

Universidad de Matanzas



Facultad de Ciencias Económicas e Informática

Ingeniería Informática



Título: Aplicación Web para el apoyo a los ajustes de planes de estudio en la Universidad de Matanzas.

Trabajo de Diploma para optar por el Título de Ingeniero en Informática.

Autor: Abel González Mondéjar.

Tutor: DrC. Julio A. Telot González.

Matanzas, Cuba

Abril, 2015

Pensamiento

“Si lo puedes imaginar, lo puedes lograr.”

Albert Einstein

Dedicatoria

A los que no están...

Agradecimientos

Esta se convierte en la página más difícil de redactar, máxime cuando han sido tantas las personas que han participado en esta larga travesía, por eso quiero agradecer a todos los que de una forma u otra contribuyeron a que este sueño se convirtiera en realidad:

Primeramente, quienes me vieron ingresar a la Universidad pero no están físicamente conmigo en este momento: mi tía Mercedita y mi papá. Seguro estoy que, dondequiera se encuentren, están muy orgullosos de mí. Te quiero, viejo.

A mi mamá y mi hermana, que sin su apoyo y comprensión durante estos 5 años, hubiera sido imposible este sueño que hoy, especialmente les dedico.

A mi sobrino Alexander, seguro estoy que será mejor que yo.

A mi familia paterna, por su ayuda cuando más lo necesité, por guiarme por el buen camino y apoyarme incondicionalmente ante cualquier decisión.

A mi tía Vivian, por ser mi segunda madre, darme las fuerzas y el optimismo necesarios para seguir adelante en la carrera.

A mi novia Silvia que ha sido y es mi motor impulsor, un faro entre tanta niebla. A su familia, por considerarme un hijo más.

A mi tutor Telot, por depositar su confianza en mí para la realización de esta investigación, por su apoyo para lograr mis objetivos.

A mis amigos y hermanos en estos cinco años: Félix, Dayana, Yuleika, Randy y Yosmely, gracias por su ayuda sincera y comprensión en todo momento, por el tiempo que compartimos juntos, los voy a extrañar.

A Félix y Alberto, vuestra ayuda fue fundamental en este trabajo, muchísimas gracias.

A la comunidad de desarrollo de Django en Stack Overflow y a Google, por sacarme del apuro en varias ocasiones.

A todas las personas que en un momento u otro me preguntaron “¿cómo va la tesis?”.

A los profesores de la Universidad y del Departamento de Informática que durante esta etapa me formaron como profesional. Espero formar parte del *staff* en breve, continuar aprendiendo de ustedes, y difundir lo aprendido en la hermosa y noble labor del magisterio.

A todos, de corazón, muchas gracias.

Declaración de autoría

Yo, Abel González Mondéjar, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, y en especial, a la Facultad de Ciencias Económicas e Informática, a que hagan el uso que estimen pertinente de él.

Para que así conste, firmo la presente a los 6 días del mes de Abril del 2015.

Firma del autor

Abel González Mondéjar

Firma del tutor

DrC. Julio A. Telot González

Opinión del cliente y tutor del Trabajo de Diploma

El Trabajo de Diploma titulado “Aplicación Web para el apoyo a los ajustes de planes de estudio en la Universidad de Matanzas” presentado por el estudiante Abel González Mondéjar, en opción al Título Académico de Ingeniero Informático y desarrollado en el Departamento de Informática de la Facultad de Ciencias Económicas e Informática de la Universidad de Matanzas, trata una temática de suma actualidad para las condiciones actuales de desarrollo de las instituciones universitarias cubanas, en particular cuando se considera la alta movilidad de estudiantes entre carreras, tipos de curso e incluso la opción de estudiar una segunda carrera a la que se acogen muchos egresados en la actualidad.

Esta aplicación contribuye a la elevación de la eficiencia y confiabilidad en el manejo y toma de decisiones en cuanto a los ajustes que se le realizan a los Planes de Estudio de los estudiantes, a la vez que mejora el registro, la organización, actualización y acceso a la información relacionada con la personalización de estos Planes para garantizar una adecuada información a todos los elementos componentes del proceso.

En el desarrollo del Trabajo de Diploma se logran satisfacer los aspectos metodológicos de la investigación científica en cuanto a forma y contenido, lo que puede ser apreciado a lo largo de todo el material presentado en el cuerpo del documento. Existe un excelente trabajo de estudio, levantamiento de requisitos, análisis, diseño e implementación que dan cumplimiento a los requisitos para obtener el título de Ingeniero Informático. Los requisitos han requerido de un fuerte trabajo por parte del estudiante, ya que son aspectos poco formalizados en la práctica diaria.

Se han desarrollado las etapas de construcción de un sistema informático que permite automatizar los aspectos relacionados con los ajustes de planes de estudio en la Educación Superior, garantiza un fácil acceso a la información, las interfaces están construidas de forma profesional y eficiente y dan respuesta a los requerimientos planteados.

En todo el trabajo se puede apreciar rigor, manifestado desde el tratamiento de los conceptos estudiados y referenciados en la bibliografía. Las Conclusiones están adecuadamente estructuradas sobre los objetivos a lograr y las Recomendaciones son objetivas para el trabajo posterior.

El trabajo tiene un nivel de terminación elevado para el nivel de pregrado al cual está dirigido. Por otra parte, ha utilizado herramientas modernas y eficientes con el fin de obtener un producto de alto nivel funcional, ergonómico y estético.

El estudiante ha mostrado una alta capacidad de trabajo independiente, de seriedad y dedicación a su trabajo, el que ha desempeñado junto con otras responsabilidades como la impartición de docencia, en la que existen muy buenos criterios de su desempeño. Su consagración al trabajo, modestia, capacidad de asimilación de nuevos contenidos, carácter y respeto en sus relaciones hacen que se vislumbre un futuro promisorio en su vida profesional.

Por todo lo anteriormente señalado, considero que el estudiante de Ingeniería Informática Abel González Mondéjar reúne sobradamente los requisitos para la obtención del Título Académico de Ingeniero Informático.

Dr. C. Julio A. Telot González

Emitida: 5 de abril de 2015

Resumen

El presente trabajo hace un estudio de la situación existente en la Universidad de Matanzas en el transcurso del proceso de ajuste de plan de estudio en el cual se debe efectuar una extensa labor en la creación y manipulación de los documentos de cada estudiante así como en el almacenamiento de las modificaciones hechas. Actualmente para solucionar este problema se emplean soluciones basadas en Microsoft Excel las cuales no proporcionan una respuesta adecuada a las demandas de información, a raíz de esto se pretende desarrollar una aplicación web que permita organizar y sistematizar el proceso de ajuste de plan de estudio en la Universidad de Matanzas, de forma tal que la experiencia de los casos existentes pueda aprovecharse y se logre una adecuada y más rápida toma de decisiones. La implementación de un sistema integral permite obtener una alternativa basada en la web, adaptada al contexto de la Universidad de Matanzas, lo que permitirá facilitar el ajuste de planes de estudio para estudiantes que cambian el mismo. Para su descripción y desarrollo fue utilizada la metodología de desarrollo ágil *eXtreme Programming* (XP), el *framework* Django 1.7.1, ambiente de programación PyCharm 4.0 con el lenguaje Python 3.4.2, Apache 2.4.9 como servidor web y MySQL 5.6.17 como sistema de gestor de base de datos. La presente aplicación web está dirigida a la Universidad de Matanzas aunque el producto final puede ser extendido y utilizado en las demás universidades del país.

Palabras clave:

Ajustes, plan de estudio, Universidad de Matanzas

Abstract

This work makes a study of the situation at the University of Matanzas during the process of adjusting curriculum in which they must perform extensive work in the creation and manipulation of documents for each student as well as the storing modifications. Currently to solve this problem based on Microsoft Excel solutions which do not provide an adequate response to the demands for information, because of this is to develop a web application that allows to organize and systematize the process of setting curriculum in used the University of Matanzas, so that the experience of existing cases can take advantage and proper and faster decision making is achieved. The implementation of a comprehensive system enables a web based alternative adapted to the context of the University of Matanzas, allowing easy adjustment of curricula for students who change it. For description and development was used agile development methodology eXtreme Programming (XP), the framework Django 1.7.1, environment PyCharm 4.0 programming language Python 3.4.2, Apache 2.4.9 and web server and MySQL 5.6.17 as system manager database. This web application is directed to the University of Matanzas although the final product can be extended and used in other universities.

Keywords: settings, curriculum, University of Matanzas

Índice

Introducción.....	1
Capítulo 1: Marco teórico referencial.....	6
1.1 Introducción	6
1.2 Antecedentes del trabajo	7
1.2.1 Ámbito internacional	7
1.2.2 Ámbito nacional:	8
1.3 Objeto de estudio	9
1.3.1 Descripción de los documentos asociados al objeto de estudio	9
1.3.2 Flujo actual de los procesos involucrados en el campo de acción.....	11
1.3.3 Análisis crítico de cómo se ejecutan actualmente estos procesos, las causas que originan la situación problemática y las consecuencias	14
1.4 Análisis comparativo de la propuesta existente con la propuesta de trabajo	15
1.5 Métodos de la investigación	16
1.5.1 Métodos teóricos empleados	16
1.5.2 Métodos empíricos empleados	16
1.6 Herramientas, tecnologías y metodologías de desarrollo	17
1.6.1 Tendencias tecnológicas a considerar.....	17
1.6.2 Metodologías de desarrollo.....	17
1.6.3 Tecnologías	20
1.7 Análisis crítico de las fuentes bibliográficas utilizadas	27
1.8 Conclusiones parciales	28
Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta	29
2.1 Introducción	29
2.2 Descripción de la solución	30
2.3 Etapa de planificación	31
2.3.1 Equipo de trabajo y roles	31
2.3.2 Historias de usuario iniciales	31
2.3.3 Planificación de iteraciones	39
2.3.4 Reuniones.....	39
2.3.5 Plan de entregas.....	40
2.4 Etapa de diseño	41

2.4.1 Prototipo de interfaz de usuario	41
2.4.2 Incidencias	42
2.4.3 Metáfora	42
2.4.4 Tareas a desarrollar	42
2.4.5 Tarjetas CRC	51
2.4.6 Modelo físico de la base de datos	54
2.5 Conclusiones parciales	55
Capítulo 3: Validación de la solución propuesta	56
3.1 Introducción	56
3.2 Pruebas al software	57
3.2.1 Pruebas de caja blanca	58
3.2.2 Pruebas de caja negra	61
3.2.3 Pruebas de aceptación	62
3.4 Análisis de los resultados obtenidos	72
3.5 Conclusiones del capítulo	73
Conclusiones generales	74
Recomendaciones	75
Bibliografía	76
Anexos	81
Anexo 1: Diferencias entre metodologías ágiles y tradicionales	81
Anexo 2: Top 39 departamentos de ciencias de la computación norteamericanos y los lenguajes de programación que emplean	74
Anexo 3: Funcionamiento del MTV de Django	75
Anexo 4: Comparación entre algunos <i>frameworks</i> en cuanto a cantidad de carpetas y ficheros	76
Anexo 5: Comparación entre algunos <i>frameworks</i> y sus licencias	77
Anexo 6: Funcionalidades nativas en algunos <i>frameworks</i>	78

Índice de tablas

Tabla 1 Equipo de trabajo y roles.....	31
Tabla 2 Historias de usuario planificadas.....	33
Tabla 3 HU-1 Diseño y creación de la Base de Datos	34
Tabla 4 HU-2 Diseño de la interfaz de usuario.....	35
Tabla 5 HU-3 Seguridad y autenticación de usuario	35
Tabla 6 HU-4 Gestionar Administración del Sitio (Backend)	36
Tabla 7 HU-5 Gestionar plan de estudio	36
Tabla 8 HU-6 Operaciones con cohortes	37
Tabla 9 HU-7 Gestionar expedientes	37
Tabla 10 HU-8 Gestionar ajustes	38
Tabla 11 HU-9 Generar reportes.....	38
Tabla 12 Tareas de ingeniería del proyecto	45
Tabla 13 HU1 TI-1 Diseño de la base de datos.....	45
Tabla 14 HU1 TI-2 Creación de la base de datos	46
Tabla 15 HU2 TI-3 Diseño de la interfaz principal.....	46
Tabla 16 HU2 TI-4 Diseño de otras interfaces	47
Tabla 17 HU3 TI-5 Gestionar usuarios.....	47
Tabla 18 HU4 TI-9 Gestionar Departamento.....	48
Tabla 19 HU7 TI-63 Listar expediente estudiantil.....	48
Tabla 20 HU7 TI-64 Buscar expediente estudiantil	49
Tabla 21 HU7 TI-65 Nuevo expediente estudiantil	49
Tabla 22 HU7 TI-67 Editar expediente estudiantil	50
Tabla 23 HU7 TI-67 Eliminar expediente estudiantil	50
Tabla 24 HU7 TI-69 Detalle de expediente estudiantil	51
Tabla 25 Tarjeta CRC – Estudiante.....	52
Tabla 26 Tarjeta CRC - Plan de estudio.....	53
Tabla 27 Casos de prueba para el método autenticar.....	61
Tabla 28 HU2 PA-2 Prueba del diseño de la interfaz de usuario	62
Tabla 29 HU7 PA-7 Prueba de la gestión de expedientes estudiantiles	66
Tabla 30 HU8 PA-8 Prueba de la gestión de ajustes a plan	69

Índice de figuras

Figura 1 Flujo de trabajo	13
Figura 2 Iteraciones del proyecto	39
Figura 3 Página principal de la aplicación	41
Figura 4 Modelo físico de la base de datos	54
Figura 5 Filosofía de prueba de caja blanca y negra.....	58
Figura 6 Método a realizar prueba de caja blanca (Autenticar)	59
Figura 7 Grafo de flujo del método Autenticar	60
Figura 8 Resultado de la prueba de aceptación HU2 PA-2 en Google Chrome 36.0.1948.0	63
Figura 9 Resultado de la prueba de aceptación HU2 PA-2 en Mozilla Firefox 35.0	63
Figura 10 Resultado de la prueba de aceptación HU2 PA-2 en Internet Explorer 11.0.9600.17031	64
Figura 11 Resultado de la prueba de aceptación HU2 PA-2 (móvil @ 250px x 500px)	65
Figura 12 Resultado de la prueba de aceptación HU2 PA-2 (tablet @ 880px x 460px)	65
Figura 13 Resultado de la prueba de aceptación HU2 PA-2 (ordenador @ 1366px x595px)	65
Figura 14 Resultados de la prueba de aceptación HU7 PA-7 (primer paso)	67
Figura 15 Resultados de la prueba de aceptación HU7 PA-7 (segundo paso)	67
Figura 16 Resultado de la prueba de aceptación HU7 PA-7 (tercer paso)	68
Figura 17 Resultados de la prueba de aceptación HU7 PA-7 (cuarto paso)	68
Figura 18 Resultados de la prueba de aceptación HU8 PA-8 (primer paso)	70
Figura 19 Resultados de la prueba de aceptación HU8 PA-8 (segundo paso)	70
Figura 20 Resultados de la prueba de aceptación HU8 PA-8 (tercer paso)	71
Figura 21 Resultados de la prueba de aceptación HU8 PA-8 (cuarto paso)	71

Introducción

El progresivo empleo de la Informática y las Comunicaciones (TIC) para la educación, ha causado los cambios que hoy se aprecian en el sistema educacional. La revolución que ha supuesto el desarrollo de las TIC, ha dotado al hombre de los instrumentos que le permiten potenciar cambios en todos los ámbitos de la actividad humana. Un salto cualitativo particularmente se ha visto, desde la introducción de estas técnicas, en el sector educativo. Las nuevas tecnologías facilitan enormemente la adquisición y absorción de conocimiento, es por ello que la Universidad, vista como la base de toda sociedad, se ha sumado a la gran transformación que significa la inserción de estos avances en el sistema educacional.

Para resolver un problema muy común por estos días, como es la gestión y recuperación de información de una manera eficiente, sin tener una estructura común que la soporte, se han creado sistemas para la gestión de documentos a través del uso de las facilidades de la informatización. Al analizar estos documentos puede encontrarse la documentación asociada a este proceso, elaborada en sistemas tales como procesadores de texto, páginas en formato HTML o incluso en alguna que otra hoja de cálculo (A. G. V. Fernández Orquín, Yoan; Pérez Martínez, Abel; García Vasconcelos, Yaniseth; Miranda, Lisandra, 2010).

Estos programas de computadoras se convierten en herramientas para ayudar a los usuarios que trabajan con documentos, independientemente del formato en que se crearon. Estas aplicaciones para la gestión de la información permiten la recuperación de forma rápida y eficiente, de archivos creados electrónicamente. En realidad existe hoy muy poco aprovechamiento en el empleo de estas formas de almacenamiento, salvo que se cuenta con versiones electrónicas de esta documentación, la que se puede imprimir o mostrar relativamente fácil, sin embargo, se pudieran obtener muchas otras ventajas de una verdadera informatización (A. Fernández Orquín, E. D. Expósito., Z. R. LLorca., Y. B. González., Gutiérrez Vasconselo, Yoan; Pérez Martínez, Abel; J. D. Blanco., L. P. Alfonso., I. G. Valiente; Pérez Chávez, Roger, 2004).

Al hacer una revisión de aplicaciones informáticas con cierta similitud al tema de investigación expuesto en la presente investigación se localizaron algunas soluciones dentro y fuera de Cuba (Alonso Rodríguez, 2014; Aluicio Sarduy, 2011; Delgado Expósito, 2006; Rodríguez Díaz, 2012; s.a, 2014a, 2014b, 2014c, 2014d; Vera León, 2008). Algunos de ellos se acercan al tema que se investiga, por otra parte hay otros que están desarrollado con tecnologías actualmente desfasadas, lo cual va en contra de los estándares que se buscan en Cuba, sin embargo, otro grupo de aplicaciones informáticas están desarrolladas con PHP que, comparándolo con Django (Aloy, 2009, 2010; Bom Malla, 2010) en velocidad de ejecución, facilidad de mantenimiento y evolución posterior, Python y Django suelen ser una mejor elección para el desarrollo de la nueva aplicación web.

Después de estudiar y analizar algunos sistemas informáticos empleados dentro y fuera de Cuba para el ajuste de plan de estudios se llegó a la conclusión de no utilizar ninguno porque no abordan específicamente el tema que se quiere investigar. A pesar de los inconvenientes detectados, es importante aprovechar toda la ardua labor realizada por otros desarrolladores que investigaron en temas similares y, posteriormente, desarrollar una aplicación que permita satisfacer las necesidades específicas del cliente.

En la Universidad de Matanzas, en el transcurso del proceso de ajuste de plan de estudio se debe efectuar una extensa labor en la creación y manipulación de los documentos de cada estudiante así como en el almacenamiento de las modificaciones hechas, provocando la **situación problemática** del presente trabajo el cual hace un estudio en la Universidad de Matanzas, donde, a pesar del desarrollo tecnológico actual, para llevar a cabo la actividad de ajuste de los planes de estudio a estudiantes que cambian de plan de estudio por disímiles razones, se emplean soluciones basadas en Microsoft Excel (o simplemente un estudio de los documentos presentados y la elaboración de un documento Word) lo que hace muy lento e inseguro su trabajo en cuanto a organización y control de los mismos, por otra parte, impide utilizar racionalmente la experiencia de casos anteriores para agilizar el trabajo de los actores implicados en esta labor. Por otra parte, el proceso

de ubicación de los estudiantes en un año determinado y la decisión de qué asignaturas debe cursar o no el estudiante al cambiar de Plan de Estudio se realiza de forma manual, basada en la experiencia personal de cada coordinador de carrera, por lo que se presentan, con frecuencia, dificultades para tomar las decisiones correspondientes.

Todo lo antes expuesto permite definir el siguiente **problema científico**: ¿cómo organizar y sistematizar el proceso de ajuste de plan de estudio en la Universidad de Matanzas, de forma tal que la experiencia de los casos existentes pueda aprovecharse y se logre una adecuada y más rápida toma de decisiones?

Los avances tecnológicos han hecho este proceso asequible, eficiente y confiable, donde la institución y sus directivos pueden ahorrar tiempo y elevar la confiabilidad en la información, a partir de las amplias posibilidades que se obtendrán con el producto final, la presente investigación posee como **objeto de estudio** la gestión de los documentos relacionados con los planes de estudio de Educación Superior y los ajustes frecuentes que se realizan a los estudiantes; se define como **campo de acción** el desarrollo y empleo de herramientas informáticas para la gestión de los planes de estudio y el **alcance del trabajo** está dirigido a la Universidad de Matanzas, aunque el producto final puede ser extendido y utilizado en las demás universidades del país, conjuntamente en la presente investigación se persigue como **objetivo general** desarrollar una aplicación web que permita organizar y sistematizar el proceso de ajuste de plan de estudio en la Universidad de Matanzas, de forma tal que la experiencia de los casos existentes pueda aprovecharse y se logre una adecuada y más rápida toma de decisiones, con los **objetivos específicos**:

- Realizar un estudio de los antecedentes y tendencias actuales de software para la gestión de planes de estudio en la Universidad de Matanzas.
- Implementar las funcionalidades que permitan realizar ajustes de planes de estudio en la Universidad de Matanzas.
- Efectuar pruebas al software desarrollado para verificar su correcto funcionamiento en la Universidad de Matanzas.

Estos objetivos específicos serán cumplidos con el desarrollo de **tareas de investigación** como:

- Consulta de bibliografía sobre las tendencias actuales de sistemas para la gestión del proceso docente y plan de estudios.
- Realización de entrevistas en profundidad a personal especializado.
- Análisis de las tendencias y tecnologías actuales para el desarrollo de sistemas informáticos gestores del proceso docente y plan de estudios.
- Descripción de los requisitos funcionales y no funcionales de la aplicación.
- Selección de la arquitectura que soporte la implementación de las funcionalidades de la aplicación web.
- Implementación de las funcionalidades que den cumplimiento a los requisitos identificados.
- Validación de la aplicación web.

Se pudo arribar a la formulación de la siguiente **hipótesis**: si se desarrolla una aplicación web para el ajuste de los planes de estudio se logrará una adecuada organización y sistematización del proceso de ajuste de plan de estudio en la Universidad de Matanzas.

Se define como **variable independiente**: el software y como **variable dependiente**: optimizar el trabajo, proporcionar seguridad y agilizar el proceso de ajuste de plan de estudio.

Visto lo anterior entre los **resultados esperados** es válido recalcar que la implementación de un software permitirá obtener una alternativa basada en la web, adaptada al contexto de la Universidad de Matanzas, lo que permitirá facilitar el ajuste de planes de estudio para estudiantes que cambian el mismo. La aplicación web será multiplataforma, optando por alternativas de Software Libre, por otra parte, permitirá realizar de forma muy fácil e intuitiva el proceso de ajuste de plan de estudios y agilizar la toma de decisiones por parte del personal especializado, así como garantizar una ubicación más certera del año académico que cursará el estudiante.

La investigación se estructuró en tres capítulos, como se indica a continuación:

Capítulo 1: Marco teórico referencial.

Recoge los conceptos fundamentales asociados al dominio del problema. Realiza un estudio de las tendencias y tecnologías actuales sobre las que se apoya la propuesta y analiza las soluciones existentes.

Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta.

Presenta una propuesta de solución para el sistema, donde se describen las reglas y los elementos del negocio, una planificación inicial del proyecto con el empleo de la metodología de Ingeniería del Software *Extreme Programming* (XP). La solución que se propone se basa en el análisis y diseño del sistema; tomando en cuenta los intereses originados por el cliente, los cuales se describen mediante las Historias de Usuario.

Capítulo 3: Validación de la solución propuesta.

Se hace el análisis de los resultados obtenidos. También se realizan las pruebas al *software* con el objetivo de entregarle al cliente un producto totalmente funcional, cumpliendo con todos los requisitos demandados por el mismo y satisfaciendo sus necesidades.

Durante la investigación se emplearon diversos **métodos de investigación científica**. Entre los **métodos teóricos** se utilizaron:

- Histórico-lógico.
- Analítico-sintético.
- Inducción-deducción.
- Hipotético-deductivo.
- Enfoque de sistema.

Por otra parte, los **métodos empíricos** analizados fueron los de:

- Observación.
- Entrevista.
- Análisis de documentos.

Capítulo 1: Marco teórico referencial

1.1 Introducción

Las universidades cubanas se encuentran inmersas en un proceso de avanzada en el perfeccionamiento y desarrollo de todas las esferas en las que interviene. El Plan de Estudio (PE), y específicamente la gestión de sus documentos rectores constituyen una de las tareas más importantes a tener en cuenta, pues de ella dependen muchos de los factores asociados a los objetivos perseguidos. No obstante, estos procesos se encuentran aún al margen de las más novedosas técnicas de Informatización, debido tal vez a la complejidad con que se desarrollan.

Es precisamente a la utilización de las TIC, con aras de automatizar y perfeccionar el PE, que se encuentra encaminada esta investigación, se exponen en este capítulo las bases teóricas que la sustentan. Además, se definen los principales conceptos relacionados con el objeto de estudio y se describen los elementos seleccionados para dar respuesta a la problemática planteada. Se realiza un análisis del estado del arte, a partir de sistemas existentes vinculados al campo de acción y las tendencias y tecnologías sobre las que se basa la propuesta a implementar.

1.2 Antecedentes del trabajo

Para la realización de esta investigación se tuvieron en cuenta experiencias similares en el empleo del motor de búsqueda Google (www.google.com), por otra parte, se analizó la web nacional con el objetivo de investigar en otras universidades la existencia de aplicaciones informáticas que guardaran semejanza con la idea que se propone desarrollar.

1.2.1 Ámbito internacional:

Al emplear Google, se encontraron propuestas que guardan, hasta cierto punto, semejanza con la propuesta de investigación.

Plataforma Académica Web: según su sitio web, es un sistema de información centralizado en Internet que ofrece la Universidad Católica de Colombia a sus estudiantes, egresados, profesores y personal administrativo, para que puedan realizar las labores académicas propias de su actividad. (s.a, 2014b)

SIU-Guaraní: informa su página principal que es un sistema de gestión de alumnos en el plano internacional que registra y administra todas las actividades académicas de la universidad desde que los alumnos ingresan como aspirantes hasta que obtienen el diploma. (s.a, 2014d)

GESTACAD: provee su portal que es una aplicación web desarrollada para ser implementada en instituciones educativas de nivel básico, bachillerato, técnico y tecnológico. Tiene vistas de acceso diferenciados para los diferentes tipos de usuario: rectores, secretarías, contadoras/colectoras, docentes, estudiantes y representantes (s.a, 2014a).

1.2.2 Ámbito nacional:

SIGENU: consta de dos tipos de aplicaciones: un sistema transaccional y un sistema para la toma de decisiones. Entre las funcionalidades principales del sistema de gestión se encuentra la inscripción de un estudiante, registro de asignaturas a cursar, registro de evaluaciones, control de bajas, emisión de reportes oficiales entre otros. (s.a, 2014c).

SIPRID: las dos versiones de esta aplicación web (Delgado Expósito, 2006; Vera León, 2008) se implementaron con el objetivo de perfeccionar el procesamiento, gestión y análisis de la documentación asociada al proceso docente-estudiantil.

Asistente para la Gestión de la Documentación Docente en las Universidades: Según Aluicio Sarduy (2011, p. 4), explica que el software genera automáticamente la planificación del Proceso Docente y del Gráfico Docente la cual ayudaría en gran medida a la labor de organización y control del proceso.

Módulo Informático para la Gestión de la Información Académica de los Estudiantes: este módulo, según su autora, Saybelis Rodríguez (2012, p. 10), efectúa la gestión de la información académica del estudiante. Se apoya en la implementación de un servicio web posibilita la comunicación entre diversos sistemas para consumir y brindar información.

Sistema informático para la gestión de expedientes electrónicos: expone Yunier Alonso (2014, p. 7), que este software se utiliza para la conformación de los expedientes estudiantiles, además de constituir un medio para facilitar la creación, modificación y almacenamiento de la gran cantidad de información referente al proceso docente.

1.3 Objeto de estudio

El objeto de estudio de esta investigación se enmarca en la gestión de los documentos relacionados con los planes de estudio de Educación Superior y los ajustes frecuentes que se realizan a los estudiantes, por tanto, se hace necesario hacer una descripción más exacta de este elemento que se pretende transformar y perfeccionar.

1.3.1 Descripción de los documentos asociados al objeto de estudio

Para una mejor comprensión del problema es necesario dominar los términos **aplicación, web, apoyar y ajustar**, los que se pueden considerar como intuitivos para los especialistas que manejan estas informaciones y procedimientos. Es válido aclarar un conjunto completo de características de aplicación web, propuestos por Roger Pressman (2009, pp. 4-7) los cuales hacen la diferencia a los sistemas basados en la Web de los sistemas más convencionales de basados en ordenadores.

A partir de la relevancia que posee en este trabajo el concepto de **Plan de Estudio** (PE), se acepta la definición del DRAE (Española, 2014b) el cual lo define como “un conjunto de enseñanzas y prácticas que, con determinada disposición, han de cursarse para cumplir un ciclo de estudios u obtener un título”.

Se ha podido comprobar que el PE, según el Reglamento para el Trabajo Docente y Metodológico en la Educación Superior (2007) en su artículo 21 expresa que “es el documento fundamental de carácter estatal que establece la dirección general y el contenido principal de la preparación del profesional. Está integrado por: el Modelo del Profesional, el Plan del Proceso Docente y los Programas de Disciplina (...). El plan de estudio se elabora para cada carrera en correspondencia con las necesidades sociales existentes en el país, los avances científicos y tecnológicos de la época actual y las particularidades de la profesión en cuestión.”

Cohorte: Constituye una versión de un plan de estudios para un curso específico.

Expediente estudiantil: Constituye el documento oficial donde queda registrado la información personal y docente de un estudiante en un Centro de Educación Superior (CES).

Ajuste a Plan: es “el proceso de acomodo de las asignaturas de un estudiante cuando es traslado interno o externo de un Centro de Estudio” (Superior, 2007)

Asignatura eximida o abonada: “se denomina asignatura eximida cuando por determinadas circunstancias convenidas o establecidas legalmente, se autoriza al estudiante a no cursarla. (...)” (Artículo 31, Superior (2010)).

Asignatura aprobada por suficiencia: “se denomina asignatura aprobada por suficiencia cuando el estudiante sin haberla cursado, demuestra a través de un examen que domina los contenidos esenciales y satisface los objetivos generales del programa de la misma. (...)” (Artículo 30, Superior (2010)).

Asignatura convalidada: “se denomina asignatura convalidada siempre que exista correspondencia entre los objetivos generales y contenidos esenciales del programa analítico vigente con respecto al programa de la asignatura objeto de análisis. Los contenidos esenciales deben tener una correspondencia de, al menos, el 80% (...)”. (Artículo 29, Superior (2010)).

Asignatura optativa: “(...) es aquella asignatura que se incluyen en el Plan de Estudio y de entre las cuales el estudiante selecciona una cantidad determinada para cursar en forma obligatoria. (...)” (Artículo 70, Superior (2007)).

Asignatura electiva: “(...) es aquella asignatura que el estudiante elige libremente de acuerdo con sus gustos e intereses personales, a partir de un grupo de ofertas que se brindan y que pueden, inclusive, pertenecer a otras carreras. (...)” (Artículo 71, Superior (2007)).

1.3.2 Flujo actual de los procesos involucrados en el campo de acción

Una de las tareas de mayor importancia dentro del Proceso Docente Educativo (PDE) la constituye la generación y modificación de documentos rectores, tales como Planes de Estudios.

Una característica importante en la generación de este tipo de documentos consiste en que las comisiones nacionales de carreras no elaboran todo el currículo. Su labor se centra fundamentalmente en aquellos aspectos del contenido que son necesarios para darle cumplimiento a los objetivos propuestos para esta carrera, por lo que no abordan aspectos secundarios, que pudieran variar de una universidad a otra (Horruitiner Silva, 2006; Superior, 2007).

Como se plantea en el artículo 22 del Reglamento del Ministerio de Educación Superior (2007), la Comisión Nacional de Carrera determina el currículo base, con lo que aseguran un enfoque estratégico de la misma y el dominio esencial del modo de actuación del profesional. Una vez elaborados, los planes y programas de estudio son informados a los centros de educación superior de todo el país, los que reciben estos documentos de forma de forma electrónica o impresa.

A partir de aquí cada universidad, en correspondencia con sus características decide el modo de completar su PE.

Lo planteado anteriormente no excluye el perfeccionamiento de estos planes y programas de estudio. Esta tarea es responsabilidad de las Comisiones Nacionales de Carrera, en correspondencia con lo establecido en las reglamentaciones preliminares.

Cada nuevo curso constituye una nueva oportunidad de definir intereses locales por parte de los directivos los cuales informan a las universidades sus inquietudes y necesidades de intereses a largo plazo, tomando en cuenta sus criterios, se elaboran las cohortes, las cuales constituyen una versión del plan de estudio, respetando su estructura original, pero flexibilizando en función de las necesidades de planificación y disponibilidad de profesores en un momento dado, se planifica

pues, cada curso, una nueva cohorte para cada carrera, para lo que se realizan las variaciones pertinentes.

Planteado lo anterior, cuando un estudiante resulta traslado del otro CES o bien, por alguna razón, abandona la carrera que estuvo cursando y se reincorpora posteriormente, se precisa hacerle un ajuste a plan, en función a sus intereses de continuación de estudios.

Para realizar este proceso, se requiere, inicialmente, que la secretaría del centro donde egresa el estudiante emita la información acerca de las asignaturas que el estudiante venció, sus resultados docentes y el plan temático para cada asignatura.

Al llegar el estudiante al centro donde desea ingresar, le entrega su expediente estudiantil al (la) secretario(a) docente de la Facultad de la carrera a la que aspira, la misma le conforma el expediente estudiantil, dándole así el alta en la nueva carrera. El expediente creado con vistas a realizarle posteriormente un ajuste a plan, le es enviado al coordinador de carrera a la que aspira, el mismo verifica y aprueba el expediente enviado por el (la) secretario(a), en caso de haber algún error alguna duda devuelve el mismo.

Una vez que el expediente estudiantil es aprobado por el coordinador el mismo procede a realizar el ajuste a plan, verificando las asignaturas que trae el estudiante, su plan temático, y determina si la asignatura la tiene abonada, convalidada o pendiente. Al final determina en que año y en que semestre comenzará el estudiante en cuestión. Luego de terminar el ajuste a plan, procede a enviar la propuesta de ajuste al (la) secretario(a) docente.

El (la) secretario(a) docente recibe por parte del coordinador de carrera la propuesta de ajuste y se la envía al decano, el mismo determina si aprueba la propuesta hecha por el coordinador, en caso de no estar conforme, procede a devolver dicha propuesta notificando al (la) secretario (a), la misma informa al coordinador de la negativa por parte del decano y procede a hacer una nueva propuesta de ajuste, le envía al (la) secretario(a) y si el decano la aprueba, el (la) secretario (a) docente le entrega al estudiante el ajuste a plan realizado.

El proceso antes expuesto se grafica en la Figura 1, con el objetivo de tener una mejor comprensión del negocio.

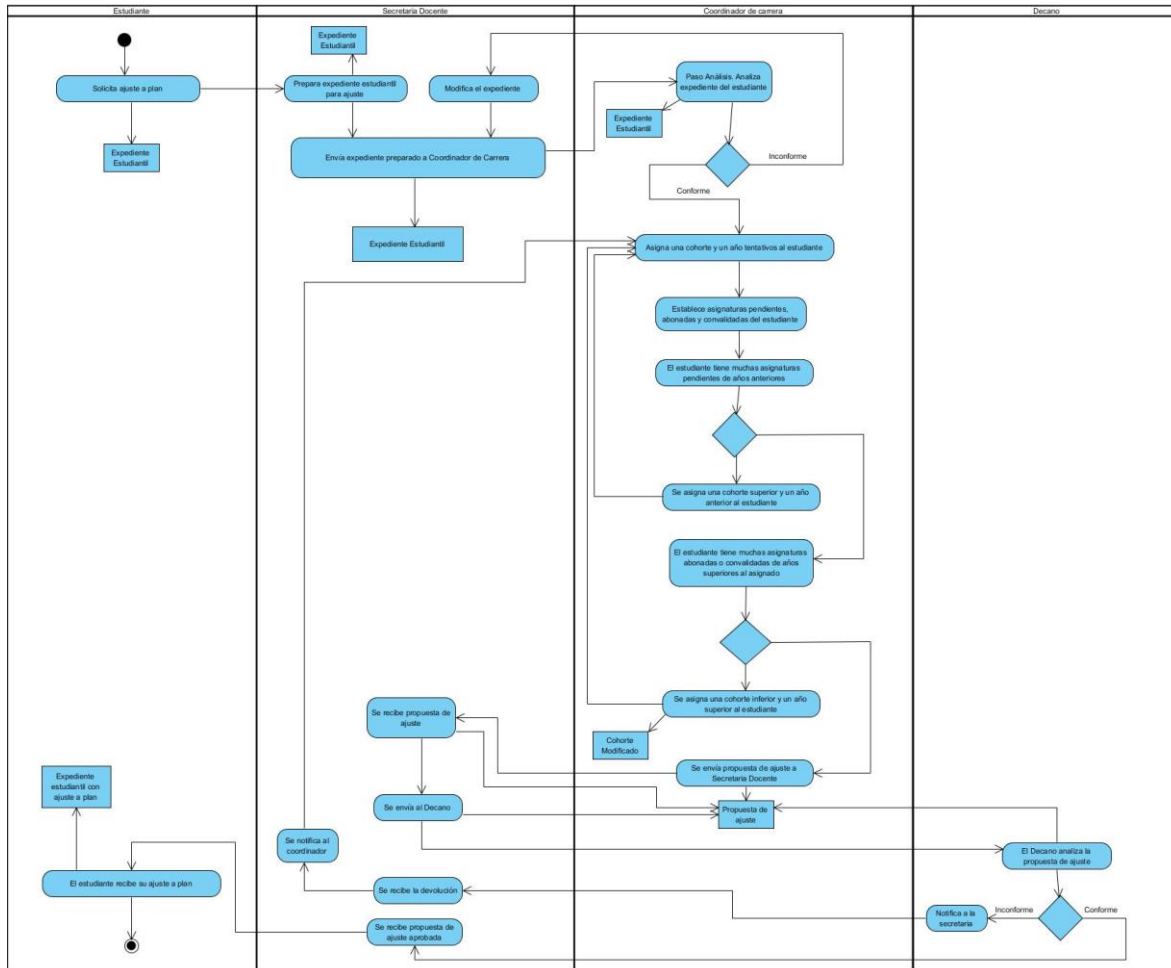


Figura 1 Flujo de trabajo

1.3.3 Análisis crítico de cómo se ejecutan actualmente estos procesos, las causas que originan la situación problemática y las consecuencias

Según la investigación realizada al personal que intervienen en el proceso de realizar los ajustes a plan de estudio se pudo conocer que no explotaban las facilidades que brindan las TIC haciendo un pobre uso de ella, a raíz de esto, esta tarea se realiza de forma manual, convirtiéndose así en un proceso difícil, en el que se pierde tiempo y esfuerzo por parte de los directivos. Es importante recalcar que a veces el proceso de ajuste a plan se realiza en papel, corriendo el riesgo de perder la información.

A raíz de esta situación, se comenzó a trabajar, inicialmente, con Departamento de Informática, logrando varios pasos de avances con la obtención de las primeras versión del Sistema SIPRID (A. Fernández Orquín, E. D. Expósito., Z. R. LLorca., Y. B. González., Gutiérrez Vasconselo, Yoan; Pérez Martínez, Abel; J. D. Blanco., L. P. Alfonso., I. G. Valiente; Pérez Chávez, Roger, 2004). No obstante, debido a la necesidad de incorporarle varios elementos necesarios para su adecuada implantación, no se cuenta aún con un sistema funcional que automatice de forma eficiente este proceso.

Uno de los problemas que aún se enfrentan, consiste en que los documentos para el ajuste a plan se elaboran en procesadores de textos y documentos en Excel lo que hace muy lento e inseguro el trabajo en cuanto a organización y control de los mismos, por otra parte, impide utilizar racionalmente la experiencia de casos anteriores para agilizar el trabajo de los actores implicados en esta labor.

Por lo anteriormente expuesto y por la investigación y análisis de esta problemática en otras universidades cubanas por el autor del trabajo y el tutor, queda demostrada la falta de una aplicación web que permita la solución eficiente de la problemática existente.

1.4 Análisis comparativo de la propuesta existente con la propuesta de trabajo

Para la realización del presente trabajo se han tenido en cuenta investigaciones precedentes relacionadas directa o indirectamente con el objeto de estudio y el campo de acción y se han tomado los elementos que se considera que constituyen brechas o insuficiencias que deben ser resueltas.

En la investigación se obtuvo que la **Plataforma Académica Web, SIU-Guaraní, Módulo informático para la Gestión de la Información Académica de los estudiantes y Sistema informático para la gestión de expedientes electrónicos**, no gestionan nada referente a ajustes del plan de estudios. Por otra parte, **GESTACAD** dejó de utilizarse por orientación del organismo superior hace algún tiempo, **SIGENU** cuenta con varios módulos entre los que se encuentra el PE, más, la persona que tenga que procesar esta información debe introducir el ajuste previamente hecho, por lo que no resulta una variante que resuelva el problema de esta investigación.

SIPRID y Asistente para la Gestión de la Documentación Docente en las Universidades dieron varios pasos de avances en este sentido, no obstante quedaron pendientes varios elementos imprescindibles para su adecuada funcionalidad. Estas aplicaciones ofrecen una velocidad de respuesta que hace muy difícil el trabajo y el acceso a través de la red. Otro punto negativo a señalar es que no manipula de forma eficiente la gestión de plan de estudios.

En la búsqueda bibliográfica no se encontraron otras aplicaciones informáticas vinculadas estrechamente al campo de acción de esta investigación.

1.5 Métodos de la investigación

1.5.1 Métodos teóricos empleados

- **Histórico-lógico:** Permitió desentrañar la historicidad del problema a solucionar y cómo funciona el ajuste a plan, su definición, sus características, evolución y desarrollo.
- **Analítico-sintético:** Este se precisó durante la revisión bibliográfica, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
- **Hipotético-deductivo:** Desempeñó un papel esencial en el proceso de verificación del supuesto de partida. A partir de este supuesto se pueden inferir conclusiones y establecer predicciones.
- **Enfoque de sistema:** Proporcionó la orientación general del estudio como una realidad integral formada por elementos que interaccionan unos con otros.

1.5.2 Métodos empíricos empleados

- **Observación:** Se observó cómo se llevaba a cabo el ajuste de plan de estudio y las dificultades que ello conlleva, lo que condujo a la realización de la aplicación web.
- **Entrevista:** Permitió conocer cuáles eran las inquietudes y sugerencias por parte del personal especializado en el tema y qué hacer facilitar y agilizar su trabajo.
- **Análisis de documentos:** Incluye la revisión de documentos como son los reglamentos y disposiciones vigentes, así como trabajos de diploma relacionados con el tema. También se consultaron libros y diversos artículos de bancos de información de Internet relacionados con el tema a investigar.

1.6 Herramientas, tecnologías y metodologías de desarrollo

Para desarrollar la investigación, se hace necesario el estudio de las herramientas, tecnologías y metodologías de desarrollo con la finalidad de darle cumplimiento a objetivo general. A continuación se describen aspectos a tener en cuenta.

1.6.1 Tendencias tecnológicas a considerar

Arquitectura cliente-servidor: esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un Sistema Gestor de Base de Datos (SGBD). Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. (M. A. Álvarez, 2007)

Software libre: se define que el Software Libre (SL) (Española, 2014d) son programas de ordenador que pueden ser utilizados, copiados, modificados y redistribuidos libremente por sus usuarios.

1.6.2 Metodologías de desarrollo

Según Andrew Phillips (2010) las metodologías de software tienen que ver con el proceso de creación de software, no tanto el aspecto técnico, pero sí los aspectos organizativos. Una metodología debe ser adaptable para poder aplicarse en distintos proyectos, sencilla para que no resulte muy engorrosa su utilización y a la vez sea completa como para que su utilización por parte de un equipo sea provechosa.

S.D Amaro y S.D Valdeverde (2007) establecen que las metodologías de desarrollo de software (MDS) proveen una dirección a seguir para la correcta aplicación de los demás elementos R.A Ccoica (2006) explica que las Metodologías de Desarrollo de

Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software.

Actualmente existen dos grandes grupos de metodologías de desarrollo, las **metodologías tradicionales** y las **metodologías ágiles**; las primeras se centran en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, mientras que las segundas dan mayor importancia a la capacidad de respuesta a los cambios. A continuación se presenta una breve comparación entre ellas.

Metodologías ágiles: dan mayor valor a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar *software* que funcionen, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan.

Metodologías tradicionales: llevan un control estricto del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto.

Después de realizar una comparación entre las metodologías ágiles y las tradicionales en el Anexo 1 se observa que las ágiles se adecuan más para guiar el desarrollo del software propuesto, principalmente porque el proyecto no es altamente complejo, se cuenta con pocos roles y no se dispone de mucho tiempo para su puesta en marcha.

Entre las metodologías ágiles más conocidas se encuentran: *Scrum*, *Crystal Methodologies* y *Extreme Programming* (XP).

Programación Extrema (en inglés *Extreme Programming*): está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y propicia 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. La programación extrema (XP) fue diseñada para entregar el software que el cliente necesita, en el momento que lo necesita, además promueve el uso de prácticas para aumentar la productividad del equipo de desarrollo y mejorar la adaptabilidad a los frecuentes cambios dentro del ciclo de vida del proyecto, es decir, para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Justificación para su uso en el proyecto:

Para el desarrollo de este proyecto se ha decidido utilizar como metodología de desarrollo XP para darle solución a la propuesta, pues, por estar dentro del grupo de metodologías ágiles, se caracteriza por su sencillez tanto en su aprendizaje, como en su aplicación. Es adecuada para cualquier tipo de proyecto y para equipos de cualquier tamaño. XP es de todas las metodologías ágiles citadas, con la que más se ha trabajado y con la que más experiencia se ha adquirido por parte del autor de este trabajo.

Las metodologías pesadas en equipos pequeños exigen un mayor esfuerzo para su adaptación, por lo que su uso conlleva un gasto de tiempo innecesario. Además, XP cuenta con sus propios artefactos, que son generados ágilmente, con menos protocolo y son iguales de exhaustivos. XP es un método estupendo para equipos extremadamente pequeños que se centran en un solo cliente.

1.6.3 Tecnologías

Lenguajes de programación web: existen, en la actualidad distintos lenguajes de programación para la Web, principalmente se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente - Servidor.

El primer grupo está formado por los lenguajes que se ejecutan en el **lado cliente**: HTML, JavaScript y CSS, todos encargados de aportar dinamismo a la aplicación en los navegadores.

También existe un segundo grupo de lenguajes que se ejecutan en el **lado servidor** entre ellos se encuentran ASP, PHP, Java y Python, estos se caracterizan por desarrollar la lógica de negocio, además de ser los encargados del acceso a Bases de Datos y el tratamiento de la información. Las características de los lenguajes empleados se muestran a continuación.

Del lado del cliente:

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon las siguientes tecnologías:

HTML5: es la actualización de HTML, el lenguaje en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript (Freddy Vega, 2011).

Por otra parte Collins (2012) plantea que HTML5 promete cambiar la forma en que las aplicaciones web se desarrollan a través de su abanico de funcionalidades de lado del cliente incluyendo hojas de estilos de código (CSS, por sus siglas en inglés) y otras características compatibles con los navegadores actuales.

CSS3: es la última versión de CSS. El término “CSS3” no es solo una referencia a nuevas características en CSS, pero si al tercer nivel en el progreso de las especificaciones de CSS. Contiene todo lo incluido en CSS2.1 (la versión anterior) y añade nuevas características para ayudar a resolver a los desarrolladores de

aplicaciones web numerosos problemas sin la ayuda de marcadores semánticos, scripts complejos o imágenes extra. Las nuevas características en CSS3 incluyen soporte adicional para selectores, sombras, esquinas redondeadas, múltiples fondos, animaciones, transparencia y mucho más. (Goldstein, 2011, p. 5)

Javascript: es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Flanagan (2006)

Por otra parte, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Eguíluz Pérez (2014, p. Cap. 1)

Del lado servidor:

Python: Plantea Guido (Van Rossum, 2014) que Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Justificación para su uso en el proyecto:

En la actualidad existen varios lenguajes de programación que son empleados en disímiles ámbitos, entre ellos C#, y Java, por mencionar algunos, basada en la experiencia del autor de esta investigación, se decide emplear Python dada la experiencia en el trabajo con este lenguaje. Por otra parte, expresa Philip Guo (2014) que Python es actualmente el lenguaje más popular para la enseñanza en cursos introductorios de ciencias de la computación mejor calificado en los

departamentos estadounidenses. En el Anexo 2 se puede apreciar una comparativa con otros lenguajes de programación que aseveran la anterior proposición.

Frameworks: un *framework* constituye una estructura de soporte definida (o plataforma de desarrollo enfocada al entorno web) en la cual otro proyecto de *software* puede ser organizado y desarrollado. (Alcalde, 2013).

Django: Django es un entorno de desarrollo web escrito en Python que fomenta el desarrollo rápido y el diseño limpio y pragmático. De código abierto, escrito en Python, permite desarrollar aplicaciones web más rápido y con menos código. Django fue inicialmente desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de *World Online*, más tarde se liberó bajo licencia BSD. Django se centra en automatizar todo lo posible y se adhiere al principio DRY (*Don't Repeat Yourself*). (Felipe, 2013; Picca, 2014)

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes y el desarrollo rápido. Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos. (García Fernández, 2013)

Al tratarse de Python, Django permite que el desarrollador escriba código ágilmente. El resultado son menos líneas de código y, en consecuencia, menos probabilidades de que haya *bugs*. De ahí que se dice que “fomenta el desarrollo rápido y el diseño limpio y pragmático”. (Rodríguez-Guerra, 2014)

Plantea Sergio Infante (2012) que Django es un *framework* MTV (una modificación de MVC), esto se debe a que los desarrolladores no tuvieron la intención de seguir algún patrón de desarrollo, sino hacer el *framework* lo más funcional posible.

En el Anexo 3 se puede ver el funcionamiento del MTV de Django

Justificación para su uso en el proyecto:

Basado en la experiencia del autor principal de esta investigación con otros *frameworks* web tales como CodeIgniter, de PHP y OpenXava, de Java, se decide emplear Django como *framework* de desarrollo web a raíz del análisis realizado por Andrés García (2013), Juan Felipe (2013) y la propia experiencia del autor de este trabajo pues este *framework*, entre sus ventajas se encuentran que:

- Es de código abierto.
- Trabaja con el patrón MVC (Modelo Vista Controlador) o específicamente con el MVT (Modelo Vista Template).
- Es mejorado casi diariamente.
- Existe, por parte de la comunidad de desarrolladores, asegurarse que Django ahorre tiempo al desarrollar.
- Las aplicaciones son fáciles de mantener y rinden bajo mucha carga.
- Respeto el principio DRY (*Don't Repeat Yourself*).
- Provee Url's limpias y amigables.
- Los módulos (apps) son poco dependientes.

Entre los sitios webs que utilizan o utilizaron el *framework* Django se encuentran: Google, Yahoo!, Discovery Communication, Instagram, Pinterest, National Geographic, Walt Disney, The New York Times, Mozilla Foundation.

En el Anexo 4 se comparan Django y otros *frameworks* con sus ficheros y carpetas, se puede apreciar claramente la simplicidad de éste. Por otra parte, en el Anexo 5 se comparan los *frameworks* a partir de sus licencias. Se aprecia en el

Anexo 6 una comparativa de funcionalidades nativas que posee Django.

Los anteriores puntos proveen un sustento lo suficientemente fuerte para decantar la solución del proyecto con el empleo de Django.

Sistemas de Gestión de Base de Datos (SGBD): los Sistemas Gestores de Bases de Datos (SGBD, en inglés DBMS: *DataBase Management System*) son un tipo de software que permiten la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. (Marcano, 2009)

MySQL: es un gestor de base de datos sencillo de usar e increíblemente rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos posibilitando realizar múltiples y rápidas consultas. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. Es gratis para aplicaciones no comerciales. Dentro de las principales características se encuentra, la gran portabilidad entre sistemas, soporta gran cantidad de tipos de datos para las columnas y hasta 32 índices por tabla, manteniendo un buen nivel de seguridad pues gestiona los datos de los usuarios y contraseñas, dispone de API's en gran cantidad de lenguajes (Python, C, C++, Java, PHP) y aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo. (Martínez Peco, 2014).

Justificación para su uso en el proyecto:

En la actualidad, varios son los autores que evidencian la factibilidad de MySQL sobre otros SGBD (Martínez Peco, 2014; Mikoluk, 2013), se refieren que MySQL es el sistema de base de datos más popular de los que actualmente están en la red. Coinciden en la velocidad de MySQL sobre PostgreSQL.

Por otra parte, el factor principal de la popularidad de MySQL es su uso establecido en gran cantidad de proyectos, su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema, las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.

Basado en la experiencia personal del autor de este trabajo se decide emplear este SGBD para el desarrollo de la aplicación web.

Servidor web: un servidor web o servidor HTTP es un programa que, utilizando el modelo cliente-servidor y el protocolo HTTP, procesa una aplicación del lado del servidor, a través de conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y genera o cede una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. (Lingan, 2007)

HTTP Apache: es generalmente reconocido como el servidor web más popular del mundo (servidor HTTP). El servidor Web Apache proporciona una gama completa de características de servidor Web, incluyendo CGI, SSL y dominios virtuales. Apache también es compatible con los módulos de plug-in para la extensibilidad. Apache es el software libre, distribuido por la *Apache Software Foundation* que promueve diversas tecnologías Web avanzadas libres y de código abierto. (Mitchell, 2014)

Justificación para su uso en el proyecto:

El servidor HTTP Apache es, de los servidores web, con el que más se ha trabajado y con el que más experiencia se ha adquirido por parte del autor del presente trabajo, además de ser un servidor de código libre. Su robustez y gran lista de prestaciones justifican su selección como servidor en la solución propuesta de la plataforma Web. Además, Apache junto con Python y MySQL forman un buen equipo para servir páginas Web con contenido dinámico en las que la velocidad y el número de accesos concurrentes sean algo primordial, como es el caso de la aplicación web que se quiere implementar. Para la fase de desarrollo se empleará el servidor de desarrollo que provee Python, posteriormente se utilizará Apache como servidor web al publicar el sitio.

Entorno de desarrollo integrado (IDE): un entorno integrado de desarrollo (EDI, en inglés IDE: *Integrated Development Environment*) es ambiente de programación que ha sido empacado en un programa, típicamente consiste en un editor de código, un compilador, un debugger, asistentes, colores a la hora de codificar y una interfaz gráfica para el usuario (GUI, por sus siglas en inglés). El IDE puede ser una aplicación independiente o puede ser incluido como la parte de una o varias aplicaciones existentes y compatibles.

Los IDEs proveen a los desarrolladores un *framework* amigable para muchos lenguajes de programación modernos como C#, PHP o Python. (Nawrocki, 2013; Rose, 2007).

PyCharm: al igual que otros IDE de JetBrains, PyCharm incluye un editor de código inteligente que entiende las capacidades específicas de Python y ofrece destacados impulsores de productividad, formateo de código automático, finalización del código, navegación de código de un solo clic, y más. PyCharm hereda naturalmente sus funciones de edición JavaScript, HTML y CSS que todos los desarrolladores web apreciarán. Respaldadas por rutinas de análisis de código avanzadas, estas funciones hacen de PyCharm una eficaz herramienta en manos de los desarrolladores profesionales de Python y los que acaban de iniciarse en la tecnología. (Oreshnikova, 2010)

Justificación para su uso en el proyecto:

A juzgar por las ventajas que nos ofrece este IDE sobre los demás según Andrew Brookins (2014) y la opinión personal del autor de este trabajo se elige PyCharm para el desarrollo de la aplicación web.

1.7 Análisis crítico de las fuentes bibliográficas utilizadas

La bibliografía utilizada se obtuvo en su mayoría de los motores de búsqueda de internet: Bing (<http://www.bing.com>) y Google (<https://www.google.com.cu>), así como trabajos de diploma que fueron elaborados por estudiantes graduados en cursos anteriores.

La bibliografía en idioma extranjero fue traducida empleando Google Translator (<https://translate.google.com/>) y corregida por el autor de este trabajo.

La mayoría de la bibliografía consultada data de períodos actuales de no más de cinco años de su última actualización (salvo algunos documentos rectores) puesto que las tecnologías utilizadas se consideran entre las más actuales en el campo de la informática.

1.8 Conclusiones parciales

Con la elaboración de este capítulo quedaron sentadas las bases teóricas que sustentan esta investigación, a partir del estudio de los antecedentes y tendencias actuales de software para la gestión de planes de estudio en la Universidad de Matanzas y se ha llegado a las siguientes conclusiones:

- El problema planteado debe ser resuelto con un adecuado uso de las TIC.
- En la bibliografía consultada no se encontró ninguna aplicación que dé solución al problema planteado.
- Se demuestra la necesidad de la utilización de software libre como una tendencia a asumir.
- La implementación del software será posible con la utilización del lenguaje de programación Python 3.4.1, el framework Django 1.7.1, Apache 2.4.9 como servidor web y MySql 5.6.17 como sistema de gestor de base de datos.
- En sentido general se ha contribuido a la mejor comprensión del objeto de estudio y se han establecido las bases para las siguientes fases de la investigación.

Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta

2.1 Introducción

Desarrollar una aplicación web con calidad, ajustada a las necesidades del cliente es uno de los principales aspectos que se persigue en esta investigación. En este capítulo se describirá la solución propuesta, a partir del análisis de los requerimientos del software. Se expondrá el inicio y evolución del proyecto de desarrollo, para lo cual se utiliza la metodología XP con el objetivo de garantizar el diseño de un programa lo más ajustado posible, con la ventaja de la incorporación del cliente como un miembro del equipo de desarrollo.

2.2 Descripción de la solución

Se propone el desarrollo de una aplicación web que tenga como principio la sistematización del proceso de ajuste de plan de estudio en la Universidad de Matanzas de forma tal que la experiencia de los datos existentes pueda aprovecharse y se logre una adecuada y más rápida toma de decisiones.

A continuación se mencionarán las funcionalidades a las cuales podrán tener acceso cada uno de los usuarios que interactúan con el sistema:

Secretaria docente: se encarga de procesar la información de los expedientes estudiantiles y las asignaturas vencidas de los estudiantes.

Coordinador de carrera: se encarga realizar los ajustes plan.

Decano: se encarga de aprobar los ajustes a plan.

Administrador de sistema: tiene la posibilidad de asignar permisos, crear los usuarios y configurar la aplicación.

Las opciones para trabajar con la aplicación web se habilitarán de acuerdo al perfil del usuario que ingrese.

Para garantizar la seguridad y la confiabilidad en la información que se procesa es muy importante implementar una aplicación web que permita la autenticación para identificar los usuarios que utilizan el programa y por tanto incluirle un mecanismo para almacenar toda la actividad del usuario en el sistema.

2.3 Etapa de planificación

La etapa de planificación es la etapa inicial de todo el proyecto. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. Por otra parte se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto.

2.3.1 Equipo de trabajo y roles

La metodología XP exige que el desarrollo se realice en parejas para agregar calidad al código que se escribe, en este caso se tomará como principal fuente de aporte de ideas todas aquellas propuestas hechas por el cliente en cada una de las pruebas, haciendo necesario adaptar la metodología a un sólo programador. Se trabajan cuarenta horas semanales, plasmando el principio de que horas extras no aumentan la productividad sino que desmotivan al equipo, programadores cansados son más propensos a cometer errores y son menos productivos.

El equipo de trabajo está formado por un programador y el jefe de proyecto el cual es cliente según se muestra en la Tabla 1.

Miembros	Roles
Julio Telot González	Jefe de proyecto, Cliente, Tester
Abel González Mondéjar	Programador, Tester

Tabla 1 Equipo de trabajo y roles

2.3.2 Historias de usuario iniciales

En las historias de usuarios (HU) se especifican los requisitos que debe cumplir el software. El cliente describe las características que a su entender debe tener el sistema. Se logra un acuerdo entre el programador y el usuario sobre lo que se quiere y como debe lograrse.

Escalas equivalentes a la prioridad en el negocio:

Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.

Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

El resumen de historias de usuario iniciales que se muestra en la Tabla 2, y cubre todos los requerimientos que fueron planteados por el cliente, aunque la planificación es flexible ante los cambios que puedan ocurrir durante el desarrollo del proyecto. Queda definido el nivel de prioridad (P) con el que deben darle solución a las HU que sirve como guía para el desarrollo de las mismas. Se determina el riesgo (R). El número de la iteración (I) donde se realizará. Se definen también los puntos estimados (E) requeridos por historia, que no son más que los tiempos en los que se concibió inicialmente el desarrollo de cada HU. Por último se precisa la duración (D) en días y la fecha de entrega (F.E) acordada.

No H.U	Nombre	P	R	I	E	D	F.E
1	Diseño y creación de la Base de Datos	Alta	Alto	1	1,4	10	03/12/2014
2	Diseño de la interfaz de usuario	Media	Bajo		1	7	12/12/2014
3	Seguridad y Autenticación de usuario	Media	Medio	2	1	7	23/12/2014
4	Gestionar Administración del Sitio (<i>Backend</i>)	Alta	Alto		1	7	01/01/2015
5	Gestionar Plan de Estudio	Alta	Medio	3	2	14	21/01/2015
6	Operaciones con cohortes	Media	Alto	4	1,4	10	04/02/2015
7	Gestionar Expedientes	Alta	Alto	5	1,7	12	20/02/2015
8	Gestionar Ajustes	Alta	Alto	6	1,7	12	10/03/2015
9	Generar Reportes	Alta	Bajo	7	1,6	11	25/03/2015

Tabla 2 Historias de usuario planificadas

Según las características de las HU expuestas por el cliente en cuanto a similitud y funcionalidad se definieron cinco módulos fundamentales que agrupan el grueso de la aplicación en un conjunto de tareas necesarias para el funcionamiento del mismo, estos son:

Gestionar Administración del Sitio: este módulo es el encargado de administrar toda la información que se le provee al personal autorizado. En él se administran elementos como: la asignatura, tipo de cursos, ubicación y todo lo encargado con la creación de usuarios y sus respectivos roles para acceder a las diferentes funcionalidades del sistema.

Gestionar Plan de Estudio: este módulo es el encargado de controlar los planes de estudio existentes. En el mismo se crea, edita, elimina y busca un plan de estudio y se le asigna a cada uno de ellos las asignaturas correspondientes a través de las cohortes, la cantidad de horas clases para cada una de ellas, currículo al que pertenece, semestre en que se imparte, y otros datos de interés.

Gestionar Expediente Estudiantil: este módulo es el encargado de recoger las acciones elementales para su entrada en la aplicación web y posterior procesamiento. En el mismo se crea, edita, elimina y se busca un expediente estudiantil. Por las características de la aplicación no es necesario guardar toda la información referente a los estudiantes, según indicaciones del cliente.

Gestionar Ajuste a Plan: este módulo es el encargado de controlar los ajustes a plan de los estudiantes. En el mismo se crea, edita, elimina y busca un ajuste a plan realizado / a realizar, provee, en caso de crear un nuevo ajuste a plan un asistente que facilita dicho proceso.

Generar reportes: este módulo es el encargado de brindar información de interés a los usuarios que interactúen con la aplicación web.

En la Tabla 3 se muestra la HU 1: Diseño y creación de la base de datos.

Historia de Usuario	
Número: 1	Usuario: Administrador de Sistema
Nombre de la historia: Diseño y creación de la Base de Datos	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1,4	Iteración asignada: 1
Programador responsable: Abel González Mondéjar	
Descripción: Se diseña e implementa la base de datos en el SGBD MySQL.	
Observaciones: Verificar si a la base de datos se le estableció una contraseña.	

Tabla 3 HU-1 Diseño y creación de la Base de Datos

En la Tabla 4 se muestra la HU 2: Diseño de la interfaz de usuario.

Historia de Usuario	
Número: 2	Usuario: Administrador de Sistema
Nombre de la historia: Diseño de la interfaz de usuario	
Prioridad en negocio: Media	Riesgo en Desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Abel González Mondéjar	
Descripción: Se diseña la interfaz con la que el usuario va a interactuar.	
Observaciones: Para el diseño de la interfaz se empleó el <i>framework</i> CSS Metro UI CSS. Durante el progreso del proyecto se va modificando en función de las necesidades del cliente.	

Tabla 4 HU-2 Diseño de la interfaz de usuario

En la Tabla 5 se muestra la HU 3: Seguridad y Autenticación de usuario.

Historia de Usuario	
Número: 3	Usuario: Administrador de Sistema
Nombre de la historia: Seguridad y Autenticación de usuario	
Prioridad en negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Abel González Mondéjar	
Descripción: Inicia cuando un usuario intenta acceder al sitio, surge la necesidad de restringir el acceso al sistema mediante un conjunto de grupos (o roles). Se agregan los usuarios que interactúan con la aplicación.	
Observaciones: <ul style="list-style-type: none"> - Se debe verificar si son correctos los datos para acceder al sistema y si el usuario está registrado en la BD. - La contraseña almacenada en la BD la encripta el propio <i>framework</i>. 	

Tabla 5 HU-3 Seguridad y autenticación de usuario

En la Tabla 6 se muestra la HU 4: Gestionar Administración del Sitio (*Backend*).

Historia de Usuario	
Número: 4	Usuario: Administrador de Sistema
Nombre de la historia: Gestionar Administración del Sitio (<i>Backend</i>)	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Abel González Mondéjar	
Descripción: Inicia cuando el administrador del sistema provee los datos iniciales necesarios para que los usuarios puedan trabajar.	
Observaciones: El administrador de sistema tiene privilegios globales durante la fase de desarrollo solamente, al entregar el producto se restringen los permisos, los usuarios con el grupo “Secretaria Docente” tiene un acceso restringido a modificar los datos que están en el <i>backend</i> .	

Tabla 6 HU-4 Gestionar Administración del Sitio (*Backend*)

En la Tabla 7 se muestra la HU 5: Gestionar Plan de Estudio.

Historia de Usuario	
Número: 5	Usuario: Secretaria docente.
Nombre de la historia: Gestionar Plan de Estudio.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Abel González Mondéjar	
Descripción: Se inicia cuando se necesita gestionar la parte relacionada con los datos de los estudiantes. En el mismo se debe guardar toda la información inicial relacionada con el estudiante.	
Observaciones: <ul style="list-style-type: none"> - Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos. - Tener en cuenta que los datos no se eliminan, se deshabilitan. 	

Tabla 7 HU-5 Gestionar plan de estudio

En la Tabla 8 se muestra la HU 6: Operaciones con cohortes.

Historia de Usuario	
Número: 6	Usuario: Secretaria docente.
Nombre de la historia: Operaciones con cohortes.	
Prioridad en negocio: Media	Riesgo en Desarrollo: Alto
Puntos estimados: 1,4	Iteración asignada: 4
Programador responsable: Abel González Mondéjar	
Descripción: Se inicia cuando se necesita un cohorte asociado a un plan de estudio.	
Observaciones: <ul style="list-style-type: none"> - Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos. - Tener en cuenta que los datos no se eliminan, se deshabilitan. 	

Tabla 8 HU-6 Operaciones con cohortes

En la Tabla 9 se muestra la HU 7: Gestionar Expedientes.

Historia de Usuario	
Número: 7	Usuario: Secretaria docente.
Nombre de la historia: Gestionar Expedientes.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1,7	Iteración asignada: 5
Programador responsable: Abel González Mondéjar	
Descripción: Se inicia cuando se necesita gestionar la parte relacionada con los datos de los expedientes estudiantiles. En el mismo se debe guardar toda la información inicial relacionada con el estudiante.	
Observaciones: Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

Tabla 9 HU-7 Gestionar expedientes

En la Tabla 10 se muestra la HU 8: Gestionar Ajustes.

Historia de Usuario	
Número: 8	Usuario: Coordinador de carrera.
Nombre de la historia: Gestionar Ajustes.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1,7	Iteración asignada: 6
Programador responsable: Abel González Mondéjar	
Descripción: Se inicia cuando se necesita realizar el ajuste a plan de un estudiante.	
Observaciones: <ul style="list-style-type: none"> - Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos. - Tener en cuenta que los datos no se eliminan, se deshabilitan. 	

Tabla 10 HU-8 Gestionar ajustes

En la Tabla 11 se muestra la HU 9: Generar Reportes.

Historia de Usuario	
Número: 9	Usuario: Administrador de Sistema, Coordinador de carrera, Secretaria docente, Decano.
Nombre de la historia: Generar Reportes.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos estimados: 1,6	Iteración asignada: 9
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrarán los diferentes reportes que se ofrecen en la aplicación.	
Observaciones: Proveer la facilidad de exportar los reportes generados.	

Tabla 11 HU-9 Generar reportes

2.3.3 Planificación de iteraciones

Teniendo las historias de usuario del sistema definidas se pasa a realizar la planificación de las iteraciones. Atendiendo a lo mencionado con anterioridad se decide realizar siete iteraciones durante el desarrollo de la aplicación web, en la Figura 2 se detallan las mismas.

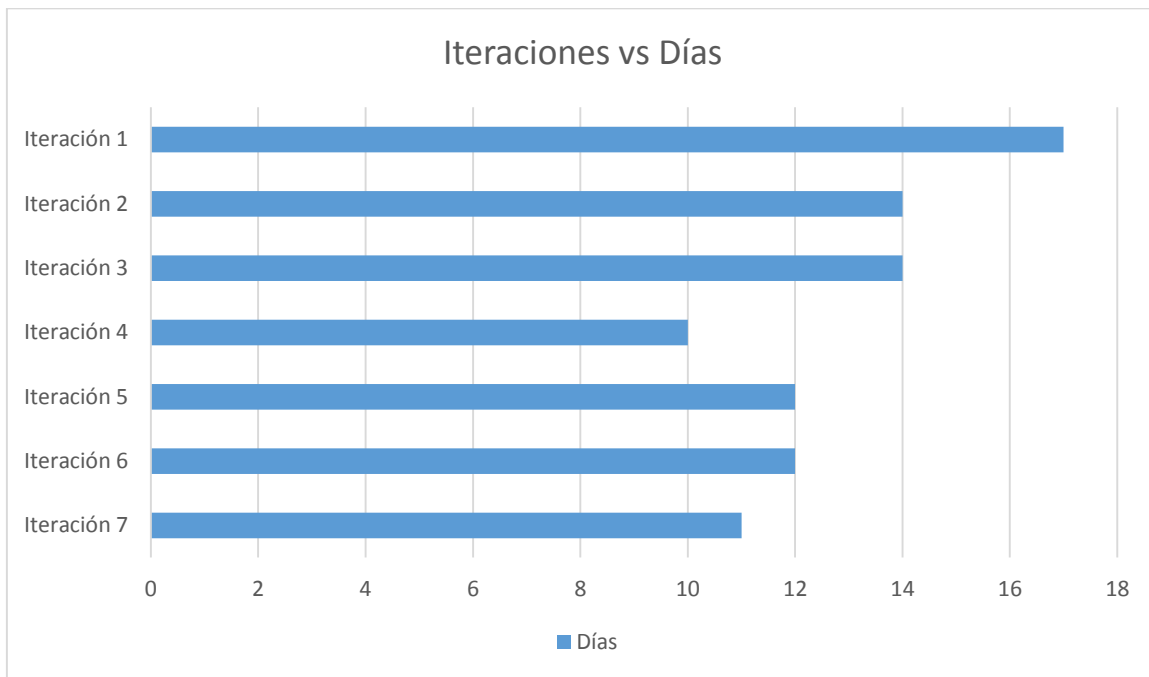


Figura 2 Iteraciones del proyecto

2.3.4 Reuniones

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de ser una metodología que evita la documentación exagerada, es muy estricta en la organización del trabajo.

Quincenalmente se realizará una reunión general con el equipo de trabajo con el objetivo de ver el progreso de cada iteración y en caso de algún inconveniente se deberá planificar una reunión aparte que deberá ser dentro de dicha quincena.

2.3.5 Plan de entregas

Una entrega está compuesta por una o más iteraciones, es planificada en base a funcionalidades concluidas. Durante el proceso de construcción de la solución al finalizar cada iteración, se debe realizar un análisis de seguimiento al plan de entregas que se reflejará en las incidencias. El hecho de que se realice una adecuada distribución de las iteraciones implica que se pueda cumplir satisfactoriamente el plan acordado con el cliente.

En este proyecto, según acuerdo con el cliente, se determinó hace una entrega por cada iteración concluida, por tanto, el plan de entregas coincide con el de iteraciones.

2.4 Etapa de diseño

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto. Los aspectos que se tratarán a continuación son: prototipo inicial de interfaz de usuario, simplicidad en el diseño, metáfora del sistema, tareas para cada iteración y tarjetas de clase, responsabilidad, colaboración (CRC).

2.4.1 Prototipo de interfaz de usuario

La Figura 3 muestra la página principal de la aplicación.

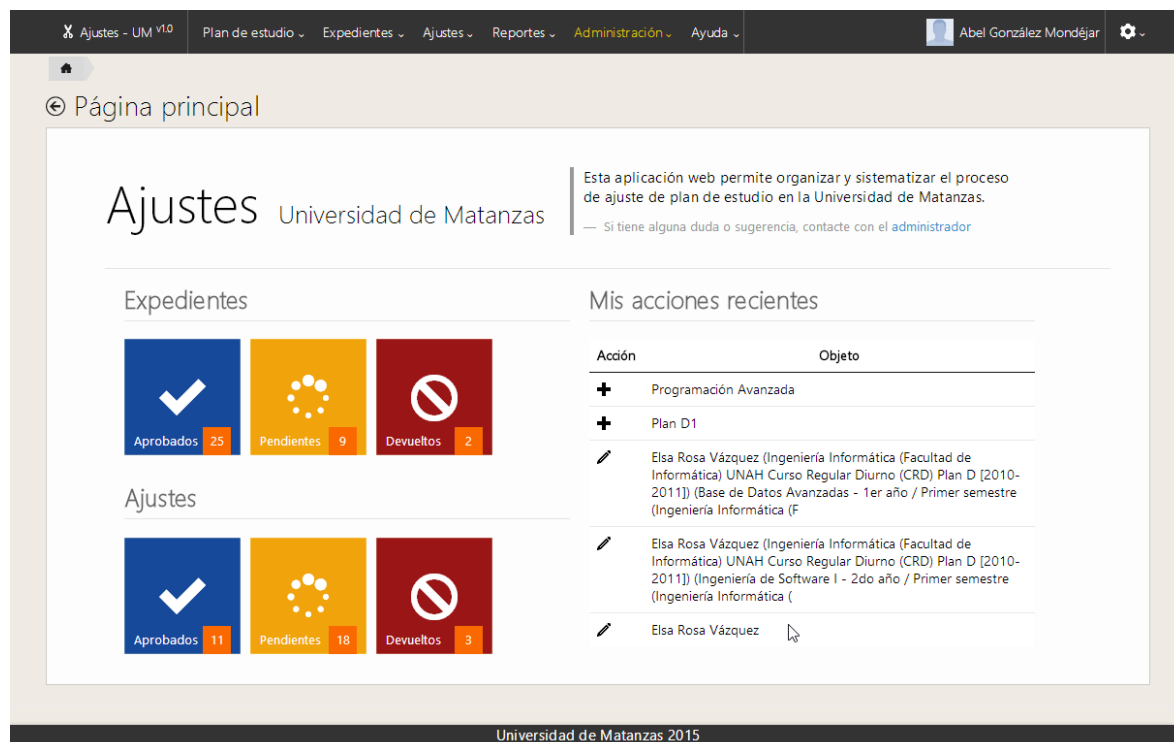


Figura 3 Página principal de la aplicación

2.4.2 Incidencias

Durante el desarrollo de la aplicación se reportaron algunas dificultades con el SGBD y su compatibilidad con Django, no obstante, haciendo las búsquedas oportunas en Stack Overflow (<http://stackoverflow.com/>) y en Google (<https://www.google.com.cu>), se pudo dar respuesta a los problemas encontrados no afectando así, el normal transcurso del desarrollo de la aplicación en el tiempo inicialmente planificado.

2.4.3 Metáfora

Una metáfora es una historia mediante la cual se plasma la arquitectura del sistema, mediante la misma se le da al grupo de desarrolladores una misma visión sobre el proyecto además de brindarles un primer vistazo muy completo de todas las herramientas que se utilizarán para desarrollar el sistema.

Es muy importante dentro del desarrollo de la metáfora darle nombres adecuados a todos los elementos del sistema constantemente, y que estos correspondan a un sistema de nombres consistente. Esto será de mucha utilidad en fases posteriores del desarrollo para identificar aspectos importantes del sistema.

Se puede decir que la aplicación web permite organizar y sistematizar el proceso de ajuste de plan de estudios en la Universidad de Matanzas.

2.4.4 Tareas a desarrollar

Cada historia de usuario se transformará en Tareas de Ingeniería (TI) según se muestra en la Tabla 12. Una historia de usuario puede transformarse en muchas tareas de programación, por esta razón solo se documentarán las más importantes.

Django, como *framework* provee facilidades para el programador en tareas de rutina, entre ellas la gestión de usuarios y roles, asimismo garantiza la gestión de tablas (llamados también modelos) a través de la interfaz de administración, por tanto, en la iteración 2 el proceso ingenieril consistió en agrupar, en la opción

“Gestionar”, las sub-tareas de Insertar, Editar, Buscar y Eliminar pues el propio *framework* lo provee, simplemente se plasma el tiempo estimado que le tomó al programador el realizar este proceso el cual, es casi el mismo para las tablas que se encuentran en la iteración mencionada. Se tendrán en cuenta y desglosarán, las tareas en las que el programador tuvo que implementar “desde cero”.

En la documentación del sistema se recoge toda la información referente a todo el proceso de las tareas y sus respectivas pruebas funcionales para cada iteración.

No. H.U	No Tarea.	Tareas de Ingeniería	Fecha de entrega
Iteración 1			
1	1	Diseño de la base de datos	03/12/2014
	2	Creación de la base de datos	
2	3	Diseño de la interfaz principal	12/12/2014
	4	Diseño de otras interfaces	
Iteración 2			
3	5	Gestionar Usuarios	23/12/2014
	6	Gestionar Grupos	
	7	Gestionar Permisos de grupos	
	8	Gestionar Permisos de usuario	
4	9	Gestionar Departamento	01/01/2015
	10	Gestionar Categoría Docente	
	11	Gestionar Grado Científico	
	12	Gestionar Trabajador	
	13	Gestionar Universidad	
	14	Gestionar Facultad	
	15	Gestionar Carrera	
	16	Gestionar Tipo de Curso	
	17	Gestionar Carrera-Tipo Curso	
	18	Gestionar Año	
	19	Gestionar Semestre	
	20	Gestionar Semestres de un año	
	21	Gestionar Curso	
	22	Gestionar País	
	23	Gestionar Provincia	
	24	Gestionar Municipio	
	25	Gestionar Ubicación	
	26	Gestionar Sexo	
	27	Gestionar Etnia	
	28	Gestionar Nacionalidad	
	29	Gestionar Asignatura	
	30	Gestionar Tipo de Asignatura	

	31	Gestionar Disciplina	
	32	Gestionar Currículo	
	33	Gestionar Tipo de Evaluación	
	34	Gestionar Motivo de ajuste	
	35	Gestionar Documentación analizada	
	36	Gestionar Situación	
Iteración 3			
5	37	Listar Plan de Estudio	21/01/2015
	38	Buscar Plan de Estudio	
	39	Nuevo Plan de Estudio	
	40	Editar Plan de Estudio	
	41	Eliminar Plan de Estudio	
	42	Detalle de Plan de Estudio	
	43	Listar Cohorte	
	44	Buscar Cohorte	
	45	Nuevo Cohorte	
	46	Editar Cohorte	
	47	Eliminar Cohorte	
	48	Detalle de Cohorte	
	49	Listar Asignatura de Cohorte	
	50	Buscar Asignatura de Cohorte	
	51	Nueva Asignatura de Cohorte	
	52	Editar Asignatura de Cohorte	
	53	Eliminar Asignatura de Cohorte	
	54	Detalle de Asignatura de Cohorte	
	55	Listar Asignatura optativa/electiva	
	56	Buscar Asignatura optativa/electiva	
	57	Nueva Asignatura optativa/electiva	
	58	Editar Asignatura optativa/electiva	
	59	Eliminar Asignatura optativa/electiva	
	60	Detalle de Asignatura optativa/electiva	
Iteración 4			
6	61	Comparar cohortes	04/02/2015
	62	Duplicar cohortes	
Iteración 5			
7	63	Listar Expediente Estudiantil	20/02/2015
	64	Buscar Expediente Estudiantil	
	65	Nuevo Expediente Estudiantil	
	66	Editar Expediente Estudiantil	
	67	Eliminar Expediente Estudiantil	
	68	Detalle del Expediente Estudiantil	
Iteración 6			
8	69	Listar Ajustes a plan	10/03/2015
	70	Buscar Ajuste a plan	
	71	Nuevo Ajuste a plan	

	72	Editar Ajuste a plan	
	73	Eliminar Ajuste a plan	
	74	Detalle de Ajustes a plan	
Iteración 7			
9	75	Expedientes aprobados	25/3/2015
	76	Expedientes pendientes	
	77	Expedientes devueltos	
	78	Ajustes aprobados	
	79	Ajustes pendientes	
	80	Ajustes devueltos	
	81	Estudiantes ajustados por cohorte	
	82	Estudiantes ajustados por curso	
	83	Resultado de comparativa	
	84	Resultado de expediente	
	85	Resultados de ajuste	

Tabla 12 Tareas de ingeniería del proyecto

En las tablas que se muestran se relacionan algunas tareas de ingeniería que tenían mayor peso en el desarrollo de esta investigación:

La Tabla 13 muestra la TI 1: Diseño de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de la tarea: Diseño de la base de datos.	
Tipo de tarea: Diseño	Puntos estimados: 1
Programador responsable: Abel González Mondéjar	
Descripción: Analizar profundamente el negocio referente al trabajo que se realiza con los estudiantes cuando se les realizar ajuste a plan para crear la base de datos que permita almacenar dicha información.	

Tabla 13 HU1 TI-1 Diseño de la base de datos

La Tabla 14 muestra la TI 2: Creación de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 2	Número de Historia de Usuario: 1
Nombre de la tarea: Creación de la base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,4
Programador responsable: Abel González Mondéjar	
Descripción: Crear en MySQL la base de datos con su respectivas relaciones y la integridad correspondiente entre las tablas.	

Tabla 14 HU1 TI-2 Creación de la base de datos

La Tabla 15 muestra la TI 3: Diseño de la interfaz principal.

Tarea de Ingeniería	
Número de Tarea: 3	Número de Historia de Usuario: 2
Nombre de la tarea: Diseño de la interfaz principal.	
Tipo de tarea: Diseño	Puntos estimados: 0,7
Programador responsable: Abel González Mondéjar	
Descripción: Apoyándose del <i>framework</i> web Metro UI CSS, diseñar la interfaz principal de la aplicación.	

Tabla 15 HU2 TI-3 Diseño de la interfaz principal

La Tabla 16 muestra la TI 4: Diseño de la interfaz principal.

Tarea de Ingeniería	
Número de Tarea: 4	Número de Historia de Usuario: 2
Nombre de la tarea: Diseño de otras interfaces.	
Tipo de tarea: Diseño	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Apoyándose de la interfaz principal, diseñar las demás interfaces que el usuario va a emplear.	

Tabla 16 HU2 TI-4 Diseño de otras interfaces

La Tabla 17 muestra la TI 5: Gestionar Usuario. Al referirse al ABCDEF, quiere decir: Agregar, Buscar, Cambiar, Desplegar (listar), Eliminar, Fichar (Ficha, cédula o Detalles de un registro), es equivalente al CRUD (del original en inglés: **Create, Read, Update and Delete**).

Tarea de Ingeniería	
Número de Tarea: 5	Número de Historia de Usuario: 3
Nombre de la tarea: Gestionar Usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,25
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará un formulario, generado por el <i>framework</i> , donde el(los) usuario(s) que pertenece(n) al grupo Administrador del Sistema podrá(n) insertar los datos del nuevo usuario. Es posible hace el ABCDEF completo.	

Tabla 17 HU3 TI-5 Gestionar usuarios

La Tabla 18 muestra la TI 9: Gestionar Departamento.

Tarea de Ingeniería	
Número de Tarea: 9	Número de Historia de Usuario: 4
Nombre de la tarea: Gestionar Departamento	
Tipo de tarea: Desarrollo	Puntos estimados: 0,04
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará un formulario, generado por el <i>framework</i> , donde el usuario con rol Secretaria Docente o Administrador de Sistema podrán insertar los datos referentes a la ubicación, estará compuesta por datos como municipio, provincia y país, deben verificarse los datos antes de agregarlos a la base de datos. Es posible hace el ABCDEF completo.	

Tabla 18 HU4 TI-9 Gestionar Departamento

La Tabla 19 muestra la TI 63: Listar Expedientes Estudiantiles.

Tarea de Ingeniería	
Número de Tarea: 63	Número de Historia de Usuario: 7
Nombre de la tarea: Listar Expedientes Estudiantiles.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará una lista de los expedientes activos en la aplicación web. Se muestran los botones Nuevo, Editar y Eliminar en función de los permisos del usuario que inició sesión. Al pie de la lista se muestra el paginador.	

Tabla 19 HU7 TI-63 Listar expediente estudiantil

La Tabla 20 muestra la TI 64: Buscar Expediente Estudiantil.

Tarea de Ingeniería	
Número de Tarea: 64	Número de Historia de Usuario: 7
Nombre de la tarea: Buscar Expediente Estudiantil.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará una lista con los resultados de la búsqueda de los expedientes activos en la aplicación web. Se muestran los botones Nuevo, Editar y Eliminar en función de los permisos del usuario que inició sesión. Al pie de la lista se muestra el paginador.	

Tabla 20 HU7 TI-64 Buscar expediente estudiantil

La Tabla 21 muestra la TI 65: Nuevo Expediente Estudiantil.

Tarea de Ingeniería	
Número de Tarea: 65	Número de Historia de Usuario: 7
Nombre de la tarea: Nuevo Expediente Estudiantil.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar los datos del expediente. Estará compuesta por datos como: nombre del estudiante, carrera de origen asignaturas vencidas y otros. Los datos de cada expediente deben validarse antes de agregarlos a la base de datos.	

Tabla 21 HU7 TI-65 Nuevo expediente estudiantil

La Tabla 22 muestra la TI 66: Nuevo Expediente Estudiantil.

Tarea de Ingeniería	
Número de Tarea: 66	Número de Historia de Usuario: 7
Nombre de la tarea: Editar Expediente Estudiantil.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará un formulario donde el usuario con los permisos necesarios podrá editar los datos del expediente. Los datos de cada expediente deben validarse antes de agregarlos a la base de datos.	

Tabla 22 HU7 TI-67 Editar expediente estudiantil

La Tabla 23 muestra la TI 67: Eliminar Expediente Estudiantil.

Tarea de Ingeniería	
Número de Tarea: 67	Número de Historia de Usuario: 7
Nombre de la tarea: Eliminar Expediente Estudiantil.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará una página para confirmar la eliminación del expediente estudiantil.	

Tabla 23 HU7 TI-67 Eliminar expediente estudiantil

La Tabla 24 muestra la TI 68: Detalle de Expediente Estudiantil.

Tarea de Ingeniería	
Número de Tarea: 68	Número de Historia de Usuario: 7
Nombre de la tarea: Detalle de Expediente Estudiantil.	
Tipo de tarea: Desarrollo	Puntos estimados: 0,3
Programador responsable: Abel González Mondéjar	
Descripción: Se mostrará una página con todos los datos del expediente estudiantil.	

Tabla 24 HU7 TI-69 Detalle de expediente estudiantil

2.4.5 Tarjetas CRC

La principal funcionalidad que tienen las tarjetas Clase-Responsabilidad-Componente (CRC) es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda están descritas las responsabilidades y a la derecha las clases que le sirven de soporte.

En el documento se muestran algunas de las tarjetas CRC que se elaboraron. La documentación del sistema recoge, más detalladamente, toda la información referente a todas las tarjetas CRC que se crearon.

La Tabla 25 muestra la tarjeta CRC para la clase **Estudiante**

Tarjeta CRC	
Clase: Estudiante	Super-clase: django.db.models.base.Model
Descripción: En esta clase se guardan los datos básicos de los estudiantes.	
Atributos	
carneIdentidad	models.CharField
nombre	models.CharField
apellidos	models.CharField
direccion	models.CharField
ubicacionId	models.ForeignKey(Ubicación)
sexoId	models.ForeignKey(Sexo)
etniaId	models.ForeignKey(Etnia)
correoElectronico	models.CharField
fechaIngESup	models.DateField
escalafon	models.CharField
indiceAcademico	models.DecimalField
nacionalidadId	models.ForeignKey(Nacionalidad)
activo	models.BooleanField
Responsabilidad	
Nombre	Colaborador
EstudianteListView	PaginationMixin, ListView
EstudianteBuscarListView	
EstudianteCreate	CreateView
EstudianteUpdate	UpdateView
EstudianteDetailView	DetailView
EstudianteDelete	DeleteView

Tabla 25 Tarjeta CRC – Estudiante

La Tabla 26 muestra la tarjeta CRC para la clase **PlanEstudio**

Tarjeta CRC	
Clase: PlanEstudio	Super-clase: django.db.models.base.Model
Descripción: En esta clase se guardan los datos de los planes de estudio.	
Atributos	
nombre	models.CharField
cantidadPeriodos	models.CharField
comentarios	models.CharField
activo	models.BooleanField
Responsabilidad	
Nombre	Colaborador
PlanEstudioListView	PaginationMixin, ListView
PlanEstudioBuscarListView	
PlanEstudioCreate	CreateView
PlanEstudioUpdate	UpdateView
PlanEstudioDetailView	DetailView
PlanEstudioDelete	DeleteView

Tabla 26 Tarjeta CRC - Plan de estudio

2.4.6 Modelo físico de la base de datos

En la Figura 4 se observa el modelo físico de la base de datos de la aplicación web, es posible hacerle zoom para agrandar la imagen.



Figura 4 Modelo físico de la base de datos

2.5 Conclusiones parciales

Como corresponde a la metodología de desarrollo XP, se crearon las historias de usuarios, planificadas en cada una de las iteraciones, se definió el tiempo de entrega de cada una de las historias, siempre a partir del hecho de que pueden sufrir cambios durante el proceso de implementación. Se pudo concretar al final de las iteraciones realizadas la entrega de la aplicación web completamente funcional.

Los elementos tratados en este capítulo sirvieron para llegar a un acuerdo entre las partes interesadas en el diseño y la estructura de la aplicación a través de la implementación de las funcionalidades que permitan realizar ajustes de planes de estudio en la Universidad de Matanzas.

Capítulo 3: Validación de la solución propuesta

3.1 Introducción

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo. Obviamente, la mejor forma de que las pruebas estén correctas es incluir al cliente en el diseño de las mismas.

3.2 Pruebas al software

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba.

La fase de pruebas es uno de los elementos que sostiene la metodología XP. En las pruebas de *software* los procesos soportados por la aplicación se cumplen completamente, es decir, los procesos fluyen desde su inicio hasta el final.

Es importante conocer que existen, según Roger Pressman (2011), principios básicos que guían el buen funcionamiento de las pruebas de software y que es necesario conocerlos antes de aplicar algún método de prueba. Entre ellos se mencionan:

- Que a todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente.
- Deben planificarse mucho antes de que empiecen.
- Deben comenzar por lo pequeño y progresar hacia lo grande.

La prueba de caja blanca del software realiza un minucioso examen de los detalles procedimentales. En ella se eligen y ejercitan una serie de caminos lógicos importantes y se examina el estado del programa en varios puntos, con el objetivo de comprobar si el estado real coincide con el esperado.

La prueba de caja negra se lleva a cabo sobre la interfaz del software, con el fin de demostrar que sus funciones son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

La Figura 5 representa gráficamente la filosofía de las pruebas de caja blanca y caja negra. Como se puede observar las pruebas de caja blanca necesitan conocer los detalles procedimentales del código, mientras que las de caja negra únicamente necesitan saber el objetivo o funcionalidad que el código ha de proporcionar.

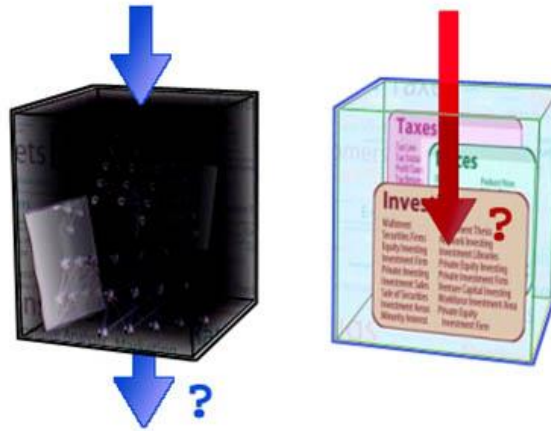


Figura 5 Filosofía de prueba de caja blanca y negra

3.2.1 Pruebas de caja blanca

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal, pruebas estructurales o pruebas unitarias en XP) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente (Oré, 2009).

Esta prueba usa una técnica denominada **prueba del camino básico** que le permite al diseñador obtener una medida de la complejidad lógica de un diseño y la usa para definir un conjunto básico de caminos de ejecución que garantizan que se ejecuten al menos una vez cada sentencia del programa.

También es importante definir la **complejidad ciclomática** como una métrica que proporciona una medida cualitativa de la complejidad lógica de un programa. Este valor se reduce al número de caminos independientes del camino básico de dicho programa y se calcula a partir de un diagrama o grafo de flujo.

Seguidamente se expone un ejemplo que ilustra la aplicación de esta técnica sobre el código del programa. El código seleccionado es el método para la autenticación de usuario el cual se muestra en la Figura 6.

$V(G)$: Marca el límite de casos de prueba para un programa. Cuando $V(G) > 10$ la probabilidad de defectos en el módulo o programa crece mucho, entonces quizás sea interesante dividir el módulo.

1. $V(G) = a - n + 2$
2. $V(G) = r$
3. $V(G) = c + 1$

Siendo:

- a: número de arcos o aristas del grafo.
- n: número de nodos del grafo.
- r: número regiones cerradas del grafo
- c: número de nodos prediados del grafo.

```
def autenticar(request):  
    if request.method == 'POST':  
        username = request.POST['username']  
        password = request.POST['password']  
        user = authenticate(username=username, password=password)  
        if user is not None:  
            if user.is_active:  
                login(request, user)  
                return HttpResponseRedirect(reverse('inicio'))  
            else:  
                return render(request, 'autenticar.html', {'error': True})  
        else:  
            return render(request, 'autenticar.html', {'error': True})  
    else:  
        return render(request, 'autenticar.html')
```

Figura 6 Método a realizar prueba de caja blanca (Autenticar)

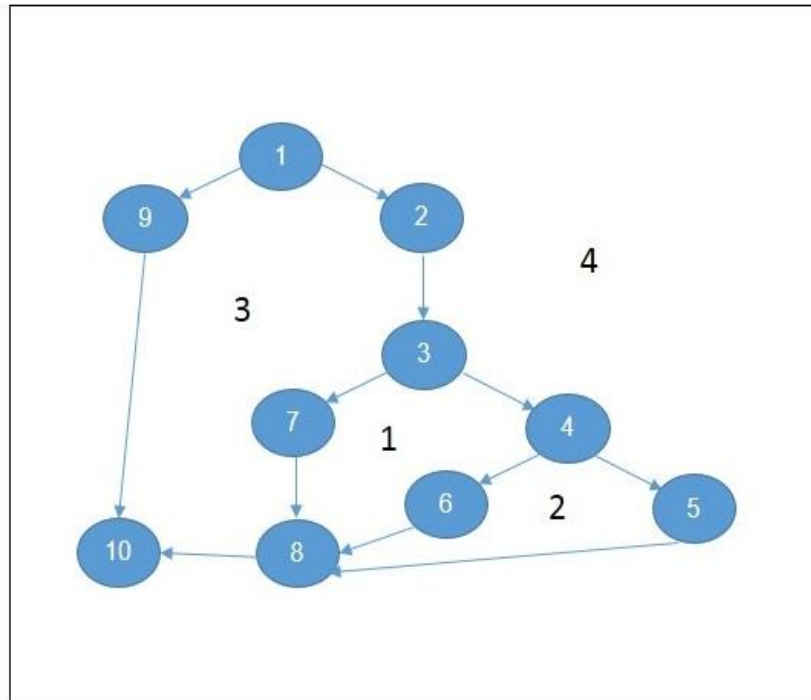


Figura 7 Grafo de flujo del método Autenticar

Complejidad Ciclomática del Grafo.

En el Grafo que se muestra en la Figura 7, los parámetros empleados fueron:

$$a = 12, n = 10, r = 4 \text{ y } c = 3$$

1. $V(G) = a - n + 2 = 12 - 10 + 2 = 4$
2. $V(G) = r = 4$
3. $V(G) = c + 1 = 3 + 1 = 4$

Camino Independientes.

- Camino 1: 1, 2, 3, 4, 5, 8, 10.
- Camino 2: 1, 2, 3, 4, 6, 8, 10.
- Camino 3: 1, 2, 3, 7, 8, 10.
- Camino 4: 1, 9, 10.

Los casos de pruebas para el método en la Figura 6 a emplear se muestran en la Tabla 27.

Camino	Caso de prueba	Resultado Esperado
1	Se verifica que la petición sea por POST, si existe el usuario con la contraseña y esta su cuenta activada se le permite entrar a la aplicación.	La aplicación redireccionará a su página de inicio, mostrando acceso a funciones según su grupo (rol).
2	Se verifica que la petición sea por POST, si existe el usuario con la contraseña y su cuenta no está activada no se permite la entrada.	El sistema mostrará un error indicando que la cuenta esta deshabilitada.
3	Se verifica que la petición sea por POST, de no existir el usuario con la contraseña no se permite la entrada a la aplicación.	El software mostrará un error alertando que verifique el usuario y contraseña.
4	Si la petición no es por POST el software no permite la entrada.	La aplicación mostrará la página de autenticación.

Tabla 27 Casos de prueba para el método autenticar

3.2.2 Pruebas de caja negra

Las pruebas de caja negra o pruebas de comportamiento se centran fundamentalmente en los requisitos funcionales del software, lo que permite obtener un conjunto de condiciones de entrada y que se ejerciten completamente todos los requisitos funcionales de un programa mediante un enfoque complementario que intenta descubrir diferentes tipos de errores como:

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

Durante el proceso de solución de la propuesta, en cada iteración por la que transitó el proceso de implementación del sistema se desarrolló un grupo de pruebas funcionales que quedaron reflejadas en la documentación del sistema.

3.2.3 Pruebas de aceptación

Las Pruebas de Aceptación (PA) también llamadas pruebas del cliente las especifica el cliente y se enfocan en las características generales y las funcionalidades de la aplicación. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado.

Las pruebas de aceptación se llevarán a cabo mediante la redacción de los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades. Luego se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo, para así reunir los miembros del proyecto seleccionados para realizarlas. Finalmente, se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

En la Tabla 28 se muestra la PA 2 que comprueba el diseño de la interfaz de usuario.

Prueba de Aceptación	
Número del caso de prueba: 2	Número de Historia de Usuario: 2
Nombre del caso de prueba: Prueba del diseño de la interfaz de usuario	
Descripción: Verificar si funciona el <i>responsive</i> y el funcionamiento de los botones.	
Condiciones de ejecución: Que la aplicación web esté en ejecución.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> - Se ajusta el tamaño de la ventana del navegador para comprobar si el <i>responsive</i> funciona en distintos de dispositivos con características de pantallas distintas (móvil, tablet y ordenador personal). - Se prueba la aplicación en distintos navegadores para comprobar su compatibilidad. - Se prueba el funcionamiento correcto de los botones, en su correspondiente ejecución. 	
Resultado esperado: Que la aplicación se ajuste a las distintas resoluciones de pantalla y sea compatible con varios navegadores que se emplean en la actualidad. Que todos los botones funcionen correctamente.	
Evaluación de la prueba: Satisfactorio.	

Tabla 28 HU2 PA-2 Prueba del diseño de la interfaz de usuario

La Figura 8 muestra el resultado de la PA 2, empleando el navegador Google Chrome 36.0.1948.0.

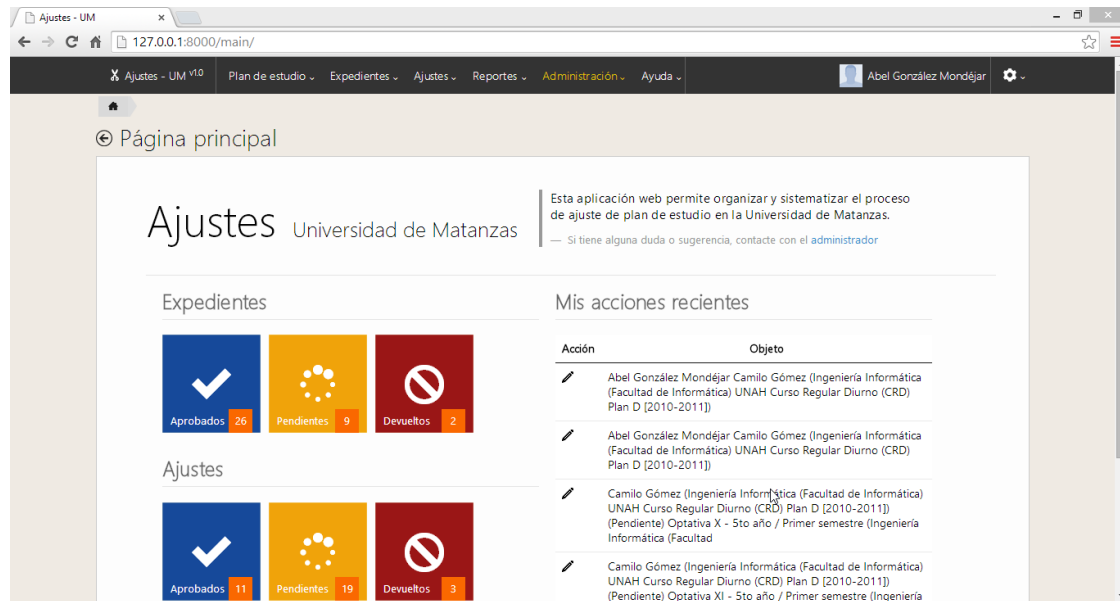


Figura 8 Resultado de la prueba de aceptación HU2 PA-2 en Google Chrome 36.0.1948.0

La Figura 9 muestra el resultado de la PA 2, empleando el navegador Mozilla Firefox 35.0

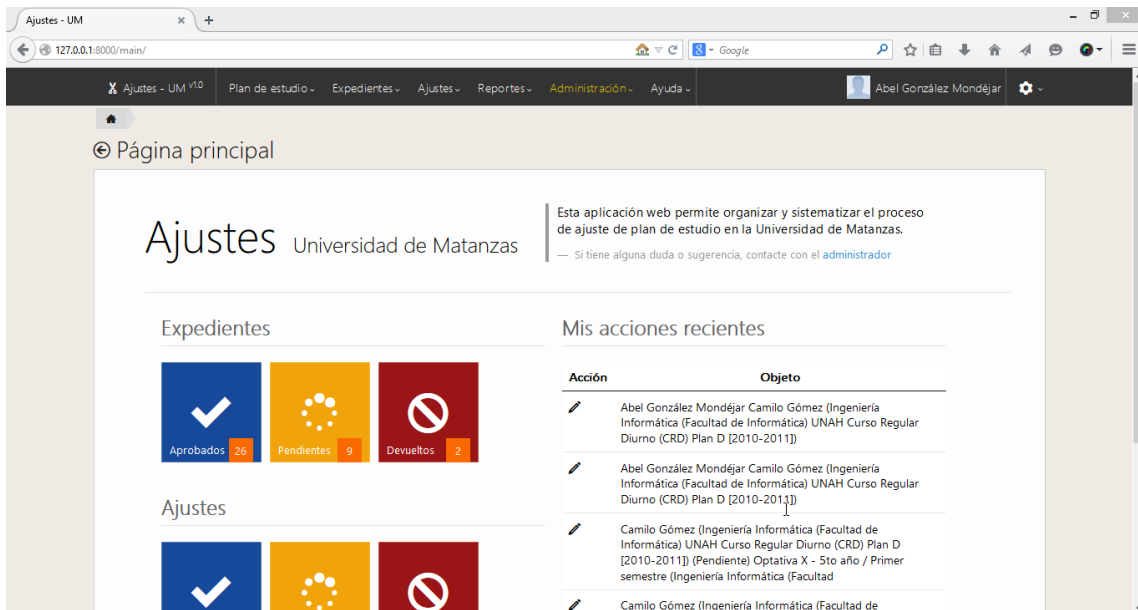


Figura 9 Resultado de la prueba de aceptación HU2 PA-2 en Mozilla Firefox 35.0

La Figura 10 muestra el resultado de la PA 2, empleando el navegador Internet Explorer 11.0.9600.17031

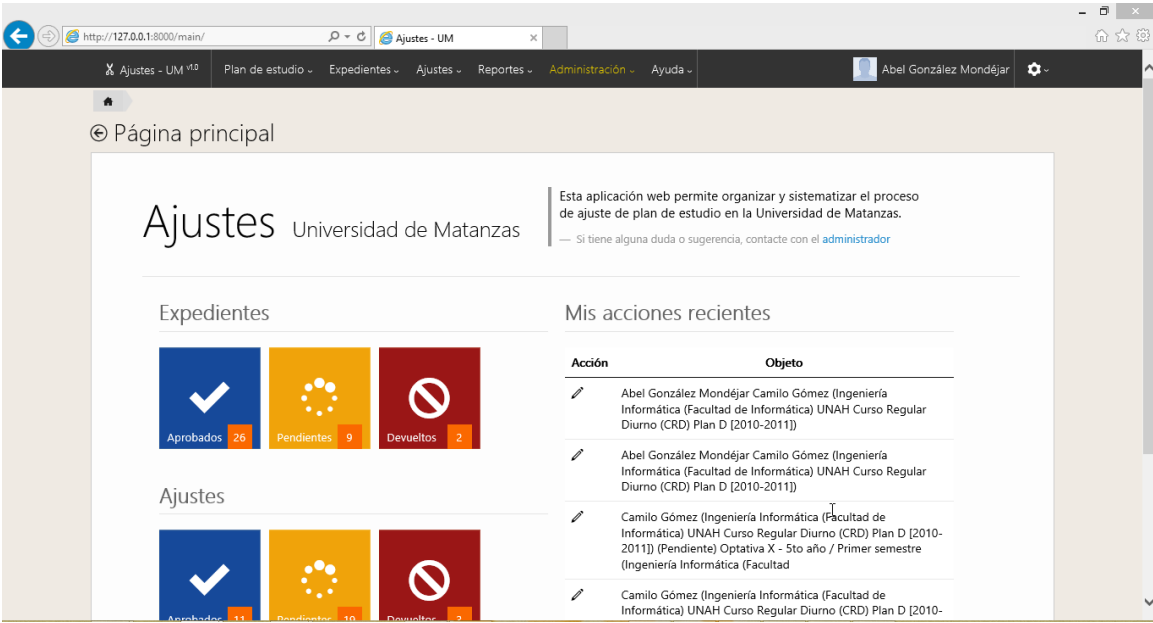


Figura 10 Resultado de la prueba de aceptación HU2 PA-2 en Internet Explorer 11.0.9600.17031

Se comprobó el *responsive* de la aplicación para comprobar la compatibilidad con distintas resoluciones de pantalla y dispositivos (Figura 11, Figura 12, Figura 13).

Capítulo 3: Validación de la solución propuesta

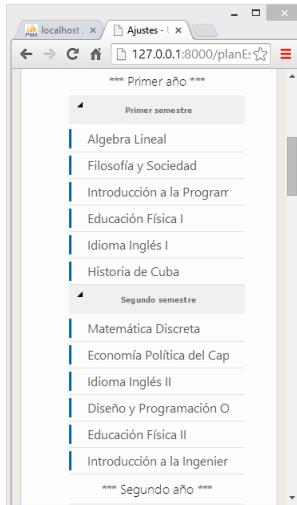


Figura 11 Resultado de la prueba de aceptación HU2 PA-2 (móvil @ 250px x 500px)

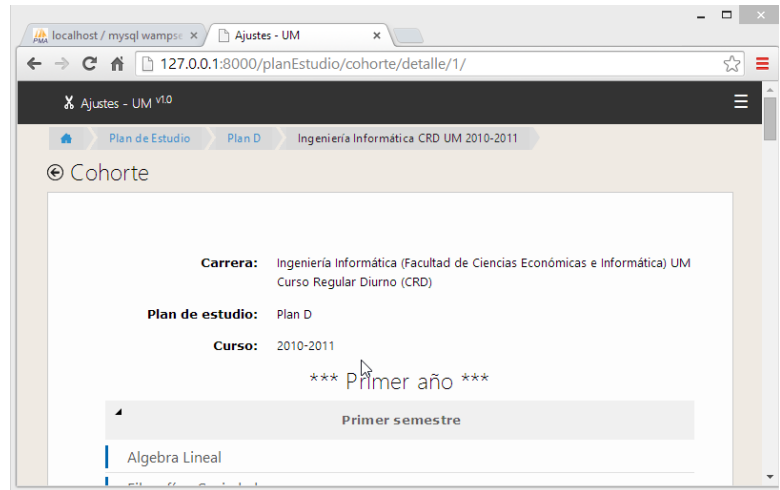


Figura 12 Resultado de la prueba de aceptación HU2 PA-2 (tablet @ 880px x 460px)

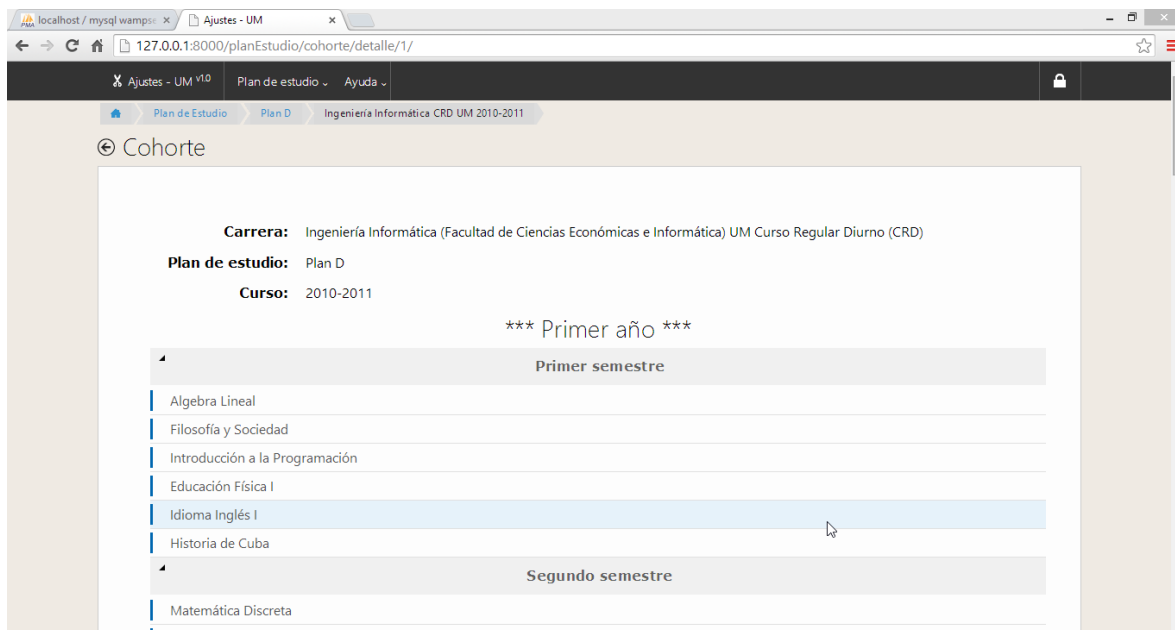


Figura 13 Resultado de la prueba de aceptación HU2 PA-2 (ordenador @ 1366px x595px)

En la Tabla 29 se orientan las operaciones necesarias para hacer la prueba de aceptación para la HU-7, la misma se realiza en un asistente de cuatro pasos.

Prueba de Aceptación	
Número del caso de prueba: 7	Número de Historia de Usuario: 7
Nombre del caso de prueba: Prueba de la gestión de expedientes estudiantiles.	
Descripción: Se insertan los datos necesarios para crear un expediente estudiantil a través de un asistente en cuatro pasos. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, luego se insertarán de manera correcta para comprobar que los datos sean almacenados. Se modifican los datos de un expediente estudiantil. Se modificarán de forma incorrecta, dejando campos en blanco para verificar la validación, luego se modificarán de manera correcta para comprobar que los datos sean almacenados y cargados. Se eliminará un expediente estudiantil. Primero se presionará el botón eliminar sin seleccionar un expediente estudiantil para verificar la validación, luego se eliminará de forma correcta para comprobar que se eliminen los datos.	
Condiciones de ejecución: El usuario debe tener los permisos suficientes para realizar esta operación.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> - Dejar en blanco campos obligatorios. - Insertar el nombre de un elemento existente. - Insertar los datos correctamente. - Modificar los datos dejando en blanco campos obligatorios. - Modificar los datos correctamente. - Verificar que se muestren los nuevos datos. 	
Resultado esperado: La aplicación debe alertar al usuario cuando se inserten datos erróneos, en caso de que ya exista un elemento con el mismo nombre y si no se ha seleccionado un elemento al presionar el botón eliminar. Cuando se inserten o modifiquen los datos correctamente, el software debe almacenarlos en la base de datos y mostrarlos.	
Evaluación de la prueba: Satisfactorio.	

Tabla 29 HU7 PA-7 Prueba de la gestión de expedientes estudiantiles

En la Figura 14 se observa el primer paso del asistente de la PA-7

The screenshot shows the 'Expedientes estudiantiles' (Student Records) interface. The 'Datos del estudiante' (Student Data) tab is active. The form contains the following fields:

- CI:** 89112825226
- Nombre:** Abel
- Apellidos:** González Mondéjar
- Dirección:** Ruiz 964 apto 2 e/ 20 y 21
- Ubicación:** Matanzas, Matanzas, Cuba
- Nacionalidad:** Cubana
- Sexo:** Masculino
- Etnia:** Blanco(a)
- E-mail:** abelglez89@gmail.com
- Ingresó a la ES:** 26/9/2010

Figura 14 Resultados de la prueba de aceptación HU7 PA-7 (primer paso)

En la Figura 15 se observa el segundo paso del asistente de la PA-7

The screenshot shows the 'Expedientes estudiantiles' (Student Records) interface. The 'Cohorte de origen' (Origin Cohort) tab is active. The form contains the following fields:

- Universidad:** Universidad de Matanzas
- Facultad:** Facultad de Ciencias Económicas e Informática
- Carrera:** Ingeniería Informática
- Tipo de curso:** Curso Regular Diurno
- Plan de estudio:** Plan D
- Curso:** 2010-2011

Navigation buttons are located at the bottom:

- Anterior (Previous)
- Siguiente (Next)
- No, cancelar (No, cancel)

Figura 15 Resultados de la prueba de aceptación HU7 PA-7 (segundo paso)

En la Figura 16 se observa el tercer paso del asistente de la PA-7

*** Primer año ***

Primer

- Algebra Lineal
- Filosofía y Sociedad
- Introducción a la Programación
- Educación Física I
- Idioma Inglés I
- Historia de Cuba
- Procesos tecnológicos

Segundo

- Matemática Discreta
- Economía Política del Capitalismo
- Idioma Inglés II
- Diseño y Programación Orientada a Objetos
- Educación Física II
- Introducción a la Ingeniería de Software

Figura 16 Resultado de la prueba de aceptación HU7 PA-7 (tercer paso)

En la Figura 17 se observa el cuarto paso del asistente de la PA-7

CI: 89112825226 **Sexo:** Masculino

Nombre: Abel **Etnia:** Blanco(a)

Apellidos: González Mondéjar **Email:** abelglez89@gmail.com

Dirección: Ruíz 964 apto 2 e/ 20 y 21 **Ingreso al ES:** 26/9/2010

Ubicación: Matanzas, Matanzas, Cuba **Índice académico:** 4.44

Nacionalidad: Cubana

Asignaturas vencidas

Asignatura	Año	Semestre	Cantidad de horas totales
Algebra Lineal	1er	1er - S	80
Filosofía y Sociedad	1er	1er - S	80

Finalizar
Guardar cambios

Figura 17 Resultados de la prueba de aceptación HU7 PA-7 (cuarto paso)

En la Tabla 30 se orientan las operaciones para hacer la prueba de aceptación para la HU-9, la misma se realiza en un asistente de cuatro pasos.

Prueba de Aceptación	
Número del caso de prueba: 9	Número de Historia de Usuario: 9
Nombre del caso de prueba: Prueba gestión de ajuste de un plan.	
Descripción: Se insertan los datos necesarios para crear un ajuste a través de un asistente en cuatro pasos. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, luego se insertarán de manera correcta para comprobar que los datos sean almacenados. Se modificarán los datos de un ajuste. Se modificarán de forma incorrecta, dejando campos en blanco para verificar la validación, luego se modificarán de manera correcta para comprobar que los datos sean almacenados y cargados.	
Condiciones de ejecución: El usuario debe tener los permisos suficientes para realizar esta operación.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> - Dejar en blanco campos obligatorios. - Insertar el nombre de un elemento existente. - Insertar los datos correctamente. - Modificar los datos dejando en blanco campos obligatorios. - Modificar los datos correctamente. - Verificar que se muestren los nuevos datos. 	
Resultado esperado: El software debe alertar al usuario cuando se inserten datos erróneos, en caso de que ya exista un elemento con el mismo nombre y si no se ha seleccionado un elemento al presionar el botón eliminar. Cuando se inserten o modifiquen los datos correctamente, la aplicación debe almacenarlos en la base de datos y mostrarlos.	
Evaluación de la prueba: Satisfactorio.	

Tabla 30 HU8 PA-8 Prueba de la gestión de ajustes a plan

En la Figura 18 se observa el primero paso del asistente de la PA-9

Ajustes - UM v1.0 | Plan de estudio | Expedientes | Ajustes | Reportes | Administración | Ayuda | Abel González Mondéjar

Ajustes | Nuevo | Ajuste a plan para Abel González Mondéjar

Ajustes

Datos del ajuste | Cohorte a la que aspira | Ajuste de asignaturas | Resultados

Estudiante: Abel González Mondéjar

Motivo de ajuste: Baja

Documentación: Expediente académico

Curso actual: 2014-2015

Siguiete

No, cancelar

Universidad de Matanzas 2015

Figura 18 Resultados de la prueba de aceptación HU8 PA-8 (primer paso)

En la Figura 19 se observa el segundo paso del asistente de la PA-9

Ajustes - UM v1.0 | Plan de estudio | Expedientes | Ajustes | Reportes | Administración | Ayuda | Abel González Mondéjar

Ajustes | Nuevo | Ajuste a plan para Abel González Mondéjar

Ajustes

Datos del ajuste | Cohorte a la que aspira | Ajuste de asignaturas | Resultados

Universidad: Universidad de Matanzas

Facultad: Facultad de Ciencias Económicas e Informática

Carrera: Ingeniería Informática

Tipo de curso: Curso Regular Diurno

Plan de estudio: Plan D

Curso: 2014-2015

Anterior | Siguiete

No, cancelar

Universidad de Matanzas 2015

Figura 19 Resultados de la prueba de aceptación HU8 PA-8 (segundo paso)

En la Figura 20 se observa el tercer paso del asistente de la PA-9

*** Primer año ***	
Primer semestre	
Matemática I	Situación: Abonada
Álgebra Lineal	Situación: Convalidada Con cuál: Álgebra Lineal
Filosofía y Sociedad	Situación: Convalidada Con cuál: Filosofía y Sociedad
Introducción a la Programación	Situación: Pendiente
Educación Física I	Situación: Pendiente
Idioma Inglés I	Situación:

Figura 20 Resultados de la prueba de aceptación HU8 PA-8 (tercer paso)

En la Figura 21 se observa el cuarto paso del asistente de la PA-9

Estudiante: Abel González Mondéjar
Motivo de ajuste: Baja
Documentación: Expediente académico
Curso de ajuste: 2014-2015

Ingeniería Informática CRD
 Plan de estudio — Plan D
 Curso — 2014-2015
 Universidad — Universidad de Matanzas

***** Primer año *****

Primer	
Matemática I	
Álgebra Lineal	Convalidada con: Álgebra Lineal
Filosofía y Sociedad	Convalidada con: Filosofía y Sociedad
Introducción a la Programación	
Educación Física I	
Idioma Inglés I	
Historia de Cuba	

***** Primer año *****

Primer	
Álgebra Lineal	
Filosofía y Sociedad	
Introducción a la Programación	
Educación Física I	
Idioma Inglés I	
Historia de Cuba	
Procesos tecnológicos	

Segundo

Figura 21 Resultados de la prueba de aceptación HU8 PA-8 (cuarto paso)

3.4 Análisis de los resultados obtenidos

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las realizadas, se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión. Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa y a medida que fueron avanzando se desplazaron a módulos integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentran en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento.

El desarrollo de la aplicación cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad.

3.5 Conclusiones del capítulo

Las pruebas realizadas utilizando las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de los requerimientos del cliente. El cliente confirma que la aplicación web facilita en gran medida el ajuste a plan y agiliza así el trabajo de los coordinadores de carrera, por otra parte queda anexado en la documentación del sistema el Manual de Usuario donde se explica claramente cada una de las funcionalidades de la aplicación.

Conclusiones generales

Como resultado de esta investigación quedaron satisfechos los objetivos trazados arribando a las siguientes conclusiones:

- El ajuste a plan es un proceso engorroso y complicado debido a su magnitud.
- Los softwares encontrados, vinculados al tema no le dan solución al problema planteado por lo que no es factible su utilización.
- El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió aportar los elementos necesarios para dar solución a la problemática planteada.
- Se utilizaron las herramientas de software más factibles para la construcción de la solución.
- Se realizó el análisis y diseño del sistema utilizando como metodología de desarrollo XP.
- La implementación del software y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados por el cliente.

De forma general, se concluye que la aplicación web desarrollada es una herramienta confiable y fácil de utilizar pues eleva la calidad del proceso de ajuste a plan de estudio, reduce los errores a cometer, facilita y perfecciona el proceso y brinda reportes de interés a los directivos. Por estas razones, queda respondida la pregunta científica planteada en la introducción.

Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

- Poner en marcha la aplicación y posteriormente extenderlo a otras universidades.
- Integrar con SIGENU a través de los servicios web que ofrece.
- Aprovechar las posibilidades de la información almacenada en la base de datos e implementar un módulo para los servicios docentes que ofrecen los departamentos de la Universidad.
- Agregarle nuevos reportes que sean de interés para los usuarios que interactúen con la aplicación.

Bibliografía

- Alcalde, A. (2013). El baúl del programador. from <http://elbauldelprogramador.com/los-10-mejores-frameworks-gratis-de-aplicaciones-web/>
- Alonso Rodríguez, Y. (2014). *Sistema informático para la gestión de expedientes electrónicos*. Universidad de Matanzas, Matanzas.
- Aloy, A. (2009). Django vs PHP frameworks. Retrieved from <http://trespams.com/blog/2009/05/10/django-vs-php-frameworks/>
- Aloy, A. (2010). ¿Va a desaparecer Python? Retrieved from <http://blog.apsl.net/weblog/2010/07/29/va-a-desaparecer-python/>
- Aluicio Sarduy, D. A. (2011). *Asistente para la Gestión de la Documentación Docente en las Universidades*. Universidad de Matanzas Camilo Cienfuegos, Matanzas.
- Álvarez, M. A. (2007). Arquitectura cliente-servidor. from <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
- Álvarez, R. (2001). Introducción a HTML. Retrieved from Desarrollo web website: <http://www.desarrolloweb.com/articulos/534.php>
- Amaro Calderón, S. D. V. R., S.D. y J.C (2007). *Metodologías Ágiles*. Universidad Nacional de Trujillo, Perú, Perú.
- Bom Malla, N. (2010). PHP vs Python: Analysis. Retrieved from EnBeeone3 website: <http://enbeeone3.com/php-vs-python-analysis/>
- Brookins, A. (2014). An Epic Review of PyCharm 3 from a Vim User's Perspective.
- Ccoica Falcón, R. A. M. O., W. A. E. (2006). *Tecnología de la información. Procesos de ciclo de vida del software*. Universidad Nacional Tecnológica del Cono Sur de Lima, Lima, Perú.
- Collins, M. J. (2012). Pro HTML5 with Visual Studio 2012 (pp. 409 (se encuentra en la página 405)): Apress.
- Delgado Expósito, E. R. L. Z. (2006). *Sistema Informático para la Recuperación de Información Docente*. Universidad de Matanzas, Matanzas. (página 131)
- Eguíluz Pérez , J. (2009). Introducción a XHTML. In R. R. Ortega (Ed.), (pp. 179). Universidad de Matanzas.

- Eguíluz Pérez, J. (2014). Introducción a CSS. Retrieved from http://librosweb.es/css/capitulo_1.html.
- Eguíluz Pérez, J. (2014). ¿Qué es Javascript? Retrieved from http://librosweb.es/javascript/capitulo_1.html.
- Española, D. d. I. R. A. (2014a). Definición de estudiante.
- Española, D. d. I. R. A. (Ed.) (2014b) Real Academia Española.
- Española, D. d. I. R. A. (Ed.) (2014c).
- Española, D. d. I. R. A. (2014d). Definición de Software Libre Retrieved 13-10, 2014, from <http://www.libre.org/es/libre/software-libre/definicion-de-software-libre>
- Felipe, J. (2013). Principio DRY. <http://collectioncode.com/principio-dry/>
- Fernández Orquín, A., E. D. Expósito., Z. R. LLorca., Y. B. González., Gutiérrez Vasconcelo, Yoan; Pérez Martínez, Abel; J. D. Blanco., L. P. Alfonso., I. G. Valiente; Pérez Chávez, Roger. (2004). *SIPRID. Sistema Informático para la Recuperación de Información* Retrieved from http://www.academia.edu/2247967/SIPRID._Sistema_Inform%C3%A1tico_para_la_Recuperaci%C3%B3n_de_Informaci%C3%B3n_Docente
- Fernández Orquín, A. G. V., Yoan; Pérez Martínez, Abel; García Vasconcelos, Yaniseth; Miranda, Lisandra. (2010). *Extracción de Información en documentos no etiquetados del entorno educacional* (pp. página 5 in XII Simposio Internacional de Comunicación Social 2010). Retrieved from http://scholar.google.com/citations?view_op=view_citation&hl=en&user=M4FJQtsAAAAJ&citation_for_view=M4FJQtsAAAAJ:2osOgNQ5qMEC
- Flanagan, D. (2006). *JavaScript: the definitive guide*: O'Reilly Media, Incorporated.
- Freddy Vega, J. V. D. H., Christian. (2011). *Guía HTML5*.
- García Fernández, A. (2013). ¿Qué es Django Framework? Retrieved 2014, from <http://vanuta.com/articulos/que-es-django-framework/>
- Goldstein, A. L., Louis; Weyl, Estelle. (2011). *HTML5 & CSS3 for the real world* (K. Steele Ed.). Collingwood, Australia.
- González, O. (2014). Cómo escribir código mantenible - parte 2. Retrieved from Codehero website: <http://codehero.co/como-escribir-codigo-mantenible-parte-2/>

- Guo, P. (2014). Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. from <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>
- Horruitiner Silva, P. (2006). *La universidad cubana: el modelo de formación*: Editorial Félix Varela.
- Infante Moreno, S. (2012). Curso Django: Entendiendo como trabaja Django. from <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/>
- Gaceta Oficial de la República de Cuba, 29 C.F.R. (2014).
- Kennedy, C. M. B. (1999). *HTML - The complete guide* (E. A. M. Yazmh Juárez Parra, Trans. 2 ed.): McGraw-Hill interamericana Editores, S.A de C.V.
- Lingan, J. B. (2007). Web server. from <http://whatis.techtarget.com/definition/Web-server>
- Marcano, H. (2009). *Desarrollo de una aplicación educativa bajo ambiente web, como apoyo para la enseñanza de la asignatura bases de datos orientadas a objetos, para la carrera de Licenciatura en Informática de la Universidad de Oriente*.
- Martínez Pecoz, D. (2014). PostgreSQL vs MySQL. from <http://danielpecos.com/documents/postgresql-vs-mysql/>
- Merchán Quijije, R. E. (2009). *Los métodos aplicados en la enseñanza de lenguaje y comunicación y su incidencia en el aprendizaje de los-as estudiantes de los octavos años de educación básica, paralelo "A, B y C" del colegio Juan Montalvo*. (Tesis de grado para la obtención del título de Licenciada en Ciencias de la Educación), Universidad Laica Eloy Alfaro de Manabí, Manta, . Retrieved from <http://repositorio.ulead.edu.ec/bitstream/26000/1088/1/T-ULEAM-054-0025.pdf>
- Mikoluk, K. (2013). MySQL vs PostgreSQL: Por Qué MySQL es Superior A PostgreSQL. Retrieved 2014-11-05, from <https://www.udemy.com/blog/mysql-vs-postgresql-por-que-mysql-es-superior-a-postgresql/>

- Mitchell, B. (2014). Apache. from http://compnetworking.about.com/cs/webservers/g/bldef_apache.htm
- Nawrocki, M. (2013). Five Integrated Development Environment applications. from <http://www.techrepublic.com/blog/five-apps/five-integrated-development-environment-applications/>
- Oré, A. (2009). Calidad y software. from http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php
- Oreshnikova, A. (2010). Con PyCharm, los desarrolladores de Python obtienen finalmente una IDE potente. <http://www.europapress.es/economia/noticia-comunicado-pycharm-desarrolladores-python-obtienen-finalmente-ide-potente-20101014090555.html>
- Phillips, A. (2010). Software Development Methodologies. Retrieved from Codeproject website: <http://www.codeproject.com/Articles/124732/Software-Development-Methodologies>
- Picca, C. (2014). Django desde Cero - Instalación y herramientas. Retrieved from Codehero website: <http://codehero.co/django-desde-cero-instalacion-y-herramientas/>
- Pressman, R. S. (2011). Técnicas de prueba del software, en Ingeniería del Software. Un enfoque práctico.
- Pressman, R. S. L., David. (2009). *Web engineering - A practitioner's approach*: McGraw-Hill.
- Rodríguez-Guerra, J. (2014). Descubre qué es Django, el framework web de moda. from <http://computerhoy.com/noticias/internet/descubre-que-es-django-framework-web-moda-8641>
- Rodríguez Díaz, S. (2012). *Módulo Informático para la Gestión de la Información Académica de los Estudiantes*. Universidad de Matanzas, Matanzas.
- Rose, M. (2007). Integrated development environment. from <http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>

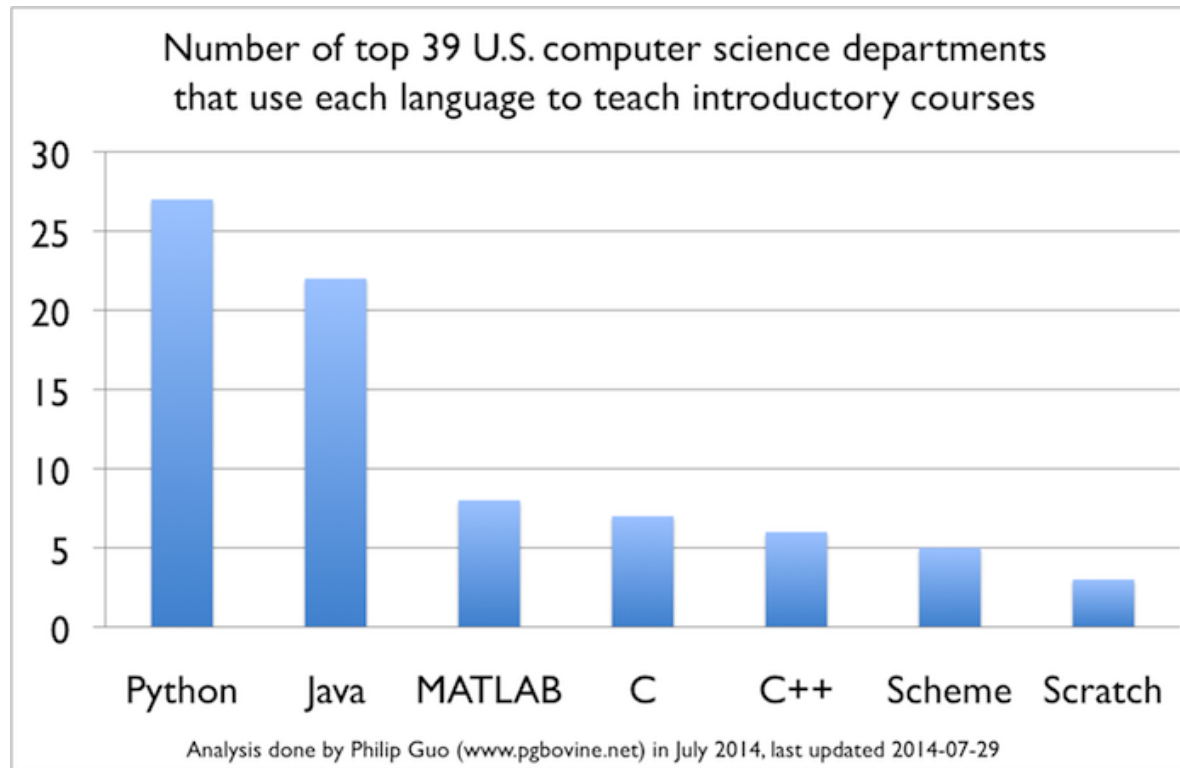
- s.a. (2014a). Ecuasupport. Retrieved 10-06, 2014, from http://www.ecuasupport.com/ecuasupport/index.php?option=com_content&view=article&id=10:gestacad&catid=6:aplicaciones-web&Itemid=9
- s.a. (2014b). Paw, lugar de coincidencia en internet. from http://www.ucatolica.edu.co/paw/estudiantes/index.php?option=com_xmap&sitemap=1&Itemid=42
- s.a. (2014c). SIGENU - Quienes somos. Retrieved 10-08, 2014, from <http://sigenu.mes.edu.cu:8080/dmmes/pages/info/aboutUsWelcome.faces>
- s.a. (2014d). Sistema de Gestión Universitaria. Retrieved 10-06, 2014, from <http://guarani.unne.edu.ar/>
- Reglamento para el Trabajo Docente y Metodológico en la Educación Superior, 40 C.F.R. (2007).
- Resolución 120/2010 (2010).
- Van Rossum, G. (2014). El tutorial de Python. In F. J. Louis Drake (Series Ed.). Retrieved from <http://python.org.ar/pyar/Tutorial>.
- Vera León, L. R. V., Dunia. (2008). *Sistema Informático para la Gestión de Información Docente*. Universidad de Matanzas, Matanzas.

Anexos

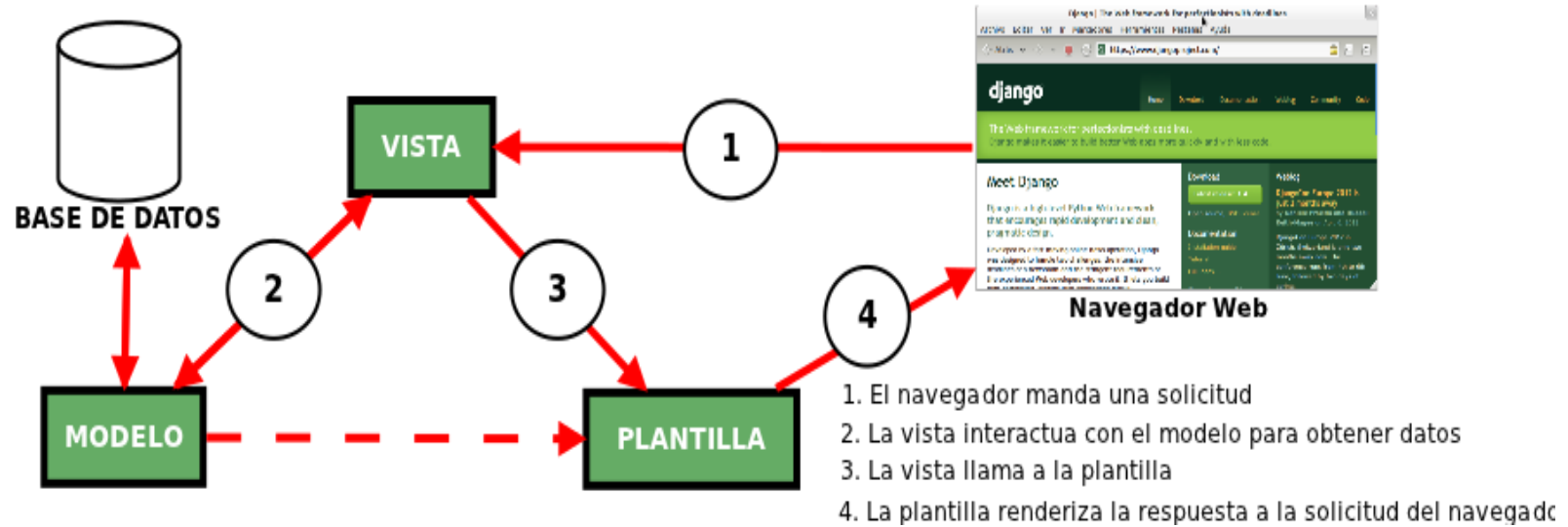
Anexo 1: Diferencias entre metodologías ágiles y tradicionales

Metodología Ágil	Metodología Tradicional
Pocos artefactos.	Más artefactos.
Pocos roles, más genéricos y flexibles.	Más roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Se hace énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Se hace énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.

Anexo 2: Top 39 departamentos de ciencias de la computación norteamericanos y los lenguajes de programación que emplean



Anexo 3: Funcionamiento del MTV de Django



Anexo 4: Comparación entre algunos *frameworks* en cuanto a cantidad de carpetas y ficheros

Framework	Cantidad de carpetas	Cantidad de ficheros
Ruby on Rails 4.1.5	35	149
Symfony 2	29	117
Django 1.7.1	1	4

Anexo 5: Comparación entre algunos *frameworks* y sus licencias

Framework	Última versión estable	Fecha de lanzamiento	Lenguaje	Licencia
OpenXava	4.7	02-04-2013	Java	LGPL
CodeIgniter	2.2.0	05-06-2014	PHP	OSLv3
Django	1.7.1	22-10-2014	Python	BSD

Anexo 6: Funcionalidades nativas en algunos *frameworks*

Framework	MVC	Migraciones de la BD	ORM	Framework de seguridad	Framework de plantillas	Framework de validación
OpenXava 4.7	Si	Si	De terceros	A través de plugins	La interfaz gráfica es automáticamente generada	Si
CodeIgniter 2.2.0	Si	Si	De terceros	A través de plugins	Si	Si
Django 1.7.1	Si	Si	Nativo (Django ORM)	Basado en ACL	Si, nativo (Django Template Language)	Si, nativo (Django Forms API)