

Universidad de Ciencias Informáticas

Facultad 4



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

Sistema de Control Docente de la Facultad 4

Autores: Dariel Corchado López del Castillo

Maikel Ruíz Pol

Tutor: Lic. Héctor Matías

Co-Tutor: M. Sc. Prof. Aux. Maritza Calaña Hernández

**La Habana, 12 de junio de 2015**

**“Año 54 de la Revolución Cubana.”**



*"Solo podemos ver poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer"*

*Alan Mathison Turing*

## **Declaración de Autoría:**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ de 2015.

\_\_\_\_\_  
Dariel Corchado López del Castillo

Autor

\_\_\_\_\_  
Lic. Héctor Matías

Tutor

\_\_\_\_\_  
Maikel Ruiz Pol

Autor

\_\_\_\_\_  
M. Sc. Prof. Aux. Maritza Calaña

Co-tutor

*Dedicatoria*

*A mi madre, que aunque no esté  
conmigo aquí, mucho luchó por verme  
hecho un hombre realizado, mi papa, mi  
hermana, por ser la razón de ser de mi  
vida.*

*Maikel*

*A mi madre, padre, novia y mis dos  
hermanos queridos que tanto han  
contribuido para llegar a conseguir lo  
que hoy soy.*

*Dariel*

## *Agradecimientos*

*De Maikel:*

*A mi mamá y mi papá por todo el sacrificio, amor, entrega, dedicación y confianza que siempre me han brindado y por esperar siempre lo mejor de mí.*

*A mi compañero de tesis Daríel, por ser mi apoyo más grande para llegar a la meta y uno de mis mejores amigos prácticamente desde que nacimos.*

*A todos los que de una forma y otra influyeron en mis resultados académicos en especial a Daynier, por ser mi repasador cuando no entendía y mis ojos cuando no veía en la pizarra.*

*A todos los que han sido mis compañeros de apartamento, en especial a Adrián, Ocleidy, Ángel Luis, Eddy, Orlando, Virgilio, Carlos, Abel, Raúl, Gino, Alberto, Pedro y demás compañeros que han jugado un papel importante en mi vida residencial.*

*A todo el claustro de profesores que me ayudó a obtener conocimientos básicos para hoy estar aquí como ingeniero, en especial a mis tutores Maritza Calaña Hernández y Hector Matías González, de veras muchas gracias a todos.*

*De Daríel*

*A mi madre, padre, a mi hermanita linda y a mi hermano por apoyarme tanto en la vida y especialmente en mi graduación.*

*A mi novia y toda su familia por su apoyo incondicional y por quererme como uno más en su familia.*

*Al resto de mi familia, primas, primos, tías, tíos, abuelos y abuelas que siempre me han brindado su cariño.*

*A mis amigos de la Universidad y del aula que de forma directa e indirecta han contribuido en que hoy sea ingeniero.*

*A mis compañeros de apartamento por ayudarme de una forma u otra en el desarrollo de la tesis.*

*A Albernova por la foto de las recomendaciones.*

*A los tutores Maritza y Héctor por todo su apoyo y asesoría. A los profesores que me impartieron clase a lo largo de la carrera.*

*Y finalmente a mi compañero de tesis Maikel primeramente por ser mi amigo de toda la vida y compañero de tesis.*

## **Resumen**

El Vicedecanato de Formación es el encargado de realizar la gestión y el control de los procesos docentes-metodológicos en la Facultad 4, entre los que se incluyen los controles a clase (CC) y la matrícula de asignaturas optativas y electivas (MOE) que se brindan a los estudiantes.

En la Universidad se emplea un sistema de gestión académica denominado Sistema de Gestión Universitaria pero este aún no da respuesta a todas las necesidades de gestión de información sobre los CC y la MOE, información que es de vital importancia para el trabajo de los vicedecanos docentes y jefes de departamentos. Estos dos procesos se realizan de forma manual por no contar con un sistema a nivel de universidad que permita su automatización, lo cual se torna más complejo en algunas ocasiones.

Por este motivo surge la necesidad de desarrollar un sistema informático para la gestión de controles a clases, asignaturas optativas y electivas, así como la matrícula de los estudiantes a estas asignaturas, brindando recomendaciones automatizadas. Un sistema que además genere reportes y facilite el trabajo al vicedecano docente.

En el presente trabajo diploma se explica todo el proceso de selección de las tecnologías y herramientas a utilizar que más se adecuaron a las características del proyecto y a las políticas de uso de software libre de la Universidad. Se documenta detalladamente el desarrollo del sistema informático propuesto, el cual abarca todas las fases que propone la metodología Extreme Programming.

## Índice

Introducción.....	1
Métodos de obtención de información.....	4
Capítulo 1.....	5
Fundamentación Teórica.....	5
Introducción .....	5
1.1.1 Sistemas de Gestión Académica .....	5
1.1.2 Sistema de Gestión Universitaria (SGU) .....	6
1.1.3 Sistemas en Línea para la Matrícula de Cursos .....	7
1.2 Generalidades de los Sistemas de Recomendación .....	8
1.2.1 Técnicas de Recomendación .....	10
1.3 Metodología de Desarrollo de Software .....	12
1.3.1 ¿Por qué seleccionar XP? .....	13
1.4 Lenguaje de programación. ....	14
1.4.1 Python .....	14
1.4.2 Java .....	15
1.5 Frameworks de desarrollo de PHP .....	16
1.5.1 Symfony.....	17
1.5.2 Yes It Is (Yii) .....	17
1.5.3 Zend Framework (ZF) .....	18
1.5.4 jQuery .....	19
1.5.5 Bootstrap .....	19
1.6 Gestor de Base de Datos.....	20
1.7 Herramientas de desarrollo.....	20
1.7.1 Herramientas para la creación de sistemas recomendadores.....	20
1.7.2 Weka .....	20
1.8 Herramienta de modelado.....	23
1.9 Herramientas de programación.....	23
1.9.1 IDE.....	23
1.9.2 Eclipse .....	23
1.9.3 NetBeans 8.0 .....	24
1.9.4 PhpStorm.....	24
1.9.5 Fundamentos de la selección.....	24
1.10 Servidor web.....	24
Conclusiones del Capítulo .....	25
Capítulo 2.....	26
Propuesta de Solución .....	26

Introducción .....	26
2.1 Usuarios del Sistema .....	26
2.2 Lista de reserva del producto.....	26
2.3 Modelo conceptual.....	28
2.3.1 Definiciones del Modelo Conceptual .....	28
2.4 Historias de Usuario (HU) .....	29
2.5 Planeación.....	33
2.5.1 Estimación de Esfuerzos por (HU) .....	33
2.5.2 Plan e iteraciones para entregables.....	34
2.6 Diseño de la Aplicación.....	35
2.6.1 Tarjetas CRC .....	35
2.7 Arquitectura .....	38
2.7.1 Patrón MVC .....	38
2.7.2 Patrones de Diseño .....	40
2.8 Modelo de datos .....	41
Conclusiones parciales.....	42
Capítulo 3. ....	43
Implementación y prueba. ....	43
Introducción .....	43
3.1 Implementación .....	43
3.2 Pruebas del sistema .....	47
3.2.1 Pruebas unitarias.....	48
3.2.1 Pruebas de aceptación .....	50
3.2.2 Prueba de recomendación .....	62
3.3 Estándares de codificación .....	63
Conclusiones .....	65
Conclusiones Generales.....	66
Recomendaciones .....	67
Referencias Bibliográficas .....	68
Anexos .....	<b>¡Error! Marcador no definido.</b>

## Introducción

Los avances tecnológicos alcanzados en el área del desarrollo de software han revolucionado en gran medida la forma de adquirir conocimientos que contribuyen al desarrollo gradual de la sociedad. Existen numerosos proyectos de software enfocados en dinamizar y automatizar procesos, logrando aliviar de cierta forma los esfuerzos humanos para alcanzar calidad, seguridad y eficiencia en cada estación de trabajo. En la Universidad de las Ciencias Informáticas se desarrollan trabajos que responden a necesidades reales de automatización de procesos. Estos surgen como respuesta a **situaciones problemáticas** como la que se plantea a continuación:

El Vicedecanato de Formación es el encargado de realizar la gestión y el control de los procesos docentes-metodológicos en la Facultad, entre los que se incluyen los controles a clase y la matrícula de asignaturas optativas y electivas que se brindan a los estudiantes.

El control se ha convertido en un proceso cotidiano de las instituciones de Educación Superior. El control a los procesos universitarios está dirigido a los resultados a alcanzar, a partir de lo planificado en los objetivos de trabajo a nivel institucional, que comprende procesos docentes y no docentes.

Según se establece en el Artículo 90 de la Resolución No. 210, por la que se aprobó el “Reglamento para el Trabajo Docente y Metodológico en la educación superior”, el control al proceso docente educativo es el medio fundamental para conocer la calidad de dicho proceso, evaluar sus resultados y dirigirlo hacia el cumplimiento de sus objetivos. [1]

El control docente presupone dirección, supervisión y control de varios procesos y actividades que forman parte del trabajo docente metodológico, de normativas, resultados docentes, cumplimiento del plan de estudio, entre otras actividades.

Para algunos autores el control docente está dirigido a la organización de esfuerzos, a la determinación de objetivos académicos, orientación de las líneas de trabajo docente - metodológico; a la adecuada normativa para alumnos, personal docente, administrativo con la finalidad de establecer objetivos y metas para un trabajo eficiente y exitoso. [2]

Los autores de la presente investigación asumen el control docente, en el contexto que se investiga, como un mecanismo de seguimiento, control y valoración

acerca del trabajo de un área docente. En este caso, para el Vicedecanato de Formación, Secretaría Docente y Departamentos Docentes de la Facultad 4.

Según la Resolución 210 del MES [3]:

ARTÍCULO 70: Las asignaturas optativas son aquellas que se incluyen en el plan de estudio y de entre las cuales el estudiante selecciona una cantidad determinada para cursar en forma obligatoria. Los contenidos de estas asignaturas tienen como propósito ampliar y actualizar a los estudiantes sobre temas científicos relacionados con la profesión.

ARTÍCULO 71: Las asignaturas electivas son aquellas 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.

En la Universidad se emplea un Sistema de Gestión Académica denominado Sistema de Gestión Universitaria [4] pero este aún no da respuesta a todas las necesidades de gestión de información sobre los controles a clase y la matrícula de asignaturas optativas y electivas que se brindan a los estudiantes. Esta información es de vital importancia para el trabajo de los vicedecanos docentes y jefes de departamentos.

Los vicedecanos de formación son los responsables de ofertar las asignaturas optativas y electivas de las Facultades. Estos dos procesos se realizan de forma manual por no contar con un sistema a nivel de universidad que permita su automatización, lo cual se torna más complejo en algunas ocasiones.

Por lo anteriormente expuesto se tiene como **problema a resolver**: ¿Cómo facilitar la matrícula de los estudiantes en las asignaturas optativas y electivas, así como la gestión de información de controles a clases y su cumplimiento en la Facultad 4?

El **objeto de estudio** son los sistemas de gestión académica, definiéndose como **campo de acción**: Los sistemas en línea para la matrícula de cursos con recomendación automática y la gestión de información de controles a clases y su cumplimiento en la Facultad 4.

Para solucionar el problema antes mencionado se propone como **objetivo general**: Desarrollar un sistema informático para la gestión de controles a clases, asignaturas optativas y electivas, así como la matrícula de los estudiantes a estas asignaturas, brindando recomendaciones automatizadas.

Para el cumplimiento del objetivo propuesto se plantean los siguientes **objetivos específicos:**

- ✓ Construir el marco teórico, relacionado con los aspectos fundamentales a tener en cuenta para el desarrollo del sistema propuesto.
- ✓ Realizar el análisis, diseño e implementación del sistema.
- ✓ Validar la solución propuesta, a través de los métodos definidos en la investigación.

Para el logro de estos objetivos se plantean las siguientes **tareas específicas:**

1. Caracterización del Sistema de Gestión Universitaria de la Universidad de Ciencias Informáticas.
2. Análisis acerca de los sistemas en línea para la matrícula de cursos con recomendación automática.
3. Caracterización de los sistemas recomendadores.
4. Selección de una metodología adecuada para desarrollar la solución propuesta.
5. Selección del Lenguaje de programación y framework de desarrollo.
6. Selección del gestor de base de datos a utilizar en la aplicación.
7. Selección de las herramientas a utilizar en la concepción del sistema.
8. Diagnóstico del estado de la gestión de información del trabajo metodológico de la Facultad 4 mediante una entrevista a la vicedecana de formación.
9. Entrevista a los directivos del Sistema de Gestión Universitaria para conocer que funcionalidades necesarias para el vicedecanato de formación no están implementadas.
10. Análisis de las técnicas de inteligencia artificial y algoritmos más usados en la concepción del tipo de recomendación a utilizar en la propuesta.
11. Diseño e implementación del sistema propuesto.
12. Validación de la solución sugerida y verificación de que se ajuste a las necesidades del vicedecanato de formación.

Como **hipótesis** se plantea lo siguiente:

La existencia de un sistema en línea para la gestión de los controles a clase, las asignaturas optativas o electivas, así como la matrícula de los estudiantes a las mismas, con recomendaciones automáticas, agilizará el trabajo del vicedecanato de formación de la Facultad 4 y mejorará la elección de asignaturas optativas y electivas por los estudiantes.

## **Métodos de obtención de información**

Para la obtención de la información se aplicaron diferentes métodos científicos, teóricos y empíricos. Se utilizó el método **Analítico-Sintético** para el estudio de la documentación referente a los sistemas de gestión académica, de matrícula de cursos en línea y recomendadores, así como las técnicas de inteligencia artificial empleadas en la confección de estos sistemas con el objetivo de encontrar una solución al problema a resolver. El método **Deductivo** permitió arribar a las conclusiones parciales para la selección de cada herramienta y tecnología a utilizar.

Como **método empírico** se utilizó la Entrevista. Se realizaron entrevistas a varias personas con basto conocimiento sobre el tema de los reportes que se generan en el Vicedecanato de Formación, las formas de gestión de las asignaturas optativas o electivas de la Facultad 4 y la forma en que se brindan los servicios web en la Universidad entre otras. Las mismas permitieron obtener información necesaria para realizar el diagnóstico. Las entrevistas se realizaron de forma presencial a la vicedecana de formación y a la secretaria de la Facultad 4, así como directivos autorizados para brindar información disponible en el Sistema de Gestión Universitaria. Fue una entrevista semi-estructurada, puesto que se realizaron preguntas no concebidas en el guión lo que dio dinamismo y fluidez en las conversaciones.

La tesis está estructurada en tres capítulos:

Capítulo 1: Se analiza y se exponen los elementos teóricos, con el objetivo de generar el marco teórico y describir los principales elementos de desarrollo, seleccionar las herramientas y tecnologías a utilizar.

Capítulo 2: Se describe la propuesta de solución tomando en cuenta los elementos que se definen en la fase de Exploración y Planificación. Se detallan las Historias de Usuario que recogen todas las funcionalidades a implementar. Además, se muestran otros artefactos propios de la metodología de desarrollo seleccionada.

Capítulo 3: Se definen las pautas de la implementación, así como el desarrollo de pruebas y validaciones del sistema, lo que permitirá establecer estándares de calidad para con los requisitos y funcionalidades que debe suplir la solución propuesta.

# Capítulo 1.

## Fundamentación Teórica.

### Introducción

En la actualidad, los cambios tecnológicos han hecho que los sistemas de software resulten cada vez más necesarios en la gestión de procesos que se realizan en las diferentes áreas, lo que implica muchas más exigencias en cuanto a tiempo de desarrollo y usabilidad. Atendiendo a la necesidad de apoyar y automatizar cada proceso surgen en las universidades sistemas de apoyo a la gestión de información y a la enseñanza - aprendizaje.

En el presente capítulo se muestran los resultados de las investigaciones realizadas acerca de los sistemas de gestión académica y los sistemas para la matrícula de cursos en línea, teniendo en cuenta sus principales características y conceptos. Se lleva a cabo un estudio sobre los tipos de recomendaciones que se utilizan en estas aplicaciones. Se hace una breve descripción de las metodologías, lenguajes de programación, herramientas y técnicas a utilizar en la propuesta.

#### 1.1.1 Sistemas de Gestión Académica

Antes de definir brevemente el término Sistema de Gestión Académica se debe señalar que, “un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización”. [5] Se define además, como un conjunto de componentes que interactúan entre sí, con el fin de apoyar las actividades que se realizan en una organización o para informatizar los procesos de trabajo que se efectúan dentro de esta. [6]

De lo anterior expresado, se puede decir que un sistema de gestión académica es un grupo de componentes estructurados informáticamente con el fin de mejorar los procesos y actividades docentes de instituciones educacionales.

A nivel mundial existen diversos sistemas que realizan el proceso de gestión académica en las universidades, ejemplo de esto son PRISMA: Sistema de Gestión Académica de la UPC(Universidad Politécnica de Cataluña) [7], SIU: Sistema de Información Universitario de la UNSJ (Universidad Nacional de San Juan) [8] y el sistema nombrado MINERVA de la UMA (Universidad de Málaga. [9] Estos sistemas gestionan los procesos necesarios para llevar a cabo las tareas relacionadas con la trayectoria académica de los alumnos teniendo en cuenta las características de cada

universidad. Sus principales funcionalidades son la preinscripción, matrícula, registro de evaluaciones, expedientes, generan estadísticas y brindan soporte a la toma de decisiones.

Cuba no está exenta del empleo de este tipo de aplicaciones. Su uso se ha extendido a la mayoría de las universidades del país. Ejemplo de esto son el Sistema para la Gestión Académica (GESTACAD) de la Universidad de Matanzas, la Universidad Virtual de Postgrado (IPLAC-Virtual), perteneciente al Ministerio de Educación (MINED), el Sistema de Gestión de la Nueva Universidad (SIGENU) empleado en la Universidad Agraria de la Habana y Cienfuegos, el Instituto Superior de Diseño, Instituto Superior de Relaciones Internacionales entre otros [10] y el Sistema de Gestión Universitaria (SGU) de la Universidad de Las Ciencias Informáticas. Estas aplicaciones son implementadas para satisfacer las necesidades propias de cada universidad brindándoles la posibilidad de automatizar y con ello agilizar muchos procesos docentes. A continuación se explica detalladamente el SGU por su gran importancia en esta investigación.

### **1.1.2 Sistema de Gestión Universitaria (SGU)**

El Sistema de Gestión Universitaria fue creado con el objetivo de mantener la mayor cantidad de información legalmente manejada por la universidad, centralizada para el uso de los encargados de manejarla constantemente. Muestra una interfaz común para todos los usuarios muy intuitiva para el intercambio con la misma. Brinda información necesaria para todas las facultades de la Universidad. Está compuesto por diferentes módulos entre los que se encuentran: Postgrado, Cooperación, Residencia, Ingreso, Investigación, Producción, Laboratorios, Biblioteca, Desarrollo, Tecnología, Extensión, Teleformación y Egreso.

En el módulo de Pregrado se gestiona toda la información referente a los estudiantes. Esta información no es más que el expediente del estudiante y su información como universitario. Dentro del expediente del estudiante se encuentra su plan de estudio, trámites docentes, registro de asignaturas y resumen de evaluaciones, en este último se registran todas las incidencias del estudiante a lo largo de los diferentes semestres.

Desde el punto de vista de obtención de información centralizada para los vicedecanatos de formación de las facultades es muy necesario el uso de esta herramienta, pero cabe señalar que se puede mejorar e incorporar algunas nuevas funcionalidades tales como: la gestión de matrícula, on-line, de las asignaturas

optativas o electivas y la gestión de los controles a clases, los cuáles son muy necesarios para el trabajo metodológico y a la hora de generar informes completos para la máxima dirección de la Universidad. Es por ello que se hace imprescindible implementar la propuesta.

### **1.1.3 Sistemas en Línea para la Matrícula de Cursos**

La educación a distancia es un sistema tecnológico de comunicación masiva y bidireccional que sustituye la interacción personal en el aula de profesor y alumno como medio preferente de enseñanza, acción sistémica y conjunta de diversos recursos didácticos y de apoyo de una organización tutorial, que propician el aprendizaje autónomo de los estudiantes. [11]

Las tendencias actuales de la educación a distancia señalan una educación más abierta, libre y orientada al uso de las tecnologías, mayormente con el uso de la web e internet. A raíz de ésta surgen los MOOC o Cursos en Línea Masivos y Abiertos, los cuales fueron creados por Dave Cormier y Bryan Alexander en el año 2008. MOOC es una clase apoyada por la web, diseñada para aceptar un gran número de participantes. [12]

Existen numerosas plataformas creadas para el empleo de los MOOC entre las que se puede mencionar, la plataforma EDX, la cual fue creada por la Universidad de Massachusetts y más tarde contó con la colaboración de la Universidad de Harvard. EDX posibilita la matrícula, on-line, a los diferentes cursos que se ofertan en las diferentes universidades. Otra de las plataformas más usadas en la actualidad es el COURSERA. La misma consta de numerosos cursos a distancia que se imparten en las más prestigiosas universidades.

Numerosas aplicaciones educativas que se encuentran en la web son desarrolladas empleando LMS (por sus siglas en inglés, Learning Management System).

Un LMS es un programa (aplicación de software) instalado en un servidor, que se emplea para administrar, distribuir y controlar las actividades de formación presencial o e-Learning de una institución u organización. Las principales funciones del LMS son: gestionar usuarios, recursos tales como materiales y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros. [13]

Entre los LMS más reconocidos se encuentra la plataforma de teleformación Moodle. Es una aplicación web gratuita que los educadores pueden utilizar para crear

cursos y darle un seguimiento detallado a las acciones del estudiante; cuestiones necesarias para dirigir un proceso de aprendizaje en línea.

Según su página web oficial, Moodle es una plataforma de aprendizaje diseñado para proporcionar a los educadores, administradores y estudiantes; un único sistema robusto, seguro e integrado para crear entornos de aprendizaje personalizados. [14]

La Universidad de las Ciencias Informáticas es un vivo ejemplo de la implementación de este tipo de aplicaciones. Se cuenta con varios portales educativos entre los que se destaca el Entorno Virtual de Aprendizaje (EVA). El EVA es un espacio de apoyo al proceso de formación de la carrera de Ingeniería en Ciencias Informáticas, donde los profesores pueden implementar estrategias de enseñanza-aprendizaje complementarias a las clases presenciales, así como diseñar cursos semi-presenciales o totalmente a distancia. Los estudiantes disponen de un poderoso medio en el cual pueden obtener, utilizar o compartir materiales didácticos. [15]

El EVA, en la Universidad de Ciencias Informáticas, se emplea solamente para publicar los contenidos de las asignaturas y no como sistema de gestión de matrícula, la matrícula a las asignaturas es responsabilidad del vicedecanato de formación. Esto no afecta las asignaturas ordinarias de la carrera de Ingeniería en Ciencias Informáticas porque todos los estudiantes deben cursarlas, sin embargo las asignaturas optativas y electivas son opcionales. La matrícula de estas asignaturas se hace de forma manual, sin el apoyo de las TIC, lo que hace complejo este proceso y necesaria la implementación de la solución propuesta.

Con el análisis de las soluciones anteriormente estudiadas se pudo detectar que la gestión de las asignaturas optativas o electivas es uno de los procesos a los que cuales se les puede dar mejoramiento pues no se encuentra un sistema que gestione todas las propuestas brindadas por la Universidad. Por otro lado sería de gran importancia implementar recomendaciones de estas asignaturas para los estudiantes, lo que implica el estudio de sistemas y técnicas de recomendación.

## **1.2 Generalidades de los Sistemas de Recomendación**

Los sistemas recomendadores son herramientas de software desarrolladas para ayudar a los usuarios en el proceso de toma de decisiones con respecto a un tema específico. Estos sistemas comienzan a emerger a mediados de los años 90, como solución ante el desbordamiento de información que desde aquel entonces ya estaban ocasionando a nivel global las nuevas tecnologías de la información y las comunicaciones (TIC) y en particular la World Wide Web (WWW), haciendo difícil para

una persona obtener la mejor información dentro de este cúmulo, al no tener esta, en muchos casos, detalles de cada una de las alternativas a seleccionar. [12]

Actualmente, el uso de estos sistemas se ha extendido a una gran variedad de aplicaciones, destacándose las de E-commerce y las de E-learning. El término "ítem", se utiliza para denominar el objeto que el sistema debe recomendar, por ejemplo, canciones, películas o noticias. Por lo general un sistema recomendador se especializa en un ítem específico, por lo cual elementos como el diseño, la interfaz gráfica y el algoritmo usado para generar las recomendaciones son personalizados para obtener sugerencias efectivas para dicho ítem. De manera general, se puede decir que los datos que utilizan los sistemas recomendadores pueden separarse en tres grupos:

- ✓ Ítems: Como se mencionó anteriormente son los elementos que son recomendados. El valor de un ítem puede ser positivo si resulta útil para el usuario, así como puede ser negativo si el usuario considera que fue una mala decisión seleccionarlo. Los ítems tienen además, características que pueden indicar cuan necesario resultan para un usuario determinado.
- ✓ Usuarios: Un usuario puede tener diversidad de metas o preferencias cuando accede a un software. Para poder ofrecer las recomendaciones adecuadas el sistema debe explorar toda la información disponible acerca de dicho usuario, ya sea la que inicialmente se haya solicitado o la que se va recopilando a lo largo de la interacción de la persona con el sistema.
- ✓ Transacciones: Genéricamente se le denomina transacciones a la interacción de los usuarios con el sistema de recomendación. De esta manera se puede almacenar información acerca de las preferencias de los usuarios sobre los ítems. Una de las formas más frecuentes y populares de ver las transacciones es a través de los "ratings". Este término se refiere a la evaluación que realiza un individuo sobre un ítem que es sometido a su consideración, de esta forma se puede almacenar el criterio que tiene el usuario sobre un ítem determinado. [13] A partir de estas generalidades se impone ampliar conocimientos acerca de técnicas empleadas para la recomendación.

## **1.2.1 Técnicas de Recomendación**

### **Basados en Contenido (RBC):**

Los recomendadores basados en el contenido son aquellos sistemas que recomiendan los ítems a los usuarios basándose exclusivamente en la descripción del ítem y de un perfil que contenga los intereses del propio usuario. Este tipo de recomendador determina cuán relacionado está un objeto con un determinado usuario, para lo cual utiliza la medida de similitud. Esto le permite al sistema recomendar el objeto, partiendo de la descripción del mismo que mejor se adapte a las características de un usuario recogidas en su perfil. Dicho perfil puede ser creado de manera explícita, preguntando directamente al usuario por sus preferencias. La información puede ser recogida además de manera indirecta, considerando elementos que revelen las preferencias del usuario, entre estos elementos se pueden encontrar: sitios visitados, tipos de archivos descargados, tiempo de permanencia en un sitio y cualquier otra información que resulte útil.

Una de las ventajas más importantes de este tipo de sistema recomendador es que no se basa en las acciones previas del usuario, siempre lo hará basándose en la descripción del objeto y las preferencias del usuario. La principal desventaja es que las recomendaciones serán muy parecidas, ya que tiene como base los mismos datos. [7]

### **Filtrado Colaborativo (RFC):**

El Filtrado Colaborativo (RFC) es una técnica utilizada por algunos sistemas recomendadores. Tiene como base identificar personas con similitudes en sus preferencias para así crear las recomendaciones. Este puede ser implementado basando en ítems o basado en usuarios. Con el RFC los usuarios expresan sus intereses a partir de algunos ítems que se les presentan, ayudando al sistema, de esta manera, a crear un perfil. El sistema compara el perfil de este con los restantes usuarios construyendo una lista de usuarios con similitudes, la cual se puede denominar como “usuarios más cercanos”. Combinando esta lista, se devuelve un listado de recomendaciones, conteniendo los ítems mejor evaluados por estos usuarios y que no han sido aún evaluados por el nuevo individuo.

Para que este sistema tenga éxito es necesario que sea utilizado por varios usuarios, para que cuando ingrese uno nuevo pueda encontrar grupos de personas con intereses comunes, esto sería una forma sencilla de obtener las preferencias, para

que la comparación de los mismos sea más sencilla y así disponer de algoritmos capaces de relacionar a los usuarios. [16]

Su principal desventaja es la de nuevos ítems. Se refiere a la situación en la cual un ítem no puede ser recomendado a menos que haya sido valorado por un número sustancial de usuarios. Sin información sobre nuevos ítems, no hay manera de recomendarlo. [17]

### **Sistemas Recomendadores Híbridos (RH)**

Aún conociendo las características de los tipos de sistemas recomendadores existentes se decide no aplicar alguna de estas técnicas por separado pues para llegar a obtener una solución eficiente en cuanto a recomendación fue necesario elegir lo mejor de cada técnica lo que se conoce como sistema híbrido. Estos sistemas combinan dos o más técnicas de la recomendación para ganar un funcionamiento mejor con menos desventajas, aprovechando los puntos fuertes de estas técnicas. Generalmente, el filtrado colaborativo se combina con otra técnica en una tentativa de evitar el problema de nuevos ítems". Según lo indicado anteriormente, Recomendadores Basados en Contenido (RBC) proporcionan recomendaciones basadas en la correlación "ítem-ítem ", mientras que los Filtrado Colaborativo (RFC) utilizan la correlación "gente-gente".

Para crear un sistema híbrido de RBC y RFC, se utilizan los perfiles de usuarios basados en el análisis del contenido, y se comparan directamente estos perfiles para determinar usuarios similares para la recomendación de colaboración. Los Recomendadores Híbridos (RH) evitan las limitaciones de sistemas de RBC y RFC.

Un RH genera típicamente un modelo del usuario activo supervisando el comportamiento del usuario o analizando su interés o feedback. El modelo generado del usuario se combina generalmente con las necesidades de información del usuario y se presenta a un motor de búsqueda un pedido de recomendaciones. Además, el sistema mantiene un espacio de perfiles de otros usuarios, haciendo posible el uso de RFC para refinar más la lista de recomendaciones seleccionada. [17]

Luego de analizados estos tipos de sistemas de recomendación se decide emplear los filtros colaborativos, centrándose en los filtros colaborativos basados en ítems, pues se está en presencia de más usuarios que asignaturas optativas o electivas lo que permite obtener más similitudes en las preferencias de los usuarios

con respecto a dichas asignaturas. En el caso del sistema propuesto, el usuario tendrá una noción de las asignaturas que les serán recomendadas, esto hace que tomen mayor peso las recomendaciones basándose en los ítems y no en los perfiles de cada usuario. Por otra parte, en las librerías de Mahout se encuentran implementados los filtros escogidos de forma sencilla, basados en diferentes medidas (Pearson, Euclidean, Log-likelihood y Tanimoto) que dan solución al problema de la recomendación. También se escoge esta técnica porque estas métricas anteriormente mencionadas pueden ser evaluadas de acuerdo a la exactitud de la recomendación mejorando así la calidad de la misma.

### **1.3 Metodología de Desarrollo de Software**

Una metodología de desarrollo es la guía que contiene aquellos pasos y procedimientos que deben seguirse para desarrollar un software[18]. Estas se clasifican en tradicionales y ágiles.

Las Metodologías Tradicionales son aquellas que ponen mayor énfasis en la planificación y control del proyecto y en la especificación precisa de requisitos y modelado. Las Metodologías Ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso [19].

Para la elección de la metodología se analizan tres de las más utilizadas en la Universidad. Las mismas son Proceso Unificado de Desarrollo (RUP) que es una metodología tradicional y las metodologías ágiles Extreme Programming (XP) y Scrum.

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte (personalización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de modelado. Realiza procesos de continuas pruebas y retroalimentación. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto. El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos

(documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas [20].

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. Las iteraciones en general tienen una duración entre 2 y 4 semanas. [20]

Por su parte XP es lo más destacado de los procesos ágiles de desarrollo de software formulados por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. [20] Esta metodología considera los cambios de requisitos sobre la marcha como un aspecto natural, inevitable e incluso deseable para el desarrollo de proyectos. Plantea que ser capaces de adaptarse a los cambios de requerimientos que puedan surgir sobre cualquier etapa de desarrollo constituye una aproximación mejor y más realista del producto que necesita el cliente para satisfacer sus necesidades. [20]

### **1.3.1 ¿Por qué seleccionar XP?**

Luego de analizar estas metodologías se escoge XP por las siguientes razones:

- ✓ Teniendo en cuenta las características del equipo de desarrollo, el cual está conformado por dos individuos, resulta apropiado la implementación de prácticas correspondientes a la metodología XP, tales como la programación en pares, la permanente interacción con el cliente, la refactorización constante, entre otras ventajas que brinda dicha metodología.
- ✓ Aunque RUP es un proceso que define actividades más detalladas de cara al desarrollo, más formal, de proyectos de software, no posee mucha interacción con el cliente dejándolo en una posición secundaria donde podría sentirse incómodo.
- ✓ A diferencia de XP, RUP posee una muy compleja evaluación de riesgos lo que no permite, a grandes rasgos, tener absoluta seguridad que el producto se esté desarrollando con la calidad requerida.
- ✓ La aplicación que se desarrolla es muy práctica y guarda mucha relación con los procesos que realiza el cliente en su quehacer diario, por lo que no requiere demasiada documentación sino muy buena calidad para su puesta en práctica.

- ✓ XP reduce el número de cambios necesarios para el proyecto lo que permite mejorar el tiempo de implementación. Por otra parte existe propiedad de código compartido y simplicidad en el código lo que permite que este sea más entendible y modificable por los desarrolladores.

SCRUM brinda muchas facilidades pero esta metodología no se selecciona por que no se cuenta con un jefe de proyecto (Scrum master) que oriente al equipo de desarrollo, siendo este uno de los roles más importantes en el proceso. Además no presenta alguna de las prácticas que muestra XP como es la programación en parejas y el modelo de 40 horas de trabajo semanales, las que considera el equipo son de vital importancia para el desarrollo en equipo.

A raíz del análisis anteriormente realizado y teniendo en cuenta que el cliente es también tutor del proyecto, se puede afirmar que el sistema a desarrollar puede culminarse con resultados exitosos aplicando la metodología XP.

#### **1.4 Lenguaje de programación.**

Para la selección del lenguaje de programación se tuvo en cuenta los siguientes parámetros:

- ✓ Debe permitir desarrollar aplicaciones web.
- ✓ Debe ser distribuido bajo una licencia libre para lograr una correspondencia con las políticas de la Universidad.
- ✓ De permitir la implementación del patrón de diseño Modelo-Vista-Controlador (MVC) porque separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos [18], filosofía ideal para implementar la solución propuesta.

Luego de analizados estos parámetros se designan como posibles lenguajes a utilizar Python, Java y PHP.

##### **1.4.1 Python**

Es un lenguaje de programación integrado cuya filosofía hace hincapié en una sintaxis limpia y código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Posee una licencia de código abierto,

denominada Python Software Foundation License [21], que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores. Python puede incluirse en aplicaciones que necesitan una interfaz programable [22].

### **1.4.2 Java**

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos, basado en clases y destinado a correr en cualquier plataforma [23]. Sun Microsystems describe a Java como simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico. [24] Se ha desarrollado unido a Internet y es por ello que es ampliamente utilizado en la construcción tanto de sistemas distribuidos, como de complejas aplicaciones de gestión. El hecho de la dependencia de una máquina virtual, como mismo constituye una ventaja, es también una desventaja. El depender de un intermediario para ejecutar los programas hace a estos más lentos y los vuelve dependientes de la exactitud de la mencionada máquina virtual: en ocasiones el código está bien escrito, pero el programa no trabaja correctamente debido a errores de la máquina virtual.

### **1.4.3 PHP (Hypertext Pre-processores)**

Es un lenguaje de programación de código del lado del servidor. El código es interpretado por un servidor web con un módulo de procesador de PHP generando esta la página web. Puede ser usado en casi todos los servidores web y sistemas operativos sin costo alguno. [25]

#### **Principales usos de PHP:**

Este lenguaje es utilizado para la programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC (Open DataBase Connectivity), lo que amplía en gran medida sus posibilidades de conexión.

- ✓ Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- ✓ Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS.

### **Ventajas de PHP:**

- ✓ PHP permite ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS, y puede interactuar con los servidores web más populares.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad tales como MySQL, PostgreSQL, Oracle, ODBC (Open DataBase Connectivity), IBM DB2, Microsoft SQL Server y SQLite, lo cual permite la creación de aplicaciones web muy robustas.
- ✓ Capacidad de leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- ✓ Posibilidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- ✓ Posee muy buena documentación en su página oficial [www.php.net](http://www.php.net).
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite las técnicas de Programación Orientada a Objetos (POO). [25]

#### **1.4.4 Fundamentación de la selección**

Cualquiera de las tres propuestas anteriormente analizadas es factible para utilizar en la implementación de la solución. Sin embargo, se decide utilizar el lenguaje PHP por las siguientes razones:

- ✓ Es un lenguaje cuya línea de expansión y desarrollo se corresponde con la política trazada por la UCI y por el país.
- ✓ Posee bibliotecas para integrarse a aplicaciones desarrolladas en otros lenguajes.
- ✓ Existe una amplia documentación referente al lenguaje y los desarrolladores tienen dominio del mismo.

#### **1.5 Frameworks de desarrollo de PHP**

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. El propósito de los framework es permitir a los desarrolladores construir aplicaciones web y centrarse en los aspectos interesantes, aliviando la típica tarea repetitiva asociada con patrones comunes de desarrollo web. La mayoría de los framework de aplicaciones web proporcionan los tipos de funcionalidad básica común, tales como sistemas de templates (plantillas),

manejo de sesiones de usuario, interfaces comunes con el disco o el almacenamiento en base de datos de contenido cacheado, y persistencia de datos. Normalmente los framework de aplicación web promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos [6].

Existen muchos tipos de framework PHP entre ellos resaltan Symfony, Yii y Zend Framework.

### **1.5.1 Symfony**

Symfony es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional [26].

#### **Principales características de Symfony**

- ✓ Utiliza el patrón de diseño Modelo Vista Controlador (MVC).
- ✓ Su código, y el de todos los componentes y librerías que incluye, se publican bajo la licencia MIT de software libre.
- ✓ La documentación del proyecto también es libre e incluye varios libros y decenas de tutoriales específicos.
- ✓ Aprender a programar con Symfony permite acceder a una gran variedad de proyectos: el framework Symfony2 para crear aplicaciones complejas, el micro framework Silex para sitios web sencillos y los componentes Symfony para otras aplicaciones PHP.
- ✓ Según GitHub, Symfony es el proyecto PHP más activo, lo que garantiza que nunca te quedarás atrapado en un proyecto sin actividad.
- ✓ Aunque en su desarrollo participan cientos de programadores de todo el mundo, las decisiones técnicas importantes siempre las toma Fabien Potencier. Esto evita el peligro de que surjan forks absurdos y la comunidad se fragmente.
- ✓ Los componentes de Symfony son tan útiles y están tan probados, que proyectos tan gigantescos como Drupal 8 están contruidos con ellos.

### **1.5.2 Yes It Is (Yii)**

Yii es un framework orientado a objetos, software libre, de alto rendimiento basado en componentes, PHP y framework de aplicaciones web. Yii se pronuncia en

español como se escribe y es un acrónimo para "Yes It Is!" (En español: ¡Sí lo es!).  
[27]

### **Principales características de Yii**

- ✓ Este framework utiliza el patrón de diseño Modelo Vista Controlador (MVC).
- ✓ Database Access Objects (DAO), query builder, Active Record y migración de base de datos.
- ✓ Integración con jQuery.
- ✓ Entradas de formulario y validación.
- ✓ Widgets de Ajax, como autocompletado de campos de texto y demás.
- ✓ Soporte de Autenticación incorporado. Además soporta autorización vía role-based access control (RBAC) jerárquico.
- ✓ Personalización de aspectos y temas.

### **1.5.3 Zend Framework (ZF)**

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. En la estructura de los componentes de ZF; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad) [28]

En el estudio de los framework se decide elegir Symfony en su versión 2.5 para la implementación de la solución pues tiene grandes opciones para adaptarse a diferentes tipos de sistemas de bases de datos, es muy utilizado en la Universidad de las Ciencias informáticas (UCI) y los desarrolladores se encuentran capacitados. Por otra parte es necesario señalar que este marco de trabajo consta de una comunidad de desarrollo activa que trabaja constantemente en mejorar cada vez más las comodidades de desarrollo, esto hace que surjan nuevas funcionalidades, lo que favorece la propuesta a implementar en cuanto a la utilización de tecnologías novedosas.

### **1.5.4 jQuery**

Uno de los marcos de trabajo, correspondiente a JavaScript, utilizados para el desarrollo del sistema es el jQuery, el cual fue creado inicialmente por John Resig. Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Fue presentado el 14 de enero de 2006 en el BarCamp NYC [29]

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio [29]. Para el desarrollo de la solución se utiliza esta librería dada las funcionalidades que implementa, el manejo de eventos y el dinamismo que ofrece, además de ser libre y de código abierto, bajo dos tipos de licencias.

### **1.5.5 Bootstrap**

Se seleccionó para el diseño de la interfaz de usuario Bootstrap que no es más que una colección gratuita y de código abierto de herramientas para la creación de sitios y aplicaciones web. Su objetivo es proporcionar una biblioteca refinada, bien documentada y extensa de componentes de diseño flexibles creados con HTML, CSS y JavaScript para que otros puedan construir e innovar. [30]

Este framework fue escogido por que trae incorporado CSS y LESS (lenguaje de hojas de estilo) entre otros, lo que facilita crear interfaces rápidamente supliendo el engorroso trabajo de hacerlo con CSS desde cero, usa componentes de JavaScript para mejorar la programación, maneja sintaxis HTML lo que hace que el diseño sea más ágil, brinda un útil diseño en malla, facilitando la división del contenido del sistema haciéndolo flexible a todo dispositivo que pueda acceder al sistema, por último se impone hablar de la gran documentación que existe de dicho framework, bien detallada, ejemplificada y fácil para su uso. [36]

## **1.6 Gestor de Base de Datos**

Para selección del gestor de bases de datos (GBD) que se emplea en la implementación de la aplicación se tomaron en cuenta MySQL y PostgreSQL pues ambos contienen componentes de desarrollo que aportan importantes ventajas a las tecnologías libres.

Aun conociendo que MySQL es más veloz para realizar consultas y conexiones a las BD, consume muy pocos recursos, tiene mayor rendimiento, mejores utilidades de administración, mejor integración con PHP entre otras características, PostgreSQL ofrece una mejor garantía de integridad referencial, por su arquitectura de diseño, funciona muy bien al aumentar el número de CPUs y la calidad de memoria RAM, soporta transacciones y desde la versión 7.0, claves ajenas, tiene mejor soporte para triggers y procedimientos en el servidor entre otras características [31].

En consecuencia con las ventajas comparativas de ambos gestores de bases de datos y teniendo en cuenta las líneas tecnológicas trazadas por la UCI, se selecciona el PostgreSQL para el desarrollo de la aplicación.

## **1.7 Herramientas de desarrollo**

### **1.7.1 Herramientas para la creación de sistemas recomendadores.**

Para la creación de sistemas recomendadores existe un gran número de herramientas. Estas implementan una gran variedad de algoritmos para la minería de datos como por ejemplo los algoritmos de clusterización, clasificación y de recomendación. A continuación se exponen características de dos de estas herramientas: Weka y Mahout con el objetivo de seleccionar una posteriormente.

### **1.7.2 Weka**

“La herramienta Weka fue desarrollada en 1993 en la Universidad de Waikato de Nueva Zelanda utilizando TCL/TK y C. y en el año de 1997 se reescribió su código al lenguaje Java incluyendo implementaciones de algoritmos de modelado.

El paquete Weka contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue un front-end en TCL/TK para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para preprocesamiento de datos

desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación.”

**Los puntos fuertes de Weka son:**

- ✓ Está disponible libremente bajo la licencia pública general de GNU.
- ✓ Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- ✓ Contiene una extensa colección de técnicas para pre-procesamiento de datos y modelado.
- ✓ Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

Weka soporta varias tareas estándar de minería de datos, especialmente, pre-procesamiento de datos, clustering, clasificación, regresión, visualización y selección. Todas las técnicas de Weka se fundamentan en la presunción que los datos están disponibles en un archivo plano (flat file, archivo de texto sin formato) o una relación, en la que cada registro de datos está descrito por un número fijo de atributos (normalmente numéricos o nominales, aunque también se soportan otros tipos). Weka también proporciona acceso a bases de datos vía SQL gracias a la conexión JDBC (Java Database Connectivity) y puede procesar el resultado devuelto por una consulta hecha a la base de datos. No puede realizar minería de datos multi-relacional, pero existen aplicaciones que pueden convertir una colección de tablas relacionadas de una base de datos en una única tabla que ya puede ser procesada con Weka. Esta herramienta se ha vuelto tan importante en la actualidad que se le dedica una parte completa del libro *Data Mining: Practical Machine Learning Tools and Techniques* [32].

### **1.7.3 Mahout**

El Apache Mahout es un nuevo proyecto de código abierto fundado por la Apache Software Foundation (ASF) con la meta primaria de crear algoritmos escalables de aprendizaje de máquina gratis bajo la licencia apache. Mahout contiene implementaciones para clustering, categorización, filtros colaborativos (RFC) y

programación evolutiva. Además, incorpora la biblioteca Apache Hadoop para permitirle a Mahout escalar eficazmente en la nube [33]

### **La historia de Mahout**

El proyecto Mahout comenzó por varias personas involucrado en la Apache Licence (búsqueda de código abierto), comunidad con un interés activo en el aprendizaje de máquina y un deseo para implementaciones robustas, bien documentadas, escalables algoritmos de aprendizajes de máquina comunes para clusterización y la clasificación. La comunidad fue inicialmente conducida por Ngetal's paper "MapReduce for Machine Learning on Multicore" pero desde entonces ha evolucionado a cubrir los enfoques de aprendizaje de máquinas mucho más amplios [34].

### **Mahout también tiene como objetivo:**

- ✓ Construir y apoyar a una comunidad de usuarios y contribuyentes de tal manera que el código sobrevive a la participación de cualquier contribuyente en particular, empresa o la financiación de una universidad.
- ✓ Concentrarse en el mundo real, los casos de uso práctico en oposición a la investigación bleeding-edge o técnicas no probadas.
- ✓ Proporcionar documentación de calidad y ejemplos.

### **Mahout posee:**

- ✓ Varias implementaciones de clustering incluyendo K-medias, KMeans difusos, Canopy, Dirichlet y media-Shift.
- ✓ Implementaciones de clasificación basadas clasificadores Bayesianos.
- ✓ Capacidades de función de fitness distribuidos para la programación evolutiva.
- ✓ Bibliotecas de Matrices y vectores.
- ✓ Ejemplos de todos los algoritmos anteriores.
- ✓ La construcción de un motor de recomendación para FC

### **Motor de recomendación**

Mahout actualmente proporciona herramientas para la construcción de un motor de recomendación a través de la biblioteca de Taste - un motor rápido y flexible para RFC. Taste soporta tanto las recomendaciones basadas en el usuario y basadas

en ítems y viene con muchas opciones para la formulación de recomendaciones, así como las interfaces para definir recomendaciones personalizadas. Tase consta de cinco componentes principales que trabajan con los usuarios, ítems y preferencias:

1. DataModel: Almacenamiento para los usuarios, objetos y preferencias
2. UserSimilarity: Interfaz que define la similitud entre dos usuarios
3. ItemSimilarity: Interfaz que define la similitud entre dos elementos
4. Recommendation: Interfaz para proporcionar recomendaciones
5. UserNeighborhood: Interfaz para el cálculo de una vecindad de los usuarios similares que puede entonces ser utilizado por los Recomendadores.

## **1.8 Herramienta de modelado**

Como herramienta de modelado se selecciona el Visual Paradigm porque es una herramienta profesional que posibilita modelar el ciclo de vida del desarrollo de software: Análisis y Diseño, Implementación, Pruebas y Despliegue. Posee una serie de características entre las que incluye generación de código. Además es capaz de generar diagramas de ingeniería directa e inversa desde el código y permite el control de versiones. Presenta una interfaz amigable y muy intuitiva para el modelado de diagramas. Posee numerosa documentación, en cuanto a tutoriales, libros, blogs, demostraciones interactivas y proyectos.

## **1.9 Herramientas de programación**

### **1.9.1 IDE**

Un Entorno Integrado de Desarrollo (IDE) es un conjunto de aplicaciones informáticas que consisten en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede dedicarse exclusivamente a un lenguaje de programación o bien puede utilizarse en varios lenguajes. A continuación se analiza Eclipse, Netbeans y PhpStorm, tres de los IDEs más populares con el objetivo de seleccionar el más adecuado para el desarrollo del sistema propuesto.

### **1.9.2 Eclipse**

Eclipse es un entorno de desarrollo integrado de código abierto. Posee una funcionalidad muy grande, pero esa funcionalidad es muy genérica. Permite que nuevos componentes puedan utilizar distintos tipos de contenido, para realizar determinadas tareas con contenidos existentes. La plataforma Eclipse permite descubrir, e invocar funcionalidad implementada en componentes llamados plug-ins. Contiene funciones que permiten manipular diferentes contenidos como son HTML,

Java, C, JSP, EJB, XML, y GIF. Facilita una integración transparente entre todas las herramientas y tipos de contenidos sin tener en cuenta al proveedor. Proporciona entornos de desarrollo gráfico (GUI por sus siglas en inglés) o no gráficos. Permite ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux. [35]

### **1.9.3 NetBeans 8.0**

Es un software multiplataforma, de código abierto (open-source), libre bajo licencia GPL (General Public License) y gratuito sin restricciones de uso, aunque fue creado principalmente para Java, soporta PHP. Es iterativo, ofrece plantillas de código y la generación, la refactorización, información sobre herramientas de parámetros, consejos y soluciones rápidas (aplicación de métodos abstractos). Tiene un atractivo balance entre la interfaz con múltiples opciones y un aceptable completamiento de código (33). Además soporta gran cantidad de módulos para el desarrollo con frameworks como Symfony en todas sus versiones.

### **1.9.4 PhpStorm**

Es uno de los entornos de programación más completos de la actualidad, permite editar código no solo PHP sino también HTML, CSS, JavaScript, entre otros. Algo que destaca en PhpStorm es la ejecución de nuestro código en la misma interfaz del IDE. Así como también la interpretación y visualización inmediata de código PHP en los navegadores web más populares como Google Chrome, Mozilla Firefox e Internet Explorer. También brinda soporte para el trabajo con muchos frameworks entre los que se encuentra Symfony2. [36]

### **1.9.5 Fundamentos de la selección**

El entorno de desarrollo seleccionado es NetBeans 8.0, por todas las funcionalidades y la potencia que este posee a la hora de trabajar con el lenguaje de programación PHP y el framework Symfony2. Además, por ser una aplicación libre, multiplataforma y principalmente porque el equipo de desarrollo tiene varios años de experiencia en el empleo de esta herramienta.

## **1.10 Servidor web**

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada) [37]

Como servidor web se emplea el Apache en su versión 2.4 para montar el sistema. Este servidor fue seleccionado por ser libre, de código abierto y multiplataforma, lo que lo hace prácticamente universal y con uso muy extendido en la universidad. También por ser una tecnología gratuita de código fuente abierto y altamente configurable.

### **Conclusiones del Capítulo**

En el presente capítulo se realizó un estudio de los sistemas de gestión académica, de matrícula de cursos online y de los sistemas recomendadores más empleados en la actualidad, obteniendo las bases teóricas para la implementación de la solución propuesta, la cual utilizará algunas funcionalidades de cada uno de estos sistemas. Se resuelve utilizar como metodología de desarrollo XP, lenguaje de programación PHP 5.3, framework de desarrollo Symfony2 2.5, como servidor web Apache 2.2, sistema gestor de bases de datos PostgreSQL 9.3, como herramienta de modelado el Visual Paradigm 8.0, como herramienta para la programación el Netbean 8.0 y finalmente y no menos importante se elige la técnica de filtrados colaborativos basados en ítems mediante funciones métricas de evaluación y recomendación.

## Capítulo 2.

# Propuesta de Solución

### Introducción

En este capítulo se presenta una propuesta de solución del Sistema de Control Docente de la Facultad 4 con el cual se puede brindar servicios de matrículas a las asignaturas optativas o electivas, y también se brindará la posibilidad de generar reportes de controles a clases para el Vicedecanato de Formación. Se muestran artefactos propios de la metodología escogida (XP), tales como las Historias de Usuario con las que se guía el proceso de desarrollo, las tarjetas CRC como metodología para el diseño del paradigma orientado a objetos [38]. Por otra parte se hace referencia a los usuarios del sistema, la arquitectura, el modelo de datos del mismo, y también se muestra la lista de reserva del producto, modelo conceptual e iteraciones para la producción de entregables del sistema.

### 2.1 Usuarios del Sistema

Los usuarios tomados en cuenta para interactuar con el sistema son los siguientes:

**Administrador:** Es quien dirige, en todo su dominio, el sistema, en este caso la vicedecana de formación, puede ver, modificar, añadir y eliminar toda la información que fluye en la aplicación.

**Jefe de Departamento:** Tiene la posibilidad de actualizar la información del departamento: información de los profesores de acuerdo a categoría docente, asignatura que imparte y responsabilidad. También maneja la información relacionada con los controles a clase.

**Secretaria:** Es la encargada de matricular, junto a la vicedecana de formación, a los estudiantes en el sistema y actualizar la información si aprobó o no la asignatura.

**Estudiante:** Puede matricularse en las asignaturas optativas y electivas, ver detalles de las mismas, observar los estudiantes matriculados y dar puntuación a las asignaturas.

### 2.2 Lista de reserva del producto

Con motivo de establecer una organización a la hora de desarrollar un proyecto de software se realiza la lista de reserva de productos. Este artefacto típico de la

metodología XP constituye una opción para dar prioridad a las funcionalidades a implementar en un proyecto. Es por ello que se muestra a continuación la lista de reserva de producto confeccionada para el desarrollo de las funcionalidades presentes en la propuesta de solución.

Tabla 1: Lista de reserva de producto

Ítem	Descripción	Estimación	Estimado por
Prioridad: Alta			
1	Listar Departamento	1	Programador
2	Gestionar control a clase	1	Programador
3	Desglose de controles por asignaturas	1	Analista
4	Desglose de controles por Departamentos	1	Programador
5	Totales de controles por facultad	1	Analista
6	Gestionar Asignaturas Optativas o Electivas	2	Analista
7	Matricular Estudiantes	1	Analista
8	Sugerir Asignatura Optativa o Electiva	2	Programador
Prioridad: Media			
9	Añadir profesor a departamento	0.5	Programador
10	Nuevo Departamento	0.5	Programador
11	Ver detalles de departamento	0.5	Analista
12	Editar departamento	0.5	Analista
13	Gestionar Usuario	0.5	Programador
Prioridad: Baja			

14	Otorgar rating	1	Analista
15	Profesores evaluados de mal	0.5	Analista
16	Generar PDF	0.5	Programador
17	Autenticar Usuario	0.5	Programador

## 2.3 Modelo conceptual

Una de las formas de poder llegar al entendimiento de las funcionalidades que contiene la solución propuesta es a través del modelo conceptual el cual no es más que una descripción de las características de los datos del sistema, los atributos y relaciones de interés para el negocio a representar [39]. Cada modelo conceptual puede tener un mayor o menor grado de precisión de acuerdo con la etapa de estudio en que se encuentre el negocio o función a representar. Según Jesús Barranco de Areba [39] existen modelos conceptuales globales y modelos conceptuales detallados. A continuación se muestra un modelo conceptual global asociado al sistema implementado y correspondiente al estudio de las entidades y sus relaciones, como resultado de un estudio estratégico previo a la estructuración de la misma.

### 2.3.1 Definiciones del Modelo Conceptual

Antes de presentar el modelo conceptual asociado al sistema, es necesario puntualizar las definiciones más importantes que interactúan en él, lo que se presenta a continuación:

Vicedecanato de Formación: área de trabajo donde se realizan todos los procesos gestión de información del trabajo metodológico y matrícula de asignaturas optativas o electivas.

Trabajo Docente Metodológicos: son las actividades docentes que se realizan en la Facultad.

Controles a clase: son visitas que se realizan a los profesores donde se emite una evaluación.

Profesor: es quien imparte las asignaturas tanto ordinarias como optativas o electivas.

Jefe: es la máxima dirección de los departamentos o las asignaturas.

Departamento: área donde se gestiona la información de los controles a clases de las asignaturas.

Asignatura: son los elementos para los que se obtienen reportes por cada departamento.

Asignaturas optativas o electivas: son los ítems que se van a recomendar para la matrícula de los estudiantes.

Estudiante: es quien se matricula en las asignaturas optativas o electivas.

Reporte: es la información que se genera de cada control a clase.

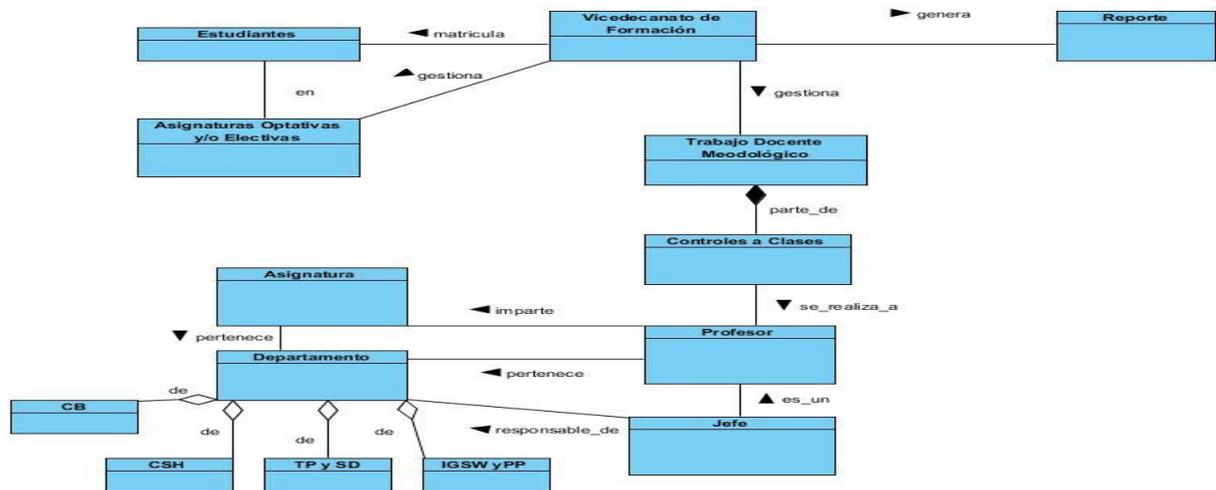


Imagen 1: Modelo conceptual, Sistema de Control Docente de la Facultad 4.

## 2.4 Historias de Usuario (HU)

Las HU son el artefacto eje de la metodología XP debido a que estas describen las funcionalidades que comprenderá el sistema. Las mismas son escritas por el cliente empleando un lenguaje transparente y conciso que deje bien claro las necesidades a suplir con el sistema. Permiten darle respuesta inmediatamente a las funcionalidades cambiantes en el transcurso del desarrollo y sirven de base para las pruebas de funcionamiento del sistema. También se emplean para estimar el tiempo y para la realización de los planes de entregas. Las HU están divididas por iteraciones:

Iteración 1: Se plantean las funcionalidades relacionadas con la gestión de los departamentos.

Iteración 2: Se identificaron las funcionalidades relacionadas con la gestión de información de los controles a clases y reportes del sistema.

Iteración 3: Se le da tratamiento a la gestión de asignaturas optativas o electivas, recomendación y sugerencia de las mismas.

Iteración 4 y última: Se le da tratamiento a la seguridad del sistema teniendo en cuenta los usuarios y permisos definidos.

A continuación se muestran algunas de las HU realizadas durante la fase de Exploración que abarcan las funcionalidades correspondientes en el sistema propuesto. Es importante apuntar que todas las HU se encuentran en los anexos.

Tabla 2: HU Listar Departamento

Historia de Usuario	
Número: 1	Nombre: Listar Departamento
Usuario: Administrador.	
Modificación de la Historia de Usuario: 1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgos de desarrollo: Medio	Puntos Reales: 1
Descripción: Inicialmente se listan los departamentos guardados en la base de datos. También se permite añadir nuevos departamentos, ver los detalles y editar dichos departamentos.	
Observación: Para realizar todas las acciones sobre todos los departamentos el usuario debe estar registrado como administrador. Si el usuario está registrado como jefe departamento solo puede realizar acciones sobre su departamento. En otro caso, no tiene privilegios para realizar alguna acción en el módulo.	

Tabla 3: HU Listar control a clase

Historia de Usuario	
Número: 2	Nombre: Gestionar control a clase.
Usuario: Administrador, Jefe de departamento.	
Modificación de la Historia de Usuario: 1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgos de desarrollo: Alta	Puntos Reales: 5
Descripción: Inicialmente se listan los controles a clase guardados en la base de datos. También se permite, añadir un control, ver detalles, editar y eliminar dicho control.	
Observación: Para realizar todas las acciones sobre todos los departamentos el usuario	

debe estar registrado como administrador. Si el usuario está registrado como jefe departamento solo puede realizar acciones sobre su departamento. En otro caso no se permite realizar estas acciones.

Tabla 4: HU Totales de controles por Facultad

Historia de Usuario	
Número: 3	Nombre: Totales de controles por facultad.
Usuario: Administrador.	
Modificación de la Historia de Usuario:	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgos de desarrollo: Media	Puntos Reales: 4
Descripción: El usuario registrado como administrador accede al módulo Reportes\Desglose de controles de la Facultad, donde se permite visualizar las estadísticas de los controles a clases planificados en cada departamento de la facultad y los realizados atendiendo a la cantidad de profesores que imparten docencia, para los cuales también se registrará el promedio por categoría docente. Los reportes se brindan en forma de tabla y en forma de gráfico.	
Observación: Para poder generar los reportes, los usuarios deben estar registrados en el sistema cumpliendo el rol de administrador.	

Tabla 5: HU Gestionar Asignaturas Optativas o Electivas

Historia de Usuario	
Número: 4	Nombre: Gestionar Asignaturas Optativas o Electivas
Usuario: Administrador.	
Modificación de la Historia de Usuario:	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgos de desarrollo: Alto	Puntos Reales: 5

Descripción: Se permite añadir, eliminar, editar, ver asignaturas optativas o electivas guardadas en el sistema.

Observación: Las acciones a realizar sobre las asignaturas se asignaran a cada rol, teniendo en cuenta los permisos que cada usuario tenga. El usuario con rol de estudiante solo puede matricularse en las asignaturas de su preferencia cuando le haya otorgado un rating a la misma.

Tabla 6: HU Sugerir Asignaturas Optativas o Electivas

Historia de Usuario	
Número: 5	Nombre: Sugerir Asignaturas Optativas o Electivas.
Usuario: Administrador, Secretaria.	
Modificación de la Historia de Usuario:	Iteración Asignada: 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgos de desarrollo: Alto	Puntos Reales: 5
Descripción: El usuario registrado como estudiante accede al sistema donde se permite visualizar las asignaturas sugeridas por el mismo. Desde su sesión el estudiante puede aceptar matricularse en las asignaturas sugeridas por el sistema y otorgar un rating a la misma o rechazar la sugerencia.	
Observación: El sistema solo sugiere asignaturas al usuario registrado como estudiante.	

Tabla7: HU Gestionar Usuario

Historia de Usuario	
Número: 6	Nombre: Autenticar Usuario
Usuario: Administrador, Jefe de departamento, Secretaria, Estudiante.	
Modificación de la Historia de Usuario:	Iteración Asignada: 4
Prioridad en Negocio: Medio	Puntos Estimados: 0.5
Riesgos de desarrollo: Medio	Puntos Reales: 1

Descripción: El individuo debe introducir su usuario y contraseña para autenticarse. Según sus privilegios accederá a la sesión correspondiente con su rol en la aplicación.

Observación:

## 2.5 Planeación

Kent Beck [38] en su libro define la planeación como una de las fases de la metodología XP, la cual se describe en la siguiente sección. Durante la planificación se priorizan las HU y se acuerda el alcance de la entrega. Esta fase consiste en una o varias reuniones grupales de planificación y el resultado es un plan de entregas. Se realiza una estimación del esfuerzo que costará implementar cada historia de usuario utilizando como medida el punto (una semana ideal de trabajo sin ningún tipo de interrupción) [38].

### 2.5.1 Estimación de Esfuerzos por (HU)

Con vista al correcto desarrollo del sistema propuesto, se realizó una estimación por HU identificadas y definidas anteriormente para alcanzar una medida, lo más cercana a la realidad, del progreso del proyecto. Como consecuencia de este proceso se obtuvo el siguiente resultado:

Tabla 8: Estimación de Esfuerzos por (HU)

No.	Historia de Usuario	Puntos Estimados
1	Listar departamento	1
2	Nuevo departamento	0.5
3	Ver detalles de departamento	0.5
4	Editar departamento	0.5
5	Gestionar control a clase	1
6	Desglose de controles por asignaturas	1
7	Desglose de controles por departamentos	1
8	Totales de controles en la Facultad	1
9	Profesores evaluados de mal	0.5
10	Relación estudiante-asignatura	0.5
11	Gestionar asignaturas optativas o electivas	2
12	Matricular estudiantes en las asignaturas optativas o electivas	1
13	Sugerir asignaturas optativas o electivas	2
14	Otorgar rating	0.5
15	Generar PDF	0.5
16	Autenticar usuarios	0.5
17	Añadir usuario	0.5

## 2.5.2 Plan e iteraciones para entregables.

El plan de entrega está compuesto por iteraciones de no más de 4 semanas. En la primera iteración se definen todos los elementos arquitectónicos y técnicos para dar estructura al sistema, la cual pueda ser utilizada durante el transcurso de desarrollo del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción [40]

El plan de entregas se basa en las Historias de Usuario. Una vez identificadas las mismas, se definieron niveles de prioridad lo que posibilitó dividir el desarrollo en 4 iteraciones de acuerdo a la prioridad de cada HU y la relación en cuanto a funcionalidad entre ellas.

Iteración 1: En esta iteración se realizan las HU relacionadas con los departamentos.

Iteración 2: En esta iteración se realizan las HU relacionadas con los controles a clases y la generación de reportes.

Iteración 3: En esta iteración se realizan las HU relacionadas con las asignaturas optativas o electivas.

Iteración 4: En la cuarta iteración y última se le da tratamiento a la gestión de usuarios y permisos asociados a estos y se prepara el sistema para realizar pruebas.

Tabla 9: Plan de entregas

Historias de Usuario	Tiempo Estimado(días)	Iteración Asignada	Entrega Asignada
Listar departamento	1	1	1
Nuevo departamento	3	1	1
Ver detalles de departamento	2	1	1
Editar departamento	3	1	1
Gestionar control a clase	5	2	2
Desglose de resultados de los controles a clases por asignaturas	5	3	3
Desglose de controles a clases por departamentos	5	3	3
Totales de controles en la Facultad	4	3	3
Relación estudiante-asignatura	1	3	3
Reporte de profesores evaluados de mal	2	3	3
Generar PDF	2	3	3
Gestionar asignaturas optativas o electivas	5	4	4

Matricular estudiantes	7	4	4
Sugerir asignatura optativas o electivas	5	4	4
Otorgar rating	3	4	4
Autenticar usuario	1	4	4
Anadir usuario	1	4	4

## 2.6 Diseño de la Aplicación

### 2.6.1 Tarjetas CRC

Las tarjetas CRC (clases, responsabilidades y colaboración) son una representación de las entidades del sistema a las cuales se les asignan responsabilidades y colaboraciones. El formato físico de las tarjetas CRC facilita la interacción entre clientes y el equipo de desarrollo, en sesiones en las que se aplican técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir de especificación de requisitos o historias de usuarios. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones. Luego en un estado de diseño avanzado o ya en la implementación del sistema, las tarjetas CRC se convierten en clases con métodos, atributos, relaciones de herencia, composición o dependencia. [41]

Durante el diseño de la aplicación se elaboraron un total de 9 tarjetas CRC las que se exponen a continuación:

Tabla 10: Tarjeta CRC Estudiante

Tarjeta CRC	
Clase: Estudiante	
Responsabilidad	Colaboración
Contiene toda la información referente a un estudiante. Permite cambiar los atributos de uno o más estudiantes.	

Tabla 11: Tarjeta CRC Profesor

Tarjeta CRC
-------------

Clase: Profesor	
Responsabilidad	Colaboración
Contiene la información referente uno o más profesores. Permite cambiar los atributos de uno o más profesores.	

Tabla 12: Tarjeta CRC Grupo

Tarjeta CRC	
Clase: Grupo	
Responsabilidad	Colaboración
Contiene la información referente uno o más grupos. Permite cambiar los atributos de uno o más grupos.	

Tabla 13: Tarjeta CRC Asignatura

Tarjeta CRC	
Clase: Asignatura	
Responsabilidad	Colaboración
Contiene la información referente uno o más asignaturas. Permite cambiar los atributos de uno o más asignaturas.	

Tabla 14: Tarjeta CRC Departamento

Tarjeta CRC	
Clase: Departamento	
Responsabilidad	Colaboración
Contiene la información referente uno o más departamentos.	

Permite cambiar los atributos de uno o más departamentos.	
---	--

Tabla 15: Tarjeta CRC controlProfesorController

Tarjeta CRC	
Clase: controlProfesorController	
Responsabilidad	Colaboración
<p>Contiene la información referente uno o más controles a profesores.          Permite cambiar los atributos de uno o más controles a profesores.          Elimina uno o más controles realizados.          Lista los controles realizados.</p>	<p>Grupo          Profesor</p>

Tabla 16: Tarjeta CRC DepartamentoController

Tarjeta CRC	
Clase: DepartamentoController	
Responsabilidad	Colaboración
<p>Crea un nuevo departamento.          Modifica uno o varios atributos a un departamento en específico          Elimina un departamento.          Lista los departamentos existentes.          Crea nuevas asignaturas.          Añade asignaturas al departamento.          Añade profesores a departamentos.          Modifica uno o varios atributos de uno o varios profesores y/o asignaturas en cada departamento.          Quita uno o varios profesores y/o asignaturas de los departamentos.</p>	<p>Departamento          Asignaturas          Profesor</p>

Tabla 17: Tarjeta CRC Web\_Service\_Driver

Tarjeta CRC	
Clase: Web_Service_Driver	
Responsabilidad	Colaboración
Permite buscar profesores en la base de datos de la UCI y los envía a la clase departamentoController.	departamentoController

Tabla 18: Tarjeta CRC Control\_Profesor

Tarjeta CRC	
Clase: Control_Profesor	
Responsabilidad	Colaboración
Crea un nuevo control de profesores. Modifica uno o varios atributos a uno o más controles de profesores. Lista los controles existentes. Elimina controles existentes.	Profesor Grupo Asignatura

## 2.7 Arquitectura

Según Isidro Ramos “la arquitectura de software es la representación de alto nivel de la estructura de un sistema o aplicación, que describe las partes que la integran, las interacciones entre ellas, los patrones que supervisan su composición y las restricciones a la hora de aplicar esos patrones.” [42] Para la realización del sistema se utiliza el Symfony2 v2.5, tal como se mencionó en secciones anteriores. Este framework está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles. [43]

### 2.7.1 Patrón MVC

El patrón Modelo-Vista-Controlador separa los datos y la lógica del negocio de una aplicación de la interfaz de usuario y la gestión de eventos y comunicaciones. Este patrón de diseño se basa en la reutilización de código y la separación de

conceptos, características que buscan facilitar la tarea de aplicaciones y su superior mantenimiento. [44]

- ✓ El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ✓ El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.[43]

A continuación se muestra a través de un ejemplo la forma en que se aplican los principios de la arquitectura MVC en el SCD-F4:

Cuando el usuario accede a algún módulo para solicitar alguna información a través de una ruta, el sistema determina cual Controlador es el que está asociado a la ruta especificada. En el SCD-F4, los Controladores se encuentran en las carpetas Controller pertenecientes a cada bundle.

Symfony2 ejecuta el Controlador correspondiente a la ruta. El Controlador solicita al Modelo los datos de la clase. Una vez que el Modelo devuelve dichos datos, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos obtenidos por el Modelo. El Controlador entrega al servidor los datos la página creada por la Vista, la cual se le muestra al usuario.

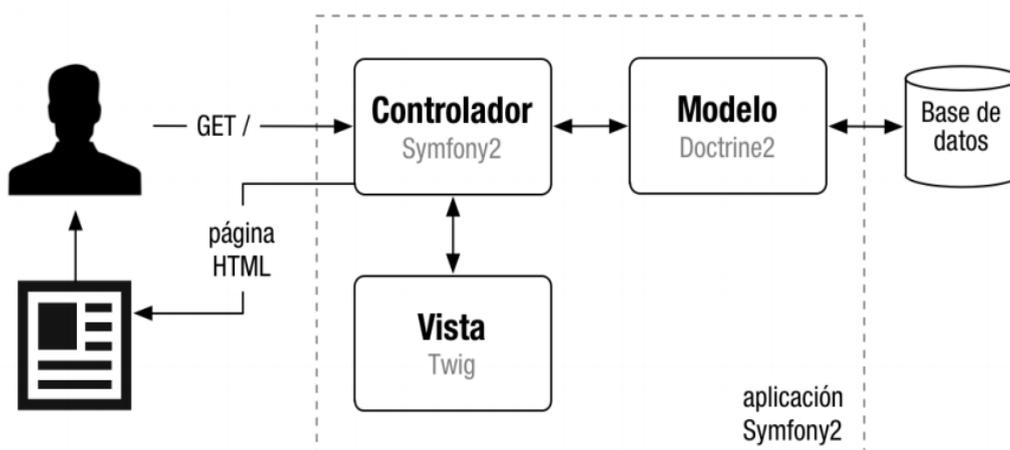


Imagen 2: Modelo-Vista-Controlador

## 2.7.2 Patrones de Diseño

Un patrón es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Estos patrones identifican Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. [45]. Para el diseño del sistema se emplea el GRASP acrónimo de General Responsibility Assignment Software Patterns, patrones generales de software para asignar responsabilidades. [46]

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. [46]

Dentro del diseño de la aplicación se detectó el uso de los siguientes patrones:

**Alta Cohesión:** En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. En el SCD/F4 se ejemplifica este patrón cuando se delegan responsabilidades a la clase `departamentoController.php` para gestionar toda la información manejada para los departamentos, se le delegan responsabilidades a la clase `controlProfesorController.php` para gestionar la información relacionadas con los profesores y sus categorías.

**Bajo Acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases y con que recurre a ellas. Una clase con bajo o débil acoplamiento no depende de muchas otras. [46] Para la construcción del sistema se dividió el mismo en dos bundles (`OptativasBundle` y `ControlProfesorBundle`) distribuidos de forma tal que cada uno englobe entidades y funcionalidades comunes e independientes que no dependa casi totalmente uno del otro para su funcionamiento, lo cual provee un diseño con bajo acoplamiento y alta cohesión.

**Controlador:** Este patrón se encarga que una clase actúe como intermediaria para el manejo de eventos. Sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Por ejemplo, cuando se solicita un reporte del total de controles de la Facultad, el controlador `controlProfesor` es el intermediario para obtener toda la información solicitada.

Por otra parte, se aplicaron algunos de los patrones de diseño llamado Gang of Four (Gof) entre los que se encuentran:

Builder o de “Creación”: Este patrón indica a qué clase se le asigna la responsabilidad de la creación de instancias, puesto que esta posee la información necesaria para la creación de este objeto. Por ejemplo, la clase controlProfesorController.php se encuentran todas las acciones que se pueden hacer a la hora de gestionar un control a clases. Entre dichas acciones se crean nuevos controles en representación lo que evidencia que la clase controlProfesorController.php es “creador” de dichas entidades.

Singleton: Es un patrón creador que permite obtener, siempre que se solicite, un objeto de una clase determinada, obtener la misma instancia. Es utilizado por el framework propuesto para la solución. Es muy utilizado por Symfony en sus clases sfContext, sfRequest, sfResponse, etc.

## 2.8 Modelo de datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia “. (Database System. [47] A continuación se muestra una propuesta del modelo de datos del Sistema de Control Docente de la Facultad 4.

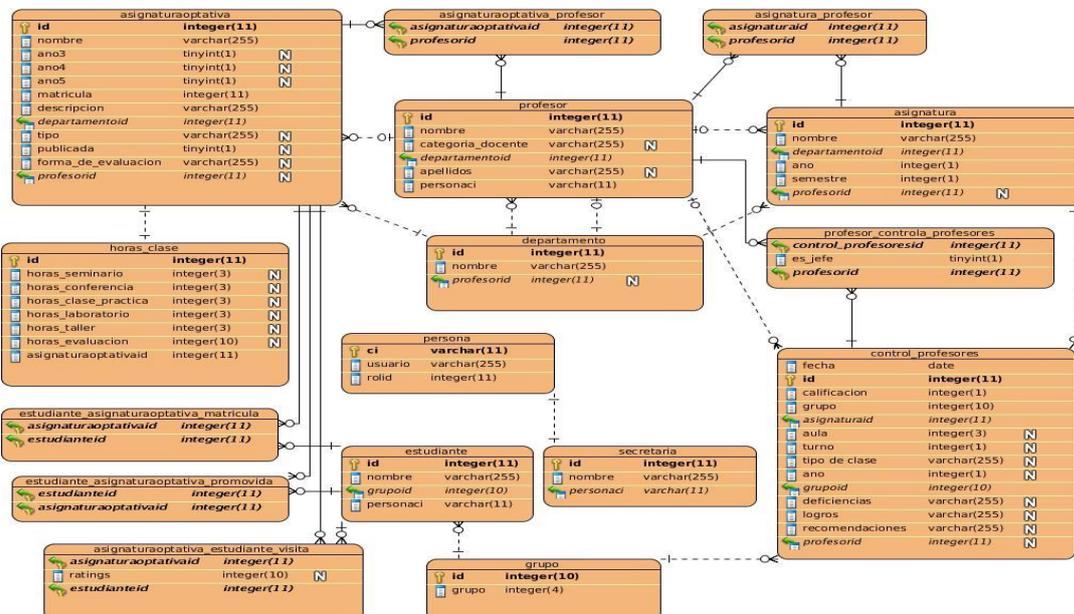


Imagen 3: Diagrama de Entidad-Relación

## **Conclusiones parciales**

Se analizaron las funcionalidades a incluir en la propuesta. Se elaboraron y describieron los principales artefactos que propone la metodología XP para la planificación, como las HU y el Plan de entregas. Se definió como patrón de diseño arquitectónico el MVC, así como algunos patrones GRASP y GOF entre los que se destacan Controlador, Alta Cohesión, Bajo Acoplamiento, Creador y Singleton. Esto permitió obtener una mejor organización y calidad en la solución, evidenciándose a través de las tarjetas CRC.

## Capítulo 3.

### Implementación y prueba.

#### Introducción

En este capítulo se estarán mostrando las tareas de implementación generadas por cada historia de usuario, obteniendo una versión del Sistema de Gestión Docente de la Facultad 4 en cada iteración, así como las pruebas realizadas para el mismo. La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

#### 3.1 Implementación

En la fase de Planeación se definieron los elementos funcionales necesarios para realizar el desarrollo de la aplicación, lo que implica que ya se está en disponibilidad para continuar a la fase de implementación. Para esta fase se dividen las funcionalidades operativas que conforman las HU en tareas más pequeñas, las que toman el nombre de tareas de ingeniería. Estas tienen el objetivo de realizar un análisis más detallado de cada funcionalidad y lograr una estimación más real de su tiempo de desarrollo. Lo principal de las tareas de ingeniería es que son importantes para guiar la implementación por lo que no necesariamente deben ser entendibles para el cliente por ser descritas en un lenguaje técnico.

Las HU pueden ser desglosadas en uno a más tareas de ingeniería, en dependencia de la complejidad que posea la funcionalidad a desarrollar. En la etapa anterior se desglosaron algunas HU, las que coinciden con algunas tareas, para bajar la complejidad de cada funcionalidad y ganar en tiempo de desarrollo. Atendiendo a la planificación realizada en el capítulo anterior, se realizaron cuatro iteraciones para las que se obtuvieron las siguientes tareas de ingeniería.

##### Iteración 1

Tabla 19: Tarea 1 de la HU Listar departamento.

Número tarea: 1	Historia de Usuario: 1
Nombre tarea: Añadir profesor a departamento.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha inicio: 28/02/2015	Fecha fin: 01/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: Se seleccionan profesores ya registrados en la base de datos del sistema o se buscan en la base de datos UCI, se especifica si es jefe de departamento y luego se asigna a cada departamento.	

Tabla 20: Tarea 2 de la HU Listar departamento.

Número tarea: 2	Historia de Usuario: 1
Nombre tarea: Quitar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha inicio: 02/03/2015	Fecha fin: 03/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona los profesores que no pertenecen al departamento y los elimina. La eliminación solo se realiza en el departamento no en la base de datos como tal.	

Tabla 21: Tarea 3 de la HU Ver detalles de departamento.

Número tarea: 1	Historia de Usuario: 3
Nombre tarea: Editar datos del profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.10
Fecha inicio: 04/03/2015	Fecha fin: 05/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona el profesor al que desea actualizar los datos y los modifica.	

Tabla 22: Tarea 4 de la HU Ver detalles de departamento.

Número tarea: 2	Historia de Usuario: 3
Nombre tarea: Editar asignaturas.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.15
Fecha inicio: 06/03/2015	Fecha fin: 07/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona la asignatura que desea modificar y edita la información.	

Tabla 23: Tarea 5 de la HU Ver detalles de departamento.

Número tarea: 4	Historia de Usuario: 3
Nombre tarea: Quitar asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.10
Fecha inicio: 08/03/2015	Fecha fin: 09/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona la asignatura que desea quitar y la retira.	

## Iteración 2

Tabla 24: Tarea 6 de la HU Listar control a clase.

Número tarea: 1	Historia de Usuario: 6
Nombre tarea: Nuevo control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha inicio: 10/03/2015	Fecha fin: 12/03/2015
Programador responsable: Dariel Corchado López del Castillo	
Descripción: El usuario indica crear un nuevo control a clase y adiciona sus datos.	

Tabla 25: Tarea 7 de la HU Listar control a clase.

Número tarea: 2	Historia de Usuario: 6
Nombre tarea: Editar control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25

Fecha inicio: 13/03/2015	Fecha fin: 14/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona el control a clase que desea modificar y añade la nueva descripción.	

Tabla 26: Tarea 8 de la HU Listar control a clase.

Número tarea: 3	Historia de Usuario: 6
Nombre tarea: Ver detalles de control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha inicio: 15/03/2015	Fecha fin: 16/03/2015
Programador responsable: Dariel corchado López del Castillo	
Descripción: Se muestra una lista con todos los controles a clase realizados hasta el momento.	

Tabla 27: Tarea 9 de la HU Listar control a clase.

Número tarea: 4	Historia de Usuario: 6
Nombre tarea: Eliminar control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha inicio: 17/03/2015	Fecha fin: 18/03/2015
Programador responsable: Maikel Ruiz Pol	
Descripción: El usuario selecciona los controles a clases y los elimina si desea.	

### Iteración 3

Tabla 28: Tarea 10 de la HU Gestionar asignatura optativa o electiva.

Número tarea: 11	Historia de Usuario: 11
Nombre tarea: Nueva asignatura.	
Tipo de tarea: Programación	Puntos estimados: 0.15
Fecha inicio: 19/03/2015	Fecha fin: 21/03/2015

Programador responsable: Dariel Corchado López del Castillo
Descripción: El usuario selecciona la opción nueva asignatura e introduce los datos de la nueva asignatura optativa o electiva y la crea. El usuario autorizado para realizar esta acción es el administrador o jefe de departamento.

Tabla 29: Tarea 11 de la HU Gestionar asignaturas optativas o electivas

Número tarea: 2	Historia de Usuario: 11
Nombre tarea: Detalles de asignatura.	
Tipo de tarea: Programación	Puntos estimados: 0.25
Fecha inicio: 21/03/2015	Fecha fin: 22/03/2015
Programador responsable: Dariel Corchado López del Castillo	
Descripción: el usuario selecciona la asignatura optativa o electiva de su preferencia y observa sus detalles.	

Tabla 30: Tarea 12 de la HU Gestionar asignaturas optativas o electivas.

Número tarea: 3	Historia de Usuario: 11
Nombre tarea: Editar asignatura.	
Tipo de tarea: Programación	Puntos estimados: 0.15
Fecha inicio: 23/03/2015	Fecha fin: 24/03/2015
Programador responsable: Dariel Corchado López del Castillo	
Descripción: El usuario selecciona la asignatura optativa o electiva que desea modificar y edita la información. El usuario autorizado para realizar esta acción es el administrador o jefe de departamento.	

### 3.2 Pruebas del sistema

Para la metodología XP el uso de pruebas es fundamental puesto que es la manera de comprobar que las funcionalidades que se van implementando funcionan correctamente y cumplen con lo requerido por el cliente. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo

transcurrido entre la aparición de un error y su detección. La metodología de desarrollo empleada para el trabajo de diploma divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. Seguidamente se brinda una ejemplificación de las pruebas más importantes realizadas al sistema.

### **3.2.1 Pruebas unitarias**

Las pruebas unitarias son una forma de comprobar el correcto funcionamiento de los módulos de código, comprobando su funcionamiento por separado. [48] Generalmente son realizadas por el mismo programador, debido a que al conocer con mayor detalle el código, se les simplifica la tarea de elaborar conjuntos de datos de prueba para testarlo. Symfony2 es un framework que permite, de manera sencilla, realizar pruebas unitarias pues combina las facilidades que propone con la potencia de la librería PHPUnit que prácticamente se ha convertido en un estándar en el mundo PHP. [49]

#### **¿Por qué realizar pruebas unitarias?**

- ✓ Asegura calidad del código entregado. Es la mejor forma de detectar errores tempranamente en el desarrollo. No obstante, esto no asegura detectar todos los errores, por tanto las pruebas aceptación siguen siendo necesarias.
- ✓ Ayuda a definir los requerimientos y responsabilidades de cada método en cada clase probada.
- ✓ Permite hacer refactoring tempranamente en el código. No es necesario todo un ciclo de integración para hacer refactoring en la aplicación, basta con ver cómo se comporta un caso de prueba para hacer refactoring unitario sobre la clase que estamos probando en cuestión.
- ✓ Permite incluso hacer pruebas de estrés tempranamente en el código. Por ejemplo un método que realice una consulta SQL que exceda los tiempos de aceptación es posible optimizarla antes de integrar con la aplicación.

#### **Validando entidades**

Asegurar que la información creada por la aplicación sea correcta es fundamental para su buen funcionamiento. Para ello no basta con añadir reglas de validación a las entidades, sino que es

imprescindible comprobar que todas se están cumpliendo satisfactoriamente. Las pruebas unitarias realizadas al sistema están enfocadas en el correcto funcionamiento de las entidades, es por ello que se presenta a continuación un ejemplo de algunas pruebas realizadas.

```
<?php
2
3 use Symfony\Component\Validator\Validation;
4 use scd\ControlProfesorBundle\Entity\controlProfesor;
5
6 class controlProfesorTest extends \PHPUnit_Framework_TestCase {
7
8     private $validator;
9
10    public function setUp() {
11        $this->validator = Validation::createValidatorBuilder()
12            ->enableAnnotationMapping()
13            ->getValidator();
14    }
15
16    public function testValidarFecha() {
17        $control = new controlProfesor();
18        $control->setAño(1);
19        $control->setCalificacion(2);
20        $control->setDeficiencias("Hay muchas");
21        $control->setLogros("hay pocos");
22        $control->setRecomendaciones("anja");
23        $control->setTurno(1);
24        $control->setTipoDeClase('Conferencia');
25        $control->setFecha(null);
26        $listaErrores = $this->validator->validate($control);
27        $this->assertGreaterThan(0, $listaErrores->count(), 'La fecha no puede estar vacía');
28        $error = $listaErrores[0];
29        $this->assertEquals('La fecha no puede estar vacía', $error->getMessage());
30        $this->assertEquals('fecha', $error->getPropertyPath());
31    }
32}
```

Imagen 4: Código de la prueba de validación aplicado a la entidad controlProfesor.

```
Terminal - ...hy-linux: /var/www/html/scd-facultad4
corchy@corchy-linux:~$ sudo su
[sudo] password for corchy:
root@corchy-linux:/home/corchy# phpunit -c app src/scd/ControlProfesorBundle/Tests/Entity/controlProfesorTest.php
Could not read "app".
root@corchy-linux:/home/corchy# cd /var/www/html/scd-facultad4/
root@corchy-linux:/var/www/html/scd-facultad4# phpunit -c app src/scd/ControlProfesorBundle/Tests/Entity/controlProfesorTest.php
PHPUnit 3.7.28 by Sebastian Bergmann.

Configuration read from /var/www/html/scd-facultad4/app/phpunit.xml.dist

.....

Time: 265 ms, Memory: 3.75Mb

OK (5 tests, 15 assertions)
root@corchy-linux:/var/www/html/scd-facultad4#
```

Imagen 5: Resultado de la prueba de validación aplicado a la entidad controlProfesor.

Las pruebas se realizaron por iteraciones obteniéndose algunas no conformidades las cuales fueron solucionadas en su totalidad. El siguiente gráfico ejemplifica como fue este proceso.

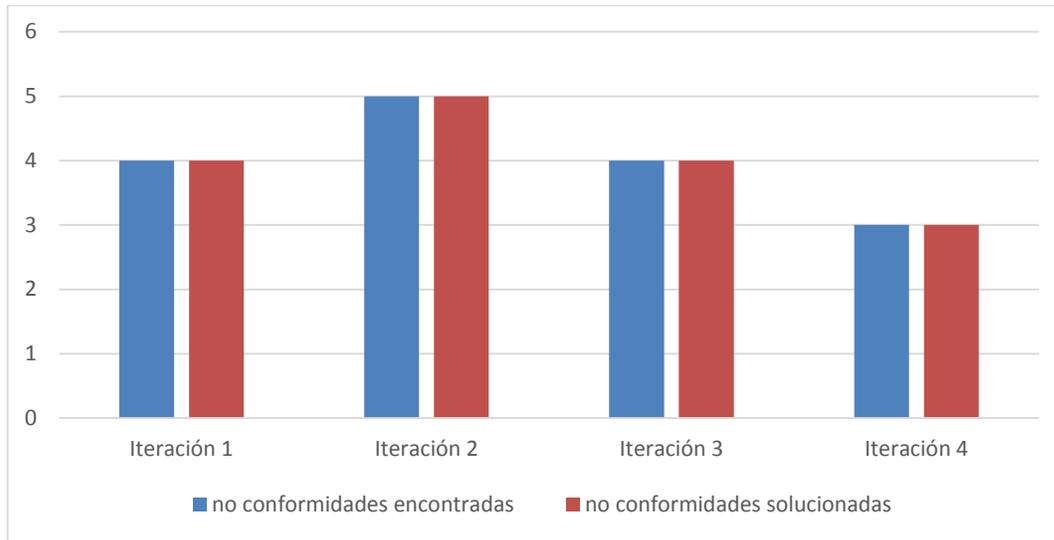


Imagen 6: Estadísticas de las pruebas unitarias hechas por iteración

### 3.2.1 Pruebas de aceptación

Las pruebas de aceptación (PA) se hacen con el propósito de demostrar al cliente el cumplimiento de un requisito del software. Se caracterizan por:

- ✓ Describir un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del cliente.
- ✓ Estar asociada a un requisito funcional o requisito no funcional.
- ✓ Cubrir desde escenarios típicos/frecuentes hasta los más excepcionales.
- ✓ Tener infinitas instancias (ejecuciones con valores concretos).
- ✓ Ser el tester quien realice el trabajo de diseñar las instancias y su aplicación. [50]

Atendiendo a lo anteriormente especificado se muestran los resultados obtenidos de las PA realizadas.

Tabla 31: Prueba de Aceptación # 1

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Listar Departamento	
<b>Descripción:</b> Prueba para la funcionalidad de obtener una lista de departamentos y sus características.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción gestionar departamento y accede al módulo. También tiene la opción de crear, editar o ver detalles de los departamentos, si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra una lista con los departamentos existentes y las opciones de ver detalles, editar departamento y añadir departamento en caso de ser administrador.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 32: Prueba de Aceptación # 2

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F2	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Añadir profesor a departamento.	
<b>Descripción:</b> Prueba para la funcionalidad de adicionar profesores a los departamentos.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos en la base de datos y que el usuario este registrado como administrador o jefe de departamento.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción añadir profesor al departamento y lo añade desde la base de datos UCI o desde la base de datos del sistema.	
<b>Resultado Esperado:</b> El sistema muestra los detalles del departamento incluyendo al nuevo profesor.	

**Evaluación de la Prueba:** Satisfactoria.

Tabla 33: Prueba de Aceptación # 3

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F3	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Nuevo departamento.	
<b>Descripción:</b> Prueba para la funcionalidad de adicionar departamentos.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción nuevo departamento y lo añade con todas sus características.	
<b>Resultado Esperado:</b> El sistema muestra los detalles del nuevo departamento así como las opciones, editar y eliminar.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 34: Prueba de Aceptación # 4

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F4	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Ver detalles de departamento.	
<b>Descripción:</b> Prueba para la funcionalidad ver los detalles de cada departamento.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos guardados en la base de datos y que el usuario esté registrado como administrador o jefe de departamento.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un departamento y accede a la opción ver detalles.	
<b>Resultado Esperado:</b> El sistema muestra la información del departamento seleccionado.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 35: Prueba de Aceptación # 5

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F5	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Editar departamento.	
<b>Descripción:</b> Prueba para la funcionalidad de modificar la información de los departamentos.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos guardados en la base de datos y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un departamento y accede a la opción editar para modificar la información que necesite en el departamento o eliminar el mismo.	
<b>Resultado Esperado:</b> El sistema muestra la información modificada del departamento.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 36: Prueba de Aceptación # 6

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F6	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Listar control a clase.	
<b>Descripción:</b> Prueba para la funcionalidad de listar los controles a clase existentes hasta el momento.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos, profesores, asignaturas y controles a clase guardados en la base de datos y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción gestionar controles y accede al módulo. También tiene la opción de crear, editar o ver detalles de los controles, si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra una lista con todos los controles existentes en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 37: Prueba de Aceptación # 7

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F2	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Nuevo control a clase.	
<b>Descripción:</b> Prueba para la funcionalidad de crear un nuevo control a clase.	
<b>Condiciones de Ejecución:</b> Verificar que existan profesores, asignaturas y controles a clase guardados en la base de datos y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción nuevo control y añade un control a clase con sus características.	
<b>Resultado Esperado:</b> El sistema muestra la información del nuevo control a clase.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 38: Prueba de Aceptación # 8

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F2	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Editar control a clase.	
<b>Descripción:</b> Prueba para la funcionalidad de editar controles a clase existentes en el sistema.	
<b>Condiciones de Ejecución:</b> Verificar que existan controles a clase almacenados y que el usuario este registrado como administrador o jefe de departamento.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un control a clase, elige la opción editar y modifica la información que necesita.	
<b>Resultado Esperado:</b> El sistema muestra la información del control editado recientemente.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 39: Prueba de Aceptación # 9

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F3	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Ver detalles de control a clase.	
<b>Descripción:</b> Prueba para la funcionalidad de ver detalles de los controles.	
<b>Condiciones de Ejecución:</b> Verificar que existan controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un control a clase, elige la opción detalles.	
<b>Resultado Esperado:</b> El sistema muestra la información del control seleccionado.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 40: Prueba de Aceptación # 10

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F4	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Eliminar control a clase.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar controles a clase.	
<b>Condiciones de Ejecución:</b> Verificar que existan controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un control a clase, elige la opción detalles o modificar control y luego elige la opción eliminar.	
<b>Resultado Esperado:</b> El sistema elimina el control seleccionado y luego muestra la información de los controles que aún persisten en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 41: Prueba de Aceptación # 11

Caso de Prueba de Aceptación
------------------------------

<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Desglose de controles por asignaturas.	
<b>Descripción:</b> Prueba para la funcionalidad de obtener reportes por cada asignatura ordinaria presente en el plan de estudio.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos, controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción desglose de controles a clase por asignatura y elige la opción generar PDF si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra información graficada del desglose generado de los controles a clase por asignaturas en cada departamento además de una tabla con la misma información.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 42: Prueba de Aceptación # 12

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Desglose de controles por departamento.	
<b>Descripción:</b> Prueba para la funcionalidad de obtener reportes por cada departamento.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos, controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción desglose de controles a clase por departamento y elige la opción generar PDF si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra información graficada del desglose generado de los controles a clase por departamento además de una tabla con la misma información.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 43: Prueba de Aceptación # 13

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Totales de controles por facultad.	
<b>Descripción:</b> Prueba para la funcionalidad de obtener reportes totales por facultad, son reportes generales.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos, controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción totales de controles por facultad y elige la opción generar PDF si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra información graficada del total de controles de los controles a clase general de la facultad en un gráfico de barra además de una tabla con la misma información.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 44: Prueba de Aceptación # 14

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 9
<b>Nombre:</b> Profesores evaluados de mal.	
<b>Descripción:</b> Prueba para la funcionalidad de obtener reportes de los profesores evaluados de mal en la facultad, de acuerdo a los controles realizados.	
<b>Condiciones de Ejecución:</b> Verificar que existan departamentos, controles a clase almacenados y que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción profesores evaluados de mal y elige la opción generar PDF si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra una tabla con la información de los profesores evaluados de mal en la facultad.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 45: Prueba de Aceptación # 15

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Relación Estudiante-Asignatura.	
<b>Descripción:</b> Prueba para la funcionalidad de generar reportes de la relación de los estudiantes matriculados en las asignaturas ofertadas por la facultad.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción de generar un reporte de la Relación Estudiante-Asignatura.	
<b>Resultado Esperado:</b> El sistema muestra una tabla con la relación de los estudiantes matriculados en las asignaturas optativas o electivas ofertadas por la facultad, cuantas han cursado y la relación de cuantas les faltan por cursar.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 46: Prueba de Aceptación # 16

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 11
<b>Nombre:</b> Gestionar Asignaturas Optativas o Electivas.	
<b>Descripción:</b> Prueba para la funcionalidad de gestionar las asignaturas optativas o electivas ofertadas por la facultad.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción gestionar asignaturas optativas o electivas y accede al módulo. También tiene la opción de crear, editar o ver detalles de las asignaturas, si lo desea.	
<b>Resultado Esperado:</b> El sistema muestra una tabla con la información de las asignaturas optativas o electivas creadas listadas.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 47: Prueba de Aceptación # 17

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 12
<b>Nombre:</b> Matricular Estudiantes en las Asignaturas Optativas o Electivas.	
<b>Descripción:</b> Prueba para la funcionalidad de matricular estudiantes en las asignaturas optativas o electivas ofertadas por la facultad.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador, secretaria o estudiante.	
<b>Entradas/Pasos de Ejecución:</b> El usuario de tipo administrador o secretaria selecciona la opción estudiante, luego la opción matricular estudiante, introduce el su usuario y su grupo y acepta la matrícula. Por otra parte el usuario de tipo estudiante accede a la aplicación donde se le muestra el listado de las asignaturas disponible y posteriormente se matricula en la que desea.	
<b>Resultado Esperado:</b> El sistema muestra una notificación de que fue matriculado exitosamente. Si el usuario registrado es administrador o secretaria se muestra la información de los estudiantes matriculados hasta el momento incluyendo el nuevo matriculado. Si es de tipo estudiante se elimina la asignatura de las disponibles.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 48: Prueba de Aceptación # 18

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 13
<b>Nombre:</b> Sugerir Asignaturas Optativas o Electivas.	
<b>Descripción:</b> Prueba para la funcionalidad de sugerir asignaturas por parte de la secretaria a los estudiantes.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador o secretaria.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona la opción sugerir asignaturas y accede al módulo donde introduce su usuario.	
<b>Resultado Esperado:</b> El sistema muestra la sugerencia de las asignaturas optativas o electivas para el estudiante.	

**Evaluación de la Prueba:** Satisfactoria.

Tabla 49: Prueba de Aceptación # 19

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 14
<b>Nombre:</b> Otorgar rating.	
<b>Descripción:</b> Prueba para la funcionalidad de otorgarle un rating a las asignaturas para mostrar las preferencias de los estudiantes.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador o estudiante, que existan asignaturas guardadas en el sistema.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona un rating aumentando los puntos con el sombreado de las estrellas de cada asignatura.	
<b>Resultado Esperado:</b> El sistema muestra el rating proporcionado por el estudiante y guarda dicho rating para generar las recomendaciones.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 50: Prueba de Aceptación # 20

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 15
<b>Nombre:</b> Generar PDF.	
<b>Descripción:</b> Prueba para la funcionalidad de generar los reportes realizados por el sistema en formato PDF.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El usuario selecciona cualquiera de las opciones de generar reportes de los controles a clases y elige la opción generar PDF.	
<b>Resultado Esperado:</b> El sistema obtiene la información generada y la exporta a formato PDF.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 51: Prueba de Aceptación # 21

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 16
<b>Nombre:</b> Autenticar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad de gestionar los usuarios del sistema.	
<b>Condiciones de Ejecución:</b>	
<b>Entradas/Pasos de Ejecución:</b> El cliente accede a la interfaz de inicio de sección del sistema, introduce su usuario y contraseña en el sistema y selecciona la opción autenticar.	
<b>Resultado Esperado:</b> El sistema obtiene la información y muestra la interfaz de usuario.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 52: Prueba de Aceptación # 22

Caso de Prueba de Aceptación	
<b>Código:</b> HU_F1	<b>Historia de Usuario:</b> 17
<b>Nombre:</b> Añadir Usuario local.	
<b>Descripción:</b> Prueba para la funcionalidad de añadir usuario local al sistema.	
<b>Condiciones de Ejecución:</b> Verificar que el usuario este registrado como administrador.	
<b>Entradas/Pasos de Ejecución:</b> El administrador selecciona la opción registrar usuario y añade un nuevo usuario al sistema completando todos los campos necesarios.	
<b>Resultado Esperado:</b> El sistema obtiene muestra una notificación.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

### 3.2.2 Prueba de recomendación

Con el objetivo de garantizar el correcto funcionamiento de las sugerencias de asignaturas optativas o electivas que realiza el sistema a los estudiantes, se conciben las pruebas a los algoritmos de recomendación.

En el acápite 1 se especificó que se llevaría a cabo la recomendación basada en ítems utilizando como herramienta principal el Mahout. Para la implementación de este tipo de recomendación esta herramienta propone cuatro funciones de similitud fundamentales las cuales son Pearson, Euclidean, Log-likelihood y Tanimoto. Las pruebas que se realizan en este epígrafe están enfocadas en cuan certeras son dichas funciones de similitud a la hora de recomendar un ítem.

Mahout también permite evaluar la precisión de la recomendación. Para esto es necesario contar con dos conjuntos de datos: el conjunto con la información real (CR) y un conjunto pequeño de prueba (CP) que se obtiene del CR. Utilizando el CP Mahout recomienda un ítem al usuario con un valor  $x$  de preferencia. Luego esto se compara con los datos reales obteniendo la diferencia entre los valores de los datos reales y los recomendados. Seguidamente se calcula el promedio de estas diferencias, el cual consiste en el valor de precisión de la función de similitud empleada en la recomendación. Mientras más bajo sea el valor obtenido, más efectiva es la función y por consiguiente más efectivo es el método de recomendación.

Las pruebas ejecutadas consistieron en recomendarle asignaturas a un grupo de 90 estudiantes, los cuales previamente realizaron una encuesta donde asignaron valores de preferencia (rating) a un conjunto de 15 asignaturas optativas. Las recomendaciones fueron realizadas empleando cada una de las funciones de similitud mencionadas anteriormente. El promedio de las diferencias, en cada caso, se fue almacenando lo cual se muestra en la siguiente tabla:

Funciones de similitud.	Puntuación de recomendación.	Ítems recomendados.
Euclidean	1,241	3,9,2
Pearson	1.714	3,9,2

Log-likelihood	1,293	3,9,2
Tanimoto	1.246	3,9,2

Tabla 53: Muestra de la prueba realizada a los algoritmos de recomendación.

Como se puede apreciar la función Euclidean es la que arroja mejores resultados y por tanto se escoge para implementar las recomendaciones del sistema.

Luego de implementar el recomendador se procedió a verificar si lo que se recomendaba era bastante semejante a lo que habían descrito previamente los estudiantes en la encuesta y se comprobó que en un 97 % de los casos las recomendaciones coincidían con lo planteado por los estudiantes.

### 3.3 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe quedar como si un único programador hubiera escrito cada línea de una sola vez. [51] Es muy buena práctica, establecer estándares de codificación para garantizar que los programadores realicen un trabajo coherente. Emplear diferentes técnicas de codificación sólidas contribuye a mejorar la calidad del software y obtener mejores índices de rendimiento.

XP propone una serie de convenciones o estándares de códigos enfocados a la estructura, apariencia física de la aplicación, comprensión y mantenimiento del código. Por otra parte Symfony sigue los estándares definidos en los documentos PSR-0, PSR-1 y PSR-2 lo que hace inevitable el uso de los mismos en la implementación de la solución propuesta. [52] A continuación se muestran algunas de las reglas tenidas en cuenta para la implementación de la aplicación. Es necesario aclarar que en los anexos se amplían detalles visuales obtenidos de dicha implementación.

➤ **Identación:**

- ✓ Se debe emplear cuatro espacios como unidad de indentación, no tabs.

➤ **Longitud de líneas**

- ✓ No debe emplearse límites exagerados de longitud de línea, evitando tener más de 80 caracteres por cada una.

## ➤ **Comentarios**

Los comentarios a usar son los siguientes:

- ✓ para documentación `/**...*/`
- ✓ para programación `//...`

## ➤ **Estructura**

- ✓ Se emplea una declaración por línea lo que facilita los comentarios.
- ✓ Se añade un solo espacio alrededor de los operadores (`==`, `&&`,...).
- ✓ Se añade una coma después de cada elemento del arreglo en un arreglo multilínea, incluso después del último.
- ✓ Define una clase por archivo

## ➤ **Convenciones de nomenclatura**

- ✓ Utiliza mayúsculas intercaladas —sin guiones bajos— en nombres de variable, función, método o argumentos.
- ✓ Utiliza espacios de nombres para todas las clases.
- ✓ Sufija las excepciones con `Exception`.

## ➤ **Documentación**

- ✓ Añade bloques PHPDoc a todas las clases, métodos y funciones;
- ✓ Omite la etiqueta `@return` si el método no devuelve nada;
- ✓ Las anotaciones `@package` y `@subpackage` no se utilizan.

## **Conclusiones**

En este capítulo se detallaron elementos técnicos de la implementación y prueba del sistema de forma tal que se evidenciaran los resultados obtenidos y asegurara la calidad del producto a través de las diferentes pruebas realizadas. En un primer momento se desglosaron y resolvieron las historias de usuarios en pequeñas funcionalidades que constituyeron tareas de ingeniería mejorando así la estimación realizada en el acápite anterior. Por otra parte se verificó la calidad del sistema con la aplicación de pruebas unitarias realizadas al código, las pruebas de recomendación que permitieron comprobar la calidad de sugerencia de las asignaturas recomendadas. Las pruebas de aceptación propuestas por la metodología seleccionada, permitieron verificar si los resultados obtenidos fueron los esperados, documentando todos los resultados arrojados, quedando el producto listo para usarse en el Vicedecanato de Formación de la Facultad 4. En un último momento se analizaron elementos importantes para la codificación, lo que le dio mejor estructura para el entendimiento del código fuente de la aplicación.

## Conclusiones Generales

A lo largo de la investigación se describen los pasos para darle cumplimiento al objetivo de desarrollar una herramienta que permita gestionar la matrícula de cursos optativos o electivos y su recomendación automática, los controles a clase en la Facultad 4.

Una vez finalizado el trabajo se plantean las siguientes conclusiones:

- ✓ Los métodos científicos y las técnicas de recopilación de información empleadas permitieron desarrollar las bases teóricas que sustenta el estudio realizado, logrando conocer que no existe en Cuba, ni en la Universidad de Ciencias Informáticas un sistema que combine la matrícula de cursos en línea con la recomendación de los mismos, y los que fueron estudiados a nivel internacional no poseen todas las funcionalidades necesarias, sin embargo aportan elementos significativos para la solución propuesta.
- ✓ La selección de las herramientas, metodología de desarrollo, lenguajes de programación y técnicas más apropiadas para dar solución al objetivo general fueron resultado del análisis realizado.
- ✓ La metodología seleccionada permitió satisfacer la necesidad del cliente mediante las tareas de ingeniería de cada HU y la implementación, partiendo de las tarjetas CRC, cuyo desarrollo se llevó a cabo en el diseño de la aplicación.
- ✓ Las pruebas realizadas mostraron el grado de cumplimiento respecto a las especificaciones iniciales del sistema, garantizando la calidad del software y la satisfacción del cliente.
- ✓ El cumplimiento de las necesidades funcionales que demanda el área para la cual se propone el sistema creado muestra la calidad y la utilidad del mismo.

## **Recomendaciones**

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones.

- ✓ Continuar desarrollando las funcionalidades que brinda el sistema adaptándolo cada vez más a las comodidades de los usuarios que interactúan con el mismo.
- ✓ Brindar servicios web que puedan integrarse o ser de gran utilidad para el Sistema de Gestión Universitaria.
- ✓ Desarrollar un manual de usuario para uso de los clientes, con el objetivo de lograr un mayor entendimiento del sistema y su funcionamiento.
- ✓ Perfeccionar la adaptabilidad del sistema a diferentes resoluciones de pantalla.

## Referencias Bibliográficas

- [1] «Administración educativa - Monografias.com». [En línea]. Disponible en: <http://www.monografias.com/trabajos93/la-administracion-educativa/la-administracion-educativa.shtml>. [Accedido: 05-jun-2015].
- [2] Yadisbel Arencibia-Rivera, Yosvany Barr, ios-Hernández, Nexos Cabrera-Padrón, Osvaldo Domínguez-Junco, y Naidelys García-Delgado, «El Control Interno en la Educación Superior con enfoque en el Proceso Docente Educativo», vol. 15, p. 11, mar. 2013.
- [3] «Resolución no. 210-07 del mes». [En línea]. Disponible en: [https://www.google.com/cu/webhp?ie=utf-8&oe=utf-8&gws\\_rd=cr&ei=yP9hVbbDIKv7sAS5\\_4LwCw#q=Resoluci%C3%B3n+no.+210-07+del+mes](https://www.google.com/cu/webhp?ie=utf-8&oe=utf-8&gws_rd=cr&ei=yP9hVbbDIKv7sAS5_4LwCw#q=Resoluci%C3%B3n+no.+210-07+del+mes). [Accedido: 24-may-2015].
- [4] Universidad de las Ciencias Informáticas, «Sistema de Gestión Universitaria». [En línea]. Disponible en: <https://sgu.uci.cu/#>. [Accedido: 15-feb-2015].
- [5] British Standards Institution, *British Standards*. Universidad de Michigan: British Standards Institution, 1983.
- [6] «Sistemas de Gestión Normalizados | THINK&SELL». [En línea]. Disponible en: <http://thinkandsell.com/servicios/consultoria/software-y-sistemas/sistemas-de-gestion-normalizados/>. [Accedido: 26-abr-2015].
- [7] «Prisma: Sistema de gestión académica de la UPC | inLab FIB», 06-feb-2015. [En línea]. Disponible en: <http://inlab.fib.upc.edu/es/prisma-sistema-de-gestion-academica-de-la-upc>. [Accedido: 06-feb-2015].
- [8] «siu.edu.ar», 07-feb-2015. [En línea]. Disponible en: <http://www.siu.edu.ar/>. [Accedido: 07-feb-2015].
- [9] «Servicio Central de Informática - Gestión\_Académica - Universidad de Málaga», 07-feb-2015. [En línea]. Disponible en: [http://www.uma.es/servicio-central-de-informatica/info/7892/gestion\\_academica/](http://www.uma.es/servicio-central-de-informatica/info/7892/gestion_academica/). [Accedido: 07-feb-2015].
- [10] «La Nueva Universidad en Informática 2009 | Cuba-L Direct – News and Views from Cuba». [En línea]. Disponible en: <http://www.cuba-l.com/la-nueva-universidad-en-informtica-2009/>. [Accedido: 07-may-2015].
- [11] Rodrigo Barrantes Echavarría, *Educación a Distancia*, Primera Edición. Universidad Estatal a Distancia, 1998.

- [12] Lorna Uden, Jane Sinclair, Yu/Hai, Diario Liberona, *Learning Technology for Education in Cloud, MOOC and Big Data*. Chile, Santiago de Chile: Springer International Publishing Switzerland, 2014.
- [13] «la web en la educación: LMS. DEFINICIÓN». [En línea]. Disponible en: [lawebenlaeducacion2010.blogspot.com/p/lms-definicion.html](http://lawebenlaeducacion2010.blogspot.com/p/lms-definicion.html). [Accedido: 15-feb-2015].
- [14] «Moodle - Open-source learning platform | Moodle.org». [En línea]. Disponible en: <https://moodle.org/about/>. [Accedido: 15-feb-2015].
- [15] «Bienvenidos al Entorno Virtual de Aprendizaje». [En línea]. Disponible en: <http://eva.uci.cu/>. [Accedido: 15-feb-2015].
- [16] Francesco Ricci, Lior Rokach, Bracha Shapira, y Paul B. Kantor, *Recommender Systems Handbook*. .
- [17] Jamil A.S. Itmazi, «Sistema Flexible de Gestión del eLearning para soportar el aprendizaje en las Universidades Tradicionales», Tesis Doctoral, Universidad de Granada.
- [18] Félix Óscar Rubio and Santos y Crescencio Bravo García, «Metodologías de Desarrollo de Software», *Metodologías de Desarrollo de Software*, 2010-2009. .
- [19] Dagmar Fernández Cuesta, «Tesis de Maestría en Informática Aplicada», La Habana, La Habana, 2011.
- [20] R. G. Figueroa, Armando A. Cabrera, y Camilo J. Solís, «Metodologías Tradicionales vs. Metodologías Ágiles». .
- [21] Bruno Gorjón Salvador y Alex Martelli, *Python. Guía de referencia*, Primera Edición. Anaya Multimedia-Anaya Interactiva, 2007.
- [22] Mark Lutz, *Learning Python*, Cuarta Edición. O'Reilly Media, Inc. (ed.), 1999.
- [23] *Web Development with java. Using Hibernate, JSPs, and Servlets*. .
- [24] Javier García de Jalón, Iñigo Mingo, Alfonso Brazález, Aitor Imaz, Alberto Larzabal, Jesús Calleja, y Jon García, *Aprenda Java como si estuviera en primero*. Escuela Superior de Ingenieros Industriales, 2000.
- [25] «PHP: Hypertext Preprocessor». [En línea]. Disponible en: <http://www.php.net/>. [Accedido: 28-abr-2014].
- [26] « ¿Que es Symfony?», *Desarrollo con Symfony*, 2014. .
- [27] Qiang Xue y Xiang Wei Zhuo, «The Definitive Guide to Yii 1.1», 2010 2008.
- [28] Pedro Boda, Kattia Ninahuanca, Javier Martinez, y Benjamín Gonzales, *Zend Framework Manual en Español*, 2005.<sup>a</sup>-2009.<sup>a</sup> ed. 2009.

- [29] Jonathan Chaffer y Karl Swedberg, *Learning jQuery*, 4ta ed. Birmingham B3 2PB, UK: Packt Publishing Ltd, 2013.
- [30] David Cochran, *Twitter Bootstrap Web Development How-To*, 4ta ed. Birmingham B3 2PB, UK: Packt Publishing Ltd, 2012.
- [31] Abraham Otero, «MySQL vs PostgreSQL ¿cuándo emplear cada una de ellas? javaHispano». [En línea]. Disponible en: [http://www.javahispano.org/contenidos/es/mysql\\_vs\\_postgresql\\_cuando\\_emplear\\_cada\\_una\\_de\\_ellas\\_11/](http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11/). [Accedido: 15-feb-2015].
- [32] Lan Witten y Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Tercera Edición. .
- [33] «Apache Mahout: Scalable machine learning and data mining». [En línea]. Disponible en: <http://mahout.apache.org/>. [Accedido: 09-feb-2015].
- [34] «Introducing Apache Mahout», 09-feb-2015. [En línea]. Disponible en: <http://www.ibm.com/developerworks/java/library/j-mahout/>. [Accedido: 09-feb-2015].
- [35] «Eclipse - The Eclipse Foundation open source community website.» [En línea]. Disponible en: [http://www.eclipse.org/#sec\\_ide](http://www.eclipse.org/#sec_ide). [Accedido: 05-jun-2015].
- [36] «PHP IDE :: JetBrains PhpStorm». [En línea]. Disponible en: <https://www.jetbrains.com/phpstorm/>. [Accedido: 05-jun-2015].
- [37] «Conceptos básicos del servidor web». [En línea]. Disponible en: [http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php). [Accedido: 05-jun-2015].
- [38] Kent Beck, *Una explicación de la programación extrema: aceptar el cambio*. Pearson Educación.
- [39] Jesús Barranco de Areba, *Modelización Conceptual de Datos*, ICAI, ICADE. Madrid, 2001.
- [40] Patricio Letelier, «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)», presentado en Metodologías ágiles para el desarrollo de software, Departamento de Sistemas Informáticos y Computación (DSIC), Universidad Politécnica de Valencia (UPV), 2006, vol. 05.
- [41] «Desarrollo de software. Tarjetas CRC | Jummp». [En línea]. Disponible en: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>. [Accedido: 03-jun-2015].

- [42] Isidro Ramos Salavert y Maria Dolores Lozano Perez, *Ingeniería de Software y Bases de Datos. Tendencias Actuales*, 4ta ed. Ediciones de la Universidad de Castilla-La Mancha, 2000.
- [43] «2.1. El patrón MVC (Symfony 1.2, la guía definitiva)». [En línea]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_2/capitulo\\_2/el\\_patron\\_mvc.html](http://librosweb.es/libro/symfony_1_2/capitulo_2/el_patron_mvc.html). [Accedido: 11-abr-2015].
- [44] Vicente Javier Esleva Muñoz, *El nuevo PHP. Conceptos avanzados*. España: Bubok Publishing S.L., 2013.
- [45] Craig Larman, *UML y Patrones: introducción al análisis y diseño orientado a objetos*, Segunda. 1999.
- [46] Larman, Craig, *UML y Patrones. 2002*, Segunda. Pearson Prentice Hall, Craig Larman.
- [47] Abraham Silberschatz y Henry F. Korth, *Database System Concepts*, Sexta. McGraw-Hill, 2015.
- [48] Lic. Pablo Andrés Barrientos, «Enfoque para pruebas de unidad basado en la generación aleatoria de objetos», Universidad Nacional de la Plata, La Plata, Argentina, 2014.
- [49] «Probando — Manual de Symfony2 en Español». [En línea]. Disponible en: <http://gitnacho.github.io/symfony-docs-es/book/testing.html>. [Accedido: 03-jun-2015].
- [50] Patricio Letelier Torres, «¿Qué es una Prueba de Aceptación? | LinkedIn», 21-oct-2010. [En línea]. Disponible en: [https://www.linkedin.com/grp/post/3636186-48805747?goback=%2Egde\\_3636186\\_member\\_48805747](https://www.linkedin.com/grp/post/3636186-48805747?goback=%2Egde_3636186_member_48805747). [Accedido: 11-may-2015].
- [51] ing. Ernesto Vladimir Pereda Díaz y ing. Yaismel Miranda Pons, «Pautas de Codificación». 09-oct-2013.
- [52] «Estándares de codificación — Manual de Symfony2 en Español». [En línea]. Disponible en: <http://gitnacho.github.io/symfony-docs-es/contributing/code/standards.html>. [Accedido: 03-jun-2015].