



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 1**

**“Sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.”**

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

**Autores:**

Osiel Victor Vargas Díaz

Taimi Dalila Corzo Guzmán

**Tutores:**

MsC Ailec Granda Dihigo

MsC Violena Hernández Aguilar

Ing. Geidis Sánchez Michel

La Habana, Junio de 2013

“Año 55 de la Revolución.”

## Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a que haga uso ilimitado de este. Queda confirmada la presente declaración a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2013.

### **Autores:**

Taimi Dalila Corzo Guzmán

Osiel Víctor Vargas Díaz

\_\_\_\_\_

\_\_\_\_\_

### **Tutores:**

Ailec Granda Dihigo

Violena Hernández

Geidis Michel Sánchez

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

*La posibilidad de realizar un sueño es lo que hace que la vida  
sea interesante.*

*Paulo Coelho*

## **Dedicatoria**

***Osiel Victor:***

*A mi mamá, a mi hermanita Anaisel, a mi novia, a mis cuñadas y a mis suegros.*

***Taimi Dalila:***

*A mi mamá, a mi papá Orlando, a mi abuela, a mis hermanas Marifyn y litay y a mi novio Osiel Victor.*

## **Agradecimientos**

### ***Osiel Victor:***

*A mi mamá y mi papá "pepe" por siempre confiar en mí y guiarme cuando aún no sabía que era lo que deseaba para mi futuro.*

*A mis suegros por ayudarme en todo lo que necesité y apoyarme en cada momento.*

*A mi novia por siempre confiar en mí y brindarme todo su amor y cariño.*

*A mis tutoras Ailec, Geidis y Violena por ser tan atentas en el transcurso de este trabajo, por compartir conmigo su tiempo libre y prestarme toda la atención del mundo.*

*En fin, a todo aquel que de una manera u otra, aportó su grano de arena, por así decirlo, en el desarrollo de esta tesis, que con tanto trabajo y dedicación la realizamos mi compañera y yo.*

### ***Taimi Dalila:***

*Esta carrera para mí fue un gran reto en mi vida, ya que muchas personas dudaron de que yo rindiera en la misma, pero con la ayuda de mis seres queridos avancé, aunque en algunas ocasiones veía que estos agradecimientos se me alejaban de mi vida.*

*Agradezco:*

*A mi mamá Sonia María, a mi abuela María Paneque, a mi papá Orlando Sánchez, mis hermanas Litay y Marilyn, ya que lloraron junto a mí en cada*

*momento triste y feliz de mi carrera. Mamita linda, te agradezco todo porque eres lo más lindo de mi vida y gracias por estar aquí conmigo en este momento feliz.*

*A mi carú (Victor) ya que nunca me abandonó en los momentos duros y siempre me dio su dedicación, para así demostrarme de que yo si podía lograr lo que quisiera en la vida.*

*A mis amigas María y Mailen por estar en el momento indicado de mi vida, y por confiar en mí, las quiero.*

*A Leonardo y a PB (Juan Manuel) por siempre ser mis mejores amigos y quererme mucho, sabiendo lo pesadita que soy.*

*A Mayelín, Damián, Matilde mi mamita, Geidis Sánchez, Eileén, Yari, Ilmaris, Enrique mi profesor de programación que confió mucho en mí, al profesor Hilario, Renier Limonta profesor de tele, gracias por confiar en mí, a mi amigo Walber por siempre estar ahí para mí y a Llorente.*

*A todas las personas que siempre se alegraron de las cosas malas de mi vida, ya que gracias a eso me volví más fuerte y pude levantarme de mis caídas con la cabeza en alto. Quiero que miren bien porque “Taimi Dalila llegó a la meta”.*

*A mis tutoras Ailec, Violena y Geidis por apoyarnos mucho en este trabajo y darnos a mi compañero y a mí, fuerzas para llegar al final de esta carrera.*

*Gracias a todos por hacer posible mi sueño y estar siempre conmigo, los quiero.*

## Resumen

En el Departamento Metodológico Central de Ingeniería y Gestión de Software de la Universidad de las Ciencias Informáticas se desarrollan diferentes procesos, como es el caso de la gestión de las preparaciones metodológicas, la gestión de los planes de trabajo de los asesores y la planificación y seguimiento de los controles a clases que se realizan en las diferentes asignaturas. En la actualidad estos procesos se gestionan de forma manual, lo que hace el trabajo más engorroso. En ocasiones existe duplicidad de información, retraso en la entrega de la documentación necesaria en algún proceso y se crean volúmenes muy extensos de información. Todo esto hace necesario el desarrollo de un sistema informático que contribuya a mejorar la gestión de los procesos que se desarrollan en el departamento, dando solución a los problemas y necesidades que existen actualmente en el mismo. Para la realización de dicho sistema fue necesaria la utilización de las herramientas y tecnologías adecuadas. Al concluir este trabajo se logró satisfacer las necesidades anteriormente planteadas, a partir del diseño e implementación de las funcionalidades previstas, las cuales fueron validadas a través del desarrollo de diferentes tipos de pruebas.

**Palabras claves:** Departamento Metodológico Central, Ingeniería y Gestión de Software, procesos, preparaciones metodológicas, controles a clases, plan de trabajo.

## Índice general

<b>Introducción</b> .....	1
<b>Capítulo 1: “Fundamentación Teórica”</b> .....	4
1.1 Introducción.....	4
1.2 El proceso docente en la educación superior .....	4
1.2.1 Conceptos asociados al tema .....	4
1.3 Departamento de Ingeniería y Gestión de Software en la Universidad de las Ciencias Informáticas.	6
1.4 Sistemas para gestionar procesos en departamentos docentes.....	8
1.5 Propuesta de solución .....	11
1.6 Tecnologías que se utilizarán en la propuesta de solución .....	11
1.6.1 Metodologías de desarrollo .....	11
1.6.2 Lenguaje de modelado .....	13
1.6.3 Lenguajes de programación web .....	13
1.6.4 Marco de trabajo a utilizar .....	15
1.6.5 Herramienta de modelado .....	17
1.6.6 Sistemas gestores de bases de datos .....	17
1.6.7 Entorno de desarrollo integrado .....	18
1.6.8 Servidor Web Apache.....	19
1.7 Conclusiones parciales.....	19
<b>Capítulo 2: “Características del sistema. Elaboración y Planificación”</b> .....	20
2.1 Introducción.....	20
2.2 Flujo actual de los procesos .....	20
2.3 Objeto de automatización .....	22
2.4 Requerimientos del sistema .....	22
2.5 Propuesta de la aplicación.....	24
2.5.1 Identificación de los roles .....	25
2.6 Fase de Exploración .....	25
2.6.1 Historias de usuarios .....	25
2.7 Fase de Planificación.....	28
2.7.1 Estimación por historia de usuario .....	28
2.7.2 Plan de iteraciones .....	30
2.7.3 Plan de duración de iteraciones .....	31
2.7.4 Plan de entrega .....	33
2.8 Conclusiones parciales.....	35
<b>Capítulo 3: “Implementación y Pruebas”</b> .....	36
3.1 Introducción.....	36
3.2 Diseño del sistema.....	36
3.2.1 Tarjetas CRC (Clase-Responsabilidad-Colaboración). .....	38
3.2.2 Modelo de datos .....	39



3.3	Patrones de diseño .....	40
3.3.1	Modelo de despliegue .....	42
3.3.2	Estándar de codificación .....	42
3.4	Pruebas .....	43
3.5	Conclusiones parciales.....	48
	<b>Conclusiones generales.....</b>	<b>49</b>
	<b>Recomendaciones .....</b>	<b>50</b>
	<b>Referencias bibliográficas .....</b>	<b>51</b>
	<b>Bibliografías .....</b>	<b>54</b>
	<b>Glosario de términos.....</b>	<b>57</b>
	<b>Anexos .....</b>	<b>59</b>
	Anexo_1: Fichas de los principales procesos .....	59
	Anexo_2: Flujo de actividades de los principales procesos .....	60
	Anexo_3: Historias de usuarios .....	64
	Anexo_4: Prototipos de interfaz .....	76
	Anexo_5: Tarjetas CRC.....	79
	Anexo_6: Casos de pruebas .....	82

## Índice de tablas

Tabla 1: Ficha de proceso “Gestión de los controles a clases” .....	20
Tabla 2: Flujo de actividades del proceso “Gestión de los controles a clases” .....	21
Tabla 3: Lista de requisitos del sistema .....	22
Tabla 4: Relación de roles del sistema .....	25
Tabla 5: HU “Crear la planificación de los controles a clases” .....	26
Tabla 6: HU “Crear la planificación de las preparaciones metodológicas” .....	27
Tabla 7: Estimación de esfuerzo por HU .....	28
Tabla 8: Plan de duración de las iteraciones .....	31
Tabla 9: HU abarcadas en módulos .....	33
Tabla 10: Plan de duración de la entrega del sistema.....	34
Tabla 11: Tarjeta CRC de los controles a clases.....	38
Tabla 12: Tarjeta CRC de las preparaciones metodológicas .....	38
Tabla 13: Tarjeta CRC del plan de trabajo mensual.....	39
Tabla 14: Transformación de la base de datos .....	39
Tabla 15: Caso de prueba "Crear un grupo" .....	45
Tabla 16: Ficha de proceso “Gestión de las preparaciones metodológicas” .....	59
Tabla 17: Ficha de proceso “Elaboración de los planes de trabajo” .....	59
Tabla 18: Flujo de actividades del proceso “Gestión de las preparaciones metodológicas” .....	60
Tabla 19: Flujo de actividades del proceso “Elaboración de los planes de trabajo” .....	62
Tabla 20: HU “Modificar la planificación de los controles a clases” .....	64
Tabla 21: HU “Adicionar un control a clase a un asesor” .....	64
Tabla 22: HU “Modificar un control a clase a un asesor” .....	64
Tabla 23: HU “Eliminar un control a clase a un asesor” .....	64
Tabla 24: HU “Eliminar la planificación de los controles a clases” .....	65
Tabla 25: HU “Mostrar la planificación de los controles a clases” .....	65
Tabla 26: HU “Dar seguimiento a un control a clase” .....	65
Tabla 27: HU “Generar reporte total de evaluaciones de controles a clases por asignatura” .....	66
Tabla 28: HU “Generar reporte total de evaluaciones de controles a clases” .....	66
Tabla 29: HU “Generar reporte del estado actual de controles a clases por asignatura” .....	66
Tabla 30: HU “Generar reporte del estado actual de la planificación de controles a clases” .....	66
Tabla 31: HU “Adicionar nomenclador” .....	67
Tabla 32: HU “Modificar nomenclador” .....	67
Tabla 33: HU “Eliminar nomenclador” .....	67
Tabla 34: HU “Modificar la planificación de las preparaciones metodológicas” .....	68
Tabla 35: HU “Eliminar preparaciones metodológicas” .....	68
Tabla 36: HU “Mostrar la planificación de las preparaciones metodológicas” .....	68
Tabla 37: HU “Dar seguimiento a las preparaciones metodológicas” .....	69
Tabla 38: HU “Generar actas de las preparaciones metodológicas” .....	69
Tabla 39: HU “Imprimir actas de las preparaciones metodológicas” .....	69
Tabla 40: HU “Modificar acta de las preparaciones metodológicas” .....	70
Tabla 41: HU “Eliminar acta de las preparaciones metodológicas” .....	70
Tabla 42: HU “Planificar visitas a las preparaciones metodológicas” .....	70
Tabla 43: HU “Visualizar la planificación de las visitas a las preparaciones metodológicas” .....	70

Tabla 44: HU “Modificar la planificación de las visitas a las preparaciones metodológicas” .....	71
Tabla 45: HU “Eliminar visitas a las preparaciones metodológicas” .....	71
Tabla 46: HU “Emitir valoración de la preparación metodológica” .....	71
Tabla 47: HU “Crear plan de trabajo a los asesores” .....	72
Tabla 48: HU “Mostrar plan de trabajo de los asesores” .....	72
Tabla 49: HU “Modificar plan de trabajo a los asesores” .....	72
Tabla 50: HU “Crear evaluación del profesor” .....	72
Tabla 51: HU “Modificar evaluación del profesor” .....	73
Tabla 52: HU “Eliminar evaluación del profesor” .....	73
Tabla 53: HU “Mostrar las evaluaciones de los profesores” .....	73
Tabla 54: HU “Generar reportes de las evaluaciones de los profesores” .....	74
Tabla 55: HU “Generar corte de resumen de evaluaciones de los profesores” .....	74
Tabla 56: HU “Adicionar usuario” .....	74
Tabla 57: HU “Modificar usuario” .....	74
Tabla 58: HU “Eliminar usuario” .....	75
Tabla 59: HU “Autenticar usuario” .....	75
Tabla 60: Tarjeta CRC de facultad .....	79
Tabla 61: Tarjeta CRC de local .....	79
Tabla 62: Tarjeta CRC de grupo .....	79
Tabla 63: Tarjeta CRC de departamento .....	79
Tabla 64: Tarjeta CRC de disciplina .....	80
Tabla 65: Tarjeta CRC de asignatura .....	80
Tabla 66: Tarjeta CRC de usuario .....	80
Tabla 67: Tarjeta CRC de categoría docente .....	80
Tabla 68: Tarjeta CRC de profesor .....	81
Tabla 69: Tarjeta CRC de tarea .....	81
Tabla 70: Caso de prueba “Crear departamento” .....	82
Tabla 71: Caso de prueba “Modificar departamento” .....	84
Tabla 72: Caso de prueba “Eliminar departamento” .....	86
Tabla 73: Caso de prueba “Crear facultad” .....	87
Tabla 74: Caso de prueba “Modificar facultad” .....	89
Tabla 75: Caso de prueba “Eliminar facultad” .....	92
Tabla 76: Caso de prueba “Modificar grupo” .....	93
Tabla 77: Caso de prueba “Eliminar grupo” .....	96
Tabla 78: Caso de prueba “Crear local” .....	97
Tabla 79: Caso de prueba “Modificar local” .....	100
Tabla 80: Caso de prueba “Eliminar local” .....	101
Tabla 81: Caso de prueba “Crear Disciplina” .....	102
Tabla 82: Caso de prueba “Eliminar disciplina” .....	105
Tabla 83: Caso de prueba “Crear asignatura” .....	106
Tabla 84: Caso de prueba “Modificar asignatura” .....	109
Tabla 85: Caso de prueba “Eliminar asignatura” .....	110
Tabla 86: Caso de prueba “Crear categoría docente” .....	111
Tabla 87: Caso de prueba “Modificar categoría docente” .....	112

Tabla 88: Caso de prueba "Eliminar categoría docente" .....	113
Tabla 89: Caso de prueba "Criterio de búsqueda de los controles a clases por asesor" .....	114
Tabla 90: Caso de prueba "Criterio de búsqueda de los controles a clases por asignatura" .....	114
Tabla 91: Caso de prueba "Criterio de búsqueda por rango de fecha de los controles a clases" .....	115
Tabla 92: Caso de prueba "Asignar visitante a las preparaciones metodológicas" .....	116
Tabla 93: Caso de prueba "Criterio de búsqueda para las preparaciones por asignatura" .....	116
Tabla 94: Caso de prueba "Criterio de búsqueda para las preparaciones metodológicas por disciplina" ..	117
Tabla 95: Caso de prueba "Criterio de búsqueda para las preparaciones metodológicas por rango de fecha" .....	118
Tabla 96: Caso de prueba "Asignar tareas a un plan de trabajo" .....	118
Tabla 97: Caso de prueba "Modificar un plan de trabajo" .....	119
Tabla 98: Caso de prueba "Eliminar un plan de trabajo" .....	119
Tabla 99: Caso de prueba "Criterio de búsqueda para de los planes de trabajo por asesor" .....	120
Tabla 100: Caso de prueba "Criterio de búsqueda de los planes de trabajo por rango de fecha" .....	121
Tabla 101: Caso de prueba "Crear evaluación" .....	121
Tabla 102: Caso de prueba "Eliminar tareas al plan de trabajo" .....	122
Tabla 103: Caso de prueba "Dar cumplimiento a las tareas del plan de trabajo" .....	123
Tabla 104: caso de prueba "Generar reporte por asignatura de los controles a clases" .....	123
Tabla 105: Caso de prueba "Generar reporte del estado actual de los controles a clases por asignatura" .....	124
Tabla 106: Caso de prueba "Generar reporte del estado actual de los controles a clases" .....	124
Tabla 107: Caso de prueba "Generar reporte del estado actual de las evaluaciones de los asesores" ....	125

## Índice de ilustraciones

Ilustración 1: Prototipo de interfaz “Crear la planificación de los controles a clases” .....	27
Ilustración 2: Prototipo de interfaz “Crear la planificación de las preparaciones metodológicas”.....	28
Ilustración 3: Modelo Vista Controlador (Potencier, 2011) .....	37
Ilustración 4: Modelo de datos .....	40
Ilustración 5: Modelo de despliegue .....	42
Ilustración 6: Validación de la entidad “Facultad” .....	44
Ilustración 7: Resultado de la prueba unitaria “Validación de la entidad Facultad” .....	44
Ilustración 8: Diagrama de deficiencias.....	48
Ilustración 9: Prototipo de interfaz “Dar seguimiento al control a clases” .....	76
Ilustración 10: Prototipo de interfaz “Asignar visitante a la preparación metodológica” .....	76
Ilustración 11: Prototipo de interfaz “Dar seguimiento a la preparación metodológica” .....	77
Ilustración 12: Prototipo de interfaz “Crear planificación de los planes de trabajo” .....	77
Ilustración 13: Prototipo de interfaz “Asignar tareas a un plan de trabajo” .....	78
Ilustración 14: Prototipo de interfaz “Autenticar usuario” .....	78

## Introducción

El desarrollo de la humanidad crece cada día gracias a todos los avances ocurridos en las diferentes esferas; por ejemplo, los avances científico-técnicos en las distintas ramas, tales como, la informática y las comunicaciones. La industria del software ha alcanzado una posición relevante, por su característica de controlar o hacer accesible los adelantos informáticos, en la mayoría de los casos. Esta industria posee una variedad de aplicaciones que pueden ser utilizadas en cualquier otra rama de la ciencia o la economía de un país. La industria de las tecnologías de la información y las comunicaciones (TIC) está fuertemente sujeta al desarrollo económico de cada nación, por lo que la industria del software, como apéndice de la industria de las nuevas tecnologías, se encuentra bajo las mismas condiciones (Hernández, 2005).

La gestión de procesos en cualquier organización es vital, puesto que conlleva a alcanzar resultados satisfactorios a partir de la planificación que se realice. Los mismos benefician a la organización en su participación competitiva en los actuales escenarios, contribuyendo a una eficiente organización y desarrollo de las actividades. A través del modelado de procesos puede lograrse un mejor entendimiento del negocio. En el mundo existen diversos sistemas para gestionar la información, los que han contribuido a que se gestione mejor los procesos y que la información que se genere esté más organizada.

En las universidades actuales se hace necesaria la informatización de los procesos que se desarrollan para así poder mantener un mayor control sobre estos. Cuba no está ajena a este fenómeno tan necesario de informatizar la sociedad, por esa razón se crea la Universidad de las Ciencias Informáticas (UCI), teniendo entre sus tareas más importantes, informatizar el país. En esta institución se aplica un modelo de integración que vincula los procesos de docencia, investigación y producción. En el caso de la docencia, la universidad tiene definida una estructura, la cual está compuesta por departamentos docentes en las facultades y departamentos metodológicos centrales. Estos últimos son los encargados de orientar metodológicamente las actividades relacionadas con las disciplinas existentes en cada uno de ellos, ya sea Programación, Matemática, Ingeniería y Gestión de Software, entre otros. En este caso se encuentra el Departamento Metodológico Central de Ingeniería y Gestión de Software.

En dicho departamento se desarrollan diversos procesos, como es la gestión de las preparaciones metodológicas, la gestión de los controles a clases, la gestión del plan de trabajo individual de los asesores. Estos procesos en la actualidad se realizan de forma manual, por lo que el trabajo del personal involucrado es más engorroso, identificándose los siguientes problemas:

- Se crea un volumen de información muy extenso.
- A veces existe duplicidad de la información.
- Existe retraso en la entrega de la documentación necesaria en algún proceso.

- Cada trabajador guarda su información de forma independiente, lo cual dificulta la gestión de la misma.

Todo esto hace necesario la creación de un mecanismo que mejore la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.

Luego del análisis anterior, surge el siguiente **problema a resolver**: ¿cómo mejorar la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software de la Universidad de las Ciencias Informáticas? Esta problemática se enmarca en el **objeto de estudio**: la gestión de los procesos docentes en la educación superior. Para dar una solución efectiva al problema se propone como **objetivo**: desarrollar un sistema utilizando tecnologías web que contribuya a mejorar la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software de la Universidad de las Ciencias Informáticas. El objetivo planteado anteriormente delimita el **campo de acción**: sistemas automatizados para gestionar procesos docentes en departamentos docentes-metodológicos de la educación superior.

Como **objetivos específicos** se pueden mencionar los siguientes:

1. Definir los componentes teóricos y tecnológicos necesarios para la implementación del sistema.
2. Caracterizar los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software para su futura informatización.
3. Diseñar una aplicación web que gestione los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.
4. Implementar una aplicación web que gestione los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.
5. Verificar las funcionalidades implementadas realizando pruebas.

Para dar cumplimiento a los objetivos de la investigación se desarrollarán las **tareas** siguientes:

1. Estudio de sistemas de gestión de procesos docentes más significativos a nivel nacional e internacional.
2. Análisis crítico de las tecnologías necesarias para implementar aplicaciones web que permita la definición de cuáles utilizar en el desarrollo de la aplicación.
3. Análisis de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software para la definición de los participantes, datos de entrada y salida y futuras entidades del sistema.
4. Identificación de las actividades a informatizar en los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software, así como la estructura de la información generada por los mismos.
5. Aplicación de una metodología de desarrollo de software con todos sus flujos de trabajo.

6. Diseño de una base de datos que soporte el sistema.
7. Diseño de los elementos que componen el modelo, la vista y el controlador de la aplicación propuesta.
8. Implementación del sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.
9. Desarrollo de pruebas funcionales y unitarias al sistema implementado.

En el trabajo se hace necesaria la utilización de varios métodos de investigación, teóricos y empíricos. Como **métodos teóricos** se pueden mencionar el **histórico-lógico**, el cual es de gran utilidad para realizar un estudio de la evolución de las herramientas necesarias para el desarrollo de esta investigación. Además, el **analítico-sintético** para realizar un análisis de las tendencias en la personalización de las herramientas, el sistema y los procesos que se desarrollan en el departamento, necesarios para el desarrollo del software. Por último, la **modelación** para modelar los procesos del negocio que se desarrollan en el departamento, así como en el análisis y diseño de la investigación efectuada. Como **método empírico** se puede citar las **entrevistas**, que son un ejemplo de método de recopilación de información, con el objetivo de poder identificar los posibles procesos que se desarrollan en el departamento y las principales funcionalidades que debe cumplir la aplicación a desarrollar.

El documento se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

**Capítulo 1: “Fundamentación teórica”:** se realiza un análisis del estado del arte referente al tema que se trabaja en esta investigación, abordándose los principales conceptos asociados al dominio del problema y las tecnologías a utilizar para el desarrollo de la solución.

**Capítulo 2: “Características de la aplicación. Elaboración y Planificación”:** se presentan los resultados de las fases de exploración y planificación de la propuesta de solución.

**Capítulo 3: “Implementación y Pruebas”:** se realiza el diseño del sistema, presentándose además los resultados de las fases de implementación y pruebas.



# Capítulo 1: “Fundamentación Teórica”

## 1.1 Introducción

En el presente capítulo se realiza un análisis del estado del arte referente al tema que se trabaja en esta investigación. Se abordan los principales conceptos asociados al dominio del problema y las tecnologías a utilizar para el desarrollo de la solución. Se realiza además, un estudio de los sistemas existentes asociados al campo de acción planteado, con el objetivo de analizar si pueden ser utilizados en este entorno o, en caso negativo, tomar de estos las funcionalidades que contribuyan a resolver la situación problemática.

## 1.2 El proceso docente en la educación superior

En la actualidad el mundo necesita que se desarrollen cada vez más los procesos formativos, en los cuales se garantiza una formación básica de los estudiantes que pasan por los diferentes niveles de enseñanza. En el caso de la educación superior, esto adquiere una importancia significativa, pues es el momento donde los jóvenes se forman como futuros profesionales. Es por ello que las universidades establecen como una prioridad, el adecuado desarrollo del proceso docente, garantizando el cumplimiento de los planes de estudios, así como todo el aseguramiento necesario para ello.

Cuba no se encuentra ajena a esto, por lo que ha definido en sus diferentes instituciones de educación superior, una estructura departamental, que gestione el proceso docente, incluyendo los niveles de carreras, disciplinas y asignaturas. En este sentido, los diferentes departamentos, gestionan un grupo de procesos, que tributan al logro de los objetivos y metas definidas en cada área.

### 1.2.1 Conceptos asociados al tema

Durante el desarrollo de la investigación y en el presente documento, se utilizan términos que por su importancia, deben ser abordadas las definiciones que existen relacionadas a ellos.

A continuación se muestran diferentes definiciones de **proceso**:

- Es un conjunto lógico de actividades relacionadas y conectadas que toma entrada de proveedores, les añade valor y produce unas salidas para sus clientes (Álvaro, 2001).
- Es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso tiene sus entradas, funciones y salidas. Las entradas son requisitos que deben tenerse antes de que una función pueda ser aplicada. Cuando

una función es aplicada a las entradas de un método, se tendrán ciertas salidas resultantes (Garrido, 2011).

- Secuencia de actividades lógicas, diseñadas para generar una salida preestablecida para unos clientes identificados a partir de un conjunto de entradas necesarias que van añadiendo valor (Camisón, 2009).

En la investigación se decide tomar como concepto de referencia el dado por Miguel Montero Garrido, pues se considera que reúne todos los elementos que caracterizan de forma ampliada el concepto de proceso.

Otro término a abordar es el de **proceso docente**, el cual también ha sido definido por varios autores:

- Proceso docente es una parte de una institución docente y se proyecta en la sociedad y es la encargada de educar al hombre para la vida, a partir de compromisos sociales (Flores, 2007).
- El proceso docente es un fenómeno universal requerido para la continuidad cultural, a través del cual una generación prepara a otra que le sucede (Triana, 2010).

Como definición de referencia, se toma en la investigación, la planteada por Israel Mazarío Triana, puesto que se considera la más abarcadora del concepto de proceso docente.

Como término importante se tiene además, el de **gestión por procesos**, diferentes autores lo definen de la siguiente forma:

- La gestión por procesos se centra en la identificación, control y mejora de los diferentes procesos, para así obtener un resultado conforme al cliente (García, 2010).
- La gestión por procesos es una disciplina de gestión que ayuda a la dirección de la empresa a identificar, representar, diseñar, formalizar, controlar, mejorar y hacer más productivos los procesos de la organización para lograr la confianza del cliente (Carrasco, 2011).

Para la investigación se asume el dado por Juan Bravo Carrasco, ya que este es uno de los que engloba de mejor forma las características de la gestión por procesos.

El último concepto a tratar, es el de **departamento docente**. Entre las definiciones encontradas se tienen las siguientes:

- El departamento docente es el nivel de dirección que rige el trabajo de los profesores y se ocupa directamente de proyectar un trabajo metodológico que priorice la formación y desarrollo de valores en los estudiantes, desde el contenido de las asignaturas y disciplinas que son de su competencia (Batista, 2011).
- Es el órgano metodológico donde tiene lugar la reunión e intercambio de los docentes de las asignaturas de un área de conocimientos, con el objetivo de lograr puntos de encuentro y cooperación entre las diferentes disciplinas y evitar la multiplicidad de influencias en los estudiantes según el diagnóstico, para asegurar la calidad del proceso de enseñanza-aprendizaje. Es la célula básica del trabajo metodológico (Zayas, 1999).

Se decide tomar como concepto de referencia el dado por Carlos Álvarez de Zayas, por ser este el concepto más completo, referente a los departamentos docentes.

### **1.3 Departamento de Ingeniería y Gestión de Software en la Universidad de las Ciencias Informáticas**

La Universidad de las Ciencias Informáticas (UCI) tiene entre sus principales misiones: (Gil-Morell, 2006)

- ✓ Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la informática.
- ✓ Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.
- ✓ Servir de soporte a la industria cubana de la informática.

En esta universidad se desarrollan una serie de actividades relacionadas con la docencia, producción e investigación, ejes fundamentales del modelo de integración que en ella se desarrolla.

La universidad cuenta con varios departamentos metodológicos centrales, como es el caso del Departamento Metodológico Central de Ingeniería y Gestión de Software, el cual tiene como principal función, orientar metodológicamente al claustro de las diferentes disciplinas que lo conforman. Actualmente se desarrollan diferentes procesos en el departamento, como es el caso de la gestión de evaluaciones profesoras de los asesores, la planificación de los controles a clases, la gestión de las preparaciones metodológicas y la gestión de los planes de trabajo individual de los asesores.

#### **Descripción preliminar de los procesos**

Evaluaciones profesoriales de los asesores: se evalúa anualmente a cada profesor o asesor, teniendo en cuenta las actividades desarrolladas en el curso, entre ellas: actividad docente, investigativa y productiva. Actualmente se hace engorrosa y difícil la elaboración de la evaluación, pues no existe un espacio común donde se tenga el plan de trabajo del profesor, el cumplimiento del mismo y los principales resultados obtenidos.

Preparaciones metodológicas: se desarrollan con el objetivo de orientar metodológicamente a los profesores, a partir del análisis de las principales actividades, el desarrollo de ciclos de trabajo metodológico y la realización de actividades de superación en contenidos de la disciplina. Tienen una gran importancia, puesto que contribuyen a aumentar la calidad de las clases impartidas por los diferentes docentes. En ocasiones, la información relacionada con su planificación y desarrollo no llega a todos los involucrados, es decir, no tienen claridad de la fecha, lugar y puntos a debatir en la misma. Además, no se tiene automatizado el tema del control de la asistencia y la generación del acta de la preparación metodológica, lo cual dificulta también, el desarrollo del proceso.

La gestión de los controles a clases es otro de los principales procesos desarrollados en el departamento, estos tienen como principal objetivo controlar el cumplimiento de los objetivos y orientaciones realizadas durante los ciclos de trabajo metodológico, lo cual se debe desencadenar en el desarrollo de clases de calidad. En ocasiones se dificulta la planificación y seguimiento de los controles realizados, haciéndose engorrosa la generación de reportes o resúmenes de cantidad de controles, principales resultados, entre otros.

La elaboración de los planes de trabajo individual de los asesores es otro de los procesos que se desarrollan en el departamento, el cual les sirve a los asesores como guía para su labor diaria, pues en este se reflejan todas las actividades propuestas por la facultad y el departamento. Estos planes se realizan mensualmente y constituyen la base para la elaboración de las evaluaciones profesoriales. En ocasiones este plan no se confecciona ni publica con anterioridad, lo cual provoca que los miembros del departamento estén desorientados y se ausenten a alguna de las actividades previstas en el plan. Por otra parte, el jefe de departamento debe tener conocimiento de las tareas en las cuales están involucrados sus subordinados, siendo engorrosa la obtención de esta información actualmente, por la carencia de un sistema que informatice esta gestión.

Lo expresado anteriormente, conlleva a la necesidad de gestionar los procesos que se desarrollan en el departamento de forma automatizada.

## 1.4 Sistemas para gestionar procesos en departamentos docentes

A continuación se hace una caracterización de los diferentes sistemas internacionales estudiados para el desarrollo de la investigación, de los cuales se señalarán sus beneficios y deficiencias.

**SFIUJB:** Sistema de la Facultad de Ingeniería de la Universidad Javeriana de Bogotá. (Rivera, 2000)

Dentro de sus principales funcionalidades, se debe destacar que permite conocer todo lo referido a los diferentes departamentos existentes en la universidad, pues se visualizan las actividades de estos. Fue desarrollado con HTML, CSS y JavaScript, como lenguajes del lado del cliente, además, utiliza librerías de *jQuery*. Se debe señalar que los procesos automatizados en dicho sistema, no se asemejan a los que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software de la UCI, por lo que no es viable utilizarlo para dar solución a la problemática planteada en esta investigación. Además, no cumple con los parámetros visuales que el cliente desea para su aplicación.

**SFCEUNCBA:** Sistema de la Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la provincia de Buenos Aires. (Vénere, 1998)

El SFCEUNCBA se centra en las caracterizaciones de las investigaciones de las ciencias exactas, donde se incluyen las diferentes informaciones referidas a los estudiantes y graduados. Está desarrollado con la combinación de los lenguajes del lado del cliente: HTML, JavaScript y CSS, aunque algunas de sus páginas están hechas en XML, además, utiliza *jQuery* para la realización de consultas. Presenta una funcionalidad que se pudiera seleccionar para ser incluida en la propuesta de solución; es la referida a los planes de trabajo. En la misma se muestra información sobre las acciones y actividades que se realizan en la universidad durante un período determinado. Presenta algunas dificultades, pues no tiene la información de forma centralizada, lo que conlleva a que el visitante no se oriente de una manera adecuada. Además, el diseño visual del mismo no responde a los requerimientos del usuario de la propuesta de solución.

**UNL:** Universidad Nacional de Luján de Buenos Aires. (Dibbern, 1999)

Este sistema gestiona diferentes procesos educativos. Presenta un diseño visual adecuado y brinda una amplia explicación de dichos procesos, entre ellos: carreras de pregrado, carreras de postgrado, ofertas educativas e información académica. Está compuesto por diferentes departamentos docentes, mostrándose las funciones de cada uno de estos. Del mismo, se puede tomar para el sistema a desarrollar, la funcionalidad: calendario de actividades, ya que las mismas se planifican con una periodicidad mensual. El sistema está desarrollado con HTML, CSS y JavaScript, como lenguajes del lado

del cliente. Por lo demás, presenta como deficiencia que no tiene la información organizada, ya que está dividido por muchas páginas, afectando la usabilidad del mismo.

**SUCDSI:** Sistema de la Universidad de Castilla-La Mancha del Departamento de Sistemas Informáticos. (Barra, 2001)

De este sistema se puede tomar la seguridad que presenta en los datos de los estudiantes y profesores. La funcionalidad referente al plan de estudio, proporciona una breve explicación de lo que se trata en la rama de la Ingeniería. Fue desarrollado utilizando la combinación de XHTML, JavaScript y CSS, como programación del lado del cliente. Tiene como dificultad que no muestra los departamentos que lo componen y está más centrado en la información para los estudiantes que cursan en esta universidad que para cualquier otro usuario interesado en el tema.

**SUNAM:** Sistema de la Universidad Nacional Autónoma de México. (Herrera, 2011)

Visualiza información a toda la comunidad universitaria sobre becas, estancias, aulas virtuales, calendarios, recursos educativos y demás servicios de interés, tanto para los estudiantes, como para los que desean ingresar a la UNAM. A su vez, ayuda a los diferentes departamentos existentes en la universidad a gestionar todos sus procesos, con sus respectivas informaciones sobre las reuniones y actividades orientadas en un tiempo determinado. El sistema está desarrollado con tecnologías web, HTML, JavaScript y CSS, también utiliza librerías de *JQuery*. Para una mejor visualización se requiere de la instalación de *Flash Player* y *Acrobat Reader*. El mismo tiene como deficiencia que sus páginas están muy cargadas de información, lo cual dificulta la orientación del usuario a la hora de realizar sus operaciones.

**SUNLP:** Sistema de la Universidad Nacional de La Plata (Tauber, 2008)

El SUNLP permite conocer, a todos los estudiantes y demás usuarios, todo lo relativo a la gestión del plan estratégico de la universidad, del plan de obras, así como la gestión presupuestaria para poder estudiar en esta escuela de nivel superior. Está desarrollado con los lenguajes del lado del cliente: HTML, JavaScript y CSS, además utiliza librerías de *JQuery*, permitiendo entre otras cosas, que el menú principal se despliegue solamente pasando el *mouse* por encima de este.

El sistema contribuye a una mayor orientación de las actividades que se llevan a cabo en los diferentes departamentos existentes. Posee una alta seguridad en sus datos, pues solo pueden acceder a estos los usuarios autorizados, lo cual es de vital importancia en los sistemas de gestión. Sin embargo, sus

funcionalidades no pueden ser utilizadas en la propuesta de solución, pues no están acorde a las pautas planteadas por el cliente de la misma.

Se analizaron además, sistemas nacionales desarrollados para este fin, buscando funcionalidades que pudieran ser utilizadas en el trabajo que se desarrolla, así como insuficiencias que imposibilitarán la utilización del mismo.

**SUCMH:** Sistema de la Universidad de Ciencias Médicas de La Habana (Pérez, 1999).

El SUCMH presenta en su portada una buena fundamentación de las tareas, informaciones y convocatorias de actividades que realiza la universidad. En su diseño gráfico, utiliza colores agradables a la vista del usuario. Trabaja de manera centralizada las informaciones de sus centros académicos y de sus eventos universitarios. El mismo se relaciona con diferentes sitios del mundo que trabajan la misma rama de las ciencias médicas. Proporciona información de los cursos de posgrado, de las especialidades, de la síntesis de los planes de estudios por carreras, entre otros. Está desarrollado con HTML, JavaScript y CSS, como lenguajes del lado del cliente. Aunque es un sistema adecuadamente diseñado y abarca un gran número de funcionalidades, se debe señalar que no contiene las que son necesarias gestionar en la propuesta de solución.

**SUCMM:** Sistema de la Universidad de Ciencias Médicas de Matanza (García, 2008).

Este sistema presenta una adecuada gama de colores en su portada. Los principales procesos que gestionan son los siguientes: desarrollo de talleres relacionados con la rama, los planes de trabajos mensuales, curiosidades, noticias y las informaciones referentes a las carreras. El mismo está desarrollado con CSS, HTML, JavaScript; y al igual que el sistema anterior, no contiene las funcionalidades que exige el cliente de la propuesta de solución.

**SUCMC:** Sistema de la Universidad de Ciencias Médicas de Cienfuegos (Baños, 2012).

Este sistema gestiona la información de las diferentes carreras que se estudian, convocatorias para números especiales de investigación pedagógica, eventos virtuales, salas virtuales y listas de correo, secciones de documentos y conferencias. Fue desarrollado con HTML, JavaScript y CSS, como lenguajes del lado del cliente, además, utiliza librerías de *JQuery* para mejorar su apariencia. Presenta como inconveniente, que no posee mucha información sobre los departamentos que lo componen, además, su interfaz de inicio está muy cargada de información, lo que dificulta a los usuarios a la hora de realizar cualquier búsqueda.

El estudio realizado demostró que los sistemas estudiados no cumplen con los requerimientos necesarios para solucionar los problemas existentes en el Departamento Metodológico Central de Ingeniería y Gestión de Software. En algunos casos, se toman funcionalidades de dichos sistemas, para ser utilizadas en la propuesta de solución, pero en general, los requisitos de diseño y procesos que se gestionan en los mismos, no se relacionan con los que se deben implementar en el sistema a desarrollar. En función de esto, se hace necesario buscar nuevas alternativas para dar una solución efectiva a la problemática planteada. A continuación, en el siguiente capítulo, se describirá la propuesta de solución para así satisfacer las necesidades y resolver los problemas existentes en dicho departamento.

### **1.5 Propuesta de solución**

A partir del análisis realizado en el epígrafe anterior, se hace necesario construir un sistema que solucione el problema planteado. Dicho sistema debe permitir la gestión de todos los procesos descritos con anterioridad, tales como: las evaluaciones de los profesores, las planificaciones de las diferentes preparaciones metodológicas que se realizarán en un periodo, la planificación de los controles a las clases, así como los planes de trabajos de cada uno de los asesores. Sus principales funcionalidades deben estar enmarcadas en el objetivo de contribuir a la mejora del trabajo de los asesores del departamento y a gestionar de una mejor forma la información que se maneja.

### **1.6 Tecnologías que se utilizarán en la propuesta de solución**

Cuando se trata de desarrollar un sistema informático que satisfaga las necesidades de los clientes, así como de sus usuarios finales, que cumpla con normas de calidad y funcione sobre máquinas reales, resulta muy importante contar con un proceso o metodología de desarrollo de software que oriente el camino a seguir. Es responsabilidad entonces de los desarrolladores escoger la que se considere más adecuada, para cumplir con los requisitos de tiempo y documentación, entre otros, que hayan sido establecidos como requerimientos del sistema (Gómez, 2006). Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio sobre las tecnologías y herramientas actuales posibles a emplear, descritas a continuación.

#### **1.6.1 Metodologías de desarrollo**

Por ser un equipo de trabajo pequeño y contar con poco tiempo para el desarrollo de la solución, se decide utilizar una metodología ágil. Estas permiten realizar entregas rápidas al cliente y así poder satisfacer sus necesidades. A continuación, se muestra el estudio realizado de varias de estas metodologías, para así llegar a la selección de una de ellas:



**XP (eXtreme Programming):** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Además, se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Letelier, 2007). Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados en una misma dirección, permitiendo además seguir de forma clara el avance de las tareas a realizar, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando el nivel de interés del equipo (Romero, 2008).

A continuación se describen las fases de esta metodología (Letelier, 2006):

- **Planificación del proyecto:** en esta fase se definen las historias de usuario con el cliente, ya que es el primer paso de cualquier proyecto que siga la metodología XP.
- **Diseño:** esta fase sugiere la obtención de diseños simples y sencillos, lo menos complicado posible, fácilmente entendible a la hora de su desarrollo.
- **Codificación:** la codificación debe hacerse ajustándose a estándares de codificación ya creados. Programar bajo estos estándares, mantiene el código consistente, facilita su comprensión y escalabilidad.
- **Pruebas:** uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando.

**SCRUM:** es una estrategia de gestión donde se aplican de manera regular un conjunto de prácticas para mejorar el trabajo colaborativo y obtener el mejor resultado posible en la gestión de un proyecto de software. Es una metodología de trabajo ágil diseñada para acortar el ciclo de desarrollo, conseguir una mejor aproximación entre las funcionalidades del software y los requerimientos del cliente. Presenta una mayor versatilidad frente a los cambios, manejo más eficiente de los requerimientos cambiantes en un proyecto y proporciona una mejor comunicación entre el cliente y el equipo desarrollador (Tello, 2003).

**MSF (Microsoft Solution Framework):** está enfocada a dirigir proyectos o soluciones de innovación. En ella no se detalla ni se hace énfasis en la organización, ni el tamaño del equipo de desarrollo, está más bien centrada en la gestión y administración del proyecto para lograr el impacto deseado. Involucra

indudablemente la calidad. Aún tiene fallos o desperfectos, para ello propone seleccionar un grupo de prueba piloto, el cual es una versión beta y cumplido un tiempo de prueba ya es liberada la versión formal o versión alfa, en la cual está garantizada la calidad (Fraile, 2006). Esta metodología cuenta con cinco fases: visión, planificación, desarrollo, estabilización y despliegue o implementación.

**Featured Driven Development (FDD):** la iniciativa de desarrollo dirigida por rasgos es un tipo de metodología ágil que rige su arquitectura por las funcionalidades del programa a implementar. No enfatiza los requerimientos y la generación de documentación previa, sino que se centra en realizar las fases de diseño y construcción. Sin embargo, se preocupa mucho de la calidad del producto, ya que insiste en el monitoreo constante (Gómez, 2006). Las fases de esta metodología son: realización de un modelo global, lista de funcionalidades, planificación por funcionalidad y diseño y construcción.

De las metodologías estudiadas y descritas anteriormente, el grupo de trabajo en conjunto con el cliente decide seleccionar para el desarrollo de la aplicación, prácticas de XP y SCRUM. Su característica de presentar una alta flexibilidad, ayudando a que el equipo trabaje unido y con los objetivos bien claros, además de poseer la ventaja de incluir al cliente en el equipo de desarrollo, hacen viable la selección de las mejores prácticas de estas metodologías. Se puede afirmar además, que permite eliminar los impedimentos en el desarrollo del producto, aumentando la satisfacción de los clientes, mediante la realización de entregas frecuentes de resultados tangibles e integrándolos activamente en el ciclo de desarrollo. Todo esto proporciona una mayor adaptación y adecuación a las necesidades del cliente.

### **1.6.2 Lenguaje de modelado**

El lenguaje UML (*Unified Modeling Language*) tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue. En fin, es un lenguaje de modelado que permite la representación conceptual y física de un sistema (Booch, 1999).

Se decide la utilización de este lenguaje, en su versión 2.0, para el análisis y diseño del sistema a desarrollar. De manera general es común la utilización del mismo, para construir artefactos y diagramas en proyectos realizados con las metodologías seleccionadas.

### **1.6.3 Lenguajes de programación web**

Cuando se va a utilizar un lenguaje de programación web, es necesario conocer con exactitud qué es lo que se quiere hacer y si el lenguaje permite o no su realización. Estos lenguajes de programación se

dividen en dos grupos fundamentales: los lenguajes del lado del cliente y los lenguajes del lado del servidor.

### **Lenguajes del lado del cliente**

**HTML 5** proporciona una plataforma con la que desarrollan aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Permite que los diseñadores de páginas web "rendericen" las imágenes, con lo cual se pueden crear imágenes dinámicas con las que el usuario puede interactuar. Las etiquetas de cabecera son de reciente introducción y se definen bajo las principales áreas y temáticas de un sitio web. Se puede decir que los motores de búsqueda van a mantener siempre en cuenta los enlaces para la navegación, para lo cual se utilizan las etiquetas creadas. HTML 5 viene con una serie de características avanzadas, para hacer más fácil, al desarrollador o diseñador, el uso de este lenguaje (Andyberna, 2011).

**JavaScript** es un lenguaje de programación interpretado por el cliente (no compilado). Está basado en objetos, pero no es un lenguaje de programación orientada a objetos. Basado en prototipos, imperativo, débilmente tipado y dinámico, pues responde a eventos en tiempo real. Es simple, maneja objetos dentro de la página web y sobre ese objeto se puede definir diferentes eventos (Torre, 2006).

**CSS** (*Cascading Style Sheets/Hoja de estilo cascada*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (*World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Presenta un control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Además, los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad (Cortez, 2012).

Como lenguajes del lado del cliente se utilizarán precisamente, la combinación HTML, JavaScript y CSS que es la más utilizada en la actualidad para desarrollar el contenido referente a la parte del cliente en las aplicaciones web, pues los tres se complementan de forma ideal para brindar así una amplia gama de posibilidades a los desarrolladores.

### **Lenguajes del lado del servidor**

**Perl** es un lenguaje de programación interpretado, donde el código de los *scripts* no se compila, sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.

Además, es extensible a partir de otros lenguajes, ya que desde Perl se hacen llamadas a subprogramas escritos en otros lenguajes (Torre, 2006).

**ASP** (*Active Server Pages*) es la tecnología desarrollada por *Microsoft* para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje *Visual Basic Script* o JavaScript de *Microsoft* (Torre, 2006).

**PHP** (*Hypertext Pre-processor*) es un lenguaje interpretado de propósito general, ampliamente usado. Está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. PHP es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación (Achour, 1997). Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas. Entre ellas se pueden mencionar: (Achour, 1997).

- ✓ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, SybasemSQL, Informix, entre otras.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas del web de fácil programación.

De los lenguajes estudiados, del lado del servidor, se selecciona PHP para el desarrollo del sistema, por ser un lenguaje multicapa, por presentar una rapidez de ejecución formidable, además de tener una gran seguridad y poca probabilidad de corromper los datos.

#### **1.6.4 Marco de trabajo a utilizar**

Se decide utilizar un *framework* porque simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (Beneyto, 2012).

Un *framework* es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta (Gutiérrez, 2009).

**Symfony** es un marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web, además de que proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Está desarrollado completamente con PHP 5 y ha sido probado en numerosos proyectos reales. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas *Windows*. Symfony automatiza la mayoría de los elementos comunes de los proyectos web. Su capa de internacionalización, permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos. Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos. Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos. La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario (Zaninotto, 2010).

**CodeIgniter** presenta una gran versatilidad, pues es capaz de trabajar en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde solo se tiene un acceso por FTP para enviar los archivos al servidor y donde no hay acceso a su configuración. Es compatible con PHP, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. CodeIgniter es bastante menos rígido que otros *frameworks*. Define una manera de trabajar específica, pero en muchos de los casos se puede o no seguir y sus reglas de codificación muchas veces se pueden saltar para trabajar como más a gusto se encuentre. Algunos módulos como el uso de plantillas, son totalmente opcionales. El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, solamente cuando se van a utilizar realmente. No tiene una línea de comandos, para brindarle al desarrollador posibilidades con su trabajo (Álvarez, 2011).

En la presente investigación, se decidió utilizar como marco de trabajo Symfony porque además de ser muy fácil su aprendizaje, es muy flexible con la introducción de *bundles* (Beneyto, 2011), es muy completo y rápido. Además, posee un generador de plantillas muy amplio, el cual se puede integrar a muchos IDEs; al igual que una línea de comandos impresionante con muchísimas posibilidades.

### 1.6.5 Herramienta de modelado

Visual Paradigm es una herramienta de modelado que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar diagramas de procesos de negocio, ingeniería inversa (de código a modelo), generación de código (de modelo a código), generación de bases de datos (transformación de Diagramas Entidad-Relación a tablas de bases de datos), ingeniería inversa de bases de datos (a partir del modelo físico de la base de datos obtener su Diagrama Entidad-Relación), generación de informes de documentación, entre otros. Existen licencias gratuitas y comerciales de este producto (Zayas, 2001).

Para el modelado de los diferentes diagramas del presente trabajo, se utilizará como herramienta Visual Paradigm.

### 1.6.6 Sistemas gestores de bases de datos

A continuación se muestra el estudio realizado sobre varios gestores de base de datos, para así seleccionar el que se va a utilizar en el desarrollo de la aplicación.

**PostgreSQL** es un sistema gestor de base datos que corre en casi todos los sistemas operativos, como es el caso de Linux, Unix, BSDs, Mac OS, *Beos*, *Windows*. Presenta una documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Altamente adaptable a las necesidades del cliente. Posee soporte nativo para los lenguajes más populares del medio como PHP, C, C++, *Perl*, *Python*; además de todas las características de una base de datos profesional. Tiene extensiones para alta disponibilidad, nuevos tipos de índices, minería de datos. Además, presenta utilidades para limpieza de la base de datos y utilidades para análisis y optimización de *Querys* (Quiñones, 2010).

**Microsoft SQL Server** es un sistema para la gestión de bases de datos creado por *Microsoft*, el mismo se basa en el modelo relacional. Este gestor utiliza como lenguajes de consulta T-SQL y ANSI SQL. Reúne en un solo producto la potencia necesaria para cualquier aplicación empresarial. Entre sus principales características se pueden mencionar que soporta transacciones, procedimientos almacenados, proporciona escalabilidad, estabilidad y seguridad, permitiendo trabajar en modo cliente-servidor: Incluye un potente entorno gráfico de administración (Meneses, 2006).

**Oracle** es una herramienta de administración gráfica que es intuitiva y cómoda de utilizar. Ayuda a analizar datos y efectuar recomendaciones concernientes a mejorar el rendimiento y la eficiencia en el manejo de aquellos datos que se encuentran almacenados. Apoya en el diseño y optimización de modelos de datos. Apoya además, en la definición de estándares de diseño y nomenclatura de objetos. Permite

documentar y mantener un registro periódico de las mantenciones, actualizaciones de hardware y software, cambios en las aplicaciones y, en general, todos aquellos eventos relacionados con cambios en el entorno de utilización de una base de datos (Cerón, 2010).

De los sistemas gestores de base de datos descritos con anterioridad, se decide escoger para el trabajo con la base de datos del sistema que se desea implementar, PostgreSQL, puesto que es altamente adaptable a las necesidades del cliente. Además, sus requerimientos de administración y mantenimiento son relativamente bajos respecto al resto de bases de datos comerciales. Es muy aceptado por toda la comunidad y puede ser ejecutado en los principales sistemas operativos. Su código es 100 % libre y es muy fácil su aprendizaje.

### 1.6.7 Entorno de desarrollo integrado

**NetBeans** es un entorno de desarrollo integrado (IDE) para programadores, pensado para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Expande el soporte de los lenguajes dinámicos con apoyo para PHP 5.3, así como el esquema de Symfony, y acelera el desarrollo de aplicaciones web con PHP. Es un IDE multilenguaje completo y modular, una plataforma para construir aplicaciones completas para el cliente (Ruiz, 2008).

**Eclipse** es un IDE que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. Proporciona solamente el entorno de desarrollo, siendo necesario además de otras herramientas para poder compilar y ejecutar las aplicaciones desarrolladas (Martínez, 2007).

La arquitectura de *plugins* de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo, tales como: herramientas UML, editores visuales de interfaces y ayuda en línea para librerías.

Después de haber caracterizado los IDE´s anteriores se decide utilizar, para el desarrollo de la aplicación, **NetBeans**, por presentar facilidad de uso. Solamente se debe insertar un código php dentro del código HTML de un sitio web, de esta manera, cuando un cliente visita la página web que contiene este código, el servidor lo ejecuta y el cliente solo recibe el resultado. Además, ofrece soporte a un número considerable de sistemas gestores de bases de datos. Permite que las aplicaciones sean desarrolladas a partir de un

conjunto de componentes de software denominados módulos, que pueden ser desarrollados de forma independiente y así extender las aplicaciones.

### **1.6.8 Servidor Web Apache**

Apache está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando, qué módulos se van a cargar, ya sea al compilar o al ejecutar el mismo. Corre en multitud de sistemas operativos, esto lo hace prácticamente universal. Apache es una tecnología gratuita de código de fuente abierta (Booch, 1999).

Apache es en la actualidad el principal servidor web. Es el más rápido, eficiente y el que evoluciona a mayor velocidad. Por su naturaleza de software abierto, es ideal para instalar en máquinas GNU/Linux, que aseguran un sistema operativo con unas comunicaciones excelentes. Para la propuesta de solución se decide utilizar este servidor.

## **1.7 Conclusiones parciales**

En el capítulo se realizó un análisis de los principales conceptos asociados al tema, además de un estudio de los sistemas homólogos existentes en el mundo y en el país. Finalmente, se llegó a la conclusión de que los mismos no reúnen las condiciones necesarias para garantizar la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software, por lo que se decidió desarrollar un nuevo sistema que solucionará la problemática identificada. Como se fue presentando durante el desarrollo del capítulo, este sistema será desarrollado con el uso del marco de trabajo Symfony en su versión 2.1, como gestor de base de datos PostgreSQL en su versión 9.1, como lenguaje del lado del cliente la combinación de HTML5, JavaScript 1.5 y CSS 3, como lenguaje del lado del servidor PHP 5.3, como entorno de desarrollo integrado *NetBeans* 7.2 y como servidor web Apache en su versión 2.2. Para el modelado se definió el uso del lenguaje UML 2.0, apoyándose en la herramienta Visual Paradigm, en su versión 8.0 y como metodología de desarrollo, prácticas de XP y SCRUM.



## Capítulo 2: “Características del sistema. Elaboración y Planificación”

### 2.1 Introducción

El presente capítulo tiene como objetivo presentar los resultados de las fases de exploración y planificación de la propuesta de solución. Se especifican además, los requisitos funcionales y no funcionales que el sistema debe cumplir para satisfacer las necesidades del cliente, así como los diferentes artefactos que establece la metodología de desarrollo de software utilizada.

### 2.2 Flujo actual de los procesos

Actualmente la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software, se realiza de forma manual, lo cual implica que en ocasiones exista demora en obtener alguna información referente al trabajo que se realiza en el departamento. En algunos casos, son los mismos profesores los encargados de guardar dicha información, trayendo consigo que se duplique y a la vez se cree un volumen extenso de esta. Los procesos fundamentales que se desarrollan están relacionados a: la planificación de los controles a clases, planificación de preparaciones metodológicas y la confección y evaluación de los planes de trabajo de los asesores.

A continuación se muestra en la tabla 1, la ficha del proceso “Gestión de los controles a clases”, el resto se pueden ver en el **Anexo\_1**.

Tabla 1: Ficha de proceso “Gestión de los controles a clases”

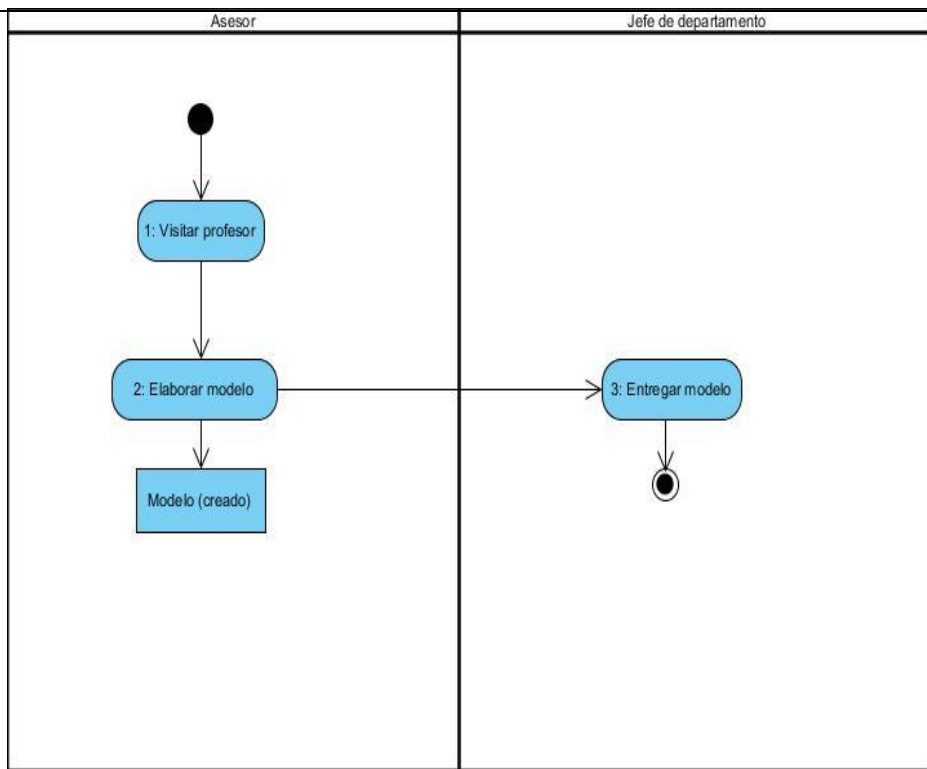
Ficha de procesos			
<b>Identificador:</b> 1	<b>Nombre del proceso:</b> Gestión de los controles a clases.	<b>Tipo:</b> Principal	<b>Responsable:</b> Jefe de Departamento
<b>Misión:</b> La gestión de los controles a clases tiene como principal objetivo controlar el cumplimiento de las orientaciones realizadas durante los ciclos de trabajo metodológico, lo cual se debe desencadenar en el desarrollo de clases de calidad.			
<b>Precondición:</b>			
<b>Síntesis del proceso:</b> Inicialmente el jefe del departamento realiza la planificación de los controles a clases, es decir, le asigna a los asesores del departamento varios controles. Luego se da una evaluación del control efectuado y se guarda dicha información para poder consultarla en otro momento.			

A continuación, en la tabla 2 se representa el ciclo del proceso “Gestión de los controles a clases”, el resto puede verse en el **Anexo\_2**.

**Tabla 2: Flujo de actividades del proceso “Gestión de los controles a clases”**

Actividades	Representación	
<p>1- El jefe del departamento es el encargado de realizar la planificación de los controles a clases. Luego ejecuta dicha planificación y la refleja en un modelo.</p>		
<p>2- Para realizar la planificación el jefe del departamento revisa el horario, para luego definir a qué profesor efectuarle el control a clases. Luego asigna el asesor encargado de realizar dicho control. Después actualiza la planificación y la envía a los subordinados.</p>		

3- Para la ejecución de la planificación el asesor seleccionado visita al profesor, reflejando en el modelo de control a clases, los resultados del mismo. Finalmente, el modelo se entrega al jefe de departamento.



### 2.3 Objeto de automatización

Los procesos descritos anteriormente, deben ser automatizados, pues la forma en que se están gestionando, no es la más óptima. El objetivo de este trabajo es mejorar la gestión de estos procesos, facilitándoles la tarea a los encargados de realizar las diferentes actividades. Se trabajará directamente en la gestión de los controles a clases, las preparaciones metodológicas, y los planes de trabajo de los asesores.

### 2.4 Requerimientos del sistema

Para el desarrollo de una aplicación es muy importante tener claro cuáles van a ser las principales funcionalidades y restricciones del mismo. Esto se puede obtener con la definición de los diferentes requisitos que debe cumplir el sistema, ya que constituyen la descripción de las necesidades del producto. A continuación se muestra la lista de requisitos funcionales y no funcionales del sistema, en la tabla 3:

**Tabla 3: Lista de requisitos del sistema**

Código	Descripción	Prioridad
RF (Requisitos funcionales)		
RF 1	Crear la planificación de controles a clases por asesores.	Alta

RF 2	Modificar la planificación de controles a clases por asesores.	Alta
RF 3	Adicionar un control a clase a un asesor.	Alta
RF 4	Modificar un control a clase a un asesor.	Alta
RF 5	Eliminar un control a clase a un asesor.	Alta
RF 6	Eliminar la planificación de controles a clases (al final del semestre).	Alta
RF 7	Mostrar la planificación de los controles a clases	Medio
RF 8	Dar seguimiento a los controles a clases.	Alta
RF 9	Generar reporte total de las evaluaciones de controles a clases por asignaturas.	Medio
RF 10	Generar reporte total de las evaluaciones de controles a clases.	Medio
RF 11	Generar reporte del estado actual de la planificación de controles a clases por asignaturas.	Medio
RF 12	Generar reporte del estado actual de la planificación de controles a clases.	Medio
RF 13	Adicionar nomenclador	Alta
RF 14	Modificar nomenclador	Alta
RF 15	Eliminar nomenclador	Alta
RF 16	Crear la planificación de las preparaciones metodológicas	Alta
RF 17	Modificar la planificación de las preparaciones metodológicas	Alta
RF 18	Eliminar preparaciones metodológicas	Alta
RF 19	Mostrar las planificaciones metodológicas	Media
RF 20	Dar seguimiento a las preparaciones metodológicas	Alta
RF 21	Generar acta de la preparación metodológica	Alta
RF 22	Imprimir acta de la preparación metodológicas	Media
RF 23	Modificar acta de las preparaciones metodológicas	Media
RF 24	Eliminar acta de las preparaciones metodológicas	Media
RF 25	Planificar visitas a las preparaciones metodológicas	Alta
RF 26	Visualizar la planificación de las visitas a las preparaciones metodológicas.	Media
RF 27	Modificar la planificación de las visitas	Alta
RF 28	Eliminar visitas a las preparaciones metodológicas	Alta
RF 29	Emitir valoración de la preparación metodológica	Alta
RF 30	Crear plan de trabajo de los asesores.	Alta
RF 31	Mostrar plan de trabajo de los asesores.	Alta
RF 32	Modificar plan de trabajo de los asesores.	Alta
RF 33	Crear evaluación del profesor	Alta

RF 34	Modificar evaluación del profesor	Alta
RF 35	Eliminar evaluación del profesor	Alta
RF 36	Mostrar las evaluaciones de los profesores	Media
RF 37	Generar reportes de las evaluaciones de los profesores	Media
RF 38	Generar corte de resumen de evaluaciones de los profesores	Media
RF 39	Adicionar usuario.	Alta
RF 40	Modificar usuario.	Alta
RF 41	Eliminar usuario.	Alta
RF 42	Autenticar usuario.	Alta
RNF ( Requisitos no funcionales)		
<b>RNF 1</b>	<b>Apariencia o interfaz externa.</b>	
RNF1.1	Se muestran todos los módulos por sus nombres.	
RNF1.2	La interfaz es sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo.	
RNF1.3	El sistema debe presentar una serie de menú tanto laterales como en barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.	
<b>RNF 2</b>	<b>Seguridad</b>	
RNF 2.1	La seguridad de la base de datos está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.	
<b>RNF 3</b>	<b>Portabilidad</b>	
RNF 3.1	El sistema podrá ser ejecutado bajo los distintos sistemas operativos, tanto en los privativos como en los de software libre, en sus distintas versiones.	
<b>RNF 4</b>	<b>Restricciones de diseño</b>	
RNF4.1	Servidor de base de datos con PostgreSQL 9.1 o superior.	
RNF4.2	Servidor de aplicaciones Web: Apache 2.2 o superior.	
RNF4.3	Navegador web: <i>Internet Explorer</i> o <i>Mozilla Firefox</i>	
<b>RNF 5</b>	<b>Interfaces Hardware</b>	
RNF 5.1	La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.	
<b>RNF 6</b>	<b>Requisitos Legales, de Derecho de Autor y otros</b>	
RNF6.1	El sistema debe ser sometido a un análisis legal por parte de los abogados y personal autorizado con vistas a declarar su autenticidad y evitar restricciones legales para su uso y comercialización; así mismo se debe proceder a una evaluación y certificación por parte del cliente del producto.	

## 2.5 Propuesta de la aplicación

Se propone la implementación de un sistema que brinde las funcionalidades identificadas y esté acorde al flujo de trabajo actual. Como se explicó anteriormente, el mismo gestionará la planificación de controles a clases, de las preparaciones metodológicas y la confección y seguimiento de los planes de trabajo de los asesores.

## 2.5.1 Identificación de los roles

Interactuarán con el sistema varios roles, que se le llaman usuarios, los cuales son el jefe del departamento y los asesores de este. A continuación, en la tabla 4 se muestra la relación de los roles existentes en el sistema:

Tabla 4: Relación de roles del sistema

Rol	Descripción
Jefe de departamento (administrador)	Es el rol encargado de realizar las funciones administrativas, ya sea administrar usuarios, publicar las diferentes actividades, velar por la seguridad de los datos, así como crear la planificación de controles a clases y las preparaciones metodológicas.
Asesor del departamento (usuario)	Es el encargado de publicar las informaciones relacionadas con los resultados de los controles, de las preparaciones metodológicas y la elaboración del plan de trabajo.

## 2.6 Fase de Exploración

En esta fase se enmarca el alcance del producto, por lo que el cliente define sus necesidades a través de las historias de usuarios (HU), por las cuales se estima el tiempo que llevará el desarrollo del mismo. Aunque estas estimaciones de tiempo no son estáticas y pueden variar cuando se analicen más, en cada iteración.

### 2.6.1 Historias de usuarios

Como se había dicho anteriormente, las HU son escritas por el cliente, describiendo con sus propias palabras lo que debe hacer el sistema. Las mismas presentan los detalles, para que los programadores puedan realizar la estimación del tiempo que llevará el desarrollo. Los desarrolladores dialogarán de forma directa con los clientes cuando llegue la hora de implementación, para así obtener los detalles necesarios (Joskowicz, 2008).

A continuación, en la tabla 5 y 6, se muestran las historias de usuario de los requisitos funcionales “Crear la planificación de los controles a clases” y “Crear la planificación de las preparaciones metodológicas”, el resto de las historias de usuario se encuentran en el **Anexo\_3**. En la ilustración 1 y 2, se muestran los prototipos de interfaz de las historias de usuario mencionadas anteriormente, el resto de los prototipos se pueden ver en el **Anexo\_4**.

**Tabla 5: HU “Crear la planificación de los controles a clases”**

<b>Historia de usuario.</b>	
No.:1	Nombre: Crear la planificación de los controles a clases.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite crear la planificación de los controles a clases que se realizarán en el periodo establecido. El usuario selecciona en el menú principal “Controles a clases”, ya dentro del módulo escoge la opción de crear y procede a llenar los datos correspondientes a todos los controles que se desean efectuar.	
Observaciones: La planificación de los controles a clases solo se podrá eliminar cuando culmine el semestre.	



Ilustración 1: Prototipo de interfaz “Crear la planificación de los controles a clases”

Tabla 6: HU “Crear la planificación de las preparaciones metodológicas”

Historia de usuario.	
No.:16	Nombre: Crear la planificación de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que crear la planificación de las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, ya dentro del módulo escoge la opción “Crear la	



planificación”, llenando los datos necesarios para cada preparación.  
 Observaciones:



Ilustración 2: Prototipo de interfaz “Crear la planificación de las preparaciones metodológicas”

## 2.7 Fase de Planificación

En esta fase es donde el cliente establece la prioridad que llevará cada una de las HU. También se tomarán los acuerdos referidos a la primera entrega del producto y se establece un cronograma, todo esto es en trabajo conjunto con el cliente.

### 2.7.1 Estimación por historia de usuario

Se realizó la siguiente estimación de las diferentes HU, para el desarrollo de la aplicación propuesta. A continuación se muestra en la tabla 7:

Tabla 7: Estimación de esfuerzo por HU

Historias de usuario	Puntos de estimación
----------------------	----------------------

Crear la planificación de controles.	2
Modificar la planificación de controles a clases por asesores (adicionar, modificar y eliminar).	2
Adicionar un control a clase a un asesor.	2
Modificar un control a clase a un asesor.	2
Eliminar un control a clase a un asesor.	2
Eliminar la planificación de controles a clases (al final del semestre).	2
Mostrar la planificación de los controles a clases	2
Dar seguimiento a los controles a clases.	2
Generar reporte total de las evaluaciones de controles a clases por asignaturas.	2
Generar reporte total de las evaluaciones de controles a clases.	2
Generar reporte del estado actual de la planificación de controles a clases por asignaturas.	2
Generar reporte del estado actual de la planificación de controles a clases.	2
Adicionar nomenclador	3
Modificar nomenclador	3
Eliminar nomenclador	3
Crear la planificación de las preparaciones metodológicas	1
Modificar la planificación de las preparaciones metodológicas	1
Eliminar preparaciones metodológicas	1
Mostrar las planificaciones metodológicas	1
Dar seguimiento a las preparaciones metodológicas	1
Generar acta de la preparación metodológica	1
Imprimir actas de las preparaciones metodológicas	1
Modificar acta de las preparaciones metodológicas	1
Eliminar acta de las preparaciones metodológicas	1
Planificar visitas a las preparaciones metodológicas	1

Visualizar la planificación de las visitas a las preparaciones metodológicas.	1
Modificar la planificación de las visitas	1
Eliminar visitas a las preparaciones metodológicas	1
Emitir valoración de la preparación metodológica	1
Crear plan de trabajo de los asesores.	2
Mostrar plan de trabajo de los asesores	2
Modificar plan de trabajo de los asesores.	2
Generar reportes sobre los planes de trabajo.	2
Crear evaluación del profesor	2
Modificar evaluación del profesor	2
Eliminar evaluación del profesor	2
Mostrar las evaluaciones de los profesores	2
Generar reportes de las evaluaciones de los profesores	2
Generar corte de resumen de evaluaciones de los profesores	2
Adicionar usuario.	3
Modificar usuario.	3
Eliminar usuario.	3
Autenticar usuario.	3

Los puntos de estimación anteriores significan la prioridad de la implementación de cada uno de los requisitos funcionales, es decir, los que presentan un punto de estimación con valor 3 serían las primeras funcionalidades a implementar, luego los de valor 2 y por último los que tienen un valor 1.

### **2.7.2 Plan de iteraciones**

Con la descripción de las HU y la estimación del esfuerzo para la realización de las mismas, por parte de los desarrolladores, es necesaria una planificación de cómo se va a proceder la implementación de la aplicación. Se agrupan las HU por iteraciones, para así especificar, en cada iteración del proceso de implementación, cuáles serán desarrolladas en cada una de estas iteraciones. Para el desarrollo del sistema se deciden desarrollar tres iteraciones, las cuales serán descritas a continuación:

#### **Iteración# 1:**

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 13, 14, 15, 39, 40, 41 y 42. Estas HU seleccionadas para desarrollarse en esta primera iteración, permiten que se creen, modifiquen o eliminen en el sistema los nomencladores generales, tales como profesores, disciplinas, asignaturas, locales, grupos, facultades, departamentos, los cuales son de vital importancia para el trabajo en el departamento. En estas HU se permitirá el control de acceso en el sistema y a su vez la gestión de los usuarios. Al concluir esta iteración se llevará a cabo la primera entrega del sistema, con el objetivo de mostrarle al cliente lo realizado hasta el momento y recibir además una retroalimentación del mismo.

**Iteración# 2:**

Esta iteración tiene como objetivo implementar las historias de usuario 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 30, 31, 32, 33, 34, 35, 36, 37 y 38. Con el desarrollo de las HU antes mencionadas, se da cumplimiento a un gran grupo de funcionalidades que requiere el sistema. Se muestran a los asesores del departamento los controles a clases que se llevarán a cabo durante un periodo establecido, así como todo lo relacionado con el plan de trabajo de los mismos. La entrega efectuada, luego de haber terminado esta iteración, servirá en conjunto con la anterior, para verificar con el cliente el cumplimiento de lo pactado.

**Iteración# 3:**

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 y 29. Estas están estrechamente relacionadas con las preparaciones metodológicas, las cuales son muy importantes para contribuir a la preparación de los diferentes profesores. Al finalizar esta iteración, ya se tiene lista la primera versión del producto, para mostrárselo al cliente.

**2.7.3 Plan de duración de iteraciones**

Con la utilización de las prácticas de las metodologías seleccionadas como parte de su ciclo de vida, es necesaria la creación de un plan de duración por iteraciones establecidas, con el objetivo de mostrar la duración de cada una de las iteraciones, además del orden en que serán implementadas las HU. A continuación, en la tabla 8 se muestra el plan de duración de las iteraciones:

**Tabla 8: Plan de duración de las iteraciones**

Iteración	Historias de usuarios	Duración de la iteración
<b>#1</b>	Adicionar nomenclador	4 semanas
	Modificar nomenclador	
	Eliminar nomenclador	
	Adicionar usuario.	
	Eliminar usuario.	

	Modificar usuario.	
	Autenticar usuario.	
<b>#2</b>	Crear plan de trabajo de los asesores.	4 semanas
	Mostrar la planificación de los planes de trabajo.	
	Modificar plan de trabajo de los asesores.	
	Crear evaluación del profesor	
	Modificar evaluación del profesor	
	Eliminar evaluación del profesor	
	Mostrar las evaluaciones de los profesores	
	Generar reportes de las evaluaciones de los profesores	
	Generar corte de resumen de evaluaciones de los profesores	
	Crear la planificación de controles a clases por asesores.	
	Modificar la planificación de controles a clases por asesores (adicionar, modificar y eliminar).	
	Adicionar un control a clase a un asesor.	
	Modificar un control a clase a un asesor.	
	Eliminar un control a clase a un asesor.	
	Eliminar la planificación de controles a clases (al final del semestre).	
	Mostrar la planificación de los controles a clases	
	Dar seguimiento a los controles a clases.	
	Generar reporte total de las evaluaciones de controles a clases por asignaturas.	
	Generar reporte total de las evaluaciones de controles a clases.	
	Generar reporte del estado actual de la planificación de controles a clases por asignaturas.	
	Generar reporte del estado actual de la planificación de controles a clases.	
<b>#3</b>	Crear la planificación de las preparaciones metodológicas	4 semanas
	Modificar la planificación de las preparaciones metodológicas	
	Eliminar preparaciones metodológicas	
	Mostrar las planificaciones metodológicas	
	Dar seguimiento a las preparaciones metodológicas	
	Generar acta de la preparación metodológica	
	Imprimir el acta de las preparaciones metodológicas.	
	Modificar acta de las preparaciones metodológicas	
	Eliminar acta de las preparaciones metodológicas	

	Planificar visitas a las preparaciones metodológicas	
	Visualizar la planificación de las visitas a las preparaciones metodológicas.	
	Modificar la planificación de las visitas	
	Eliminar visitas a las preparaciones metodológicas	
	Emitir valoración de la preparación metodológica	

## 2.7.4 Plan de entrega

Para un mejor entendimiento de la estructura del sistema, el mismo está conformado por 5 módulos, a continuación en la tabla 9 se hace referencia a las funcionalidades por cada módulo:

**Tabla 9: HU abarcadas en módulos**

<b>Módulo</b>	<b>Historia de usuario</b>
Administración.	Adicionar nomenclador
	Modificar nomenclador
	Eliminar nomenclador
	Adicionar usuario.
	Eliminar usuario.
	Modificar usuario.
	Autenticar usuario.
Plan de trabajo.	Crear plan de trabajo de los asesores.
	Mostrar la planificación de los planes de trabajo.
	Modificar plan de trabajo de los asesores.
	Crear evaluación del profesor
	Modificar evaluación del profesor
	Eliminar evaluación del profesor
	Mostrar las evaluaciones de los profesores
	Generar reportes de las evaluaciones de los profesores
	Generar corte de resumen de evaluaciones de los profesores
Controles a clases.	Crear la planificación de controles a clases por asesores.
	Modificar la planificación de controles a clases por asesores (adicionar, modificar y eliminar).
	Adicionar un control a clase a un asesor.
	Modificar un control a clase a un asesor.
	Eliminar un control a clase a un asesor.
	Eliminar la planificación de controles a clases (al final del semestre).
	Mostrar la planificación de los controles a clases
	Dar seguimiento a los controles a clases.
	Generar reporte total de las evaluaciones de controles

	a clases por asignaturas.
	Generar reporte total de las evaluaciones de controles a clases.
	Generar reporte del estado actual de la planificación de controles a clases por asignaturas.
	Generar reporte del estado actual de la planificación de controles a clases.
Preparaciones Metodológicas	Crear la planificación de las preparaciones metodológicas
	Modificar la planificación de las preparaciones metodológicas
	Eliminar preparaciones metodológicas
	Mostrar las planificaciones metodológicas
	Dar seguimiento a las preparaciones metodológicas
	Generar acta de la preparación metodológica
	Imprimir el acta de las preparaciones metodológicas.
	Modificar acta de las preparaciones metodológicas
	Eliminar acta de las preparaciones metodológicas
	Planificar visitas a las preparaciones metodológicas
	Visualizar la planificación de las visitas a las preparaciones metodológicas.
	Modificar la planificación de las visitas
	Eliminar visitas a las preparaciones metodológicas
	Emitir valoración de la preparación metodológica

Teniendo en cuenta la tabla 9 se hace necesaria la especificación del plan de entrega de las diferentes versiones del sistema. Las mismas incorporan los diferentes módulos de la aplicación. A continuación, en la tabla 10 se muestran las fechas de entrega, de las versiones antes mencionadas:

**Tabla 10: Plan de duración de la entrega del sistema**

<b>Módulo</b>	<b>Final 1ra iteración 1ra semana de abril</b>	<b>Final 2da iteración 3ra semana de abril</b>	<b>Final 3ra iteración 1ra semana de mayo</b>
Administración.	0.1	0.2	1.0
Plan de trabajo.		0.1	1.0
Controles a clases.		0.1	1.0
Evaluaciones.		0.1	1.0
Preparaciones Metodológicas.			1.0

El plan de duración de entrega cuando tiene el valor 0.1 significa que ya ha comenzado la implementación en la iteración, cuando tiene el valor 0.2 es que aún no se ha terminado con la implementación de las funcionalidades de dicha iteración y con 1.0 es que ya culminó la implementación de las funcionalidades en la iteración determinada.

## **2.8 Conclusiones parciales**

En este capítulo se comenzó a profundizar en el desarrollo de la propuesta de solución, obteniéndose una lista de las funcionalidades que debe tener el sistema. Fueron descritas todas las acciones que realizarán los usuarios en el sistema a desarrollar. Se presentaron además, las historias de usuario, la planificación de las iteraciones y el plan de duración de las entregas del sistema. La información obtenida en este capítulo constituye la base para continuar el diseño y construcción de la propuesta de solución.



## Capítulo 3: “Implementación y Pruebas”

### 3.1 Introducción

En este capítulo se realiza el diseño del sistema. Como resultado del mismo se obtienen las tarjetas CRC, se definen los patrones del diseño, la arquitectura del sistema y se diseñan los casos de prueba para verificar las funcionalidades de la aplicación.

### 3.2 Diseño del sistema

El patrón arquitectónico utilizado para el desarrollo del sistema es el Modelo Vista Controlador (MVC), pues se trabaja con Symfony 2, el cual brinda la utilización de este patrón por defecto, reduciendo el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Entre sus características principales se puede mencionar que el modelo, las vistas y los controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el modelo se refleje automáticamente en cada una de las vista (Mariñán, 2002). A continuación se muestran algunas de las ventajas que brinda la utilización del patrón antes mencionado (Mariñán, 2002):

- ✓ Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- ✓ La conexión entre el **modelo** y sus **vistas** es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- ✓ Es posible tener diferentes vistas para un mismo modelo (ejemplo: representación de un conjunto de datos como una tabla o como un diagrama de barras).
- ✓ Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
- ✓ Proporciona un mecanismo de configuración a componentes complejos más tratables que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).

La ilustración 3 muestra la representación del patrón arquitectónico MVC, este se basa en que un usuario le hace una petición a una vista del sistema, la cual envía la petición a la clase controladora y esta es la encargada de buscar la respuesta en la base de datos y se la muestra al usuario.

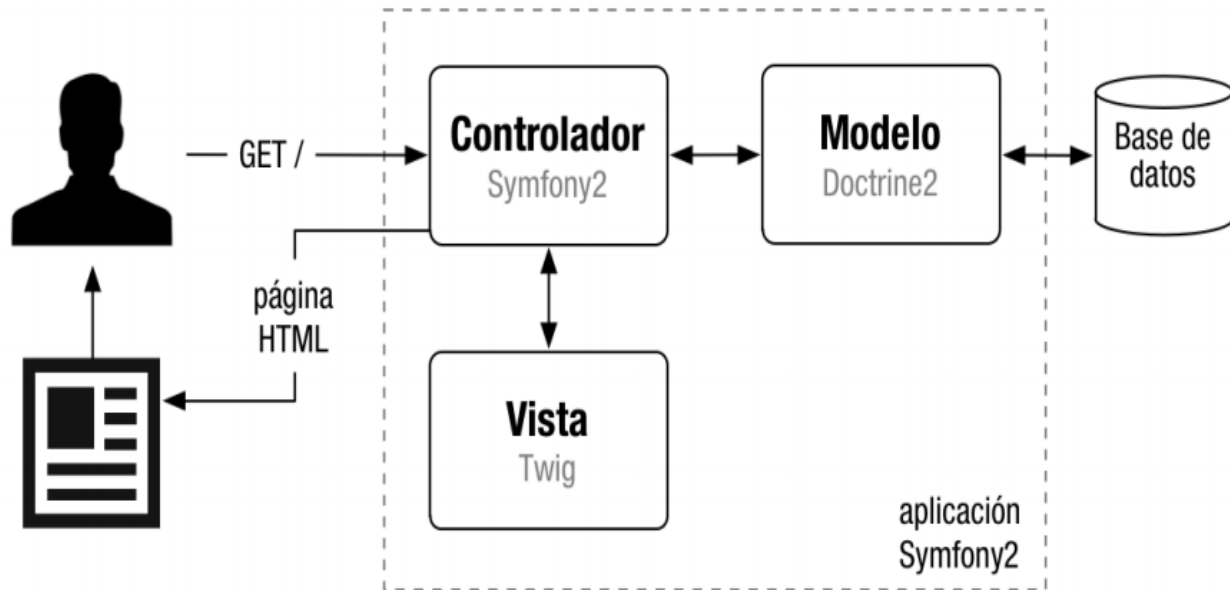


Ilustración 3: Modelo Vista Controlador (Potencier, 2011)

### Descripción de los elementos del patrón:

En el modelo se encuentran las clases, que son generadas de forma automática según la estructura de la base de datos. En Symfony, el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos (Potencier, 2011). Aquí es donde se guarda todo el código referente a la manipulación de los datos. Como ejemplo de estas clases se tiene: `PlanTrabajoMensual ()`, `ControlClase ()`, `PreparacionesMetodologicas ()`, entre otras.

La vista es la encargada de originar las páginas que son mostradas como resultado de las acciones, donde se encuentra el *layout*, que es común para todas las páginas de la aplicación. La vista en Symfony está conformada por varias partes, preparadas cada una de ellas especialmente para ser fácilmente modificada por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones (Potencier, 2011). En las vistas se guarda todo el código de presentación a los usuarios.

En el controlador se encuentran las acciones, las cuales son el núcleo de la aplicación, pues contienen toda la lógica de esta. Estas acciones utilizan el modelo y precisan las variables para la vista. Al realizarse una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición (Potencier, 2011). Las clases controladoras son las encargadas de la lógica de procesamiento de las

peticiones de los usuarios al sistema. Como ejemplo se puede mencionar la clase *PlanTrabajoMensualController* (), entre otras.

### 3.2.1 Tarjetas CRC (Clase-Responsabilidad-Colaboración).

Las tarjetas CRC son de gran ayuda para el usuario que no tiene entendimiento concreto del sistema, ya que estas muestran las funcionalidades que son de vital importancia en el diseño del mismo. A continuación se representan en las tablas 11, 12 y 13 las tarjetas CRC de los controles a clases, de las preparaciones metodológicas y del plan de trabajo mensual de los asesores, el resto puede verse en el **Anexo 5**.

**Tabla 11: Tarjeta CRC de los controles a clases**

<b>ControlClase</b>	
<b>Funcionalidades</b>	<b>Colaboración</b>
crearControlClaseAction( )	public function crearProfesorAction(Request \$request)
nuevoControlClaseAction( )	public function crearGrupoAction (Request \$request)
listarControlClaseAction( )	public function crearAsignaturaAction(Request \$request)
mostrarControlClaseAction( )	public function crearLocalAction (Request \$request)
editarControlClaseAction( )	
actualizarControlClaseAction( )	
eliminarControlClaseAction( )	
evaluarControlClaseAction( )	
actualizarEvaluacionControlAction( )	
buscarPorAsesorCCAction( )	
buscarAsesorCCAction( )	
buscarPorAsignaturaCCAction( )	
buscarAsignaturaCCAction( )	
buscarPorFechaCCAction( )	
buscarFechaCCAction( )	

**Tabla 12: Tarjeta CRC de las preparaciones metodológicas**

<b>PreparacionesMetodologicas</b>	
<b>Funcionalidades</b>	<b>Colaboración</b>
crearPreparacionAction( )	public function crearDisciplinaAction(Request \$request)
nuevaPreparacionAction( )	public function crearAsignaturaAction(Request \$request)
listarPreparacionAction( )	public function crearLocalAction (Request \$request)
mostrarPreparacionAction( )	
editarPreparacionAction( )	
actualizarPreparacionAction( )	
eliminarPreparacionAction( )	

asignarVisitanteAction( )	
actualizarVisitanteAction( )	
darSeguimientoPMAction( )	
actualizarSeguimientoPMAction( )	
generarActaAction( )	
buscarPorFechaPMAction( )	
buscarFechaPMAction( )	
buscarPorAsignaturaPMAction( )	
buscarAsignaturaPMAction( )	
buscarPorDisciplinaPMAction( )	
buscarDisciplinaPMAction( )	

**Tabla 13: Tarjeta CRC del plan de trabajo mensual**

PlanTrabajoMensual	
Funcionalidades	Colaboración
crearPlanTrabajoMensualAction( )	public function crearProfesorAction(Request \$request)
nuevoPlanTrabajoMensualAction( )	
listarPlanTrabajoMensualAction( )	
mostrarPlanTrabajoMensualAction( )	
editarPlanTrabajoMensualAction( )	
eliminarPlanTrabajoMensualAction( )	
asignarTareasAction( )	
evaluarPlanTrabajoAction( )	
actualizarEvaluacionPlanTrabajoAction( )	
buscarPorFechaPTAction( )	
mostraBusquedaPorFechaPTAction( )	
buscarPorAsesorPTAction( )	
mostraBusquedaPorAsesorPTAction( )	

### 3.2.2 Modelo de datos

El correcto diseño de la base de datos de la aplicación web es muy importante a la hora de implementarla, ya que esta es quien lo soporta. El *framework* de desarrollo utilizado posee una interfaz para el mapeo de objetos a bases de datos, la cual mapea la lógica de objetos a la lógica relacional. Se logra el acceso a la base de datos en las aplicaciones desarrolladas mediante el uso de objetos. A continuación, se muestra en la tabla 14, un ejemplo de las transformaciones que ocurren.

**Tabla 14: Transformación de la base de datos**

Relacional	Orientado a objetos(OO)
Tabla	Clase
Fila	Objeto
Campo	Atributo

A continuación se muestra en la ilustración 4 el modelo de datos:

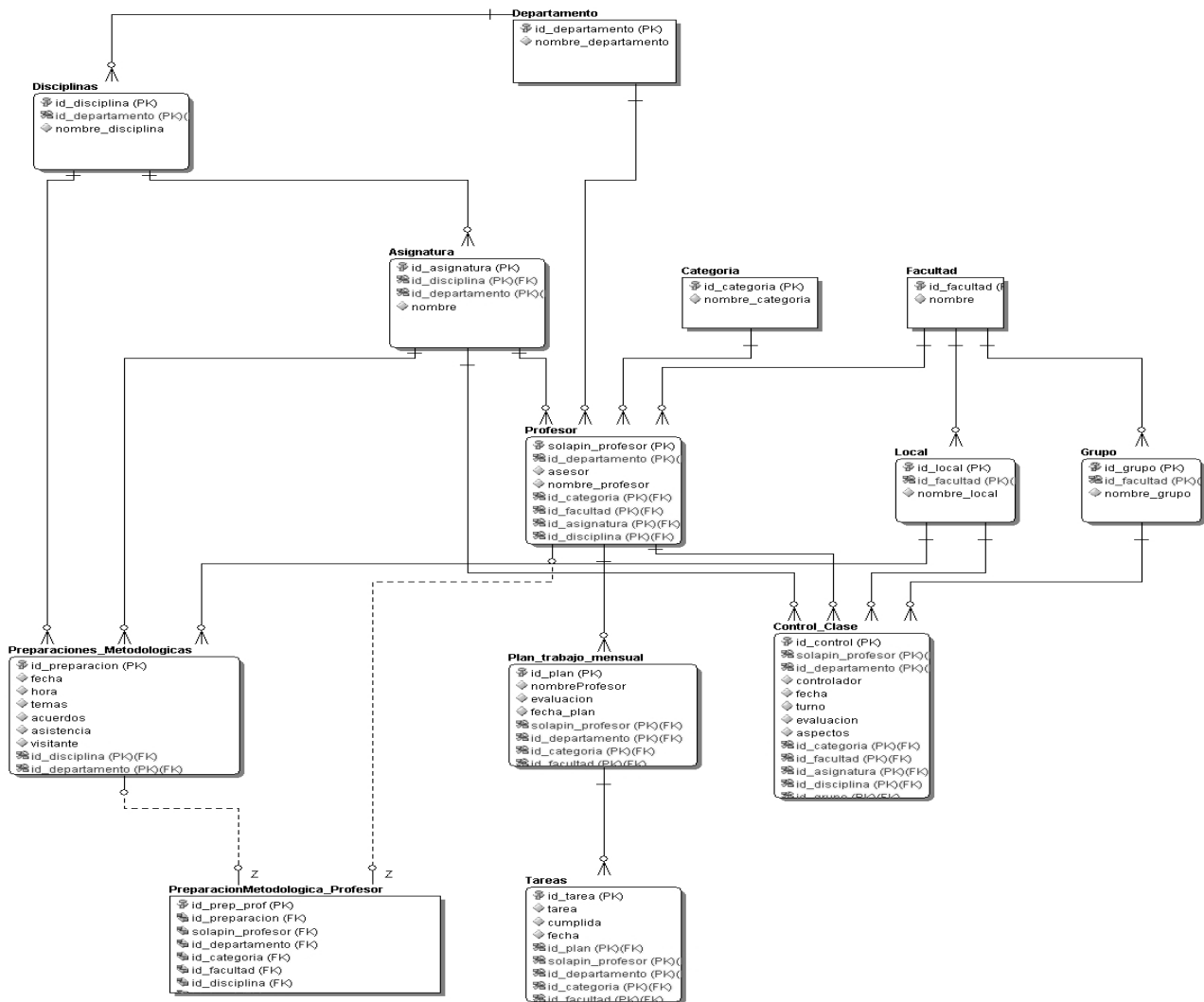


Ilustración 4: Modelo de datos

### 3.3 Patrones de diseño

Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse varias veces sin tener que hacer dos veces lo mismo (Gamma, 1995). Durante el desarrollo del sistema se utilizaron algunos de los patrones GRASP y GoF. A continuación se describen los mismos:

**Patrones GRASP:** (Patrones de software para la asignación general de responsabilidad)

**Experto:** es uno de los patrones que más se utiliza cuando se trabaja con Symfony, ya que es el encargado de asignar responsabilidades. Pues existen en el sistema clases que tienen la información necesaria para cumplir cualquier responsabilidad, tal es el caso de la clase Profesor (), que puede ser lo mismo un profesor que un asesor.

**Creador:** este patrón es el encargado de crear instancias entre las clases. Está reflejado en el código del sistema, ya que este presenta clase tales como ControlClase (), la cual contiene objetos de las clases Local (), Grupo () y Profesor ().

**Alta Cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Profesor (), la cual está formada por varias funcionalidades que están estrechamente relacionadas, y colabora con otras para realizar diferentes operaciones.

**Bajo Acoplamiento:** las clases Local () y Grupo () dependen en gran manera de la clase Facultad () para alcanzar un bajo acoplamiento. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

**Controlador:** todas las peticiones web son manipuladas por varios controladores, que son el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases PreparacionesMetodologicasController (), ControlClaseController () y PlanTrabajoController ().

**Patrones GoF (gang of four/banda de los cuatro):**

**Patrón Singleton:** el objeto *parameters* almacena una serie de valores tipo parámetros de configuración, los cuales son comunes para toda la aplicación. Se guardan en una base de datos y si son modificados por un administrador, quedan modificados para todos los usuarios que acceden a la página.

**Patrón Command:** este patrón se observa en la clase ControlClaseController (), esta clase se encarga de establecer el módulo y la acción que se va a usar según la petición del usuario, pues ya parseada la URL, se precisan los parámetros de la misma y de esta forma se puede conocer el *Action* que debe responder a la petición.

**Patrón Decorator:** este patrón se ve reflejado en el archivo nombrado *layout\_user*, el cual es el encargado de guardar el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada una de estas.

### 3.3.1 Modelo de despliegue

A continuación, se despliega la solución en la ilustración 5. El usuario hace una petición mediante el protocolo https al servidor web desde una PC cliente, este servidor realiza la búsqueda en la base de datos mediante el protocolo TCP-IP, mostrando la respuesta al usuario en la PC cliente. Se representa además el dispositivo impresora, la cual está conectada a la PC cliente.

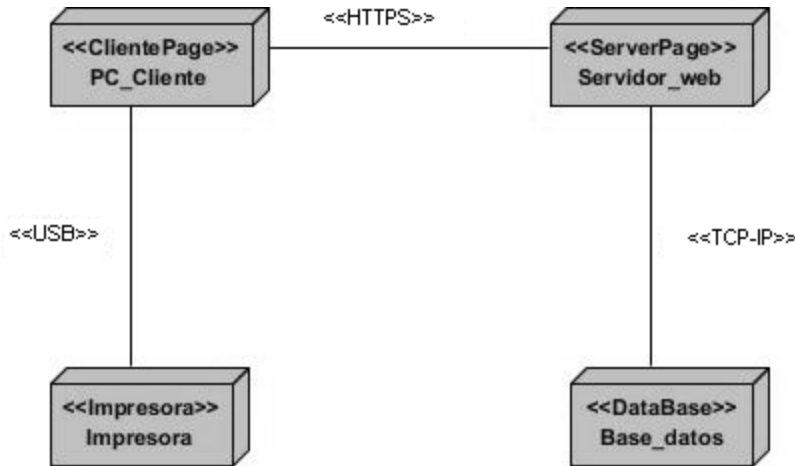


Ilustración 5: Modelo de despliegue

### 3.3.2 Estándar de codificación

Como estándar de codificación se utilizaron los convenios que se describen a continuación:

Todas las clases controladoras se capitalizan, es decir, comienzan con letra mayúsculas y terminan con la palabra *Controller*. En el caso de que las clases controladoras contengan más de una palabra, las iniciales de cada palabra comienzan con mayúsculas y el resto con minúscula, ejemplo de estas clases se tiene la siguiente: `ControlClaseController ()`.

Para nombrar las clases no se recomienda utilizar palabras reservadas del lenguaje, como es el caso de *namespace*, *user* entre otras.

Se definió además, que los formularios comiencen con letra mayúscula y terminen con la palabra “*Type*”, ejemplo de ello: `ControlClaseType ()`.

Todas las funciones comienzan con minúsculas y terminan con la palabra “*Action*”. En caso de que la función tenga más de dos palabras las demás comienzan con mayúsculas y terminaría con “*Action*”, esta descripción se define como “*camel case*”, ejemplo de este estándar se puede mencionar la función `listarControlClaseAction ()`.

### 3.4 Pruebas

Toda prueba de software desempeña un papel fundamental en el desarrollo de cualquier tipo de aplicación, pero si se estudia la mejor forma de hacerlo, siguiendo los pasos de acuerdo con los especialistas en el tema, se incrementan las posibilidades de que esta llegue a un feliz término y arroje resultados más cercanos a los esperados, permitiendo así, realizar a posterioridad un mejor análisis de la situación (Lovellette, 2002).

Existen diferentes niveles de pruebas, entre ellos se pueden mencionar la prueba de unidad y la prueba de sistema. La prueba de unidad se concreta en el esfuerzo de verificación de la unidad más pequeña del diseño del software: el componente o módulo de software. Tomando como guía la descripción del diseño al nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que éstas descubren. Las pruebas de unidad se concretan en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente (Pressman, 2005).

La prueba de sistema abarca una serie de pruebas diferentes cuyo propósito principal es ejercitar profundamente el sistema de cómputo. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas (Pressman, 2005).

Para validar la propuesta de solución se realizaron pruebas unitarias y pruebas funcionales, comprendidas en los niveles de prueba de unidad y de sistema respectivamente.

#### Pruebas unitarias

Para realizar las pruebas unitarias se utiliza la técnica de caja blanca. La prueba de caja blanca del software se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejercen conjuntos específicos de condiciones, bucles o ambos (Pressman, 2005).

Symfony2 integra una biblioteca independiente, llamada PHPUnit, para proporcionar una rica plataforma de estas pruebas. Cuando en Symfony2 se genera un *bundle*, se define una clase cuyo nombre termina con *Test* (Potencier, 2011), un ejemplo de ello es la clase `FacultadControllerTest` (). A continuación, se refleja en la ilustración 6, la validación de la entidad "Facultad":



```

class FacultadControllerTest extends \PHPUnit_Framework_TestCase
{
    private $validator;

    public function setUp()
    {
        $this->validator = Validation::createValidatorBuilder()
            ->enableAnnotationMapping()
            ->getValidator();
    }

    public function testValidarResultado()
    {
        $facultad = new Facultad();
        $facultad->setNombre('Facultad 1');
        $listaErrores = $this->validator->validate($facultad);
        $this->assertGreaterThan(0, $listaErrores->count(),
            'El nombre no puede dejarse en blanco.');
```

Ilustración 6: Validación de la entidad “Facultad”

### Resultados de las pruebas unitarias

Los resultados los muestra *PHPUnit* en la consola, con el siguiente comando: “*phpunit -c app -tap*”. A continuación, en la ilustración 7, se reflejan dichos resultados.

```

$ phpunit -c app --tap

TAP version 1
ok 1 - Sitio \ FacultadBundle \ Test \ Entity \ Facultad : : testValidarResultado
```

Ilustración 7: Resultado de la prueba unitaria “Validación de la entidad Facultad”

### Pruebas funcionales

Para realizar las pruebas funcionales se utiliza la técnica de caja negra, haciendo uso del método de partición equivalente. Las pruebas de caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software (Pressman, 2005). La partición equivalente es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de los cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de pruebas que deben desarrollarse (Pressman, 2005).

A continuación, se muestra en la tabla 15, un ejemplo de las pruebas funcionales: caso de prueba “Crear un Grupo”. El resto de estas pruebas se encuentran en el **Anexo\_6**.

**Tabla 15: Caso de prueba "Crear un grupo"**

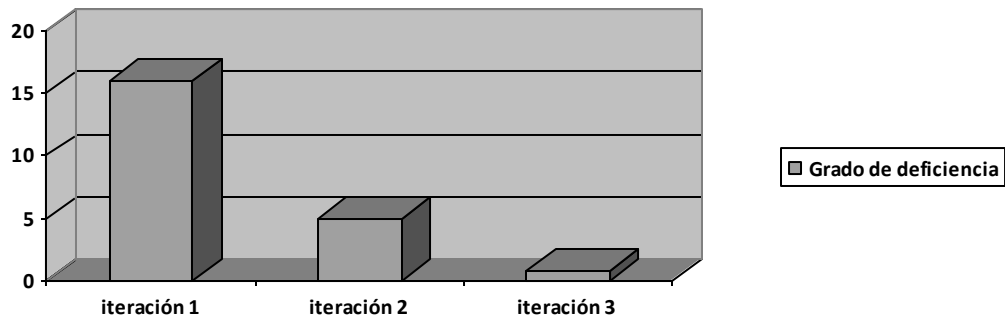
Escenario	Descripción	Variable 1 “Nombre del grupo”	Variable 2 “facultad”	Respuesta del sistema	Flujo central
EC1.Crear grupo correctamente	Mediante este escenario se crea el grupo correctamente en el sistema.	V “Grupo”	V “Facultad 1”	El sistema crea correctamente el grupo y muestra al usuario los datos creados.	El usuario una vez autenticado con los privilegios otorgados selecciona “Grupo” en el menú principal, el sistema muestra todos los datos de los grupos en caso de que existan, este da la opción de crear un nuevo grupo. El usuario escoge la opción de “Crear un nuevo grupo”. El sistema muestra los campos a llenar por el usuario: - facultad -nombre del grupo  El usuario llena todos los campos satisfactoriamente y presiona el botón “Crear” y se agrega el nuevo grupo a la base de datos.
		V “Nombre de 25 caracteres”	V “Facultad 1”		
EC2.El	Mediante este	I	V	El sistema muestra	El usuario una vez

elemento existe.	escenario se verifica que el grupo ya existe en el sistema.	"Grupo"	"Facultad 1"	un mensaje de error "El elemento ya existe" y no lo suscribe.	autenticado con los privilegios otorgados selecciona "Grupo" en el menú principal, el sistema muestra todos los datos de los grupos en caso de que existan, este te da la opción de crear un nuevo grupo. El usuario escoge la opción de "Crear un nuevo grupo". El sistema muestra los campos a llenar por el usuario: - facultad - nombre del grupo  El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear". El sistema le muestra el mensaje "El elemento ya existe" y no se procede con dicha creación.
EC3.Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear un grupo.	I "Vacío"	I "Vacío"	El sistema muestra el mensaje "Por favor rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona "Grupo" en el menú principal, el sistema muestra todos los datos de los grupo en caso de que existan, este te da la opción de crear un nuevo grupo El usuario escoge la opción de "Crear un nuevo grupo". El sistema muestra los campos a llenar por el usuario: - facultad - nombre del grupo Estos campos siguiendo las precondiciones del caso de prueba.
		V "Grupo"	I "Vacío"		
		I "Vacío"	V "Facultad"		

					El usuario no llena todos los campos y presiona el botón "Crear". El sistema muestra el mensaje "Existen campos vacíos"
EC4.Insertar datos incorrectos.	Mediante este escenario se permite no introducir datos incorrectos.	I "Grupo\$%"	V "Facultad"	El sistema marca el error en rojo y muestra un mensaje "Entre solo letras, espacios y números".	El usuario una vez autenticado con los privilegios otorgados selecciona "Grupo" en el menú principal, el sistema muestra todos los datos de los grupo en caso de que existan, este te da la opción de crear un nuevo grupo El usuario escoge la opción de "Crear un nuevo grupo". El sistema muestra los campos a llenar por el usuario: - facultad -nombre del grupo  Estos campos siguiendo las precondiciones del caso de prueba. El usuario inserta datos incorrectos y presiona el botón "Crear":  El sistema marca el campo en rojo y muestra el mensaje "Este campo solo acepta letras y números".
		V "Grupo"	NA		
		I "Nombre de 25 caracteres"	V "Facultad"	El sistema no permite seguir introduciendo caracteres.	

### Resultados de las pruebas funcionales

Las pruebas funcionales al sistema se realizaron en 3 iteraciones, a continuación, en la ilustración 8 se refleja la cantidad de no conformidades detectadas en cada una de estas iteraciones:



**Ilustración 8: Diagrama de deficiencias**

En la iteración 1 se detectaron 16 no conformidades, la mayoría de ellas eran no conformidades referentes a la validación del código, errores ortográficos en la interfaz de la página.

Se realizó una segunda iteración para rectificar los errores detectados en la primera iteración de pruebas. Fueron detectadas 5 no conformidades, las mismas no se correspondían con las de la iteración anterior, las cuales se reflejaban en respuesta no adecuada, ya que el sistema no validaba entrada de datos incorrectos y la cantidad de caracteres a insertar en el mismo.

En la tercera iteración de prueba no se detectaron no conformidades, por lo que se puede confirmar que el software tiene una calidad aceptable.

### **3.5 Conclusiones parciales**

En este capítulo se describieron los pasos a seguir para llevar a cabo el diseño, la implementación y las pruebas al sistema. Se mencionan las pruebas realizadas a la propuesta de solución: unitarias y funcionales. Las pruebas funcionales se realizaron en 3 iteraciones, en la primera y segunda iteración, se detectaron las no conformidades referentes a la validación de código, errores ortográficos, la validación de entrada de datos incorrectos y la cantidad de caracteres a insertar por el usuario. En la última iteración no se detectaron no conformidades demostrándose que la calidad del sistema es aceptable. Además, se realizaron pruebas unitarias, demostrando el correcto funcionamiento de cada una de las funcionalidades de la aplicación.

## **Conclusiones generales**

Se realizó un estudio de los principales conceptos y sistemas homólogos, determinándose la necesidad de un nuevo sistema para gestionar los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.

Se desarrolló un sistema informático para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software, el cual puede ser generalizado para otros departamentos metodológicos centrales en la Universidad de las Ciencias Informáticas.

Al sistema desarrollado se le realizaron pruebas unitarias y funcionales, detectándose los errores existentes, a los cuales se les dio una solución inmediata. Estas pruebas sirvieron para validar la calidad del producto, la cual es aceptable.

Los resultados obtenidos en el transcurso de la investigación fueron recopilados en el presente documento, el cual podrá consultarse en futuras investigaciones.

## Recomendaciones

Con el objetivo de dar continuidad al presente trabajo se recomienda:

- ✓ La integración a otros sistemas pertenecientes a la universidad, entre ellos al de Gestión Universitaria.
- ✓ Permitir a los usuarios autenticarse por el dominio.

## Referencias bibliográficas

1. Álvarez, Ernesto Quiñones (2010). PostgreSQL [Libro]. - [s.l.] : *University Javeriana* – 2010.
2. Álvarez, Miguel Angel (2011). Manual de CodeIgniter [Libro]. - [s.l.] : *Empresrials* - 2011.
3. Álvaro, José Antonio (2001). Sistema de indicadores para la mejora y el control integrado de la calidad de los procesos [Libro]. - [s.l.] : *Universitat Jaume I*, 2001.
4. Andyberna, Lucas (2011). HTML5, sus ventajas [Libro]. - [s.l.] : *Universitat Jaume I* - 2011.
5. Baños, Roberto (2012) Universidad de Ciencias Médicas de Cienfuegos [En línea] Disponible en: <http://www.rcm.cu> - 2012. – (Consultado en Enero de 2013).
6. Barra, Benjamin Alonso (2001) Departamento de Sistemas Informáticos [En línea] Disponible en: <http://www.info-ab.uclm.es>. - 2001. – (Consultado en Diciembre de 2012). -.
7. Batista, M. ; Arana y N. (2011) La educación en valores: una propuesta pedagógica para la formación profesional [Libro]. - [s.l.] : *Empresrials* - 2011.
8. Booch, G. ; Rumbaugh, J. ; Jacobson, I. (1999) Lenguaje de modelado [Libro]. - [s.l.] : *Edition* 1999.
9. Camisón, César (2009) La gestión de la calidad por procesos. Técnicas y herramientas de calidad [Libro]. - [s.l.] : *Editorials Joshua* - 2009.
10. Carrasco, Juan Bravo. (2011) Gestión por procesos [Libro]. - [s.l.] : 204.790, 2011.
11. Cortez, Jessica (2012) Hojas de estilo en cascada (CSS) [Libro]. - [s.l.] : 105.600 - 2012.
12. Dibbern, Alberto (2001) Universidad Nacional de Luján [En línea] Disponible en: <http://www.unlu.edu.ar> . – (Consultado en Diciembre de 2012). - 2001
13. Flores, Miguel Sosa (2007) El proceso docente educativo: sus componentes [Libro]. - [s.l.] : Edición Lid Editor, 2007.
14. Gamma, E. ; Helm, R. (1995) Fundamentos de Ingeniería de Software. Patrones de diseño. [Libro]. - 1995.
15. García, Ricardo Fernández. (2010) Sistema de la gestión de la calidad, ambiente y prevención de riesgos [Libro]. - [s.l.] : Club universitario Casa del Libro, 2010.



16. García, Sonia Fuentes. (2008) Universidad de Ciencias Médicas de Matanza [En línea] Disponible en: <http://www.cpimtz.sld.cu> - (Consultado en Enero de 2013) - 2008.
17. Garrido, Miguel Montero (2011) Procesos de negocio [Libro]. - [s.l.] : *Legendary university* - 2011.
18. Gil-Morell, M. (2006). Carta del Rector de la Universidad de las Ciencias Informáticas a los visitantes al Sitio Web de la universidad - (Consultado en Mayo de 2013).
19. Gómez, Juan (2006) Metodologías ágiles de desarrollo de software [Informe]. Disponible en: <http://moodle.urv.net/moodle/file.php> - (Consultado en Enero de 2013) - 2006.
20. Gutiérrez, Javier (2009) Herramientas web [Libro]. - [s.l.] : 209-222 - 2009.
21. Hernández, B. (2005) Reflejos actuales del uso de las tecnologías modernas. [En línea] Disponible en: <http://tecfa.unige.ch/tecfa/publicat> - (Consultado en Febrero de 2013) -2005
22. Herrera, Elizabeth Arriaga (2011) Sistema de la Universidad Nacional Autónoma de México [En línea] Disponible en: <http://www.distancia.cuaed.unam.mx> – ( Consultado en Enero de 2013). - 2011.
23. Joskowicz, José (2008) Reglas y prácticas en *extreme programming* [Libro]. - [s.l.] : 206145 - 2008.
24. Mariñán, Alberto; Molpeceres, Touris; Martín Pérez (2002) Arquitectura empresarial y software libre [Libro]. - [s.l.] : *Edition 46* - 2002.
25. Martínez, Jairo Chapela (2007) Introducción al entorno de desarrollo Eclipse [Libro]. - [s.l.] : *Legendary university* - 2007.
26. Mehdi, Achour; Friedhelm, Betz; Antony, Dovgal; Nuno, Lopes; Hannes, Magnusson; Georg, Richter; Damien, Seguy; Jakub, Vrana (1997) Manual de PHP [Libro]. - [s.l.] : 20946 - 1997.
27. Meneses, Luis Eduardo Arriola. (2008) Sistema de información para la evaluación del programa emprendedor universitario [Libro]. - [s.l.] : *Legendary university* - 2006.
28. Mónica, Joshwa (2009) Módulos de Multiprocesamiento [En línea] Disponible en: <http://httpd.apache.org/docs/trunk/es/mpm> - (Consultado en Febrero de 2013). - 2009. -
29. Pérez, Malena Zelada (1999) Universidad de Ciencias Médicas de la Habana [En línea]. Disponible en: <http://instituciones.sld.cu> - (Consultado en Enero de 2013.) - 1999.

30. Peñalver, Romero G.M. (2008) "Trabajo de diploma: Metodología ágil para proyectos de software libre". Ciudad de La Habana, Universidad de las Ciencias Informáticas. 2008.
31. Potencier, Fabien Symphony.(2011) La guía definitiva de Symfony 2. [Libro]. - 2011.
32. Pressman, Roger (2005) Ingeniería del Software: Un Enfoque Práctico. Sexta Edición ed. McGraw-Hill Companies, 614 p. ISBN 8448132149.- 2005.
33. Rivera, Rafael Andrés González (2000) Sistema de la Facultad de Ingeniería de la Universidad Javeriana de Bogotá. [En línea] Disponible en: [http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/fac\\_presentacion](http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/fac_presentacion) (Consultado en Noviembre de 2012) - 2000.
34. Ruiz, Marcelo Hernán (2008) Programación web avanzada [Libro]. - [s.l.] : Editorial MP Ediciones, 2008.
35. Tauber, Fernando (2008) Universidad Nacional de La Plata [En línea] Disponible en: <http://www.unlp.edu.ar> - (Consultado en Enero de 2013) - 2008.
36. Triana, Israel Mazarío (2010) Enseñar y aprender: conceptos y contextos [Libro]. - [s.l.] : Editorial 10025 - 2010.
37. Véneré, Marcelo (1998) Facultad de Ciencias Exactas|UNICEN [En línea] Disponible en: <http://www.unicen.edu.ar/content/facultad-de-ciencias-exactas> - (Consultado en Diciembre de 2012)- 1998.
38. Zayas, Carlos Álvarez (1999) La interdisciplinariedad en los Departamentos de Ciencias Exactas de la Educación Media. Resúmenes del Congreso Internacional de Didáctica de las Ciencias [Libro]. -1999.

## Bibliografías

1. Álvarez, Ernesto Quiñones (2010). PostgreSQL [Libro]. - [s.l.] : *University Javeriana* – 2010.
2. Álvarez, Miguel Angel (2011). Manual de CodeIgniter [Libro]. - [s.l.] : *Empresrials* - 2011.
3. Álvaro, José Antonio (2001). Sistema de indicadores para la mejora y el control integrado de la calidad de los procesos [Libro]. - [s.l.] : *Universitat Jaume I*, 2001.
4. Andyberna, Lucas (2011). HTML5, sus ventajas [Libro]. - [s.l.] : *Universitat Jaume I* - 2011.
5. Baños, Roberto (2012) Universidad de Ciencias Médicas de Cienfuegos [En línea] Disponible en: <http://www.rcm.cu> - 2012. – (Consultado en Enero de 2013).
6. Barra, Benjamin Alonso (2001) Departamento de Sistemas Informáticos [En línea] Disponible en: <http://www.info-ab.uclm.es>. - 2001. – (Consultado en Diciembre de 2012). -.
7. Batista, M. ; Arana y N. (2011) La educación en valores: una propuesta pedagógica para la formación profesional [Libro]. - [s.l.] : *Empresrials* - 2011.
8. Booch, G. ; Rumbaugh, J. ; Jacobson, I. (1999) Lenguaje de modelado [Libro]. - [s.l.] : *Edition* 1999.
9. Camisón, César (2009) La gestión de la calidad por procesos. Técnicas y herramientas de calidad [Libro]. - [s.l.] : *Editorials Joshua* - 2009.
10. Carrasco, Juan Bravo. (2011) Gestión por procesos [Libro]. - [s.l.] : 204.790, 2011.
11. Cortez, Jessica (2012) Hojas de estilo en cascada (CSS) [Libro]. - [s.l.] : 105.600 - 2012.
12. Dibbern, Alberto (2001) Universidad Nacional de Luján [En línea] Disponible en: <http://www.unlu.edu.ar> – (Consultado en Diciembre de 2012). - 2001
13. Flores, Miguel Sosa (2007) El proceso docente educativo: sus componentes [Libro]. - [s.l.] : Edición Lid Editor, 2007.
14. Gamma, E. ; Helm, R. (1995) Fundamentos de Ingeniería de Software. Patrones de diseño. [Libro]. - 1995.
15. García, Ricardo Fernández. (2010) Sistema de la gestión de la calidad, ambiente y prevención de riesgos [Libro]. - [s.l.] : Club universitario Casa del Libro, 2010.

16. García, Sonia Fuentes. (2008) Universidad de Ciencias Médicas de Matanza [En línea] Disponible en: <http://www.cpimtz.sld.cu> - (Consultado en Enero de 2013) - 2008.
17. Garrido, Miguel Montero (2011) Procesos de negocio [Libro]. - [s.l.] : *Legendary university* - 2011.
18. Gil-Morell, M. (2006). Carta del Rector de la Universidad de las Ciencias Informáticas a los visitantes al Sitio Web de la universidad - (Consultado en Mayo de 2013).
19. Gómez, Juan (2006) Metodologías ágiles de desarrollo de software [Informe]. Disponible en: <http://moodle.urv.net/moodle/file.php> - (Consultado en Enero de 2013) - 2006.
20. Gutiérrez, Javier (2009) Herramientas web [Libro]. - [s.l.] : 209-222 - 2009.
21. Hernández, B. (2005) Reflejos actuales del uso de las tecnologías modernas. [En línea] Disponible en: <http://tecfa.unige.ch/tecfa/publicat> - (Consultado en Febrero de 2013) -2005
22. Herrera, Elizabeth Arriaga (2011) Sistema de la Universidad Nacional Autónoma de México [En línea] Disponible en: <http://www.distancia.cuaed.unam.mx> – ( Consultado en Enero de 2013). - 2011.
23. Joskowicz, José (2008) Reglas y prácticas en *extreme programming* [Libro]. - [s.l.] : 206145 - 2008.
24. Mariñán, Alberto; Molpeceres, Touris; Martín Pérez (2002) Arquitectura empresarial y software libre [Libro]. - [s.l.] : *Edition 46* - 2002.
25. Martínez, Jairo Chapela (2007) Introducción al entorno de desarrollo Eclipse [Libro]. - [s.l.] : *Legendary university* - 2007.
26. Mehdi, Achour; Friedhelm, Betz; Antony, Dovgal; Nuno, Lopes; Hannes, Magnusson; Georg, Richter; Damien, Seguy; Jakub, Vrana (1997) Manual de PHP [Libro]. - [s.l.] : 20946 - 1997.
27. Meneses, Luis Eduardo Arriola. (2008) Sistema de información para la evaluación del programa emprendedor universitario [Libro]. - [s.l.] : *Legendary university* - 2006.
28. Mónica, Joshwa (2009) Módulos de Multiprocesamiento [En línea] Disponible en: <http://httpd.apache.org/docs/trunk/es/mpm> - (Consultado en Febrero de 2013). - 2009. -
29. Pérez, Malena Zelada (1999) Universidad de Ciencias Médicas de la Habana [En línea]. Disponible en: <http://instituciones.sld.cu> - (Consultado en Enero de 2013) - 1999.

30. Peñalver, Romero G.M. (2008) "Trabajo de diploma: Metodología ágil para proyectos de software libre". Ciudad de La Habana, Universidad de las Ciencias Informáticas. 2008.
31. Potencier, Fabien Symphony.(2011) La guía definitiva de Symfony 2. [Libro]. - 2011.
32. Rivera, Rafael Andrés González (2000) Sistema de la Facultad de Ingeniería de la Universidad Javeriana de Bogotá. [En línea] Disponible en: [http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/fac\\_presentacion](http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/fac_presentacion) (Consultado en Noviembre de 2012) - 2000.
33. Ruiz, Marcelo Hernán (2008) Programación web avanzada [Libro]. - [s.l.] : Editorial MP Ediciones, 2008.
34. Tauber, Fernando (2008) Universidad Nacional de La Plata [En línea] Disponible en: <http://www.unlp.edu.ar> - (Consultado en Enero de 2013) - 2008.
35. Triana, Israel Mazarío (2010) Enseñar y aprender: conceptos y contextos [Libro]. - [s.l.] : Editorial 10025 - 2010.
36. Vénere, Marcelo (1998) Facultad de Ciencias Exactas|UNICEN [En línea] Disponible en: <http://www.unicen.edu.ar/content/facultad-de-ciencias-exactas> - (Consultado en Diciembre de 2012)- 1998.
37. Zayas, Carlos Álvarez (1999) La interdisciplinariedad en los Departamentos de Ciencias Exactas de la Educación Media. Resúmenes del Congreso Internacional de Didáctica de las Ciencias [Libro]. -1999.

## Glosario de términos

**ASP:** (*Active Server Pages*), es un lenguaje de programación de servidores para generar páginas web dinámicamente.

**CodeIgniter:** es un framework PHP para la creación rápida de aplicaciones web.

**CSS:** es el acrónimo de *Cascading Style Sheets* (hojas de estilo en cascada).

**GPL/GNU:** *GNU General Public License*. Es una licencia creada por la *Free Software Foundation* en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de *software*. Su propósito es declarar que el *software* cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**HTML:** es el acrónimo de *HyperText Markup Language* (Lenguaje de Marcado de Hipertexto) y es el lenguaje que se utiliza para crear las páginas web. Este lenguaje indica a los navegadores cómo deben mostrar el contenido de una página web.

**IDE:** del acrónimo de *Integrated Development Environment* (Entorno de desarrollo integrado).

**JavaScript:** es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

**MySQL:** es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

**OOP:** (*Object Oriented Programming*) programación orientada a objetos.

**Oracle:** es un sistema de gestión de base de datos objeto-relacional (por el acrónimo en inglés de *Object-Relational Data Base Management System*)

**Perl:** es un lenguaje de programación, es un acrónimo de *Practical Extracting and Reporting Language*.

**PHP:** acrónimo *Hypertext Pre-processor*, es un lenguaje muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.

**PostgreSQL:** es un sistema de gestión de bases de datos objeto-relacional y con su código fuente disponible libremente.

**Symfony:** es un completo *framework* diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web.

**UCI:** Universidad de las Ciencias Informáticas.

**UML:** (*Unified Modeling Language*) es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño.

**W3C:** son las siglas de *World Wide Web Consortium*, y es una comunidad internacional donde los estados miembros trabajan para poder desarrollar estándares para el desarrollo web y así ayudar a un mejor desarrollo del Internet mundialmente.

**XHTML:** acrónimo inglés de *eXtensible Hyper Text Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

**XML:** son las siglas de *Extensible Markup Language*, una especificación/lenguaje de programación. Es un sistema estándar de codificación de información. Los programas que utilizan el formato XML pueden intercambiar fácilmente sus datos, ya que responden a una misma lógica interna.

## Anexos

### Anexo\_1: Fichas de los principales procesos

Tabla 16: Ficha de proceso “Gestión de las preparaciones metodológicas”

Ficha de procesos			
<b>Identificador:</b> 2	<b>Nombre del proceso:</b> Gestión de las preparaciones metodológicas.	<b>Tipo:</b> Principal	<b>Responsable:</b> Jefe de departamento
<b>Misión:</b> Se desarrollan con el objetivo de orientar metodológicamente a los profesores, a partir del análisis de las principales actividades, el desarrollo de ciclos de trabajo metodológico y la realización de actividades de superación en contenidos de la disciplina. Tienen una gran importancia, puesto que contribuyen a aumentar la calidad de las clases impartidas por los diferentes profesores.			
<b>Precondición:</b>			
<b>Síntesis del proceso:</b> Inicialmente el jefe del departamento realiza la planificación de las preparaciones metodológicas en un periodo determinado, designando visitantes a estas. Las visitas son las encargadas de tomar la asistencia y los principales acuerdos de la preparación. Se genera el acta y se guarda para poder consultarla en otro momento.			

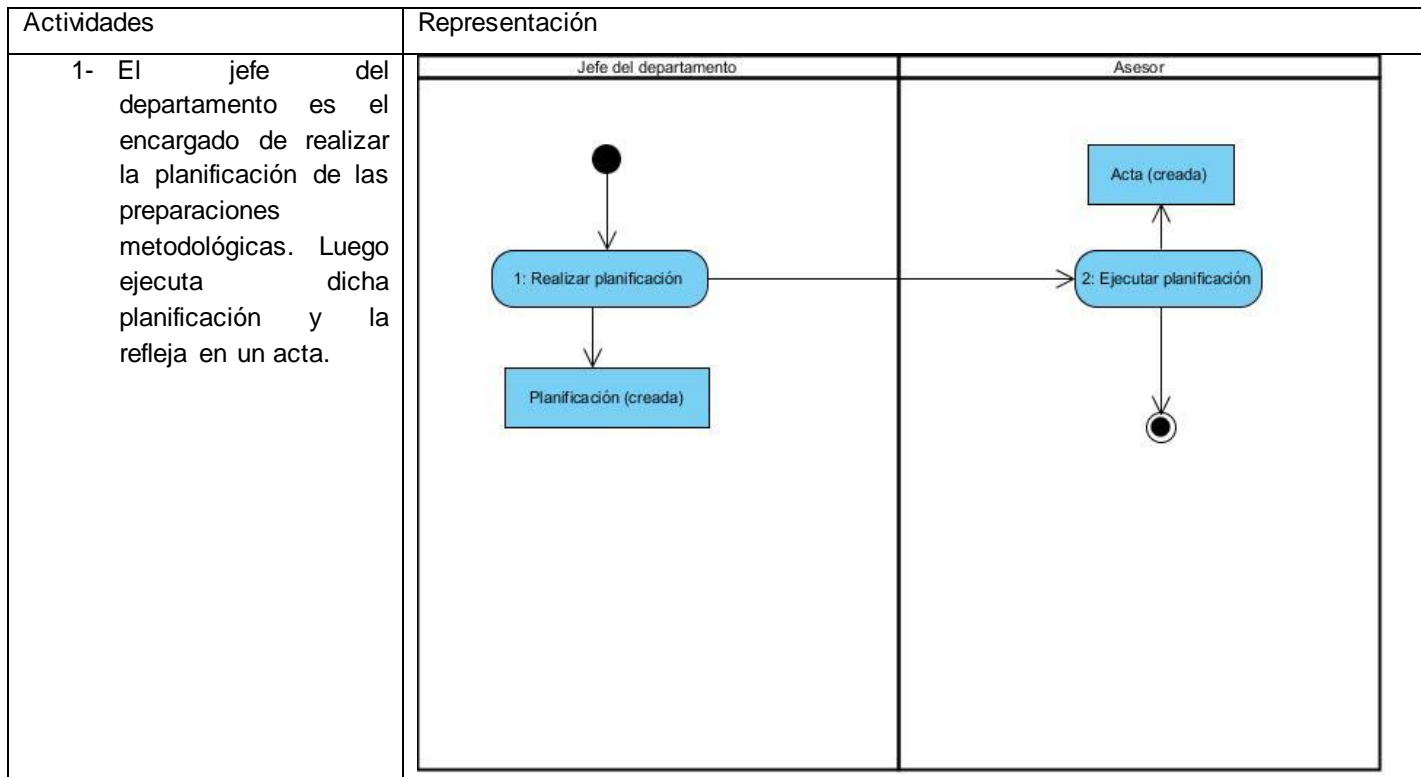
Tabla 17: Ficha de proceso “Elaboración de los planes de trabajo”

Ficha de procesos			
<b>Identificador:</b> 3	<b>Nombre del proceso:</b> Elaboración de los planes de trabajo.	<b>Tipo:</b> Principal	<b>Responsable:</b> Jefe de departamento
<b>Misión:</b> La elaboración de los planes de trabajo les sirve a los asesores como guía para su labor diaria, pues en este se reflejan todas las actividades propuestas por la facultad y el departamento. Estos planes se realizan mensualmente y constituyen la base para la elaboración de las evaluaciones profesoriales.			
<b>Precondición:</b>			
<b>Síntesis del proceso:</b> Inicialmente el jefe del departamento realiza la elaboración de la planificación de los planes de trabajo para cada uno de sus subordinados. El plan contiene las actividades a realizar en un mes determinado, sobre la base de las cuales se le da una evaluación al asesor del mes. Las evaluaciones de cada asesor se guardan para poder consultarlas en otro momento que sea necesario.			

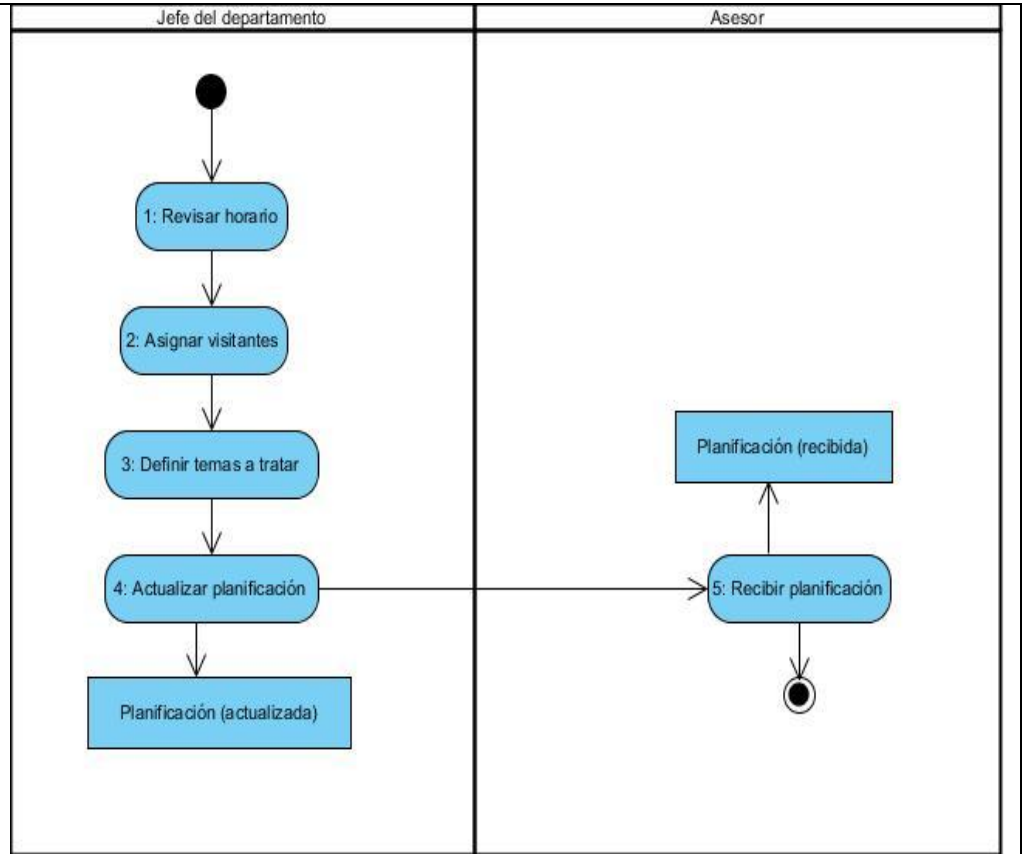


## Anexo\_2: Flujo de actividades de los principales procesos

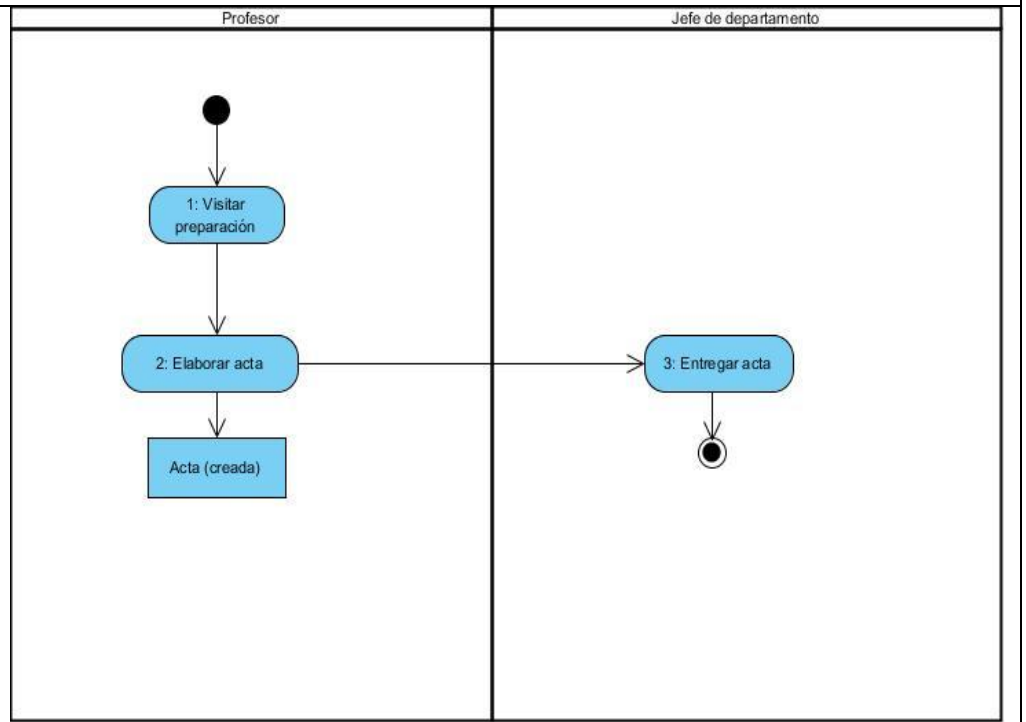
Tabla 18: Flujo de actividades del proceso “Gestión de las preparaciones metodológicas”



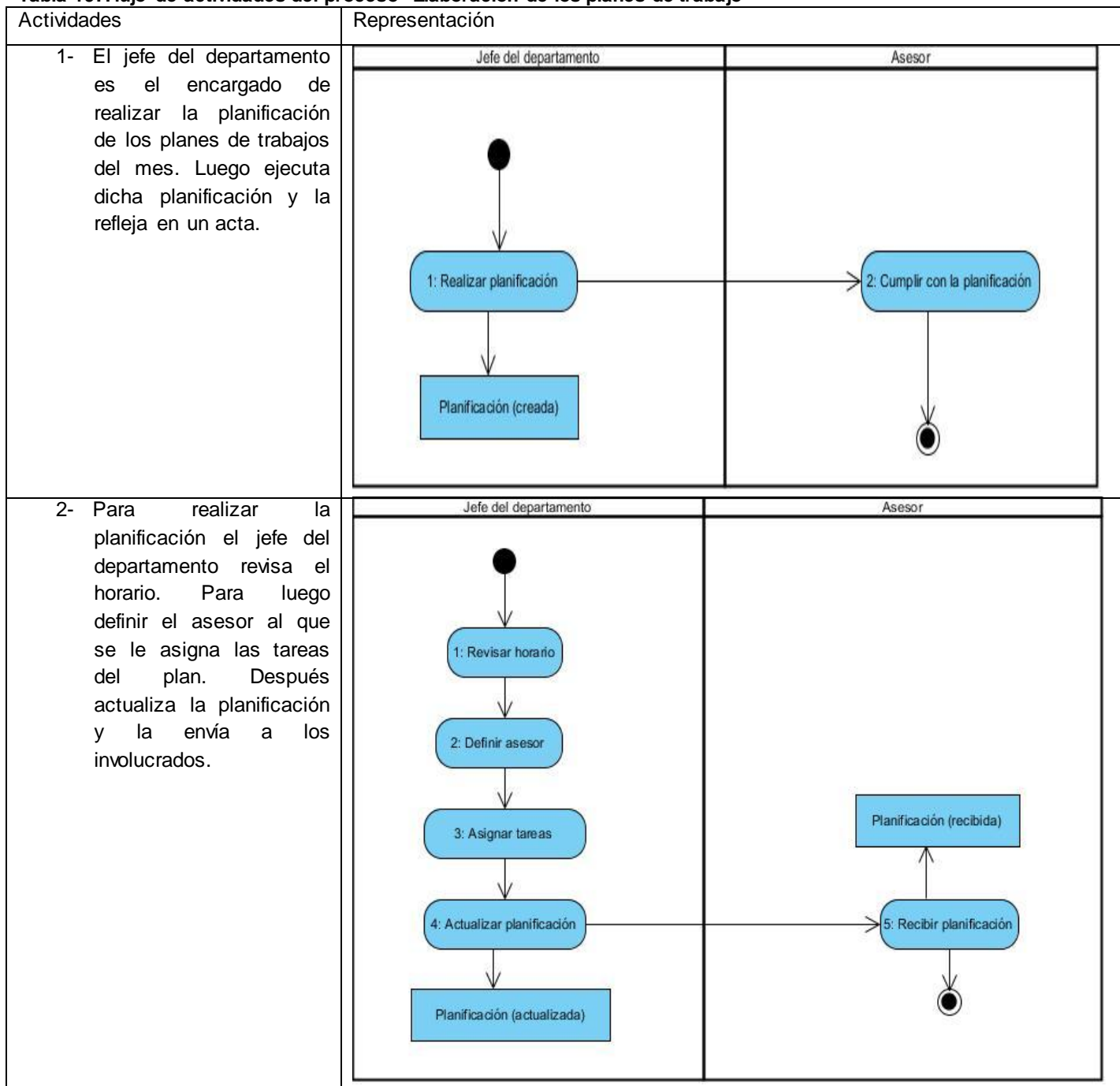
2- Para realizar la planificación el jefe del departamento revisa el horario. Para luego asignar los visitantes a esta y definir los temas a tratar en la misma. Después actualiza la planificación y la envía a los involucrados.



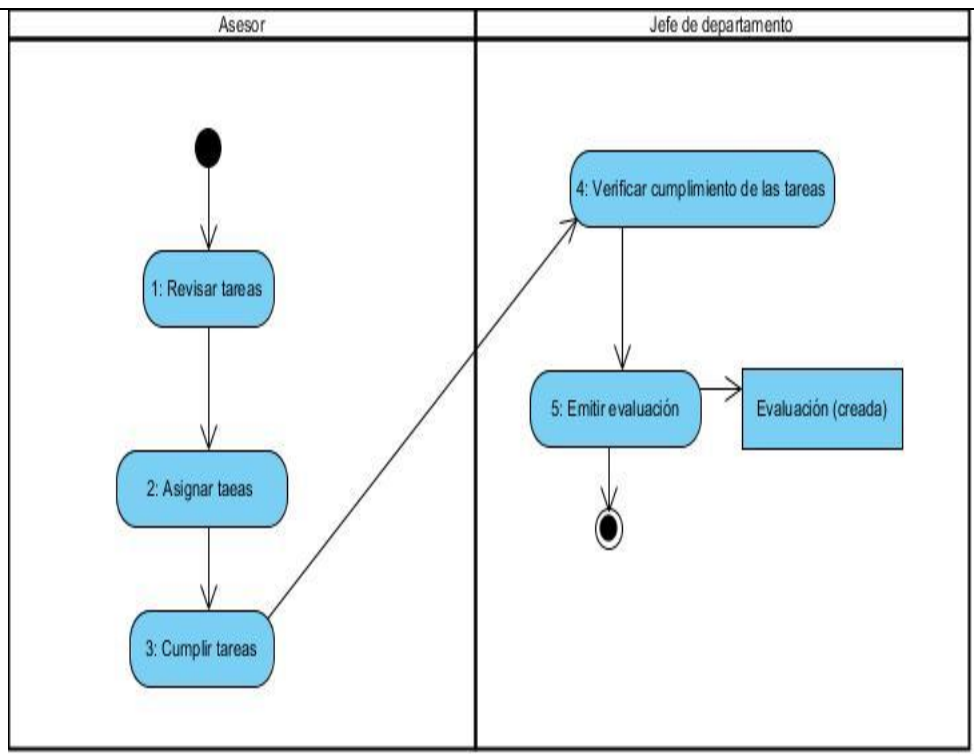
3- Para la ejecución de la planificación los visitantes elaboran un acta con los principales acuerdos tomados es esta preparación metodológica y se la envían al jefe de departamento.



**Tabla 19: Flujo de actividades del proceso “Elaboración de los planes de trabajo”**



3- Para la ejecución de la planificación se revisan las tareas asignadas, las cuales se les da el cumplimiento y se envía al jefe del departamento.



### Anexo\_3: Historias de usuarios

Tabla 20: HU “Modificar la planificación de los controles a clases”

Historia de usuario.	
No.:2	Nombre: Modificar la planificación de los controles a clases.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se modifique la planificación de los controles a clases que se realizarán en el periodo establecido. El usuario selecciona en el menú principal “Controles a clases”, el sistema le muestra al usuario la lista de los controles planificados y el usuario selecciona el control a modificar.	
Observaciones:	

Tabla 21: HU “Adicionar un control a clase a un asesor”

Historia de usuario.	
No.:3	Nombre: Adicionar un control a clase a un asesor.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se adicione un control a clase a un asesor determinado. El usuario selecciona en el menú principal “Controles a clases” y allí accede a crear un nuevo control, el sistema muestra el formulario donde el usuario llena los datos y se crea el nuevo control a clases.	
Observaciones:	

Tabla 22: HU “Modificar un control a clase a un asesor”

Historia de usuario.	
No.:4	Nombre: Modificar un control a clase a un asesor.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite modificar un control a clase a un asesor determinado. El usuario selecciona en el menú principal “Controles”, el sistema muestra una lista con todos los controles planificados. Allí se selecciona el control que se desea modificar. El sistema muestra los datos del control seleccionado, permitiendo así la modificación del control.	
Observaciones:	

Tabla 23: HU “Eliminar un control a clase a un asesor”

Historia de usuario.	
No.:5	Nombre: Eliminar un control a clase a un asesor.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio

Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se elimine un control a clase a un asesor determinado. El usuario selecciona en el menú principal “Controles”, el sistema muestra una lista con todos los controles planificados. Allí se selecciona el control que se desea eliminar. El sistema muestra los datos del control a eliminar, el usuario selecciona la opción de eliminar, eliminándose el control al asesor.	
Observaciones:	

**Tabla 24: HU “Eliminar la planificación de los controles a clases”**

Historia de usuario.	
No.:6	Nombre: Eliminar la planificación de los controles a clases.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se elimine la planificación de los controles a clases. El usuario selecciona en el menú principal “Controles a clases”, ya dentro del módulo escoge la opción de “Eliminar la planificación” y el sistema elimina toda la planificación de los controles existentes.	
Observaciones: Solo se elimina la planificación luego de que haya culminado el semestre.	

**Tabla 25: HU “Mostrar la planificación de los controles a clases”**

Historia de usuario.	
No.:7	Nombre: Mostrar la planificación de los controles a clases.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestre la planificación de los controles a clases que se realizarán en el periodo establecido. El usuario selecciona en el menú principal “Controles” y el sistema muestra una lista con todos los controles planificados.	
Observaciones:	

**Tabla 26: HU “Dar seguimiento a un control a clase”**

Historia de usuario.	
No.:8	Nombre: Dar seguimiento a un control a clase.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se le dé un seguimiento al control a clase seleccionado. El usuario selecciona en el menú principal “Controles” y el sistema muestra una lista con todos los controles planificados. Allí se selecciona el control al que se le desea dar un resultado, luego se procesa a llenar los campos del resultado.	

Observaciones:
----------------

**Tabla 27: HU “Generar reporte total de evaluaciones de controles a clases por asignatura”**

Historia de usuario.	
No.:9	Nombre: Generar reporte total de evaluaciones de controles a clases por asignatura.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestre y genere un reporte total de las evaluaciones a los controles a clases por asignatura. El usuario selecciona en el menú principal “Reportes por asignaturas”, allí se selecciona la asignatura y el sistema genera tal reporte.	
Observaciones:	

**Tabla 28: HU “Generar reporte total de evaluaciones de controles a clases”**

Historia de usuario.	
No.:10	Nombre: Generar reporte total de evaluaciones de controles a clases.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestre y genere un reporte total de las evaluaciones a los controles a clases. El usuario selecciona en el menú principal “Reporte total” y el sistema genera un reporte de todos los controles a clases.	
Observaciones:	

**Tabla 29: HU “Generar reporte del estado actual de controles a clases por asignatura”**

Historia de usuario.	
No.:11	Nombre: Generar reporte del estado actual de controles a clases por asignatura.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestre y genere un reporte del estado actual de los controles a clases por asignatura. El usuario selecciona en el menú principal “Reporte del estado actual por asignaturas”, allí se selecciona la asignatura y el sistema genera tal reporte.	
Observaciones:	

**Tabla 30: HU “Generar reporte del estado actual de la planificación de controles a clases”**

Historia de usuario.	
No.:12	Nombre: Generar reporte del estado actual de la planificación

	de controles a clases.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestre y genere un reporte del estado actual de los controles a clases. El usuario selecciona en el menú principal "Reporte del estado actual" y el sistema genera un reporte de todos los controles a clases.	
Observaciones:	

**Tabla 31: HU "Adicionar nomenclador"**

Historia de usuario.	
No.:13	Nombre: Adicionar nomenclador.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se adicione un nomenclador del tipo seleccionado, ya sea departamento, disciplina, asignatura, facultad, local, grupo y categoría docente. El usuario selecciona en el menú principal "Administración", ya dentro del módulo escoge el nomenclador que desea adicionar, llenando los datos en el formulario correspondiente.	
Observaciones:	

**Tabla 32: HU "Modificar nomenclador"**

Historia de usuario.	
No.:14	Nombre: Modificar nomenclador.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se modifique un nomenclador del tipo seleccionado, ya sea departamento, disciplina, asignatura, facultad, local, grupo y categoría docente. El usuario selecciona en el menú principal "Administración", ya dentro del módulo escoge el nomenclador que desea modificar, procediendo a modificar los datos del nomenclador deseado.	
Observaciones:	

**Tabla 33: HU "Eliminar nomenclador"**

Historia de usuario.	
No.:15	Nombre: Eliminar nomenclador.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se elimine un nomenclador del tipo seleccionado, ya sea departamento, disciplina, asignatura, facultad, local,	



grupo y categoría docente. El usuario selecciona en el menú principal “Administración”, ya dentro del módulo escoge el nomenclador que desea eliminar, el sistema le muestra los datos del sistema que desea eliminar y escoge la opción de “Eliminar” y sistema elimina el nomenclador seleccionado.
Observaciones:

**Tabla 34: HU “Modificar la planificación de las preparaciones metodológicas”**

Historia de usuario.	
No.:17	Nombre: Modificar la planificación de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite modificar la planificación de las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones planificadas, allí se selecciona la que se quiere modificar y procede con la modificación de sus datos.	
Observaciones:	

**Tabla 35: HU “Eliminar preparaciones metodológicas”**

Historia de usuario.	
No.:18	Nombre: Eliminar preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se eliminen preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones que se desean realizar, allí se selecciona la que se quiere eliminar, y el sistema le muestra los datos de la preparación que se desea eliminar, allí se escoge la opción “Eliminar” y el sistema elimina dicha preparación.	
Observaciones:	

**Tabla 36: HU “Mostrar la planificación de las preparaciones metodológicas”**

Historia de usuario.	
No.:19	Nombre: Mostrar la planificación de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Medio	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se muestre la planificación de las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones” y el sistema le muestra todas las preparaciones planificadas en el periodo.	

Observaciones:
----------------

**Tabla 37: HU “Dar seguimiento a las preparaciones metodológicas”**

Historia de usuario.	
No.:20	Nombre: Dar seguimiento a las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite darle un seguimiento a las preparaciones metodológicas efectuadas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones que se desean realizar, allí se selecciona la que se quiere dar seguimiento, llenando los datos requeridos.	
Observaciones:	

**Tabla 38: HU “Generar actas de las preparaciones metodológicas”**

Historia de usuario.	
No.:21	Nombre: Generar actas de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se genere el acta de las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones, allí se selecciona la preparación a la que se le quiere generar el acta y se escoge la opción de “Generar acta”, generando así dicha acta.	
Observaciones:	

**Tabla 39: HU “Imprimir actas de las preparaciones metodológicas”**

Historia de usuario.	
No.:22	Nombre: Imprimir actas de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se imprima el acta de las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones, allí se selecciona la preparación a la que se le quiere generar el acta y se escoge la opción de “Generar acta”, luego de generada dicha acta se selecciona la opción de “Imprimir” y el sistema la imprime.	
Observaciones:	

**Tabla 40: HU “Modificar acta de las preparaciones metodológicas”**

Historia de usuario.	
No.:23	Nombre: Modificar acta de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se modifique el acta de la preparación metodológica. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones, allí se selecciona la preparación a la que se le quiere generar el acta y antes de generar dicha acta se modifica.	
Observaciones:	

**Tabla 41: HU “Eliminar acta de las preparaciones metodológicas”**

Historia de usuario.	
No.:24	Nombre: Eliminar acta de las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se elimine el acta de la preparación metodológica. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones, allí se selecciona la preparación a la que se le quiere eliminar el acta y se procede a eliminar dicha acta.	
Observaciones:	

**Tabla 42: HU “Planificar visitas a las preparaciones metodológicas”**

Historia de usuario.	
No.:25	Nombre: Planificar visitas a las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se planifiquen las visitas a las preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones, se selecciona la preparación a la que se le quiere asignar un visitante en la opción “Asignar visita”, luego se procede a llenar los datos de las diferentes visitas que se desean efectuar a la preparación metodológica escogida.	
Observaciones:	

**Tabla 43: HU “Visualizar la planificación de las visitas a las preparaciones metodológicas”**

Historia de usuario.	
No.:26	Nombre: Visualizar la planificación de las visitas a las

	preparaciones metodológicas.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se visualice la planificación de las visitas a las preparaciones metodológicas. El usuario selecciona en el menú principal "Preparaciones", el sistema muestra las preparaciones metodológicas con sus visitantes asignados.	
Observaciones:	

**Tabla 44: HU "Modificar la planificación de las visitas a las preparaciones metodológicas"**

Historia de usuario.	
No.:27	Nombre: Modificar la planificación de las visitas a las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se modifique la planificación de las visitas a las preparaciones metodológicas. El usuario selecciona en el menú principal "Preparaciones", el sistema muestra las preparaciones metodológicas con sus visitantes asignados, se escoge la opción de "Asignar visitante" y se procede a modificar el visitante asignado.	
Observaciones:	

**Tabla 45: HU "Eliminar visitas a las preparaciones metodológicas"**

Historia de usuario.	
No.:28	Nombre: Eliminar visitas a las preparaciones metodológicas.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que se elimine la planificación de las visitas a una preparación metodológica. El usuario selecciona en el menú principal "Preparaciones", el sistema muestra las preparaciones metodológicas con sus visitantes asignados, se escoge la opción de "Asignar visitante" y allí se eliminar este.	
Observaciones:	

**Tabla 46: HU "Emitir valoración de la preparación metodológica"**

Historia de usuario.	
No.:29	Nombre: Emitir valoración de la preparación metodológica.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 3
Descripción: La historia de usuario permite que emita una valoración de a las	

preparaciones metodológicas. El usuario selecciona en el menú principal “Preparaciones”, el sistema le muestra todas las preparaciones que se desean realizar y este selecciona a la que se le va a emitir la valoración y se llenan los datos.
Observaciones:

**Tabla 47: HU “Crear plan de trabajo a los asesores”**

Historia de usuario.	
No.:30	Nombre: Crear plan de trabajo a los asesores.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que creen los planes de trabajo de los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, ya dentro del módulo escoge la opción “Crear plan” y se procede a llenar los datos para cada plan de trabajo a crear.	
Observaciones:	

**Tabla 48: HU “Mostrar plan de trabajo de los asesores”**

Historia de usuario.	
No.:31	Nombre: Mostrar plan de trabajo de los asesores.
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestren los planes de trabajo de los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, y el sistema muestra la lista con todos los planes de trabajo.	
Observaciones:	

**Tabla 49: HU “Modificar plan de trabajo a los asesores”**

Historia de usuario.	
No.:32	Nombre: Modificar plan de trabajo a los asesores.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se modifiquen los planes de trabajo de los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, el sistema muestra los planes, seleccionando luego el que se desea modificar y se procede a la modificación deseada.	
Observaciones:	

**Tabla 50: HU “Crear evaluación del profesor”**

Historia de usuario.	
No.:33	Nombre: Crear evaluación del profesor.

Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite dar una evaluación a los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, el sistema muestra la lista con todos los planes planificados, allí se selecciona al que se le desea dar una evaluación y se procede a escoger dicha evaluación.	
Observaciones:	

**Tabla 51: HU “Modificar evaluación del profesor”**

Historia de usuario.	
No.:34	Nombre: Modificar evaluación del profesor.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite modificar la evaluación a los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, el sistema muestra la lista con todos los planes planificados, allí se selecciona al que se le desea modificar la evaluación y se procede con la modificación de la misma.	
Observaciones:	

**Tabla 52: HU “Eliminar evaluación del profesor”**

Historia de usuario.	
No.:35	Nombre: Eliminar evaluación del profesor.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite eliminar la evaluación a los asesores del departamento. El usuario selecciona en el menú principal “Plan de trabajo”, el sistema muestra la lista con todos los planes planificados, allí se selecciona al que se le desea eliminar la evaluación y se elimina dicha evaluación.	
Observaciones:	

**Tabla 53: HU “Mostrar las evaluaciones de los profesores”**

Historia de usuario.	
No.:36	Nombre: Mostrar las evaluaciones de los profesores
Usuario: Administrador, asesores del departamento.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se muestren las evaluaciones de los planes de trabajo de los asesores del departamento. El usuario	

selecciona en el menú principal “Plan de trabajo”, y el sistema muestra la lista con todos los planes de trabajo y su respectiva evaluación.
Observaciones:

**Tabla 54: HU “Generar reportes de las evaluaciones de los profesores”**

Historia de usuario.	
No.:37	Nombre: Generar reportes de las evaluaciones de los profesores.
Usuario: Administrador.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se generen reportes sobre las evaluaciones de los asesores del departamento. El usuario selecciona en el menú principal “Reporte de evaluaciones”, y el sistema genera el reporte de estas.	
Observaciones:	

**Tabla 55: HU “Generar corte de resumen de evaluaciones de los profesores”**

Historia de usuario.	
No.:38	Nombre: Generar corte de resumen de evaluaciones de los profesores.
Usuario: Administrador.	
Prioridad en el negocio: Media	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 2
Descripción: La historia de usuario permite que se generen cortes de resumen sobre las evaluaciones de los asesores del departamento. El usuario selecciona en el menú principal “Reporte de resumen de evaluaciones”, y el sistema genera el reporte de estas.	
Observaciones:	

**Tabla 56: HU “Adicionar usuario”**

Historia de usuario.	
No.:39	Nombre: Adicionar usuario.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se adicione un usuario al sistema. El usuario selecciona en el menú principal “Administración”, ya dentro del módulo escoge la opción de adicionar usuario y procese a llenar los datos del usuario de la aplicación.	
Observaciones:	

**Tabla 57: HU “Modificar usuario”**

Historia de usuario.
----------------------

No.:40	Nombre: Modificar usuario.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se modifique un usuario seleccionado. El usuario selecciona en el menú principal "Administración", ya dentro del módulo escoge la opción de modificar usuario y procese a modificar los datos del usuario.	
Observaciones:	

**Tabla 58: HU "Eliminar usuario"**

<b>Historia de usuario.</b>	
No.:41	Nombre: Eliminar usuario.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se elimine un usuario del sistema. El usuario selecciona en el menú principal "Administración", ya dentro del módulo escoge la opción de eliminar usuario y se elimina el usuario.	
Observaciones:	

**Tabla 59: HU "Autenticar usuario"**

<b>Historia de usuario.</b>	
No.:42	Nombre: Autenticar usuario.
Usuario: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 1
Descripción: La historia de usuario permite que se autentique un usuario al sistema.	
Observaciones: Antes de alguna operación en el sistema, el usuario debe estar autenticado en él.	



## Anexo\_4: Prototipos de interfaz

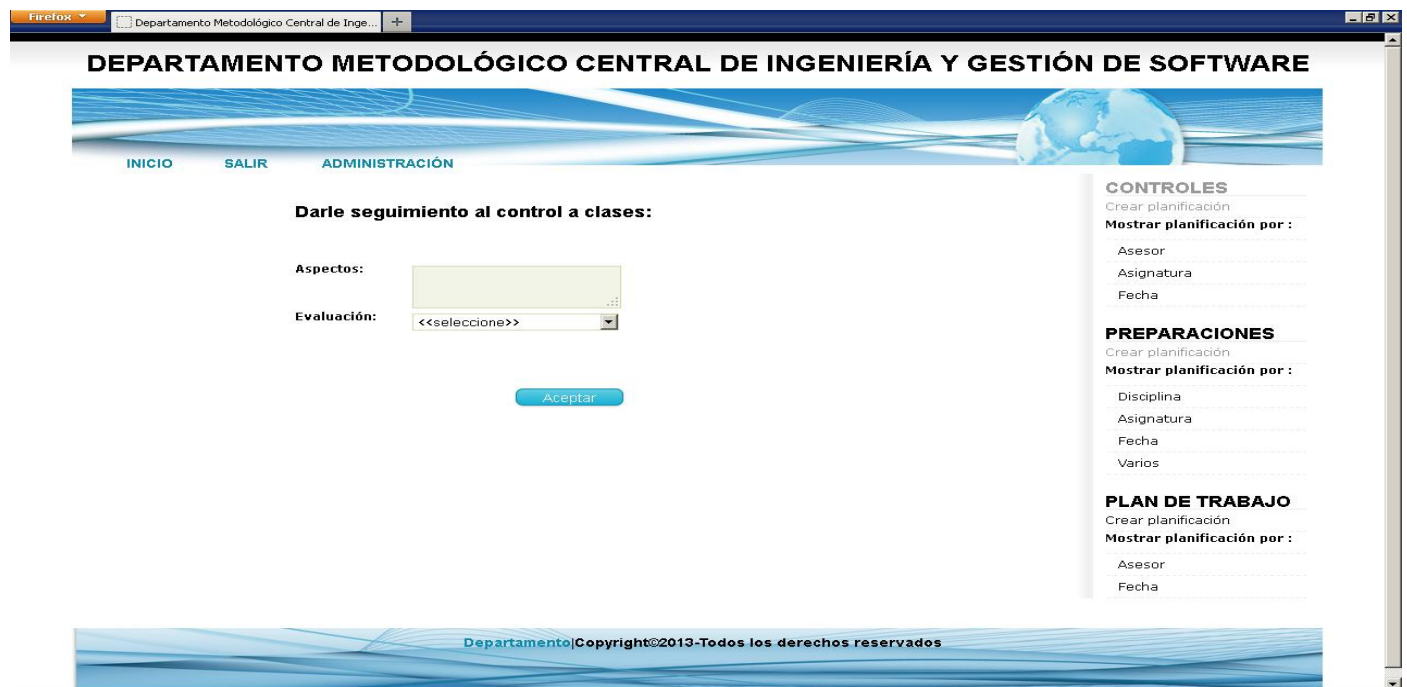


Ilustración 9: Prototipo de interfaz “Dar seguimiento al control a clases”

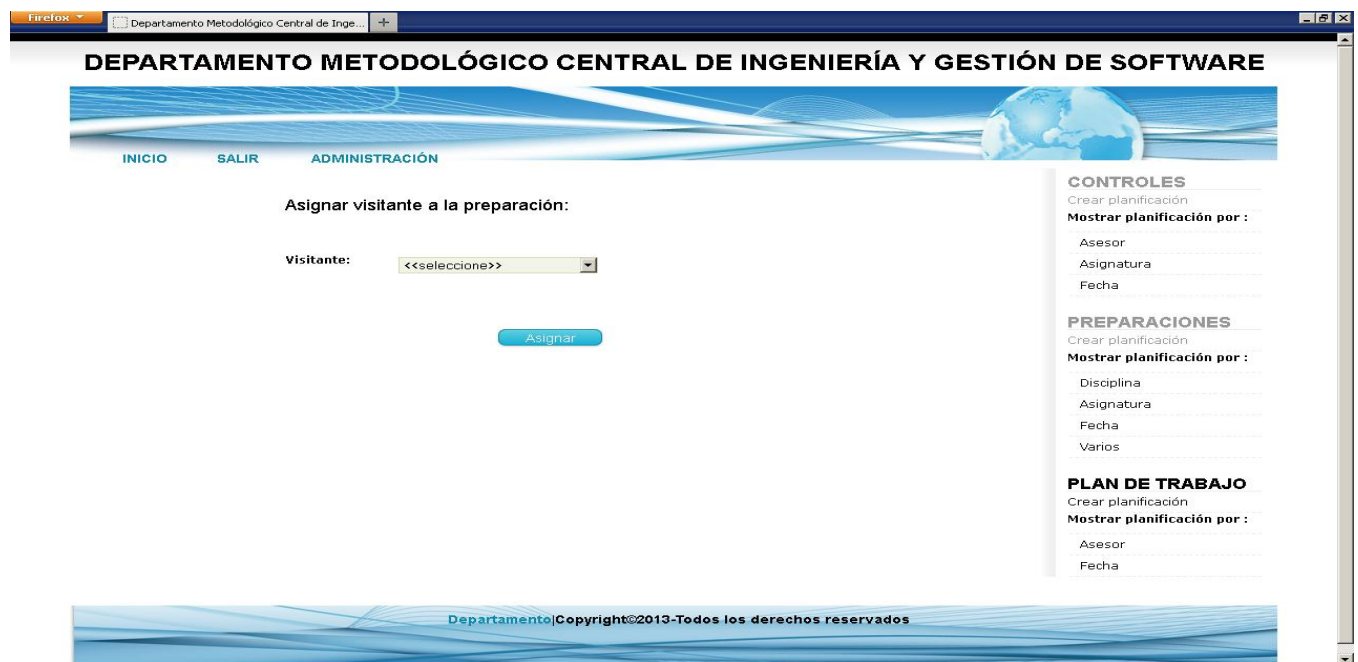


Ilustración 10: Prototipo de interfaz “Asignar visitante a la preparación metodológica”

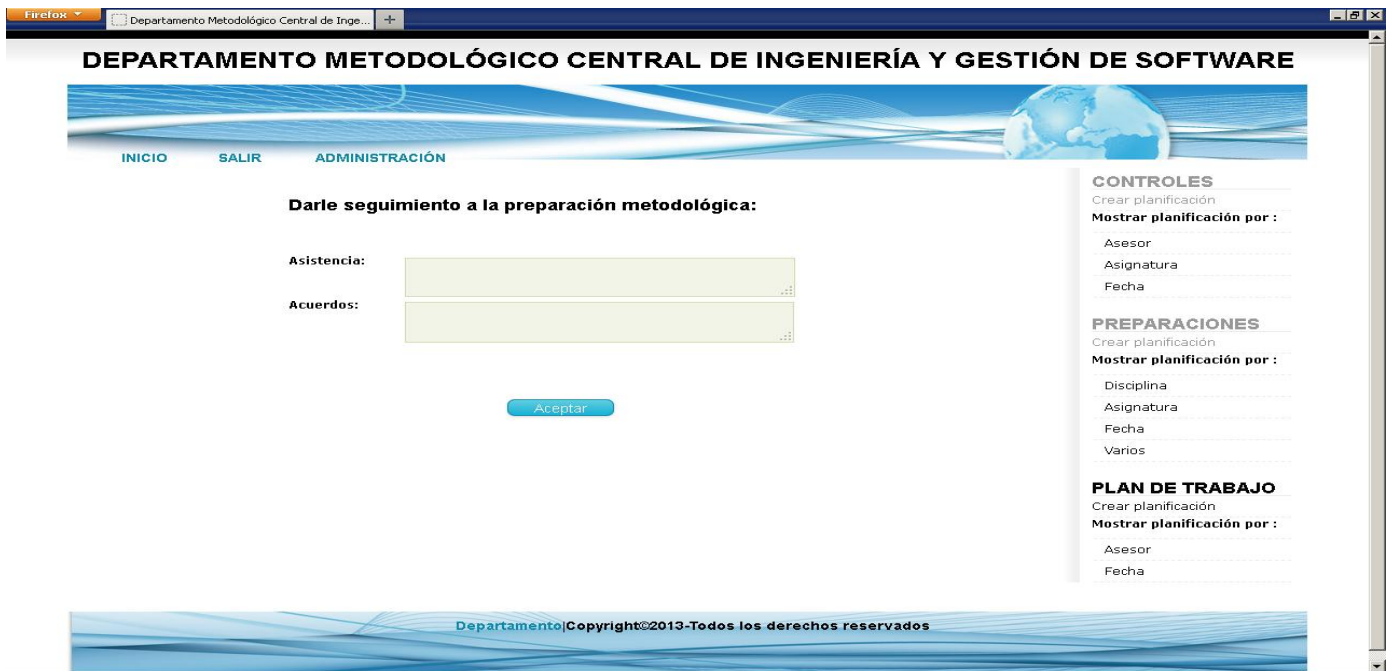


Ilustración 11: Prototipo de interfaz “Dar seguimiento a la preparación metodológica”

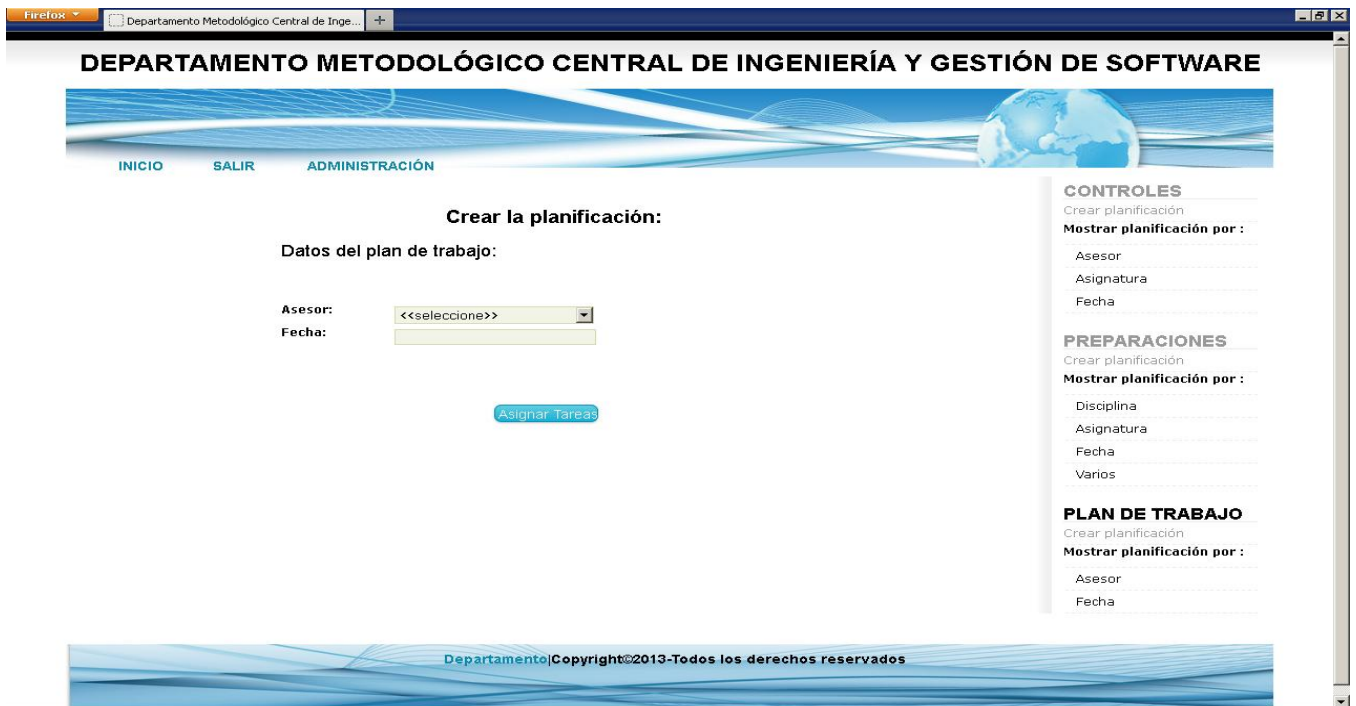


Ilustración 12: Prototipo de interfaz “Crear planificación de los planes de trabajo”

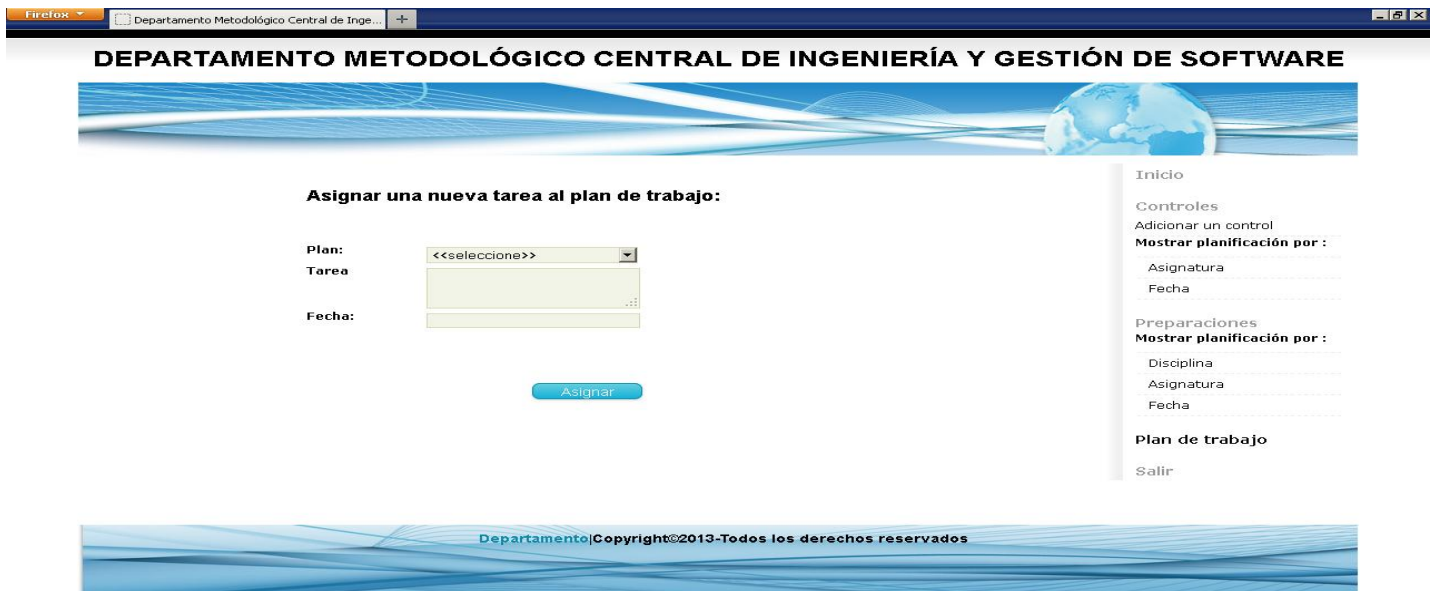


Ilustración 13: Prototipo de interfaz “Asignar tareas a un plan de trabajo”

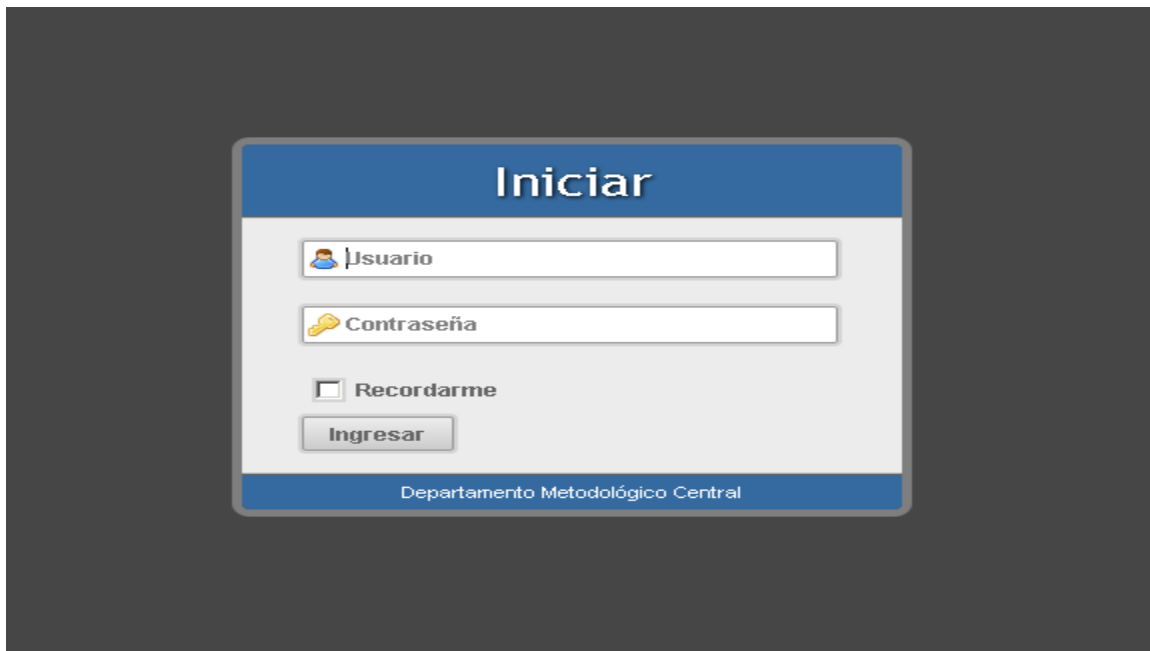


Ilustración 14: Prototipo de interfaz “Autenticar usuario”

## Anexo\_5: Tarjetas CRC

Tabla 60: Tarjeta CRC de facultad

Facultad	
Funcionalidades	Colaboración
nuevaFacultadAction ( )	
crearFacultadAction ( )	
listarFacultadAction ( )	
mostrarFacultadAction ( )	
editarFacultadAction ( )	
actualizarFacultadAction ( )	
eliminarFacultadAction ( )	

Tabla 61: Tarjeta CRC de local

Local	
Funcionalidades	Colaboración
crearLocalAction( )	public function crearFacultadAction (Request \$request)
nuevoLocalAction( )	
listarLocalAction( )	
mostrarLocalAction( )	
editarLocalAction( )	
actualizarLocalAction( )	
eliminarLocalAction( )	

Tabla 62: Tarjeta CRC de grupo

Grupo	
Funcionalidades	Colaboración
crearGrupoAction( )	public function crearFacultadAction(Request \$request)
nuevoGrupoAction( )	
listarGrupoAction( )	
mostrarGrupoAction( )	
editarGrupoAction( )	
actualizarGrupoAction( )	
eliminarGrupoAction( )	

Tabla 63: Tarjeta CRC de departamento

Departamento	
Funcionalidades	Colaboración
crearDepartamentoAction( )	
nuevoDepartamentoAction( )	
listarDepartamentoAction( )	
mostrarDepartamentoAction( )	

editarDepartamentoAction( )	
actualizarDepartamentoAction( )	
eliminarDepartamentoAction( )	

**Tabla 64: Tarjeta CRC de disciplina**

Disciplina	
Funcionalidades	Colaboración
crearDisciplinaAction( )	public function crearDepartamentoAction(Request \$request)
nuevaDisciplinaAction( )	
listarDisciplinaAction( )	
mostrarDisciplinaAction( )	
editarDisciplinaAction( )	
actualizarDisciplinaAction( )	
eliminarDisciplinaAction( )	

**Tabla 65: Tarjeta CRC de asignatura**

Asignatura	
Funcionalidades	Colaboración
crearAsignaturaAction( )	public function crearDisciplinaAction(Request \$request)
nuevaAsignaturaAction( )	
listarAsignaturaAction( )	
mostrarAsignaturaAction( )	
editarAsignaturaAction( )	
actualizarAsignaturaAction( )	
eliminarAsignaturaAction( )	

**Tabla 66: Tarjeta CRC de usuario**

Usuario	
Funcionalidades	Colaboración
crearUsuarioAction( )	
nuevoUsuarioAction( )	
listarUsuarioAction( )	
mostrarUsuarioAction( )	
editarUsuarioAction( )	
actualizarUsuarioAction( )	
eliminarUsuarioAction( )	

**Tabla 67: Tarjeta CRC de categoría docente**

Categoría	
Funcionalidades	Colaboración
crearCategoriaAction( )	

listarCategoriaAction( )	
mostrarCategoriaAction( )	
editarCategoriaAction( )	
actualizarCategoriaAction( )	
eliminarCategoriaAction( )	

**Tabla 68: Tarjeta CRC de profesor**

<b>Profesor</b>	
<b>Funcionalidades</b>	<b>Colaboración</b>
crearProfesorAction( )	public function crearFacultadAction(Request \$request)
nuevoProfesorAction( )	public function crearCategoriaAction(Request \$request)
listarProfesorAction( )	public function crearAsignaturaAction(Request \$request)
mostrarProfesorAction( )	
editarProfesorAction( )	
actualizarProfesorAction( )	
eliminarProfesorAction( )	

**Tabla 69: Tarjeta CRC de tarea**

<b>Tarea</b>	
<b>Funcionalidades</b>	<b>Colaboración</b>
crearTareaAction( )	public function crearPlanTrabajoMensualAction (Request \$request)
listarTareaAction( )	
mostrarTareaAction( )	
modificarTareaAction( )	
eliminarTareaAction( )	

## Anexo\_6: Casos de pruebas

Tabla 70: Caso de prueba “Crear departamento”

Escenario	Descripción	Variable 1 “Nombre del Departamento”	Respuesta del sistema	Flujo central
EC1.Crear departamento correctamente	Mediante este escenario se crea el departamento correctamente en el sistema.	V “Departamento”	El sistema crea el departamento correctamente.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Departamento”, el sistema muestra todos los datos de los departamentos, en caso de que existan, este da la opción de crear un nuevo departamento.  El usuario escoge la opción de “crear un nuevo departamento”.  El sistema muestra los campos a llenar por el usuario:  -nombre del departamento  El usuario llena el campo satisfactoriamente y presiona el botón “Crear”.  El sistema actualiza el listado y muestra los departamentos creados.
		V “Ingeniería gestión de software”		
		V “Nombre de 5 a 75 caracteres”		
EC2.El elemento existe.	Mediante este escenario se verifica que el departamento ya existe en el sistema.	I “Departamento”  V “Nombre de 5 a 75 caracteres”	El sistema muestra un mensaje de error “Este valor ya se ha utilizado” y no lo suscribe.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona

				<p>“Departamento”, el sistema muestra todos los datos de los departamentos, en caso de que existan, este da la opción de crear un nuevo departamento.</p> <p>El usuario escoge la opción de “crear un nuevo departamento”.</p> <p>El sistema muestra el campo a llenar por el usuario:</p> <p>-nombre del departamento</p> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón “Crear”.</p> <p>El sistema muestra un error “El departamento ya existe” y no actualiza la lista.</p>
EC3. Insertar datos incompletos	Mediante este escenario no se introducen todos los datos para crear un departamento.	/ “Vacío”	El sistema muestra el mensaje “Por favor, rellene los campos” y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Departamento”, el



EC4. Insertar datos incorrectos.	Mediante este escenario no se permite introducir datos incorrectos.	<p>I "@#departamento"</p> <p>V "Nombre de 5 a 75 caracteres"</p>	<p>El sistema muestra un mensaje de error "Este campo acepta solo letras. El sistema no permite seguir introduciendo caracteres."</p>	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", el sistema muestra todos los datos de los departamentos, en caso de que existan.</p> <p>El usuario escoge la opción de "crear un nuevo departamento".</p> <p>El sistema muestra el campo a llenar por el usuario: -nombre del departamento.</p> <p>El usuario llena incorrectamente el campo y presiona el botón "Crear".</p> <p>El sistema muestra un mensaje de error "Este campo acepta solo letras".</p>
		<p>I "Nombre de 5 a 75 caracteres"</p>		
		<p>V "Departamento"</p>		

**Tabla 71: Caso de prueba "Modificar departamento"**

Escenario	Descripción	Variable 1 "Nombre del departamento"	Respuesta del sistema	Flujo central
EC1. Modificar departamento correctamente	Mediante este escenario se modifica el departamento correctamente en el sistema.	<p>V "Departamento"</p> <p>V "Nombre de 5 a 75 caracteres"</p>	El sistema permite modificar el departamento correctamente y actualiza la lista de los departamentos.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", el sistema muestra todos los datos de los

				<p>departamentos, en caso de que existan, este da la opción de modificar departamento.</p> <p>El usuario escoge la opción de "Editar departamento".</p> <p>El sistema muestra los campos a actualizar por el usuario:</p> <p>-nombre del departamento</p> <p>El usuario llena el campo satisfactoriamente y presiona el botón "Editar".</p> <p>El sistema modificar el departamento correctamente y actualiza la lista de los departamentos.</p>
EC2. El elemento existe.	Mediante este escenario se verifica que el departamento ya existe en el sistema.	<p>I "Departamento"</p> <p>V "Nombre de 5 a 75 caracteres"</p>	El sistema muestra el mensaje "Este valor ya se ha utilizado" y no actualiza la lista de los departamentos.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", el sistema muestra todos los datos de los departamentos, en caso de que existan.</p>
				<p>El usuario escoge la opción de "Editar departamento".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre del departamento.</p> <p>El usuario llena el campo y presiona el botón "Editar".</p> <p>El sistema muestra el mensaje "Este valor ya se ha utilizado" y no actualiza la lista de los departamentos.</p>
EC3. Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar un departamento.	I "Vacío"	El sistema muestra el mensaje "Por favor rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", y el sistema muestra todos los datos de los departamentos, en caso de que existan.</p> <p>El usuario escoge la opción de "Editar un departamento".</p>
				<p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre del departamento.</p>

				El usuario no llena el campo y presiona el botón "Editar". El sistema muestra el mensaje "Por favor, rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.
EC4. Insertar datos incorrectos.	Mediante este escenario se permite no introducir datos incorrectos.	I "@#departamento"	El sistema muestra un mensaje de error "Solo se aceptan letras".	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", el sistema muestra todos los datos de los departamentos, en caso de que existan.  El usuario escoge la opción de "Editar departamento".  El sistema muestra los campos a llenar por el usuario: -nombre del departamento. El usuario llena el campo con datos incorrectos y presiona el botón "Editar". El sistema muestra un mensaje de error "Solo se acepta letras".
		V "Nombre de 5 a 75 caracteres"		
		V "Departamento"		

**Tabla 72: Caso de prueba "Eliminar departamento"**

Escenario	Descripción	Variable 1 "Seleccionar el Departamento"	Respuesta del sistema	Flujo central
EC1. Eliminar departamento correctamente.	Mediante este escenario se elimina el departamento correctamente en el sistema.	V "Seleccionar departamento"	El sistema elimina correctamente el departamento y actualiza la lista de los departamentos.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Departamento", y el sistema muestra todos los datos de los departamentos, en caso de que existan.  El usuario escoge la opción de "Eliminar"

				<p>departamento”.</p> <p>El sistema muestra el departamento que desea eliminar.</p> <p>El usuario selecciona el botón “Eliminar” y el sistema elimina correctamente el departamento y actualiza la lista de los departamentos.</p>
--	--	--	--	--

**Tabla 73: Caso de prueba “Crear facultad”**

Escenario	Descripción	Variable 1 “Nombre de la facultad”	Respuesta del sistema	Flujo central
EC1.Insertar los datos correctamente.	Mediante este escenario se crea una facultad correctamente en el sistema.	<p>V “facultad”</p> <p>V “Nombre de 5 a 25 caracteres”</p>	El sistema crea correctamente la facultad y muestra al usuario los datos creados.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Facultad”, y el sistema muestra todos los datos de las facultades, en caso de que existan.</p> <p>El usuario escoge la opción de “Crear una nueva facultad”.</p> <p>El sistema muestra los campos a llenar por el usuario: -nombre de la facultad El usuario llena el campo satisfactoriamente y presiona el botón “Crear”.</p> <p>El sistema actualiza el listado de las facultades.</p>
EC2.El elemento existe.	Mediante este escenario se verifica que la facultad ya	I “facultad”	El sistema muestra un mensaje de error “El elemento	El usuario una vez autenticado con los privilegios otorgados selecciona

	existe en el sistema.		ya existe" y no lo suscribe.	"Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en caso de que existan.
		V "Nombre de 5 a 25 caracteres"		El usuario escoge la opción de "Crear una nueva facultad".  El sistema muestra los campos a llenar por el usuario: -nombre de la facultad El usuario llena el campo satisfactoriamente y presiona el botón "Crear". El sistema muestra un error "El elemento existe" y este no actualiza la lista de facultades.
EC3.Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para insertar una facultad en la aplicación.	I "Vacío"	El sistema muestra el mensaje "Por favor rellene este campo" y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en caso de que existan.
				El usuario escoge la opción de "Crear una nueva facultad".  El sistema muestra los campos a llenar por el usuario: -nombre de la facultad El usuario no llena el campo y presiona el botón "Crear". El sistema muestra un mensaje "Por favor,

				rellene este campo” y se muestra en rojo el campo que debe ser llenado obligatorio.
EC4.Insertar datos incorrectos.	Mediante este escenario se permite no introducir datos incorrectos.	I “@#facultad”	El sistema marca el error en rojo y muestra un mensaje "Solo se aceptan letras y números".	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Facultad”, y el sistema muestra todos los datos de las facultades, en caso de que existan.  El usuario escoge la opción de “Crear una nueva facultad”.  El sistema muestra los campos a llenar por el usuario: -nombre de la facultad El usuario inserta datos incorrectos y presiona el botón “Crear”. El sistema marca el error en rojo y muestra un mensaje” Solo se aceptan letras y números”.
		V “Nombre de hasta 25 caracteres”		
		V “Facultad”		
		I “Nombre de 5 hasta 25 caracteres” ”	El sistema no permite seguir introduciendo caracteres.	

**Tabla 74: Caso de prueba “Modificar facultad”**

<b>Escenario</b>	<b>Descripción</b>	<b>Variable 1 “Nombre de la facultad”</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC1.Modificar facultad correctamente	Mediante este escenario se modifica la facultad correctamente en el sistema.	V “Facultad”  V “Nombre de hasta 25 caracteres”	El sistema modifica correctamente la facultad.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Facultad”, y el sistema muestra todos los datos de

				<p>las facultades, en caso de que existan.</p> <p>El usuario escoge la opción de "Editar facultad".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre de la facultad</p> <p>El usuario llena el campo satisfactoriamente y presiona el botón "Editar".</p> <p>El sistema modifica correctamente la facultad y actualiza el listado de las facultades.</p>
EC2.El elemento existe.	Mediante este escenario se verifica que la facultad ya existe en el sistema.	I "Facultad"	El sistema muestra un mensaje de error "Este valor ya se ha utilizado", y no lo suscribe.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en caso de que existan.
		V "Nombre de hasta 25 caracteres"		
				<p>El usuario escoge la opción de "Editar facultad".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre de la facultad</p> <p>El usuario llena el campo y presiona el botón "Editar".</p> <p>El sistema muestra un mensaje de error "Este valor ya se ha utilizado", y no lo</p>

				suscribe.
EC3. Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar una facultad.	I "Vacío"	El sistema muestra el mensaje "Por favor rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en caso de que existan.  El usuario escoge la opción de "Editar facultad".  El sistema muestra los campos a llenar por el usuario:  -nombre de la facultad El usuario no llena el campo y presiona el botón "Editar". El sistema muestra el mensaje "Por favor, rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.
		I "Nombre de 3 hasta 25 caracteres"	El sistema muestra el mensaje "Este valor es demasiado corto. Debería tener 3 caracteres o más".	
EC4. Insertar datos incorrectos.	Mediante este escenario se permite no introducir datos incorrectos.	I "@#facultad"	El sistema muestra un mensaje de error "Solo se aceptan letras y números".	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en
		V "Nombre de hasta 25 caracteres"		



		V "Facultad"		<p>caso de que existan.</p> <p>El usuario escoge la opción de "Editar facultad".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre de la facultad</p> <p>El usuario llena incorrectamente el campo y presiona el botón "Editar".</p> <p>El sistema muestra un mensaje de error "Solo se aceptan letras y números".</p> <p>El sistema no procede con la actualización de la base de datos.</p>
--	--	-----------------	--	--

**Tabla 75: Caso de prueba "Eliminar facultad"**

<b>Escenario</b>	<b>Descripción</b>	<b>Variable 1 "Seleccionar la facultad"</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC1.Eliminar facultad correctamente	Mediante este escenario se elimina la facultad correctamente en el sistema.	V "facultad"	El sistema elimina correctamente la facultad y actualiza el listado de las facultades.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Facultad", y el sistema muestra todos los datos de las facultades, en caso de que existan.
				<p>El usuario escoge la opción de "Eliminar facultad".</p> <p>El sistema muestra</p>

				<p>los datos de la facultad que desea eliminar.</p> <p>El usuario selecciona el botón "Eliminar" y el sistema elimina correctamente la facultad y actualiza el listado de las facultades.</p>
--	--	--	--	---

**Tabla 76: Caso de prueba "Modificar grupo"**

Escenario	Descripción	Variable 1 "Nombre del Grupo"	Variable 2 "facultad"	Respuesta del sistema	Flujo central
EC1.Modificar un grupo correctamente.	Mediante este escenario se modifica un grupo correctamente en el sistema.	V "Grupo"	V "Facultad 1"	El sistema modifica correctamente el grupo.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Grupo", y el sistema muestra todos los datos de los grupos, en caso de que existan.
		V "Nombre de 25 caracteres"	V "Facultad 1"		
					<p>El usuario escoge la opción de "Editar grupo".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- facultad</li> <li>- nombre del grupo</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón</p>

					<p>“Editar”.</p> <p>El sistema modifica correctamente el grupo.</p>
EC2.Dejar campos vacíos.	Mediante este escenario no se introducen todos los datos para modificar un grupo.	I “Vacío”	I “Vacío”	El sistema muestra el mensaje “Por favor, rellene los campos” y se muestra en rojo el campo que debe ser llenado obligatorio.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona “Grupo”, y el sistema muestra todos los datos de los grupos, en caso de que existan.</p> <p>El usuario escoge la opción de “Editar grupo”.</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- facultad</li> <li>- nombre del grupo</li> </ul> <p>El usuario no llena todos los campos y presiona el botón “Editar”.</p> <p>El sistema muestra un error “Por favor, rellene los campos”.</p>
		V “Grupo”	I “Vacío”		
		I “Vacío”	V “Facultad”		

EC3.El elemento existe.	Mediante este escenario se verifica que el grupo ya existe en el sistema.	I "Grupo"	I "Facultad"	El sistema muestra un mensaje de error "El elemento ya existe", y no lo suscribe.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú principal, dentro de ese módulo selecciona "Grupo", y el sistema muestra todos los datos de los grupos, en caso de que existan.  El usuario escoge la opción de "Editar grupo".  El sistema muestra los campos a llenar por el usuario: - facultad - nombre del grupo  El usuario llena todos los campos y presiona el botón "Editar".  El sistema muestra un mensaje de error "El elemento ya existe" y no lo suscribe.
EC4. Insertar datos incorrectos.	Mediante este escenario se permite no introducir datos incorrectos.	I "@#Grupo"	V "Facultad1"	El sistema muestra un mensaje de error "Solo se aceptan letras y números".	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración" en el menú
		V "Grupo"	NA		
		I	V		

		“Nombre de 25 caracteres”	“Facultad1”	El sistema no permite seguir introduciendo caracteres.	principal, dentro de ese módulo selecciona “Grupo”, y el sistema muestra todos los datos de los grupos, en caso de que existan.  El usuario escoge la opción de “Editar grupo”. El sistema muestra los campos a llenar por el usuario: - facultad - nombre del grupo El usuario llena el campo incorrectamente y presiona el botón “Editar”.  El sistema muestra un mensaje de error "Solo se aceptan letras y números"
		V “Nombre de 25 caracteres”	NA		

**Tabla 77: Caso de prueba “Eliminar grupo”**

<b>Escenario</b>	<b>Descripción</b>	<b>Variable 1 “Nombre del grupo”</b>	<b>Variable 2 “Facultad”</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC1. Eliminar grupo correctamente.	Mediante este escenario se elimina un grupo correctamente en el sistema.	V “Grupo”	V “Facultad”	El sistema elimina correctamente el grupo y actualiza la lista de los grupos.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración” en el menú principal, dentro de ese módulo selecciona

					<p>“Grupo”, y el sistema muestra todos los datos de los grupos, en caso de que existan.</p> <p>El usuario escoge la opción de “Eliminar grupo”.</p> <p>El sistema muestra los datos del grupo que desea eliminar.</p> <p>El sistema elimina correctamente el grupo y actualiza la lista de los grupos.</p>
--	--	--	--	--	--

**Tabla 78: Caso de prueba “Crear local”**

<b>Escenario</b>	<b>Descripción</b>	<b>Variable 1 “Local”</b>	<b>Variable 2 “facultad”</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC1.Crear local correctamente.	Mediante este escenario se crea un local correctamente en el sistema.	V “Local”	V “Facultad 1”	El sistema crea correctamente el local.	El usuario una vez autenticado con los privilegios otorgados selecciona “Local” en el menú principal, el sistema muestra todos los datos de los locales en caso de que existan.
		V “Nombre de3 hasta 25 caracteres”	V “Facultad 1”		
					<p>El usuario escoge la opción de “Crear un nuevo local”.</p> <p>El sistema muestra los campos a llenar por el usuario:</p>

					<p>- facultad</p> <p>-nombre del local</p> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear".</p> <p>El sistema actualiza el listado.</p>
EC 2. El elemento existe.	Mediante este escenario se verifica que el local ya existe en el sistema.	I "Local"	I "Facultad 1"	El sistema muestra un mensaje de error "El elemento ya existe" y no lo suscribe.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "local" en el menú principal, el sistema muestra todos los datos de los locales, en caso de que existan.</p> <p>El usuario escoge la opción de "Crear un nuevo local".</p>
		V "Local"	I "Facultad 1"		
		I "Local"	V "Facultad 1"		
					<p>El sistema muestra los campos a llenar por el usuario:</p> <p>- facultad</p> <p>-nombre del local.</p> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear".</p> <p>El sistema muestra un error "El elemento ya existe" y no actualiza la lista de los locales.</p>
EC3.Campos vacíos.	Mediante este escenario no se introducen todos	I "Vacío"	I "Vacío"	El sistema muestra el mensaje "Por	El usuario una vez autenticado con los privilegios

	los datos para crear un local.	I "Local"	V "Vacío"	favor, rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio	otorgados selecciona "Local" en el menú principal, el sistema muestra todos los datos de los locales en caso de que exista.
		I "Vacío"	V "Facultad"		El usuario escoge la opción de "Crear un nuevo local"  El sistema muestra los campos a llenar por el usuario: - facultad - nombre del local  El usuario no llena todos los campos y presiona el botón "Crear".  El sistema muestra un error "Por favor, rellene los campos".
EC4.Insertar datos incorrectos	Mediante este escenario se permite no introducir datos incorrectos.	I "Local\$%"	V "Facultad"	El sistema marca el error en rojo y muestra un mensaje "Solo se aceptan letras y números".	El usuario una vez autenticado con los privilegios otorgados selecciona "Local" en el menú principal, el sistema muestra todos los datos de los locales en caso de que exista.
		V "Local"	NA		El usuario escoge la opción de "Crear un nuevo local".
		I "Nombre de 3 hasta 25 caracteres"	V "Facultad"	El sistema no permite seguir introduciendo caracteres.	El sistema muestra los campos a llenar por el usuario: - facultad



					<p>-nombre del local</p> <p>El usuario inserta los datos incorrectos y presiona el botón "Crear".</p> <p>El sistema marca el error en rojo y muestra un mensaje "Solo se aceptan letras y números".</p>
--	--	--	--	--	---

**Tabla 79: Caso de prueba "Modificar local"**

Escenario	Descripción	Variable 1 "Local"	Variable 2 "facultad"	Respuesta del sistema	Flujo central
EC1.Modificar un local correctamente.	Mediante este escenario se modifica un local correctamente en el sistema.	V "Local"	V "Facultad 1"	El sistema modifica correctamente el local.	El usuario una vez autenticado con los privilegios otorgados selecciona "Local" en el menú principal, el sistema muestra todos los datos de los locales en caso de que existan.
		V "Nombre de 25 caracteres"	V "Facultad 1"		
					El usuario escoge la opción de "Editar local". El sistema muestra los campos a llenar por el usuario: - facultad - nombre del local El usuario llena todos los campos satisfactoriamente y presiona el botón "Editar". El sistema modifica el listado de los

					locales.
EC2. Dejar campos vacíos.	Mediante este escenario no se introducen todos los datos para modificar un local.	I "Vacío"	I "Vacío"	El sistema muestra el mensaje "Por favor, rellene los campos" y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Local", el sistema muestra todos los datos de los locales, en caso de que existan.  El usuario escoge la opción de "Editar local".  El sistema muestra los campos a llenar por el usuario: - facultad - nombre del local  El usuario no llena todos los campos y presiona el botón "Editar".  El sistema muestra un error "Por favor, rellene los campos".
		I "Local"	V "Facultad"		
		V "Local"	I "Vacío"		

**Tabla 80: Caso de prueba "Eliminar local"**

Escenario	Descripción	Variable 1 "Local"	Variable 2 "Facultad"	Respuesta del sistema	Flujo central
EC1. Eliminar local correctamente	Mediante este escenario se elimina un local correctamente en el sistema.	V "Local"	V "Facultad"	El sistema elimina correctamente el local y actualiza el listado de los locales.	El usuario una vez autenticado con los privilegios otorgados selecciona "Local" en el menú principal, el sistema

					<p>muestra todos los datos de los locales en caso de que existan.</p> <p>El usuario escoge la opción de "Eliminar local".</p> <p>El sistema muestra los datos del local que desea eliminar, y el usuario elimina el local.</p>
--	--	--	--	--	--

**Tabla 81: Caso de prueba "Crear Disciplina"**

Escenario	Descripción	Variable 1 "disciplina"	Variable 1 "departamento"	Respuesta del sistema	Flujo central
EC1. Insertar los datos correctamente.	Mediante este escenario se inserta una disciplina correctamente al sistema.	V "disciplina"	V "departamento"	El sistema inserta correctamente la disciplina.	El usuario una vez autenticado con los privilegios otorgados selecciona "disciplina" en el menú principal, el sistema muestra todos los datos de la disciplina en caso de que existan.
		V "BaseD1"	V "IGSW"		
		V "Nombre de 3 hasta 25 caracteres"	V "departamento"		
					<p>El usuario escoge la opción de "Crear una nueva disciplina".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>-nombre de la disciplina.</li> <li>-nombre del departamento.</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el</p>

					<p>botón "Crear".</p> <p>El sistema actualiza el listado de las disciplinas.</p>
EC2.El elemento existe.	Mediante este escenario se verifica que la disciplina ya existe en el sistema.	I "disciplina"	I "departamento"	El sistema muestra un mensaje de error "El elemento ya existe" y no lo suscribe.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "disciplina" en el menú principal, el sistema muestra todos los datos de las disciplinas en caso de que existan.</p> <p>El usuario escoge la opción de "Crear una disciplina".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>-nombre de la disciplina</li> <li>-nombre del departamento</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear".</p> <p>El sistema muestra un error "El elemento ya existe" y este no actualiza el listado de las disciplinas.</p>
		V "BaseD1"	I "IGSW"		
		I "BaseD1"	V "IGSW"		
		V "Nombre de 3 hasta 25 caracteres"	I "departamento"		
EC3.Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para insertar una disciplina en la	I "Vacío"	I "Vacío"	El sistema muestra el mensaje "Por favor, rellene este campo" y se muestra en rojo	El usuario una vez autenticado con los privilegios otorgados selecciona "disciplina" en el menú principal, el
		V "BaseD1"	I "Vacío"		

	aplicación.	I "Vacío"	V "Departamento"	el campo que debe ser llenado obligatorio.	sistema muestra todos los datos de las disciplinas en caso de que existan.
		I "Nombre de 3 hasta 25 caracteres"	V "Departamento"	El sistema muestra el mensaje "Este valor es demasiado corto. Debería tener 3 caracteres o más".	El usuario escoge la opción de "Crear una nueva disciplina". El sistema muestra los campos a llenar por el usuario: -nombre de la disciplina. -nombre del departamento.
					El usuario no llena el campo y presiona el botón "Crear". El sistema muestra un mensaje "Por favor, rellene este campo" y se muestra en rojo el campo que debe ser llenado obligatorio.
EC4.Insertar datos incorrectos.	Mediante este escenario no se permite introducir datos incorrectos.	I "Progra\$%"	V "Departamento"	El sistema marca el error en rojo y muestra un mensaje "Solo se aceptan letras y números".	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Disciplina" el sistema muestra todos los datos de las disciplinas en caso de que existan.
		I "Base dato234"	V "Departamento"	El sistema no permite seguir introduciendo caracteres.	
		I "Nombre de 3 hasta 25 caracteres"	V "Departamento"		
					El usuario escoge la opción de "Crear una nueva

					<p>disciplina”.</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- nombre de la disciplina.</li> <li>- nombre del departamento.</li> </ul> <p>El usuario no llena el campo y presiona el botón “Crear”.</p> <p>El sistema marca el error en rojo y muestra un mensaje "Solo se aceptan letras y números".</p>
--	--	--	--	--	---

**Tabla 82: Caso de prueba “Eliminar disciplina”**

Escenario	Descripción	Variable 1 “disciplina”	Variable 2 “departamento”	Respuesta del sistema	Flujo central
EC1.Eliminar disciplina correctamente.	Mediante este escenario se elimina la disciplina correctamente en el sistema.	V “disciplina”	V “departamento”	El sistema elimina la disciplina correctamente.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Disciplina" el sistema muestra todos los datos de las disciplinas en caso de que existan.
		V “Programación”	V “departamentoProgramción”		
					El usuario escoge la opción de “Eliminar disciplina”.

				El sistema muestra los datos de la disciplina que desea eliminar el usuario selecciona el botón "Eliminar" y se elimina la disciplina del sistema y se actualiza la base dato.
--	--	--	--	--

**Tabla 83: Caso de prueba "Crear asignatura"**

Escenario	Descripción	Variable 1 "Nombre de la Asignatura"	Variable 2 "Nombre de la disciplina"	Respuesta del sistema	Flujo central
EC1. Insertar los datos correctamente.	Mediante este escenario se crea una asignatura correctamente al sistema.	V "Asignatura"	V "Disciplina"	El sistema crea la asignatura.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Asignatura" en el menú principal, el sistema te muestra todos los datos de las asignaturas en caso de que existan.  El usuario escoge la opción de "Crear una nueva asignaturas".  El sistema muestra los campos a llenar por el usuario: - nombre de la asignatura -disciplina  El usuario llena todos los campos satisfactoriamente y presiona el botón
		V "Nombre de hasta 20 caracteres"	V "Nombre de hasta 20 caracteres"		

					<p>“Crear”.</p> <p>El sistema actualiza el listado de las asignaturas.</p>
EC2.El elemento existe.	Mediante este escenario se verifica que la asignatura ya existe en el sistema.	I “Asignatura”	I “Disciplina”	El sistema muestra un mensaje de error “El elemento ya existe” y no lo suscribe.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona “Administración”, ya dentro del módulo escoge “Asignatura” en el menú principal, el sistema te muestra todos los datos de las asignaturas en caso de que existan.</p> <p>El usuario escoge la opción de “Crear una asignatura”.</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>-nombre de la asignatura.</li> <li>-disciplina</li> </ul> <p>El usuario no llena todos los campos satisfactoriamente y presiona el botón “Crear”.</p> <p>El sistema muestra un error “El elemento ya existe”.</p>
		I “Base datos”	V “Ingeniería”		
		V “Base datos”	I “Ingeniería”		
EC3.Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear una asignatura en la aplicación.	I “Vacío”	I “Vacío”	El sistema muestra el mensaje “Por favor, rellene este campo” y se muestra en rojo el campo que debe ser llenado obligatorio.	El usuario una vez autenticado con los privilegios otorgados selecciona “Administración”, ya dentro del módulo escoge “Asignatura” en el
		V “Matemática”	I “Vacío”		



I  
"Vacío"

v  
"Disciplina"

menú principal, el sistema te muestra todos los datos de las asignaturas en caso de que existan.

El usuario escoge la opción de "Crear una asignatura".

El sistema muestra los campos a llenar por el usuario:

-nombre de la asignatura.

-disciplina

El usuario no llena todos los campos satisfactoriamente y presiona el botón "Crear".

El sistema muestra un mensaje "Por favor, rellene estos campos" y se muestra en rojo los campos que deben ser llenados obligatorio.

Tabla 84: Caso de prueba “Modificar asignatura”

Escenario	Descripción	Variable 1 “Nombre de la asignatura”	Variable 2 “Nombre de la disciplina”	Respuesta del sistema	Flujo central
EC1.Modificar asignatura correctamente	Mediante este escenario se modifica la asignatura correctamente en el sistema.	V “Asignatura”	V “Disciplina”	El sistema modifica correctamente la asignatura.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Asignatura" en el menú principal, el sistema te muestra todos los datos de las asignaturas en caso de que existan.</p> <p>El usuario escoge la opción de "Crear una asignatura".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>-nombre de la asignatura.</li> <li>-disciplina</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Editar".</p> <p>El sistema modifica el listado de las asignaturas.</p>
		V “Física”	V “Ingeniería”		
		V “Nombre de 25 caracteres”	V “Ingeniería”		
EC2.Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar una asignatura.	I “Vacío”	I “Vacío”	El sistema muestra el mensaje “Por favor, rellene los campos” y se muestra en rojo el campo que debe ser llenado obligatorio.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Asignatura" en el menú principal, el sistema te muestra todos los datos de las asignaturas en</p>
		V “Física”	I “Vacío”		
		I “Vacío”	V		

			"Ingeniería"		<p>caso de que existan.</p> <p>El usuario escoge la opción de "Crear una asignatura".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <p>-nombre de la asignatura.</p> <p>-disciplina</p> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Editar".</p> <p>El sistema muestra un error "Por favor, rellene los campos".</p>
--	--	--	--------------	--	---

**Tabla 85: Caso de prueba "Eliminar asignatura"**

Escenario	Descripción	Variable 1 "Seleccionar la Asignatura"	Variable 1 "Nombre de la Disciplina"	Respuesta del sistema	Flujo central
EC1.Eliminar asignatura correctamente	Mediante este escenario se elimina la asignatura correctamente en el sistema.	V " asignatura"	V "Disciplina"	El sistema elimina correctamente la asignatura y actualiza el listado de las asignaturas.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Asignatura" en el menú principal, el sistema te muestra todos los datos de las asignaturas en caso de que existan.</p> <p>El usuario escoge la opción de "Eliminar asignatura".</p> <p>El sistema muestra los datos de la asignatura que desea eliminar.</p> <p>El usuario selecciona el botón "Eliminar" y</p>
		V "Español"	V "Disciplina"		

					se elimina esta asignatura.
--	--	--	--	--	-----------------------------

Tabla 86: Caso de prueba “Crear categoría docente”

Escenario	Descripción	Variable 1 “categoría docente”	Respuesta del sistema	Flujo central
EC1. Insertar los datos correctamente.	Mediante este escenario se crea una categoría docente correctamente al sistema.	V “Categoría Docente”	El sistema inserta correctamente la categoría docente.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Categoría docente" en el menú principal, el sistema las muestra en el caso en que existan.</p> <p>El usuario escoge la opción de "Crear una nueva categoría docente".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- nombre de la categoría docente</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear":</p> <p>El sistema actualiza el listado y muestra los datos anteriormente entrados por el usuario.</p>
		V “Nombre de hasta 25 caracteres”		
EC2. El elemento existe.	Mediante este escenario se verifica que la categoría docente ya existe en el sistema.	I “Categoría Docente”	El sistema muestra un mensaje de error “El elemento ya existe” y no lo suscribe.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Categoría docente" en el menú principal, el sistema las muestra en el caso en que existan.</p> <p>El usuario escoge la opción de "Crear una nueva categoría docente".</p> <p>El sistema muestra los</p>
		V “Nombre de hasta 25 caracteres”		

				<p>campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- nombre de la categoría docente</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Crear":</p> <p>El sistema muestra un error "El elemento ya existe".</p>
--	--	--	--	--

**Tabla 87: Caso de prueba "Modificar categoría docente"**

Escenario	Descripción	Variable 1 "categoría docente"	Respuesta del sistema	Flujo central
EC1.Modificar categoría docente correctamente.	Mediante este escenario se modifica la categoría docente correctamente en el sistema.	<p>V "categoría docente"</p> <p>V "Nombre hasta 25 caracteres"</p>	El sistema modifica correctamente la categoría docente y muestra.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Categoría docente" en el menú principal, el sistema las muestra en el caso en que existan.</p> <p>El usuario escoge la opción de "Editar categoría docente".</p> <p>El sistema muestra los campos a llenar por el usuario:</p> <ul style="list-style-type: none"> <li>- nombre de la categoría docente</li> </ul> <p>El usuario llena todos los campos satisfactoriamente y presiona el botón "Editar".</p> <p>El sistema modifica el listado de las categorías docentes.</p>
EC2.Dejar campos vacíos.	Mediante este escenario no se introducen todos los datos para modificar	I "Vacío"	El sistema muestra el mensaje "Por favor, rellene los campos" y se muestra en rojo el campo que debe ser	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya

	una categoría docente.		llenado obligatorio.	dentro del módulo escoge "Categoría docente" en el menú principal, el sistema las muestra en el caso en que existan.  El usuario escoge la opción de "Editar categoría docente".  El sistema muestra los campos a llenar por el usuario: - nombre de la categoría docente	
		V "Nombre hasta 25 caracteres"			El sistema no permite seguir introduciendo caracteres.
		V "Instructor"			
		I "Nombre hasta 25 caracteres"			

**Tabla 88: Caso de prueba "Eliminar categoría docente"**

Escenario	Descripción	Variable 1 "Nombre de la Categoría docente"	Respuesta del sistema	Flujo central
EC1.Eliminar categoría docente correctamente.	Mediante este escenario se elimina la categoría docente correctamente en el sistema.	V "Categoría docente"	El sistema elimina correctamente la categoría docente.	El usuario una vez autenticado con los privilegios otorgados selecciona "Administración", ya dentro del módulo escoge "Categoría docente" en el menú principal, el sistema las muestra en el caso en que existan.  El usuario escoge la opción de "Eliminar
		V "Nombre hasta 25 caracteres"		

				<p>categoria docente”.</p> <p>El sistema muestra los datos de la categoría docente que desea eliminar, el usuario selecciona el botón “Eliminar” y se elimina la categoría docente del sistema y se actualiza la base de datos.</p>
--	--	--	--	---

**Tabla 89: Caso de prueba “Criterio de búsqueda de los controles a clases por asesor”**

Escenario	Descripción	Variable 1 “Asesores”	Respuesta del sistema	Flujo central
EC1. Buscar controles a clases por asesor correctamente	Mediante este escenario se puede buscar los controles a clase por asesor en el sistema.	V “Asesor”	El sistema busca los controles a clases por asesor y muestra el resultado de la búsqueda al usuario.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona “Reportes” en el menú principal, el sistema muestra los tipos de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda de los controles a clases por asesores.</p> <p>El sistema muestra la lista de los asesores para que el usuario escoja por cual buscar.</p> <p>El usuario selecciona el asesor y da al botón "Mostrar".</p> <p>El sistema muestra un listado de todos los controles a clase del asesor seleccionado.</p>

**Tabla 90: Caso de prueba “Criterio de búsqueda de los controles a clases por asignatura”**

Escenario	Descripción	Variable 1 “asignatura”	Respuesta del sistema	Flujo central
-----------	-------------	----------------------------	-----------------------	---------------

EC1. Buscar controles a clases por asignatura correctamente	Mediante este escenario se puede buscar los controles a clase por asignatura en el sistema.	V "asignatura"	El sistema busca los controles a clase por asignatura y se las muestra al usuario.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra los tipos de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda de los controles a clases por asignatura.</p> <p>El sistema muestra la lista de las asignaturas para que el usuario escoja por cual buscar.</p> <p>El usuario selecciona el asesor y da al botón "Mostrar".</p> <p>El sistema muestra un listado de todos los controles a clase de la asignatura seleccionada.</p>

**Tabla 91: Caso de prueba "Criterio de búsqueda por rango de fecha de los controles a clases"**

Escenario	Descripción	Variable 1 "Fecha inicio"	Variable 2 "Fecha fin"	Respuesta del sistema	Flujo central
EC1. Buscar controles a clases por fecha correctamente	Mediante este escenario se puede buscar los controles a clases por fecha inicio y fin en el sistema.	V "29/5/2013"	V "21/5/2013"	El sistema busca los controles a clase por fecha y se las muestra al usuario correctamente.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra los tipos de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda de los controles a clases por fecha.</p> <p>El sistema permite que el usuario entre la fecha de inicio y la fecha de fin y da al botón "Mostrar".</p> <p>El sistema muestra</p>
		V "Fecha inicio"	V "Fecha"		



					un listado de todos los controles a clases existentes dentro de ese rango de fecha.
--	--	--	--	--	---

**Tabla 92: Caso de prueba "Asignar visitante a las preparaciones metodológicas"**

Escenario	Descripción	Variable 1 "Visitante"	Respuesta del sistema	Flujo central
EC1. Asignar visitantes a las preparaciones metodológicas correctamente.	Mediante este escenario se asignan visitantes a las preparaciones metodológicas correctamente en el sistema.	V "Seleccionar Visitante"	El sistema asigna los visitantes a las preparaciones metodológicas.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Preparaciones metodológicas" en el menú principal, el sistema muestra todos los datos de las preparaciones, en caso de que existan.</p> <p>El usuario escoge la opción de "Asignar visitante".</p> <p>El sistema muestra el campo a llenar -seleccionar visitante</p> <p>El usuario selecciona los visitantes y da al botón "Asignar".</p> <p>El sistema actualiza los datos de las preparaciones metodológicas.</p>
		V "Visitante"		

**Tabla 93: Caso de prueba "Criterio de búsqueda para las preparaciones por asignatura"**

Escenario	Descripción	Variable 1 "Asignatura"	Respuesta del sistema	Flujo central
-----------	-------------	----------------------------	-----------------------	---------------

EC1. Buscar preparaciones metodológicas por asignatura correctamente	Mediante este escenario se pueden buscar las preparaciones metodológicas por asignatura.	V "Seleccionar Asignatura"	El sistema busca las preparaciones metodológicas por asignatura y se las muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra los criterios de búsquedas, el usuario selecciona buscar preparaciones metodológicas por asignatura.  El sistema muestra las asignaturas y el usuario selecciona por cual quiere realizar su búsqueda.  El sistema muestra las preparaciones correspondientes al criterio de búsqueda.

**Tabla 94: Caso de prueba "Criterio de búsqueda para las preparaciones metodológicas por disciplina"**

Escenario	Descripción	Variable 1 "Disciplina"	Respuesta del sistema	Flujo central
EC1. Buscar las preparaciones metodológicas por disciplina correctamente	Mediante este escenario se pueden buscar las preparaciones metodológicas por disciplina en el sistema.	V "Seleccionar Disciplina"	El sistema busca las preparaciones metodológicas por disciplina y se las muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra los criterios de búsquedas, el usuario selecciona buscar preparaciones metodológicas por disciplina.  El sistema muestra las disciplinas y el usuario selecciona por cual quiere realizar su búsqueda.  El sistema muestra las preparaciones correspondientes al criterio de búsqueda.

**Tabla 95: Caso de prueba "Criterio de búsqueda para las preparaciones metodológicas por rango de fecha"**

Escenario	Descripción	Variable 1 "Fecha inicio"	Variable 2 "Fecha fin"	Respuesta del sistema	Flujo central
EC1. Buscar preparaciones por fecha correctamente	Mediante este Buscar preparaciones por fecha inicio y fin en el sistema.	V "29/5/2013"	V "21/5/2013"	El sistema busca las preparaciones por fecha y se las muestra al usuario correctamente.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra los tipos de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda de las preparaciones metodológicas por fecha.</p> <p>El sistema permite que el usuario entre la fecha de inicio y la fecha de fin y da al botón "Mostrar".</p> <p>El sistema muestra un listado con todas las preparaciones metodológicas existentes dentro de ese rango de fecha.</p>
		V "FechaInicio"	V "Fecha"		

**Tabla 96: Caso de prueba "Asignar tareas a un plan de trabajo"**

Escenario	Descripción	Variable 1 "Fecha"	Variable 2 "Asesor"	Respuesta del sistema	Flujo Central
EC1. Insertar los datos correctamente.	Mediante este escenario se insertan las tareas de los planes de trabajo correctamente	V "21/2/2013"	V "Osiel"	El sistema inserta correctamente las tareas del plan de trabajo correctamente.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "plan de trabajo" en el menú principal, el sistema da la opción de asignar las tareas al plan de trabajo.</p> <p>El usuario selecciona la opción "Asignar tarea".</p> <p>El sistema muestra los campos a llenar por el usuario: -fecha</p>
		V "Fecha"	V "nombre"		
		V "Fecha"	V "Cantidad de caracteres hasta 10 "		

					-asesor El usuario llena todos los campos satisfactoriamente y presiona el botón "Asignar". El sistema actualiza las tareas del plan de trabajo satisfactoriamente.
--	--	--	--	--	---

**Tabla 97: Caso de prueba "Modificar un plan de trabajo"**

Escenario	Descripción	Variable 1 "Fecha"	Variable 2 "Asesor"	Respuesta del sistema	Flujo Central
EC1. Modifica los planes de trabajo correctamente.	Mediante este escenario se modifican los planes de trabajo correctamente.	V "21/2/2013"	V "Osiel"	El sistema modifica correctamente los planes de trabajo.	El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra el listado de los planes de trabajo. El usuario escoge la opción "Editar plan de trabajo". El sistema muestra los campos a modificar por el usuario: -fecha -asesor El usuario llena todos los campos satisfactoriamente y presiona el botón "Editar". El sistema actualiza los datos del plan de trabajo modificado.
		V "Fecha"	V "nombre"		
		V "Fecha"	V "nombre"		

**Tabla 98: Caso de prueba "Eliminar un plan de trabajo"**

Escenario	Descripción	Variable 1 "Fecha"	Variable 2 "Asesor"	Respuesta del sistema	Flujo Central
EC1. Eliminar los planes de trabajo correctamente.	Mediante este escenario se eliminan los planes de trabajo correctamente.	V "21/2/2013"	V "Osiel"	El sistema elimina correctamente los planes de trabajo.	El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra el listado de los planes de trabajo. El usuario escoge la opción
		V "Fecha"	V "nombre"		

					<p>"Eliminar plan de trabajo".</p> <p>El sistema muestra los campos a los datos del plan de trabajo a eliminar:</p> <ul style="list-style-type: none"> <li>-fecha</li> <li>-asesor</li> </ul> <p>El usuario presiona el botón "Eliminar":</p> <p>El sistema elimina de la base de datos el plan y actualiza el listado de los planes de trabajo.</p>
--	--	--	--	--	--

**Tabla 99: Caso de prueba "Criterio de búsqueda para de los planes de trabajo por asesor"**

Escenario	Descripción	Variable 1 "Asesor"	Respuesta del sistema	Flujo central
EC1. Busca los planes de trabajo por asesor correctamente	Mediante este escenario se pueden buscar los planes de trabajo por asesor en el sistema.	V "Asesor"	El sistema busca los planes de trabajo por asesor y se las muestra al usuario.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra los criterios de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda "Por asesor".</p> <p>El sistema muestra una lista con los asesores del departamento para que el usuario escoja por cual buscar.</p> <p>El usuario selecciona el asesor y da al botón "Mostrar".</p> <p>El sistema muestra un</p>

				listado de todos los planes de trabajo del asesor seleccionado.
--	--	--	--	---

**Tabla 100: Caso de prueba "Criterio de búsqueda de los planes de trabajo por rango de fecha"**

Escenario	Descripción	Variable 1 "Fecha inicio"	Variable 2 "Fecha fin"	Respuesta del sistema	Flujo central
EC1. Buscar planes de trabajo por fecha correctamente	Mediante este escenario se buscan los planes de trabajo por fecha en el sistema.	V "29/5/2013"	V "21/5/2013"	El sistema busca los planes de trabajo por fecha y se los muestra al usuario.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra los criterios de búsqueda que puede realizar el usuario.</p> <p>El usuario selecciona la búsqueda "Por fecha".</p> <p>El sistema permite introducir la fecha de inicio y la fecha de fin para realizar la búsqueda en ese rango de fechas.</p> <p>El usuario introduce las fechas y da al botón "Mostrar".</p> <p>El sistema muestra un listado de todos los planes de trabajo comprendidos en ese rango de fechas.</p>
		V "Fecha inicio"	V "Fecha"		

**Tabla 101: Caso de prueba "Crear evaluación"**

Escenario	Descripción	Variable 1 "Evaluación"	Respuesta del sistema	Flujo central

EC1. Evaluar plan de trabajo correctamente	Mediante este escenario de evalúa el plan de trabajo correctamente en el sistema.	V "Adecuado"	El sistema evalúa el plan de trabajo correctamente y se lo muestra al usuario.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra el listado de los planes de trabajo existentes</p> <p>El usuario selecciona la opción "Evaluar".</p> <p>El sistema muestra el campo a llenar.</p> <p>-evaluación</p> <p>El usuario escoge la evaluación que desea darle a ese plan de trabajo y le da al botón "Aceptar".</p> <p>El sistema actualiza los datos del plan de trabajo evaluado.</p>

**Tabla 102: Caso de prueba "Eliminar tareas al plan de trabajo"**

Escenario	Descripción	Variable 1 "Tarea"	Variable 2 "Cumplida"	Respuesta del sistema	Flujo Central
EC1. Eliminar las tareas al plan de trabajo correctamente.	Mediante este escenario se eliminan las tareas al plan de trabajo correctamente	V "Tarea"	V "marcado"	El sistema elimina correctamente las tareas al plan de trabajo.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra el listado de los planes de trabajo existentes</p> <p>El usuario escoge la opción "Mostrar tareas".</p> <p>El sistema muestra un listado de las tareas del plan de trabajo.</p> <p>El usuario selecciona la opción "Eliminar tarea".</p> <p>El sistema le muestra los</p>
		V "visitar clase"	V "desmarcado"		

					<p>datos de la tarea que desea eliminar, y el usuario selecciona el botón "Eliminar".</p> <p>El sistema elimina correctamente la tarea y actualiza el listado de las mismas.</p>
--	--	--	--	--	--

**Tabla 103: Caso de prueba "Dar cumplimiento a las tareas del plan de trabajo"**

Escenario	Descripción	Variable 1 "Cumplimiento"	Respuesta del sistema	Flujo central
EC1. Evaluar el cumplimiento de las tareas del plan de trabajo correctamente	Mediante este escenario se les da cumplimiento a las tareas del plan de trabajo.	V "Marcado"	El sistema da cumplimiento a las tareas del plan de trabajo.	<p>El usuario una vez autenticado con los privilegios otorgados selecciona "Plan de trabajo" en el menú principal, el sistema muestra el listado de los planes de trabajo existentes</p> <p>El usuario escoge la opción "Mostrar tareas".</p> <p>El sistema muestra un listado de las tareas del plan de trabajo.</p> <p>El usuario selecciona la opción "Dar cumplimiento".</p> <p>El sistema actualiza la tarea con su cumplimiento</p>
		V "Desmarcado"		

**Tabla 104: caso de prueba "Generar reporte por asignatura de los controles a clases"**

Escenario	Descripción	Variable 1 "asignatura"	Respuesta del sistema	Flujo central



EC1. Generar reporte por asignatura de los controles a clases correctamente	Mediante este escenario se muestran los reportes de las asignaturas.	V "asignatura"	El sistema busca los reportes por asignatura y se las muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra varios reportes y el usuario selecciona "Controles a clases por asignaturas"  El usuario selecciona la asignatura.  El sistema muestra el reporte total de la asignatura seleccionada por el usuario.

**Tabla 105: Caso de prueba "Generar reporte del estado actual de los controles a clases por asignatura"**

Escenario	Descripción	Variable 1 "asignatura"	Respuesta del sistema	Flujo central
EC1. Buscar reporte actual por asignatura correctamente	Mediante este escenario se buscan los reportes actuales de las asignaturas correctamente.	V "asignatura"	El sistema busca los reportes actuales por asignatura y se los muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra varios reportes y el usuario selecciona "Controles a clases del estado actual por asignaturas"  El usuario selecciona la asignatura.  El sistema muestra el reporte del estado actual de la asignatura seleccionada por el usuario.

**Tabla 106: Caso de prueba "Generar reporte del estado actual de los controles a clases"**

Escenario	Descripción	Variable 1 "Controles"	Respuesta del sistema	Flujo central
-----------	-------------	---------------------------	-----------------------	---------------

EC1. Buscar reporte actual de los controles a clase correctamente	Mediante este escenario se buscan los reportes del estado actual de los controles a clases correctamente.	V "Controles"	El sistema busca los reportes actuales de los controles a clases y se los muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra varios reportes y el usuario selecciona "Controles a clases del estado actual"  El sistema muestra el reporte del estado actual de los controles a clases.

**Tabla 107: Caso de prueba "Generar reporte del estado actual de las evaluaciones de los asesores"**

Escenario	Descripción	Variable 1 "Planes de trabajo"	Respuesta del sistema	Flujo central
EC1. Buscar reporte actual de los planes de trabajo correctamente	Mediante este escenario de busca los reporte del estado actual los planes de trabajo correctamente.	V "Planes de trabajo"	El sistema busca los reportes actuales de los planes de trabajo y se las muestra al usuario.	El usuario una vez autenticado con los privilegios otorgados selecciona "Reportes" en el menú principal, el sistema muestra varios reportes y el usuario selecciona "Estado actual de las evaluaciones"  El sistema muestra el reporte del estado actual las evaluaciones de los asesores.