

**Universidad de las Ciencias Informáticas  
Facultad 1**



**Título:** Sistema de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deporte para la Facultad 1 de la Universidad de las Ciencias Informáticas

**Trabajo de Diploma para optar por el título de  
Ingenieros en Ciencias Informáticas**

**Autores:** Yailin Sardiña Díaz

Rafael Tizol Rosado

**Tutores:** Ing. Geidis Sánchez Michel

Ing. Manuel A. Quert Gómez

Ciudad de la Habana  
Junio, 2010

***Declaración de autoría***

Declaramos que somos los únicos autores del trabajo titulado:

Sistema de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deporte para la Facultad 1 de la Universidad de las Ciencias Informáticas.

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_  
Yailin Sardiña Díaz

\_\_\_\_\_  
Rafael Tizol Rosado

\_\_\_\_\_  
Ing. Geidis Sánchez Michel

\_\_\_\_\_  
Ing. Manuel Alejandro Quert Gómez

*Agradecimientos:*

*A mis padres por siempre confiar en mí, a mi hermana Mailín por su apoyo incondicional y por siempre darme fuerzas para seguir adelante.*

*A mis amistades por estar conmigo en todos momentos especialmente a Ivis por ayudarme siempre, a Ramón y a Ane por ayudarme cuando los necesité.*

*A mi hermana negra Yuritza por preocuparse siempre por mí y darme fuerzas para seguir adelante.*

*A Lilian por demostrarme ser mi amiga y siempre darme su apoyo.*

*A mi tía María por ser como una madre para mí y a mi prima Magdelais por ayudarme siempre y quererme tanto.*

*A mi abuela (madre) Onelia por darme su apoyo y comprensión y quererme tanto y a mi abuelo (padre) Antonio por ayudarme desde el cielo.*

*En general a toda mi familia.*

*A mi novio Josue por siempre estar a mi lado guiándome y dándome fuerzas para seguir adelante.*

*A mi compañero de tesis Rafael por su ayuda y preocupación.*

*A mis tutores Geidis y Manuel por apoyarnos y ayudarnos siempre.*

*Yailin*

*Agradecimientos:*

*A mi familia en general y especialmente a mis padres y hermana por darme su apoyo, comprensión y preocupación.*

*A mi novia y a su familia, por aceptarme entre ellos y darme su apoyo y cariño.*

*A mis amigos, por estar siempre a mi lado.*

*A mi compañera de tesis Yailin por su ayuda y preocupación.*

*A mis tutores Geidis y Manuel por apoyarnos y ayudarnos siempre.*

*Rafael*

*Dedicatoria:*

*Yailin*

*A mis padres*

*A mi hermana*

*A mi familia*

*A mi novio*

*Rafael*

*A mi familia*

*A mi novia*

En la Universidad de las Ciencias Informáticas (UCI) se desarrolla un proceso de seguimiento de la trayectoria de los estudiantes durante el transcurso de su carrera, para este proceso la Facultad 1 necesita llevar el control de la participación de sus estudiantes en las esferas de investigación, cultura y deporte.

La FEU lleva el control del proceso de seguimiento de cada estudiante de la Facultad. En estos momentos este proceso es realizado de forma manual, ocasionando que el trabajo resulte tedioso y complicado ya que en algunos casos existen duplicados, errores o ausencia de la información, dificultando el proceso de caracterización de los estudiantes, por lo que surge la necesidad de desarrollar un sistema de seguimiento a la trayectoria estudiantil, en las esferas de investigación, cultura y deporte para la Facultad 1 de la UCI con el objetivo de agilizar el trabajo de los directivos de la FEU en dicha Facultad, mejorar la calidad de la información, disminuir los riesgos por pérdida de la información y aumentar la seguridad de la misma.

**Palabras clave:** gestión de la información, trayectoria estudiantil.

---

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación teórica</b> .....	<b>4</b>
<b>1.1 Introducción</b> .....	<b>4</b>
<b>1.2 Conceptos asociados al problema</b> .....	<b>4</b>
<b>1.2.1 Gestión</b> .....	<b>4</b>
<b>1.2.2 Funciones de la gestión</b> .....	<b>4</b>
<b>1.2.3 Gestión de la información</b> .....	<b>5</b>
<b>1.2.4 Software de gestión</b> .....	<b>5</b>
<b>1.2.5 Trayectoria estudiantil</b> .....	<b>5</b>
<b>1.3 Estudio de sistemas de gestión relacionados con la trayectoria estudiantil</b> .....	<b>6</b>
<b>1.3.1 Sistemas de gestión relacionados con la trayectoria estudiantil en el mundo</b> .....	<b>6</b>
<b>1.3.2 Sistemas de gestión relacionados con la trayectoria estudiantil en Cuba</b> .....	<b>9</b>
<b>1.3.3 Sistemas de gestión relacionados con la trayectoria estudiantil en la UCI</b> .....	<b>11</b>
<b>1.4 Metodologías, herramientas y tecnologías para el proceso de desarrollo de software</b> .....	<b>12</b>
<b>1.4.1 Metodologías ágiles</b> .....	<b>12</b>
<b>1.4.1.1 XP:Extreme Programming</b> .....	<b>12</b>
<b>1.4.1.2 Scrum</b> .....	<b>13</b>
<b>1.4.2 UML: Lenguaje Unificado de Modelado</b> .....	<b>14</b>
<b>1.4.3 Herramientas CASE</b> .....	<b>15</b>
<b>1.4.3.1 Visual Paradigm v6.4</b> .....	<b>15</b>
<b>1.4.4 Lenguajes de programación</b> .....	<b>16</b>
<b>1.4.4.1 PHP</b> .....	<b>16</b>
<b>1.4.4.2 Java Script</b> .....	<b>17</b>
<b>1.4.5 CMS :Sistema Gestor de Contenido</b> .....	<b>17</b>

---

1.4.5.1 Drupal v 6.16.....	17
1.4.6 SGBD: Sistema gestor de base de datos.....	17
1.4.6.1 MySQL v5.0.....	17
1.4.7 Servidor web .....	18
1.4.7.1 Apache v2.0f.....	18
1.4.8 IDE: Entorno de Desarrollo Integrado.....	19
1.5 Justificación de las tecnologías a usar .....	19
1.6 Conclusiones .....	21
Capítulo2 Propuesta de solución.....	22
2.1 Introducción .....	22
2.2 Descripción del proceso de seguimiento a la trayectoria estudiantil en la Facultad 1. ....	22
2.3 Propósito del sistema .....	23
2.4 Descripción del sistema propuesto.....	23
2.4.2 Arquitectura y diseño .....	24
2.4.3 Despliegue.....	26
2.4.4 Modelo de datos.....	27
2.4.5 Seguridad del sistema.....	27
2.4.6 Product Backlog .....	28
2.4.7 Requerimientos no funcionales.....	40
2.4.8 Planificación de los sprints .....	41
2.5 Construcción de la propuesta de solución .....	50
2.5.1 Diagramas de pantallas por funcionalidades.....	50
2.5.2 Tratamiento de excepciones .....	51
2.5.3 Estándares de codificación .....	51
2.6 Conclusiones .....	53



<b>3.1 Introducción .....</b>	<b>54</b>
<b>3.2 Pruebas.....</b>	<b>54</b>
<b>3.3 Pruebas de aceptación.....</b>	<b>54</b>
<b>3.4 Resultado de las pruebas .....</b>	<b>68</b>
<b>3.5 Conclusiones .....</b>	<b>69</b>
<b>Conclusiones generales .....</b>	<b>70</b>
<b>Recomendaciones.....</b>	<b>71</b>
<b>Referencia bibliográfica.....</b>	<b>72</b>
<b>Bibliografía consultada .....</b>	<b>74</b>
<b>Glosario de términos.....</b>	<b>75</b>

## INTRODUCCIÓN

Debido al creciente desarrollo de la sociedad actual y al avance de las tecnologías, cada día el uso de las computadoras es más necesario en el mundo, así como la informatización de numerosas actividades y procesos que antes se realizaban de forma manual permitiendo una mayor calidad, agilidad y seguridad en la información.

Cuba no ha estado excluido de estos formidables cambios, que tienen lugar tanto en los países desarrollados como en los que poseen menos adelanto. Con el objetivo de lograr avances en la rama de la informática, se creó la UCI, la cual cuenta con 9 facultades en su sede central y 3 facultades regionales, cada una con perfiles diferentes, donde se vincula la docencia con la producción.

El perfil de la Facultad 1 se orienta hacia las tecnologías web, en la cual se vinculan varios proyectos productivos encaminados a la ejecución y perfeccionamiento de software para la Universidad y para su exportación. En la actualidad, la Facultad está desarrollando software con el objetivo de informatizar todo el trabajo relacionado con la docencia y la producción de los estudiantes. En la misma se hace necesario el control de la trayectoria de los estudiantes en las esferas de investigación, cultura y deporte.

La Federación Estudiantil Universitaria (FEU) lleva a cabo el proceso de caracterización de cada estudiante durante su estancia en la Universidad. En estos momentos todo este proceso se realiza de forma manual lo que hace más engorroso el control de esta información, la que se almacena en el mejor de los casos en documento Word y es enviada a niveles superiores a través de correo electrónico, dispositivos de almacenamiento de datos o por otros medios, lo que resulta compleja su manipulación y almacenamiento, afectándose así la agilidad y calidad del proceso, esto trae consigo que:

- Existan duplicados o ausencia de la información.
- Dificultades debido a la gestión manual de grandes volúmenes de información.
- Dificultad en la confección de la caracterización de los estudiantes.

Por tal situación se plantea como **problema a resolver**: ¿Cómo garantizar una mejor gestión de la trayectoria estudiantil en las esferas de investigación, cultura y deporte, en la Facultad 1 de la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio**: los procesos relacionados con la trayectoria estudiantil en las esferas de investigación, cultura y deporte en la Facultad 1 de la Universidad de las Ciencias Informáticas, donde el **campo de acción** estaría enmarcado en las herramientas y tecnologías para el desarrollo de sistemas de gestión de seguimiento a la trayectoria estudiantil.

Para darle solución al problema existente surge como **objetivo general**: desarrollar un sistema de gestión de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deporte, para la Facultad 1 de la Universidad de las Ciencias Informáticas.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Diseñar el marco teórico de los procesos relacionados con las esferas de investigación, cultura y deporte.
- Modelar los procesos relacionados con el seguimiento a la trayectoria estudiantil.
- Implementar el sistema de gestión de seguimiento a la trayectoria estudiantil.
- Probar el sistema de gestión de seguimiento a la trayectoria estudiantil.

La investigación se sustenta en la siguiente **hipótesis**: si se desarrolla un sistema de gestión de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deporte, para la Facultad 1 de la Universidad de las Ciencias Informáticas, se garantizará un control fiable de la información.

**Variables:**

**Variable independiente:** sistema de gestión de seguimiento a la trayectoria estudiantil.

**Variable dependiente:** control fiable de la información

**Operacionalización de las variables** Ver Anexo 1.

**Tareas de investigación:**

- Diagnóstico de los procesos relacionados con el seguimiento a la trayectoria estudiantil.
- Estudio del estado del arte para caracterizar los sistemas de gestión de seguimiento a la trayectoria estudiantil.

- Análisis y diseño del sistema de gestión de seguimiento a la trayectoria estudiantil
- Desarrollar un sistema de gestión de seguimiento a la trayectoria estudiantil.
- Diseño de casos de pruebas para el sistema de gestión de seguimiento a la trayectoria estudiantil.

Para el desarrollo de este trabajo se utilizan los siguientes métodos de investigación:

**Histórico-lógico:** se utiliza para conocer el uso de los sistemas de gestión existentes a nivel internacional y nacional que tengan similitud con el que se va a implementar, así como las investigaciones que se realizaron anteriormente referentes al tema.

**Analítico-sintético:** se utiliza para el análisis bibliográfico de la investigación y para sintetizar las características de los procesos y sistemas estudiados.

**Modelación:** se utiliza para crear abstracciones con el objetivo de explicar la realidad. Se modelan todos los procesos de la trayectoria estudiantil relacionados con las esferas de cultura, deporte e investigación en la Facultad 1 de la UCI.

### **Métodos empíricos**

**Observación:** se utiliza para percibir cómo fluyen los procesos involucrados en la trayectoria estudiantil relacionados con las esferas de cultura, deporte e investigación en la Facultad 1 de la UCI.

**Entrevista:** la entrevista se utiliza como técnica de recopilación de información para entender los procesos involucrados en la trayectoria estudiantil relacionados con las esferas de cultura, deporte e investigación en la Facultad 1 de la UCI.

# 1

## Capítulo Fundamentación teórica

### 1.1 Introducción

En el presente capítulo se explican los principales conceptos asociados al problema, se realiza la investigación de algunos sistemas similares a nivel internacional y nacional. Se definen las herramientas, tecnologías y metodologías que serán utilizadas para el desarrollo del sistema.

### 1.2 Conceptos asociados al problema

#### 1.2.1 Gestión

**Gestión:** “Conjunto de decisiones y acciones que llevan al logro de objetivos previamente establecidos”. [1]

#### 1.2.2 Funciones de la gestión

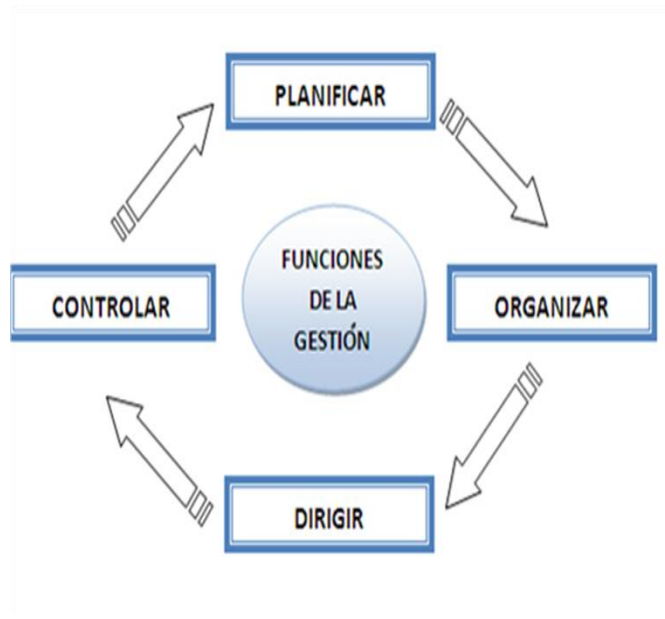
**Planificar:** proceso de establecer objetivos con el fin de alcanzar determinados resultados. [2]

**Organizar:** proceso de dividir el trabajo y de coordinar el logro de resultados que tienen un propósito común. [2]

**Dirigir:** proceso de conducir y coordinar esfuerzos laborales de las personas que integran una organización. Función mediante la cual se ponen en marcha las tareas programadas. [2]

**Controlar:** proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. [2]

En la figura 1 se puede ver como se relacionan estas funciones.



**Figura 1. Funciones de la gestión**

### 1.2.3 Gestión de la información

La gestión de la información comprende un conjunto de actividades relacionadas al proceso de registro, análisis y difusión de la información que se genera en las redes del conocimiento.[3]

### 1.2.4 Software de gestión

Las aplicaciones o software de gestión son aquellas diseñadas para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial o por un software que permite realizar al cliente los mismos procedimientos de forma no presencial, disminuyendo el esfuerzo empleado para la realización de los mismos.

### 1.2.5 Trayectoria estudiantil

La trayectoria de un estudiante es el comportamiento del mismo a lo largo de su estancia en un centro educacional, registrándose su desempeño en las diferentes actividades, tanto dentro como fuera de la institución y sus resultados.

### 1.3 Estudio de sistemas de gestión relacionados con la trayectoria estudiantil.

#### 1.3.1 Sistemas de gestión relacionados con la trayectoria estudiantil en el mundo.

##### **ALBA: Sistema Libre de Gestión Educativa**

Desarrollado por un grupo de estudiantes argentinos, es un proyecto de desarrollo de Software Libre para la realización de un “Sistema Informático Abierto de Gestión para Unidades Educativas” que brinda una herramienta para mejorar el trabajo cotidiano en las escuelas. El objetivo principal es crear un sistema de gestión abierto para centros educativos, capaz de adaptarse a las necesidades de la unidad donde sea utilizado. El sistema es desarrollado en Symfony y PHP 5, utiliza como servidor de base de datos MySQL 4.1 y el sistema operativo Linux.

El software está dividido en varias secciones, entre las que se encuentran: seguridad, gestión escolar, docentes, calendario, horario escolar, entre otras. Entre las funcionalidades que brinda se encuentran la gestión de los alumnos dentro del centro de estudios, posibilita que sean dados de alta y baja, que se realicen listados y búsquedas de alumnos dadas ciertas características. Permite, también, la gestión de notas, asistencia y ciertos registros médicos de los alumnos. Se encarga de la gestión de los profesores de los centros de estudios y de la confección de los horarios para cada uno de estos. Otra de las principales funcionalidades que ofrece es la gestión escolar, en esta se definen los grados/años que componen un plan de estudio, así como la gestión de secciones/divisiones en la que son agrupadas las materias/actividades. Posee secciones de informes o consultas donde se puede obtener información acerca de los estudiantes y docentes que componen el centro. Brinda una ayuda detallada del sistema a los usuarios del mismo, la cual puede ser consultada a través de la aplicación o de internet. [4]

**Desventajas:** este sistema, aunque cuenta con un gran número de funcionalidades relacionadas con la gestión académica estudiantil y con la gestión de información de sus profesores, no posee un registro de la información referente a la participación de los estudiantes en las esferas de investigación, cultura y deportes siendo estas las principales características demandadas por la institución con el objetivo de llevar una evaluación eficiente de los estudiantes.

### **SIGA: Software Integrado de Gestión Académica**

Aplicación informática utilizada por la Universidad de Córdoba (España) para la Automatización de la Gestión Académica. La versatilidad y flexibilidad del sistema permite organizar la informatización de diversas instituciones en un espectro que abarca a conservatorios, academias, colegios, institutos y universidades, así como centros de formación de empresas, y eventos especiales como maestrías y cursos de postgrado.[5]

**Ventajas y desventajas:** los estudiantes interactúan con el sistema directamente, existen varios módulos destinados a satisfacer las necesidades de estos, como son el módulo de Opciones de Alumnos, donde los estudiantes a través de los submódulos Gestión de Asistencias y Gestión de Calificaciones pueden conocer todos los detalles de su asistencia a clases y evaluaciones. Para los estudiantes matriculados de manera no presencial en el centro, está el submódulo Gestión de Tutorías que controla el número y tipo de contactos del alumno con el centro a través de su tutor. SIGA posee una gran desventaja, y es que su interfaz no es amigable lo cual puede provocar serios problemas de utilización teniendo en cuenta la diversidad de usuarios que interactúan con el sistema (alumnos, profesores, personal del centro, entre otros). Además de no contar con un registro de la información referente a la participación de los estudiantes en eventos de investigación, cultura y deportes, siendo estas las principales características demandadas por la UCI con el objetivo de llevar una evaluación eficiente de los estudiantes.

### **Ágora**

Ágora es un producto de software estándar de gran calidad para la Gestión de Centros Docentes y Academias de todo tipo, desarrollado por la empresa española Kherian Soft. Se adapta a cualquier tipo de centro o formación, ya sea de tipo oficial o de carácter libre (academias de enseñanza general, de idiomas, informática, música u oposiciones entre otras), ya se trate de un pequeño centro o una gran empresa de formación con una gestión centralizada de varias sucursales distantes geográficamente. (Kherian Soft, 2009) Para ello, la aplicación cuenta con varias versiones y múltiples opciones de organización docente para utilizar en cada caso.[6]

**Ventajas:** aporta soluciones en diversos sentidos, como son la gestión centralizada de datos a través de la cual se gestionan de forma centralizada todas las bases de datos de alumnos, profesores, aulas y clases; la generación automática de la documentación. Gestiona con gran facilidad excepciones que dificultan la planificación tales como: alumnos que sólo pueden asistir a la mitad de las clases



proyectadas de su grupo, o que no se adaptan a ningún grupo y toman clases de varios, clases que cambian de horario, profesores que están de baja y han de ser sustituidos, entre otras. Con el control automatizado de docencia real, el programa calculará cada día la docencia que en función de los horarios y las matrículas, teóricamente se habrá impartido, generando los registros correspondientes, permite al usuario verificarlos periódicamente para ajustar - con un simple clic de ratón - las incidencias (clase aplazada, no impartida, impartida aunque no esté planificada). Es especialmente eficaz para calcular los pagos a profesores. Así mismo, calculará de forma automática la disponibilidad horaria de profesores y aulas, y de plazas en grupos y clases. El control automatizado de asistencia real permite un registro automatizado de la asistencia.

**Desventajas:** la principal desventaja de este software radica en su carácter propietario, por lo que habría que pagar para su utilización e instalación en la UCI y por otro lado no cuenta con la capacidad de gestionar los datos relacionados con la trayectoria de los estudiantes, las cuales comprenden la participación de los estudiantes en eventos de investigación, cultura y deportes.

### **SIGIAEN: Sistema de Gestión Académica del Instituto de Altos Estudios Nacionales**

Desarrollado en Quito, Ecuador con el objetivo de custodiar y procesar todo el historial académico de los estudiantes, de esta forma, la Unidad de Estudiantes proporciona soporte a la administración docente a través del manejo del Sistema de Gestión Académica del IAEN (SIGIAEN) y del expediente de los estudiantes. [7]

Una de las funciones principales de la Unidad de Estudiantes es la de operar el sistema de información académica, vinculando las actividades de todas las áreas involucradas con la actividad académica tales como: bienestar estudiantil, finanzas y recursos humanos. Además, la Unidad de Estudiantes organiza el proceso de selección de estudiantes, de acuerdo con lo establecido en el reglamento que expide el Consejo Académico.

Entre sus **desventajas** se encuentra que solo se centra en el control académico, dejando fuera el trabajo con la información relacionada con la trayectoria no académica de los estudiantes, en este caso la participación de los estudiantes en eventos de investigación, cultura y deportes, siendo estas características de vital importancia para llevar una evaluación del estudiante.

### **FaMAF: Facultad de Matemática, Astronomía y Física**

La Secretaría de Asuntos Estudiantiles (SAE) en FaMAF de la Universidad de Córdoba, es el área de la Facultad con responsabilidad política e institucional sobre las diferentes problemáticas de la población estudiantil.

Se entiende al claustro estudiantil como una población por la diversidad de sus intereses y problemáticas. Durante su vida universitaria, los estudiantes no sólo cursan materias y seminarios sino también practican deportes, asisten a diferentes actividades extracurriculares como congresos o encuentros, necesitan atención médica, recurren a becas y todo tipo de cuestiones referidas a su ingreso, permanencia y egreso en las carreras de grado y pregrado.

En FaMAF, la Secretaría de Asuntos Estudiantiles cuenta con una Oficina de Becas y se encuentra en planificación un Gabinete Psicopedagógico que estudia y planifica en función de generar una institución más inclusiva y de calidad para los estudiantes, desde un punto de vista psicofísico integral tanto en lo individual como en lo colectivo. [8]

**Desventaja:** es software propietario, lleva el control de otros tipos de información las cuales no son de interés a la hora de caracterizar a un estudiante en su trayectoria estudiantil. Además de carecer del manejo de la información referente a la participación del estudiante en eventos culturales, el cual también forman parte del proceso de caracterización de los estudiantes.

### **1.3.2 Sistemas de gestión relacionados con la trayectoria estudiantil en Cuba.**

#### **GESTACAD**

Sistema informativo automatizado, desarrollado por el Departamento de Informática de la Universidad de Matanzas "Camilo Cienfuegos". Permite actualizar y mantener la información sobre estudiantes y profesores de una universidad y obtener determinados resultados propios del trabajo de las áreas implicadas, aunque el grueso de la información se obtiene mediante el acceso al sitio web de la universidad. El sistema es capaz de brindar las siguientes funcionalidades:

- Búsqueda de un alumno: brinda la ubicación según el horario docente.
- Listado de estudiantes por grupo.

- Reportes dinámicos de la información existente: se le da la posibilidad al usuario de seleccionar los campos de datos que desea obtener en el reporte así como el título de este y las condiciones que debe cumplir la información a mostrar.
- Reporte de notas por asignaturas y grupo: examen final, extraordinario, especial, premio.
- Tabla con los resultados docentes de un grupo en un semestre.
- Reporte de los resultados académicos de un estudiante en toda su carrera: Hoja de rendimiento.
- Actas de exámenes de las diferentes asignaturas.
- Registro de características de un grupo de estudiantes.
- Dar baja a un estudiante.

**Desventajas:** el sistema no brinda la posibilidad de llevar un seguimiento de la participación de los estudiantes en eventos de investigación, cultura y deportes, los cuales forman parte del proceso de caracterización de los estudiantes.

### **SIGENUS**

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un software desarrollado en el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) en el 2007 con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la Universidad Cubana, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad e integridad de la información, y a la vez, tan flexible que permita ser adaptado a todos los centros de educación superior del país con sus diversas particularidades y distintas maneras de realizar determinados procedimientos.

El sistema permite obtener un conjunto importante de reportes muy usados cotidianamente en el mundo de la educación superior. Consta con los siguientes módulos: Codificadores, Matrícula, Control de estudiantes, Plan de Estudio, Evaluaciones y Reportes.

El sistema se limita a hacer la gestión de los datos de los estudiantes sin permitir la gestión de ningún dato referente a los profesores, ni las relaciones de estos con los estudiantes y grupos académicos. Su interfaz principal necesita ser instalada en cada computadora que se va a utilizar por ser esta una aplicación de escritorio, lo que ocasiona pérdidas de tiempo y limitaciones en su uso.

### 1.3.3 Sistemas de gestión relacionados con la trayectoria estudiantil en la UCI.

#### UCIMAT

UCIMAT fue desarrollado por la UCI con el objetivo de agilizar el censo de estudiantes por determinados criterios, incluye la matrícula y modificación de los datos de los estudiantes, así como reportes generales, pero solo los que traía por defecto, en caso necesario, había que ir directamente a la base de datos; impidiendo que se pudieran realizar otras operaciones importantes como el reingreso, registro de traslado u otros datos requeridos. Fue utilizado durante los dos primeros cursos, brindando gran provecho al centro. [9]

**Desventaja:** este sistema no cuenta con la capacidad de llevar el control de los datos referentes a la trayectoria estudiantil, enfocándose solamente en llevar el control de las matrículas y modificación de los datos de los estudiantes, así como reportes generales de los mismos. Dejando fuera la gestión relacionada con la participación de los estudiantes en eventos de investigación, cultura y deportes, siendo estas características de vital importancia para llevar la evaluación del estudiante.

#### Akademios

Sistema de Gestión Académica que mantiene informado a los estudiantes sobre su desempeño académico y lleva el control del desarrollo del proceso docente, notas y asistencia. Además de la definición de los planes de estudio, asignaturas, disciplinas y perfiles. Esta herramienta se ha convertido en una referencia para resolver los problemas de esta índole. [10]

**Desventaja:** se basa totalmente en la plataforma .Net y su sistema gestor de base de datos es SQLServer, ambas tecnologías propietarias descartadas para el desarrollo en la UCI. Además deja fuera el trabajo relacionado con la trayectoria no académica de los estudiantes.

## Proyecto Kainos

Tiene como objetivo desarrollar una aplicación para la gestión de la información de la Unión de Jóvenes Comunistas (UJC) a todos sus niveles. También se propone controlar la información actualizada de cada uno de sus miembros y cuenta con un sistema de reportes que facilita el cierre del funcionamiento mensual de la organización. [11]

Este software no cuenta con la capacidad de llevar el control, mantenimiento y gestión de la información relacionada con la trayectoria de los estudiantes y su desempeño en eventos de investigación, cultura y deportes centrándose exclusivamente en la manipulación y gestión de la información de la UJC.

### 1.4 Metodologías, herramientas y tecnologías para el proceso de desarrollo de software.

Debido a que la Facultad 1 no cuenta con un sistema que le permita gestionar la trayectoria de los estudiantes vinculados a las áreas de investigación, cultura y deportes, se ha decidido implementar una aplicación que disponga de las funcionalidades fundamentales para garantizar una respuesta adecuada de los procesos que se llevan a cabo. Para conseguir desarrollar un sistema que tenga la calidad y la eficiencia requerida se utilizaron las siguientes herramientas, tecnologías y metodologías para su implementación.

#### 1.4.1 Metodologías ágiles

##### 1.4.1.1 XP: Extreme Programming

XP es una de las metodologías de desarrollo de software más exitosa en la actualidad, utilizada para proyectos de corto plazo y equipos de trabajos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Tiene como objetivo la satisfacción del cliente y potenciar al máximo el trabajo en equipo, surge como respuesta y posible solución a los problemas derivados del cambio en los requisitos, esta define cuatro variables para proyectos de software como son: costo, tiempo, calidad y ámbito, y como fases se definen: Planificación, Diseño, Desarrollo y Prueba. [12]

La metodología XP se basa en las pruebas unitarias, la prefabricación y la programación en pares como características fundamentales. Lo esencial de este tipo de metodología es la comunicación entre los usuarios y los desarrolladores, así como la simplicidad al desarrollar y codificar los módulos del sistema.[12]

### 1.4.1.2 Scrum

Se basa en la planificación de tareas modulares y pequeñas, que son realizadas por equipos de desarrollo, en períodos de trabajo intensivo llamados Sprint. Esta metodología define tres conceptos fundamentales: herramientas, prácticas, roles y responsabilidades.

Las herramientas contienen las tareas a entregar en cada momento del desarrollo del software. Estos listados están sujetos a cambios frecuentes.

Las prácticas se basan en mantener un desarrollo ágil y que se ajuste a los cambios.

Los roles y responsabilidades controlan el cumplimiento de las tareas, retroalimentándose de los resultados.

Esta metodología de desarrollo consta de tres fases:

- **Pregame:** se capturan los requisitos fundamentales, se crea el Product Backlog List y se prepara el primer *sprint* en el cual se define una propuesta de arquitectura, diseño y estándares de desarrollo.
- **Development:** es la etapa de desarrollo intensivo, donde ocurren los cambios impredecibles. Scrum posee una serie de prácticas para controlar las variables de tiempo, calidad, tecnología, recursos, requerimientos y herramientas para adaptarse a dichos cambios de forma flexible.
- **PostGame:** se llega a un consenso sobre la terminación del producto. Se realizan las pruebas, la integración y la documentación.

Los roles fundamentales de SCRUM son:

- **SCRUM Master:** garantiza que el desarrollo del producto se realice con el buen uso de las prácticas de Scrum. Su principal tarea es eliminar impedimentos y reducir riesgos. Este rol es desempeñado generalmente por el líder del proyecto.
- **Product Owner:** garantiza la integridad del Product Backlog List y se asegura de que las tareas a desarrollar tributen a la visión del proyecto.
- **SCRUM Team:** es la base del desarrollo de Scrum. Es un grupo de 5 a 10 personas, auto-dirigido y que realiza estimación de esfuerzos, sprint Backlog y señala los obstáculos a ser

removidos durante el proceso de desarrollo. Está compuesto por programadores, arquitectos, y diseñadores.

- Customer: es la representación del cliente que trabajará junto al equipo de desarrollo.
- Management: es el responsable de tomar las decisiones finales en cuanto a convenciones y estándares. Trabaja muy unido al SCRUM Master.

El ciclo de vida de Scrum es iterativo y se desarrolla a través de equipos auto-dirigidos. No define prácticas de ingeniería específica, delimita pocos artefactos y flujos de trabajo fundamentales. Cada iteración se denomina sprint.

Product Backlog List: es la lista de requerimientos funcionalidades o historias de usuario. Se elabora según se vaya desarrollando el sistema, lo que significa que mientras exista el producto existirá el Product Backlog. Las historias a medida que van surgiendo, se agregan a esta lista y son implementadas en orden, según su importancia o necesidad. Generalmente representan requerimientos funcionales, pero en ocasiones también representan requerimientos no funcionales que no están relacionados directamente con ninguna historia de usuario específica, pero aun así son necesarias para la entrega final del producto. Este tipo de historias se denominan historias técnicas.[13]

Sprint: es un período de tiempo que oscila entre una y seis semanas, donde se implementan las funcionalidades y se realizan las tareas definidas para el sprint. En esta fase, el producto experimenta los cambios señalados por el cliente en el diseño, codificación y pruebas. Además, la arquitectura es propensa a variar en dependencia de dichos cambios. Cada sprint es diseñado con mucho detalle, en una reunión de inicio de aproximadamente 4 horas de duración y es evaluado diariamente con el equipo en reuniones de 15 minutos de duración. [13]

### 1.4.2 UML: Lenguaje Unificado de Modelado

Es uno de los lenguajes más conocidos y utilizados en la actualidad. Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como: procesos de negocios, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante resaltar que UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software Orientado a Objetos (OO) y no para describir métodos o procesos. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no

especifica en sí mismo qué metodología o proceso usar. Es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. [14]

### 1.4.3 Herramientas CASE

Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

El empleo de herramientas CASE permite integrar el proceso de ciclo de vida:[15]

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento.

#### 1.4.3.1 Visual Paradigm v6.4

Visual Paradigm para UML es una de las herramientas CASE considerada muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Es fácil de instalar y actualizar. Genera código para varios lenguajes. También proporciona características tales como: generación del código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Cabe destacar igualmente su robustez, usabilidad y portabilidad. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto. Incorpora soporte para trabajo en equipo, esto permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios realizados por sus compañeros. [16]



### 1.4.4 Lenguajes de programación

#### 1.4.4.1 PHP

PHP es un lenguaje script que corre del lado del servidor. Es muy utilizado por los desarrolladores de páginas web dinámicas, aún más cuando su versión PHP 5 (v5.05), tiene incorporado un grupo de mejoras en cuanto a Programación Orientada a Objetos (POO), funciones y extensiones. Puede ser utilizado para la creación de otro tipo de programa incluyendo aplicaciones con interfaz gráfica usando las librerías. PHP, permite embeber sus pequeños fragmentos de código dentro de la página *HTML* y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas íntegramente en un lenguaje distinto de *HTML*. Presenta ventajas como:[17]

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL.
- Permite las técnicas de Programación Orientada a Objetos (POO).
- Se puede hacer todo lo que se pueda transmitir por vía *HTTP*.
- Es software libre. Se puede obtener en la web y su código está disponible bajo la licencia GPL.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, *XML* y creación de PDF).
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios *HTML*.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (extensiones).
- Puede interactuar con muchos motores de bases de datos
- Posee gran rapidez, generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

### 1.4.4.2 Java Script

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a los lenguajes Java y C. A diferencia de Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *DOM*. [18]

### 1.4.5 CMS: Sistema Gestor de Contenido

#### 1.4.5.1 Drupal v 6.16

Drupal es un sistema de administración de contenido para sitios web. Permite publicar artículos, imágenes u otros archivos y servicios añadidos como: foros, encuestas, votaciones, blogs, administración de usuarios y permisos. Es un sistema dinámico, en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones es almacenado en una base de datos y se edita utilizando un entorno web incluido en el producto. Contiene las principales características de un CMS en el núcleo del sistema, es utilizado por una gran comunidad de usuarios, es muy flexible, ampliable, presenta una buena arquitectura de programación ya que es seguro y posee un buen rendimiento. Al mismo tiempo es multilenguaje, posee un sistema de búsqueda integrada, genera un informe de errores y estadísticas e incorpora un sistema de caché que agiliza la velocidad de las páginas. [19]

### 1.4.6 SGBD: Sistema gestor de base de datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Permite realizar las siguientes acciones: [20]

- Definición de los datos
- Mantenimiento de la integridad de los datos dentro de la base de datos
- Control de la seguridad y privacidad de los datos

#### 1.4.6.1 MySQL v5.0

Sistema gestor de base de datos, su característica principal es la velocidad. Razón por la cual en un principio sus desarrolladores decidieron sacrificar ciertas prestaciones como: transacciones, llaves foráneas y subselects, a cambio de obtener un mayor rendimiento.

Fue escrito en C y C++. Se adapta fácilmente a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl, Java y su integración en los diferentes sistemas operativos. Utiliza múltiples tablas para almacenar y organizar la información. Permite gestionar archivos llamados de bases de datos. Cuenta con una gran cantidad de herramientas que administran cómodamente la base de datos. En las últimas versiones se incorporaron algunas transacciones de las que se carecía al inicio.[21]

Es muy destacable su condición de código abierto, de utilización gratuita, que puede modificarse con total libertad con solo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, convirtiéndolo en una de las herramientas más utilizadas por los programadores orientados a Internet.

### 1.4.7 Servidor web

Un servidor web puede definirse como un programa que recibe las peticiones de los usuarios o navegantes y las atiende o satisface. Por medio de la especificación de la búsqueda el servidor web buscará una página específica o ejecutará un programa, pero necesariamente enviará algún resultado sobre la búsqueda solicitada. Los sistemas operativos más utilizados por los servidores son Windows y Linux, siendo este último más estable y por lo tanto de uso más frecuente. Algunos servidores web son:[22]

- Apache
- Internet Information Server
- Sun Java System
- Zeus
- Light http

#### 1.4.7.1 Apache v2.0f

Apache es un servidor web y de aplicaciones de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, entre otras. Es el más extendido, por versátil y potente, entre los servidores web en la Internet. Apache presenta entre otras características, la emisión de mensajes de error altamente configurables. Su arquitectura es modular y consta de una sección con diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web. La configuración de cada módulo se hace mediante la configuración de las directivas que

están contenidas dentro del módulo. Los módulos de Apache se pueden clasificar en tres categorías:[23]

1. Módulos Base: contienen las funciones básicas de Apache.
2. Módulos Multiprocesos: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
3. Módulos Adicionales: cualquier otro módulo que le añada una funcionalidad al servidor. Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. De este último se han diseñado varios para cada uno de los sistemas operativos sobre los que se ejecuta Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle uno nuevo, de forma que no es necesario volver a instalar el software. Apache puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la *API* de programación de estos.

### 1.4.8 IDE: Entorno de Desarrollo Integrado

Un IDE es un programa que integra varias herramientas que facilitan el desarrollo de software sobre uno o varios lenguajes de programación. Como ventajas se tienen:[24]

- Una comunidad de millones de usuarios y miles de desarrolladores.
- Cientos de plugins.
- Soporte multi-lenguaje en una única herramienta (Eclipse).
- Coloreado de sintaxis PHP, autocompletado de código e inspección de métodos y atributos.
- Soporte básico de depuración de scripts PHP.

Entre los IDEs más reconocidos mundialmente se encuentran: Visual Studio .NET, Java para Eclipse y Borland C++. Para desarrollo Web figuran Zend Framework, Zend Studio, Cake PHP, Symfony entre otros. Estos últimos son herramientas poderosas para el desarrollo de software sobre PHP.

### 1.5 Justificación de las tecnologías a usar

Se propone hacer uso del Sistema Gestor de Contenido (CMS) Drupal para la creación del sistema. Debido a que es altamente configurable y con una flexibilidad que lo distingue de los demás en la gestión de usuarios y roles. También considerado uno de los más seguros, con alta velocidad de

respuesta a los usuarios y posee independencia entre la persistencia de datos y la presentación del contenido. Como lenguaje de programación se usa PHP debido a que es el lenguaje en el cual fue desarrollado Drupal, además de que es fácil de usar y a la vez seguro, es multiplataforma y cuenta con una amplia comunidad de desarrolladores.

MySQL se integra perfectamente con Apache como servidor de aplicaciones y PHP como lenguaje de programación web. La combinación de estos tres elementos es considerada por muchos desarrolladores como la mejor integración para el desarrollo web. Como servidor web Apache, es el seleccionado ya que al igual que MySQL y Drupal es multiplataforma, rápido, eficiente, seguro y altamente configurable en dependencia de las necesidades del usuario y de la aplicación, puede adaptarse a diferentes entornos de trabajo. Posee varios módulos de apoyo que permiten un mejor uso del mismo y extienden sus funcionalidades. Por versátil y potente es el más usado entre los servidores web en Internet. Tanto el CMS como el sistema gestor de base de datos y el servidor web son libres y cuentan con licencia *GPL*, siendo esta una de las mayores ventajas para hacer uso de ellos. Como herramienta CASE se utilizará Visual Paradigm, ya que es considerada una herramienta muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Se decidió optar por Zend Studio para Eclipse 2008 para el desarrollo de este trabajo, debido a que integra lo mejor de las prácticas actuales para desarrollo web, permite la integración con Eclipse y simplifica en gran medida las tareas de integración. Posee herramientas para editar, analizar, optimizar y depurar el código generado. Permite trabajar con bases de datos y otros lenguajes relacionados como JavaScript, *HTML* y *XML*. Está diseñado para adaptarse a los procesos de las metodologías de desarrollo ágiles.

Como metodología de desarrollo se utilizará la combinación de dos metodologías ágiles: Scrum y XP. Debido a que el desarrollo del producto final debe realizarse en un breve período de tiempo. Esta combinación se ajusta perfectamente a las necesidades y condiciones del equipo de desarrollo.

La combinación de SCRUM y XP se traduce en un proceso de desarrollo del software marcado por continuas entregas al cliente, el cual es consciente del avance del proyecto. La documentación generada es la estrictamente necesaria, por lo que es recomendable para proyectos pequeños o medianos que requieran una entrega inmediata y que cuentan con pocos integrantes en su equipo de trabajo.

### **1.6 Conclusiones**

En este capítulo se realizó un análisis del estado del arte y de los conceptos necesarios para el entendimiento de la creación del sistema a desarrollar. Además de profundizar en los aspectos teóricos necesarios para el desarrollo del trabajo. Se fundamentó el ambiente de desarrollo que facilitará la implementación de la solución.

# 2

## Capítulo

## Propuesta de solución

### 2.1 Introducción

En este capítulo se abordan los aspectos fundamentales de la propuesta de solución de este trabajo. Se especifican los requisitos que debe cumplir la aplicación. Se describen las características del sistema en cuanto a la arquitectura y los patrones de diseño utilizados y algunos parámetros relacionados con la seguridad, el tratamiento de excepciones y los estándares de codificación utilizados en su construcción.

### 2.2 Descripción del proceso de seguimiento a la trayectoria estudiantil en la Facultad 1.

Este trabajo pretende llevar el control de seguimiento a la trayectoria estudiantil en la Facultad 1 de la UCI que se lleva a cabo en el local de la FEU de la Facultad. Para ello es necesario conocer la forma en que se realiza este proceso en la actualidad.

Este proceso de integralidad se empieza a medir desde el 1er año del estudiante, llevando el control de toda la información referente a la participación del estudiante en las diferentes esferas de la vida estudiantil. El presidente de brigada es el encargado de llevar el control de esta información mensualmente, registrando la participación del estudiante en las diferentes actividades que se realizan. Luego es entregada al organizador de la FEU, el cual verifica toda la información recibida; para saber si un determinado estudiante ha participado en algún evento o en cuántos ha participado, los organizadores proceden a realizar una búsqueda manual en los documentos archivados, en vista de la elaboración del informe de integralidad y la caracterización, en la que se registra toda la información almacenada.

### 2.3 Propósito del sistema

En estos momentos en la Facultad 1 no existe un sistema que gestione el seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deportes. Es por eso que surgen dificultades a la hora de caracterizar a un estudiante tales como:

- Existan duplicados o ausencia de la información.
- Que a la hora de caracterizar a un estudiante falten datos.
- Gestión manual de grandes volúmenes de información.

Sobre la base de eliminar esos problemas y debido a la necesidad que tiene la Facultad 1 de gestionar los procesos relacionados con la trayectoria estudiantil en las esferas de investigación, cultura y deportes, se desarrollará un sistema que gestione el seguimiento de la trayectoria estudiantil. El sistema tendrá como objetivo automatizar los procesos de gestión que dan seguimiento a la misma, con lo que se pretende eliminar parte de los problemas presentes y agilizar este proceso.

### 2.4 Descripción del sistema propuesto

El sistema propuesto centralizará su función en el agrupamiento de la información referente a la trayectoria estudiantil en las esferas de investigación, cultura y deportes de la Facultad 1, facilitando agilidad en el proceso de búsqueda de datos referente a cualquier estudiante para llevar a cabo su caracterización.

### Roles de usuarios

A continuación se relacionan los roles definidos para actuar sobre el sistema de gestión de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deportes de la Facultad 1.



Roles	Justificación
Presidente de brigada	Se encarga de revisar las caracterizaciones de los estudiantes y registrar toda la información referente a la participación de los estudiantes en los diferentes eventos.
Administrador	Tiene acceso a todas las funcionalidades del sistema con el fin de administrar la aplicación.
Organizador FEU	Tiene acceso a toda la información registrada y a las funcionalidades del sistema.
Estudiante	Solo consulta información.

Tabla 2.1 Definición de los roles de usuarios

### 2.4.2 Arquitectura y diseño

Uno de los objetivos que se persigue con la implementación de un sistema que gestione el seguimiento de la trayectoria estudiantil es permitir el acceso rápido a la información de los estudiantes durante su estancia en la UCI referente a las esferas de investigación, cultura y deportes. Por esto es necesario realizar un correcto diseño que permita darle una arquitectura apropiada al sistema, con el fin de facilitar la construcción del mismo.

En múltiples foros y espacios de discusión especializada, muchos desarrolladores asignan a Drupal una arquitectura modular. Sin embargo, a pesar de lo sugerente y clasificador, esta no se encuentra definida o avalada por ningún estándar internacional.

Se destacan dos categorías de estilos arquitectónicos capaces de ajustarse a una aplicación construida utilizando Drupal: Llamada y retorno y Componentes independientes. Antes de analizarlas con más profundidad, es preciso describir detalladamente el funcionamiento de Drupal.

En Drupal existe una clasificación básica llamada nodo, que contiene cualquier información almacenada. Cada nodo en específico pertenece a un tipo de contenido determinado. Los tipos de contenido son plantillas que definen como almacenar la información de los nodos. A manera de ejemplo, en Drupal, el tipo de contenido *Page* (Página) guarda un nodo de información que contiene un título y un cuerpo del contenido. Por otra parte, un tipo de contenido *Evento*, podría guardar nodos de información con los campos nombre del evento, tipo de evento y fecha, entre otros. Las funcionalidades de Drupal están separadas en módulos. Cada módulo gestiona su generación de interfaz, lógica de negocio y acceso a datos. Estos módulos pueden clasificarse de manera intuitiva en dos categorías: los que componen el *Core* y los que extienden las funcionalidades de la aplicación. En el *Core* se encuentra lo necesario para que el sistema comprenda el trabajo de los demás módulos y el CMS funcione.

El resto de los módulos son desarrollados por cualquier miembro de la comunidad de Drupal. Para que el *Core* interprete correctamente el funcionamiento de cada módulo, estos presentan un grupo de funciones llamadas *hooks* (ganchos). Cada *hook* tiene una función específica, como definir el menú de opciones, el menú de administración, la generación de las páginas al usuario, entre otras. Tienen parámetros muy bien definidos. Cuando el sistema necesita mostrar cualquier información al usuario, llama a todos los *hooks* que tienen el nombre adecuado y luego define cuál de ellos se encuentra en el módulo correcto y lo ejecuta.

Una vez explicado de forma elemental el funcionamiento de Drupal, se puede concluir que no es un sistema orientado a objetos, aunque la estructura de nodos y tipos de contenido sugieran un comportamiento similar. Tampoco puede ubicarse en una aplicación construida por capas, pues como se dijo anteriormente, cada módulo gestiona su propia interfaz, lógica de negocio y acceso a datos. No existe una capa uniforme a ningún nivel de la aplicación. Sin embargo, un sistema desarrollado en Drupal se ajusta muy bien a la definición de componentes independientes, dado en gran medida por el nivel de independencia de sus módulos. Los *hooks* tienen el mismo nombre en cada uno de los módulos, pero describen un comportamiento diferente, ideal para ubicarse como un sistema orientado a eventos. Drupal presenta comportamientos descritos en varios patrones de diseño:

*Singleton*: limita a solo una la cantidad de instancias de una clase y proporciona un punto de acceso global a la misma. Los nodos de Drupal son tratados como objetos, de los cuales solo existe

una única instancia que se referencia teniendo en cuenta los niveles de acceso permitidos por el usuario.

*Observer*: plantea las relaciones entre objetos de forma tal que cuando alguno de ellos varía alguna de sus propiedades, el observador informa al resto de los objetos del cambio. Es una forma eficiente de capturar eventos, al igual que el tratamiento que da Drupal a sus *hooks*.

*Command*: permite realizar peticiones a un objeto sin conocer la petición en sí, ni conocer el destino de la misma. Se logra encapsulando los datos de la petición en un objeto. Es útil cuando los parámetros de la petición pueden ser de un amplio espectro. Drupal no tiene objetos de clases pero utiliza sus tablas en la base de datos para guardar algunas de las acciones de los usuarios.

*Reflection Pattern*: este patrón es predominante en Drupal. Permite adaptarse dinámicamente a los cambios en las solicitudes a la información almacenada. Estos cambios pueden afectar la estructura de la información y otros aspectos importantes, por lo que el uso de este patrón permite a una aplicación mantenerse confiable y segura en un entorno altamente variable.

### 2.4.3 Despliegue

El diagrama de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software. La figura 2 presenta el diagrama de despliegue propuesto teniendo en cuenta los recursos con los que contaba el cliente.

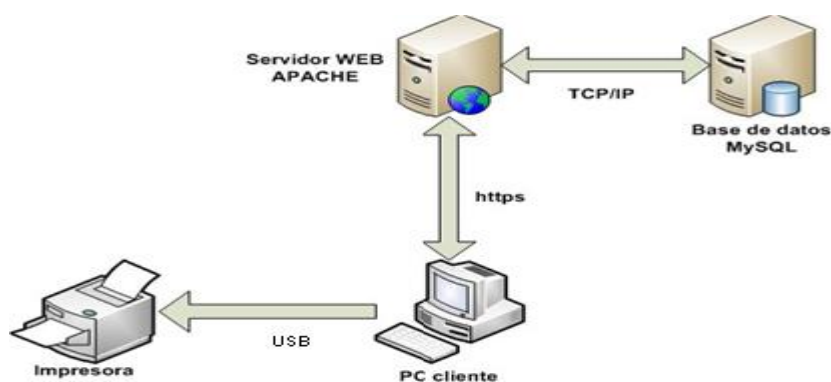


Figura 2. Diagrama de despliegue.

## 2.4.4 Modelo de datos

Un modelo de datos está orientado a describir una base de datos, en el cual se definen los tipos de datos y la forma en que se relacionan. Drupal autogenera la estructura de su base de datos, la cual está formada por un gran número de entidades y sus relaciones, destinadas a garantizar su funcionamiento. En la figura 3 se muestra el modelo de datos del sistema y sus entidades más significativas.

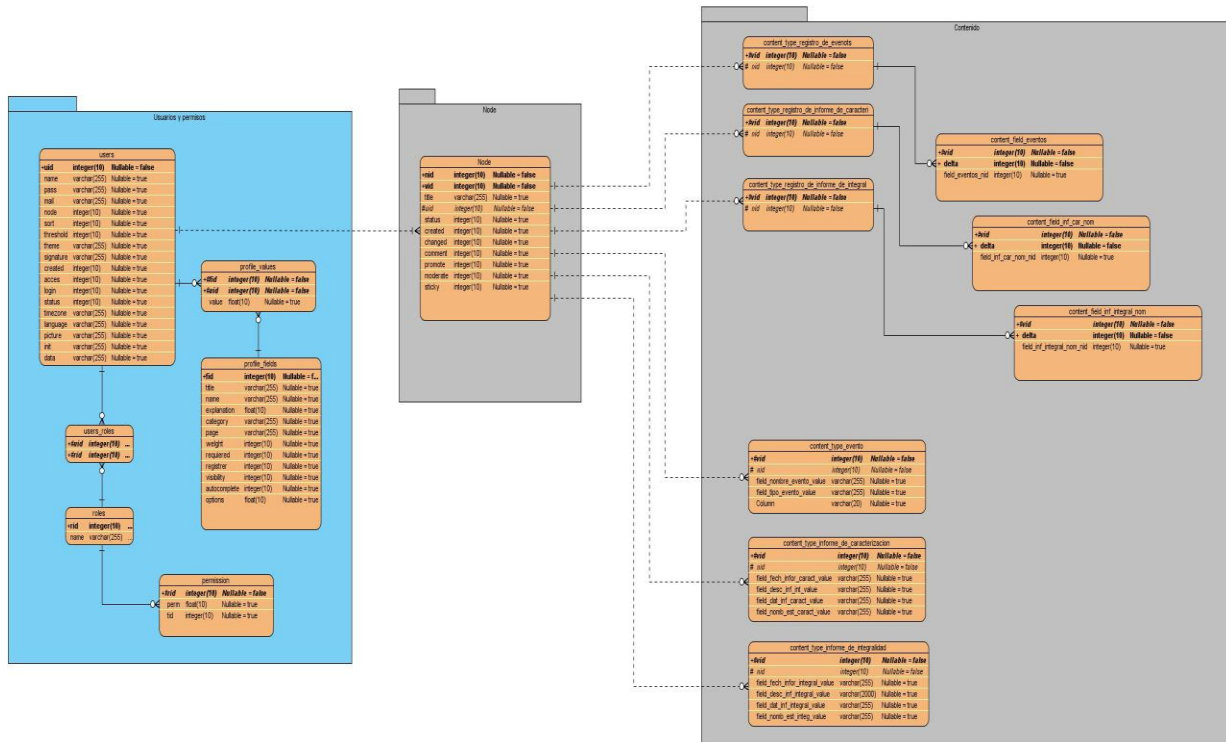


Figura 3. Modelo de datos

## 2.4.5 Seguridad del sistema

Al tratarse de un sistema web, las mayores amenazas a la seguridad provienen de los datos que se introducen en los formularios del mismo. Esto se refiere principalmente a scripts que traspasan los permisos asignados a los usuarios finales y a la inyección de código SQL. El Core de Drupal provee validación de los datos desde el lado del servidor, controlando estas agresiones.

La barra de direcciones es también un blanco frecuente de ataques. Estos consisten en escribir las direcciones a las que se desea acceder en la misma barra de direcciones y burlar los permisos asignados por el administrador. Para evitar esto, Drupal cuenta con el módulo `phpid`, que regula los permisos de las URLs solicitadas al servidor.

La tercera línea defensiva contra ataques malintencionados constituye el trabajo y permisos por roles. Cada usuario final del sistema accede al mismo para realizar tareas específicas de su rol. Una estricta asignación de permisos permite controlar el acceso de cada usuario solamente a las áreas que le están asignadas. Por último, el sistema regula las direcciones IP, los usuarios y las direcciones de correo electrónico que pueden acceder por la red. Estas reglas de acceso pueden ser fácilmente añadidas, modificadas y eliminadas por el administrador.

Esta medida posibilita habilitar el acceso de una nueva computadora o usuario. Por supuesto, también permite lo opuesto, la negación del servicio a usuarios, ordenadores y direcciones de correo electrónico.

### 2.4.6 Product Backlog

En el Product Backlog se enuncian y describen todos los requisitos o funcionalidades que deben estar presentes en la aplicación. Se llaman historias de usuario o elementos de la pila y se describen utilizando los términos del cliente. El Product Backlog se confecciona con el fin de identificar las necesidades del cliente para que se pueda desarrollar un producto que cumpla con sus expectativas.

Historia de usuario	
<b>Número: 1</b>	<b>Usuario:</b> Presidente de brigada, Organizador FEU y Estudiante
<b>Nombre historia:</b> Autenticar usuario	
<b>Puntos estimados: 7</b>	<b>Iteración asignada: 4</b>
<b>Descripción:</b> para entrar al sistema el usuario tiene que autenticarse en el sistema haciendo uso de un usuario y una contraseña.	
<b>Como probarlo:</b> el usuario debe autenticarse en el sistema. Comprobar que pueda acceder al sistema y obtener un rol con los permisos correspondientes.	
<b>Nota:</b> los usuarios solo podrán autenticarse por el dominio de la UCI. -Se necesitan los datos del LDAP_ UCI.	

Tabla 2.2 Historia de usuario Autenticar usuario

Historia de usuario	
<b>Número: 2</b>	<b>Usuario:</b> Administrador y Organizador FEU
<b>Nombre historia:</b> Administrar usuario	
<b>Puntos estimados: 7</b>	<b>Iteración asignada: 4</b>
<b>Descripción:</b> el sistema debe permitir asignarle permisos a los roles del sistema y los roles a los usuarios para controlar que la información sea accedida por las personas autorizadas.	
<b>Cómo probarlo:</b> a un usuario registrado en la base de datos asignarle un rol. Cuando el usuario vuelva a autenticarse comprobar que cuente con los permisos del rol asignado.	
<b>Nota:</b> cada usuario del sistema tendrá acceso según su rol.	

Tabla 2.3 Historia de usuario Administrar usuarios

Historia de usuario	
<b>Número: 3</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Buscar listado de estudiantes por evento	
<b>Puntos estimados: 10</b>	<b>Iteración asignada: 3</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de buscar los diferentes tipos de eventos de: investigación, cultura y, deporte, y mostrar un listado de los estudiantes que participaron en él.	
<b>Como probarlo:</b> seleccionar el tipo de evento que desea buscar, filtrar los campos por los que desea realizar la búsqueda y comprobar el resultado.	
<b>Nota:</b> los campos para filtrar deben permitir diferentes operadores de comparación, como: igual, contiene y no es igual.	

Tabla 2.4 Historia de usuario Buscar listado de estudiantes por evento

Historia de usuario	
<b>Número: 4</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Buscar listado de eventos en los que participa un estudiante	
<b>Puntos estimados: 10</b>	<b>Iteración asignada: 3</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de buscar y listar los diferentes tipos de eventos: investigativos, culturales, deportivos y otros, en los que participa un estudiante.	
<b>Como probarlo:</b> seleccionar el estudiante, filtrar los campos por los que desea realizar la búsqueda de los eventos y comprobar el resultado.	
<b>Notas:</b> los campos para filtrar deben permitir diferentes operadores de comparación, como: igual, contiene y no es igual.	

Tabla 2.5 Historia de usuario Buscar listado de eventos en los que participa un estudiante

Historia de usuario	
<b>Número: 5</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Mostrar listado de eventos	
<b>Puntos estimados: 10</b>	<b>Iteración asignada: 3</b>
<b>Descripción:</b> el sistema debe generar listados de eventos.	
<b>Como probarlo:</b> acceder a los elementos mencionados en la descripción y comprobar que se genera el listado con los datos correctos.	
<b>Notas:</b>	

Tabla 2.6 Historia de usuario Mostrar listado de eventos

Historia de usuario	
<b>Número: 6</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Mostrar listado de estudiantes	
<b>Puntos estimados: 10</b>	<b>Iteración asignada: 3</b>
<b>Descripción:</b> el sistema debe generar listado de estudiantes que participan en un evento específico.	
<b>Como probarlo:</b> acceder a los elementos mencionados en la descripción y comprobar que se genera el listado con los datos correctos.	
<b>Notas:</b>	

Tabla 2.7 Historia de usuario Mostrar listado de estudiantes

Historia de usuario	
<b>Número: 7</b>	<b>Usuario:</b> Organizador FEU
<b>Nombre historia:</b> Añadir evento	
<b>Puntos estimados: 8</b>	<b>Iteración asignada: 1</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de añadir un evento en el cual el estudiante haya participado.	
<b>Como probarlo:</b> acceder a la opción añadir un evento. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b>	

Tabla 2.8 Historia de usuario Añadir evento



Historia de usuario	
<b>Número: 8</b>	<b>Usuario:</b> Organizador FEU
<b>Nombre historia:</b> Modificar evento	
<b>Puntos estimados: 7</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar cualquier información referente a un evento cualquiera, en el cual el estudiante haya participado y no se encuentre correctamente en el sistema.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del evento seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.9 Historia de usuario Modificar evento

Historia de usuario	
<b>Número: 9</b>	<b>Usuario:</b> Organizador FEU
<b>Nombre historia:</b> Eliminar eventos	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar un evento teniendo en cuenta las restricciones de eliminación, o eliminarlo del listado de eventos asignado a los estudiantes.	
<b>Como probarlo:</b> seleccionar y eliminar evento. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.10 Historia de usuario Eliminar eventos

Historia de usuario	
<b>Número: 10</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Crear un informe de integralidad	
<b>Puntos estimados: 8</b>	<b>Iteración asignada: 1</b>
<b>Descripción:</b> el sistema debe permitir la creación de un informe de integralidad a partir de la recopilación de la información registrada de cada estudiante.	
<b>Como probarlo:</b> acceder a la opción Crear informe de integralidad. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b> solo se creará un nuevo informe de integralidad a los estudiantes que no cuenten con uno creado.	

Tabla 2.11 Historia de usuario Crear un informe de integralidad

Historia de usuario	
<b>Número: 11</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Modificar informe de integralidad	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar cualquier informe de integralidad referente a un estudiante.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del informe seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.12 Historia de usuario Modificar informe de integralidad

Historia de usuario	
<b>Número: 12</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Eliminar informe de integralidad	
<b>Puntos estimados: 7</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar un informe de integralidad asignado a un estudiante.	
<b>Como probarlo:</b> seleccionar y eliminar informe. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.13 Historia de usuario Eliminar infome de integralidad

Historia de usuario	
<b>Número: 13</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Crear un informe de caracterización	
<b>Puntos estimados: 8</b>	<b>Iteración asignada: 1</b>
<b>Descripción:</b> el sistema debe permitir la creación de un informe de caracterización a partir de la recopilación de la información registrada de cada estudiante.	
<b>Como probarlo:</b> acceder a la opción Crear informe de caracterización. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b> Solo se creará un nuevo informe de caracterización a los estudiantes que no cuenten con uno creado.	

Tabla 2.14 Historia de usuario Crear un informe de caracterización

Historia de usuario	
<b>Número: 14</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Modificar informe de caracterización	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar cualquier informe de caracterización referente a un estudiante.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del informe seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.15 Historia de usuario Modificar informe de caracterización

Historia de usuario	
<b>Número: 15</b>	<b>Usuario:</b> Presidente de brigada y Organizador FEU
<b>Nombre historia:</b> Eliminar informe de caracterización	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar un informe de caracterización asignado a los estudiantes.	
<b>Como probarlo:</b> Seleccionar y eliminar informe. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.16 Historia de usuario Eliminar informe de caracterización

Historia de usuario	
<b>Número: 16</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Añadir estudiantes	
<b>Puntos estimados: 8</b>	<b>Iteración asignada: 1</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de añadir estudiantes.	
<b>Como probarlo:</b> acceder a la opción insertar estudiante. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b>	

Tabla 2.17 Historia de usuario Añadir estudiantes

Historia de usuario	
<b>Número: 17</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Modificar estudiante	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar los datos de un estudiante.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del estudiante seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.18 Historia de usuario Modificar estudiante

Historia de usuario	
<b>Número: 18</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Eliminar estudiante	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar a un estudiante.	
<b>Como probarlo:</b> seleccionar y eliminar a un estudiante. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.19 Historia de usuario Eliminar estudiante

Historia de usuario	
<b>Número: 19</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Añadir presidente de brigada	
<b>Puntos estimados: 8</b>	<b>Iteración asignada: 1</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de añadir un presidente de brigada en la aplicación.	
<b>Como probarlo:</b> Acceder a la opción insertar presidente de brigada. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b>	

Tabla 2.20 Historia de usuario Añadir presidente de brigada

Historia de usuario	
<b>Número: 20</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Modificar presidente de brigada	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar los datos de un presidente de brigada.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del presidente de brigada seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.21 Historia de usuario Modificar presidente de brigada

Historia de usuario	
<b>Número: 21</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Eliminar presidente de brigada	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar un presidente de brigada de la aplicación.	
<b>Como probarlo:</b> seleccionar y eliminar un presidente de brigada. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.22 Historia de usuario Eliminar presidente de brigada

Historia de usuario	
<b>Número: 22</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Añadir organizador FEU	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de añadir a un organizador FEU en la aplicación.	
<b>Como probarlo:</b> acceder a la opción insertar organizador FEU. Llenar y enviar todos los datos del formulario. Comprobar si los datos fueron guardados.	
<b>Notas:</b>	

Tabla 2.23 Historia de usuario Añadir organizador FEU

Historia de usuario	
<b>Número: 23</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Modificar organizador FEU	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de modificar los datos de un organizador FEU.	
<b>Como probarlo:</b> acceder al formulario que contiene los datos del organizador FEU seleccionado y modificar los datos necesarios. Enviar y comprobar que se guardaron las modificaciones correctamente.	
<b>Notas:</b>	

Tabla 2.24 Historia de usuario Modificar organizador FEU



Historia de usuario	
<b>Número: 24</b>	<b>Usuario:</b> Organizador FEU y Administrador
<b>Nombre historia:</b> Eliminar organizador FEU	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 2</b>
<b>Descripción:</b> el sistema debe brindar la posibilidad de eliminar a un organizador FEU de la aplicación.	
<b>Como probarlo:</b> seleccionar y eliminar a un organizador FEU. Verificar que no queden registros en la base de datos.	
<b>Notas:</b>	

Tabla 2.25 Historia de usuario Eliminar organizador FEU

Historia de usuario	
<b>Número: 25</b>	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Diseñar interfaz	
<b>Puntos estimados: 6</b>	<b>Iteración asignada: 4</b>
<b>Descripción:</b> la interfaz del sistema debe tener un diseño sencillo, con colores pocos llamativos. Debe mostrar información relacionada solamente con las funcionalidades del sistema, para no facilitar distracción en el trabajo que se desarrolle.	
<b>Como probarlo</b>	
<b>Nota s:</b>	

Tabla 2.26 Historia técnica Diseñar interfaz

### 2.4.7 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener.

➤ **Software:**

- Apache como servidor de aplicaciones.
- MySQL como gestor de base de datos.
- PHP como lenguaje de programación.
- **Restricciones en el diseño.**
  - Para organizar el diseño del sistema, debe ser utilizadas las metodologías Scrum y XP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Visual Paradigm.
- **Rendimiento:**
  - El sistema estará implementado sobre una tecnología web, facilitando su uso a través de la red.
  - El sistema deberá ser capaz de gestionar toda la información y dar respuesta en el menor tiempo posible.
- **Hardware:**
  - Para la explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 128 MB RAM mínimos.
- **Seguridad:**
  - Solo tendrán acceso al sistema aquellos usuarios que puedan autenticarse por el dominio de la Universidad o tengan determinados permisos dados por el organizador FEU y el Administrador del sistema.

### 2.4.8 Planificación de los sprints

Cada sprint tiene una reunión inicial de planificación en la que se decide que historias se van a incluir en el mismo.

El Sprint Backlog es la lista de tareas en las que se dividen las historias, con el fin de convertirlas en funcionalidades del sistema, logrando que se incremente el desarrollo del producto según se vayan elaborando cada uno de los sprints. Son una especie de guía que el equipo confecciona con

el propósito de orientar sus trabajos mientras implementan el sistema. Para una correcta planificación de los sprints son necesarios una serie de datos:

- **Cantidad de días-hombre:** es la suma de los días que puede dedicar cada miembro del equipo de trabajo a la producción durante el sprint.
- **Factor de dedicación del equipo:** es una estimación del aprovechamiento del tiempo por parte del equipo de trabajo durante el sprint. Se representa en %.
- **Velocidad estimada del sprint:** se calcula multiplicando la cantidad de días hombre por el factor de dedicación.
- **Velocidad final del sprint:** es la suma de las estimaciones iniciales de las historias de usuarios que se incluyeron en el sprint.

A continuación, aparecen las planificaciones de cada uno de los Sprint creados para construir la aplicación:

### **Sprint 1 (Crear información)**

Cantidad de días que debe durar el sprint: 56. Del 13/10/2009 al 8/12/2009

Cantidad de días hombre: son dos miembros en el equipo. Cada uno puede aportar en ese período un total de 25 días. Esto hace un total de 50 días hombre.

Factor de dedicación: 85%.

Velocidad estimada:  $50 \times 85\% = 42.5$  puntos de historias. Esto significa que en este sprint solo se pueden acumular historias de usuario hasta que la suma de sus estimaciones esté por debajo de 42.5.

Las historias seleccionadas para este sprint son las siguientes:

- 1- Historia de usuario Añadir un evento.
- 2- Historia de usuario Crear informe de integralidad.
- 3- Historia de usuario Crear informe de caracterización.
- 4- Historia de usuario Añadir estudiante.

5- Historia de usuario Añadir presidente de brigada.

En total, las estimaciones de las 5 historias de usuario suman 40 puntos. Por tanto, la velocidad final del sprint es de 40 puntos.

Aquí se muestran algunas de las tareas de este sprint. Para ver las restantes [Ver anexo 2](#)

### Sprint Backlog 1:

Tarea	
<b>Número tarea:</b> 1	<b>Nombre de la historia:</b> Añadir un evento
<b>Nombre tarea:</b> Añadir un evento	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación inicial:</b> 4
<b>Fecha inicio:</b> 13/10/2009	<b>Fecha fin:</b> 18/10/2009
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> crear un formulario que recoja los datos del evento entre los que están el nombre del evento, tipo y fecha.	

Tabla 2.28 Tarea Crear formulario Añadir un evento

Tarea	
<b>Número tarea:</b> 2	<b>Nombre de la historia:</b> Añadir un evento
<b>Nombre tarea:</b> Validar la correcta entrada de los datos y el llenado de los campos obligatorios del formulario Añadir evento.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación inicial:</b> 4
<b>Fecha inicio:</b> 19/10/2009	<b>Fecha fin:</b> 21/10/2009
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> definir como campos obligatorios del formulario: nombre del evento, tipo de evento y fecha. Verificar la correcta entrada de los datos.	

Tabla 2.29 Tarea Validar la correcta entrada de los datos y el llenado de los campos obligatorios del formulario Añadir evento

Tarea	
<b>Número tarea:</b> 3	<b>Nombre de la historia:</b> Crear informe de integralidad
<b>Nombre tarea:</b> Crear informe de integralidad	
<b>Tipo de tarea :</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 21/10/2009	<b>Fecha fin:</b> 26/10/2009
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> crear un formulario que recoja los datos del informe de integralidad, se mostrará un campo donde se elabore el informe de integralidad del estudiante teniendo en cuenta : el nombre del informe, el nombre del estudiante, la fecha y la descripción	

Tabla 2.30 Tarea elaborar formulario Crear informe de integralidad

### **Sprint 2 (Crear, Mdificar y Eliminar información)**

Cantidad de días que debe durar el sprint: 72. Del 7/1/2010 al 20/3/2010

Cantidad de días hombre: Son dos miembros en el equipo. Cada uno puede aportar en ese período un total de 60 días. Esto hace un total de 120 días hombre.

Factor de dedicación: 70%.

Velocidad estimada:  $120 \times 70\% = 84$  puntos de historias. Esto significa que en este sprint solo se pueden acumular historias de usuario hasta que la suma de sus estimaciones esté por debajo de 84. Las historias seleccionadas para este sprint son las siguientes:

- 1- Historia de usuario Añadir organizador FEU.
- 2- Historia de usuario Modificar evento.
- 3- Historia de usuario Modificar informe de integralidad.
- 4- Historia de usuario Modificar informe de caracterización.
- 5- Historia de usuario Modificar estudiante.
- 6- Historia de usuario Modificar presidente de brigada.

7- Historia de usuario Modificar organizador FEU.

8 - Historia de usuario Eliminar evento

9- Historia de usuario Eliminar informe de integralidad.

10-Historia de usuario Eliminar informe de caracterización.

11- Historia de usuario Eliminar estudiante.

12- Historia de usuario Eliminar presidente de brigada.

13- Historia de usuario Eliminar organizador FEU.

En total, las estimaciones de las 13 historias de usuario suman 80 puntos. Por tanto, la velocidad final del sprint es de 80 puntos.

Aquí se muestran algunas de las tareas de este sprint .Para ver las restantes [Ver anexo 3](#)

**Sprint Backlog 2:**

<b>Tarea</b>	
<b>Número tarea:</b> 11	<b>Nombre de la historia:</b> Modificar evento
<b>Nombre tarea:</b> Crear enlace Modificar a los formularios de los eventos	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 16/1/2010	<b>Fecha fin:</b> 21/1/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> insertar en el buscador de eventos un hipervínculo al formulario del evento seleccionado para modificar sus datos. Los datos modificados se guardan mediante el submit del formulario.	

Tabla 2.40 Tarea Crear enlace Modificar a los formularios de los eventos

Tarea	
<b>Número tarea:</b> 12	<b>Nombre de la historia:</b> Modificar informe de integralidad
<b>Nombre tarea:</b> Crear enlace Modificar a los formularios de los informes de integralidad	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 21/1/2010	<b>Fecha fin:</b> 25/1/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> insertar en el buscador de informes de integralidad un hipervínculo al formulario del informe seleccionado para modificar sus datos. Los datos modificados se guardan mediante el submit del formulario.	

Tabla 2.41 Tarea Crear enlace Modificar a los formularios de los informes de integralidad

Tarea	
<b>Número tarea:</b> 13	<b>Nombre de la historia:</b> Modificar informe de caracterización
<b>Nombre tarea:</b> Crear enlace Modificar a los formularios de los informes de caracterización	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 26/1/2010	<b>Fecha fin:</b> 31/1/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> insertar en el buscador de informes de caracterización un hipervínculo al formulario del informe seleccionado para modificar sus datos. Los datos modificados se guardan mediante el submit del formulario.	

Tabla 2.42 Tarea Crear enlace Modificar a los formularios de los informes de caracterización

### Sprint 3 (Buscar y Mostrar información)

Cantidad de días que debe durar el sprint: 31. Del 21/3/2010 al 21/4/2010

Cantidad de días hombre: Son dos miembros en el equipo. Cada uno puede aportar en ese período un total de 28 días. Esto hace un total de 56 días hombre.

Factor de dedicación: 90%.

Velocidad estimada:  $56 \times 90\% = 50.4$  puntos de historias. Esto significa que en este sprint solo se pueden acumular historias de usuario hasta que la suma de sus estimaciones esté por debajo de 50.4. Las historias seleccionadas para este sprint son las siguientes:

- 1- Historia de usuario Buscar listado de estudiantes por evento.
- 2- Historia de usuario Buscar listado de eventos en los que participa un estudiante.
- 3- Historia de usuario Mostrar listado de eventos.
- 4- Historia de usuario Mostrar listado de estudiantes.

En total, las estimaciones de las 4 historias de usuario suman 40 puntos. Por tanto, la velocidad final del sprint es de 40 puntos.

Aquí se muestran algunas de las tareas de este sprint. Para ver las restantes [Ver anexo 4](#)

### Sprint Backlog 3:

Tarea	
<b>Número tarea:</b> 28	<b>Nombre de la historia:</b> Buscar listado de estudiantes por evento
<b>Nombre tarea:</b> Crear buscador de estudiantes por evento	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 21/3/2010	<b>Fecha fin:</b> 28/3/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	



**Descripción:** crear un buscador que dado un evento liste los estudiantes que han participado en él y que permita filtrar por: tipo de evento, fecha y nombre del evento.

Tabla 2.52 Tarea Crear buscador de listado de estudiantes por evento

Tarea	
<b>Número tarea:</b> 29	<b>Nombre de la historia:</b> Buscar listado de eventos en los que participa un estudiante
<b>Nombre tarea:</b> Crear buscador de listado de eventos en los que participa un estudiante	
<b>Tipo de tarea :</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 29/3/2010	<b>Fecha fin:</b> 5/4/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> crear un buscador que dado un estudiante liste los eventos en los que han participado y que permita filtrar por: nombre, grupo y año.	

Tabla 2.53 Tarea Crear buscador de listado de eventos en los que participa un estudiante

### **Sprint 4 (Configuración de la interfaz y Seguridad del sistema)**

Cantidad de días que debe durar el sprint: 10. Del 22/4/2010 al 2/5/2010

Cantidad de días hombre: Son dos miembros en el equipo. Cada uno puede aportar en ese período un total de 10 días. Esto hace un total de 20 días hombre.

Factor de dedicación: 90%.

Velocidad estimada:  $20 \times 90\% = 18$  puntos de historias. Esto significa que en este sprint solo se pueden acumular historias de usuario hasta que la suma de sus estimaciones esté por debajo de 20. Las historias seleccionadas para este sprint son las siguientes:

- 1- Historia de usuario Autenticar usuarios.
- 2- Historia de usuario Administrar usuarios.
- 3- Historia de usuario Diseñar interfaz.

En total, las estimaciones de las 3 historias de usuario suman 20 puntos. Por tanto, la velocidad final del sprint es de 20 puntos.

Aquí se muestran algunas de las tareas de este sprint .Para ver las restantes [Ver anexo 5](#)

### Sprint Backlog 4:

Tarea	
<b>Número tarea:</b> 32	<b>Nombre de la historia:</b> Autenticar usuarios
<b>Nombre tarea:</b> Configurar la autenticación en el sistema por el dominio de la Universidad	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 22/4/2010	<b>Fecha fin:</b> 24/4/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> configuración del módulo LDAP con los parámetros de conexión de la UCI.	

Tabla 2.56 Tarea Configurar la autenticación en el sistema

Tarea	
<b>Número tarea:</b> 33	<b>Nombre de la historia:</b> Administrar usuarios
<b>Nombre tarea:</b> Crear y configurar el rol presidente de brigada.	
<b>Tipo de tarea:</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 24/4/2010	<b>Fecha fin:</b> 26/4/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> crear el rol presidente de brigada con los permisos correspondientes.	

Tabla 2.57 Tarea Crear y configurar el rol presidente de brigada

Tarea	
<b>Número tarea:</b> 34	<b>Nombre de la historia:</b> Administrar usuarios
<b>Nombre tarea:</b> Crear y configurar el rol organizador FEU.	
<b>Tipo de tarea :</b> Desarrollo	Estimación inicial: 4
<b>Fecha inicio:</b> 26/4/2010	<b>Fecha fin:</b> 28/4/2010
<b>Programador responsable:</b> Rafael Tizol y Yailin Sardiña	
<b>Descripción:</b> crear el rol organizador FEU con los permisos correspondientes	

Tabla 2.58 Tarea Crear y configurar el rol organizador FEU

## 2.5 Construcción de la propuesta de solución

### 2.5.1 Diagramas de pantallas por funcionalidades

Los diagramas de pantallas permiten esquematizar al detalle la interfaz gráfica y los contenidos de la aplicación.

- **Añadir un evento:** desde la página de inicio se podrá acceder al formulario que contiene los campos que deberán llenarse para insertar un evento, a través del botón Inicio se puede acceder al enlace Evento del cual se despliega la opción Añadir evento. Después de llenar los campos y enviar la información se mostrará una vista de los eventos recién guardados. Ver Anexo 6
- **Buscar listado de estudiantes por evento:** desde la página de inicio se podrá acceder a buscar un evento y mostrar los estudiantes que participan en él, a través de la opción Estudiantes por evento, en el Menú de búsqueda. Desde ahí se podrá filtrar por nombre del evento. Ver Anexo 7
- **Buscar listado de eventos en los que participa un estudiante:** desde la página de inicio se podrá acceder a buscar los eventos en los que ha participado un estudiante dado, a través

de la opción Eventos por estudiantes en el Menú de búsqueda. Desde ahí se podrá filtrar por nombre del estudiante. [Ver Anexo 8](#)

- Crear un informe de caracterización: desde la página de inicio se podrá acceder a insertar un informe a través del botón Inicio, donde aparece el enlace Informes del cual se despliega la opción Añadir informe de caracterización y ahí se llenarán los campos requeridos para el registro del informe. [Ver Anexo 9](#)
- Crear un informe de integralidad: desde la página de inicio se podrá acceder a insertar un informe a través del botón Inicio, donde aparece el enlace Informes del cual se despliega la opción Añadir informe de integralidad y ahí se llenarán los campos requeridos para el registro del informe. [Ver Anexo 10](#)
- Registro: desde la página de inicio a través del botón Inicio, se podrá acceder a la opción Registro en la que se podrá buscar al estudiante filtrando la búsqueda por grupo y nombre. Luego de seleccionado el estudiante se le podrá asignar un determinado evento o informe. [Ver Anexo 11](#)

### 2.5.2 Tratamiento de excepciones

Las excepciones manejadas en los diferentes procesos del sistema tienen que ver principalmente con la validación de los campos de los formularios. Entre los más comunes figura, la omisión de campos obligatorios, introducir una cadena de texto en un campo numérico, escribir direcciones en formatos incorrectos, entre otras. Esta validación se logra de manera robusta mediante los módulos CCK (Content Construction Kit) y JSTools (Javascript Tools), los cuales se encargan de la generación de formularios y componentes Javascript. Los módulos CCK como Date, Number, Address, Field y Text definen el tipo de los valores que serán entrados por los formularios. Como una validación adicional, el módulo JavaScript Validator permite mostrar alertas de error para estos tipos de datos antes que los datos sean enviados al servidor.

### 2.5.3 Estándares de codificación

Los estándares de codificación son las formas o estilos de programación que utilizan los programadores en la implementación de los diferentes sistemas. Los mismos deben estar

establecidos antes de comenzar a desarrollar un proyecto de software, de esta forma se asegura que todos los programadores trabajen de la misma manera y de que si en un futuro se quieren realizar cambios en el código de la aplicación los estándares indicarán como trabajar con el mismo. A continuación se describen algunos de los estándares que utiliza Drupal en el desarrollo de aplicaciones.

Indentación: Utilizar una indentación de dos espacios sin tabulaciones.

Estructuras de control de flujo: Incluye las sentencias if, while, for, switch. El nombre de la sentencia debe ir separado por un espacio del paréntesis de apertura para evitar confundirlo con las llamadas a funciones. Las llaves ( { } ) serán siempre utilizadas para delimitar los bloques de instrucciones de las estructuras, incluso cuando su uso sea opcional. De esta manera, aumenta la legibilidad del código y disminuyen los errores lógicos al agregarse nuevas instrucciones. if (condición1 || condición2) { acción1; }

- Llamadas a funciones: No deben existir espacios entre el nombre de la función, el primer paréntesis y el primer parámetro. Para el resto de los parámetros se insertará un espacio entre las comas y los parámetros. Tampoco existirán espacios entre el último parámetro, el paréntesis de cierre y el punto y coma (;).

```
$variable = funcion($parametro1, $parametro2 , $parametro3);
```

- En caso de varias asignaciones, se asignarán espacios extra para aumentar la legibilidad.  
\$nombre\_corto = funcion(\$parametro);  
\$nombre\_mucho\_mas\_largo = funcion(\$parametro);
- Declaración de funciones: Los parámetros con valores por defecto u opcionales irán al final de la lista de parámetros. Siempre se debe tratar de retornar una variable con un nombre significativo para la función.
- Arreglos: Los elementos y los operadores de asignación estarán separados por un espacio. En caso de que la sentencia sea mayor de los 80 caracteres o se realicen varias asignaciones en la sentencia, cada asignación se representará en una línea con un nivel de indentación. Puede observarse el siguiente fragmento de código de un módulo de Drupal:

```
$form['title'] = array(  
  '#type' => 'textfield',  
  '#title' => t('Title'),  
  '#size' => 60,
```

```
'#maxlength' => 128,  
'#description' => t (Título de la página.),  
);
```

- Concatenación de cadenas: Mantener un espacio a ambos lados de los operadores de concatenación (.) y (.=).
- Comentarios: Pueden utilizarse los símbolos (//) y (/ \* \*/) indistintamente. Los comentarios deben terminar con la puntuación adecuada.
- Nombres de funciones y miembros de clases: Las funciones deben nombrarse con letras minúsculas y las palabras separadas por un guión bajo (\_). Los nombres de los miembros privados de las clases deben comenzar con un guión bajo.
- Constantes: Las constantes serán nombradas completamente con letras mayúsculas y las partes del nombre serán separadas por un guión bajo.

### 2.6 Conclusiones

En este capítulo se describieron los procesos del negocio referente a la trayectoria estudiantil en la Facultad 1. Posibilitando un mejor entendimiento de dichos procesos, permitiendo identificar las dificultades presentes en su desarrollo y definiendo las funcionalidades con las que contará el sistema.

# 3

## Capítulo Implementación y prueba

### 3.1 Introducción

En este capítulo se representan los casos de prueba necesarios para dar cumplimiento a los requisitos planteados, mediante las pruebas de aceptación, las cuales brindan la posibilidad de comprobar si la aplicación funciona como desea el cliente lo cual es el objetivo principal que se persigue con la realización de este trabajo.

### 3.2 Pruebas

El proceso de desarrollo de las pruebas ayuda al cliente a clarificar y concretar la funcionalidad de la historia de usuario y favorece la comunicación entre el cliente y el equipo de desarrollo. Además, ayuda a identificar y corregir fallos u omisiones en las historias de usuario. También permite corregir errores en las ideas del cliente, por ejemplo encontrar resultados que el cliente espera encontrar en la implementación, pero para los que no existe ningún camino de ejecución que conduzca a ello. Facilita la identificación de historias adicionales que no fueran obvias para el cliente o en las que el cliente no hubiese pensado de no enfrentarse a dicha situación.

### 3.3 Pruebas de aceptación

Las pruebas de aceptación permiten confirmar que la HU ha sido implementada correctamente al final de cada iteración. Este período de prueba se conoce también como período de caja negra donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

## Capítulo 3: Implementación y Prueba

A continuación se representan las pruebas de aceptación realizadas para las HU del sistema:

Caso de prueba de aceptación:	
<b>Nombre:</b> Autenticarse correctamente	<b>Nombre de la historia:</b> Autenticar usuarios
<b>Descripción:</b> un usuario se conecta al sistema e intenta autenticarse insertando los datos correctamente.	
<b>Condiciones de ejecución:</b> el sistema debe tener habilitado el módulo LDAP_ UCI.	
<b>Entrada:</b> el usuario llena los campos nombre de usuario y contraseña y pulsa el botón Iniciar sesión.	
<b>Resultados:</b> el usuario entra al sistema, se le muestran únicamente las opciones a las que tiene permiso según su rol.	
<b>Evaluación de las pruebas:</b> Satisfactoria	

Tabla 3.1 Caso de prueba Autenticarse correctamente

Caso de prueba de aceptación:	
<b>Nombre:</b> Autenticarse incorrectamente	<b>Nombre de la historia:</b> Autenticar usuarios
<b>Descripción:</b> el organizador FEU o el presidente de brigada intentan autenticarse introduciendo sus datos de autenticación incorrectos.	
<b>Condiciones de ejecución:</b> el sistema debe tener habilitado el módulo LDAP_ UCI.	
<b>Entrada:</b> el usuario llena los campos nombre de usuario y contraseña y pulsa el botón Iniciar sesión.	
<b>Resultados:</b> el sistema muestra el mensaje: "Lo sentimos. No reconocemos el nombre de usuario o la contraseña."	
<b>Evaluación de las pruebas:</b> Satisfactoria	

Tabla 3.2 Caso de prueba Autenticarse incorrectamente

Caso de prueba de aceptación:	
<b>Nombre:</b> Control de acceso de usuarios no autorizado	<b>Nombre de la historia:</b> Autenticar usuarios
<b>Descripción:</b> un usuario que no está autorizado a trabajar en el sistema intenta autenticarse insertando un nombre de usuario y contraseña.	
<b>Condiciones de ejecución:</b> el sistema debe tener habilitado el módulo LDAP_ UCI.	
<b>Entrada:</b> el usuario llena los campos nombre de usuario y contraseña pulsando el botón Iniciar sesión, intenta acceder a un área en la que no está autorizado a entrar.	



<b>Resultados:</b> el sistema muestra el mensaje. "Acceso denegado. Usted no está autorizado para visitar esta página".
<b>Evaluación de las pruebas:</b> Satisfactoria.

Tabla 3.3 Caso de prueba Control de acceso de usuarios no autorizado

Caso de prueba de aceptación:	
<b>Nombre:</b> Listado de estudiantes por cada evento	<b>Nombre de la historia:</b> Buscar listado de estudiantes por evento
<b>Descripción:</b> el usuario después de autenticarse en el sistema selecciona la opción Estudiantes por evento que se encuentra en el Menú de búsqueda.	
<b>Condiciones de ejecución:</b> debe existir al menos un estudiante por evento.	
<b>Entrada:</b> el usuario selecciona la opción Estudiantes por evento.	
<b>Resultados:</b> se despliega una lista con todos los estudiantes que han participado en ese evento, además de posibilitar el filtrado por evento.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.4 Caso de prueba Listado de estudiantes por cada evento

Caso de prueba de aceptación:	
<b>Nombre:</b> Listado de eventos en los que participa un estudiante	<b>Nombre de la historia:</b> Buscar listado de eventos en los que participa un estudiante
<b>Descripción:</b> el usuario después de autenticarse en el sistema selecciona la opción Eventos por estudiantes que se encuentra en el Menú de búsqueda.	
<b>Condiciones de ejecución:</b> debe existir al menos un evento en el que haya participado un estudiante.	
<b>Entrada:</b> el usuario selecciona la opción Eventos por estudiantes.	
<b>Resultados:</b> se despliega la lista de eventos en los que ha participado un estudiante además de posibilitar el filtrado por nombre de usuario.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.5 Caso de prueba Listado de eventos en los que participa un estudiante

Caso de prueba de aceptación:

## Capítulo 3: Implementación y Prueba

<b>Nombre:</b> Listado de eventos	<b>Nombre de la historia</b> Mostrar listado de eventos
<b>Descripción:</b> el organizador FEU o el presidente de brigada después de autenticarse en el sistema seleccionan la opción Listado de eventos en el Menú de búsqueda.	
<b>Condiciones de ejecución:</b> debe existir al menos un evento.	
<b>Entrada:</b> el organizador FEU o el presidente de brigada seleccionan la opción Listado de eventos.	
<b>Resultados:</b> se despliega el listado de eventos además de brindar la posibilidad de filtrar por nombre, tipo y fecha del evento.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.6 Caso de prueba Listado de eventos

Caso de prueba de aceptación:	
<b>Nombre:</b> Listado de estudiantes	<b>Nombre de la historia</b> Mostrar listado de estudiantes
<b>Descripción:</b> el organizador FEU o el presidente de brigada después de autenticarse en el sistema pulsando el botón Inicio para seleccionar la opción Registro.	
<b>Condiciones de ejecución:</b> debe existir al menos un estudiante en el sistema.	
<b>Entrada:</b> el organizador FEU o el presidente de brigada seleccionan la opción Registros en el botón inicio, la cual para los usuarios con rol organizador FEU lista los usuarios con roles de estudiante y presidente de brigada, mientras que para los usuarios con rol presidente de brigada se muestran los estudiantes que pertenecen a su grupo.	
<b>Resultados:</b> se despliega la lista de estudiantes con la posibilidad de filtrar por el nombre de usuario.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.7 Caso de prueba Listado de estudiantes

Caso de prueba de aceptación:	
<b>Nombre:</b> Validación de los campos obligatorios del formulario Añadir evento	<b>Nombre de la historia:</b> Añadir evento
<b>Descripción:</b> el organizador FEU después que entra al sistema pulsa el botón Inicio para seleccionar la opción Evento y ahí en el menú desplegable Añadir evento. Seguidamente aparece una página que contiene un formulario en el que aparecen los campos que deben llenarse para registrar el evento, indicando con un asterisco rojo los campos que tienen carácter obligatorio. El usuario pulsa el botón Guardar.	

<b>Condiciones de ejecución:</b>
<b>Entrada:</b> el organizador FEU deja sin llenar uno o todos los campos obligatorios.
<b>Resultados:</b> el sistema muestra el mensaje: “El campo nombre del evento es obligatorio”, “El campo tipo de evento es obligatorio”, “El campo fecha del evento es obligatorio.”
<b>Evaluación de las pruebas:</b> Satisfactoria.

Tabla 3.8 Casos de prueba Validación de los campos obligatorios del formulario Añadir evento

<b>Caso de prueba de aceptación:</b>	
<b>Nombre:</b> Añadir datos de un evento	<b>Nombre de la historia:</b> Añadir evento
<b>Descripción:</b> el organizador FEU después que entra al sistema pulsa el botón Inicio para seleccionar la opción Evento y ahí en el menú desplegable Añadir evento. Posteriormente aparece una página que muestra el formulario que contiene los datos requeridos para agregar un evento. Añade los datos del evento y pulsa el botón Guardar.	
<b>Condiciones de ejecución:</b> se deben llenar los campos obligatorios del formulario.	
<b>Entrada:</b> el organizador FEU inserta los datos del evento: nombre del evento, fecha y tipo.	
<b>Resultados:</b> el sistema muestra el mensaje “Evento -nombre del evento- se ha creado” y debajo los datos guardados del evento.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.9 Casos de prueba Añadir datos de un evento

<b>Caso de prueba de aceptación:</b>	
<b>Nombre:</b> Modificar datos de un evento	<b>Nombre de la historia:</b> Modificar evento
<b>Descripción:</b> el organizador FEU después que entra al sistema pulsa el botón Inicio para seleccionar la opción Evento y ahí en el menú desplegable la opción Modificar o Eliminar eventos. Posteriormente aparece una página que muestra el listado de eventos y enlaces a la información del mismo además de posibilitar el filtrado de los eventos por nombre, tipo y fecha. Se selecciona el evento a modificar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del evento. Modifica los datos del evento y pulsa el botón Guardar.	
<b>Condiciones de ejecución:</b> debe existir al menos un evento en el sistema.	
<b>Entrada:</b> el organizador FEU debe insertar los datos nuevos.	
<b>Resultados:</b> el sistema actualiza los datos del evento y muestra el mensaje “Evento -nombre del evento- ha sido	

## Capítulo 3: Implementación y Prueba

actualizado”.
<b>Evaluación de las pruebas:</b> Satisfactoria.

Tabla 3.10 Caso de prueba Modificar datos de un evento

Caso de prueba de aceptación:	
<b>Nombre:</b> Eliminar datos de un evento	<b>Nombre de la historia:</b> Eliminar eventos
<b>Descripción:</b> el organizador FEU después que entra al sistema pulsa el botón Inicio para seleccionar la opción Evento y ahí en el menú desplegable la opción Modificar o Eliminar eventos. Posteriormente aparece una página que muestra el listado de eventos y enlaces a la información del mismo, además de posibilitar el filtrado de los eventos por nombre, tipo y fecha. Se selecciona el evento a eliminar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del evento. Se pulsa el botón Borrar al final del formulario.	
<b>Condiciones de ejecución:</b> debe existir al menos un evento registrado en el sistema.	
<b>Entrada:</b> el organizador FEU selecciona la opción Borrar de un evento y pulsa el botón Borrar que aparece en el mensaje de confirmación.	
<b>Resultados:</b> el sistema elimina el evento y muestra el mensaje: “Se ha eliminado -nombre del evento-”.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.11 Caso de prueba Eliminar datos de un evento

Caso de prueba de aceptación:	
<b>Nombre:</b> Validación de los campos obligatorios del formulario Crear un informe de integralidad	<b>Nombre de la historia:</b> Crear un informe de integralidad
<b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsan el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Añadir informe de integralidad. Seguidamente aparece una página que contiene un formulario en el que aparecen los campos que deben llenarse para registrar el informe, indicando con un asterisco rojo los campos que tienen carácter obligatorio. El usuario pulsa el botón Guardar.	
<b>Condiciones de ejecución:</b>	
<b>Entrada:</b> el organizador FEU o el presidente de brigada dejan sin llenar uno o todos los campos obligatorios.	
<b>Resultados:</b> el sistema muestra el mensaje: “El campo título del informe es obligatorio”, “El campo fecha de redacción es obligatorio.”	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.12 Casos de prueba Validación de los campos obligatorios del formulario Crear un informe de integralidad

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Crear informe de integralidad	<b>Nombre de la historia</b> Crear un informe de integralidad
<p><b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsan el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Añadir informe de integralidad. Posteriormente aparece una página que contiene un formulario en el que aparecen los campos que deben llenarse para elaborar el informe. Insertan los datos necesarios y pulsan el botón Guardar.</p>	
<p><b>Condiciones de ejecución:</b> Se deben llenar los campos obligatorios.</p>	
<p><b>Entrada:</b> el organizador FEU o el presidente de brigada llenan los campos: nombre del informe, fecha de redacción y descripción.</p>	
<p><b>Resultados:</b> el sistema muestra el mensaje: “Informe de integralidad -nombre del informe- se ha creado.” y debajo los datos del informe creado.</p>	
<p><b>Evaluación de las pruebas:</b> Satisfactoria.</p>	

Tabla 3.13 Caso de prueba Crear informe de integralidad

Caso de prueba de aceptación:	
<b>Nombre:</b> Modificar datos de un informe de integralidad	<b>Nombre de la historia</b> Modificar informe de integralidad
<p><b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsan el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Modificar o Eliminar informe de integralidad. Posteriormente aparece una página que contiene un listado de los informes creados y enlaces a la información del mismo, además de poder filtrarlos por título y fecha. Se selecciona el informe a modificar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del informe. Modifican los datos y pulsan el botón Guardar.</p>	
<p><b>Condiciones de ejecución:</b> Debe existir al menos un informe de integralidad registrado en el sistema.</p>	
<p><b>Entrada:</b> el organizador FEU o el presidente de brigada insertan los datos nuevos.</p>	
<p><b>Resultados:</b> el sistema actualiza los datos del evento y muestra el mensaje “El informe de integralidad -nombre del informe- ha sido actualizado”.</p>	
<p><b>Evaluación de las pruebas:</b> Satisfactoria.</p>	

Tabla 3.14 Caso de prueba Modificar datos de un informe de integralidad

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Eliminar datos de un informe de integralidad	<b>Nombre de la historia:</b> Eliminar informe de integralidad
<p><b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsán el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Modificar o Eliminar informe de integralidad. Posteriormente aparece una página que muestra el listado de los informes y enlaces a la información del mismo, además de poder filtrarlos por título y fecha. Se selecciona el informe a eliminar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del informe. Se pulsa el botón Borrar al final del formulario.</p>	
<p><b>Condiciones de ejecución:</b> debe existir al menos un informe de integralidad registrado en el sistema.</p>	
<p><b>Entrada:</b> el organizador FEU o el presidente de brigada seleccionan la opción Borrar de un informe y pulsán el botón Borrar que aparece en el mensaje de confirmación.</p>	
<p><b>Resultados:</b> el sistema elimina el informe y muestra el mensaje: "Se ha eliminado -nombre del informe-".</p>	
<p><b>Evaluación de las pruebas:</b> Satisfactoria.</p>	

Tabla 3.15 Caso de prueba Eliminar datos de un informe de integralidad

Caso de prueba de aceptación:	
<b>Nombre:</b> Validación de los campos obligatorios del formulario Crear un informe de caracterización	<b>Nombre de la historia:</b> Crear un informe de caracterización
<p><b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsán el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Añadir informe de caracterización. Seguidamente se muestra una página que contiene un formulario en el que aparecen los campos que deben llenarse para registrar el informe, indicando con un asterisco rojo los campos que tienen carácter obligatorio. El usuario pulsa el botón Guardar.</p>	
<p><b>Condiciones de ejecución:</b></p>	
<p><b>Entrada:</b> el organizador FEU o el presidente de brigada dejan sin llenar uno o todos los campos obligatorios.</p>	
<p><b>Resultados:</b> el sistema muestra el mensaje: "El campo título del informe es obligatorio", "El campo fecha de redacción es obligatorio."</p>	
<p><b>Evaluación de las pruebas:</b> Satisfactoria.</p>	

Tabla 3.16 Casos de prueba Validación de los campos obligatorios del formulario Crear informe de caracterización

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Crear informe de caracterización	<b>Nombre de la historia</b> Crear un informe de caracterización.
<b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsán el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Añadir informe de caracterización. Seguidamente aparece una página que contiene un formulario en el que aparecen los campos que deben llenarse para registrar el informe, indicando con un asterisco rojo los campos que tienen carácter obligatorio. El usuario pulsa el botón Guardar.	
<b>Condiciones de ejecución:</b> se deben llenar los campos obligatorios.	
<b>Entrada:</b> el organizador FEU o el presidente de brigada llena los campos: nombre del informe, fecha de redacción y descripción.	
<b>Resultados:</b> el sistema muestra el mensaje: "Informe de caracterización -nombre del informe- se ha creado.", y debajo los datos del informe creado.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.17 Caso de prueba Crear informe de caracterización

Caso de prueba de aceptación:	
<b>Nombre</b> Modificar datos de un informe de caracterización	<b>Nombre de la historia</b> Modificar informe de caracterización
<b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsán el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Modificar o Eliminar informe de caracterización. Posteriormente aparece una página que muestra el listado de informes y enlaces a la información del mismo además de poder filtrarlos por título y fecha. Se selecciona el informe a modificar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del informe. Se Modifican los datos y pulsán el botón Guardar al final del formulario	
<b>Condiciones de ejecución:</b> debe existir al menos un informe de caracterización registrado en el sistema.	
<b>Entrada:</b> el organizador FEU o el presidente de brigada insertan los datos nuevos.	
<b>Resultados:</b> el sistema actualiza los datos del informe y muestra el mensaje "Informe de caracterización -nombre del informe- ha sido actualizado".	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.18 Caso de prueba Modificar datos de un informe de caracterización

Caso de prueba de aceptación:

## Capítulo 3: Implementación y Prueba

<b>Nombre:</b> Eliminar datos de un informe de caracterización	<b>Nombre de la historia</b> Eliminar informe de Caracterización
<b>Descripción:</b> el organizador FEU o el presidente de brigada después que entran al sistema pulsán el botón Inicio para seleccionar la opción Informes y ahí en el menú desplegable la opción Modificar o Eliminar informe de caracterización. Posteriormente aparece una página que muestra el listado de informes y enlaces a la información del mismo además de poder filtrarlos por título y fecha. Se selecciona el informe a eliminar y seguidamente la opción Editar. Luego se muestra una página con el formulario que contiene los datos del informe. Se pulsa el botón Borrar al final del formulario	
<b>Condiciones de ejecución:</b> debe existir al menos un informe de integralidad registrado en el sistema.	
<b>Entrada:</b> el organizador FEU o el presidente de brigada seleccionan la opción Borrar de un informe y pulsa el botón Borrar que aparece en el mensaje de confirmación.	
<b>Resultados:</b> el sistema elimina el informe y muestra el mensaje: "Se ha eliminado -nombre del informe-".	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.19 Caso de prueba Eliminar datos de un informe de caracterización

Caso de prueba de aceptación:	
<b>Nombre:</b> Añadir los datos de un estudiante	<b>Nombre de la historia</b> Añadir estudiantes
<b>Descripción</b> el administrador del sistema después de autenticarse en el sistema selecciona la opción Actualizar sistema, en el menú principal y ahí en el menú desplegable la opción Actualizar registro de la base de datos. Posteriormente aparece una página que muestra la opción Actualizar. El sistema actualiza la base de datos de estudiantes.	
<b>Condiciones de ejecución:</b>	
<b>Entrada:</b> el administrador del sistema selecciona la opción actualizar	
<b>Resultados:</b> el sistema muestra un mensaje "Base de datos actualizada" y da la opción de ir a la página anterior a través del botón Atrás.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.20 Caso de prueba Añadir los datos de un estudiante

Caso de prueba de aceptación:
-------------------------------



<b>Nombre:</b> Modificar datos de un estudiante	<b>Nombre de la historia</b> Modificar estudiante
<b>Descripción:</b> el organizador FEU y el administrador después que entran al sistema seleccionan la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra el listado de usuarios y enlaces a la información del mismo además de poder filtrarlos por nombre y grupo. Se selecciona el usuario a modificar y seguidamente la opción Editar. Luego se muestra una página con los datos del usuario. Se modifican los datos y se pulsa el botón Guardar al final del formulario	
<b>Condiciones de ejecución:</b> debe existir al menos un estudiante en el sistema.	
<b>Entrada:</b> el organizador FEU o el administrador introducen los datos a modificar.	
<b>Resultados:</b> el sistema muestra un mensaje “Se han guardado los cambios”.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.21 Caso de prueba Modificar datos de un estudiante

Caso de prueba de aceptación:	
<b>Nombre:</b> Eliminar datos de un estudiante.	<b>Nombre de la historia</b> Eliminar estudiante.
<b>Descripción:</b> el organizador FEU y el administrador después que entran al sistema seleccionan la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra el listado de los usuarios y enlaces a la información del mismo, además de poder filtrarlos por nombre y grupo. Se selecciona el usuario a eliminar y seguidamente la opción Editar. Luego se muestra una página con los datos del usuario. Se pulsa el botón Borrar al final del formulario	
<b>Condiciones de ejecución:</b> debe existir al menos un estudiante en el sistema.	
<b>Entrada:</b> el organizador FEU o el administrador seleccionan la opción Borrar de un usuario y pulsan el botón Borrar que aparece en el mensaje de confirmación.	
<b>Resultados:</b> el sistema elimina el usuario y muestra el mensaje: “Se ha eliminado -nombre del usuario-”.	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.22 Caso de prueba Eliminar datos de un estudiante

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Añadir un presidente de brigada.	<b>Nombre de la historia</b> Añadir presidente de brigada.
<b>Descripción:</b> el organizador FEU después de entrar al sistema selecciona la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios registrados además de poder filtrarlos por nombre, grupo y rol. De la tabla selecciona al usuario que se le va a asignar el rol presidente de brigada en dependencia del grupo al que pertenece. A continuación, aparece una página nueva con los datos del usuario y la opción de asignarle el rol presidente de brigada. Luego pulsar en el botón Guardar	
<b>Condiciones de ejecución:</b> debe existir al menos un usuario en el sistema.	
<b>Entrada:</b>	
<b>Resultados:</b> el sistema muestra un mensaje "Se han guardado los cambios."	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.23 Añadir un presidente de brigada

Caso de prueba de aceptación:	
<b>Nombre:</b> Modificar datos de un presidente de brigada	<b>Nombre de la historia</b> Modificar presidente de brigada
<b>Descripción:</b> el organizador FEU después de entrar al sistema selecciona la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios registrados además de poder filtrarlos por nombre, grupo y rol. De la tabla selecciona el usuario que se va a modificar. A continuación, aparece una página nueva con los datos del usuario. Modifica los datos y pulsa en el botón Guardar	
<b>Condiciones de ejecución:</b> debe existir al menos un presidente de brigada en el sistema	
<b>Entrada:</b> el organizador FEU inserta los datos nuevos.	
<b>Resultados:</b> el sistema muestra un mensaje "Se han guardado los cambios."	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.24 Caso de prueba Modificar datos de un presidente de brigada

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Eliminar datos de un presidente de brigada	<b>Nombre de la historia</b> Eliminar presidente de brigada.
<b>Descripción:</b> el organizador FEU después de entrar al sistema selecciona la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios, además de poder filtrarlos por nombre, grupo y rol. De la tabla selecciona el usuario a eliminar. A continuación, aparece una página nueva con los datos del usuario. Luego pulsar en el botón Borrar	
<b>Condiciones de ejecución:</b> debe existir al menos un presidente de brigada en el sistema.	
<b>Entrada:</b> el organizador FEU selecciona la opción Borrar de un usuario y pulsa el botón Borrar que aparece en el mensaje de confirmación.	
<b>Resultados:</b> el sistema elimina el usuario y muestra el mensaje: "Se ha eliminado -nombre del usuario-".	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.25 Caso de prueba Eliminar datos de un presidente de brigada

Caso de prueba de aceptación:	
<b>Nombre:</b> Añadir los datos de un organizador FEU	<b>Nombre de la historia</b> Añadir organizador FEU.
<b>Descripción:</b> el organizador FEU o el administrador del sistema después de entrar al sistema seleccionan la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios registrados, además de la posibilidad de filtrarlos por nombre, grupo y rol. De la tabla selecciona al usuario que se le asignará el rol de organizador FEU. A continuación, aparece una página nueva con los datos del usuario la opción de asignarle el rol organizador FEU. Luego pulsar en el botón Guardar	
<b>Condiciones de ejecución:</b> debe existir al menos un usuario en el sistema.	
<b>Entrada:</b>	
<b>Resultados:</b> el sistema muestra un mensaje "Se han guardado los cambios"	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.26 Caso de prueba Añadir los datos de un organizador FEU

## Capítulo 3: Implementación y Prueba

Caso de prueba de aceptación:	
<b>Nombre:</b> Modificar datos de un organizador FEU	<b>Nombre de la historia</b> Modificar organizador FEU
<b>Descripción:</b> el organizador FEU o el administrador después de entrar al sistema seleccionan la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios, además de poder filtrarlos por nombre, grupo y rol. De la tabla seleccionan el usuario a modificar. A continuación, aparece una página nueva con los datos del usuario a modificar. Luego pulsan en el botón Guardar	
<b>Condiciones de ejecución:</b> debe existir al menos un organizador FEU en el sistema	
<b>Entrada:</b> el organizador FEU o el administrador insertan los datos nuevos.	
<b>Resultados:</b> El sistema muestra un mensaje "Se han guardado los cambios"	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.27 Caso de prueba Modificar datos de un organizador FEU

Caso de prueba de aceptación:	
<b>Nombre:</b> Eliminar datos de un organizador FEU	<b>Nombre de la historia</b> Eliminar organizador FEU
<b>Descripción:</b> el organizador FEU o el administrador después de entrar al sistema seleccionan la opción Administrar datos en el menú Administración de usuarios. Posteriormente aparece una página que muestra un listado de los usuarios, además de poder filtrarlos por nombre, grupo y rol. De la tabla seleccionan el usuario a eliminar. A continuación, aparece una página nueva con los datos del usuario. Luego pulsar en el botón Borrar	
<b>Condiciones de ejecución:</b> debe existir al menos un organizador FEU en el sistema.	
<b>Entrada:</b> el organizador FEU selecciona la opción Borrar de un usuario y pulsa el botón Borrar que aparece en el mensaje de confirmación.	
<b>Resultados:</b> el sistema elimina el usuario y muestra el mensaje: "Se ha eliminado -nombre del usuario-".	
<b>Evaluación de las pruebas:</b> Satisfactoria.	

Tabla 3.28 Caso de prueba Eliminar datos de un organizador FEU

3.4 Resultado de las pruebas

Iteración 1				
Elemento	No	No conformidad	Aspecto correspondiente	Importancia
HU 7 HU 8 HU 9	1	El usuario presidente de brigada tiene acceso a los datos y registro de información de todos los grupos de la Facultad 1.	HU- Añadir un evento HU-Modificar un evento HU- Eliminar un evento	Significativo
HU 7	2	Dificultad en el autocompletamiento de la información de los campos, nombre del evento y nombre del informe, en los formularios de Asignación de eventos y de Informes.	HU-Añadir un evento	Significativo
HU 2	3	El usuario estudiante no tiene acceso a consultar información en el sistema.	HU- Administrar usuario	Significativo
HU 7 HU 10	4	Errores ortográficos en formulario Crear evento y Crear informe de integralidad.	HU- Crear evento HU- Crear informe de integralidad	No significativo
HU 4	5	Errores en los formularios de búsqueda.	HU- Buscar Listado de eventos por estudiante	Significativo
Iteración 2				
HU 13	1	Errores ortográficos en formulario Crear informe de caracterización y en la página principal.	HU- Crear informe de caracterización	No significativo
HU 7	2	El sistema no permite asignar más de un evento a un estudiante.	HU- Añadir un evento	Significativo

HU 3	3	Opciones de búsqueda reducidas en el formulario de Búsqueda: Listado de estudiantes por evento.	HU- Buscar Listado de estudiantes por evento	Significativo
HU 25	4	Menú de trabajo del sistema poco funcional.	- Historia técnica Diseñar interfaz	No significativo
<b>Iteración 3</b>				
HU 25	1	Falta de información en los formularios de búsqueda.	- Historia técnica Diseñar interfaz	Significativo
HU 26	2	Interfaz de la aplicación inapropiada.	- Historia técnica Diseñar interfaz	No significativo
HU 4	3	Ausencia de un formulario para registrar datos relevantes de la participación de los estudiantes en eventos.	HU- Buscar Listado de eventos por estudiante	Significativo
HU 4	4	Error en el formulario de registros al estudiante.	HU- Buscar Listado de eventos por estudiante	Significativo
<b>Iteración 4</b>				
HU 5	1	Errores ortográficos en formulario de búsqueda de eventos.	- Mostrar listado de eventos	No significativo
HU 13	2	Errores ortográficos en formulario de búsqueda de informes de caracterización.	HU- Crear informe de caracterización	No significativo
HU 10	3	Errores ortográficos en formulario de búsqueda de informes de integralidad.	HU- Crear informe de integralidad	No significativo

### 3.5 Conclusiones

La realización de este capítulo permitió comprobar que todas las funcionalidades del sistema están bien realizadas, debido a que los resultados de todas las pruebas fueron satisfactorios, cumpliendo con el objetivo para el cual fue diseñado.

### Conclusiones generales

- Después del estudio realizado de los diferentes sistemas de gestión relacionados con la trayectoria estudiantil a nivel internacional y nacional, se demostró la necesidad de desarrollar en la Facultad 1 de la UCI un sistema que gestione la trayectoria estudiantil vinculada a los eventos de investigación, cultura y deportes, satisfaciendo las necesidades de los clientes.
- Con la utilización de las tecnologías, metodologías, lenguajes y herramientas definidas por los desarrolladores de la aplicación se logró la implementación del sistema de seguimiento a la trayectoria estudiantil en las esferas de investigación, cultura y deportes para la Facultad 1 de la UCI dándole solución a los problemas existentes con el manejo de la información referente al proceso de caracterización de cada estudiante garantizando un control fiable de la información.
- Se aplicaron un conjunto de pruebas para verificar el funcionamiento correcto del sistema, que validaron la calidad y el cumplimiento de los objetivos planteados, posibilitando obtener un producto con calidad.

### **Recomendaciones**

- Extender el alcance del sistema a todas las facultades de la UCI.
- Exportar datos al formato Microsoft Office Excel.
- Permitir que los profesores interactúen con el sistema para que puedan llevar un control de la participación de los estudiantes en los diferentes eventos.



## Referencia bibliográfica

1. Beltrán, e.M., *Indicadores de Gestión*. 1999.
2. Vinent, B.G.y.S., *Necesidad de formar profesionales para la investigación científica en biodiversidad, en Contribuciones a las Ciencias Sociales*. abril 2010.
3. Peñaranda, C.A., *RevistaCiencias.com*. 10 de March de 2006.
4. ALBA, q.d.t.p., *Sistema Libre de Gestión Educativa*.
5. *Sistema Integral de Gestión Académica*. 2008; Available from: <<http://www.gestion.uco.es/gestion/aplicaciones/siga>>
6. *Educar.org y eAprender.org*. 2009; Available from: <http://portal.educar.org/foros/agora-software-de-gestion-docente>
7. *Instituto de Altos Estudios Nacionales*. 2009; Available from: <http://www.iaen.edu.ec/wordpress>
8. *Facultad de Matemática, Astronomía y Física*. Available from: [http://www.famaf.unc.edu.ar/institucional/secretarias/secretarias.html#inst\\_secreasunt\\_estudiant](http://www.famaf.unc.edu.ar/institucional/secretarias/secretarias.html#inst_secreasunt_estudiant)
9. Dariel Enrique Tamayo Palma, Y.A.P., *Análisis y Diseño de los procesos de gestión de Carreras para el Sistema Akademos v2.0*.
10. Frank Benavides Dalmendray, N.M.H., Elisabel Pérez Urbay, Dianly Santiler Álvarez, Norges Sánchez. *Sistema Automatizado para la Gestión Académica*. 2006; Available from: <http://www.akademos.uci.cu>
11. Hernández, C.L.C.P.E.N., *AJAK: JSF + AJAX para Kainos*. 2008.
12. Patricio Letelier, M.C.P. *Metodologías ágiles para el desarrollo del software: Extreme*
13. Kniberg, H. *Scrum y XP desde las trincheras*. 2007.
14. *Unified Modeling Language*. 1997-2010; Available from: <http://www.uml.org>
15. *El Rincón del Vago*. 1998; Available from: <http://html.rincondelvago.com/herramientas-case.html>
16. Available from: <http://www.visual-paradigm.com/documentation>
17. Castillo, A. 2006.
18. Eich, B., *pergamino virtual*. 1998-2009
19. *Drupal Hispano*. 15 de Noviembre de 2008.
20. Alvarez, S., *Manual de iniciación a la programación*.
21. Huesca, J.M.P., 16-08-2005.

22. *editum.org*. 2007.
23. *Apache Software Foundation*. 29 de Septiembre de 2008; Available from:  
<http://www.apache.org>
24. *diario linux*. 2010.

**Bibliografía consultada**

1. Apache Software Foundation. Apache. [En línea] [Citado el: 29 de Septiembre de 2008.]
2. Drupal Hispano. [En línea] [Citado el: 15 de Noviembre de 2008.] <http://drupal.org.es/>
3. educación, S.A, 2000. P.464.
4. <http://www.apache.org>
5. Jacobson, Booch, Rumbaugh. El Proceso Unificado de Desarrollo de Software. Madrid, Pearson
6. Manuel A.Quert Gómez, Migdelys Sablón Loforte. Sistema de Gestión de Incidencias para la Universidad de las Ciencias Informáticas. Ciudad de la Habana 2009
7. Patricio Letelier, María Carmen Penadés. Metodologías ágiles para el desarrollo del software: Extreme. MANUAL DE PHP. [En línea] [Citado el: 26 de Septiembre de 2008.] <http://www.manualdephp.es/>
8. Paz, Jose Dapena. Desarrollo en comunidad con Extreme Programming. España: s.n.P.11.
9. Peralta, Adriana. Metodología Scrum (Universidad ORT Uruguay, Facultad de Ingeniería). Montevideo: s.n., 2003. P.9.
10. Programming (XP). Valencia: s.n.P.16.

### Glosario de términos

**CMS:** Content Management System (Sistema de gestión de contenidos) es un programa que se refiere usualmente a sitios web que tienen sistemas de administración que permiten la fácil creación y edición de contenidos, como páginas nuevas, noticias, etc.

**CSS:** Cascading Style Sheets (hojas de estilo en cascada) Conjunto de instrucciones HTML que definen la apariencia de uno o más elementos de un conjunto de páginas web con el objetivo de uniformizar su diseño.

**HTML:** HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**HTTP:** HyperText Transfer Protocol Protocolo de Transferencia de Hipertexto. HTTP es un protocolo con la ligereza y velocidad necesaria para distribuir y manejar sistemas de información hipermedia.

**IDE:** Integrated Development Environment (Entorno de desarrollo integrado) es un programa compuesto por un conjunto de herramientas para un programador.

**JavaScript:** Lenguaje desarrollado por Netscape y aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo HTML.

**LDAP:** Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

**RUP:** Rational Unified Process (Proceso Unificado de Desarrollo) es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología. Tomado de: Jacobson, Booch, Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Madrid, Pearson educación, S.A, 2000. P.464.

**SGBD:** DataBase Management System (Sistemas de gestión de base de datos) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**UML:** Unified Modeling Language (Lenguaje Unificado de Modelado) El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

**XML:** Extensible Markup Language (lenguaje de marcas) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

[http://www.babylon.com/definicion/sistema\\_de\\_gesti%C3%B3n\\_de\\_base\\_de\\_datos/Spanish](http://www.babylon.com/definicion/sistema_de_gesti%C3%B3n_de_base_de_datos/Spanish)

**GPL:** Licencia pública general.

**CASE:** Ayuda computacional de la ingeniería del software.

**DOM:** Modelo de objeto de documentos.