



**Universidad de las Ciencias Informáticas**  
**Facultad 5**

**Título:** *“Sistema para la planificación y control de la  
residencia estudiantil en la Facultad 5.”*

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

**Autor:** Erlan Rodríguez Rodríguez

**Tutores:** Msc. Zoraida Fernández Guevara  
Ing. Yoandys Hernández Alvarez

La Habana, 2014

“Año 56 de la Revolución”

Curso 2013-2014

## *Declaración de autoría*

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

**Autor:** Erlan Rodríguez Rodríguez

---

**Tutora:** MSc. Zoraida Fernández Guevara

---

**Tutor:** Yoandys Hernández Alvarez

### Generales de la Tutora:

**Nombre y apellidos:** Zoraida Fernández Guevara

**Edad:** 55

**Ciudadanía:** cubana

**Institución:** Universidad de las Ciencias Informáticas

**Título:** Filóloga en Lengua Rusa, Licenciada en Educación especialidad Lengua Inglesa, Máster en Tecnología de los procesos educativos

**E-mail:** zorlis@uci.cu

**Síntesis del tutor:** Es Filóloga en Lengua Rusa, licenciada en Educación en la especialidad de Lengua Inglesa y Máster en Tecnología de los procesos educativos. Posee 30 años de experiencia en la enseñanza superior. Ha impartido asignaturas tales como Idioma Ruso e Inglés, Comunicación Profesional, Formación Pedagógica, Multimedia, para estudiantes no filólogos. Ha impartido un gran número de cursos de postgrado, varios de ellos preparados por ella. Ha participado en investigaciones pedagógicas. Ha presentado trabajos en eventos nacionales e internacionales. Ha publicado artículos y ponencias en memorias de eventos.

### Generales del Tutor:

**Nombre y apellidos:** Ing. Yoandys Hernández Álvarez.

**Edad:** 25

**Ciudadanía:** cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en Ciencias Informáticas.

**E-mail:** yhalvarez@uci.cu

**Síntesis del tutor:** Graduado en la Universidad de las Ciencias Informáticas en el año 2012 con 4 años de experiencia en trabajos productivos reales de carácter nacional e internacional y analista de software del proyecto SCADA – Guardián del Alba.

# Agradecimientos

*A mis padres Vivian y Erlan...*

*Por estar presentes en cada etapa de mi vida brindándome su entrega incondicional; todo lo que soy es el fruto de su sacrificio, amor infinito y apoyo.*

*A mi esposa...*

*Por ser mi sustento y apoyo durante todo el tiempo que he estado en la universidad, brindándome todo su amor el cual espero tener por siempre.*

*A toda mi familia y amigos...*

*Por siempre estar tan atentos con mis estudios y ayudarme en todo lo que siempre estuvo a su alcance.*

*A mis tutores...*

*Por su ayuda incondicional.*

*A mis abuelitos que ya no están...*

*Pero sé que hubieran estado orgullosos.*

## *Dedicatoria*

*Pensé dedicarle el trabajo de diploma a muchas personas, pero creo que ellas entenderán porqué quiero regalarle esta posición especial a mi madre, que es lo que más quiero en el mundo, por su entrega incondicional y por estar siempre presente acogiéndome en su regazo durante las adversidades de la vida; dedicarle todo este tiempo de esfuerzo que he empleado para poder regalarle la alegría de graduarme como ingeniero en ciencias informáticas. Aunque creo que nada de lo que haga en mi vida va a ser suficiente para agradecerle todo lo que me ha entregado. Quisiera hacerle saber que me he esforzado para que ella y mi padre (espero que el viejo no se me ponga celoso) se sientan orgullosos de su único hijo (hasta el momento).*

***Erlan Rodríguez Rodríguez***

En los últimos años se ha venido experimentando un acelerado avance de las Tecnologías de la Información y las Comunicaciones (TIC); Cuba no se encuentra ajena a este proceso, y dado el elevado volumen de información que se genera actualmente en las entidades cubanas, su incorporación en la sociedad es de vital importancia.

La Universidad de las Ciencias Informáticas (UCI), como parte del proceso de informatización de cada una de sus áreas, ha incorporado un grupo de sistemas informáticos que son capaces de gestionar determinada información, la cual pudiera ser usada en la posterior toma de decisiones por diferentes administrativos. La presente investigación tiene como objetivo principal, desarrollar una aplicación web para facilitar el flujo e intercambio de información de la residencia estudiantil de la Facultad 5, además realizar la planificación de distintos procesos que se desarrollan en el área. Dicho sistema le permite tanto a las instructoras como al Vicedecanato de Residencia y Extensión Universitaria poder llevar un control estricto de cada una de las actividades de los estudiantes en la beca. Para realizar el diseño de la misma se utilizó la metodología XP, utilizando PHP como lenguaje de programación en el servidor y lenguaje del lado del cliente JavaScript, HTML y CSS. Se define como servidor Web: Apache, PHP Storm como IDE de desarrollo y como framework el Yii Framework. Como sistema gestor de base de datos MySQL.

**Palabras clave:** Gestión de la información, planificación, residencia estudiantil.

# Índice de Contenidos

<b>INTRODUCCIÓN</b> .....	<b>5</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.</b> .....	<b>2</b>
1.1 Introducción.....	2
1.2 Antecedentes.....	2
1.3 Gestión de la Información.....	2
1.4 Sistemas de gestión de la Información.....	3
1.5 Planificación de procesos. ....	3
1.6 Sistemas informáticos existentes en la gestión residencial universitaria. ....	4
1.6.1 Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Río “Hermanos Saiz Montes de Oca”. (SAICRE).....	4
1.6.2 Sistema de Gestión de Información de la Facultad 8 (SGIF 8).....	5
1.6.3 DATAFEU.....	6
1.7 Metodología de desarrollo de software .....	6
1.7.1 Metodología XP ( <i>Extreme Programming</i> , Programación Extrema).....	7
1.7.2 Metodología RUP ( <i>Rational Unified Process</i> ) .....	8
1.7.3 Fundamentación del uso de la metodología XP .....	9
1.8 Frameworks de desarrollo:.....	9
1.8.1 Symfony.....	10
1.8.2 CodeIgniter.....	10
1.8.3 Yii PHP Framework .....	11
1.9 Lenguajes de Programación para aplicaciones web .....	12
1.9.1 <i>Hyper Text Markup Language 5</i> (HTML 5).....	12
1.9.2 <i>JavaScript</i> .....	13
1.9.3 <i>Cascading Style Sheets</i> (CSS).....	13
1.9.4 <i>Hipertext Pre-processor</i> (PHP) .....	14
1.10 Sistema de Gestión de Bases de Datos (SGBD). ....	14
1.10.1 PostgreSQL 8.3 .....	15
1.10.2 MySQL. ....	15
1.11 Servidor Web.....	16
1.12 IDE's de desarrollo.....	16
1.12.1 Aptana Studio .....	17

1.12.2	Zend Studio .....	17
1.12.3	PHP Storm.....	18
1.13	Consideraciones parciales.....	19
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>		<b>20</b>
2.1	Introducción .....	20
2.2	Descripción de la solución propuesta.....	20
2.3	Especificación de los requerimientos .....	20
2.3.1	Requerimientos No Funcionales .....	21
2.4	Roles de usuarios .....	22
2.5	Fase de Exploración y Planificación.....	23
2.5.1	Historias de usuarios .....	23
2.5.2	Planificación de la entrega.....	25
2.5.3	Estimación de esfuerzo por historia de usuario.....	25
2.5.4	Plan de Iteraciones .....	27
2.5.5	Plan de Entregas .....	28
2.5.6	Arquitectura del sistema .....	28
2.6	Diseño del sistema.....	30
2.6.1	Tarjetas CRC.....	30
2.7	Fase de Implementación.....	35
2.8	Diagrama de despliegue.....	43
2.9	Consideraciones parciales.....	45
<b>CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN .....</b>		<b>46</b>
3.1	Introducción .....	46
3.2	Pruebas de software .....	46
3.2.1	Pruebas de Aceptación.....	46
3.3	Consideraciones parciales.....	52
<b>CONCLUSIONES GENERALES.....</b>		<b>53</b>
<b>RECOMENDACIONES .....</b>		<b>54</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>55</b>
<b>BIBLIOGRAFÍA.....</b>		<b>58</b>
<b>ANEXOS.....</b>		<b>60</b>



# Índice de Tablas

Tabla 1: Requerimientos No Funcionales del sistema.....	21
Tabla 2: Roles en el sistema.....	22
Tabla 3: Historia de Usuario - Autenticar usuario .....	24
Tabla 4: Historia de Usuario - Modificar usuario.....	24
Tabla 5: Historia de Usuario - Eliminar usuario .....	25
Tabla 6: Estimación de esfuerzo por historia de usuario .....	25
Tabla 7: Plan de Iteraciones .....	27
Tabla 8: Plan de entregas a los clientes.....	28
Tabla 9: Representación genérica de una tarjeta CRC.....	31
Tabla 10: Tarjeta CRC – Ususer.....	31
Tabla 11: Tarjeta CRC – Instructora .....	31
Tabla 12: Tarjeta CRC – Edificio.....	32
Tabla 13: Tarjeta CRC – Apto.....	32
Tabla 14: Tarjeta CRC – Estudiante .....	32
Tabla 15: Tarjeta CRC – Limpieza .....	33
Tabla 16: Tarjeta CRC – Cuartelería.....	33
Tabla 17: Tarjeta CRC – Evaluación.....	33
Tabla 18: Tarjeta CRC – Llave .....	34
Tabla 19: Tarjeta CRC – Medio .....	34
Tabla 20: Tarjeta CRC – Parada.....	34
Tabla 21: Tarjeta CRC – Tarea.....	35
Tabla 22: Tarea - Configuración del módulo de autenticación .....	36
Tabla 23: Tarea - Gestión de instructoras .....	36
Tabla 24: Tarea - Gestión de edificios .....	37
Tabla 25: Tarea - Gestión de apartamentos.....	37
Tabla 26: Tarea - Gestión de estudiantes y limpieza .....	38
Tabla 27: Tarea - Gestión de cuarterías.....	39
Tabla 28: Tarea - Gestión de evaluación .....	39
Tabla 29: Tarea - Gestión de llaves .....	40
Tabla 30: Tarea - Gestión de medios.....	41
Tabla 31: Tarea - Gestión de paradas de beca .....	41
Tabla 32: Tarea - Generar reportes .....	41

## INTRODUCCIÓN

En la actualidad con el uso de las Tecnologías de la Información y Comunicaciones (TIC), se ha logrado integrar la planificación como un proceso de mejora en la organización y agilización de labores, creándose una tendencia al manejo de sistemas informáticos capaces de realizar programaciones. Ayudando en la posterior toma de decisiones, el eficaz manejo de la información y la gestión de recursos tanto tecnológicos como humanos.

En Cuba, la informatización de sus instituciones ha alcanzado un gran auge, donde el principal objetivo radica, en lograr buena calidad y rapidez a la hora de procesar los datos que se manejan en el desarrollo de sectores tanto productivos como educacionales. Con el avance de la informática y las comunicaciones, muchas corporaciones se ven obligadas a la digitalización de la mayoría de sus procesos, desarrollándose de cada uno de ellos varios software de adquisición de datos. Las universidades, no están exentas de este desarrollo, ya que es muy factible la gestión de la información cuando se cuenta con algún sistema automatizado.

La Universidad de las Ciencias Informáticas (UCI), tiene entre sus objetivos la informatización de todas sus áreas, con el propósito de mejorar el funcionamiento de los procesos que tienen lugar en el centro. La residencia estudiantil, identificada como una de las áreas de resultados claves en el desarrollo creciente de la universidad, se sustenta sobre la base de la planificación, organización y control de todas las actividades no curriculares realizadas por los estudiantes que en ella conviven.

La Facultad 5 de la UCI y en especial su Vicedecano de Extensión en conjunto con las instructoras de los edificios estudiantiles, llevan un estricto control de todo el desempeño del educando, mostrando un seguimiento de todas sus actividades fuera del ámbito docente.

Actualmente la secretaria del vicedecano de residencia recibe toda la documentación en copia dura (papel) o en formato digital (Excel, Word, entre otros) de la participación en las actividades y del cumplimiento de los deberes de los becarios, tales como: cuartería y paradas de beca que le hacen llegar las instructoras de la residencia, los encargados de esta área por la FEU y los profesores asignados como responsables en estos tipos de controles. Sucede de igual modo con los reportes de estado de las afectaciones tanto de mantenimiento como tecnológicas. Estos datos llegan a la secretaria vía correo, a través de dispositivos de almacenamiento portables o en las plantillas impresas creadas con estos fines. El procesamiento de estos datos se hace a medida que se van recepcionando, lo que provoca atraso en la disponibilidad de la información obtenida y que en ocasiones se creen ficheros duplicados por no llevar un correcto control de versiones. Esto provoca

que no se cuente con la información actualizada disponible que permita medir correctamente el desempeño de los estudiantes en el edificio. Todo ello conlleva a que el volumen de búsqueda sea cada vez mayor, lo cual provoca que la misma en determinado momento se torne un poco lenta.

Debido a la situación existente se plantea como **Problema Científico**: ¿Cómo contribuir a la gestión de la información en el área de residencia y extensión universitaria de la Facultad 5?

Para dar solución al problema planteado se determina como **Objeto de estudio** de la investigación: Sistemas de información para los procesos de extensión universitaria. Como **Objetivo general de la investigación** se define: Desarrollar una aplicación informática basada en una plataforma Web, que viabilice la gestión de la información en el área de residencia y extensión universitaria de la Facultad 5.

Todo esto conlleva a determinar como **Campo de acción**: La información del área de extensión y residencia de la Facultad 5.

Para lograr el cumplimiento del objetivo general se proponen las siguientes **Tareas investigativas**:

- Elaboración del marco teórico a partir del estado del arte actual referente al tema.
- Realización de un análisis para la selección de la tecnología, metodología de desarrollo de software, lenguaje de programación, gestor de base de datos y herramientas a utilizar.
- Modelación de las condiciones actuales que rigen el proceso de gestión de información y planificación de la residencia estudiantil de la Facultad 5.
- Implementación de la aplicación web que brinde la solución al problema planteado.
- Realización de pruebas y validar el cumplimiento de los requerimientos.

Con el objetivo de alcanzar la solución deseada y de facilitar el desarrollo de la investigación se utilizaron los siguientes **métodos de investigación**:

#### **Métodos teóricos:**

- El método Analítico-Sintético se utiliza con el fin de analizar libros, páginas web y otras bibliografías para buscar los elementos más importantes que se relacionan con esta investigación.
- El método Histórico-Lógico se utiliza con el objetivo de verificar teóricamente cómo ha evolucionado el tema tratado en esta investigación.
- El método Modelación se utiliza con el objetivo de reproducir la interacción de los objetos en la vida real mediante la creación de modelos.

## **Métodos empíricos:**

- La Observación: sirvió para valorar las tecnologías y productos existentes, además de comparar con homólogos y así definir la propuesta, dando a conocer de manera detallada lo que se desea, hace falta hacer y cómo hacerlo.
- La Entrevista: se utilizó con el objetivo de obtener información valiosa para el desarrollo de la investigación.
- La Encuesta: se utilizó con el objetivo de conocer la opinión de las personas para conformar la idea de cómo debería ser el sistema de gestión de la residencia en Facultad 5.

## **Estructura capitular**

**Capítulo 1: “Fundamentación teórica”:** Se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se realiza un análisis de las metodologías y herramientas de desarrollo que se pueden utilizar y se justifica la selección de cada una de ellas.

**Capítulo 2: “Características del sistema”:** Se brindará una descripción de las principales características del sistema propuesto, el análisis de la información vinculada con el objeto de estudio y las características asociadas al dominio de la aplicación. Al mismo tiempo se identificarán los actores a partir de los roles de usuarios definidos. Se ofrecerán detalles de los requisitos funcionales denominados historias de usuario por la metodología de desarrollo seleccionada, así como los requisitos no funcionales.

**Capítulo 3: “Validación de la solución”:** Se definen los tipos de pruebas y los casos de prueba que se le realizarán al software, con el objetivo de poder verificar el cumplimiento de las expectativas del cliente. Se valida además la solución propuesta con un cuestionario de preguntas a una muestra de usuarios, con el objetivo de conocer el nivel de satisfacción de los mismos con el sistema.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

### 1.1 Introducción.

En este capítulo se realiza el estudio del arte correspondiente a diversos temas relacionados con la gestión de la información y la incorporación de las TIC en los procesos de planificación, y como ellos han influido en lograr un sistema educacional centrado en el uso de las tecnologías, así como las diferentes aplicaciones existentes vinculadas de una forma u otra con el tema de la investigación. Además, se describirá un conjunto de tecnologías, metodologías, herramientas, lenguajes de programación y Sistemas Gestores de Base de Datos (SGBD) más utilizados, para definir y seleccionar los que se utilizarán en el desarrollo de este trabajo de diploma.

### 1.2 Antecedentes

La Universidad de las Ciencias Informáticas cuenta con importantes servicios informatizados, uno de los principales es [gestionuniversitaria.uci.cu](http://gestionuniversitaria.uci.cu), con el cual se gestiona la información relacionada con la docencia de los estudiantes de toda la universidad. También se hace imprescindible la existencia de servicios que controlen la información concerniente al área de Extensión y Residencia. Existen varias aplicaciones que de una forma u otra gestionan los procesos en la misma, pero no incluyen algunos de esta área, tales como: cuartería, paradas de beca con sus respectivas evaluaciones, gestión y planificación de la limpieza diaria y general, así como un grupo de reportes e informes que se generan en el espacio.

### 1.3 Gestión de la Información

La gestión de la información es un amplio campo de estudio, pero esencialmente es el proceso de gestión de datos de una organización o empresa, es el proceso de desarrollo de prácticas y procedimientos relativos a los datos (1). La correcta gestión de la información conoce, incorpora y vincula todos los tipos de datos, de todas las áreas de la organización y se relaciona con todos los procesos, desde la generación de datos internos, la selección y adquisición de documentos hasta la organización de su uso. Por lo tanto la gestión de la información implica:

- Determinar la información que se precisa.
- Recoger y analizar la información.
- Registrar y recuperar la información cuando sea necesaria.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- Utilizar la información.
- Divulgar la información. (1)

## 1.4 Sistemas de gestión de la Información

Un Sistema de Gestión de Información (SGI) es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una organización. Los sistemas de gestión de información para la gestión del conocimiento constituyen hoy una alternativa de imprescindible presencia en cada organización. Permiten operar casi todos los activos tangibles e intangibles de la institución y llegan a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización. (2)

Un Sistema de Gestión de Información realiza cuatro actividades básicas:

1. **Entrada de Información:** es el proceso mediante el cual el sistema de gestión de información toma los datos que requiere para procesar la información.
2. **Almacenamiento de información:** es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.
3. **Procesamiento de Información:** es la capacidad del sistema de gestión de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.
4. **Salida de Información:** es la capacidad de un sistema de gestión de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas y disquetes.

## 1.5 Planificación de procesos.

Según Lawrence Gitman la planificación es la selección de estrategias, políticas, programas y procedimientos para lograr determinados objetivos; es también la toma de decisiones o selección de un curso de acción entre varias alternativas. (3)

Según el autor se puede ver la planeación como un conjunto de procedimientos en función de alcanzar un objetivo definido, para ello se vale de múltiples variables que definen la toma de decisiones. En otro punto de vista. La planificación requiere también de tiempo, técnica y

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

organización, por lo que las TIC's permiten optimizar estas variables a la hora de asignar recursos para la realización de este proceso, además disminuye el nivel de incertidumbre y la ocurrencia de errores introducidos por el accionar del hombre.

Se puede entonces concluir que la planeación es un proceso en sí misma, con entradas, acciones y salidas, lo que facilita su aplicación a los procesos de una entidad.

En un sistema informático se pueden manejar sus características fundamentales, el almacenamiento histórico de los datos generados, permite su fácil acceso para su posterior uso dentro del mismo flujo del proceso. La toma de decisiones se limita a una propuesta simple y sin intervención de variables que aumenten la complejidad de la misma para que mediante el accionar del hombre se pueda ajustar y realizar una nueva propuesta. De este modo el riesgo es mínimo aunque no deja de estar presente pero no deja de ser directamente proporcional a la cantidad de variables que intervengan durante la ejecución del proceso de planeación, así se tiene que mientras más factores influyan en la misma mayor será el riesgo a tener en cuenta.

## **1.6 Sistemas informáticos existentes en la gestión residencial universitaria.**

A continuación se hace un estudio sobre algunos sistemas existentes para la gestión en la residencia estudiantil en el ámbito universitario. Realizando un análisis de algunas aplicaciones homólogas que fueron creados con este fin, con el objetivo de identificar elementos comunes que puedan caracterizar este tipo de producto y para proporcionar además una nueva visión del área investigada.

### **1.6.1 Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Río “Hermanos Saiz Montes de Oca”. (SAICRE).**

Este sistema se ha diseñado y elaborado con el objetivo de gestionar la información de los becados de la Comunidad de Residencia Estudiantil en la Universidad de Pinar del Río (UPR).

El software fue desarrollado con uso del gestor de base de datos Microsoft Access y de su lenguaje residente “*Visual Basic para Aplicaciones*”. Cuenta con un sistema de ayuda creado en RoboHELP Office 9.0 y funcionalidades para garantizar la seguridad de la información.

SAICRE permite:

- Controlar el registro de becados.
- Controlar el registro de indicadores de los becados
- Evaluaciones.
- Sanciones.
- Reconocimientos.
- Actividades.
- Tratamientos.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

SAICRE facilita la gestión de la información de los becados, permitiendo controlar los datos de cada uno como: evaluaciones semestrales y una anual en el curso, cuarterías, guardias, inspecciones a los cuartos donde el orden, la limpieza e higiene, el cuidado de la propiedad social y la estética son los principales parámetros evaluados. (4)

Este sistema utiliza como gestor de Base de Datos y herramienta de modelado software propietarios. Aunque tiene funcionalidades en común con la aplicación a desarrollar, estas están encaminadas a resolver las necesidades de la Universidad de Pinar del Río por lo que no es viable su utilización en el sistema a implementar. Carece de otras funcionalidades necesarias para la gestión de la información de la residencia en la Facultad 5, como es el registro de medios tecnológicos, la planificación y evaluación de la cuartería, registro de llaves y paradas de beca que se programan en el área. Además del uso de aplicaciones propietarias, que no corresponden a las políticas de la Universidad.

## 1.6.2 Sistema de Gestión de Información de la Facultad 8 (SGIF 8).

Es un sistema para gestionar toda la información referente a la facultad 8, actualmente facultad 4 de la Universidad de las Ciencias Informáticas. Posee 6 módulos, de estos uno es para gestionar gran parte de la información no docente que se lleva a cabo en la facultad y vinculada a la residencia estudiantil como son:

- Ubicar a los estudiantes en los apartamentos de los edificios con los que dispone la facultad.
- Conocer en qué edificio, paso de escalera y apartamento de la residencia se encuentra ubicado un estudiante de la facultad o un grupo de estudiantes.
- Emitir partes de guardia.
- Conocer cómo se ha comportado la guardia estudiantil de una brigada específica hasta el momento.
- Consultar los partes de la guardia estudiantil de cualquier día.

El sistema ha sido desarrollado utilizando una arquitectura cliente-servidor, además puede ser utilizado en múltiples plataformas. El lenguaje de programación Web utilizado es PHP (Personal Home Page), se seleccionó CodeIgniter como *frameworks* para la aplicación. (5)

Este sistema carece de funcionalidades necesarias para la gestión de la información de la residencia en la Facultad 5, no gestiona las afectaciones de los apartamentos, las evaluaciones de los estudiantes en la residencia y las paradas de beca. Además no posee el soporte necesario para la generación de reportes estadísticos que visualicen indicadores claves del desempeño en esta área que faciliten la toma de decisiones a los directivos.



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## 1.6.3 DATAFEU

La herramienta DataFEU es un sistema de gestión para la automatización de los principales procesos llevados a cabo por la Federación Estudiantil Universitaria en la UCI y apoyar a sus principales dirigentes en el proceso de toma de decisiones. Fue desarrollada sobre los marcos de trabajo Symfony y ExtJS empleando MySQL (Lenguaje de Consulta Estructurado) como gestor de bases de datos, ajustándose a la soberanía tecnológica por la que aboga Cuba hoy.

Se centra principalmente en el proceso de integralidad de los estudiantes de la UCI, registrando la trayectoria de cada uno de ellos desde el 1ero al 5to año de la carrera. Además, se apoya en las evaluaciones de los estudiantes en las diferentes esferas de la vida universitaria, en la participación en eventos desarrollados a los diferentes niveles y en los méritos y distinciones obtenidos. Además contempla las sanciones y elementos negativos de los cuales pueden ser objeto.

El sistema cuenta con módulos para la seguridad y la administración; así como también un módulo para la generación de reportes en formato PDF (Formato de Documento Portátil) con el objetivo de extraer estadísticas y centralizar la información hacia los factores de la universidad. Pero no permite que los vicedecanos de extensión y residencia de cada facultad y los psicopedagogos, monitoricen el trabajo de los instructores o accedan a ver información de su interés. Además el sistema que emplean para el envío de las notificaciones no está validado para que estas lleguen en su debido momento.

Estas soluciones de software estudiadas no constituyen alternativas para la residencia estudiantil de la Facultad 5, las mismas carecen en gran medida de funcionalidades que permitan llevar un flujo de información de los principales procesos identificados en el área residencial. No se realiza un estudio sociopolítico de cada estudiante, tarea fundamental de las instructoras de los edificios. Escasean las generaciones de reportes estadísticos que visualicen los resultados de las evaluaciones recibidas por los estudiantes mensualmente. No obstante, el estudio realizado sirve como base para la definición del sistema que se pretende obtener en este trabajo.

## 1.7 Metodología de desarrollo de software

Todo proceso de desarrollo de una aplicación tiene como propósito la producción eficaz y eficiente de un producto de software que reúna los requisitos determinados por el cliente. Un proceso de software detallado y completo suele denominarse Metodología.

Las metodologías de desarrollo de software se han clasificado en dos grandes grupos: las ágiles y las tradicionales. Las metodologías ágiles enfatizan la comunicación con el cliente mientras que

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

suelen ser criticadas por la falta de documentación técnica. Las tradicionales o pesadas son aquellas con mayor énfasis en la planificación y control del proyecto. (6)

Entre las metodologías ágiles se destacan XP (*eXtreme Programming*), OpenUP (*Open Unified Process*), Scrum y Crystal. Así mismo se puede mencionar RUP (*Rational Unified Process*) y MSF (*Microsoft Solution Framework*) como las más destacadas dentro de las metodologías pesadas.

A continuación se realiza un análisis de algunas metodologías, para de ese modo definir cuál es la indicada para utilizar en el desarrollo del sistema que se quiere lograr.

## 1.7.1 Metodología XP (*Extreme Programming, Programación Extrema*)

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. A continuación se presentan las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles y proceso.

**Historias de Usuario:** Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

**Roles XP:** Aunque en varias fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado se describirán los roles de acuerdo con la propuesta original de Beck:

- Programador.
- Cliente.
- Encargado de pruebas (Tester).
- Encargado de seguimiento (Tracker).
- Entrenador (Coach).
- Consultor.
- Gestor (Big boss).

**Proceso XP:** Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste a grandes rasgos en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

El ciclo de vida ideal de XP concibe seis fases:

- Fase I Exploración.
- Fase II Planificación de la Entrega.
- Fase III Iteraciones.
- Fase IV Producción.
- Fase V Mantenimiento.
- Fase VI Muerte del Proyecto. (6)

## 1.7.2 Metodología RUP (*Rational Unified Process*)

Rational Unified Process (RUP) es un proceso de desarrollo de software, donde se forman disciplinas de asignar tareas y responsabilidades en una organización. Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML y trabajos de muchas metodologías utilizadas por los clientes.

Como proceso define como sus principales elementos:

- Trabajadores (“quién”): Define el papel de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo. Los seis primeros son flujos de ingeniería y los tres últimos de apoyo. Cada flujo de trabajo cumple con algunas actividades específicas. En él funcionan trabajadores específicos que producen y consumen artefactos también definidos. Cada fase representa un estado del proyecto y produce un hito que sirve de entrada a la próxima fase. Todos los flujos se aplican en todas las fases, si bien algunos tienen más carga de trabajo que otros en algunas fases específicas.

Uno de los objetivos de RUP es asegurar la producción de Software de calidad dentro de plazos y presupuesto predecible, dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Es un producto desarrollado y mantenido por Rational que no es más que un

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

agrupamiento de metodologías y herramientas que abarca todos los aspectos del desarrollo. Es también actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo con la experiencia. Se centra en la producción y mantenimiento de modelos del sistema más que un producto documentado. En sí es una guía de cómo usar UML de la forma más efectiva. (6)

## 1.7.3 Fundamentación del uso de la metodología XP

Después haberse concluido el estudio y análisis de estas metodologías de desarrollo, se ha llegado a la conclusión de utilizar XP, ya que la misma es adaptable al software a desarrollar, así como a las condiciones de trabajo, de forma general. Propone que la comunicación y satisfacción del cliente es lo principal. No se hace mucho énfasis en la documentación, sólo es más importante definir los requerimientos y las pruebas de calidad.

Esta metodología es flexible cuando los requerimientos sufren cambios a menudo y permite administrarlos de forma óptima. Uno de los objetivos de importancia que aplica esta metodología, es potenciar al máximo el trabajo en grupo, donde los jefes de proyecto, los clientes y desarrolladores son parte del equipo y están involucrados en el desarrollo del software.

## 1.8 Frameworks de desarrollo:

Un framework de aplicaciones web es un tipo de framework que permite el desarrollo de sitios web dinámicos, web services (servicios web) y aplicaciones web. El propósito de este tipo de framework es permitir a los desarrolladores construir aplicaciones web y centrarse en los aspectos interesantes, aliviando la típica tarea repetitiva asociada con patrones comunes de desarrollo web. La mayoría de los frameworks de aplicaciones web proporcionan los tipos de funcionalidad básica común, tales como sistemas de templates (plantillas), manejo de sesiones de usuario, interfaces comunes con el disco o el almacenamiento en base de datos de contenido cacheado, y persistencia de datos. Normalmente, los frameworks de aplicación web además promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos. (7)

### Ventajas Generales:

- El desarrollo rápido de aplicaciones. Los componentes incluidos en un framework constituyen una capa que libera al programador de la escritura de código de bajo nivel.
- La reutilización de componentes de software al por mayor. Los frameworks son los paradigmas de la reutilización.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- El uso y la programación de componentes que siguen una política de diseño uniforme. Un framework orientado a objetos logra que los componentes sean clases que pertenezcan a una gran jerarquía de clases, lo que resulta en bibliotecas más fáciles de aprender a usar. (8)

## Desventajas:

- La dependencia del código fuente de una aplicación con respecto al framework. Si se desea cambiar de framework, la mayor parte del código debe reescribirse.
- La demanda de grandes cantidades de recursos computacionales debido a que la característica de reutilización de los frameworks tiende a generalizar la funcionalidad de los componentes. El resultado es que se incluyen características que están "de más", provocando una sobrecarga de recursos que se hace más grande en cuanto más amplio es el campo de reutilización. (8)

### 1.8.1 Symfony

Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (9)

#### Características:

Symfony fue diseñado para ajustarse a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes. (9)

### 1.8.2 CodeIgniter

Es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas. (10)

## Características generales de CodeIgniter:

Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

- Capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde se tiene un acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración.
- Define una manera de trabajar específica, no es necesario seguir sus reglas de codificación, el uso de algunos módulos son opcionales. Ayuda a que la curva de aprendizaje sea sencilla al principio.
- Núcleo ligero, permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código.
- La documentación está escrita en modo de tutorial. Esto posibilita la referencia rápida, para consultar sobre una función o un método en concreto. (10)

## 1.8.3 Yii PHP Framework

Yii es un framework PHP basado en componentes de alto performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo.

Es un framework genérico de programación Web que puede ser utilizado para todo tipo de aplicaciones Web. Gracias a que es liviano de correr y está equipado con soluciones de cacheo sofisticadas, es adecuado para desarrollar aplicaciones de gran tráfico como portales, foros, sistemas de administración de contenidos (CMS) y sistemas de comercio electrónico (e-commerce). (11)

## Características de Yii incluyen:

- Patrón de diseño Modelo Vista Controlador (MVC).
- Integración con jQuery.
- Entradas de Formulario y validación.
- *Widgets* de Ajax, como autocompletado de campos de texto y demás.
- Soporte de Autenticación incorporado. Además soporta autorización vía *role-based access control* (RBAC) jerárquico.
- Personalización de aspectos y temas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- Generación compleja automática de WSDL, especificaciones y administración de peticiones *Web services*.
- Esquema de caching por capas. Soporta el cache de datos, cache de páginas, cache por fragmentos y contenido dinámico. El medio de almacenamiento del cache puede ser cambiado.
- El manejo de errores y logging. Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Las medidas de seguridad incluyen la prevención *cross-site scripting* (XSS), prevención *cross-site request forgery* (CSRF), prevención de la manipulación de cookies, etc.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD.
- La generación de código por componentes de Yii y la herramienta por línea de comandos cumplen con los estándares de XHTML. (11)

Se selecciona como framework de desarrollo Yii, a partir de las peticiones del cliente por las facilidades que brinda en la implementación de proyectos guiados por la metodología XP, donde con pocos miembros en el equipo de desarrollo es viable su aplicación. Guiado sobre el patrón arquitectónico MVC, dividiendo en capas el flujo de los datos.

## 1.9 Lenguajes de Programación para aplicaciones web

Un lenguaje de programación es un idioma diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Definido además como un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una computadora. (12)

### 1.9.1 *Hyper Text Markup Language 5* (HTML 5)

HTML, siglas de *Hyper Text Markup Language* (lenguaje de marcado de hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web, utilizado normalmente en la *www* (*World Wide Web*). Fue creado en 1986 por el físico nuclear Tim Berners-Lee. HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos *Windows* (Bloc de notas), o el que ofrece MS Office (*Word*).

El conjunto de etiquetas que se creen, se deben guardar con la extensión *.htm* o *.html*. Estos documentos pueden ser mostrados por los visores o "*browsers*" de páginas Web en Internet, como Netscape *browser*, Mozilla Firefox, Mosaic, Opera y Microsoft Internet Explorer. (13)

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

En el caso HTML5 (versión 5), es la quinta revisión importante del lenguaje básico, especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML. Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

## 1.9.2 *JavaScript.*

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Entre sus principales características se puede mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. El HTML solo permitía crear páginas estáticas donde se podía mostrar textos con estilos, pero se necesitaba interactuar con los usuarios. El código JavaScript podemos encontrarlo dentro de las etiquetas <body></body> de las páginas web. Por lo general se insertan entre:<script></script>. También pueden estar ubicados en ficheros externos usando:

```
<script type="text/javascript" src="micodigo.js"></script>
```

JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (14)

## 1.9.3 *Cascading Style Sheets (CSS).*

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Una hoja de estilos CSS consiste en una serie de reglas. Cada regla consiste en uno o más *selectores* y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor. (13)

Algunas ventajas de utilizar CSS son:



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.

## 1.9.4 *Hipertext Pre-processor (PHP)*

El Hypertext Preprocessor, es un lenguaje de programación interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes, es su soporte para gran cantidad de bases de datos, entre las que se pueden mencionar: InterBase, MSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.

Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas. Es muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.

Debido a su amplia popularidad está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. (15)

### **Selección de los lenguajes de programación a utilizar.**

Constituye una petición del cliente, el desarrollo del producto empleado el lenguaje de programación PHP del lado del servidor, además, luego de analizar las características y ventajas que posee, expuestas anteriormente, se considera que es el más apropiado para la implementación del software que se pretende realizar. Como lenguajes del lado del cliente *JavaScript*, HTML y CSS, por las ventajas que muestra en la separación del contenido de la presentación, lo que facilita al creador, diseñador o usuario el fácil manejo de las vistas, permitiendo un amplio intercambio con los clientes.

## 1.10 Sistema de Gestión de Bases de Datos (SGBD).

Los sistemas de gestión de bases de datos (*DBMS, por sus siglas en inglés*) 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. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. (16)

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## 1.10.1 PostgreSQL 8.3

Servidor de bases de datos relacional libre, ofrecido bajo la licencia BSD (*Berkeley Software Distribution*). PostgreSQL incluye características de la orientación a objetos, como puede ser: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Las principales características de PostgreSQL son: (17)

- **Atomicidad:** Asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia:** Asegura que sólo se empiece aquello que se puede acabar. Se ejecutan las operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento:** Asegura que una operación no puede afectar a otras. Ello garantiza que dos transacciones sobre la misma información nunca generará ningún tipo de error.
- **Durabilidad:** Asegura que una vez realizada la operación, esta persistirá y no se podrá deshacer.
- **Soporte de casi todos los sistemas operativos:** Linux, Unix, Mac OS, Beos, Windows, etc.
- **Documentación muy bien organizada, pública y libre.**
- **Altamente adaptable a las necesidades del cliente**
- **Soporte nativo para los lenguajes más populares del medio:** PHP, C, C++, Perl, Python.
- **Soporte de todas las características de una base de datos profesional.** (17)

## 1.10.2 MySQL.

Es un sistema de gestión de base de datos relacional y multiusuario con un alto grado de utilización y popularidad. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. La versión comercial de MySQL, no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario. Este gestor de bases de datos es muy rápido y de fácil de uso. (18)

### Principales características de MySQL

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para los campos y columnas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

- Dispone de una Interfaz de programación de aplicaciones (API, del inglés *Application Programming Interface*) en gran cantidad de lenguajes: C, C++, Java, PHP, entre otros.
- Tiene un alto grado de portabilidad entre sistemas. (18)

Se selecciona MySQL como Sistema Gestor de Base de Datos, por las características y ventajas que presenta, antes mencionadas. Cuenta con infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Por otra parte se ofrece bajo la licencia GNU GPL.

## 1.11 Servidor Web.

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS. (19)

Se selecciona Apache 2.0 ya que es un servidor Web potente, flexible y disponible para distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

### Características:

- Es una tecnología gratuita, de código abierto y multi-plataforma, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Facilita la integración como "plug-ins" de lenguajes de programación de páginas web dinámicas
- Posee una alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.
- Fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación a dado a lugar el desarrollo de aplicaciones como el APPSERV y XAMPP los cuales instalan el Apache y el PHP configurados para su uso. (19)

## 1.12 IDE's de desarrollo

Un Entorno de Desarrollo Integrado (en lo adelante IDE, por sus siglas en inglés *Integrated Development Environment*) es un entorno de programación que ha sido empaquetado como un programa de aplicación. Consiste en un editor de código, un compilador, un depurador y en ocasiones un constructor de interfaz gráfica de usuario.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

Los IDE's pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Es un editor de código que además puede servir para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación. (20)

## 1.12.1 Aptana Studio

Aptana Studio es un entorno de desarrollo integrado gratuito basado en Eclipse y desarrollado por Aptana, Inc. Su uso es básicamente para la elaboración de aplicaciones web dinámicas que empleen PHP, es multiplataforma y de código abierto, es una distribución focalizada en el desarrollo web, con soporte a HTML, CSS y JavaScript. Aptana Studio contiene también información de soporte para los principales navegadores web: Internet Explorer, Firefox, Opera, Netscape y Safari. (21)

### Otras características de Aptana Studio:

- Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe.
- Visualización de errores de sintaxis a medida que se escribe.
- Debug en Firefox (Debug Internet Explorer también con la versión Profesional).
- Librerías de funciones en Javascript populares en Ajax/Javascript para utilizar en los proyectos.
- Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- Explorador de código en forma de árbol.
- Asistente de código para HTML y Javascript.
- Extensión de funcionalidad mediante macros y acciones.
- Servidor local para probar el código.
- Herramientas para trabajo con bases de datos.

## 1.12.2 Zend Studio

*Zend Studio* o *Zend Development Environment* es un completo entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Junto con su contraparte *Zend Server* (antes llamado *Zend Platform*), son la propuesta de *Zend Technologies* para el desarrollo de aplicaciones Web utilizando PHP, actuando *Zend Studio* como la parte cliente y *Zend Server* como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre.

A partir de la versión 6, *Zend Studio* fue tomando como base el entorno de desarrollo Eclipse. (22)

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

## Características

- No requiere la instalación previa del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- Ofrece soporte mediante PHPDocs y PHPDocumentor integrado.
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones. (22)

### 1.12.3 PHP Storm

Es un IDE de desarrollo que soporta todo la base de todas las nuevas tecnologías incluyendo Vagrant, Drupal, Zend 2, Web Components, TypeScript además de poder depurar fácilmente. Tiene soporte para la última versión de PHP (5.5) y mejoras en PHPDoc, así como una nueva forma de inspeccionar código. También provee colores en la sintaxis de PHP, la cual incluso puede ser definida por el usuario, con lo que se mejora la lectura del código. Hay cuatro colores diferentes: para declaraciones de funciones y métodos, para clases e interfaces, campos e instancias de clases de campos. (23)

#### Características relevantes en PHP Storm:

- Ayuda para la depuración y prueba de unidades. La validación de la configuración del depurador verifica la configuración de Xdebug y de Zend Debugger, y permite saber si se olvida algo, así como la forma de corregirlo.
- Tiene un Step-Into, lo cual es una característica que permite elegir la función que queremos analizar paso a paso mientras se depura y esta funcionalidad también trabaja en las llamadas a funciones dinámicas.
- Tiene un plug-in para Drupal 6 y 7, y en ocasiones, funciona bien para Drupal 8. Hay soporte para el Zend Framework 2, Symfony 2 y otros entornos de PHP.
- Introduce también un nuevo *web toolkit* con soporte a tecnologías web, incluyendo tres alternativas de templates en JavaScript: EJS, Handlebars y Mustache. (23)

## 1.13 Consideraciones parciales.

Durante el desarrollo del capítulo fueron analizados los antecedentes en el campo de acción, realizando un estudio de la gestión de la información y todo lo que conlleva a informatizar y gestionar los datos. Junto a ellos fueron analizados un grupo de aplicaciones existentes que apoyan la gestión universitaria. Luego de haber realizado un profundo estudio de algunas tecnologías utilizadas para el desarrollo de aplicaciones web se seleccionaron las herramientas para darle solución a la situación existente, se utilizará PHP como lenguaje de programación, teniendo en cuenta algunas de sus ventajas como ser un lenguaje multiplataforma, con capacidad de conectarse con la mayoría de los manejadores de base de datos que se utilizan en la actualidad; y lenguaje del lado del cliente JavaScript, HTML y CSS. Como sistema gestor de base de datos MySQL por su rapidez, confiabilidad y facilidad de uso. La metodología de desarrollo: XP debido a sus características y organización del trabajo. Servidor Web: Apache, PHP Storm como IDE de desarrollo y como framework el Yii Framework.

# CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

En el presente capítulo se brinda una descripción de las principales características del sistema propuesto, además del análisis de la información vinculada con el objeto de estudio y las características asociadas al dominio de la aplicación. Se identifican los actores a partir de los roles de usuarios definidos. Se detallan los requisitos funcionales denominados historias de usuario por la metodología de desarrollo seleccionada, así como los requisitos no funcionales y patrones de diseño.

### 2.2 Descripción de la solución propuesta.

Una vez realizado el análisis del estado del arte, elegidas las herramientas y la metodología de desarrollo a utilizar, se propone formalizar el diseño e implementación de una aplicación informática para la planificación y control de la residencia estudiantil en la Facultad 5. Dicho sistema le permite tanto a las instructoras como al Vicedecanato de Residencia y Extensión Universitaria poder llevar un control estricto de cada una de las actividades de los estudiantes en la beca. La misma consiste en una aplicación Web, diseñada principalmente para apoyar el proceso del flujo de información y consulta de diferentes actividades, como son; distribución por edificios, limpieza diaria y general de cada uno de ellos, planificación de la cuartería, reportes de mantenimiento y tecnológicos, generación de reportes en formato pdf con todas las evaluaciones obtenidas y demás aspectos significativos, entre otras acciones reflejadas en los requerimientos funcionales. Se logra entonces así que varias personas puedan acceder a los datos registrados de forma simultánea, logrando mayores resultados en un proceso de búsqueda. Permitiendo la autenticación por el servicio LDAP brindado en la Universidad, logrando así la confiabilidad de la información y la seguridad del sistema propuesto.

### 2.3 Especificación de los requerimientos

Los requisitos funcionales son las acciones que los usuarios realizarán y el sistema debe ser capaz de responder a ellas. La IEEE-STD-729 (*Institute of Electrical and Electronics Engineers*) los define como la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo (24). Otras de las definiciones consultadas plantean que un requisito funcional es la condición o capacidad que debe ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.3.1 Requerimientos No Funcionales

Los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener, imponen restricciones en el diseño o la implementación. Deben pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Por tanto son los criterios que se deben juzgar para determinar el funcionamiento de un sistema, verificando cómo este debería ser. (25)

*Tabla 1: Requerimientos No Funcionales del sistema*

No. RnF	Nombre	Descripción
RnF1	Usabilidad	El sistema debe poseer un procedimiento sencillo de puesta en marcha y uso, garantizando a los usuarios un fácil e intuitivo acceso al mismo.
RnF2	Apariencia o interfaz externa e interna	<b>Interfaz Web:</b> la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo. <b>Interfaz interna:</b> la interfaz interna estará determinada por el desarrollador, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.
RnF3	Seguridad	<b>Confidencialidad:</b> Existencia de distintos roles que establezcan que la información sólo sea vista por aquellos usuarios que posean los privilegios suficientes; restringir la ejecución de acciones a usuarios sin credenciales que intenten acceder a las mismas y la verificación de que el usuario esté autenticado antes de realizar alguna acción. <b>Integridad:</b> Validación de los datos en el servidor para evitar estados inconsistentes. La información manejada por el sistema estará protegida del acceso y divulgación no autorizada. Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones. <b>Disponibilidad:</b> El sistema estará disponible las 24 horas del día a los usuarios autorizados, garantizando el acceso a la información en cualquier momento. Los mecanismos utilizados para lograr la seguridad no obstruyen el acceso a la



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

		información. Se realizarán salvas periódicas en otros dispositivos para evitar pérdidas de información en caso de que haya fallos en el sistema.
<b>RnF4</b>	Rendimiento	Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.
<b>RnF5</b>	Software	<b>Cliente:</b> Utilizar cualquier navegador (Mozilla Firefox, Internet Explorer, entre otros). Sistema operativo Windows XP o superior, o Linux. <b>Servidor:</b> Sistema operativo Windows o Linux en cualquiera de sus distribuciones. Un servidor Apache 2.0 con módulo PHP disponible.
<b>RnF6</b>	Hardware	Capacidad de Disco Duro igual o superior a 5 Giga bytes. Se requiere un mínimo de 256 MB de RAM y 1.6 GHz de velocidad de procesamiento.

### 2.4 Roles de usuarios

Los actores del sistema suelen ser los trabajadores del negocio. Son las personas o sistemas que van a interactuar directamente con la aplicación con el objetivo de beneficiarse de sus funcionalidades. Una tarea fundamental radica en delimitar la audiencia a la cual va dirigido el mismo.

Dentro de la audiencia identificada se incluyen, como personal relacionado; al administrador, las instructoras y los estudiantes. Los usuarios del sistema serán aquellos que obtengan un resultado de la ejecución de uno o varios procesos. A continuación se muestra un listado con los roles de usuarios definidos para la aplicación:

**Tabla 2: Roles en el sistema**

Roles en el sistema	Descripción
<b>Administrador</b>	Encargado de gestionar toda la información tanto para los usuarios autenticados como para los invitados. Establece los permisos pertinentes para los distintos tipos de beneficiarios.
<b>Instructora</b>	Encargada de gestionar la información de los estudiantes, así como las evaluaciones e indicaciones pertinentes que sean

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	necesarias orientarlas mediante la aplicación.
<b>Estudiante</b>	Tiene acceso a todo lo publicado en el sitio de forma pública definido por el administrador del sistema. Tendrá privilegios extras en dependencia de los permisos que le sean asignados por el administrador.

### 2.5 Fase de Exploración y Planificación

En esta fase, los clientes plantean a grandes rasgos las historias de usuarios que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, exploran las posibilidades de arquitectura y experimentan los límites de las tecnologías a utilizar. Por otra parte la Planificación es una fase, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, asociadas a éstas; las entregas.

#### 2.5.1 Historias de usuarios

Uno de los artefactos más importantes que genera la metodología XP son las Historias de Usuario. Éstas tienen el mismo propósito que los casos de uso y son confeccionadas por el cliente. Las mismas expresan su punto de vista en cuanto a las necesidades del sistema. Son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica.

Otra de sus características es que solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. Su nivel de detalle debe ser el mínimo posible, de manera que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Las historias de usuarios quedan estructuradas de la siguiente manera:

- **Nombre:** nombre descriptivo de la HU.
- **Prioridad en el negocio:** grado de prioridad que le asigna el cliente a la HU en dependencia de su importancia. Los valores que puede tomar son: alta, media o baja.
- **Riesgo de desarrollo:** grado de complejidad que le asigna el desarrollador a la HU luego de analizarla. Los valores que puede tomar son: alta, media o baja.
- **Estimación:** unidades de tiempo (días) estimadas por el desarrollador para darle cumplimiento a la HU.
- **Iteración:** número de la iteración en la cual será implementada la HU.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- **Descripción:** descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.
- **Observaciones:** aspectos a tener en cuenta según la opinión del equipo de desarrollo para cumplir con la descripción.

A continuación una descripción de algunas historias de usuario definidas, las restantes forman parte de los Anexos.

**Tabla 3: Historia de Usuario - Autenticar usuario**

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Nombre:</b> Autenticar usuario
<b>Usuario:</b> Todos	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Los usuarios podrán autenticarse para acceder a algunas opciones del sistema: <b>Datos para autenticarse:</b> <ul style="list-style-type: none"><li>• Nombre del Usuario (Obligatorio).</li><li>• Contraseña (Obligatorio).</li></ul>	
<b>Observaciones:</b> La autenticación se realizará con el usuario del dominio UCI.	

**Tabla 4: Historia de Usuario - Modificar usuario**

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Nombre:</b> Modificar usuario
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá modificar los datos de un usuario determinado: <b>Datos de usuarios:</b> <ul style="list-style-type: none"><li>• Nombre.</li><li>• Apellidos.</li><li>• Correo electrónico.</li></ul>	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

• Rol.
<b>Observaciones:</b> Una vez que el usuario es registrado en el sistema cuando ingresa por primera vez, el administrador puede modificar sus datos.

**Tabla 5: Historia de Usuario - Eliminar usuario**

<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Nombre:</b> Eliminar usuario
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá eliminar los datos de un usuario determinado.	
<b>Observaciones:</b> Una vez que el usuario es registrado en el sistema cuando ingresa por primera vez, el administrador puede eliminar sus datos.	

### 2.5.2 Planificación de la entrega

En esta fase el cliente establece la prioridad de cada historia de usuario y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas, de manera general consiste en una o varias reuniones grupales de planificación. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

### 2.5.3 Estimación de esfuerzo por historia de usuario

**Tabla 6: Estimación de esfuerzo por historia de usuario**

Nro.	Historia de usuario	Puntos de estimación(días)
1	Autenticar usuario	1
2	Modificar usuario	0.5
3	Eliminar usuario	0.5
4	Adicionar instructora	1
5	Modificar instructora	0.5
6	Eliminar instructora	0.5

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

7	Adicionar edificio	1
8	Modificar edificio	0.5
9	Eliminar edificio	0.5
10	Adicionar apartamento	1
11	Modificar apartamento	0.5
12	Eliminar apartamento	0.5
13	Adicionar estudiante	1
14	Modificar estudiante	0.5
15	Eliminar estudiante	0.5
16	Planificar limpieza	3
17	Modificar limpieza	0.5
18	Eliminar limpieza	0.5
19	Planificar cuarterería	3
20	Modificar cuarterería	0.5
21	Eliminar cuarterería	0.5
22	Registrar evaluación	3
23	Modificar evaluación	0.5
24	Eliminar evaluación	0.5
25	Registrar llave	1
26	Modificar llave	0.5
27	Eliminar llave	0.5
28	Registrar medio	1
29	Modificar medio	0.5
30	Eliminar medio	0.5
31	Generar actas de responsabilidad material	1
32	Planificar parada de beca	1
33	Modificar parada de beca	0.5
34	Eliminar parada de beca	0.5
35	Generar reportes	5
36	Graficas informativas	5

# CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

## 2.5.4 Plan de Iteraciones

Tabla 7: Plan de Iteraciones

Iteración	Descripción de la iteración	Orden de las HU a implementar	Duración
1	En la primera iteración se implementan las historias de usuario iniciales, obteniendo al final de la misma una primera versión de prueba y dando al sistema las funcionalidades básicas.	<ol style="list-style-type: none"><li>1. Autenticar usuario</li><li>2. Modificar usuario</li><li>3. Eliminar usuario</li><li>4. Adicionar instructora</li><li>5. Modificar instructora</li><li>6. Eliminar instructora</li><li>7. Adicionar edificio</li><li>8. Modificar edificio</li><li>9. Eliminar edificio</li><li>10. Adicionar apartamento</li><li>11. Modificar apartamento</li><li>12. Eliminar apartamento</li></ol>	2 semanas
2	En la segunda iteración se realiza la implementación de las historias de usuario con prioridad de negocio alta. Además, se corregirán errores o disconformidades del usuario con las historias de usuario implementadas en la iteración anterior. De esta forma se obtiene la segunda versión de pruebas del software. Esta segunda versión será mostrada a los clientes con el único objetivo de realizar cambios en base a la aceptación del mismo.	<ol style="list-style-type: none"><li>13. Adicionar estudiante</li><li>14. Modificar estudiante</li><li>15. Eliminar estudiante</li><li>16. Adicionar limpieza</li><li>17. Modificar limpieza</li><li>18. Eliminar limpieza</li><li>19. Adicionar cuartería</li><li>20. Modificar cuartería</li><li>21. Eliminar cuartería</li><li>22. Adicionar evaluación</li><li>23. Modificar evaluación</li><li>24. Eliminar evaluación</li></ol>	3 semanas
3	En la tercera iteración, ya implementadas las funcionalidades básicas, se realiza el desarrollo de las historias de usuario con prioridad media. Se corrigen errores de iteraciones anteriores, y se crean las bases para la implementación de las últimas historias de usuario	<ol style="list-style-type: none"><li>25. Registrar llave</li><li>26. Modificar llave</li><li>27. Eliminar llave</li><li>28. Registrar medio</li><li>29. Modificar medio</li><li>30. Eliminar medio</li><li>31. Generar actas de responsabilidad material</li><li>32. Planificar parada de beca</li></ol>	3 semanas

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	que a su vez tienen una complejidad superior.	33. Modificar parada de beca 34. Eliminar parada de beca 35. Generar reportes 36. Graficas informativas	
--	---	--	--

### 2.5.5 Plan de Entregas

El plan de entregas es el compromiso final del equipo de desarrollo con los clientes. Es una cuestión de vital importancia para el negocio entre ambas partes. La estimación es uno de los temas más complicados del desarrollo de un proyecto de software, es por ello que resulta de vital importancia tener bien claros los requerimientos del cliente, el estilo de trabajo del equipo de desarrollo y el tiempo con que dispone el usuario final para tener la solución completa.

*Tabla 8: Plan de entregas a los clientes*

Entregable	Final 1ra Iteración 1ra Semana de Marzo	Final 2da Iteración 1ra Semana de Abril	Final 3ra Iteración 1ra Semana de Junio
Aplicación	Versión 1.0 de la aplicación.	Versión 2.0 de la aplicación.	Entrega Final

### 2.5.6 Arquitectura del sistema

La arquitectura de software es el conjunto de decisiones significativas sobre la organización de un sistema, la selección de los elementos estructurales y las interfaces que componen la aplicación, junto con su funcionalidad y navegabilidad. Describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información (28). El sistema que se propone presenta una arquitectura basada en el patrón arquitectónico Modelo Vista Controlador (MVC).

Por otra parte tenemos que, los patrones expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos; así como ayudan a especificar la estructura fundamental de una aplicación. (24)

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### Patrón Arquitectónico (Modelo Vista Controlador)

El patrón Modelo Vista Controlador (MVC), se utiliza cuando es necesario modularizar la interfaz de usuario, las reglas de negocios y el control de eventos. El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). (29)

Es de gran importancia en el desarrollo de la aplicación ya que describe una forma de organizar el código en la misma separando los datos, la interfaz de usuario y la lógica en tres componentes distintos. Este patrón debe su nombre a los tres elementos que lo conforman, el Modelo, la Vista y el Controlador. El Modelo administra la información de la aplicación, la Vista contiene todos los elementos que conforman la interfaz gráfica y el Controlador se encarga de separar y modificar el modelo y la vista, manejando la comunicación y la interacción entre ellos. En la siguiente figura se muestra a grandes rasgos el funcionamiento del patrón:



Ilustración 1: Patrón Modelo Vista Controlador

### Patrones de diseño

Los Patrones son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas. (30)



# CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

Se hará uso de varios patrones de diseño, a continuación se mencionan los más importantes y se hace una breve descripción:

Los patrones GRASP; por sus siglas en inglés (*General Responsibility Assignment Software Patterns*), tienen como objetivo la descripción de los principios fundamentales de diseño de objetos para la asignación de responsabilidades, y dentro de estos, los patrones Experto, Creador, Alta cohesión y Bajo acoplamiento. En el diseño de la aplicación se utilizó principalmente el patrón Experto, el cual plantea que siempre se debe asignar una responsabilidad al experto en información, o sea, la clase con toda la información necesaria para llevarla a cabo, dichas clases son: Usuario, Edificio, Evaluación. El diseño obtenido cumple con los patrones de Bajo acoplamiento y Alta cohesión permitiendo la colaboración entre los elementos del diseño, sin verse afectados la reutilización de estos y el entendimiento cuando se encuentran aislados. La creación de clases controladoras facilitó realizar las operaciones del sistema, ya que no es factible manejarse en la capa de interfaz o presentación, debido a que estas operaciones reflejan los procesos de la entidad o dominio.

## 2.6 Diseño del sistema

La metodología XP no requiere la descripción del sistema por medio de diagramas de clase utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

### 2.6.1 Tarjetas CRC

Las características más sobresalientes de las tarjetas CRC son su simpleza y adaptabilidad. Una tarjeta CRC no es más que una ficha de papel o cartón que representa a una entidad del sistema, las cuales permiten también que el equipo completo contribuya en la tarea del diseño. (26)

Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente, tal y como muestra la figura.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

**Tabla 9: Representación genérica de una tarjeta CRC**

<b>Nombre de la clase</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>

**Clase:** Es cualquier persona, cosa, evento, concepto, pantalla o reporte.

**Responsabilidades:** Las responsabilidades de una clase son las entidades que conoce y las que realizan, sus atributos y métodos.

**Colaboradores:** Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades. A continuación se muestran algunas tarjetas CRC consideradas las más importantes.

**Tabla 10: Tarjeta CRC – Usset**

<b>Tarjeta CRC</b>	
<b>Nombre de la clase: Usset</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"><li>• AutenticarUsset()</li><li>• AdicionarUsset()</li><li>• ModificarUsset()</li><li>• EliminarUsset()</li><li>• ListarUsset()</li><li>• ReportePDFUsset()</li></ul>	<ul style="list-style-type: none"><li>• Usset</li></ul>

**Tabla 11: Tarjeta CRC – Instructora**

<b>Tarjeta CRC</b>	
<b>Nombre de la clase: Instructora</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"><li>• AdicionarInstructora()</li><li>• ModificarInstructora()</li><li>• EliminarInstructora()</li><li>• ListarInstructora()</li><li>• ReportePDFInstructora()</li></ul>	<ul style="list-style-type: none"><li>• Usset</li></ul>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tabla 12: Tarjeta CRC – Edificio

Tarjeta CRC	
<b>Nombre de la clase: Edificio</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"><li>• AdicionarEdificio()</li><li>• ModificarEdificio()</li><li>• EliminarEdificio()</li><li>• ListarEdificio()</li><li>• ReportePDFEdificio()</li><li>• GraficoEdificio()</li></ul>	<ul style="list-style-type: none"><li>• Instructora</li><li>• Estudiante</li><li>• Apto</li></ul>

Tabla 13: Tarjeta CRC – Apto

Tarjeta CRC	
<b>Nombre de la clase: Apto</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"><li>• AdicionarApto()</li><li>• ModificarApto()</li><li>• EliminarApto()</li><li>• ListarApto()</li><li>• ReportePDFApto()</li><li>• GraficoApto()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li></ul>

Tabla 14: Tarjeta CRC – Estudiante

Tarjeta CRC	
<b>Nombre de la clase: Estudiante</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"><li>• AdicionarEstudiante()</li><li>• ModificarEstudiante()</li><li>• EliminarEstudiante()</li><li>• ListarEstudiante()</li><li>• ReportePDFEstudiante()</li></ul>	<ul style="list-style-type: none"><li>• Usser</li><li>• Edificio</li><li>• Apto</li></ul>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tabla 15: Tarjeta CRC – Limpieza

Tarjeta CRC	
Nombre de la clase: Limpieza	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• PlanificarLimpieza()</li><li>• ModificarLimpieza()</li><li>• EliminarLimpieza()</li><li>• ListarLimpieza()</li><li>• ReportePDFLimpieza()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li><li>• Estudiante</li><li>• Apto</li></ul>

Tabla 16: Tarjeta CRC – Cuarterería

Tarjeta CRC	
Nombre de la clase: Cuarterería	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• PlanificarCuartereria()</li><li>• AdicionarCuartereria()</li><li>• ModificarCuartereria()</li><li>• EliminarCuartereria()</li><li>• ListarCuartereria()</li><li>• ReportePDFCuartereria()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li><li>• Estudiante</li><li>• Apto</li></ul>

Tabla 17: Tarjeta CRC – Evaluación

Tarjeta CRC	
Nombre de la clase: Evaluación	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• RegistrarEvaluacion()</li><li>• ModificarEvaluacion()</li><li>• EliminarEvaluacion()</li><li>• ListarEvaluacion()</li><li>• ReportePDFEvaluacion()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li><li>• Estudiante</li><li>• Apto</li></ul>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tabla 18: Tarjeta CRC – Llave

Tarjeta CRC	
Nombre de la clase: Llave	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• RegistrarLlave()</li><li>• ModificaLlave()</li><li>• EliminarLlave()</li><li>• ListarLlave()</li><li>• ReportePDFLlave()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li><li>• Apto</li><li>• Usser</li></ul>

Tabla 19: Tarjeta CRC – Medio

Tarjeta CRC	
Nombre de la clase: Medio	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• RegistrarMedio()</li><li>• ModificarMedio()</li><li>• EliminarMedio()</li><li>• ListarMedio()</li><li>• ReportePDFMedio()</li></ul>	<ul style="list-style-type: none"><li>• Edificio</li><li>• Apto</li><li>• Usser</li></ul>

Tabla 20: Tarjeta CRC – Parada

Tarjeta CRC	
Nombre de la clase: Parada	
Responsabilidades (Métodos)	Colaboradores
<ul style="list-style-type: none"><li>• RegistrarParada()</li><li>• ModificarParada()</li><li>• EliminarParada()</li><li>• ListarParada()</li><li>• ReportePDFParada()</li></ul>	<ul style="list-style-type: none"><li>• Parada</li></ul>

# CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

**Tabla 21: Tarjeta CRC – Tarea**

Tarjeta CRC	
<b>Nombre de la clase: Tarea</b>	
<b>Responsabilidades (Métodos)</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• RegistrarTarea()</li> <li>• ModificarTarea()</li> <li>• EliminarTarea()</li> <li>• ListarTarea()</li> <li>• ReportePDFTarea()</li> </ul>	<ul style="list-style-type: none"> <li>• Tarea</li> </ul>

## 2.7 Fase de Implementación

La fase de implementación en la metodología XP, plantea para su desarrollo, llevar a cabo la implementación de cada una de las historias de usuarios. Realizando un chequeo de cada una de ellas junto con el plan de iteraciones. Como parte de este plan se crean tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU. Estas pueden escribirse en un lenguaje técnico, al contrario de las historias de usuario que son escritas en el lenguaje del cliente.

Siguiendo la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema. A continuación se detallan cada una de las iteraciones realizadas y las tareas definidas en las mismas:

### Iteración 1

Esta iteración tendrá como objetivo darle cumplimiento a las primeras historias de usuarios que serán de vital importancia para el sistema, conformándose la base de la estructura del negocio, las cuales recogen las principales funcionalidades del sistema.

Módulo / componente	Historia de usuario	Tiempo de implementación(días)	
		Estimación	Real
Usuario	Autenticar usuario	1	2
	Modificar usuario	0.5	1
	Eliminar usuario	0.5	0.5
	Adicionar instructora	1	1
	Modificar instructora	0.5	0.5

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Instructora	Eliminar instructora	0.5	0.5
Edificio	Adicionar edificio	1	1
	Modificar edificio	0.5	0.5
	Eliminar edificio	0.5	0.5
Apartamento	Adicionar apartamento	1	1
	Modificar apartamento	0.5	0.5
	Eliminar apartamento	0.5	0.5

**Tabla 22: Tarea - Configuración del módulo de autenticación**

Tarea	
<b>No. Tarea: 1</b>	<b>No. HU: 1,2,3</b>
<b>Nombre:</b> Configuración del módulo de autenticación.	
<b>Tipo de tarea:</b> configuración-desarrollo.	<b>Puntos de estimación: 2</b>
<b>Fecha de inicio:</b> 10/2/14	<b>Fecha de fin:</b> 14/2/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> se configura el servicio LDAP para la verificación de los datos que son introducidos por el usuario respecto al dominio uci.cu y se implementan las funcionalidades de modificar y eliminar usuario.	

**Tabla 23: Tarea - Gestión de instructoras**

Tarea	
<b>No. Tarea: 2</b>	<b>No. HU: 4,5,6</b>
<b>Nombre:</b> Gestión de instructoras	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación: 2</b>
<b>Fecha de inicio:</b> 14/2/14	<b>Fecha de fin:</b> 16/2/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar las instructoras que forman parte del sistema.	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

**Tabla 24: Tarea - Gestión de edificios**

Tarea	
<b>No. Tarea: 3</b>	<b>No. HU: 7,8,9</b>
<b>Nombre:</b> Gestión de edificios	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 17/2/14	<b>Fecha de fin:</b> 19/2/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar edificios en el sistema.	

**Tabla 25: Tarea - Gestión de apartamentos**

Tarea	
<b>No. Tarea: 4</b>	<b>No. HU: 10,11,12</b>
<b>Nombre:</b> Gestión de apartamentos	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 20/2/14	<b>Fecha de fin:</b> 23/2/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar apartamentos en el sistema.	

### Iteración 2

La segunda iteración está centrada en continuar el desarrollo de los requerimientos con prioridad alta en el negocio. Durante el transcurso de esta iteración se implementaran funcionalidades claves para la planificación y control de las tareas realizadas por los usuarios en la esfera residencial.

Módulo / componente	Historia de usuario	Tiempo de implementación(días)	
		Estimación	Real
Estudiante	Adicionar estudiante	1	3
	Modificar estudiante	0.5	1



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	Eliminar estudiante	0.5	0.5
Limpieza	Planificar limpieza	3	3
	Modificar limpieza	0.5	0.5
	Eliminar limpieza	0.5	0.5
Cuartería	Planificar cuartería	3	2
	Modificar cuartería	0.5	0.5
	Eliminar cuartería	0.5	0.5
Evaluación	Registrar evaluación	3	4
	Modificar evaluación	0.5	0.5
	Eliminar evaluación	0.5	0.5

**Tabla 26: Tarea - Gestión de estudiantes y limpieza**

Tarea	
<b>No. Tarea: 1</b>	<b>No. HU: 13,14,15</b>
<b>Nombre:</b> Gestión de estudiantes	
<b>Tipo de tarea:</b> configuración-desarrollo.	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 25/2/14	<b>Fecha de fin:</b> 3/3/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar los estudiantes que forman parte del sistema.	
Tarea	
<b>No. Tarea: 2</b>	<b>No. HU: 16,17,18</b>
<b>Nombre:</b> Gestión de limpieza	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 4/3/14	<b>Fecha de fin:</b> 9/3/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para planificar/modificar/eliminar	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

limpiezas en el sistema.

**Tabla 27: Tarea - Gestión de cuarterías**

Tarea	
<b>No. Tarea: 3</b>	<b>No. HU: 19,20,21</b>
<b>Nombre:</b> Gestión de cuarterías	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación: 2</b>
<b>Fecha de inicio:</b> 10/3/14	<b>Fecha de fin:</b> 13/3/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para planificar/modificar/eliminar cuarterías en el sistema.	

**Tabla 28: Tarea - Gestión de evaluación**

Tarea	
<b>No. Tarea: 4</b>	<b>No. HU: 22,23,24</b>
<b>Nombre:</b> Gestión de evaluación	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación: 2</b>
<b>Fecha de inicio:</b> 14/3/14	<b>Fecha de fin:</b> 19/3/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para registrar/modificar/eliminar evaluaciones en el sistema.	

### Iteración 3

Esta iteración permite añadir funcionalidades con prioridad media en el negocio así como a cubrir diferentes necesidades específicas abordadas por el cliente, como son priorizar la generación de reportes y grafico en el sistema.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Módulo / componente	Historia de usuario	Tiempo de implementación(días)	
		Estimación	Real
Llave	Adicionar llave	1	1
	Modificar llave	0.5	0.5
	Eliminar llave	0.5	0.5
Medio	Adicionar medio	1	1
	Modificar medio	0.5	0.5
	Eliminar medio	0.5	0.5
	Generar acta de responsabilidad material	1	1
Parada de beca	Planificar parada de beca	1	1
	Modificar parada de beca	0.5	0.5
	Eliminar parada de beca	0.5	0.5
Reporte	Generar reporte	5	5
Grafica	Registrar evaluación	5	5

**Tabla 29: Tarea - Gestión de llaves**

Tarea	
<b>No. Tarea: 1</b>	<b>No. HU: 25,26,27</b>
<b>Nombre:</b> Gestión de llaves	
<b>Tipo de tarea:</b> configuración-desarrollo.	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 28/3/14	<b>Fecha de fin:</b> 30/3/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar los estudiantes que portan llaves de los apartamentos en el sistema.	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

*Tabla 30: Tarea - Gestión de medios*

Tarea	
<b>No. Tarea: 2</b>	<b>No. HU: 28,29,30,31</b>
<b>Nombre:</b> Gestión de medios	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 1/4/14	<b>Fecha de fin:</b> 5/4/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para insertar/modificar/eliminar medios en el sistema. Además la generación de las actas de responsabilidad material.	

*Tabla 31: Tarea - Gestión de paradas de beca*

Tarea	
<b>No. Tarea: 3</b>	<b>No. HU: 32,33,34</b>
<b>Nombre:</b> Gestión de paradas de beca	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 6/4/14	<b>Fecha de fin:</b> 8/4/14
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> Se implementan las funcionalidades necesarias para planificar/modificar/eliminar paradas de beca en el sistema.	

*Tabla 32: Tarea - Generar reportes*

Tarea	
<b>No. Tarea: 4</b>	<b>No. HU: 35</b>
<b>Nombre:</b> Generar reportes	
<b>Tipo de tarea:</b> configuración-desarrollo	<b>Puntos de estimación:</b> 2
<b>Fecha de inicio:</b> 10/4/14	<b>Fecha de fin:</b> 29/4/14

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

<b>Programador responsable: Erlán Rodríguez Rodríguez</b>
---

<b>Descripción:</b> Se implementan las funcionalidades necesarias para permitir la generación de los reportes especificados por el cliente.
---

### **Estándares de codificación:**

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código (27). La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios. Para la implementación del sistema desarrollado se siguieron normas y estándares desarrollados, que se relacionan a continuación:

- **Indentación**

La Indentación debe ser a cuatro espacios sin caracteres de tabulación. Esto es debido a que ciertos IDE's de desarrollo introducen caracteres de tabulación cuando indentan un texto automáticamente.

- **Estructuras de control**

Las estructuras de control deben tener un espacio entre las palabras clave de la estructura, el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones y el signo de llaves deben estar sobre la línea de la estructura.

- **Llamadas de funciones**

Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro, espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

- **Estilos de comentarios**

El estilo de los comentarios debe ser como el estilo de comentarios para C (`/* */` o `//`), no debe de utilizarse el estilo de comentarios de Perl (`#`).

# CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

- **Inclusión de archivos**

Cuando se incluya un archivo de dependencia incondicionalmente se debe utilizar `require_once` y cuando sea condicionalmente, utilice `include_once`.

- **Bloques de código**

Siempre se debe utilizar las etiquetas `<?php?>` para abrir un bloque de código. No utilizar el método de etiquetas cortas, porque esto depende de las directivas de configuración en el archivo `PHP.INI` y hace que el script no sea tan portable.

- **Nombres**

Los nombres de las clases deben de iniciar con letra mayúscula. Los nombres de las variables y de las funciones pueden iniciar con letra minúscula, pero si estas tienen más de una palabra, cada nueva palabra debe iniciar con letra mayúscula (el nombre puede escribirse separado por signos de guión mayor).

Si una función, en una clase, es privada; deberá comenzar con el signo de guión mayor para una fácil identificación. Las constantes deben de escribirse siempre en mayúsculas y tanto estas como las variables globales deben de tener como prefijo el nombre de la clase a la que pertenecen.

- **Constantes**

Los nombres de las constantes pueden contener caracteres alfanuméricos, guiones bajos (`_`) y números. Todas las letras deben estar en mayúscula. Las constantes deben ser definidas como miembros de una clase usando el modificador `"const"`. No se recomienda definir constantes con alcance global (utilizando la función `"define"`).

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente.

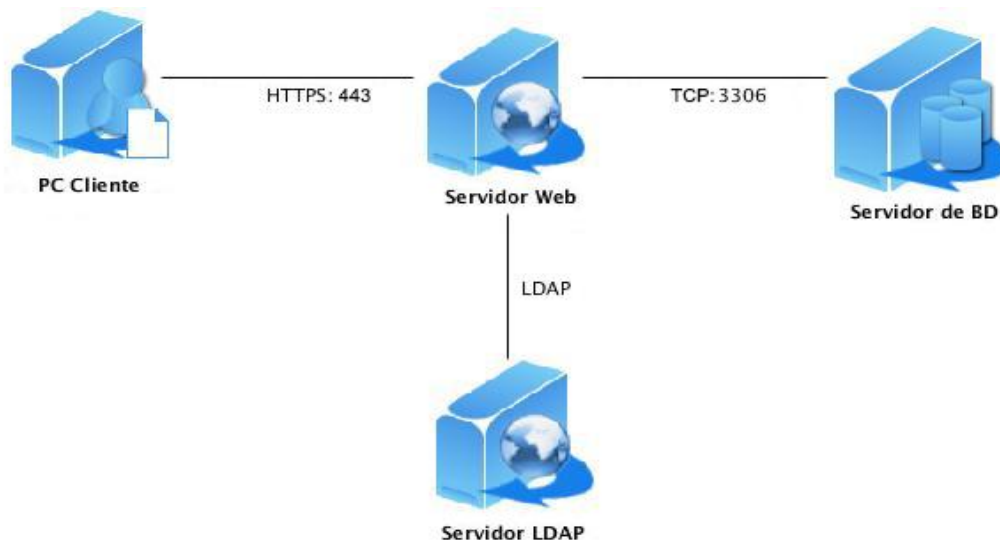
## **2.8 Diagrama de despliegue.**

El modelo de despliegue representa un modelo de objetos que muestra cómo se distribuye físicamente el sistema, teniendo en cuenta la funcionalidad entre cada nodo de cómputo. Este modelo es utilizado como base para la realización de las actividades de diseño e implementación. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El modelo de despliegue puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación, además de representar una correspondencia entre la arquitectura de software y la arquitectura del sistema (hardware).

La figura muestra el diagrama de despliegue del sistema que se propone.



**Ilustración 2: Diagrama de despliegue**

El diagrama establece la relación existente entre los nodos:

**Servidor LDAP:** Servicios web que la UCI brinda para obtener información del personal de la universidad ya sea estudiantes, profesores o trabajadores que son autenticados en el dominio uci.cu mediante LDAP.

**PC Cliente:** Espacio de trabajo por el cual el cliente interactúa con la aplicación. Esta computadora debe tener instalado el Sistema Operativo Windows o GNU/ Linux, como navegador Web Internet Explorer o Mozilla Firefox 2.0 como mínimo.

**Servidor Web:** Se utilizará el servidor Web Apache, más el lenguaje de programación PHP. Contiene toda la información referente a la aplicación.

**Servidor Base de datos:** Se utilizará MySQL como base de datos que permitirá almacenar toda la información de la aplicación.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

### **2.9 Consideraciones parciales.**

Durante el desarrollo del capítulo fueron analizadas las funcionalidades a incluir en el software a través de las historias de usuario, siguiendo el ciclo de desarrollo que propone la metodología seleccionada; XP. Se realizó la descripción de las mismas precisando por el cliente la prioridad de cada una, definiendo así el orden en el que serán implementadas. Se cuenta con 15 HU que serán implementadas en 4 iteraciones. Quedó elaborado el modelo necesario para llevar a cabo la implementación del sistema, mediante la descripción de los estándares de codificación y la arquitectura utilizada. Se definió por medio del diagrama de despliegue la distribución física, mediante la cual funcionará la aplicación.



## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

### 3.1 Introducción

En el presente capítulo se especifican las técnicas de validación empleadas para la propuesta de solución, como parte de los efectos de la ejecución de la fase de prueba, la cual refleja los resultados obtenidos de las diferentes pruebas de aceptación diseñadas para comprobar los requisitos funcionales descritos anteriormente. Se analizan los valores arrojado a partir de la aplicación de un cuestionario de preguntas a varios usuarios y/o clientes del software.

### 3.2 Pruebas de software

Una de las ventajas de la metodología XP es el uso de las pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Ello permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

La metodología ágil XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son desarrolladas por los programadores y se encargan de verificar el código automáticamente. Las pruebas de aceptación están destinadas a verificar que al final de cada iteración, las Historias de Usuario cumplen con la funcionalidad asignada y satisfacen las necesidades del cliente.

#### 3.2.1 Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se realizan a partir de las historias de usuarios. Durante las iteraciones las historias de usuarios escogidas serán traducidas a prueba de aceptación. En ella se especifican, la perspectiva del cliente, y los escenarios para probar que la misma ha sido implementada correctamente.

Una historia de usuario puede tener todas las pruebas de aceptación que desee para asegurar su funcionamiento. El objetivo específico de esta prueba es garantizar que los requerimientos han sido cumplidos y que el sistema ha sido aceptable. La ejecución de los ensayos previamente diseñados, permitió evaluar las funcionalidades de la herramienta antes de pasarla a un entorno real de explotación.

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

---

A continuación se muestran algunos casos de pruebas realizadas al sistema, las demás forman parte de los anexos.

### Iteración 1:

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Autenticar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de Ejecución:</b> El usuario debe llenar correctamente los campos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta autenticar un usuario en el sistema con los datos válidos.	
<b>Resultados Esperados:</b> El usuario se autentica correctamente en el sistema.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Autenticar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de Ejecución:</b> El usuario debe llenar correctamente los campos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta autenticar un usuario en el sistema con los datos incorrectos.	
<b>Resultados Esperados:</b> El usuario no queda autenticado en el sistema y se muestra un mensaje de error indicando el fallo de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Modificar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad modificar usuario.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El usuario al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un usuario por un administrador del sistema.	
<b>Resultados Esperados:</b> Los datos del usuario son modificados correctamente. Muestra un mensaje de éxito de la operación.	

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Modificar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad modificar usuario.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El usuario al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un usuario por un administrador del sistema y no introduce todos los campos requeridos.	
<b>Resultados Esperados:</b> Los datos del usuario no son modificados. Muestra un mensaje de error indicando que debe llenar todos los campos.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P1	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Eliminar usuario.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar usuario.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El usuario a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar un usuario por un administrador del sistema.	
<b>Resultados Esperados:</b> El usuario es eliminado correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Adicionar instructora.	
<b>Descripción:</b> Prueba para la funcionalidad adicionar instructora.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta adicionar una instructora por un administrador del sistema.	
<b>Resultados Esperados:</b> La instructora se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4
Nombre: Adicionar instructora.	
Descripción: Prueba para la funcionalidad adicionar instructora.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
Pasos de Ejecución: Se intenta adicionar una instructora por un administrador del sistema que ya se encuentra registrada.	
Resultados Esperados: La instructora no se adiciona correctamente en el sistema. Muestra un mensaje de error indicando que ya existe la instructora.	

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Modificar instructora.	
Descripción: Prueba para la funcionalidad modificar instructora.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta acción. La instructora a la que se le modificarán los datos debe existir en el sistema.	
Pasos de Ejecución: Se intenta modificar los datos de una instructora por un administrador del sistema.	
Resultados Esperados: Los datos de la instructora son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Eliminar instructora.	
Descripción: Prueba para la funcionalidad eliminar instructora.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta acción. La instructora a eliminar debe existir en el sistema.	
Pasos de Ejecución: Se intenta eliminar una instructora por un administrador del sistema.	
Resultados Esperados: La instructora es eliminada correctamente. Muestra un mensaje de éxito de la operación.	

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
<b>Nombre:</b> Adicionar edificio.	
<b>Descripción:</b> Prueba para la funcionalidad adicionar edificio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta adicionar un edificio por un administrador del sistema.	
<b>Resultados Esperados:</b> El edificio se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
<b>Nombre:</b> Modificar edificio.	
<b>Descripción:</b> Prueba para la funcionalidad modificar edificio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El edificio al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un edificio por un administrador del sistema.	
<b>Resultados Esperados:</b> Los datos del edificio son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
<b>Nombre:</b> Eliminar edificio.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar edificio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El edificio a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar un edificio por un administrador del sistema.	
<b>Resultados Esperados:</b> El edificio es eliminado correctamente. Muestra un mensaje de éxito de la operación.	

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

Caso de Prueba de Aceptación	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Eliminar edificio.	
Descripción: Prueba para la funcionalidad eliminar edificio.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta acción. El edificio a eliminar debe existir en el sistema.	
Pasos de Ejecución: Se intenta eliminar un edificio por un administrador del sistema que tiene estudiantes e instructora asignado.	
Resultados Esperados: El edificio no es eliminado del sistema. Muestra un mensaje de error indicando que debe redistribuir los estudiantes e instructoras que tiene asignado.	

Durante las tres iteraciones de pruebas fueron identificadas 15 no conformidades, agrupadas en errores ortográficos y validación del sistema, las cuales fueron corregidas en tiempo sin que afectaran el desarrollo de las demás historias de usuarios. Es importante destacar que en una cuarta iteración no se detectaron no conformidades.

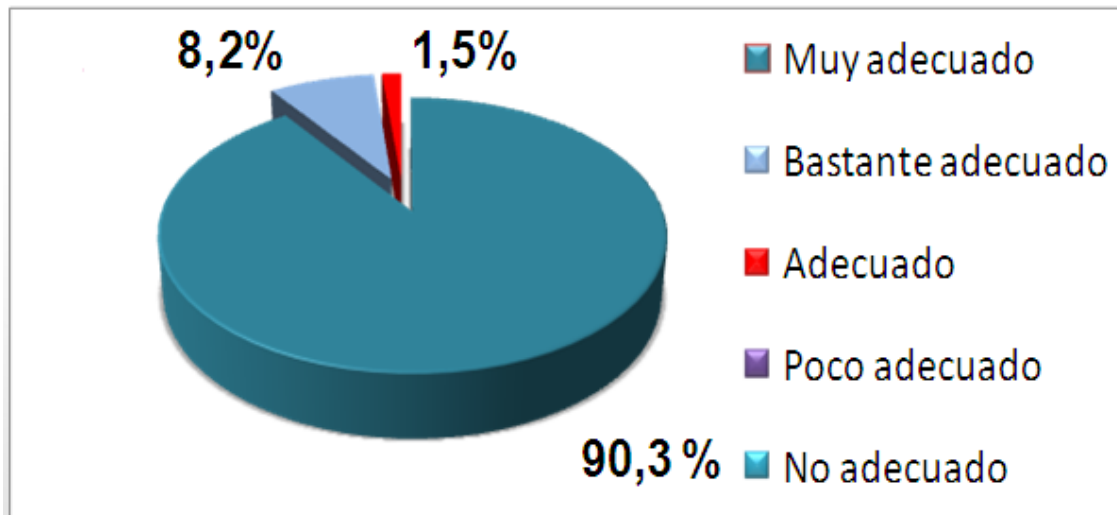
Una vez concluida la fase de pruebas del sistema y corregidas las no conformidades, se les aplicó un cuestionario de preguntas a una muestra de 20 personas (6 instructoras y 14 estudiantes), con el objetivo de conocer el nivel de satisfacción que tenían con el uso de la aplicación. La misma está integrada por 6 preguntas en las cuales se recoge toda la valoración respecto al sistema creado, resaltando lo positivo, negativo e interesante con el objetivo de poder lograr una propuesta lo más completa posible y sin error alguno. El cuestionario fue creado de forma tal que las respuestas se categorizarán en (Muy adecuado (A1), Bastante adecuado (A2), Adecuado (A3), Poco adecuado (A4) y No adecuado (A5)). Todos son equivalentes a un peso numérico de 5, 4, 3, 2, 1 respectivamente. Consultar el Anexo #1, para conocer en detalle el cuestionario de validación. Se muestra a continuación una tabulación de los resultados.

Encuestados	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6
1	5	5	5	4	4	5
2	5	5	5	5	5	5
3	4	5	5	5	5	5
4	5	5	5	5	5	5
5	5	5	5	4	5	5
6	5	4	5	5	5	4
7	5	4	3	5	5	5
8	5	5	5	5	4	4

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

9	5	5	4	5	5	5
10	5	5	5	5	5	5
11	5	5	5	5	5	5
12	5	5	5	5	5	5
13	4	5	5	5	5	5
14	5	5	5	5	5	5
15	5	5	5	5	5	5
16	5	5	4	5	4	5
17	5	5	5	5	5	5
18	5	5	5	5	5	5
19	5	5	5	5	5	5
20	5	5	5	5	5	5

Una vez concluido el proceso y registrado cada uno de los datos. Se llega a la conclusión de que el 90,3% de las respuestas fueron categorizadas de Muy adecuado, el 8,2% de Bastante adecuado y el 1,5% de Adecuado, no fueron detectadas respuestas de Poco adecuado ni No adecuado.



### 3.3 Consideraciones parciales.

En el desarrollo de este capítulo se lograron definir las técnicas de validación que serían usadas en la propuesta, haciendo uso de las revisiones de requerimientos, la construcción de prototipos y la generación de caso de pruebas, con el objetivo de comprobar que se estaba construyendo el sistema acorde con las necesidades y exigencias del usuario. Se aplicó un cuestionario de preguntas a una muestra de 20 personas arrojando resultados satisfactorios en el uso del mismo.

### **CONCLUSIONES GENERALES**

Al finalizar el presente trabajo de diploma se arriba a las siguientes conclusiones:

- La investigación realizada sobre las principales características y funcionamiento de la residencia estudiantil en la facultad 5, permitió obtener la información que sirvió como base para confeccionar la solución propuesta.
- Se realizó el diseño e implementación de la propuesta del sistema informático, siguiendo el ciclo de vida que propone XP como metodología de desarrollo, donde se definieron los conceptos necesarios para la formalización e implementación de la misma.
- Los resultados obtenidos por las pruebas realizadas, mostraron el cumplimiento de los requerimientos, logrando un correcto funcionamiento del sistema.



## RECOMENDACIONES

- Definir nuevas iteraciones que permitan incorporándole la información referente a la residencia de los profesores de la facultad.
- Integrar la solución con el sistema de la planificación de la guardia obrera-estudiantil, de manera que se cuente con un solo medio que centralice toda la información y facilite la toma de decisiones.
- Proponer el uso de esta aplicación en el resto de las facultades, estandarizándose los procesos que se automatizan en la residencia.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Aja Quiroga, Lourdes.** Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. ACIMED. 2002.
2. **Rodríguez Piña, Ramón Antonio; Peña Domínguez, Zaira.** *Cultural diagnosis of the organizational environment: a step prior to the introduction of an information management system.* ACIMED ISSN 1024-9435. 2008.
3. **GARCIA GIL, VICTOR LEONARDO.** Planeación en Administración. 2009.  
[http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/planeacionprocesoadministrativo/](http://www.elprisma.com/apuntes/administracion_de_empresas/planeacionprocesoadministrativo/)
4. **Milán, Telleria, Oliver, López Vázquez, Manuel, Salazar Alea, Caridad.** *Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Río "Hermanos Saiz Montes de Oca".* Pinar del Río, 2008.
5. **Dávila, Pérez, Yuniesky, Martínez, Padrón, Yadrian.** Sistema de Gestión de Información de la Facultad 8. Módulo para la gestión de la residencia estudiantil versión 2. Universidad de la Ciencias Informáticas, 2008.
6. **María A. Mendoza Sánchez.** Metodologías De Desarrollo De Software. 2004.  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
7. **Luzcila.** *Los 11 Mejores Frameworks gratuitos para Aplicaciones Web.* 2013  
<http://elbauldelprogramador.com/los-10-mejores-frameworks-gratis-de-aplicaciones-web/>
8. **Viveros Villaseñor, Bryan.** *JAVA MEDIA FRAMEWORK JMF.* 2010.  
<http://es.scribd.com/doc/226183010/Java-Media-Framework-1TV2>
9. **Potencier, Fabien y Zaninotto, François.** *Symfony 1.2 La guía Definitiva.* France: Sensio SA, 2008.
10. **Revilla, Luis.** *Guía CodeIgniter.* 2013.  
[https://play.google.com/store/apps/details?id=com.lectorpdf.codeigniter.pdf&hl=es\\_419](https://play.google.com/store/apps/details?id=com.lectorpdf.codeigniter.pdf&hl=es_419)

11. **Poyo Garrido, Alejandro.** *Comparativa Framework.* 2012.  
<http://es.scribd.com/doc/189203494/Comparativa-Framework>
12. Lenguaje de Programación. 2010. <http://es.scribd.com/doc/15270823/LENGUAJES-DE-PROGRAMACION-para-el-space>.
13. **Ajona, Guillermo Prado.** Página Web sobre HTML y CSS. [En línea]  
<https://belenus.unirioja.es/~guprado/pagweb/caraccss.html>.
14. **Valdeli, Ilario.** Elementos de JavaScript.2006.[http://www.htmlpoint.com/javascript/corso/js\\_02.htm](http://www.htmlpoint.com/javascript/corso/js_02.htm)
15. Procesamiento del lado del servidor.2009.<http://prograweb.com.mx/pweb/0301clienteServidor.html>
16. **Hernández, Prof: Leydis Dayana.** Introducción a las Bases de Datos. [En línea]  
<http://campus.dokeos.com/courses/BDFATLA/document/Introd.alasBasesdedatos.pdf?cidReq=BDFATLA>.
17. Investigación sobre PostgreSQL. 2006. <http://es.scribd.com/doc/36570462/postgreSQL-investigacion>.
18. **ORACLE.** Manual de referencia de MySQL 5. 2011a, vol. 2013. Available from  
Internet:<<http://manuales.guebs.com/mysql-5.0/introduction.html>>.
19. **FÉLIX, A. D. C. S.** El servidor de web Apache: Introducción práctica. Apache 1.x y 2.0 alpha. 2000.  
<http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf>.
20. **MENA, D.** Los IDEs de programación. 2011. Available from  
Internet:<<http://www.cursosporinternet.info/index.php/the-news/43-programacion/167-los-ides-de-programacion.html>>.
21. **ÁLVAREZ, M. Á.** Un IDE para el desarrollo de aplicaciones web, enfocado en Ajax y la Web 2.0. 2007, vol. 2013, no. 22 de enero de 2013. Available from Internet:<  
<http://www.desarrolloweb.com/articulos/aptana-studio.html>>.
22. **ZEND.** Zend Studio. 2013. Internet :<<http://www.zend.com/en/products/studio/features> >.
23. **López Michelone, Manuel.** 2013. PhpStorm. <http://www.unocero.com/2013/10/25/se-libera-phpstorm-7/>

24. **Pressman, Roger S.** "Ingeniería de Software. Un enfoque práctico." 5ta Edición. Madrid: Mac Graw Hill, 2001.
25. **Novelo Can, Carolina.** Requerimientos funcionales y no funcionales. 2010.  
<http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>
26. **Casas, Sandra y Reinaga, Héctor.** Identificación y Modelado de Aspectos Tempranos dirigido por Tarjetas de Responsabilidades y Colaboraciones. 2008.  
<http://www.oocities.org/espanol/profeprog2/INVPAPER25.pdf>.
27. **Bonatata, Maximiliano.** "Programación y Algoritmos.". Edición ilustrada. : MP. 2003.  
<http://sunshine.prod.uci.cu/search/programaci%C3%B3n%20orientada%2a%20objetos>. ISBN: 9789875261563.
28. **Reynoso, Billy.** Profundizando en Estilos de Arquitectura de Software. 2004.
29. **SÁNCHEZ, P. H.** Patrón MVC. Modelo Vista Controlador 2013, Available from Internet:<<http://www.devjoker.com/contenidos/articulos/525/Patroacuten-MVC-Modelo-Vista-Controlador.aspx>>.
30. **Diaz, Dustin y Harmes, Ross.** Pro JavaScript Design Patterns. New York: apress, 2008.

## BIBLIOGRAFÍA.

1. Pressman, Roger S., *Ingeniería de Software. Un enfoque práctico*. s.l.: Mc Graw Hill, 2001.
2. Sommerville, Ian. *Ingeniería del Software. Séptima Edición*. 2005.
3. Rodas Hinojosa, Raul. *Linuxcentro. Características de PHP*. [En línea]  
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
4. Ajona, Guillermo Prado. *Página Web sobre HTML y CSS. Página Web sobre HTML y CSS*.  
<https://belenus.unirioja.es/~guprado/pagweb/caraccss.html>.
5. Jacobson, I. y Booch, G. y Rumbaugh. *El Proceso Unificado de Desarrollo de software*. s.l.: Addison-Wesley, 2000. ISBN: 84-7829-036-2.
6. John. *Manuales de ayuda.com. Manuales de ayuda.com*. 2006.  
<http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.
7. Chaves, Michel Arias. *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*. Costa Rica: Revista InterSedes, 2007. Vol. VI, 10. ISSN 1409-4746.
8. Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. México: Universitarios, 1999. ISBN: 970-17-0261-1.
9. Brito Acuña, Karennny. *Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos*. [Tesis]. Cienfuegos, Cuba: s.n., 2009. Texto completo en [www.eumed.net/libros/2009c/584/](http://www.eumed.net/libros/2009c/584/).
10. Letelier, Patricio y Penadés, M<sup>a</sup> Carmen. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia.
11. Montaldo, Diego Fernando. *Patrones de Diseño de Arquitecturas de Software Enterprise*. Tesis. Noviembre 2005. Departamento de Computación. Facultad de Ingeniería. Universidad de Buenos Aires
12. Wright, E.B. y Forcier, R.C. *The Computer: A Tool for the Teacher*. Wadsworth, Estados Unidos: s.n., 1985.
13. Saint-Andre, Peter, Smith, Kevin y Troncon, Remko. *XMPP: The Definitive Guide*. San Peterburgo: O`Reilly Media, 2009.

14. Pérez, Javier Eguíluz. *Introducción a CSS*.
15. Gómez Argüello, Wilson Javier. *Metodología de desarrollo de software un enfoque práctico y global versión 1.0.11*. [En línea] <http://www.otcolombia.com/documentos/mds360-1.0.11-beta.pdf>.
16. Wahlin, Dan. *The Prototype Pattern -Techniques, Strategies and Patterns for Structuring JavaScript Code*. [En línea] 2011. <http://weblogs.asp.net/dwahlin/archive/2011/08/01/techniques-strategies-and-patterns-for-structuring-javascript-code-the-prototype-pattern.aspx>.
17. Shaw, Mary y Garlan, David. 2002. *Software Architecture: Perspectives on an Emerging Discipline*. 2002.
18. *Theoretical analysis of information systems*. 2011. <http://www.getcited.org>.
19. Wolf, Dewayne, Perry, E. y L., Alexander. 2002. *Software Engineering Notes*. 2002.

## ANEXOS.

### Anexo 1: Encuesta de validación.

**Estimado compañero/a:** Nos dirigimos a usted con el objetivo de desarrollar una encuesta que forma parte de la validación de la solución informática propuesta, que se estará desarrollando en el trabajo de diploma: *Sistema para la planificación y control de la residencia estudiantil en la Facultad 5*. En la cual mediremos con su ayuda y profesionalidad el correcto desarrollo y funcionamiento del mismo. De antemano se le asegura que nadie podrá saber quién es el encuestado. Valore el grado de factibilidad de cada pregunta o afirmación de acuerdo a la siguiente escala: Muy Adecuado (A1); Bastante Adecuado (A2); Adecuado (A3); Poco Adecuado (A4) y No adecuado (A5).

*Gracias de antemano.*

**Atentamente, Erlan Rodríguez Rodríguez.**

Preguntas	Criterios de los encuestados				
	A1	A2	A3	A4	A5
¿Considera usted que la información brindada en la solución informática es entendible?					
¿Considera usted que la gestión de la información desde la plataforma, sobre los diferentes procesos que se desarrollan en la residencia estudiantil, lo apoyan en el trabajo del área?					
¿Logra mayor intercambio con la dirección de la facultad mediante el uso del sistema?					
¿Considera usted que la aplicación cumple con un sistema de navegabilidad adecuado?					
¿Considera de beneficio para residencia de forma general contar con un sistema que apoye la planificación y gestión de información?					
¿Considera correcta la ortografía, redacción, estructura y concordancia de la información presentada?					

- Puede agregar alguna otra recomendación que considere necesario.

## Anexo 2: Historias de Usuarios.

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Adicionar instructora
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá adicionar Instructoras al sistema para ello debe introducir los siguientes datos: <ul style="list-style-type: none"><li>• Nombre (Obligatorio).</li><li>• Solapín.</li><li>• Provincia.</li><li>• Municipio.</li><li>• CI.</li></ul>	
<b>Observaciones:</b>	

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Modificar instructora
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá modificar los siguientes datos de una instructora determinada: <ul style="list-style-type: none"><li>• Solapín.</li><li>• Provincia.</li><li>• Municipio.</li><li>• CI.</li></ul>	
<b>Observaciones:</b>	



<b>Historia de Usuario</b>	
<b>Número:</b> 6	<b>Nombre:</b> Eliminar instructora
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Media	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá eliminar los datos de una instructora determinada.	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 7	<b>Nombre:</b> Adicionar Edificio
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá adicionar Edificios al sistema para ello debe introducir los siguientes datos: <ul style="list-style-type: none"> <li>• Número de edificio (Obligatorio).</li> <li>• Manzana (Obligatorio).</li> <li>• Cantidad de apartamentos (Obligatorio).</li> <li>• Instructora.</li> </ul>	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 8	<b>Nombre:</b> Modificar edificio
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá modificar los siguientes datos de un edificio determinado:	

<ul style="list-style-type: none"> <li>• Número de edificio.</li> <li>• Manzana.</li> <li>• Cantidad de apartamentos.</li> <li>• Instructora.</li> </ul>
<b>Observaciones:</b>

<b>Historia de Usuario</b>	
<b>Número:</b> 9	<b>Nombre:</b> Eliminar edificio
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá eliminar los datos de un edificio determinado.	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 10	<b>Nombre:</b> Adicionar apartamento
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá adicionar apartamentos al sistema para ello debe introducir los siguientes datos <ul style="list-style-type: none"> <li>• Número de apartamento (Obligatorio).</li> <li>• Edificio (Obligatorio).</li> <li>• Cantidad de cuartos (Obligatorio).</li> <li>• Estado (bueno, regular, malo).</li> <li>• Tipo (masculino, femenino, otro tipo).</li> </ul>	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 11	<b>Nombre:</b> Modificar apartamento
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá modificar los siguientes datos de un apartamento determinado: <ul style="list-style-type: none"> <li>• Número de apartamento.</li> <li>• Edificio.</li> <li>• Cantidad de cuartos.</li> <li>• Estado (bueno, regular, malo).</li> <li>• Tipo (masculino, femenino, otro tipo).</li> </ul>	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 12	<b>Nombre:</b> Eliminar apartamento
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá eliminar los datos de un apartamento determinado.	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 13	<b>Nombre:</b> Adicionar estudiante
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b>	

El administrador o la instructora podrán adicionar estudiantes al sistema para ello debe introducir los datos correspondientes al censo estudiantil:

- Nombre(requerido)
- Apellidos(requerido)
- Solapín(Requerido)
- Sexo(Requerido)
- Provincia(Requerido)
- Municipio(Requerido)
- Edificio(Requerido)
- Apto(Requerido)
- Año(Requerido)
- Brigada
- Teléfono
- Raza
- Organizaciones
- Estado Civil
- Cantidad de Hijos
- Hábitos
- Motivación Estudio
- Madre (ocupación, convivencia, salario)
- Padre(ocupación, convivencia, salario)
- Enfermedades
- Habilidades
- Intereses

**Observaciones:**

Historia de Usuario	
<b>Número:</b> 14	<b>Nombre:</b> Modificar estudiante
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b>	
El administrador o la instructora podrán modificar los siguientes datos de un estudiante determinado: <ul style="list-style-type: none"> <li>• Solapín.</li> <li>• Sexo.</li> <li>• Brigada.</li> <li>• Provincia.</li> <li>• Municipio.</li> </ul>	

- Edificio.
- Correo.
- Usuario.
- Contraseña.
- Apto.
- Año.
- Teléfono.
- Raza.
- Organizaciones.
- Estado Civil.
- Cantidad de Hijos.
- Hábitos.
- Motivación Estudio.
- Madre (ocupación, convivencia, salario).
- Padre (ocupación, convivencia, salario).
- Enfermedades.
- Habilidades.
- Intereses.

**Observaciones:** Las instructoras solo podrán modificar los datos de los estudiantes que forman parte de los edificios que atiende la misma.

#### Historia de Usuario

<b>Número:</b> 15	<b>Nombre:</b> Eliminar apartamento
-------------------	-------------------------------------

**Usuario:** Administrador, Instructora

<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
--------------------------------------	------------------------------------

<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
--------------------------------	------------------------------

**Programador responsable:** Erlán Rodríguez Rodríguez

#### Descripción:

El administrador o la instructora podrán eliminar los datos de un estudiante determinado.

**Observaciones:** Las instructoras solo podrán eliminar los datos de los estudiantes que forman parte de los edificios que atiende la misma.

#### Historia de Usuario

<b>Número:</b> 16	<b>Nombre:</b> Planificar limpiezas
-------------------	-------------------------------------

**Usuario:** Administrador, Instructora

<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
--------------------------------------	-----------------------------------

<b>Puntos de estimado:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán planificarle limpiezas a los estudiantes:	
<b>Datos de limpieza:</b> <ul style="list-style-type: none"> <li>• Edificio (requerido).</li> <li>• Apartamento (requerido).</li> <li>• Nombre y apellidos del estudiante (requerido).</li> <li>• Área que se limpia (sala, comedor, cocina, baño, patio)(requerido).</li> <li>• Fecha de limpieza (requerido).</li> <li>• Tipo de limpieza (general, normal) (requerido).</li> <li>• Evaluación (mal, regular, bien, muy bien, excelente).</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán planificarles limpieza a los estudiantes que forman parte de los edificios que atiende la misma.	

<b>Historia de Usuario</b>	
<b>Número:</b> 17	<b>Nombre:</b> Modificar limpieza
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán modificar los siguientes datos de una limpieza determinada: <ul style="list-style-type: none"> <li>• Área que se limpia (sala, comedor, cocina, baño, patio).</li> <li>• Fecha de limpieza.</li> <li>• Tipo de limpieza (general, normal).</li> <li>• Evaluación (mal, regular, bien, muy bien, excelente).</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán modificar los datos de las limpiezas planificadas a sus estudiantes.	

<b>Historia de Usuario</b>	
<b>Número:</b> 18	<b>Nombre:</b> Eliminar limpieza
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1

<b>Programador responsable:</b> Erlán Rodríguez Rodríguez
<b>Descripción:</b> El administrador o la instructora podrán eliminar los datos de una limpieza determinada.
<b>Observaciones:</b> Las instructoras solo podrán eliminar las limpiezas planificadas a sus estudiantes.

<b>Historia de Usuario</b>	
<b>Número:</b> 19	<b>Nombre:</b> Planificar cuartelería
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán planificarle cuartelería a los estudiantes:	
<b>Datos de cuartelería:</b>	
<ul style="list-style-type: none"> <li>• Edificio (requerido).</li> <li>• Apartamento (requerido).</li> <li>• Nombre y apellidos del estudiante (requerido).</li> <li>• Fecha (requerido).</li> <li>• Evaluación (mal, regular, bien, muy bien, excelente).</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán planificarles cuartelería a los estudiantes que forman parte de los edificios que atiende la misma.	

<b>Historia de Usuario</b>	
<b>Número:</b> 20	<b>Nombre:</b> Modificar cuartelería
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán modificar los siguientes datos de una cuartelería determinada:	
<ul style="list-style-type: none"> <li>• Fecha (requerido).</li> <li>• Evaluación (mal, regular, bien, muy bien, excelente).</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán modificar los datos de las cuarterías planificadas a sus estudiantes.	

<b>Historia de Usuario</b>	
<b>Número:</b> 21	<b>Nombre:</b> Eliminar cuartelería
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán eliminar los datos de una cuartelería determinada.	
<b>Observaciones:</b> Las instructoras solo podrán eliminar las cuartelerías planificadas a sus estudiantes.	

<b>Historia de Usuario</b>	
<b>Número:</b> 22	<b>Nombre:</b> Registrar evaluación
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán evaluar el desempeño mensual de los estudiantes en la residencia mediante los siguientes aspectos:	
<b>Datos del estudiante:</b> <ul style="list-style-type: none"> <li>• Edificio (requerido).</li> <li>• Apartamento (requerido).</li> <li>• Nombre y apellidos del estudiante (requerido).</li> <li>• Mes a evaluar (requerido).</li> </ul>	
<b>Aspectos a evaluar:</b> <ul style="list-style-type: none"> <li>• Limpieza y organización del apartamento.</li> <li>• Limpieza y organización del dormitorio.</li> <li>• Cumplimiento con la guardia estudiantil.</li> <li>• Observaciones (Algún comentario por parte de la instructora).</li> <li>• Actitud: <ul style="list-style-type: none"> <li>-Cuidado de la propiedad social.</li> <li>-Iniciativas positivas para el apartamento.</li> <li>-Buen comportamiento y disciplina.</li> </ul> </li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán evaluar a los estudiantes que forman parte de los	



edificios que atiende la misma.

#### Historia de Usuario

**Número:** 23

**Nombre:** Modificar evaluación

**Usuario:** Administrador, Instructora

**Prioridad en el negocio:** Alta

**Riesgo de desarrollo:** Medio

**Puntos de estimado:** 0.5

**Iteración asignada:** 1

**Programador responsable:** Erlán Rodríguez Rodríguez

#### Descripción:

El administrador o la instructora podrán modificar los siguientes datos de una evaluación determinada:

- Mes a evaluar (requerido).

#### Aspectos a evaluar:

- Limpieza y organización del apartamento.
- Limpieza y organización del dormitorio.
- Cumplimiento con la guardia estudiantil.
- Observaciones (Algún comentario por parte de la instructora).
- Actitud:
  - Cuidado de la propiedad social.
  - Iniciativas positivas para el apartamento.
  - Buen comportamiento y disciplina.

**Observaciones:** Las instructoras solo podrán modificar los datos de las evaluaciones planificadas a sus estudiantes.

#### Historia de Usuario

**Número:** 24

**Nombre:** Eliminar evaluación

**Usuario:** Administrador, Instructora

**Prioridad en el negocio:** Alta

**Riesgo de desarrollo:** Medio

**Puntos de estimado:** 0.5

**Iteración asignada:** 1

**Programador responsable:** Erlán Rodríguez Rodríguez

#### Descripción:

El administrador o la instructora podrán eliminar los datos de una evaluación determinada.

**Observaciones:** Las instructoras solo podrán eliminar las evaluaciones planificadas a sus estudiantes.

<b>Historia de Usuario</b>	
<b>Número:</b> 25	<b>Nombre:</b> Registrar llave
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán registrar los estudiantes que tienen llaves de los apartamentos estudiantiles, introduciendo los siguientes datos: <ul style="list-style-type: none"> <li>• Edificio (requerido).</li> <li>• Apartamento (requerido).</li> <li>• Nombre y apellidos del estudiante (requerido).</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán registrar a los estudiantes que tengan llaves de los apartamentos que atiende la misma.	

<b>Historia de Usuario</b>	
<b>Número:</b> 26	<b>Nombre:</b> Modificar llave
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán modificar los siguientes datos de un registro de llave determinado: <ul style="list-style-type: none"> <li>• Edificio.</li> <li>• Apartamento.</li> <li>• Nombre y apellidos del estudiante.</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán modificar los datos de las llaves registradas a sus estudiantes.	

<b>Historia de Usuario</b>	
<b>Número:</b> 27	<b>Nombre:</b> Eliminar llave

<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán eliminar los datos de un registro de llave determinado.	
<b>Observaciones:</b> Las instructoras solo podrán eliminar las llaves registradas a sus estudiantes.	

<b>Historia de Usuario</b>	
<b>Número:</b> 28	<b>Nombre:</b> Registrar medio
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador o la instructora podrán registrar los medios que forman parte de los apartame estudiantiles, introduciendo los siguientes datos:	
<ul style="list-style-type: none"> <li>• Edificio (requerido).</li> <li>• Apartamento (requerido).</li> <li>• Rótulo (requerido).</li> <li>• Fecha de alta (requerido).</li> <li>• Fecha actual (requerido).</li> <li>• Estado (malo, regular, bueno) (requerido).</li> <li>• Descripción.</li> <li>• Observaciones.</li> </ul>	
<b>Observaciones:</b> Las instructoras solo podrán registrar a los medios que formen parte de los apartamentos que atiende la misma.	

<b>Historia de Usuario</b>	
<b>Número:</b> 29	<b>Nombre:</b> Modificar medio
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	

**Descripción:**

El administrador o la instructora podrán modificar los siguientes datos de un medio determinado:

- Edificio.
- Apartamento.
- Rótulo.
- Fecha de alta.
- Fecha actual.
- Estado (malo, regular, bueno).
- Descripción.
- Observaciones.

**Observaciones:** Las instructoras solo podrán modificar los datos los medios que formen parte de los apartamentos que atiende la misma.

**Historia de Usuario**

**Número:** 30

**Nombre:** Eliminar medio

**Usuario:** Administrador, Instructora

**Prioridad en el negocio:** Alta

**Riesgo de desarrollo:** Medio

**Puntos de estimado:** 0.5

**Iteración asignada:** 1

**Programador responsable:** Erlán Rodríguez Rodríguez

**Descripción:**

El administrador o la instructora podrán eliminar los datos de un medio determinado.

**Observaciones:** Las instructoras solo podrán eliminar los medios que formen parte de los apartamentos que atiende la misma.

**Historia de Usuario**

**Número:** 31

**Nombre:** Generar actas de responsabilidad material

**Usuario:** Administrador, Instructora

**Prioridad en el negocio:** Alta

**Riesgo de desarrollo:** Alto

**Puntos de estimado:** 1

**Iteración asignada:** 1

**Programador responsable:** Erlán Rodríguez Rodríguez

**Descripción:**

El administrador o la instructora podrán generar las actas de responsabilidad material de los apartamentos estudiantiles. El acta contara con los siguientes datos:

**Acta de responsabilidad material:**

- Nombres, apellidos y firmas de cada estudiante que pertenezca al apartamento.
- El listado de los medios del apartamento con los siguientes datos:

<ul style="list-style-type: none"> <li>-Rótulo.</li> <li>-Estado.</li> <li>-Descripción.</li> <li>• Firma de la instructora.</li> <li>• Firma del técnico de medios básicos.</li> </ul>
<p><b>Observaciones:</b> Las instructoras solo podrán generar las actas de los apartamentos que atiende la misma.</p>

<b>Historia de Usuario</b>	
<b>Número:</b> 32	<b>Nombre:</b> Planificar parada de beca
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<p><b>Descripción:</b> El administrador podrá planificar las paradas de beca, introduciendo los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Nombre de la parada (requerido).</li> <li>• Fecha a realizarse (requerido).</li> <li>• Hora (requerido).</li> </ul>	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 33	<b>Nombre:</b> Modificar parada de beca
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<p><b>Descripción:</b> El administrador podrá modificar los siguientes datos de una parada de beca determinada:</p> <ul style="list-style-type: none"> <li>• Nombre de la parada.</li> <li>• Fecha a realizarse.</li> <li>• Hora.</li> </ul>	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 34	<b>Nombre:</b> Eliminar parada de beca
<b>Usuario:</b> Administrador	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de estimado:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El administrador podrá eliminar los datos de una parada de beca determinada.	
<b>Observaciones:</b>	

<b>Historia de Usuario</b>	
<b>Número:</b> 35	<b>Nombre:</b> Generar reportes
<b>Usuario:</b> Administrador, Instructoras	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 7	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<b>Descripción:</b> El sistema permitirá generar los siguientes reportes:	
<b>Reporte de usuarios:</b>	
<ul style="list-style-type: none"> <li>• Nombre y Apellidos.</li> <li>• Usuario (UCI).</li> <li>• Correo electrónico.</li> <li>• Rol.</li> </ul>	
<b>Reporte de instructoras:</b>	
<ul style="list-style-type: none"> <li>• Nombre y Apellidos.</li> <li>• Solapín.</li> <li>• Provincia.</li> <li>• Municipio.</li> <li>• CI.</li> </ul>	
<b>Reporte de edificios:</b>	
<ul style="list-style-type: none"> <li>• Número de edificio.</li> <li>• Instructora.</li> <li>• Cantidad de apartamentos.</li> <li>• Total de apartamentos en función de estudiantes.</li> <li>• Total de estudiantes por sexo.</li> <li>• Total de estudiantes por año.</li> <li>• Total de estudiantes por edificio.</li> </ul>	
<b>Reporte de apartamentos:</b>	

- Numero de apartamento.
- Edificio al que pertenece.
- Cantidad de cuartos.
- Estado (bueno, regular, malo).
- Tipo (masculino, femenino, otro tipo).
- Cantidad total de estudiantes.

**Reporte de estudiantes:**

- Nombre y apellidos.
- Edificio al que pertenece.
- Edificio al que pertenece.
- Solapín.
- Usuario.
- Teléfono.
- Brigada.

**Reporte de limpiezas:**

- Edificio.
- Apartamento.
- Nombre y apellidos del estudiante.
- Fecha.
- Área.
- Tipo (general, normal).
- Evaluación (mal, regular, bien, muy bien, excelente).

**Reporte de cuarterías:**

- Edificio.
- Apartamento.
- Nombre y apellidos del estudiante.
- Fecha.
- Evaluación (mal, regula, bien, muy bien, excelente).

**Reporte de evaluaciones:**

- Edificio.
- Apartamento.
- Nombre y apellidos del estudiante.
- Mes evaluado.
- Evaluaciones sobre limpieza y organización del apartamento.
- Evaluaciones sobre limpieza y organización del dormitorio.
- Evaluaciones sobre cumplimiento con la guardia estudiantil.
- Evaluaciones sobre actitud.
- Observaciones.

**Reporte de llaves:**

- Edificio.
- Apartamento.
- Nombre y apellidos del estudiante.

**Reporte de medios:**

- Edificio.
- Apartamento.

<ul style="list-style-type: none"> <li>• Rótulo.</li> <li>• Fecha de alta.</li> <li>• Fecha actual.</li> <li>• Estado (malo, regular, bueno).</li> <li>• Descripción.</li> <li>• Observaciones.</li> </ul> <p><b>Reporte de paradas de beca:</b></p> <ul style="list-style-type: none"> <li>• Nombre de la parada.</li> <li>• Fecha a realizarse.</li> <li>• Hora.</li> </ul> <p><b>Observaciones:</b> Las instructoras solo podrán generar los reportes relacionados con las funcionalidades a las que tengan permitido acceder.</p>
---

<b>Historia de Usuario</b>	
<b>Número:</b> 36	<b>Nombre:</b> Graficas informativas
<b>Usuario:</b> Administrador, Instructora	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos de estimado:</b> 7	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Erlán Rodríguez Rodríguez	
<p><b>Descripción:</b> El sistema automáticamente podrá generar graficas de varios tipos(barras, pastel) mediante las cuales el administrador y las instructoras podrán consultar los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Cantidad de apartamentos por edificio.</li> <li>• Total de apartamentos en función de estudiantes por edificio.</li> <li>• Total de estudiantes por año.</li> <li>• Cantidad de apartamentos según su estado.</li> <li>• Cantidad de apartamentos según su tipo.</li> <li>• Cantidad de estudiantes por evaluaciones de cada edificio.</li> <li>• Cantidad de estudiantes por evaluaciones de cada apartamento.</li> <li>• Cantidad de limpiezas por evaluación.</li> </ul> <p><b>Observaciones:</b> Las instructoras solo podrán observar los datos de las gráficas relacionadas con las funcionalidades a las que ella pueda acceder.</p>	

### Anexo 3: Pruebas de Aceptación.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU10_P1	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Adicionar apartamento.	



<b>Descripción:</b> Prueba para la funcionalidad adicionar apartamento.
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.
<b>Pasos de Ejecución:</b> Se intenta adicionar un apartamento por un administrador del sistema.
<b>Resultados Esperados:</b> El apartamento se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.

Caso de Prueba de Aceptación	
<b>Código:</b> HU11_P1	<b>Historia de Usuario:</b> 11
<b>Nombre:</b> Modificar apartamento.	
<b>Descripción:</b> Prueba para la funcionalidad modificar apartamento.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El apartamento al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un apartamento por un administrador del sistema.	
<b>Resultados Esperados:</b> Los datos del apartamento son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU12_P1	<b>Historia de Usuario:</b> 12
<b>Nombre:</b> Eliminar apartamento.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar apartamento.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El apartamento a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar un apartamento por un administrador del sistema.	
<b>Resultados Esperados:</b> El apartamento es eliminado correctamente. Muestra un mensaje de éxito de la operación.	

## Iteración 2:

Caso de Prueba de Aceptación	
<b>Código:</b> HU13_P1	<b>Historia de Usuario:</b> 13

<b>Nombre:</b> Adicionar estudiante.
<b>Descripción:</b> Prueba para la funcionalidad adicionar estudiante.
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.
<b>Pasos de Ejecución:</b> Se intenta adicionar un estudiante por un administrador o instructora del sistema.
<b>Resultados Esperados:</b> El estudiante se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.

Caso de Prueba de Aceptación	
<b>Código:</b> HU14_P1	<b>Historia de Usuario:</b> 14
<b>Nombre:</b> Modificar estudiante.	
<b>Descripción:</b> Prueba para la funcionalidad modificar estudiante.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un estudiante por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> Los datos del estudiante son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU15_P1	<b>Historia de Usuario:</b> 15
<b>Nombre:</b> Eliminar estudiante.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar estudiante.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El estudiante a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar un estudiante por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> El estudiante es eliminado correctamente. Muestra un mensaje de éxito de la operación.	

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

<b>Código:</b> HU16_P1	<b>Historia de Usuario:</b> 16
<b>Nombre:</b> Planificar limpieza.	
<b>Descripción:</b> Prueba para la funcionalidad planificar limpieza.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta planificar una limpieza por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La limpieza se planifica correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU17_P1	<b>Historia de Usuario:</b> 17
<b>Nombre:</b> Modificar limpieza.	
<b>Descripción:</b> Prueba para la funcionalidad modificar limpieza.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La limpieza a la que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de una limpieza por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> Los datos de la limpieza son modificados correctamente. Muestra un mensaje de éxito de la operación.	

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU18_P1	<b>Historia de Usuario:</b> 18
<b>Nombre:</b> Eliminar limpieza.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar limpieza.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La limpieza a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar una limpieza por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La limpieza es eliminada correctamente. Muestra un mensaje de éxito de la operación.	

<b>Caso de Prueba de Aceptación</b>
-------------------------------------

<b>Código:</b> HU19_P1	<b>Historia de Usuario:</b> 19
<b>Nombre:</b> Planificar cuartelería.	
<b>Descripción:</b> Prueba para la funcionalidad planificar cuartelería.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta planificar una cuartelería por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La cuartelería se planifica correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU20_P1	<b>Historia de Usuario:</b> 20
<b>Nombre:</b> Modificar cuartelería.	
<b>Descripción:</b> Prueba para la funcionalidad modificar cuartelería.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La cuartelería a la que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de una cuartelería por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> Los datos de la cuartelería son modificados correctamente. Muestra un mensaje de éxito de la operación.	

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU21_P1	<b>Historia de Usuario:</b> 21
<b>Nombre:</b> Eliminar cuartelería.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar cuartelería.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La cuartelería a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar una cuartelería por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La cuartelería es eliminada correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU22_P1	<b>Historia de Usuario:</b> 22
<b>Nombre:</b> Registrar evaluación.	
<b>Descripción:</b> Prueba para la funcionalidad registrar evaluación.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta registrar una evaluación por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La evaluación se registra correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU23_P1	<b>Historia de Usuario:</b> 23
<b>Nombre:</b> Modificar evaluación.	
<b>Descripción:</b> Prueba para la funcionalidad modificar evaluación.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La evaluación a la que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de una evaluación por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> Los datos de la evaluación son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU24_P1	<b>Historia de Usuario:</b> 24
<b>Nombre:</b> Eliminar evaluación.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar evaluación.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La evaluación a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar una evaluación por un administrador o instructora del sistema.	

**Resultados Esperados:** La evaluación es eliminada correctamente. Muestra un mensaje de éxito de la operación.

### Iteración 3:

Caso de Prueba de Aceptación	
<b>Código:</b> HU25_P1	<b>Historia de Usuario:</b> 25
<b>Nombre:</b> Registrar llave.	
<b>Descripción:</b> Prueba para la funcionalidad registrar llave.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta registrar una llave por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> La llave se registra correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU26_P1	<b>Historia de Usuario:</b> 26
<b>Nombre:</b> Modificar llave evaluación.	
<b>Descripción:</b> Prueba para la funcionalidad modificar llave.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La llave a la que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de una llave por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> Los datos de la llave son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU27_P1	<b>Historia de Usuario:</b> 27
<b>Nombre:</b> Eliminar llave.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar llave.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. La llave a eliminar debe existir en el sistema.	

<b>Pasos de Ejecución:</b> Se intenta eliminar una llave por un administrador o instructora del sistema.
<b>Resultados Esperados:</b> La llave es eliminada correctamente. Muestra un mensaje de éxito de la operación.

Caso de Prueba de Aceptación	
<b>Código:</b> HU28_P1	<b>Historia de Usuario:</b> 28
<b>Nombre:</b> Adicionar medio.	
<b>Descripción:</b> Prueba para la funcionalidad adicionar medio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.	
<b>Pasos de Ejecución:</b> Se intenta adicionar un medio por un administrador o instructora del sistema.	
<b>Resultados Esperados:</b> El medio se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU29_P1	<b>Historia de Usuario:</b> 29
<b>Nombre:</b> Modificar medio.	
<b>Descripción:</b> Prueba para la funcionalidad modificar medio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El medio al que se le modificarán los datos debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta modificar los datos de un medio por un administrador del sistema.	
<b>Resultados Esperados:</b> Los datos del medio son modificados correctamente. Muestra un mensaje de éxito de la operación.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU30_P1	<b>Historia de Usuario:</b> 30
<b>Nombre:</b> Eliminar medio.	
<b>Descripción:</b> Prueba para la funcionalidad eliminar medio.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos necesarios para realizar esta acción. El medio a eliminar debe existir en el sistema.	
<b>Pasos de Ejecución:</b> Se intenta eliminar un medio por un administrador o instructora del sistema.	

**Resultados Esperados:** El medio es eliminado correctamente. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU31\_P1

**Historia de Usuario:** 31

**Nombre:** Generar acta de responsabilidad material.

**Descripción:** Prueba para la funcionalidad generar acta de responsabilidad material.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. Se proveerán todos los datos requeridos.

**Pasos de Ejecución:** Se intenta generar un acta de responsabilidad material por un administrador o instructora del sistema.

**Resultados Esperados:** El acta de responsabilidad material es generada correctamente. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU32\_P1

**Historia de Usuario:** 32

**Nombre:** Adicionar parada de beca.

**Descripción:** Prueba para la funcionalidad adicionar parada de beca.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. Debe ingresar los datos requeridos.

**Pasos de Ejecución:** Se intenta adicionar una parada de beca por un administrador o instructora del sistema.

**Resultados Esperados:** La parada de beca se adiciona correctamente en el sistema. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU33\_P1

**Historia de Usuario:** 33

**Nombre:** Modificar parada de beca.

**Descripción:** Prueba para la funcionalidad modificar parada de beca.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. La parada de beca a la que se le modificarán los datos debe existir en el sistema.

**Pasos de Ejecución:** Se intenta modificar los datos de una parada de beca por un administrador o



instructora del sistema.

**Resultados Esperados:** Los datos de la parada de beca son modificados correctamente. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU34\_P1

**Historia de Usuario:** 34

**Nombre:** Eliminar parada de beca.

**Descripción:** Prueba para la funcionalidad eliminar parada de beca.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. La parada de beca a eliminar debe existir en el sistema.

**Pasos de Ejecución:** Se intenta eliminar una parada de beca por un administrador o instructora del sistema.

**Resultados Esperados:** La parada de beca es eliminada correctamente. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU35\_P1

**Historia de Usuario:** 35

**Nombre:** Generar reportes.

**Descripción:** Prueba para la funcionalidad generar reportes.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. Se proveerán todos los datos requeridos.

**Pasos de Ejecución:** Se intenta generar reportes por un administrador o instructora del sistema.

**Resultados Esperados:** Los reportes son generados correctamente. Muestra un mensaje de éxito de la operación.

#### Caso de Prueba de Aceptación

**Código:** HU36\_P1

**Historia de Usuario:** 36

**Nombre:** Gráficas informativas.

**Descripción:** Prueba para la funcionalidad gráficas informativas.

**Condiciones de Ejecución:** El usuario debe tener los permisos necesarios para realizar esta acción. Se proveerán todos los datos requeridos.

**Pasos de Ejecución:** Se intenta observar las gráficas informativas por un administrador o instructora del sistema.

**Resultados Esperados:** Las gráficas son mostradas correctamente. Muestra un mensaje de éxito de la operación.