



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4



**Sistema informático de apoyo a los procesos “Evaluación del estudiante” y
“Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil
2 de la Universidad de las Ciencias Informáticas**

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

Autores: Michelle Pérez Acosta

Claudia Milagros de León Martínez

Tutor(es): Ing. Yusdel Meriño Almaguer

Ing. Roanny Lamas López

La Habana, junio de 2015

“Año 57 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con el mismo.

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

Autor(es):

Claudia Milagros de León Martínez

Michelle Pérez Acosta

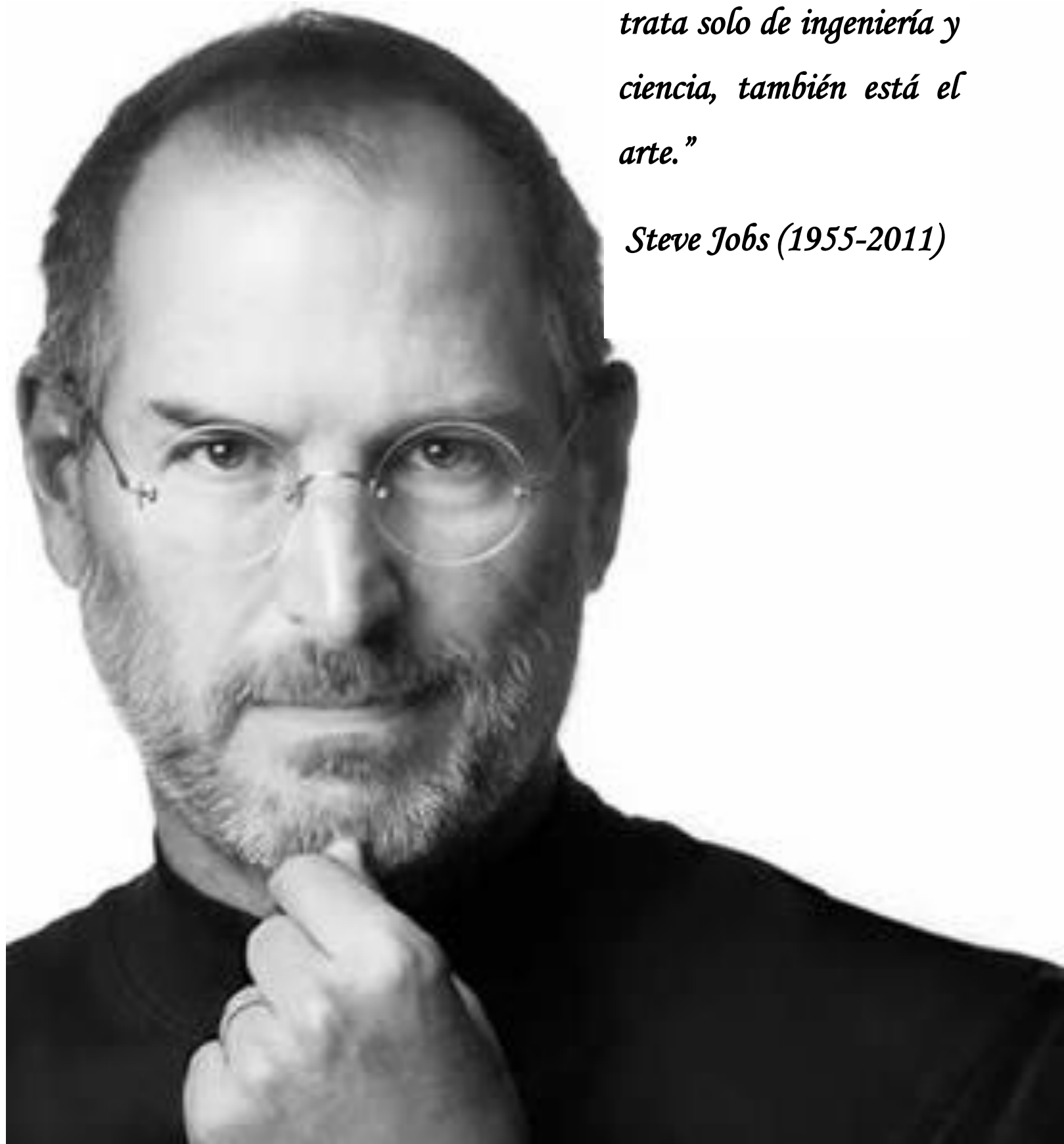
Tutores:

Ing. Yusdel Meriño Almaguer

Ing. Roanny Lamas López

“Sucede más de lo que imaginas, porque no se trata solo de ingeniería y ciencia, también está el arte.”

Steve Jobs (1955-2011)



Dedicatorias

De Claudia:

Con todo mi cariño y amor dedico este trabajo a los seres que concibieron mi existencia, a los que me motivaron en cada momento, a los que me dieron la mano cuando no veía la salida del camino, a ustedes por siempre mi corazón y mi agradecimiento.

Mamá y Papá

De Michelle:

Dedico este trabajo con todo el amor que existe en el mundo a aquellas personas a las cuales les debo la vida, por ser el motor impulsor y la razón por la cual hoy se forja una nueva profesional.

Mis padres

Agradecimientos

De Claudia:

El desarrollo de este trabajo lleva implícito una serie de gratos esfuerzos y apoyo de muchas personas a las cuales quiero agradecer:

A mis Padres que con su confianza y amor pusieron todo su empeño en mi formación como profesional.

A mis abuelos, por su cariño y dedicación.

A mis tutores por su apoyo, preocupación y eterna confianza en la realización de este trabajo.

A mi compañera de tesis por su comprensión y paciencia.

A mis amigos y compañeros de estudio durante toda mi carrera por los buenos momentos vividos.

A Todas aquellas personas que de una forma u otra contribuyeron a mi formación como profesional.

De Michelle:

A mi familia por apoyarme incondicionalmente en todo momento.

A mis madrinas por estar ahí cuando las necesito.

A mis tutores por tanta dedicación y compromiso durante la realización de esta investigación.

A mi compañera de tesis por compartir tantos momentos juntas.

A todas aquellas personas que de una forma u otra ayudaron a la realización de este sueño.

Resumen

En Cuba existen muchas instituciones educativas donde se acoge una amplia matrícula de estudiantes de pregrado. La Universidad de las Ciencias Informáticas (UCI), constituye uno de estos centros. La misma cuenta con tres Direcciones de Residencia las cuales se encargan de ubicar al personal interno de la universidad así como gestionar el estado físico de las becas y el cumplimiento de las normas establecidas en estas Direcciones para propiciar un adecuado comportamiento por parte de los residentes. En la Dirección de la Residencia Estudiantil 2 se gestionan varios procesos que involucran la vida del becario. En un estudio realizado se identificaron dificultades con la recopilación y el almacenamiento de la información que se gestiona en dos de los procesos, por tanto, la presente investigación propone un sistema para la gestión de la información asociada a los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI. Durante el progreso de la investigación se justifican las principales características que debe poseer el sistema, guiado por los principios y reglas que establece la metodología de desarrollo Programación Extrema. Como lenguajes de programación se utilizan PHP, JavaScript, HTML 5 y CSS 3, PhpStorm 8.0 como entorno de desarrollo y como servidor de aplicaciones Apache 2.2.22, como lenguaje de modelado UML 2.1, Visual Paradigm 8.0 como herramienta CASE y como servidor de base de datos MySQL 5.0.15. Una vez desarrollado el sistema se efectuaron las pruebas de *software* a la propuesta de solución, donde todas las No conformidades fueron resueltas, lo cual valida la calidad de la solución propuesta logrando una mayor satisfacción por parte del cliente y garantizando de esta manera el cumplimiento de los objetivos planteados.

Palabras clave: gestión, información, residencia estudiantil.

Índice

| | |
|--|----|
| Introducción | 4 |
| Capítulo 1: Fundamentación Teórica | 9 |
| 1.1 Conceptos generales relacionados con la investigación | 9 |
| 1.1.1 Proceso “Evaluación del estudiante” | 10 |
| 1.1.2 Proceso “Censo, capacidad y matrícula” | 10 |
| 1.2 Análisis de soluciones existentes | 10 |
| 1.2.1 Soluciones similares existentes a nivel internacional | 11 |
| 1.2.2 Soluciones similares existentes a nivel nacional | 11 |
| 1.3 Selección del Sistema Gestor de Contenido | 12 |
| 1.4 Estudio de las metodologías y estándares para el desarrollo de <i>software</i> | 16 |
| 1.4.1 Programación Extrema (XP) | 16 |
| 1.4.2 Lenguaje Unificado de Modelado (UML) | 18 |
| 1.5 Selección de las herramientas y lenguajes de desarrollo | 18 |
| 1.5.1 Lenguajes de programación | 19 |
| 1.5.2 Selección de la librería o <i>framework</i> JavaScript | 21 |
| 1.5.3 Selección del <i>framework</i> CSS | 22 |
| 1.5.4 Herramienta CASE de modelado con UML | 23 |
| 1.5.5 Servidor de Bases de Datos | 24 |
| 1.5.6 Servidor Web | 25 |
| 1.5.7 Entorno de desarrollo integrado | 26 |
| 1.6 Conclusiones | 27 |
| Capítulo 2: Exploración y Planificación | 28 |
| 2.1 Descripción de la propuesta de solución | 28 |
| 2.2 Usuarios del sistema | 29 |

| | |
|--|----|
| 2.3 Diagrama conceptual del negocio..... | 29 |
| 2.4 Funcionalidades del sistema | 30 |
| 2.5 Características del sistema..... | 32 |
| 2.6 Exploración..... | 32 |
| 2.6.1 Historia de usuario (HU)..... | 32 |
| 2.6.2 Historias de usuario de la propuesta de solución | 33 |
| 2.6.3 Prototipos de Interfaz de Usuario | 35 |
| 2.7 Planificación | 37 |
| 2.7.1 Plan de Entregas | 38 |
| 2.7.2 Plan de Iteraciones | 39 |
| 2.8 Conclusiones..... | 40 |
| Capítulo 3 Desarrollo y Pruebas | 42 |
| 3.1 Descripción de la arquitectura | 42 |
| 3.2 Gestión de tipos de contenidos en Drupal | 44 |
| 3.3 Gestión de la seguridad..... | 46 |
| 3.4 Patrones de diseño..... | 47 |
| 3.4.1 Patrones <i>Gang of Four</i> | 48 |
| 3.5 Diagrama Entidad Relación (DER) | 49 |
| 3.6 Estilos y estándares de codificación | 50 |
| 3.7 Tareas de ingeniería..... | 52 |
| 3.8 Interfaces del sistema..... | 55 |
| 3.9 Pruebas..... | 57 |
| 3.9.1 Desarrollo dirigido por pruebas (TDD) | 57 |
| 3.9.2 Pruebas unitarias..... | 57 |
| 3.9.3 Pruebas de aceptación..... | 59 |
| 3.10 Análisis de los resultados de las pruebas | 64 |

| | |
|---|----|
| 3.11 Conclusiones..... | 64 |
| Conclusiones generales y Recomendaciones..... | 66 |
| Conclusiones generales | 66 |
| Recomendaciones..... | 66 |
| Referencias Bibliográficas..... | 67 |
| Anexos..... | 73 |
| Anexo 1: Gráficos..... | 73 |
| Anexo 2: Historias de usuario..... | 75 |
| Anexo 3: Tareas de ingeniería..... | 78 |
| Anexo 4: Casos de pruebas de aceptación | 85 |

Introducción

Con el constante desarrollo de la ciencia y la tecnología, el mundo experimentó un cambio que marcó el surgimiento de una nueva era, convirtiéndose ambas en una imperiosa necesidad para alcanzar el desarrollo. Uno de los principales adelantos de estas esferas es el avance de la informática, que posibilita la informatización de las diferentes esferas de la sociedad como son la educación, la salud, el deporte y la producción, estos adelantos permiten además, el desarrollo de *software* para agilizar los procesos que se deseen informatizar, con el objetivo de disminuir costos económicos, recursos humanos y así lograr una mayor efectividad.

Las Tecnologías de la Información y las Comunicaciones (TIC), agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de las informaciones. Las mismas han alcanzado gran auge a nivel mundial, convirtiéndose en la principal tarea para alcanzar el éxito en empresas, fábricas e instituciones, lo que trae consigo el aumento de la eficiencia, la eficacia, la productividad y otros elementos que resultan indispensables para el logro de sus metas.

Cuba ha sido víctima por más de 50 años de un criminal bloqueo que impide el completo desarrollo de la esfera social y económica. La informática no queda exenta de la política hostil implementada por los Estados Unidos hacia nuestro país, sino que es uno de los sectores más afectados, por lo que imposibilita avanzar a la par de muchos otros países en el campo del desarrollo de *software*. A pesar de este impedimento, el gobierno se ha trazado una política para llevar a cabo la informatización de la sociedad cubana, debido a que la industria del *software* está llamada a convertirse en una importante fuente de ingresos a la economía nacional. Uno de los sectores más beneficiados con esta colosal tarea es la educación.

La Universidad de las Ciencias Informáticas (UCI), es una de las instituciones educacionales donde se evidencia el uso y aplicación de las denominadas TIC. Desde su creación ha sido un pilar fundamental en este proceso de informatización que se está llevando a cabo a nivel nacional. La misma cuenta con un gran número de servicios que han sido informatizados. Algunos ejemplos de estos son el Entorno Virtual de Aprendizaje (EVA), que cuenta con los materiales disponibles y necesarios para cursar cada una de las asignaturas que se imparten en la universidad, el Directorio de personas, donde se pueden encontrar datos relevantes de la universidad referente a personas, teléfonos, aplicaciones y cumpleaños, Gestión Universitaria, que contiene información referida a las áreas de la UCI dentro de las que destacan pregrado, postgrado, residencia, extensión e investigación y la Biblioteca Digital, que cuenta con una amplia bibliografía la cual facilita el estudio de la comunidad universitaria.

La residencia estudiantil aloja gran cantidad de estudiantes de pregrado procedentes de todas las provincias del país, por lo que se hace necesario tener dentro de la misma un estricto control de las actividades que en ella se efectúan. Además, es necesario resaltar que esta es una de las áreas de la universidad que carece de servicios y procesos informatizados como por ejemplo Guardia obrera, Evaluación de desempeño, Planes de trabajo y Control al destino final de los recursos.

La UCI cuenta con dos Direcciones de Residencias Estudiantiles. La presente investigación está destinada a la Dirección de la Residencia Estudiantil 2, la cual tiene entre sus objetivos principales: fortalecer la labor educativa y política ideológica de manera que contribuya a la formación integral de los educandos y realizar la evaluación cualitativa mensual del estudiante como proceso de retroalimentación que deviene en el cumplimiento de los deberes como becario con repercusión en el proceso de integralidad.

Esta Dirección abarca disímiles procesos. Un primer proceso es “Evaluación del estudiante”, que incluye aspectos importantes tales como: cuartería, guardia estudiantil y cumplimiento de las normas, donde el Técnico B en conjunto con el Especialista General son los encargados de realizar la planificación y evaluación de la cuartería además de evaluar la guardia y el cumplimiento de las normas, esta evaluación se formula mensualmente. Un segundo proceso es “Censo, Capacidad y Matrícula”, donde el Especialista Superior es el encargado de manejar la información referente al estado y capacidades físicas de la residencia por facultad, tales como cantidad de edificios, capacidades disponibles, capacidades ocupadas y capacidades no ocupadas, además de generar informes que apoyen a la toma de decisiones.

Ambos procesos se realizan de forma manual, la información se recoge en formato duro y una vez entregada se almacena en hojas de cálculo o en modelos definidos por la Dirección de la Residencia Estudiantil 2, lo que provoca deficiencias como la pérdida de la información y el difícil acceso a consultarla por parte del personal autorizado, debido a que depende de una sola persona. Además, esto conlleva a que la misma no sea precisa, se encuentre desactualizada y la búsqueda se convierta en un trabajo lento y engorroso, a pesar de que la universidad cuenta con los medios y el personal capacitado para informatizar estos procesos.

A partir de la situación planteada surge como **problema a resolver**: La inexistencia de un sistema informático que permita la automatización de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI.

El **objeto de estudio** de la presente investigación es el sistema de gestión de la información en la Dirección de la Residencia Estudiantil 2, enmarcándose como **campo de acción** los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI.

El **objetivo general** consiste en automatizar los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI para apoyar la gestión de estos procesos.

Por lo que son definidos los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Caracterizar el sistema de gestión de la información en la Dirección de la Residencia Estudiantil 2 de la UCI.
- Definir las tecnologías y herramientas necesarias para el desarrollo de la propuesta de solución.
- Desarrollar un sistema informático para la gestión de la información de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI.
- Validar mediante pruebas de *software* el funcionamiento del sistema.

Tareas a cumplir:

- Entrevista a los directivos de la Dirección de la Residencia Estudiantil 2 para conocer el funcionamiento de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula”.
- Estudio de sistemas similares existentes dentro y fuera de la universidad.
- Proposición de una solución que contenga las funcionalidades necesarias con el objetivo de automatizar los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” pertenecientes a la Dirección de la Residencia Estudiantil 2.
- Selección de la metodología adecuada para guiar el proceso de desarrollo del sistema.
- Selección de las herramientas y tecnologías a utilizar para la implementación del sistema.
- Definición de los requerimientos que debe cumplir la propuesta de solución a desarrollar.
- Confección de los artefactos que propone la metodología de desarrollo de *software* seleccionada.

- Implementación de la propuesta de solución.
- Realización de las pruebas de *software* a la propuesta de solución.

El **resultado** de esta investigación será:

Un sistema informático que permita reducir el tiempo y el esfuerzo necesario para facilitar la gestión de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI. En el mismo se almacenará de forma centralizada la información para que pueda ser consultada por personal autorizado en el momento que se requiera. Este sistema permitirá además, realizar búsquedas rápidas y de forma sencilla, mostrar y brindar datos al personal especializado en el área de forma que les facilite tomar decisiones y mostrar reportes actualizados sobre las condiciones de la residencia y los datos de los estudiantes.

A partir del objetivo propuesto y las tareas definidas, se utilizaron los **métodos científicos** siguientes: (Hernández y Coello, 2012)

- A nivel teórico:
 - **Analítico-sintético:** Se empleó para crear la división mental de los componentes y sus múltiples relaciones de la propuesta de solución y luego establecer la unión entre las partes analizadas para así descubrir las características generales que debe cumplir la solución propuesta.
 - **Modelación:** Se utilizó para crear abstracciones con el objetivo de explicar el proceso real.
- A nivel empírico:
 - **Experimentación:** Para realizar las pruebas funcionales al sistema.
 - **Entrevista (No estructurada):** Para conocer los requerimientos que debe cumplir el sistema, el equipo de desarrollo se entrevistó con el personal del área relacionado con los procesos que tienen lugar en la investigación (Hernández y Coello, 2012).

Para una mejor comprensión de la investigación, cuyo diseño metodológico se describió anteriormente, se decidió definir una **estructura capitular** que aporte cierto grado de organización y facilite el estudio del documento. Los capítulos que lo conforman, son los siguientes:

Capítulo 1: Fundamentación Teórica

En este capítulo se presentan los elementos teóricos que fundamentan el sistema de gestión de la información asociada a los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI. Se incluye un estudio del estado del arte de las herramientas necesarias para la implementación del sistema, además de incluir una investigación sobre las tecnologías actuales para el desarrollo de aplicaciones web y la fundamentación de la selección de las más adecuadas para ser usadas en el desarrollo de la propuesta de solución al problema actual.

Capítulo 2: Exploración y Planificación

Este capítulo está dedicado a las dos primeras fases de la metodología de desarrollo Programación Extrema: Exploración y Planificación. En la Exploración los clientes definen lo que necesitan mediante la redacción de las Historias de usuario, el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto. Durante la Planificación el cliente establece la prioridad de cada Historia de usuario y los programadores realizan una estimación del esfuerzo para implementarlas. Además, se realiza el Plan de Iteraciones y el Plan de Entregas del producto.

Capítulo 3: Desarrollo y Pruebas

En este capítulo se presenta una descripción de la arquitectura de la propuesta de solución y se presenta un estudio de cómo se gestiona el contenido y la seguridad en el sistema. Se realiza además, una descripción de los patrones de diseño empleados y los estilos y estándares de codificación utilizados para la implementación de la solución propuesta, así como la descripción del modelo de datos del negocio para una mejor comprensión del mismo. También se muestran las tareas de ingeniería necesarias para llevar a cabo el proceso de desarrollo. Por otra parte, se define la estrategia de pruebas a seguir y se generan los casos de pruebas para realizar las mismas.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se precisan un conjunto de conceptos y fundamentos que conforman el marco teórico relacionado con el objeto de estudio definido en la investigación. Se destacan las principales características de los elementos asociados a la temática a investigar en el entorno nacional e internacional. Se exponen además las tecnologías, metodología y herramientas que sirven de apoyo para la búsqueda de una solución a la problemática planteada.

1.1 Conceptos generales relacionados con la investigación

Para un buen entendimiento de la investigación que se abarcará en este capítulo es necesario puntualizar conceptos claves que serán usados en la misma, los cuales se muestran a continuación.

Planificación: Plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado (Real Academia Española, 2015).

Control: Es la comprobación e inspección de las tareas asignadas a ciertas personas (Real Academia Española, 2015). No es más que el proceso de chequear las actividades para cerciorarse de que se desarrollan conforme se planearon y para corregir cualquier desviación evidente.

Evaluación: Se refiere a la acción y a la consecuencia de evaluar, permitiendo indicar, valorar, establecer, apreciar o calcular la importancia de un determinado asunto. Se trata de un acto donde debe emitirse un juicio en torno a un conjunto de información y debe tomarse una decisión de acuerdo a los resultados que presente un alumno. Consiste en una operación que se realiza dentro de la actividad educativa y que tiene como objetivo alcanzar el mejoramiento continuo de un grupo de alumnos. A través de ella se consigue la información exacta sobre los resultados alcanzados por ese alumno durante un período determinado, estableciendo comparaciones entre los objetivos planeados al inicio del período y los conseguidos por el alumno (Real Academia Española, 2015).

Cuartelería: Proceso mediante el cual un estudiante designado se hace cargo de la limpieza y organización de su paso de escalera en el edificio al que pertenece, así como del control del personal ajeno que visita el mismo.

Planificación de la cuartería: Mensualmente se realiza una planificación de la cuartería que consiste en asignarle a cada estudiante de la residencia el día que le corresponde realizar la actividad.

Evaluación del estudiante: Al estudiante se le otorga una evaluación general mensual que abarca varios factores: evaluación de la cuartería, de la guardia y el cumplimiento de las normas establecidas en la residencia.

Censo: Su uso más habitual está asociado al censo poblacional o censo de población, donde se contabilizan los habitantes de un pueblo o de un país y se recogen diversos datos con fines estadísticos. El censo permite delimitar una población estadística que refleja el número total de individuos de un territorio. A diferencia de otras herramientas de la estadística, en este caso no se trabaja con muestras, sino con la población total (Real Academia Española, 2015).

Capacidad: Se refiere a los recursos y amplitud que tiene una entidad o institución para albergar cierta cantidad de objetos o personas (Real Academia Española, 2015).

Matrícula: Inscripción de una persona en un registro o lista oficial. Conjunto de personas que están inscritas en un registro o lista oficial. Documento oficial que demuestra que una persona está inscrita en un registro o lista oficial (Real Academia Española, 2015).

1.1.1 Proceso “Evaluación del estudiante”

Este proceso consiste en emitirle una evaluación mensual general a cada estudiante, la cual está dada por tres parámetros fundamentales, el cumplimiento de la cuarterería, la guardia estudiantil y las normas establecidas por la residencia para una buena convivencia. Estas normas se refieren a limpieza y organización en el apartamento e indisciplinas cometidas por los estudiantes. Esta evaluación es de Muy Bien si el estudiante cumple correcta y debidamente con los parámetros antes mencionados, Bien o Regular si lo hace parcialmente y Mal si incumple con alguno de estos requisitos.

1.1.2 Proceso “Censo, capacidad y matrícula”

Este proceso consiste en almacenar y mantener actualizada la información referente a la arquitectura y estado de la residencia de cada facultad. Esta información se refiere al estado físico de la beca, la ubicación de cada estudiante con respecto a edificio, apartamento y las bajas, altas en cada semestre por facultad. Este proceso apoya a la toma de decisiones.

1.2 Análisis de soluciones existentes

El acelerado avance de las TIC en el mundo ha hecho al hombre unir las tecnologías a todas las esferas de la vida moderna para facilitar el desarrollo de las mismas, tal es el caso de la educación. Este es un sector que debido a su importancia se ha convertido en uno de los campos con mayor crecimiento en los últimos años a nivel nacional e internacional. Debido a que en el mundo existen diversos sistemas para organizar y gestionar la información que se maneja en las residencias universitarias, se hace necesario la realización de un estudio para determinar si existen soluciones que respondan al problema planteado anteriormente, este estudio se muestra a continuación.

1.2.1 Soluciones similares existentes a nivel internacional

- **Sistema Universitario de Residencias (SUR) de la Universidad de Santiago de Compostela, España (USC, 2015).**
 - Es el sitio web oficial de la USC desarrollado con un Sistema de Gestión de Contenidos que proporciona alojamiento al estudiantado y demás miembros de la comunidad universitaria, así como a otras personas que, aunque no pertenezcan a esta universidad participan en las actividades organizadas por la misma. El sitio se realizó en el año 2007 y a pesar de que se ha venido adaptando a tecnologías más modernas la diferencia se centra en que el sistema de albergue es diferente, dado a que no es por edificios sino por villas estudiantiles por lo que las características de las universidades no son las mismas. Además, la universidad cuenta con un sistema de limpieza y guardia realizado por trabajadores contratados para cumplir con estos servicios.
- **Sistema Aloja de la Universidad Carlos III de Madrid (UC3M), España (UC3M, 2015).**
 - Es un portal web para facilitar el alojamiento del estudiantado de la Universidad Carlos III de Madrid. Describe y muestra las características físicas y el estado de los apartamentos que están disponibles para ser rentados por los estudiantes. Fue desarrollado haciendo uso de código HTML para la creación de los documentos estándares, CSS para dar formato y JavaScript para realizar validaciones. Es importante mencionar que para visualizar correctamente el sitio es necesario ejecutar el lenguaje JavaScript. Lleva más de 10 años de explotación, aunque la universidad hizo un proceso de adaptación migrando el portal a tecnologías más modernas en el año 2013.

1.2.2 Soluciones similares existentes a nivel nacional

- **Sistema de Gestión Universitaria, Universidad de las Ciencias Informáticas UCI.**
 - Es una herramienta desarrollada en el año 2012 en la UCI empleando el lenguaje PHP, que contiene varios módulos, entre los que se encuentra Residencia, donde se gestionan algunos de los procesos de la misma, comenzando por el alojamiento del personal docente una vez que ingresan al centro hasta el trabajo educativo y evaluaciones de los estudiantes. Otro de los módulos es el de Alojamiento, que gestiona todo lo referente a zonificación y permutas en la residencia, tanto de profesores internos como de estudiantes. Este sistema no cuenta con las funcionalidades necesarias que permiten resolver el

problema que se plantea en la presente investigación, dado a que no tiene concebido dentro de sus funcionalidades la planificación de la cuartería ni la situación actual del estado físico de los edificios con los que cuenta cada facultad.

- **Sistema de gestión de información para la residencia universitaria de la Universidad de Holguín Oscar Lucero Moya** (Peña y Céspedes, 2010).
 - Es una herramienta multiplataforma creada en el 2010, desarrollada en PHP que hace uso del Sistema Gestor de Contenidos Joomla y permite a los trabajadores gestionar la gran cantidad de información referente a la matrícula y al inventario de los medios básicos de este centro sin necesidad de limitar el uso del sistema, pero no cuenta con módulos que posibiliten gestionar el censo, capacidad y matrícula y la evaluación de los estudiantes de la universidad.
- **Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Río Hermanos Saiz Montes de Oca** (Tellería, Vázquez y Alea, 2008).
 - Es una aplicación creada en el año 2008 que permite gestionar las evaluaciones de los estudiantes en cuanto a la cuartería, evaluaciones, sanciones, reconocimientos, y actividades. Además, recoge los datos y estado de la residencia estudiantil, pero fue realizada con Microsoft Access como herramienta de base de datos y la implementación se desarrolló en Visual Basic para Aplicaciones, ambas tecnologías son antiguas, por lo que este sistema no es óptimo para garantizar los resultados que se esperan con el sistema de gestión de la información en la Dirección de Residencia Estudiantil 2 de la UCI.

Tras haber estudiado las soluciones similares existentes dentro y fuera del país se llegó a la conclusión que ninguna resuelve la problemática planteada anteriormente, puesto que cada una de las residencias pertenecientes a las universidades presentan características diferentes. Además, las aplicaciones son diseñadas con herramientas propietarias o no contienen módulos que permitan gestionar la información de los procesos que se tratan en la presente investigación.

1.3 Selección del Sistema Gestor de Contenido

El desarrollo de aplicaciones web ha evolucionado enormemente en la última década, sobre todo desde el punto de vista del desarrollo de *software*. Para la realización del mismo se han creado

múltiples tecnologías, ejemplo de estas son los *Framework*, *Blogger* y Sistemas Gestores de Contenidos.

La evolución de las herramientas para la creación de páginas web, ha permitido que se construyan sitios web con mayor funcionalidad y presentación visual. Dentro de estas herramientas se encuentran los Sistemas de Gestión de Contenidos (en lo adelante CMS), que brindan una mejor calidad a las páginas, facilitan la producción, administración, actualización de manera simple y dinámica y el mantenimiento de los sitios web, además separan el contenido del diseño.

Según Juan Julian Merelo Guervós¹, “...un CMS se trata generalmente de un conjunto de herramientas, apoyado habitualmente por una base de datos, que consisten en una serie de programas en un servidor web y opcionalmente, una serie de programas cliente que permitan acceder fácilmente a esos programas en el servidor. Por lo tanto, un CMS tiene dos funciones principales: facilitar la creación de contenidos y la presentación de esos contenidos. Un CMS provee las herramientas necesarias para gestionar el ciclo de vida de los contenidos: creación, gestión, presentación, mantenimiento y actualización” (Merelo Guervós, 2005).

Algunas de las ventajas que presentan los CMS son:

- Ahorro de tiempo en la elaboración de páginas web.
- Mayor flexibilidad para añadir o suprimir apartados dentro de una página web.
- Posibilidad de crear, editar y actualizar contenidos web a diario.
- Sistema de navegación sencillo.
- Gestión independiente de contenido y diseño de la página web.

Según la compañía iMajestic Internet Marketing², los CMS más usados en la actualidad en el desarrollo de aplicaciones y portales web son: Drupal, Joomla y WordPress (Compañía iMajestic Internet Marketers, 2010).

¹ Juan Julián Merelo Guervós, licenciado en Ciencias Físicas por la Universidad de Granada en 1988 y doctorado por la misma universidad en 1994. Actualmente es Profesor Titular de la Universidad, asignado al Dpto. de Arquitectura y Tecnologías de Computadores e imparte docencia en la ETS de Ingeniería Informática.

² Compañía fundada en el 2001, desde su creación, la visión de iMajestic ha sido la de realizar estudios de las actuales tecnologías de la información digital con un enfoque para cambiar el panorama económico de la comercialización del Internet. <http://www.imajestic.com/>

La Tabla 1 muestra los CMS que existen para diferentes propósitos y la Tabla 2 las características principales de los CMS para sitios web:

Tabla 1 Tipos de CMS

| CMS | |
|------------|-----------------------------------|
| Sitios Web | Drupal, Joomla, Typo3. |
| Foros | phpBB, MyBB, Vanilla. |
| Blogs | WordPress, Serendipity, Lifetype. |
| Fotologs | Pixelpost. |
| Galerías | Gallery, coppermine. |
| Wikis | MediaWiki, TikiWiki, DokuWiki. |

Tabla 2 Características de los CMS para Sitios Web.

| CMS | Drupal | Joomla | Typo3 |
|-----------------|---|--|--|
| Características | Poderoso CMS muy conocido por la calidad de su código y por la seguridad que brinda, es estable y de actualización continua, configuración sencilla, instalación ágil, importante cantidad de módulos y temas, excepcional documentación y comunidad activa y muy amigable. | Instalación muy sencilla y con muchas extensiones y módulos, la documentación es exhaustiva y concisa, interface de la administración muy intuitiva y poderosa, opciones de personalización, una gran comunidad de usuarios. | Herramienta CMS con estructura multinivel, motor de búsquedas, gestión de autoría y publicación de contenidos, mecanismo de uso de plantillas para la maquetación de páginas, multilinguaje. |

CMS Drupal 7.21

Drupal está escrito en el lenguaje PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad del código, de las páginas generadas y un énfasis especial en la usabilidad y consistencia de todo el sistema. Su flexibilidad y adaptabilidad, así como la variedad de

módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitio web. Es un CMS de código abierto con licencia GNU/GPL³.

Este CMS presenta potencialidades que lo convierten en unos de los preferidos y más utilizados a escala mundial. Sobresale entre los demás debido a que:

- Cuenta con una amplia comunidad de usuarios que se mantienen en constante desarrollo de nuevas versiones y módulos para lograr el crecimiento del CMS.
- Está basado en la inclusión de módulos elaborados por la comunidad de desarrolladores que incluyen disímiles funcionalidades.
- Posee un flexible entorno de personalización. Tanto el contenido como la presentación son individualizados de acuerdo con las preferencias definidas por el usuario, gracias al sistema de temas con que cuenta lo que permite controlar o cambiar fácilmente el aspecto del sitio web.
- Su sistema de control de versiones permite seguir y auditar totalmente las sucesivas actualizaciones del contenido creado.
- Incorpora una capa de abstracción de base de datos implementada para MySQL, PostgreSQL, además, permite incorporar soporte para otras bases de datos.
- Es un sistema multiplataforma, funciona con Apache o IIS⁴ como servidor web y en sistemas operativos como: familia GNU/Linux, BSD, Solaris, Windows y Mac OS X.
- La administración y configuración del sistema se realiza enteramente con un navegador y no precisa de ningún *software* adicional.
- Está concebido en base a una arquitectura modular que le otorga flexibilidad gracias al manejo análogo de diferentes tipos de contenido y la utilización de taxonomías para clasificar los contenidos.

Como en el desarrollo se usará una metodología ágil y el uso de un CMS es sinónimo de agilidad, se escoge como CMS para la realización de la propuesta de solución Drupal en su versión 7.21, debido a que se necesita que el sistema posea seguridad contra pérdidas de información y ataques externos y

³ La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU *General Public License*, es una licencia creada por la Free Software Foundation (Fundación de *Software* Libre). Su propósito es declarar que el *software* cubierto por esta licencia es *software* libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

⁴ IIS por sus siglas en inglés Internet Information Services, en español Servicios de Información de Internet .

que funcione correctamente con tecnologías libres. Además, se cuenta con una vasta y actualizada bibliografía tanto en Internet como en la comunidad Drupaleros de la UCI.

1.4 Estudio de las metodologías y estándares para el desarrollo de software

A principios de los años 70 la industria de *software* se vio inmersa en una crisis a nivel mundial, a causa de malas prácticas a la hora de comenzar un proyecto. Los *software* que se realizaban no solucionaban las necesidades por las cuales habían sido creados, no eran óptimos y a su vez poco flexibles, lo que provocaba la insatisfacción de los usuarios. Esta situación trajo consigo que se comenzaran a emplear metodologías y estándares para una mayor organización por parte del equipo de desarrollo y garantizar una mayor calidad del producto que se estaba desarrollando. Surge así una disciplina, la Ingeniería de Software, “...que comprende todos los aspectos de la producción de *software*” (Sommerville, 2005) según plantea uno de sus fundadores, Ian Sommerville. Una de las buenas prácticas que promueve esta disciplina para obtener los resultados esperados, es el empleo de metodologías que guíen el proceso de desarrollo de *software*.

1.4.1 Programación Extrema (XP)

“XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico” (Canós, Letelier y Penadés, 2003).

El ciclo de vida de XP consiste en seis fases: (Canós, Letelier y Penadés, 2003)

1. Exploración
2. Planificación de la Entrega (*Release*)
3. Iteraciones
4. Producción
5. Mantenimiento
6. Muerte del Proyecto.

Para guiar y entender mejor la investigación, se decidió agrupar las seis fases en cuatro, sin violar lo que establece esta metodología, las cuales se muestran a continuación:

- A. Exploración
- B. Planificación
- C. Desarrollo
- D. Pruebas

En la fase de exploración se conforman las Historias de usuario donde el cliente define los requisitos necesarios que debe poseer el sistema, además de establecer aquellos que tengan mayor prioridad. Estas Historias de usuario son las bases para una vez terminada la aplicación se puedan realizar pruebas funcionales al sistema. En la fase de planificación XP establece que las iteraciones para las entregas deben ser cortas, es decir, entre 2 o 3 semanas aproximadamente y que se prioricen aquellos requisitos que posean prioridad alta, debido a que son más importantes para el negocio. En las fases finales, se realiza la implementación y pruebas del *software*. Estas son diseñadas por parejas, el código está dirigido por las pruebas y es modular, intentado refactorizar siempre que se pueda.

El proceso de desarrollo de *software* según esta metodología consiste en los siguientes pasos: (Canós, Letelier y Penadés, 2003)

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

XP define para el equipo de desarrollo los siguientes roles: (Beck y Andres, 2004)

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

A partir del análisis anterior se puede concluir que XP será la metodología que guiará la propuesta de solución a desarrollar, debido a que el cliente en todo momento formará parte del proceso, podrá

definir los requisitos funcionales que poseerá la propuesta de solución y establecerá la prioridad en ellos, para que los desarrolladores comiencen a trabajar con los que posean prioridad alta. El tiempo de las entregas será corto, se trabajará en parejas compartiendo la misma PC y las iteraciones tendrán poco tiempo entre ellas, como máximo tres semanas. Además, no se requiere de abundante documentación, debido a que la propuesta de solución no es de gran complejidad y será de fácil entendimiento para aquellos usuarios que no posean conocimientos de informática.

1.4.2 Lenguaje Unificado de Modelado (UML⁵)

UML 2.1: Es un lenguaje de modelado estándar para los planos del *software* que contiene diagramas, elementos y sus relaciones. El modelo proporciona una comprensión de un sistema. Nunca es suficiente un único modelo, más bien para comprender cualquier cosa a menudo se necesitan múltiples modelos conectados entre sí, excepto en los sistemas más triviales. Para sistemas con gran cantidad de *software* se requiere un lenguaje que cubra las diferentes vistas de la arquitectura de un sistema mientras evoluciona a través del ciclo de vida del desarrollo de *software* (Rumbaugh, Jacobson, y Booch, 2000).

1.5 Selección de las herramientas y lenguajes de desarrollo

El amplio desarrollo de las redes y los medios de comunicación a nivel mundial, ha traído consigo que se desarrolle Internet y con esto, que aparezcan nuevas aplicaciones para este espacio.

“Una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (navegador, explorador o visualizador) como el servidor (servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones” (Mora, 2002).

Muchas son las ventajas que posee desarrollar aplicaciones web, dentro de las que destacan: (Mora, 2002)

- El problema de gestionar el código en el cliente se reduce drásticamente, para introducir un cambio solo es necesario modificar la aplicación alojada en el servidor web.
- Se evitan problemas de inconsistencia en las actualizaciones, puesto que no existen clientes con distintas versiones de la aplicación.
- Son independientes a las plataformas donde son ejecutadas, solo dependen de un navegador web.

⁵ UML: Unified Modeling Language, en español Lenguaje Unificado de Modelado.

- Facilitan la centralización de los datos.
- Son baratas, sencillas de desarrollar y actualizar.

Por todo lo anteriormente planteado se decidió que las tecnologías que se emplearán para el desarrollo de la propuesta de solución serán de tipo web, debido a que se necesita una aplicación con datos centralizados a los que pueda acceder todo aquel personal que esté autorizado en cualquier momento, además son más fáciles y menos costosas para desarrollar y dar mantenimiento, no requieren de tecnologías costosas ni de plataformas para ser ejecutadas, solo dependen de un navegador web.

1.5.1 Lenguajes de programación

1.5.1.1 PHP

Es un acrónimo recursivo que significa PHP *Hypertext Pre-processor*, además es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML, es un lenguaje de scripting, el cual puede ser embebido dentro de páginas HTML. El objetivo del lenguaje es permitir a desarrolladores web escribir páginas generadas dinámicamente con rapidez (Achour et al., 2015). La elección del mismo está determinada por seleccionar a Drupal como CMS para el desarrollo, el cual se encuentra escrito en este lenguaje.

Algunas de sus características son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, debido a que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayudas libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

1.5.1.2 JavaScript

Es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico, es más conocido como el lenguaje de script para páginas web. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Eguiluz, 2015). La selección de JavaScript como el lenguaje de script del lado del cliente, está determinada por el hecho de usar la librería jQuery, la cual está escrita en el lenguaje citado.

1.5.1.3 HTML

Lenguaje de Etiquetado de Hipertexto (*Hypertext Markup Language*): Es un lenguaje comúnmente utilizado para la publicación de hipertexto en la web y desarrollado con la idea de que cualquier persona o tipo de dispositivo pueda acceder a la información en la web (W3C, 2015).

La última versión de este lenguaje es la conocida HTML5, la misma está pensada con una mayor integración con los lenguajes CSS y JavaScript. HTML5 ha revolucionado la web y no solo se ve como el presente, sino como el futuro, por las numerosas novedades que trae con respecto a la versión anterior.

La selección de HTML5 como el lenguaje para realizar el maquetado de la propuesta de solución está sustentada en el hecho de ser este un estándar establecido por la W3C⁶.

1.5.1.4 CSS

Hojas de Estilo en Cascada (*Cascading Style Sheets*): Es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, cómo se va a imprimir, o cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos (W3C, 2015).

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (Eguiluz, 2015).

⁶ W3C por sus siglas en inglés *World Wide Web Consortium*.

La última versión del lenguaje CSS es la versión 3 (CSS 3), la cual ha incorporado valiosas novedades, como: (Luca, 2015)

- Nuevas alternativas para dibujar bordes con el uso de opciones tales como color, imágenes y radio o redondeado.
- Novedades en el trabajo con fondos, con el uso de degradados y la posibilidad de incluir múltiples imágenes.
- Uso de sombras para texto (*text shadow*).
- Se incluye la posibilidad de aplicar sombra a los elementos (*box shadow*).
- Novedades en cuanto al uso del color y de la opacidad.
- Nuevas características para el trabajo con múltiples columnas.
- Incorporación de muchas novedades en lo que se refiere a flujo de texto dentro del sitio (*text overflow*).

La selección de CSS 3 como lenguaje para escribir el estilo visual de la propuesta de solución se hizo teniendo en cuenta que el mismo es un estándar determinado por la W3C y posee un amplio uso y difusión por parte de maquetadores, diseñadores y desarrolladores web.

1.5.2 Selección de la librería o *framework* JavaScript

jQuery en su versión 1.8.3: es un *framework* que contiene una serie de funciones y métodos de JavaScript y además es uno de los complementos principales para el desarrollo web. El mismo facilita la mejora de aplicaciones enriquecidas del lado del cliente, es compatible con todos los navegadores y con CSS3.

Algunas de las principales ventajas de jQuery son:

- Flexibilidad y rapidez para el desarrollo web.
- Tiene una excelente comunidad de soporte.
- Cuenta con *plugins*.
- Tiene una excelente integración con AJAX⁷.

Es una librería de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM⁸, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery, ofrece una serie de funcionalidades basadas en JavaScript

⁷ Por sus siglas en inglés **A**synchronous **J**avascript **A**nd **X**ML, en español JavaScript Asíncrono y XML.

⁸ DOM del inglés Document Object Model, en español Modelo de Objetos del Documento.

que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Sponsors, 2015).

Algunas de sus características son:

- Permite manipular fácilmente el DOM de una página web.
- Permite manejar de una manera sencilla los eventos de los elementos que forman una página web.
- Permite modificar, eliminar o acceder a los estilos visuales de una página web.
- Posee compatibilidad con navegadores como Mozilla Firefox, Internet Explorer, Safari, Opera y Google Chrome.

Tras realizar una búsqueda en la web se pudo determinar mediante estadísticas mostradas por el sitio BuiltWith, que la librería escrita en el lenguaje JavaScript denominada jQuery, posee un amplio uso en la Internet. Esto se puede observar en el Anexo 1, figura 16, el cual muestra el uso de tecnologías JavaScript y a su vez en el mismo anexo pero en la figura 17 se muestra la utilización de librerías JavaScript.

1.5.3 Selección del *framework* CSS

Un *framework* CSS es un conjunto de herramientas, hojas de estilos y buenas prácticas que permiten al diseñador web olvidarse de las tareas repetitivas para centrarse en los elementos únicos de cada diseño en los que puede aportar valor. Además, controlan la apariencia de los documentos electrónicos creados con HTML y XHTML, su utilización consiste en diseñar páginas web perfilando por separado el contenido y el formato.

CSS mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en dispositivos diferentes.

Algunas de las ventajas de los CSS son: (García Carrero, 2008)

- Separación del contenido y presentación.
- Flexibilidad.
- Unificación del diseño y de las páginas del sitio.
- Optimización de los tiempos de carga y de tráfico en el servidor.
- Precisión o elasticidad.

- Accesibilidad o estructuración.
- Limpieza de código fuente.
- Compatibilidad y continuidad.
- Estandarización frente a especificaciones propietarias.
- Permite la diferenciación de estilos para imprimir y visualizar en pantalla.

Twitter-Bootstrap 2.2.2 es un *framework* o conjunto de herramientas de *software* libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografías, formularios, botones, cuadros y menús de navegación basado en HTML y CSS, así como extensiones de JavaScript opcionales.

Además, es de código abierto y compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores.

Este *framework* es un marco de trabajo que facilita la labor del diseño web, el mismo es una tecnología que urge como unas de las alternativas más factibles según estadísticas de BuiltWith, esto se evidencia en el Anexo 1, figura 18, donde se muestra el uso de los *framework* de diseño.

1.5.4 Herramienta CASE⁹ de modelado con UML

Es la aplicación de tecnologías informáticas a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

Aumentan la productividad y logran una reducción en el costo de tiempo y de dinero. Al utilizar estas herramientas se puede abstraer al código fuente en un nivel donde la arquitectura y el diseño son más fáciles de entender y modificar (Universidad Politécnica de Valencia, 2013).

Existen varias herramientas CASE para el modelado visual mediante UML de sistemas de *software*, entre estas se encuentran Enterprise Architect, Rational Rose y Visual Paradigm.

1.5.4.1 Visual Paradigm for UML

Es una herramienta considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo de *software* que lo automatiza y acelera, permitiendo la captura de

⁹ CASE: Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora.

requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases (Visual Paradigm, 2015).

La versión a utilizar será la 8.0, lanzada el 16 de Agosto de 2010.

1.5.5 Servidor de Bases de Datos

Un sistema gestor de base de datos (SGBD) es el *software* que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado (Cabello, 2008).

En la revisión de la literatura se evidencia el uso de los SGBD para el desarrollo de aplicaciones de gestión de contenidos. Algunos ejemplos de SGBD son Oracle, PostgreSQL, MySQL, Microsoft SQL Server. De ellos se determina el uso del SGBD MySQL en su versión 5.0.15, puesto que se integra de manera nativa con el CMS Drupal, además, ofrece un conjunto de características y ventajas que seguidamente se mencionan: (Oracle, 2015)

- Es un sistema de gestión de bases de datos relacional, multihilos y multiusuario con más de 6 millones de instalaciones.
- MySQL AB desarrolla MySQL como *software* libre en un esquema de licenciamiento dual: se encuentra bajo la licencia GNU GPL para cualquier uso compatible, pero las empresas que quieran incorporarlo a productos privativos deben comprar una licencia que le permita este uso.
- Es la base de datos de código abierto de mayor aceptación mundial, debido a que permite una creación asequible y fiable de aplicaciones de bases de datos integradas basadas en web de alto rendimiento y fácilmente ampliables.
- Además de la facilidad de uso, el alto rendimiento y la fiabilidad de MySQL, se puede beneficiar de funciones avanzadas, herramientas de gestión y soporte técnico para desarrollar, desplegar y gestionar aplicaciones MySQL.
- Posee un bajo coste total de propiedad y tiene una gran capacidad de ampliación y rendimiento.
- Ofrece soporte de producción.

1.5.5.1 EMS SQL Manager for MySQL

SQL Manager for MySQL es una herramienta de alto rendimiento para la administración y el desarrollo de bases de datos MySQL. Funciona con cualquier versión de MySQL desde 4.1 a la más reciente y soporta todas las últimas características, incluyendo desencadenamiento MySQL, vistas, procedimientos almacenados y funciones, claves foráneas InnoDB y datos Unicode. SQL Manager para MySQL permite crear y editar objetos, diseños visuales, ejecutar scripts SQL, crear y editar datos de importación y exportación de bases de datos MySQL, gestionar los usuarios de MySQL y sus privilegios. Todas estas características permiten llevar a cabo una administración eficiente de la base de datos. Posee una interfaz de usuario amigable y con baja curva de aprendizaje (SQL Manager, 2015).

1.5.6 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en un ordenador, que se encarga de atender las peticiones realizadas por las estaciones clientes, entregando como resultado una página web o información de todo tipo dependiendo de los comandos solicitados. Es un programa que implementa el protocolo HTTP¹⁰.

Uno de los servidores web más utilizados a nivel mundial es Apache (BuiltWith Pty Ltd, 2015).

Apache en su versión 2.2.22: Es un *software* libre, servidor web HTTP de código abierto para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Los ficheros binarios y el código fuente de Apache¹¹ pueden ser usados y distribuidos libremente siempre que se les reconozca el trabajo a los autores (Catalinas, Román, y Rodríguez, 2007).

Según estadísticas analizadas en diferentes fuentes se pudo constatar que Apache es el servidor web más usado en Internet, esto se puede evidenciar en el Anexo 1 figura 19.

La facilidad de configuración, la correcta integración que brinda este servidor en un ambiente de desarrollo en sistemas operativos GNU/Linux, su amplia difusión y uso en la red de redes, es el motivo por el cual fue seleccionado para el desarrollo de la propuesta de solución. La versión a utilizar será la 2.2.22, liberada el 12 de julio de 2013.

¹⁰ Del inglés *HyperText Transfer Protocol*, en español Protocolo de transferencia de hipertexto.

¹¹ Página oficial: <http://httpd.apache.org/>.

1.5.7 Entorno de desarrollo integrado

PhpStorm en su versión 8.0.1: Es un poderoso editor del lenguaje PHP centrado en la productividad del desarrollador, de navegación rápida y comprobación de errores sobre la marcha.

Características:

- Editor de código inteligente: PhpStorm cuenta con un editor de código rico e inteligente que realmente entiende el código y su estructura, con soporte para PHP 5.3, 5.4, 5.5 y 5.6, abarcando así tanto los proyectos recientes como los antiguos. Proporciona una finalización de código inteligente, resaltado de sintaxis, configuración de formato de código extendido, comprobación de errores sobre la marcha, plegado de código y mucho más.
- Depurador visual: Ofrece numerosas opciones para depurar el código PHP pudiendo:
 - Inspeccionar las variables locales relevantes al contexto y definidas por el usuario, incluyendo arreglos y objetos complejos.
 - Configurar la depuración desde un servidor remoto.
 - Evaluar una expresión en tiempo de ejecución.
 - Depurar una página en varias sesiones simultáneamente.
 - Mantener una sesión de depuración y a la vez moverse entre páginas.
- Editor HTML / CSS / JavaScript: Dispone de una función de autocompletado de código para las etiquetas JavaScript, HTML y CSS.
- Detecta y sugiere auto-correcciones para problemas tales como:
 - Falta de los atributos requeridos.
 - Atributos no válidos o valores incorrectos.
 - Referencias erróneas o enlaces rotos hacia archivos.
- Entorno de desarrollo: Dispone de los sistemas de control de versiones más populares como Git, Subversion, Mercurial y Perforce.
- Realiza tareas de forma automática como añadir, eliminar y borrar archivos.

1.6 Conclusiones

Realizado el análisis sobre el estudio del estado del arte de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula”, se arrojan las siguientes conclusiones:

- No queda constancia de la existencia de los procesos antes referidos de forma automatizada en la Dirección de la Residencia Estudiantil 2 de la UCI.
- Analizando las funcionalidades y objetivos de esta Dirección, se percibe la necesidad de automatizar los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” para contribuir a elevar la calidad de la gestión de la información en esta área, teniendo en cuenta la selección de una metodología que permita la combinación de diferentes indicadores de evaluación de la calidad.
- Las herramientas y tecnologías seleccionadas para el desarrollo de la propuesta de solución son: como metodología de desarrollo Programación Extrema, como Sistema Gestor de Contenido Drupal 7.21, PhpStorm 8.0.1 como IDE de desarrollo, como SGBD MySQL 5.0.15, Apache 2.2.22 como servidor web y CSS 3 y HTML 5 para estilos y formato.

Capítulo 2: Exploración y Planificación

En este capítulo se presenta la descripción de la propuesta de solución que incluye la captura de requisitos con la elaboración de las Historias de usuario que plantea la metodología seleccionada. Se incluye además, los usuarios y permisos que deberá tener la aplicación y se presentan los conceptos fundamentales en un diagrama conceptual del negocio.

2.1 Descripción de la propuesta de solución

La propuesta de solución tiene como objetivo apoyar los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la UCI. Dentro del sistema los usuarios podrán desempeñar los siguientes roles: Administrador, Director General, Especialista Superior, Especialista General y Técnico B. El acceso se realizará empleando el método de autenticación de la universidad, insertando usuario y contraseña válidos en el dominio *uci.cu*.

El usuario con rol Administrador será el encargado del correcto funcionamiento y mantenimiento del sistema, además, es el único que puede crear roles y asignar permisos en la aplicación. Por otra parte, el usuario con rol Director General tendrá acceso total al sistema, contando con todos los permisos de gestión de contenidos en la aplicación. El usuario con rol Especialista Superior es el encargado de gestionar el módulo Facultad, perteneciente al proceso “Censo, capacidad y matrícula”, debido a que es el que maneja la información referente a este proceso, por lo que tendrá todos los permisos en este módulo y de solo lectura en el módulo Estudiante. El rol Especialista General es el encargado de visualizar y modificar en todo momento la información que gestione el rol Técnico B. Al acceder el usuario con rol Técnico B, se le otorgará permiso de escritura en el módulo Estudiante y podrá emitir una evaluación teniendo en cuenta determinados aspectos como el cumplimiento de la cuartelería, la guardia estudiantil y las normas establecidas en la residencia. Es el encargado también de planificar la cuartelería. Además, entre sus competencias está recoger los datos físicos del área a la que pertenezca. Todos los roles podrán realizar búsquedas específicas en el momento que se requiera y enviar por correo la información o convertirla en formato PDF.

El sistema permitirá a los trabajadores de la Dirección de la Residencia Estudiantil 2 llevar un control estricto, organizado y actualizado sobre la información que se maneja en el área. Al finalizar el desarrollo de la aplicación estarán disponibles los módulos Estudiante y Facultad. Esta propuesta podrá ser visualizada en cualquier computadora que posea un navegador web para acceder al servidor donde se aloje el sistema.

2.2 Usuarios del sistema

Los usuarios del sistema son aquellas personas que interactúan con la aplicación. Para ellos existen algunas restricciones específicas. A continuación se describen los usuarios del sistema con sus responsabilidades.

Tabla 3 Usuarios del sistema.

| Usuarios | Responsabilidades |
|-----------------------|---|
| Administrador | Es el encargado del correcto funcionamiento y mantenimiento del sistema, además es el único que puede crear roles y asignar permisos en la aplicación. |
| Director General | Tiene acceso a los dos módulos del sistema, Estudiante y Facultad. Posee todos los permisos. |
| Especialista Superior | Tiene acceso a los dos módulos del sistema, Estudiante y Facultad. Es el encargado de gestionar este último debido a que es el que debe mantener actualizada la información referente al estado físico de las becas para apoyar la toma de decisiones, por lo que cuenta con todos los permisos en este módulo y de solo lectura en el módulo Estudiante. |
| Especialista General | Solo podrá tener acceso al módulo Estudiante dado a que puede insertar, modificar o eliminar cualquier dato que se registre en el mismo. |
| Técnico B | Podrá tener acceso de escritura en el módulo Estudiante puesto a que es el encargado de planificar y evaluar la cuartelería, el cumplimiento de la guardia, de las normas establecidas y finalmente emitir una evaluación mensual general a cada estudiante. |

2.3 Modelo conceptual del negocio

Un diagrama conceptual del negocio no es más que un artefacto construido bajo las reglas de UML durante la concepción de un proyecto informático. Este modelo puede ser utilizado para capturar y expresar el entendimiento ganado sobre el negocio (Preciado, 2011).

El equipo de desarrollo consideró necesario generar este artefacto para un mejor entendimiento de la propuesta de solución a desarrollar, a partir de identificar los conceptos y objetos relacionados con los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula.”

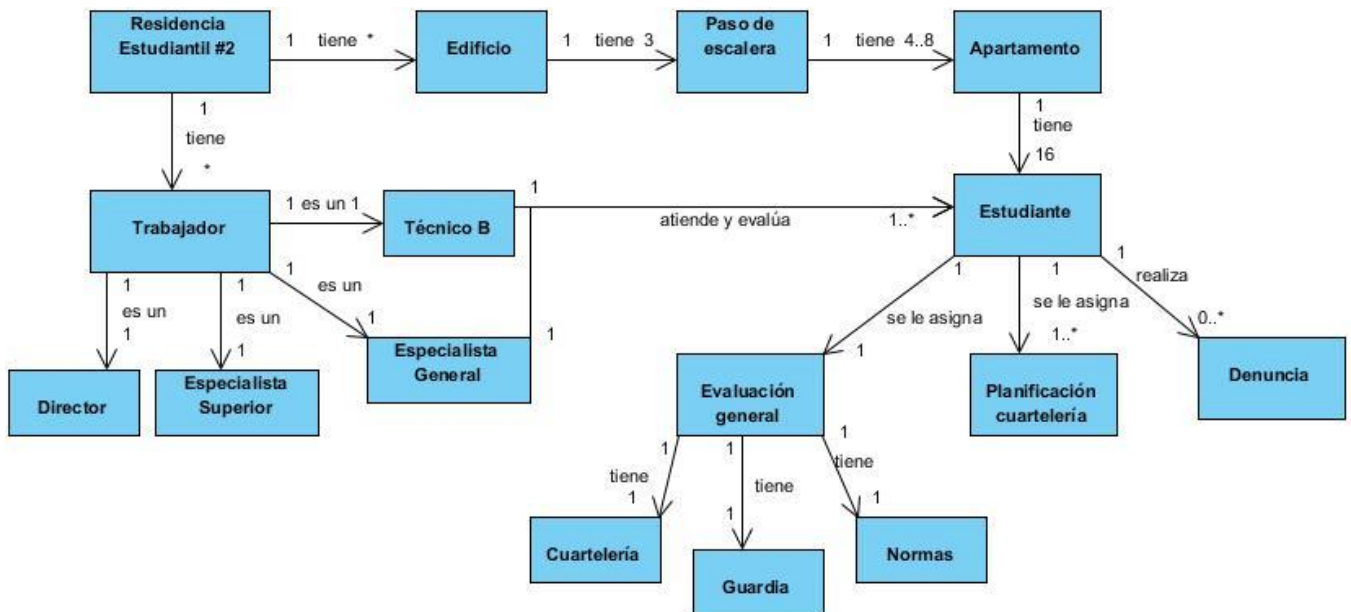


Figura 1 Modelo conceptual del negocio.

2.4 Funcionalidades del sistema

Una funcionalidad del sistema (FS) es una condición o capacidad que debe cumplir el mismo. Son especificaciones a implementar (Beck y Andres, 2004).

Las FS identificadas para llevar a cabo el desarrollo de la propuesta de solución son:

FS 1: Gestionar planificación de la cuartería: El sistema debe permitir visualizar, insertar, modificar y eliminar la planificación de la cuartería de un estudiante. De cada cuartería se registra la etapa, nombre y apellidos del estudiante, el correo, la fecha, el edificio y paso de escalera donde debe realizarla.

FS 2: Gestionar evaluación de cuartería: El sistema debe permitir visualizar, insertar, modificar y eliminar la evaluación de la cuartería del estudiante. De cada evaluación se registra nombre y apellidos del estudiante, la fecha, evaluación y observaciones en caso de que sea necesario.

FS 3: Gestionar evaluación de guardia estudiantil: El sistema debe permitir visualizar, insertar, modificar y eliminar la evaluación de la guardia estudiantil del estudiante. De cada evaluación se registra nombre y apellidos del estudiante, la fecha, evaluación y observaciones en caso de que sea necesario.

FS 4: Gestionar evaluación del cumplimiento de normas: El sistema debe permitir visualizar, insertar, modificar y eliminar la evaluación de las normas establecidas en la residencia para propiciar una buena

convivencia. De cada evaluación se registra nombre y apellidos del estudiante, la fecha, evaluación y observaciones en caso de que sea necesario.

FS 5: Gestionar evaluación mensual del estudiante: El sistema debe permitir visualizar, insertar, modificar y eliminar la evaluación mensual del estudiante. De cada evaluación se registra mes, nombre y apellidos del estudiante, el correo, la evaluación y observaciones en caso de que sea necesario.

FS 6: Enviar correo al estudiante informando fecha: El sistema debe enviar un correo al estudiante informando la planificación con respecto a día y lugar en que le corresponde realizar la cuartería.

FS 7: Enviar correo al estudiante informando evaluación: El sistema debe enviar un correo informando la evaluación mensual del estudiante.

FS 8: Filtrar evaluación para ver reincidentes: El sistema debe permitir buscar las evaluaciones de los estudiantes.

FS 9: Filtrar por edificios y por pasos de escaleras para ver cuartería: El sistema permitirá filtrar los datos por edificios y pasos de escaleras para ver los cuartereros del día seleccionado.

FS 10: Buscar datos: El sistema debe permitir filtrar la búsqueda por cuartería, cuartería por día, estudiante, estudiantes por edificio, listado de evaluaciones, evaluación mensual, y estado de permanencia.

FS 11: Filtrar por día para ver cuartería: El sistema debe permitir filtrar por día para ver los estudiantes que tienen cuartería el día seleccionado.

FS 12: Gestionar ubicación de estudiantes: El sistema debe permitir visualizar, insertar, modificar y eliminar la ubicación en cuanto a edificio, paso de escalera y apartamento del estudiante.

FS 13: Gestionar datos de edificios por facultad: El sistema debe permitir visualizar, insertar, modificar y eliminar los datos de los edificios por facultad. De cada edificio se registra el número del edificio, las capacidades disponibles, las ocupadas y las no ocupadas.

FS 14: Registrar permanencia del estudiante: El sistema debe permitir registrar el estado de permanencia de cada estudiante en la universidad. Para esto se debe almacenar el nombre y apellidos del estudiante, facultad a la que pertenece, sexo, fecha, grupo y el estado de permanencia.

FS 15: Registrar denuncia: El sistema debe permitir registrar una denuncia realizada por un estudiante.

FS 16: Exportar informes a PDF: El sistema debe permitir exportar en formato PDF los informes que se muestren en el sistema.

FS 17: Enviar informes: El sistema debe permitir enviar por correo electrónico la información que se gestiona en el sistema.

2.5 Características del sistema

Las características del sistema (CS) son condiciones o capacidades que debe poseer la aplicación para satisfacer las necesidades del cliente. Complementan los aspectos fundamentales del mismo, describiendo un comportamiento funcional y visible por el usuario (Beck y Andres, 2004).

La solución propuesta debe tener las siguientes características:

CS 1- Portabilidad: El sistema debe ser multiplataforma.

CS 2- Seguridad: El sistema debe garantizar que el acceso a la información se realice de acuerdo al rol que desempeñan los usuarios.

CS 3- *Software*: El sistema podrá ser visualizado utilizando los navegadores Firefox 19 o superior y Google Chrome 28 o superior.

2.6 Exploración

En esta fase el cliente describe con sus propias palabras las funcionalidades que debe tener el sistema, especificando cuales son las más importantes para la primera entrega del producto. Después del cliente entregar la propuesta, el equipo de desarrollo analiza y define las dificultades técnicas que presentan cada una de estas funcionalidades, así como su costo. Durante esta fase los programadores se familiarizan con las herramientas, tecnologías y prácticas que serán usadas en el desarrollo del proyecto y se exploran todas las posibilidades que puede tener la arquitectura del sistema (Calabria y Píriz, 2003).

Básicamente se realizan tres operaciones fundamentales durante la exploración: escribir las Historias de usuario, estimarlas y dividir en unas más pequeñas o agrupar en otras más generales aquellas que se considere necesario. Las estimaciones en esta fase son primarias y podrían variar cuando se analice en detalle cada iteración, logrando como resultado una visión general del sistema y un plazo total estimado. La duración de esta fase debe ser de pocas semanas a pocos meses, todo depende del tamaño del proyecto.

2.6.1 Historia de usuario (HU)

La metodología XP utiliza para representar los requisitos del sistema un artefacto denominado Historias de usuario, en lo adelante HU. Las mismas no son más que tarjetas de papel en las cuales el

cliente describe las características del sistema. En cualquier momento del desarrollo pueden ser modificadas o reemplazadas por otras más generales o específicas, o incluso añadir nuevas. Cada una es lo suficientemente entendible y está delimitada como para que los desarrolladores puedan implementarlas en unas semanas (Jeffries, Anderson, y Hendrickson, 2001).

No lograr determinar todas las HU desde un inicio no es un problema, puesto que al inicio de cada iteración se registran los cambios en estas y a partir de ahí se planifica la próxima iteración. Las HU se descomponen en tareas de ingeniería con una duración de uno a tres días, las cuales serán asignadas a los programadores para implementarlas durante una iteración (Calabria y Píriz, 2003).

2.6.2 Historias de usuario de la propuesta de solución

La metodología XP no define una plantilla específica para representar las HU, por lo que se utilizó la plantilla definida por el grupo Calidad UCI¹², que contendrá el número, el nombre del requisito, el programador encargado, la iteración asignada, el tiempo estimado y el real, el riesgo en desarrollo, una breve descripción y un apartado para agregar alguna observación si es necesario.

Fueron pactadas con el cliente un total de 20 HU, en este epígrafe se muestran las principales de acuerdo a la prioridad asignada por el cliente, las restantes se encuentran en el Anexo 2 de la investigación.

Tabla 4 HU5-Gestionar evaluación mensual del estudiante.

| | | | |
|---|--|---|--|
| Número:5 | | Nombre del requisito: Gestionar evaluación mensual del estudiante. | |
| Programador: Michelle Pérez Acosta. | | Iteración Asignada: 1 | |
| Prioridad: Alta | | Tiempo Estimado: 2 días | |
| Riesgo en Desarrollo: Alto | | Tiempo Real: 1 semana | |
| Descripción: El sistema permitirá visualizar la evaluación mensual del estudiante, insertarla, modificarla y eliminarla. | | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | | | |

Tabla 5 HU10-Buscar datos.

| | | | |
|--|--|--|--|
| Número:10 | | Nombre del requisito: Buscar datos. | |
| Programador: Claudia Milagros de León Martínez. | | Iteración Asignada: 2 | |
| Prioridad: Alta | | Tiempo Estimado: 4 días | |
| Riesgo en Desarrollo: Alto | | Tiempo Real: 1 semana | |

¹² Calidad UCI: Se encargan del aseguramiento de la calidad de los procesos y productos en la Entidad Desarrolladora.

| |
|---|
| Descripción: El sistema permitirá filtrar la búsqueda de una determinada información. |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles definidos y el sistema permitirá seleccionar el criterio de búsqueda deseado. |

Tabla 6 HU12-Gestionar ubicación del estudiante.

| | |
|---|--|
| Número: 12 | Nombre del requisito: Gestionar ubicación del estudiante. |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 3 |
| Prioridad: Alta | Tiempo Estimado: 5 días |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá visualizar la ubicación del estudiante, insertarla, modificarla y eliminarla. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B. | |

Tabla 7 HU13-Gestionar datos de edificios por facultad.

| | |
|---|---|
| Número: 13 | Nombre del requisito: Gestionar datos de edificios por facultad. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 3 |
| Prioridad: Alta | Tiempo Estimado: 5 días |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá visualizar los datos de edificios por facultad, insertarlos, modificarlos y eliminarlos. | |
| Observaciones: El usuario deberá ser autenticado en el sistema con alguno de los roles Director General o Especialista Superior. | |

Tabla 8 HU18-Portabilidad.

| | |
|---|--|
| Número: 18 | Nombre del requisito: Portabilidad. |
| Programador: Michelle Pérez Acosta, Claudia Milagros de León Martínez. | Iteración Asignada: - |
| Prioridad: - | Tiempo Estimado: - |
| Riesgo en Desarrollo: Alto | Tiempo Real: - |
| Descripción: El sistema debe ser multiplataforma. | |
| Observaciones: - | |

Tabla 9 HU19-Seguridad.

| | |
|---|---|
| Número: 19 | Nombre del requisito: Seguridad. |
| Programador: Michelle Pérez Acosta, Claudia Milagros de León Martínez. | Iteración Asignada: - |
| Prioridad: - | Tiempo Estimado: - |
| Riesgo en Desarrollo: Alto | Tiempo Real: - |
| Descripción: El sistema debe garantizar que el acceso a la información se realice de acuerdo | |

| |
|-------------------------------------|
| al rol que desempeñan los usuarios. |
| Observaciones: - |

Tabla 10 HU20-Software.

| | |
|--|--|
| | |
| Número: 20 | Nombre del requisito: Software. |
| Programador: Michelle Pérez Acosta, Claudia Milagros de León Martínez. | Iteración Asignada: - |
| Prioridad: - | Tiempo Estimado: - |
| Riesgo en Desarrollo: Alto | Tiempo Real: - |
| Descripción: Los navegadores a utilizar podrán ser Firefox 19 o superior y Google Chrome 28 o superior. | |
| Observaciones: - | |

2.6.3 Prototipos de Interfaz de Usuario

El diseño de la interfaz de usuario se puede definir como el conjunto de pasos que seguirá el usuario durante todo el tiempo que interactúe con la aplicación, detallando lo que verá en cada momento y las acciones que realizará, así como las respuestas que el sistema dará.

A continuación se muestran algunos de los prototipos de interfaz más importantes del sistema:

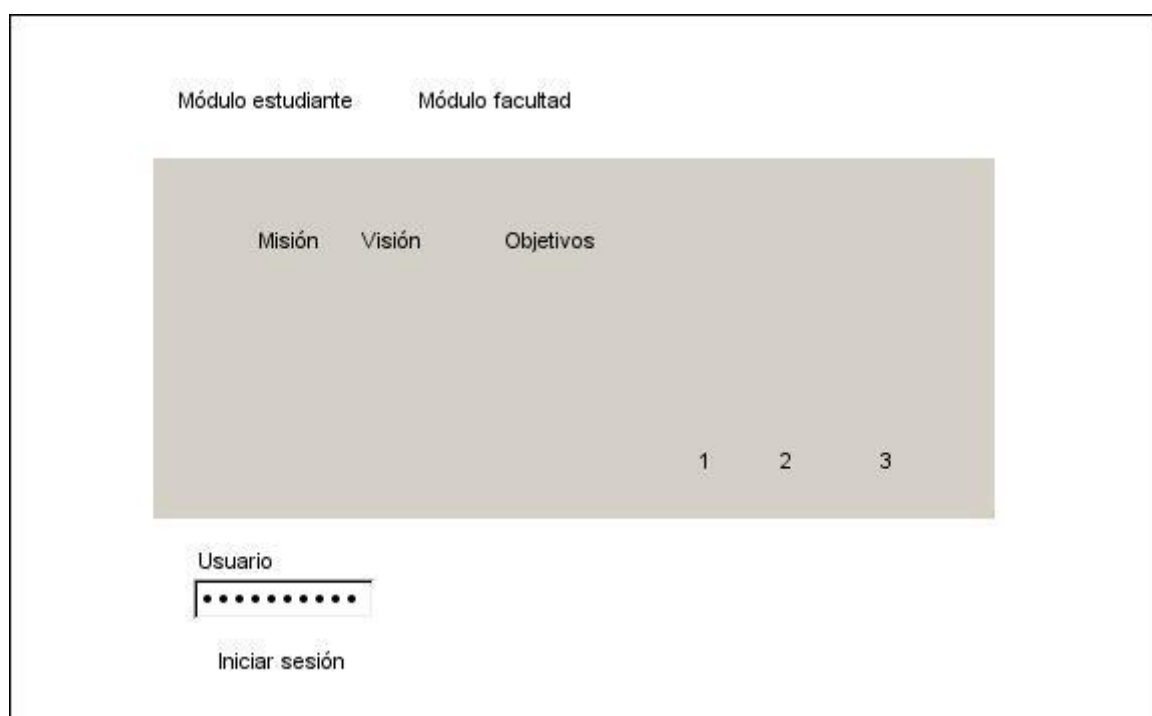


Figura 2 PI_Autenticar_usuario.

Adicionar cuarterería

Descripción

Estudiante

Correo

Fecha

Edificio

Paso de escalera

Evaluación

Observaciones

Guardar

Detailed description: This is a web form titled 'Adicionar cuarterería'. It contains several input fields: 'Descripción', 'Estudiante', and 'Correo' are text boxes. 'Fecha' is a date picker. 'Edificio' and 'Paso de escalera' are dropdown menus. 'Evaluación' is a dropdown menu. There is a large text area for 'Observaciones'. A 'Guardar' button is at the bottom.

Figura 3 PI_Gestionar_planificación_cuarterería.

Evaluación mensual

Descripción

Estudiante

Correo

Mes

Evaluación

Observaciones

Guardar

Detailed description: This is a web form titled 'Evaluación mensual'. It contains several input fields: 'Descripción', 'Estudiante', and 'Correo' are text boxes. 'Mes' and 'Evaluación' are dropdown menus. There is a large text area for 'Observaciones'. A 'Guardar' button is at the bottom.

Figura 4 PI_Gestionar_evaluación_mensual_estudiante.

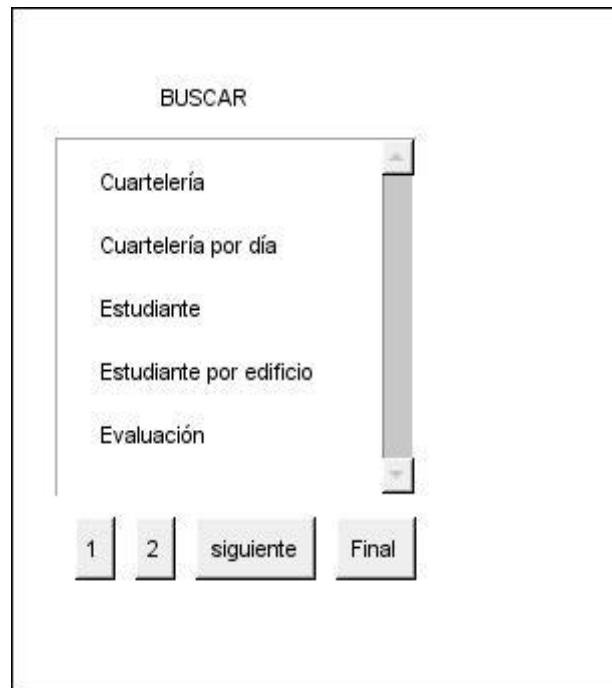


Figura 5 PI_Buscar_datos.

2.7 Planificación

XP plantea la planificación como un constante diálogo entre el cliente y los desarrolladores donde los primeros definen el alcance del proyecto, la prioridad de las HU, la composición de las versiones y la fecha de entrega de las mismas. Una vez determinados los requisitos antes mencionados los desarrolladores estiman el tiempo necesario para implementar cada HU, el cual debe ser de entre una y tres semanas. En caso de extenderse más de tres semanas la misma debe ser modificada o dividida en otras más pequeñas. Una duración de menos de una semana indica que es demasiado sencilla y deberá ser agrupada (Joskowicz, 2008).

“Toda técnica de planificación de software debe tratar de crear visibilidad, de tal manera que todo individuo involucrado en el proyecto pueda realmente ver cuán largo es el proyecto. Esto significa que se necesitan metas claras, metas que no sean dudosas y que claramente representen progreso. Las metas deben además, ser cosas que toda persona involucrada con el proyecto, incluyendo el cliente, pueda entender y aprender a confiar en estas” (Beck, 2001).

En esta fase se generan artefactos de suma importancia para el desarrollo del proyecto, como el Plan de Entregas y el Plan de Iteraciones, los cuales se muestran a continuación.

2.7.1 Plan de Entregas

El Plan de Entregas deber ser lo más real posible porque constituye un documento oficial por el cual el cliente le exige a los desarrolladores la entrega de las distintas versiones del producto. Para dar solución al problema planteado, el equipo de desarrollo a partir de una reunión con el cliente, teniendo en cuenta las prioridades planteadas por el mismo y las estimaciones realizadas por los desarrolladores, definió el siguiente Plan de Entregas: (Calabria y Píriz, 2003)

Tabla 11 Plan de Entregas.

| Entregable | 1ra entrega (3 ^{ra} semana de Marzo) | 2da entrega (3 ^{ra} semana de Abril) | 3ra entrega (3 ^{ra} semana de Mayo) |
|---|---|---|--|
| Sistema informático de apoyo a los procesos Evaluación del estudiante y Censo, capacidad y matrícula en la Dirección de Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas | Versión 0.1 | Versión 0.2 | Versión 1.0 |

Tabla 12 Funcionalidades disponibles por entregas del producto.

| Historia de usuario | 1ra entrega | 2da entrega | 3ra entrega |
|---|-------------|-------------|-------------|
| HU1. Gestionar planificación de la cuartería. | X | X | X |
| HU2. Gestionar evaluación de cuartería. | X | X | X |
| HU3. Gestionar evaluación de guardia estudiantil. | X | X | X |
| HU4. Gestionar evaluación del cumplimiento de normas. | X | X | X |
| HU5. Gestionar evaluación mensual del estudiante. | X | X | X |
| HU6. Enviar correo al estudiante informando fecha. | X | X | X |
| HU7. Enviar correo al estudiante informando evaluación. | | X | X |
| HU8. Filtrar evaluación para ver reincidentes. | | X | X |
| HU9. Filtrar por edificios y por pasos de escaleras para ver cuartería. | | X | X |

| | | | |
|--|--|---|---|
| HU10. Buscar datos. | | X | X |
| HU11. Filtrar por día para ver cuarterería. | | X | X |
| HU12. Gestionar ubicación de estudiantes. | | | X |
| HU13. Gestionar datos de edificios por facultad. | | | X |
| HU14. Registrar permanencia del estudiante. | | | X |
| HU15. Registrar denuncia. | | | X |
| HU16. Exportar informes a PDF. | | | X |
| HU17. Enviar informes. | | | X |

2.7.2 Plan de Iteraciones

Todo proyecto que se guía por la metodología XP, se planifica en iteraciones de aproximadamente tres semanas de duración. Para cada iteración se define un conjunto de HU que se van a implementar (Calabria y Píriz, 2003).

Es necesario aclarar que aunque en la presente investigación cada iteración concluye con un entregable; en XP una iteración no necesariamente termina con una entrega del producto, por lo que una versión puede implicar varias iteraciones del proceso de desarrollo, permitiendo al equipo de proyecto ubicar las Historias de usuario previstas para esta versión en tantas iteraciones como estime conveniente.

No se debe intentar implementar cualquier funcionalidad que no esté planificada en la iteración en curso. Habrá suficiente tiempo para añadir la funcionalidad extra cuando sea realmente importante según el Plan de Entregas.

Tabla 13 Planificación de las iteraciones.

| Iteraciones | Orden de las HU a implementar | Cantidad de tiempo de trabajo |
|--------------------|---|-------------------------------|
| Iteración 1 | HU1. Gestionar planificación de la cuarterería. HU2. Gestionar evaluación de cuarterería. HU3. Gestionar evaluación de guardia estudiantil. | 3 semanas |

| | | |
|--------------------|---|-----------|
| | <p>HU4. Gestionar evaluación del cumplimiento de normas.</p> <p>HU5. Gestionar evaluación mensual del estudiante.</p> <p>HU6. Enviar correo al estudiante informando fecha.</p> | |
| Iteración 2 | <p>HU7. Enviar correo al estudiante informando evaluación.</p> <p>HU8. Filtrar por evaluación para ver reincidentes.</p> <p>HU9. Filtrar por edificios y por pasos de escaleras para ver cuartelería.</p> <p>HU10. Buscar datos.</p> <p>HU11. Filtrar por día para ver cuartelería.</p> | 3 semanas |
| Iteración 3 | <p>HU12. Gestionar ubicación de estudiantes.</p> <p>HU13. Gestionar datos de edificios por facultad.</p> <p>HU14. Registrar permanencia del estudiante.</p> <p>HU15. Registrar denuncia.</p> <p>HU16. Exportar informes a PDF.</p> <p>HU17. Enviar informes.</p> | 3 semanas |

2.8 Conclusiones

Teniendo en cuenta la propuesta de solución planteada en el presente capítulo, se arribó a las siguientes conclusiones:

- La realización del Modelo conceptual le permitió al equipo de desarrollo una mejor comprensión del negocio, debido a que se manejan los conceptos fundamentales que involucran los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2.
- De la propuesta de solución planteada se detectaron un total de 17 funcionalidades del sistema y 4 características no funcionales con las que debe contar el mismo para satisfacer las necesidades del cliente.

- Luego de culminar la fase de planificación definida por la metodología seleccionada, quedó pactado con el cliente que se realizarán tres iteraciones y el tiempo máximo entre las mismas será de tres semanas.

Capítulo 3 Desarrollo y Pruebas

El presente capítulo contempla los elementos relacionados con la implementación del sistema, reflejándose en este sentido las Tareas de ingeniería y la forma en que se gestiona los tipos de contenidos. Además, se incluyen las pruebas de aceptación, las pruebas unitarias y el análisis de los resultados de estas pruebas que se realizan para validar el correcto desempeño del sistema.

3.1 Descripción de la arquitectura

El CMS Drupal está desarrollado en el lenguaje PHP, posee una arquitectura modular y sigue un modelo de programación estructurada haciendo uso de un sistema de bases de datos relacional. Gran parte de su éxito en cuanto a flexibilidad y facilidad en la creación de sitios web es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos: Nodos (*Nodes*), Módulos (*Modules*), Bloques y Menús (*Blocks & Menus*), Permisos de usuario (*User Permissions*) y Plantillas (*Templates*), esto se muestra en la siguiente figura: (Drupal, 2015)

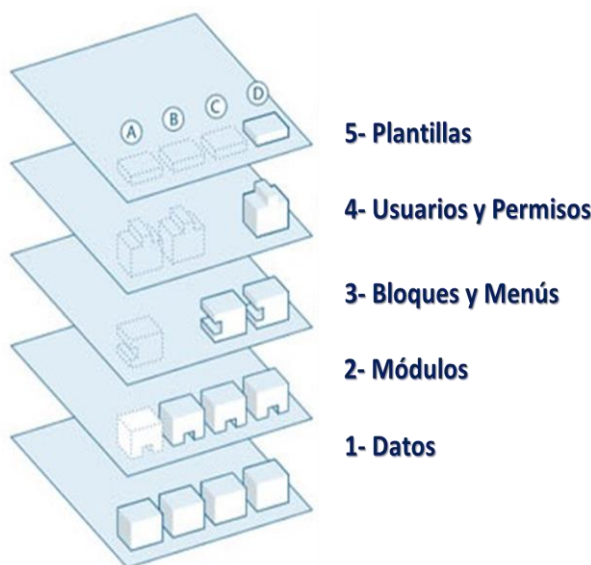


Figura 6 Arquitectura de Drupal.

Los Nodos son los elementos básicos en que Drupal almacena la información y los contenidos. Así, a medida que el sitio web crece, lo va haciendo el número de nodos, los cuales van formando un depósito

de nodos cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este depósito de nodos.

Los Módulos son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. Son como *plugins*¹³ que se instalan en el sitio web proporcionándole nuevas funcionalidades. Estos se dividen en tres grandes grupos: obligatorios, activados y opcionales. Los obligatorios no pueden desactivarse, se encuentran por defecto en la instalación. Algunos ejemplos de estos son Bloque, Campo, Filtro, Imagen, Nodo y otros. Los activados vienen también junto con el CMS, pero estos si pueden deshabilitarse en caso de no ser necesarios. Entre estos se encuentran Foro y Encuesta. Los módulos opcionales son aquellos que se instalan para cumplir funciones específicas. Ejemplos de módulos opcionales son Validación de campo y Traducción de contenido.

La siguiente capa de Drupal la constituyen los Bloques y Menús. Estos permiten estructurar y organizar los contenidos en la página web. Son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

Otra capa importante en Drupal es Control de usuarios y Permisos. Actualmente, la mayor parte de sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permite especificar qué tareas puede realizar y a qué contenidos puede acceder cada tipo de usuario. Es decir, las operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación) se encuentran limitadas por la capa de Control de usuarios y Permisos de Drupal.

La última capa es Plantillas y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño o apariencia sin necesidad de modificar los contenidos (Drupal, 2015).

Debido a la utilización de Drupal, la propuesta de solución posee también una arquitectura modular. Además de los módulos obligatorios y activados disponibles por defecto en la instalación del CMS, fue necesaria la instalación manual de otros módulos para garantizar el cumplimiento de los requerimientos descritos por el cliente, a continuación se muestra una lista de estos.

¹³ Un plugin es un módulo de *hardware* o *software* que añade una característica o un servicio específico a los sistemas.

Tabla 14 Módulos opcionales de la propuesta de solución.

| Módulo | Descripción |
|-------------------------------|---|
| Autenticación UCI | Para conectar la propuesta de solución con el sistema de autenticación de la universidad. |
| Permisos por roles | Para que la información solo se muestre al personal autorizado. |
| Traducción de contenido | Para traducir el contenido del inglés al español. |
| Validación de campo | Para validar cada uno de los campos. |
| Soporte de autenticación SMTP | Para enviar datos por correo a través del servidor SMTP de la universidad. |
| Formato PDF | Para convertir los informes al formato PDF. |

3.2 Gestión de tipos de contenidos en Drupal

Drupal permite crear contenidos de diferentes tipos, en función de los módulos instalados y activados. Es importante distinguir entre un tipo de contenido y el contenido en sí. Un tipo de contenido define una especie de plantilla a partir de la cual se crearán los contenidos y son conocidos como entidades a las que se pueden añadir campos adicionales. Los contenidos son las páginas finales que se mostrarán en el sitio, los cuales se conocen como nodos. Cada nodo se genera a partir de solo un tipo de contenido determinado. En el núcleo de Drupal se encuentran los siguientes tipos de contenidos: (Gil, 2012)

Tabla 15 Tipos de contenidos del Drupal.

| Tipo de contenido | Descripción |
|-------------------------------|--|
| Página básica (<i>page</i>) | Tipo de nodo básico que se utiliza para generar contenidos estáticos, como pueden ser las páginas de presentación y de servicios de una empresa. |
| Artículo (<i>article</i>) | Es un tipo de contenido pensado para la publicación de contenidos periódicos, como pueden ser noticias, novedades y artículos. |

| | |
|------------------------------------|--|
| Tema del foro (<i>forum</i>) | Permite la creación de nuevos temas de discusión en los foros. |
| Entrada de blog (<i>blogger</i>) | Permite generar entradas individuales en un blog o bitácora. |
| Página de libro (<i>book</i>) | Una página de libro es una página de contenido organizada en una colección de entradas relacionadas entre sí que constituyen un libro. |
| Encuesta (<i>poll</i>) | Es una pregunta con un grupo de respuestas posibles. Una vez creada proporciona automáticamente un sencillo contador del número de votos que ha recibido cada respuesta. |

Drupal permite también crear nuevos tipos de contenidos adaptados al sitio que se está construyendo. En la presente investigación, por cada entidad del Modelo lógico de datos¹⁴ fue creado un tipo de contenido específico. Estos se muestran a continuación.

Tabla 16 Tipos de contenidos del Sistema de Dirección de la Residencia Estudiantil 2.

| Tipo de contenido | Descripción |
|--------------------|---|
| Cuartelería | Contiene los datos necesarios para llevar a cabo la planificación de la cuartería, como son etapa, nombre y apellidos del estudiante, correo, fecha, edificio y paso de escalera. |
| Denuncias | Comprende todo lo referente a las denuncias, sus datos son, facultad, tipo de denuncia, nombre y apellidos del estudiante que realiza la denuncia, fecha, estado y causas. |
| Estudiante | Al registrar un estudiante se almacenan datos como nombre y apellidos, correo, grupo, edificio, paso de escalera y apartamento. |
| Evaluación mensual | Por estudiante se realiza una evaluación mensual la cual contiene mes, estudiante, correo, evaluación y observaciones. |
| Facultad | De cada facultad se registra información referente a las |

¹⁴ Se encuentra en el epígrafe 3.5

| | |
|------------------------|---|
| | capacidades disponibles, ocupadas y no ocupadas, así como la cantidad de edificios que posee la misma. |
| Gestión de permanencia | Almacena la información que involucra la estancia de un estudiante en la universidad, la misma contiene datos como nombre y apellidos del estudiante, facultad a la que pertenece, sexo, fecha, grupo y el estado de permanencia donde este puede ser alta, baja o traslado. |
| Guardia | Permite evaluar la guardia, donde se almacenan datos como nombre y apellidos del estudiante, fecha, evaluación y observaciones. |
| Normas | Permite evaluar las normas establecidas en la residencia donde se guardan datos como nombre y apellidos del estudiante, fecha, evaluación de parámetros como limpieza, organización e indisciplinas, además, de una evaluación general con respecto a los tres parámetros antes mencionados y las observaciones correspondientes. |

3.3 Gestión de la seguridad

Un sistema informático seguro que garantice la disponibilidad, integridad, confidencialidad y buen uso de la información es algo de lo que toda empresa debe disponer si no quiere sufrir daños irreparables en sus sistemas (Lorenzo, 2014).

La propuesta de solución posee un control de acceso basado en roles. Los roles no son más que un conjunto de reglas que se aplican de forma automática a todos los usuarios que lo tengan asignado, de esta forma se gestiona lo que puede hacer cada usuario. Cada rol podrá tener asignado un número de permisos disponibles en el sitio web. Solo el usuario Administrador tendrá todo el control sobre los roles y permisos (Gil, 2012).

Además de los roles por defecto, es posible crear nuevos roles para diferenciar las tareas que pueden realizar los usuarios.

El sistema desarrollado cuenta con 4 roles adicionales, los cuales son:

- Director

- Especialista Superior
- Especialista General
- Técnico B

Además, el sistema permite realizar copias de seguridad fuera del servidor donde se encuentra instalado para evitar pérdidas de la información. Cada módulo utilizado cuenta con su propia seguridad, al poseer funciones que evitan ataques de *hackers*, inyecciones SQL¹⁵ e inyecciones XSS¹⁶, todas estas características fueron adquiridas debido a la utilización del CMS Drupal.

3.4 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación.

Aunque la aplicación a desarrollar sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos y operaciones entre sistemas. En lugar de reescribir el código, se puede solucionar problemas utilizando algún patrón, puesto que son soluciones probadas y documentadas por multitud de programadores.

Nicolás Tedeschi¹⁷ define patrón de diseño como *“el esqueleto de las soluciones a problemas comunes en el desarrollo de software”*.

Para el desarrollo de la propuesta de solución que se describe en la investigación, solo fueron usados los implementados por el CMS Drupal. Estos serán descritos a continuación.

¹⁵ Inyección SQL: Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.

¹⁶ Inyección XSS: Permite a intrusos inyectar código JavaScript en páginas web vistas por el usuario.

¹⁷ **Nicolás Tedeschi**, Analista de Sistemas de la Facultad de Ingeniería de Uruguay y se encuentra desarrollando tecnología .net y SQL Server. Desde 1995 trabaja con tecnologías Microsoft de tres capas (HTML-ASP, COM+ y SQL Server). Además, es Desarrollador 5 Estrellas de Microsoft MSDN y ha obtenido 3 Estrellas de ese programa.

3.4.1 Patrones *Gang of Four*¹⁸

Es necesario destacar que debido a la manera particular que Drupal establece la escritura de módulos, en la implementación de los módulos de la solución está reflejado el patrón de diseño creacional *Singleton* y los patrones de diseño de comportamiento *Chain of Responsibility* y *Command*, que resuelven problemas relativos a la interacción entre objetos (Guerrero, Suárez y Gutiérrez, 2013).

- ***Singleton***: Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. Todos los objetos que utilicen una instancia de una clase *Singleton* utilizan la misma instancia.

A pesar de la falta de clases explícitamente declaradas en Drupal, muchos paradigmas orientados a objetos se utilizan aún en su diseño. Un módulo en Drupal es una instancia con funciones únicas, por lo que en la aplicación el módulo Autenticación UCI puede ser pensado como una clase, de ahí se puede decir que sigue el patrón de diseño creacional *Singleton*.

- ***Chain of Responsibility***: Evita emparejar el objeto que envía una petición con el objeto que recibe esa petición, al permitir que más de un objeto pueda tomar la responsabilidad de la petición enviada. Encadena los objetos que podrán recibir esa petición y la pasa a lo largo de la cadena hasta que un objeto la acepte.

El sistema de menús de Drupal sigue el patrón de diseño de comportamiento *Chain of Responsibility*. En cada solicitud de la página el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y que la función se llama para hacer el trabajo.

- ***Command***: Transforma una petición en objeto de tal manera que se pueda catalogar clientes con diferentes peticiones, mantener un registro de las peticiones y soportar el deshacer operaciones. Muchos de los ganchos de Drupal utilizan el patrón de diseño de comportamiento *Command* para reducir el número de funciones que son necesarias para la aplicación, pasando por la operación como un parámetro junto con los argumentos.

¹⁸ GoF, en español Banda de los Cuatro. Es el nombre con el que se conoce comúnmente a los autores del libro *Design Patterns* (en español *Patrones de Diseño*) (ISBN 0-201-63361-2), los mencionados autores son: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

3.5 Diagrama Entidad Relación

El Diagrama Entidad Relación mostrado en la Figura 7 representa el Modelo lógico de datos correspondiente a la base de datos de la propuesta de solución.

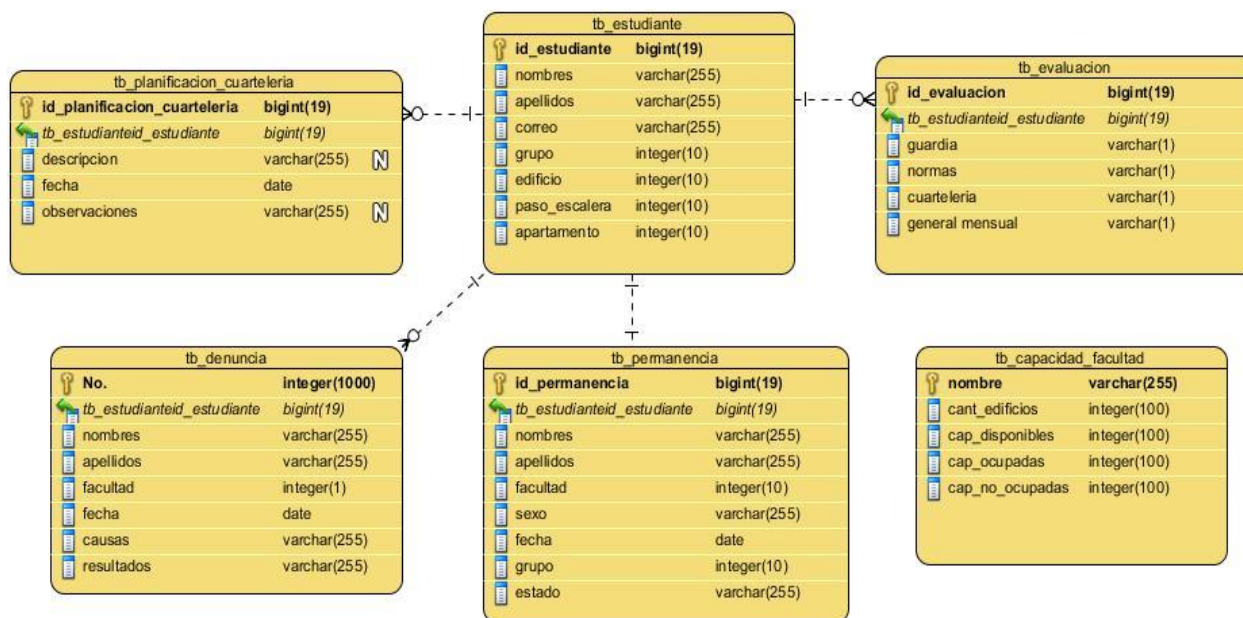


Figura 7 Modelo lógico de datos.

A continuación se muestra una descripción de las tablas que conforman la base de datos que la aplicación utiliza para hacer persistir la información que gestiona:

- **tb_planificacion_cuartereria:** Es la encargada de almacenar la información referente a la planificación de la cuarterería que se realiza en el sistema. Posee una relación de uno a muchos con la tabla tb_estudiante. Un estudiante estará relacionado con más de una planificación de la cuarterería y una planificación de la cuarterería le corresponde a solo un estudiante.
- **tb_estudiante:** Almacena los datos de un estudiante. Posee una relación de uno a muchos con las tablas tb_planificaion_cuartereria, tb_denuncia y tb_evaluacion. Un estudiante estará relacionado con más de una denuncia y una denuncia le corresponde solo a un estudiante, tal es el caso de la relación con las otras dos tablas antes mencionadas.

- **tb_evaluacion:** Contiene la evaluación de la guardia, las normas, la cuarterería y la evaluación general mensual referente a un estudiante.
- **tb_denuncia:** En ella se almacenan los datos de la denuncia realizada por un estudiante.
- **tb_permanencia:** Registra la permanencia de cada estudiante. Tiene una relación de uno a uno con la tb_estudiante, debido a que un estudiante solo puede tener un estado de permanencia y un estado de permanencia le corresponde a solo un estudiante.
- **tb_capacidad_facultad:** Contiene la información referente al estado físico de la residencia de una facultad.

3.6 Estilos y estándares de codificación

Drupal cuenta con estándares de codificación que permiten lograr una implementación limpia y organizada, siendo esta entendible para cualquier desarrollador que utilice el código desarrollado (Drupal, 2015). A continuación se definen los estándares utilizados en la solución propuesta.

Indentación: La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos. En Drupal la indentación debe ser de 2 espacios.

```
93 function uciauthentication_block_configure($delta = '') {
94     $form = array();
95     if ($delta) {
96         if (module_exists('image')) {
97             $form['uciauthentication_user_picture_style'] = array(
98                 '#type' => 'select',
99                 '#title' => t('Picture display style'),
100                 '#options' => image_style_options(TRUE),
101                 '#default_value' => variable_get('uciauthentication_user_picture_style', ''),
102                 '#description' => t("The style selected will be used on display, while the original image is retained.
103             );
104         }
105     }
106     return $form;
107 }
```

Figura 8 Indentación del código.

Estructuras de control: Con respecto a las estructuras de control, se debe tener en cuenta las siguientes normas:

- Debe haber un espacio entre el comando que define la estructura (*if*, *while*, *for*) y el paréntesis de apertura permitiendo no confundir las estructuras de control con la nomenclatura de las funciones.
- La llave de apertura ({) se sitúa en la misma línea que la definición de la estructura, separada por un espacio.
- Se recomienda usar siempre las llaves { } aún en los casos en que no sea obligatorio su uso.
- Las estructuras *else* y *elseif* se deben escribir en la línea siguiente al cierre de la sentencia anterior.

```
337 if (isset($form_state['uid']) && is_numeric($form_state['uid'])) {
338     $account = user_load($form_state['uid']);
339     if (($mode == UCIAUTHENTICATION_AUTHENTICATION_MODE_MIXED && !isset($a
340         form_clear_error();
341         return $account;
342     }
343     elseif ($mode == UCIAUTHENTICATION_AUTHENTICATION_MODE_EXCLUSIVE) {
344         $form_state['uid'] = NULL;
345     }
346 }
347
348 $account = user_load_by_name($username);
349 if (is_object($account)) {
350     $account_exists = TRUE;
351     $is_uci = isset($account->data['uciauthentication']);
352 }
353 else { // account does not exist
354     $account_exists = FALSE;
355 }
```

Figura 9 Estructuras de control.

Arreglos: Los valores dentro de un arreglo se deben separar por un espacio después de la coma que los separa. El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del arreglo supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez con 2 espacios. En este último caso, la coma de separación del último elemento también se debe escribir, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```

21 function uciauthentication_menu() {
22   $items['admin/config/people/uciauthentication'] = array(
23     'title' => decode_entities('Configuración de la Autenticación UCI'),
24     'description' => decode_entities('Configurar el modo de autenticación:
25     realizar en caso de conflicto de cuentas'),
26     'page callback' => 'drupal_get_form',
27     'page arguments' => array('uciauthentication_settings_form'),
28     'access callback' => 'user_access',
29     'access arguments' => array('administer site configuration'),
30   );

```

Figura 10 Arreglos.

3.7 Tareas de ingeniería

XP propone dividir cada HU en tareas de ingeniería con el objetivo de facilitar la implementación a los programadores, estas tareas deben tener una duración de entre uno y tres días aproximadamente. Las 17 HU redactadas fueron desglosadas en un total de 42 tareas de ingeniería, en este epígrafe solo se muestran las relacionadas con las HU que se describen en el Capítulo 2, las restantes se encuentran en el Anexo 3 de la investigación.

Tabla 17 Tarea #17.

| Tarea | |
|---|---|
| Número: 17 | Número de HU: 5 |
| Nombre: Implementar la visualización evaluación mensual del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de febrero de 2015 | Fecha fin: 28 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Visualizar evaluación mensual del estudiante en el sistema. | |

Tabla 18 Tarea #18.

| Tarea | |
|--|---|
| Número: 18 | Número de HU: 5 |
| Nombre: Implementar la inserción de la evaluación mensual del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de febrero de 2015 | Fecha fin: 28 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Insertar evaluación mensual del estudiante en el sistema. | |

Tabla 19 Tarea #19.

| Tarea | |
|---|---|
| Número: 19 | Número de HU: 5 |
| Nombre: Implementar la modificación de la evaluación mensual del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de febrero de 2015 | Fecha fin: 28 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Modificar evaluación mensual del estudiante en el sistema. | |

Tabla 20 Tarea #20.

| Tarea | |
|--|---|
| Número: 20 | Número de HU: 5 |
| Nombre: Implementar la eliminación de la evaluación mensual del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de febrero de 2015 | Fecha fin: 28 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Eliminar evaluación mensual del estudiante del sistema. | |

Tabla 21 Tarea #25.

| Tarea | |
|--|---------------------------------------|
| Número: 25 | Número de HU: 10 |
| Nombre: Implementar la búsqueda de datos. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 19 de marzo de 2015 | Fecha fin: 21 de marzo de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá contener un buscador para realizar las búsquedas por datos, ejemplo Estudiantes, Estudiantes por edificios, Cuartelería, Cuartelería por día y Evaluación. | |

Tabla 22 Tarea #27.

| Tarea | |
|--|---------------------------------------|
| Número: 27 | Número de HU: 12 |
| Nombre: Implementar la ubicación del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de marzo de 2015 | Fecha fin: 28 de marzo de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir ubicar a los estudiantes que ingresen nuevos en la universidad. | |

Tabla 23 Tarea #28.

| Tarea | |
|---|---|
| Número: 28 | Número de HU: 12 |
| Nombre: Implementar la modificación de la ubicación del estudiante. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 26 de febrero de 2015 | Fecha fin: 28 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir modificar la ubicación de los estudiantes continuantes en caso de que se produjeran cambios. | |

Tabla 24 Tarea #29.

| Tarea | |
|---|--------------------------------------|
| Número: 29 | Número de HU: 13 |
| Nombre: Implementar la visualización de datos de edificios por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 30 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá visualizar los datos de los edificios por facultad. | |

Tabla 25 Tarea #30.

| Tarea | |
|---|--------------------------------------|
| Número: 30 | Número de HU: 13 |
| Nombre: Implementar la inserción de datos de edificios por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 30 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá insertar datos de los edificios por facultad. | |

Tabla 26 Tarea #31.

| Tarea | |
|--|--------------------------------------|
| Número: 31 | Número de HU: 13 |
| Nombre: Implementar la modificación de datos de edificios por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 30 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá modificar los datos de los edificios por facultad. | |

Tabla 27 Tarea #32.

| Tarea | |
|-------------------|-------------------------|
| Número: 32 | Número de HU: 13 |

| | |
|---|--------------------------------------|
| Nombre: Implementar la eliminación de datos de edificios por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 30 de marzo de 2015 | Fecha fin: 1 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá eliminar los datos de los edificios por facultad. | |

3.8 Interfaces del sistema

A continuación se muestran las principales interfaces de la propuesta de solución que se implantará en la Dirección de la Residencia Estudiantil 2 de la UCI.

The screenshot shows a web interface for creating a student. At the top, there is a breadcrumb trail: 'Inicio » Agregar contenido'. Below this is a header bar with the text 'Crear Estudiante' and a circular icon. The form contains several fields, each with a red asterisk indicating it is required:

- Nombre y Apellidos ***: A text input field.
- Correo ***: A text input field.
- Grupo ***: A text input field.
- Edificio ***: A dropdown menu with '1' selected.
- Paso de escalera ***: A dropdown menu with '1' selected.
- Apartamento ***: A dropdown menu with '01' selected.

Figura 11 Interfaz “Adicionar estudiante”.

Nombre y Apellidos *

Facultad *

Facultad 1

Sexo *

Masculino

FECHA

Fecha

02/06/2015

Por ejemplo, 02/06/2015

Grupo *

Estado de permanencia *

Alta

Figura 12 Interfaz “Registrar permanencia del estudiante”.

Inicio » Vistas » Cuartelería

Cuartelería

Paso de escalera Edificio

1 1

| Estudiante | Descripción | Edificio | Paso de escalera | borrar | editar |
|---------------------------|----------------------|----------|------------------|----------|--------|
| Yanet Jimenez Escobar | Estar a las 8:00 am | 1 | 1 | eliminar | editar |
| Yarisleydi Soca Hernández | Cuartelería asignada | 1 | 1 | eliminar | editar |
| Michelle Pérez Acosta | Cuartelería | 1 | 1 | eliminar | editar |
| Sandy García Santos | Cuartelería | 1 | 1 | eliminar | editar |
| Yairon Consuegra Cabrera | Cuartelería | 1 | 1 | eliminar | editar |

Figura 13 Interfaz “Cuartelería”.

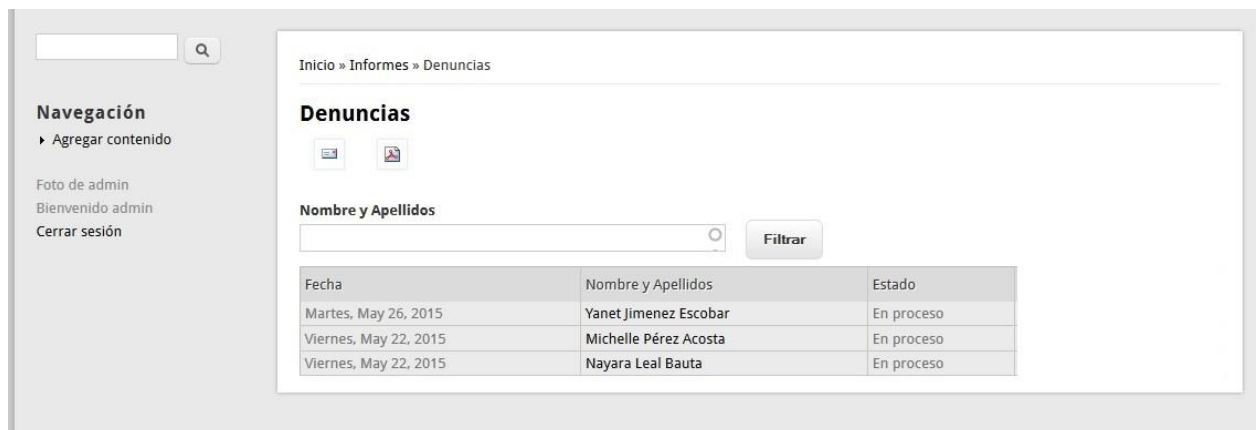


Figura 14 Interfaz “Informe de denuncias”.

3.9 Pruebas

La metodología XP enfatiza en la realización de pruebas a lo largo de todo el desarrollo del *software*, con el fin de lograr un producto con calidad, reduciendo el número de errores y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. En este proceso no solo participa el equipo de desarrollo, también es importante la colaboración del cliente, sobre todo en las pruebas de aceptación.

3.9.1 Desarrollo dirigido por pruebas (TDD¹⁹)

El desarrollo dirigido por pruebas es una característica de la metodología XP para comprobar el funcionamiento de los códigos que se van implementando. Este estilo sigue el principio *test-first*, significa que las pruebas se crean antes de comenzar la implementación. La idea es que al pensar en cómo se probará la funcionalidad, se está pensando en la propia funcionalidad desde el punto de vista de su interfaz (qué métodos tendrá y con qué parámetros), ayudando a desarrollar así un mejor diseño. De esta manera se asegura que no exista ninguna funcionalidad que no esté probada. Una vez creadas las pruebas, el código no pasa a producción hasta que el resultado de las mismas no sea satisfactorio. Luego se inicia el proceso de refactorización que consiste en limpiar y organizar el código, adaptarlo a los patrones y aumentar su legibilidad, todo esto sin modificar su comportamiento externo. (Koskela, 2008).

3.9.2 Pruebas unitarias

En la presente investigación se utilizó el nivel de pruebas unitarias utilizando la técnica de caja blanca.

19 Del inglés *Test Driven Development*.

Estas pruebas permiten determinar si un módulo del sistema está listo y correctamente terminado.

A la hora de aplicarlas se tienen en cuenta algunos parámetros, los cuales se muestran a continuación.

- Los datos de entrada son conocidos por el Encargado de Pruebas y estos deben ser preparados con minuciosidad, puesto que el resultado de las pruebas dependen de estos.
- Se debe conocer qué componentes interactúan en cada caso de prueba.
- Se debe conocer de antemano, qué resultados debe devolver el componente según los datos de entrada utilizados en la prueba.

Finalmente se deben comparar los datos obtenidos en la prueba con los datos esperados, si son idénticos el módulo supera la prueba (Tello, 2012).

3.9.2.1 Módulo *Unit Testing*

Para la realización de las pruebas unitarias en el CMS Drupal se utilizó el módulo *Unit Testing*, el cual es una práctica que consiste en tomar una pequeña porción de código para probar su corrección. Por lo general se someten a pruebas, funciones o pequeñas funcionalidades específicas con un conjunto de parámetros controlados que servirán para hacer las comprobaciones necesarias en función de los valores de retorno de la funcionalidad testeada. Dentro de sus utilidades se pueden observar:

- Ayuda a producir un código de mayor calidad.
- Detección rápida de errores cuando se programan nuevas funcionalidades o se realizan cambios en el código.
- Sirve como pequeña fuente de documentación sobre qué es lo que se espera que haga el código.
- Realizar pruebas unitarias obliga al programador a escribir el código en pequeñas porciones con el fin de que puedan ser probadas independientemente.

A continuación se muestra el resultado de las pruebas unitarias realizadas al módulo Autenticación UCI de la aplicación.

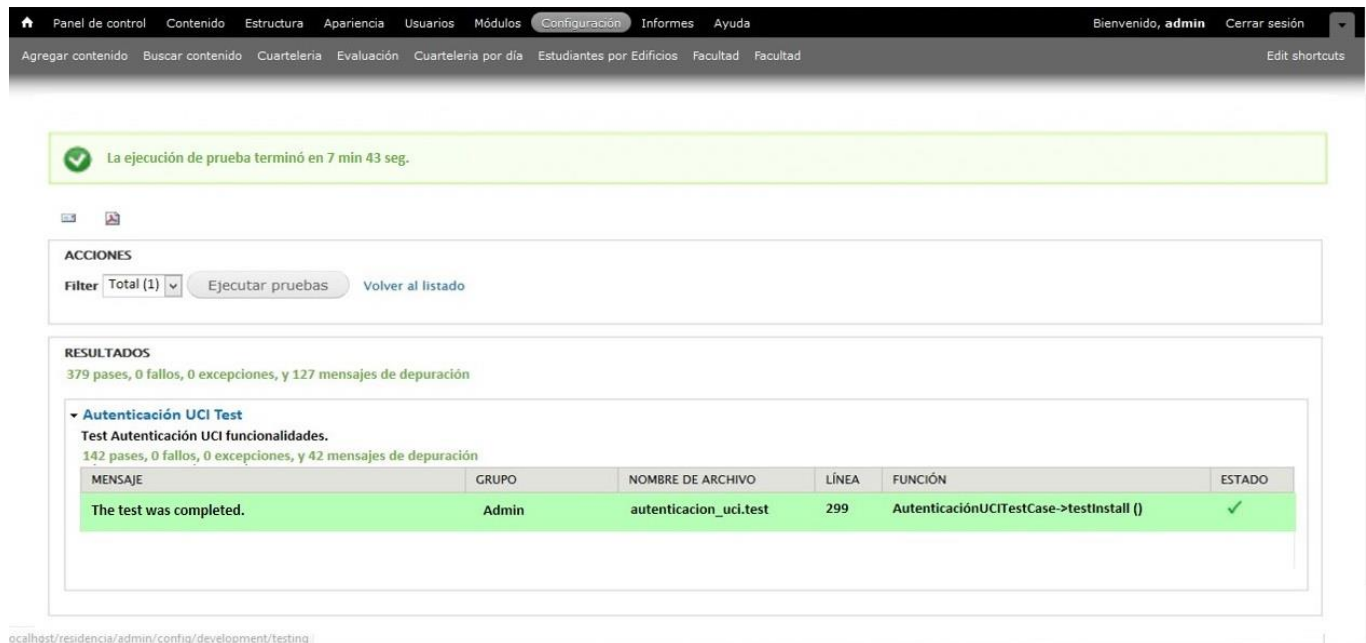


Figura 15 Prueba unitaria módulo Autenticación UCI satisfactoria.

En caso de la prueba ser insatisfactoria, se detiene el proceso indicando donde se encuentra el error y una vez los desarrolladores hayan solucionado el problema permite volver a intentar la prueba, donde se pueden encontrar nuevos errores o terminar exitosamente.

3.9.3 Pruebas de aceptación

Para realizarle pruebas a las funcionalidades del sistema se utilizó el nivel de pruebas de aceptación utilizando la técnica de caja negra.

Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Describen, por una HU dada, cómo el sistema se encarga de ciertas condiciones e insumos y con qué tipo de resultados (Koskela, 2008).

Estas pruebas fueron escritas por el equipo de desarrollo en conjunto con el cliente antes de desarrollar cada HU y aplicadas al final de cada iteración. A continuación se presentan los casos de pruebas de aceptación pertenecientes a las HU mostradas en el Capítulo 2, los restantes se encuentran en el Anexo 4 de la presente investigación.

Tabla 28 Caso de prueba de aceptación #17.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU5_CP17 | Historia de usuario: 5 |
| Nombre: Visualizar evaluación mensual del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la pestaña donde se muestran las evaluaciones mensuales de los estudiantes. Pasos de ejecución: Para visualizar la evaluación mensual del estudiante se debe ir a Vistas, Evaluación y se muestran las evaluaciones por estudiante. | |
| Resultado esperado: El sistema muestra la evaluación mensual de cada uno de los estudiantes. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 29 Caso de prueba de aceptación #18.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU5_CP18 | Historia de usuario: 5 |
| Nombre: Insertar evaluación mensual del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante y escoge la opción Crear evaluación mensual y llena los campos necesarios. Pasos de ejecución: Para insertar la evaluación mensual del estudiante se debe ir a Módulo Estudiante, Crear evaluación mensual, llenar los campos y finalmente guardar. | |
| Resultado esperado: El sistema almacena la evaluación mensual de cada uno de los estudiantes. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 30 Caso de prueba de aceptación #19.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU5_CP19 | Historia de usuario: 5 |
| Nombre: Modificar evaluación mensual del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar en la evaluación deseada. | |

| |
|---|
| <p>Pasos de ejecución: Para modificar la evaluación mensual del estudiante se debe ir a Vistas, Evaluación, clic encima del Nombre y Apellidos del estudiante, clic encima de la evaluación mensual e ir a la pestaña editar, una vez allí se modifica esta evaluación y se guarda.</p> |
| <p>Resultado esperado: El sistema almacena la nueva evaluación mensual registrada del estudiante.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria</p> |

Tabla 31 Caso de prueba de aceptación #20.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU5_CP20 | Historia de usuario: 5 |
| Nombre: Eliminar evaluación mensual del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar en la evaluación deseada. Pasos de ejecución: Para eliminar la evaluación mensual del estudiante se debe ir a Vistas, Evaluación, clic encima del Nombre y Apellidos del estudiante, clic encima de la evaluación mensual e ir a la pestaña editar, una vez allí se presiona el botón Eliminar y se confirma. | |
| Resultado esperado: El sistema elimina la evaluación deseada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 32 Caso de prueba de aceptación #25.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU10_CP25 | Historia de usuario: 10 |
| Nombre: Buscar datos. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos en el sistema. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y realiza una búsqueda en el sistema especificando los campos. Pasos de ejecución: Para buscar cualquier dato se debe ir a Vistas y una vez allí se busca por el criterio que se desee. | |
| Resultado esperado: El sistema muestra los datos encontrados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 33 Caso de prueba de aceptación #27.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU12_CP27 | Historia de usuario: 12 |
| Nombre: Ubicación del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario entra una ubicación para cada uno de los estudiantes especificando edificio y apartamento. Pasos de ejecución: Para ubicar al estudiante se debe ir a Módulo Estudiante, Adicionar Estudiante y se llenan los campos que incluyen edificio y apartamento. | |
| Resultado esperado: El sistema almacena la ubicación del estudiante. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 34 Caso de prueba de aceptación #28.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU12_CP28 | Historia de usuario: 12 |
| Nombre: Modificar ubicación del estudiante. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario escoge la opción Editar en una ubicación determinada. Pasos de ejecución: Para modificar la ubicación del estudiante se debe ir a Vistas, Estudiante, dar clic encima de Nombre y Apellidos, luego se da clic encima del correo del estudiante y una vez allí se escoge la opción Editar y se modifican los campos que se deseen. | |
| Resultado esperado: El sistema almacena la nueva ubicación insertada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 35 Caso de prueba de aceptación #30.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU14_CP30 | Historia de usuario: 13 |
| Nombre: Visualizar datos de edificios por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Informes y escoge la pestaña donde se muestra la información de los edificios por facultad. | |

| |
|---|
| Pasos de ejecución: Para visualizar datos de edificio por facultad se debe ir a Informes, Informe de facultad. |
| Resultado esperado: El sistema muestra los datos de cada uno de los edificios por facultad. |
| Evaluación de la prueba: Prueba satisfactoria |

Tabla 36 Caso de prueba de aceptación #31.

| Caso de prueba de aceptación | |
|--|--------------------------------|
| Código: HU14_CP31 | Historia de usuario: 13 |
| Nombre: Insertar datos de edificios por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario registra datos de edificios. Pasos de ejecución: Para visualizar datos de edificio por facultad se debe ir a Módulo Facultad, Registrar capacidades por facultad, se llenan los campos y finalmente se guarda. | |
| Resultado esperado: El sistema almacena los datos registrados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 37 Caso de prueba de aceptación #32.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU14_CP32 | Historia de usuario: 13 |
| Nombre: Modificar datos de edificios por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Informes y escoge la opción Editar en los datos deseados. Pasos de ejecución: Para modificar datos de edificios por facultad se debe ir a Informes, Informe de facultad, clic sobre la facultad que se desee modificar y luego se va a la pestaña Editar, se modifican los datos y finalmente se guarda. | |
| Resultado esperado: El sistema almacena la nueva información registrada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 38 Caso de prueba de aceptación #33.

| Caso de prueba de aceptación | |
|--|--------------------------------|
| Código: HU14_CP33 | Historia de usuario: 13 |
| Nombre: Eliminar datos de edificios por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Informes y escoge la opción Eliminar. Pasos de ejecución: Para eliminar datos de edificios por facultad se debe ir a Informes, Informe de facultad, clic sobre la facultad que se desee eliminar y luego se va a la pestaña Editar, se presiona el botón Eliminar y finalmente se confirma. | |
| Resultado esperado: El sistema elimina los datos seleccionados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

3.10 Análisis de los resultados de las pruebas de aceptación

Al concluir las tres iteraciones planificadas para el desarrollo de la propuesta de solución, fueron realizadas las pruebas pertinentes para las entregas pactadas con el cliente, la Tabla 38 muestra el resultado de estas pruebas en cada una de las iteraciones.

Tabla 39 Cantidad de No conformidades detectadas por iteración.

| | Iteración 1 | Iteración 2 | Iteración 3 |
|------------------------------|-------------|-------------|-------------|
| Cantidad de No conformidades | 14 | 9 | 5 |

Todas las No conformidades fueron resueltas, lo cual valida la calidad de la propuesta de solución, logrando una mayor satisfacción por parte del cliente.

3.11 Conclusiones

Realizado el desarrollo y pruebas a la propuesta de solución planteada se arriba a las siguientes conclusiones:

- La descripción de la arquitectura de la solución propuesta permitió comprender cómo gestionar contenidos para trabajar con el sistema en futuras actualizaciones del *software*.
- Se emplearon para realizar la propuesta de solución, los patrones de diseño GoF, los cuales permitieron aplicar buenas prácticas para modelar e implementar el sistema, logrando así una mayor mantenibilidad del código.

- Las tareas de ingeniería realizadas le indicaron al equipo de desarrollo los objetos que se deben construir.
- El proceso de implementación estuvo guiado por la utilización de las tecnologías seleccionadas, logrando al final de las iteraciones pactadas, un producto con las funcionalidades requeridas.
- Para la validación del *software* se realizaron pruebas unitarias y de aceptación al sistema, planteadas por la metodología XP. Con estas se detectaron un total de 28 No conformidades repartidas en cada iteración, las cuales fueron resueltas al final de cada una de estas iteraciones, mejorando así la calidad del sistema.

Conclusiones generales y Recomendaciones

Conclusiones generales

La culminación de la presente investigación permitió arribar a las siguientes conclusiones:

- Se realizó un estudio de las soluciones similares existentes a nivel nacional e internacional, donde se evidenció que no existe un sistema que garantice la automatización de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula”, demostrándose la necesidad de crear un sistema informático para la gestión de la información de estos procesos en la Dirección de la Residencia Estudiantil 2 de la UCI.
- Se construyó un sistema informático utilizando la metodología XP, como Sistema Gestor de Contenido que rigió el proceso de desarrollo Drupal 7.21 y como tecnologías y herramientas: UML 2.1 para el modelado, Visual Paradigm 8.0 como herramienta CASE, PHP, JavaScript, HTML 5 y CSS 3 como lenguajes de programación, jQuery 1.8.3 y Bootstrap 2.2.2 como *framework* para el desarrollo, PhpStorm 8.0.1 como IDE de desarrollo, Apache 2.2.22 como servidor web y MySQL 5.0.15 como SGBD.
- Se llevó a cabo el desarrollo dirigido por pruebas que define la metodología seleccionada en cuanto a pruebas unitarias y pruebas de aceptación y se utilizó el módulo Unit Testing que provee el CMS Drupal para las pruebas unitarias, en el caso de las pruebas de aceptación se realizaron en conjunto con el cliente, detectando un total de 28 No conformidades en tres iteraciones las cuales fueron resueltas al finalizar cada iteración, mejorando así la calidad del sistema.
- Se logró entregar en tiempo la solución propuesta, donde el cliente firmó el aval de aceptación de la misma, esta se implantará en la Dirección de la Residencia Estudiantil 2 de la UCI y posibilitará apoyar la gestión de los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula”, pertenecientes a esta área.

Recomendaciones

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda:

- Generalizar la propuesta a toda la residencia estudiantil.
- Incorporar a la aplicación la automatización de los restantes procesos que se gestionan en la residencia, como Guardia obrera y Evaluación de desempeño.

Referencias Bibliográficas

1. **ACHOUR, Mehdi, BETZ, Friedhelm, DOVGAL, Antony, LOPEZ, Nuno, MAGNUSSON, Hannes, RICHTER, Georg, SEGUY, Damien y VRANA, Jakub.** Manual de PHP. [En Línea]. [Accedido 13 Enero 2015]. Disponible en: <https://php.net/manual/es/index.php>
2. **BECK, Kent.** *Planning Extreme Programming.* Addison-Wesley Professional, 2001. ISBN 9780201710915.
3. **BECK, Kent y ANDRES, Cynthia.** *Extreme Programming Explained. Embrace Change.pdf.* . 2004. ISBN 0321278658
4. **BUILTWITH PTY LTD.** Web Server technologies Web Usage Statistics. [En Línea]. [Accedido 4 Diciembre 2014]. Disponible en: <http://trends.builtwith.com/web-server>
Usage Statistics for Web Servers
5. **BUILTWITH PTY LTD.** Design Framework technologies Web Usage Statistics. [En Línea]. [Accedido 7 Enero 2015]. Disponible en: <http://trends.builtwith.com/docinfo/design-framework>
Usage Statistics for Design Framework technologies
6. **BUILTWITH PTY LTD.** JavaScript Library technologies Web Usage Statistics. [En Línea]. 2015. [Accedido 16 Enero 2015]. Disponible en: <http://trends.builtwith.com/javascript/javascript-library>
Usage Statistics for JavaScript Library technologies
7. **BUYTAERT, Dries.** Drupal - Open Source CMS | Drupal.org. [En Línea]. [Accedido 4 Diciembre 2014]. Disponible en: <https://www.drupal.org/>
8. **CABELLO, Ma Victoria Nevado.** *Introduccion a Las Bases de Datos Relacionales.* Editorial Visión Libros, 2008. ISBN 9788499836171.
9. **CALABRIA, Luis y PÍRIZ, Pablo.** *Metodología_XP.pdf.* . 2003. Universidad ORT Uruguay, Facultad de Ingeniería

10. **CANÓS, José H., LETELIER, Patricio y PENADÉS, Ma. Carmen.** 2003-Metodologías ágiles en el desarrollo del software.pdf. . 2003.
11. **CATALINAS, ROMÁN y RODRÍGUEZ.** *Mantenimiento de portales de la Información: explotación de sistemas informáticos.* Editorial Paraninfo, 2007. ISBN 9788497325042.
12. **COMPAÑÍA IMAJESTIC INTERNET MARKETERS.** Comparing the Three Major CMS's: Drupal, Joomla and WordPress | Performance Based SEO company iMajestic Internet Marketing Blog. [En Línea]. 2010. [Accedido 23 Febrero 2015]. Disponible en: <http://www.imajestic.com/blog/2009/07/comparing-the-three-major-cms%e2%80%99s-drupal-joomla-and-wordpress/> Comparing the Three Major CMS's: Drupal, Joomla and WordPress.
13. **COMUNICARE.** Principales CMS Open Source: Joomla, Wordpress y Drupal - Comunicare. [En Línea]. 2015. [Accedido 9 Febrero 2015]. Disponible en: <http://www.comunicare.es/cms-joomla-wordpress-magento/#>
14. **DRUPAL. Arquitectura | Tutorial Drupal.** [En Línea]. 2015. [Accedido 27 Abril 2015]. Disponible en: <http://www.cursosdrupal.com/content/arquitectura>
15. **DRUPAL. Coding standars | Drupal.org.** [En Línea]. 2015. [Accedido 28 Abril 2015]. Disponible en: <https://www.drupal.org/node/908844>
16. **EGUILUZ, Javier.** Introducción a CSS [En Línea]. 2015. [Accedido 16 Enero 2015]. Disponible en: <http://librosweb.es/css/>
17. **EGUILUZ, Javier.** CSS avanzado [En Línea]. 2015. [Accedido 7 Enero 2015]. Disponible en: http://librosweb.es/css_avanzado/capitulo_5.html
18. **EGUILUZ, Javier.** Introducción a JavaScript [En Línea]. 2015. [Accedido 13 Enero 2015]. Disponible en: <http://librosweb.es/javascript/>
19. **GARCÍA CARRERO, Raúl.** 10 razones para usar CSS. *Maestros del Web* [En Línea]. 2008. [Accedido 24 Febrero 2015]. Disponible en: <http://www.maestrosdelweb.com/usarcss/>

20. **GIL, Fran. Forcontu** - Experto en Drupal 7 - 01 - Nivel inicial.pdf. 2012.
21. **GUERRERO, Carlos A., SUÁREZ, Johanna M. y GUTIÉRREZ, Luz E.** Información tecnológica - Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [En Línea]. 2013. [Accedido 2 Junio 2015]. Disponible en: http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext
22. **HERNÁNDEZ, León Rolando Alfredo y COELLO, González Sayda.** El proceso de investigación científica (2 edición) - Libro Rolando II Edición 2012.pdf
23. **JEFFRIES, Ron, ANDERSON, Ann y HENDRICKSON, Chet.** *Extreme Programming Installed.* Addison-Wesley Professional, 2001. ISBN 9780201708424. Extreme programming.
24. **JOSKOWICZ, José.** Reglas y Prácticas en eXtreme Programming. . 2008.
25. **KOSKELA, LASSE.** TEST DRIVEN, Practical TDD and Acceptance TDD for Java Developers. Manning Publication Co., 2008. ISBN 1-932394-85-0.
26. **LORENZO, Manuel.** La importancia de la seguridad informática. COLIBRIS Openpartners [En Línea]. 2014. [Accedido 12 Mayo 2015]. Disponible en: <http://www.colibris.es/blog/importancia-seguridad-informatica>
27. **LUCA, Damian.** HTML5 [En Línea]. USERSHOP, 2015. ISBN 9789871773794. Disponible en: <http://books.google.com.cu/books?id=DGwjKIXWPm8C&lpg=PA29&dq=ventajas%20de%20css&hl=es&pg=PA30#v=onepage&q=ventajas%20de%20css&f=true>
28. **MORA, Sergio Luján.** *Programación de aplicaciones web: Historia, principios básicos y clientes web* [En Línea]. Editorial Club Universitario, 2002. ISBN 9788484542063. Disponible en: http://www.google.com.cu/books?hl=en&lr=&id=r9CqDYh2-loC&oi=fnd&pg=PR3&dq=Programaci%C3%B3n+de+aplicaciones+web:+historia,+principios+b%C3%A1sicos+y+clientes+web+&ots=MiyOWh5QC1&sig=6_A7osdNx_rxWJjV7_0JrqOpEVk&r

edir_esc=y#v=onepage&q=Programaci%C3%B3n%20de%20aplicaciones%20web%3A%20historia%2C%20principios%20b%C3%A1sicos%20y%20clientes%20web&f=false

29. **MERELO GUERVÓS, Juan Julian.** Introducción a los sistemas de gestión de contenidos. [En Línea]. 2005. [Accedido 23 Febrero 2015]. Disponible en: <http://geneura.ugr.es/~jmerelo/tutoriales/cms/>
30. **ORACLE.** Oracle y MySQL | Database | Oracle. [En Línea]. [Accedido 18 Marzo 2015]. Disponible en: <http://www.oracle.com/es/products/mysql/index.html>
31. **PRECIADO, Mariana Palacios.** Modelos de Negocio: Propuesta de un Marco Conceptual para Centros de Productividad. [En Línea]. 2011. [Accedido 28 Abril 2015]. Disponible en: <http://www.bdigital.unal.edu.co/5152/1/940794.2011.pdf>
32. **PEÑA, Jenny Ruiz de la y CÉSPEDES, Indira Cuba.** Sistema de gestión de información para la Residencia Universitaria de la Universidad de Holguín Oscar Lucero Moya. Ciencias Holguín [En línea]. 15 Julio 2010. Vol. 16, no. 1. [Accedido 11 Diciembre 2014]. Disponible en: <http://www.ciencias.holguin.cu/index.php/cienciasholguin/article/view/547>
Sistema de gestión de información para la Residencia Universitaria de la Universidad de Holguín Oscar Lucero Moya.
33. **REAL ACADEMIA ESPAÑOLA.** Diccionario de la lengua española. In: [En Línea]. 2015. [Accedido 9 febrero 2015]. Disponible en: <http://lema.rae.es/drae/?val=control>.
34. **REAL ACADEMIA ESPAÑOLA.** Diccionario de la lengua española. In: [En Línea]. 2015. [Accedido 9 febrero 2015]. Disponible en: <http://lema.rae.es/drae/?val=planificacion>.
35. **RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady.** El lenguaje Unificado de Modelado. Manual de Referencia.pdf. [En Línea]. 2000. [Accedido 4 Diciembre 2014]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>
36. **SOMMERVILLE, Ian.** Ingeniería del Software 7ma. Ed. - Ian Sommerville.pdf. [En Línea]. 2005. [Accedido 6 Enero 2015]. Disponible en:

- <http://zeus.inf.ucv.cl/~bcrawford/Modelado%20UML/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf>
ISBN 8478290745
37. **SPONSORS, Members.** JQuery. [En Línea]. 2015. [Accedido 16 Enero 2015]. Disponible en:
<http://jquery.com/>
JQuery: The Write Less, Do More, JavaScript Library
38. **SQL MANAGER. EMS SQL Manager** - EMS SQL Manager for MySQL: MySQL Database Tool, MySQL Management and Development, MySQL Server Administration, Export and Import Tools for MySQL. [En Línea]. [Accedido 18 Marzo 2015]. Disponible en:
<http://www.sqlmanager.net/products/mysql/manager/>
39. **TELLERIA, Oliver Milan, VÁZQUEZ, Manuel López y ALEA, Caridad Salazar.** Sistema automatizado informativo para la comunidad de residencia estudiantil de la Universidad de Pinar del Rio “Hermanos Saiz Montes de Oca.” [En Línea]. 2008. [Accedido 2 Junio 2015]. Disponible en: <http://www.upr.edu.cu/>
40. **TELLO, Eduardo A. Rodríguez.** Estrategias y técnicas de prueba del software.pdf. 2012.
41. **UC3M.** Accesibilidad | UC3M. [En Línea]. [Accedido 11 Marzo 2015]. Disponible en:
<http://www.uc3m.es/ss/Satellite?c=Page&cid=1371207865836&idioma=es&pagename=UC3MInstitucional%2FPage%2FTextoMixta&rendermode=preview&site=UC3MInstitucional>
42. **UNIVERSIDAD DE COLUMBIA.** HOME | CU Facilities - Housing. [En Línea]. 2015. [Accedido 9 Febrero 2015]. Disponible en: <http://facilities.columbia.edu/housing/intro-ocha-3>
43. **UNIVERSIDAD POLITÉCNICA DE VALENCIA.** Introducción a Herramientas CASE y System Architect.pdf. [En Línea]. 2013. [Accedido 2 Junio 2015]. Disponible en:
http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf
44. **USC.** Contacto - Enlaces del web - USC. [En Línea]. 2015. [Accedido 9 Febrero 2015]. Disponible en: <https://www.usc.es/es/web/contacto.html>

45. **What's New in Visual Paradigm 12.1** [En Línea]. 2015. [Accedido 20 Mayo 2015]. Disponible en: <http://www.visual-paradigm.com/whats-new/>
46. **W3C**. W3C Oficina Española - El W3C de la A a la Z. [En Línea]. [Accedido 16 Enero 2015]. Disponible en: <http://www.w3c.es/Divulgacion/a-z/#h>
47. **W3C**. Guía Breve de CSS. [En Línea]. [Accedido 16 Enero 2015]. Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

Anexos

Anexo 1: Gráficos

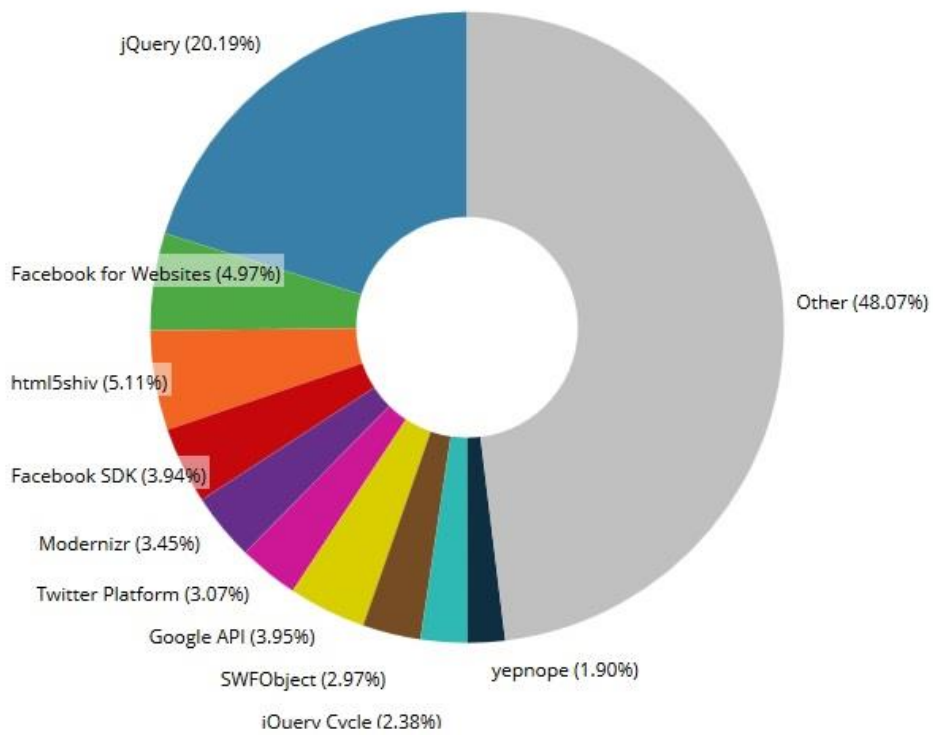


Figura 16: Estadísticas del uso de tecnologías JavaScript (BuiltWith Pty Ltd, 2015).

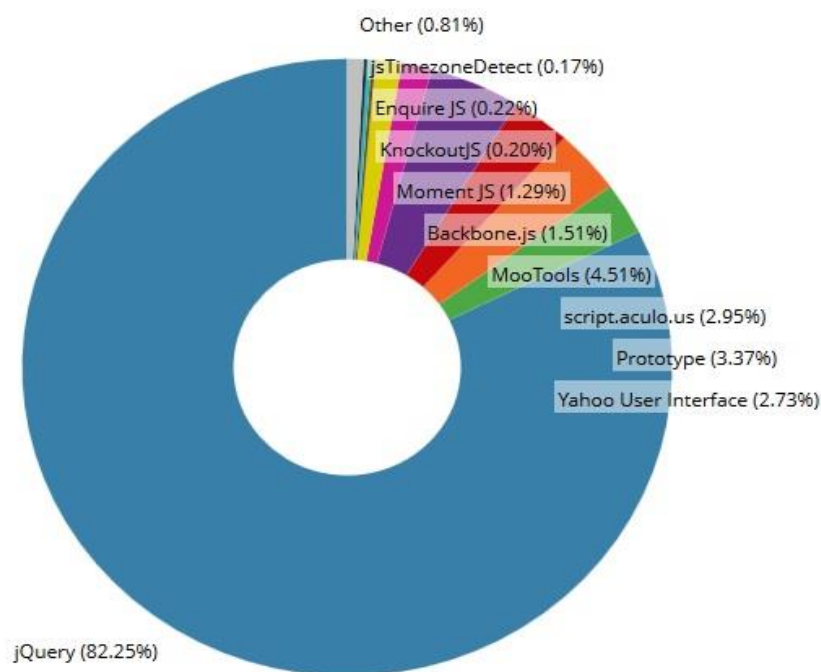


Figura 17: Estadísticas del uso de librerías JavaScript (BuiltWith Pty Ltd, 2015).

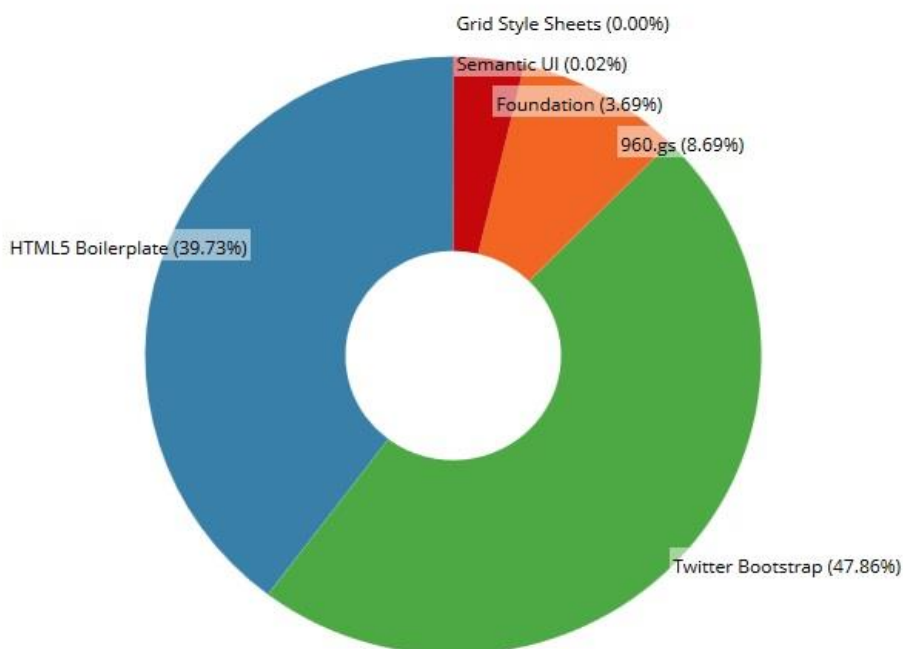


Figura 18: Estadísticas del uso de los *framework* de diseño web (BuiltWith Pty Ltd, 2015).

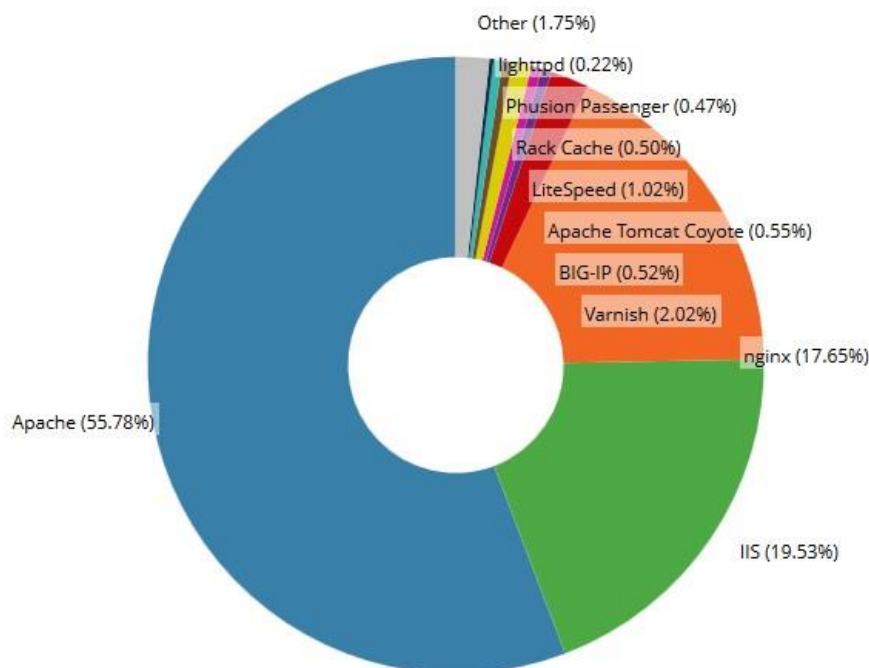


Figura 19: Estadísticas para sitios web usando servidores web (BuiltWith Pty Ltd, 2014).

Anexo 2: Historias de usuario

Tabla 41 HU1-Gestionar planificación de la cuarterería.

| | | | |
|---|--|---|--|
| Número: 1 | | Nombre del requisito: Gestionar planificación de la cuarterería. | |
| Programador: Michelle Pérez Acosta. | | Iteración Asignada: 1 | |
| Prioridad: Alta | | Tiempo Estimado: 3 días | |
| Riesgo en Desarrollo: Alto | | Tiempo Real: 1 semana | |
| Descripción: El sistema permitirá visualizar la planificación de la cuarterería, insertarla, modificarla y eliminarla. | | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | | | |

Tabla 42 HU2-Gestionar evaluación de cuarterería.

| | | | |
|--|--|---|--|
| Número: 2 | | Nombre del requisito: Gestionar evaluación de cuarterería. | |
| Programador: Michelle Pérez Acosta. | | Iteración Asignada: 1 | |
| Prioridad: Alta | | Tiempo Estimado: 3 días | |
| Riesgo en Desarrollo: Alto | | Tiempo Real: 1 semana | |
| Descripción: El sistema permitirá visualizar la evaluación de la cuarterería, insertarla, modificarla y eliminarla. | | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, | | | |

Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar.

Tabla 43 HU3-Gestionar evaluación de guardia estudiantil.

| | | |
|---|---|--|
| Número: 3 | Nombre del requisito: Gestionar evaluación de guardia estudiantil. | |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 1 | |
| Prioridad: Alta | Tiempo Estimado: 3 días | |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana | |
| Descripción: El sistema permitirá visualizar la evaluación de guardia estudiantil, insertarla, modificarla y eliminarla. | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | | |

Tabla 44 HU4-Gestionar evaluación del cumplimiento de normas.

| | | |
|---|---|--|
| Número: 4 | Nombre del requisito: Gestionar evaluación del cumplimiento de normas. | |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 1 | |
| Prioridad: Alta | Tiempo Estimado: 3 días | |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana | |
| Descripción: El sistema permitirá visualizar la evaluación del cumplimiento de normas, insertarla, modificarla y eliminarla. | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | | |

Tabla 45 HU6-Enviar correo al estudiante informando fecha.

| | | |
|---|--|--|
| Número: 6 | Nombre del requisito: Enviar correo al estudiante informando fecha. | |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 1 | |
| Prioridad: Media | Tiempo Estimado: 2 días | |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana | |
| Descripción: El sistema enviará un correo al estudiante informando la planificación en que le corresponde realizar la cuartería. | | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | | |

Tabla 46 HU7-Enviar correo al estudiante informando evaluación mensual.

| | | |
|--|---|--|
| Número: 7 | Nombre del requisito: Enviar correo al estudiante informando evaluación mensual. | |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 2 | |
| Prioridad: Media | Tiempo Estimado: 2 días | |
| Riesgo en Desarrollo: Medio | Tiempo Real: 1 semana | |

| |
|---|
| Descripción: El sistema enviará un correo al estudiante informando la evaluación mensual. |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. |

Tabla 46 HU8-Filtrar por evaluación para ver reincidentes.

| | |
|---|--|
| Número: 8 | Nombre del requisito: Filtrar por evaluación para ver reincidentes. |
| Programador: Michelle Pérez Acosta. | Iteración Asignada: 2 |
| Prioridad: Media | Tiempo Estimado: 5 días |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá filtrar las evaluaciones de los estudiantes. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | |

Tabla 48 HU9-Filtrar por edificios y por pasos de escaleras para ver cuartería.

| | |
|---|---|
| Número: 9 | Nombre del requisito: Filtrar por edificios y por pasos de escaleras para ver cuartería. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 2 |
| Prioridad: Media | Tiempo Estimado: 5 días |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá filtrar los datos por edificios y pasos de escaleras para ver los cuartereros del día seleccionado. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | |

Tabla 49 HU11-Filtrar por día para ver cuartería.

| | |
|---|--|
| Número: 11 | Nombre del requisito: Filtrar por día para ver cuartería. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 2 |
| Prioridad: Media | Tiempo Estimado: 3 días |
| Riesgo en Desarrollo: Alto | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá filtrar por día para ver los estudiantes que tienen cuartería el día determinado. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista General o Técnico B y el sistema le indicará los campos que son obligatorios llenar. | |

Tabla 50 HU14-Registrar permanencia del estudiante.

| | |
|--|--|
| Número: 14 | Nombre del requisito: Registrar permanencia del estudiante. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 3 |

| | |
|--|-------------------------------|
| Prioridad: Baja | Tiempo Estimado: 2 día |
| Riesgo en Desarrollo: Medio | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá registrar el estado de permanencia de cada estudiante en la universidad. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista Superior y el sistema le indicará los campos que son obligatorios llenar. | |

Tabla 51 HU15-Resgistrar denuncia.

| | |
|--|--|
| Número: 15 | Nombre del requisito: Registrar denuncia. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 3 |
| Prioridad: Media | Tiempo Estimado: 2 días |
| Riesgo en Desarrollo: Medio | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá registrar una denuncia realizada por un estudiante. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles Director General, Especialista Superior y el sistema le indicará los campos que son obligatorios llenar. | |

Tabla 52 HU16-Exportar informes a PDF.

| | |
|--|---|
| Número: 16 | Nombre del requisito: Exportar informes a PDF. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 3 |
| Prioridad: Baja | Tiempo Estimado: 2 días |
| Riesgo en Desarrollo: Bajo | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá exportar los informes en formato PDF. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles definidos. | |

Tabla 53 HU17-Enviar informes.

| | |
|--|---|
| Número: 17 | Nombre del requisito: Enviar informes. |
| Programador: Claudia Milagros de León Martínez. | Iteración Asignada: 3 |
| Prioridad: Baja | Tiempo Estimado: 2 días |
| Riesgo en Desarrollo: Bajo | Tiempo Real: 1 semana |
| Descripción: El sistema permitirá enviar por correo electrónico la información que se gestiona en el sistema. | |
| Observaciones: El usuario deberá ser autenticado con alguno de los roles definidos. | |

Anexo 3: Tareas de ingeniería

Tabla 54 Tarea#1.

| Tarea | |
|--|---------------------------|
| Número: 1 | Número de HU: 1 |
| Nombre: Implementar la visualización de la planificación de la cuartería. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |

| | |
|---|---|
| Fecha inicio: 9 de febrero de 2015 | Fecha fin: 11 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá mostrar la planificación de la cuarterería. | |

Tabla 55 Tarea#2.

| Tarea | |
|---|---|
| Número: 2 | Número de HU: 1 |
| Nombre: Implementar la Inserción de la planificación de la cuarterería. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 9 de febrero de 2015 | Fecha fin: 11 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir insertar la planificación de la cuarterería. | |

Tabla 56 Tarea#3.

| Tarea | |
|--|---|
| Número: 3 | Número de HU: 1 |
| Nombre: Implementar la modificación de la planificación de la cuarterería. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 9 de febrero de 2015 | Fecha fin: 11 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir modificar la planificación de la cuarterería. | |

Tabla 57 Tarea#4.

| Tarea | |
|---|---|
| Número: 4 | Número de HU: 1 |
| Nombre: Implementar la eliminación de la planificación de la cuarterería. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 9 de febrero de 2015 | Fecha fin: 11 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir eliminar la planificación de la cuarterería. | |

Tabla 58 Tarea#5.

| Tarea | |
|---|---|
| Número: 5 | Número de HU: 2 |
| Nombre: Implementar la visualización de la evaluación de la cuarterería. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 12 de febrero de 2015 | Fecha fin: 16 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá mostrar la evaluación de la cuarterería. | |

Tabla 59 Tarea#6.

| Tarea | |
|---|---|
| Número: 6 | Número de HU: 2 |
| Nombre: Implementar la inserción de la evaluación de la cuarterería. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 12 de febrero de 2015 | Fecha fin: 16 de febrero de 2015 |

| |
|--|
| Programador responsable: Michelle Pérez Acosta. |
| Descripción: El sistema deberá permitir insertar la evaluación de la cuartería. |

Tabla 60 Tarea#7.

| Tarea | |
|---|---|
| Número: 7 | Número de HU: 2 |
| Nombre: Implementar la modificación de la evaluación de la cuartería. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 12 de febrero de 2015 | Fecha fin: 16 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir modificar la evaluación de la cuartería. | |

Tabla 61 Tarea#8.

| Tarea | |
|--|---|
| Número: 8 | Número de HU: 2 |
| Nombre: Implementar la eliminación de la evaluación de la cuartería. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 12 de febrero de 2015 | Fecha fin: 16 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá permitir eliminar la evaluación de la cuartería. | |

Tabla 62 Tarea#9.

| Tarea | |
|---|---|
| Número: 9 | Número de HU: 3 |
| Nombre: Implementar la visualización de la evaluación de la guardia estudiantil. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 17 de febrero de 2015 | Fecha fin: 20 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Visualizar evaluación de la guardia estudiantil en el sistema. | |

Tabla 63 Tarea#10.

| Tarea | |
|---|---|
| Número: 10 | Número de HU: 3 |
| Nombre: Implementar la inserción de la evaluación de la guardia estudiantil. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 17 de febrero de 2015 | Fecha fin: 20 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Insertar evaluación de la guardia estudiantil en el sistema. | |

Tabla 64 Tarea#11.

| Tarea | |
|--|---|
| Número: 11 | Número de HU: 3 |
| Nombre: Implementar la modificación de la evaluación de la guardia estudiantil. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 17 de febrero de 2015 | Fecha fin: 20 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |

Descripción: Modificar evaluación de la guardia estudiantil en el sistema.

Tabla 65 Tarea#12.

| Tarea | |
|---|---|
| Número: 12 | Número de HU: 3 |
| Nombre: Implementar la eliminación de la evaluación de la guardia estudiantil. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 17 de febrero de 2015 | Fecha fin: 20 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Eliminar evaluación de la guardia estudiantil del sistema. | |

Tabla 66 Tarea#13.

| Tarea | |
|--|---|
| Número: 13 | Número de HU: 4 |
| Nombre: Implementar la visualización de la evaluación del cumplimiento de las normas. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 23 de febrero de 2015 | Fecha fin: 25 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Visualizar evaluación del cumplimiento de las normas en el sistema. | |

Tabla 67 Tarea#14.

| Tarea | |
|--|---|
| Número: 14 | Número de HU: 4 |
| Nombre: Implementar la inserción de la evaluación del cumplimiento de las normas. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 23 de febrero de 2015 | Fecha fin: 25 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Insertar evaluación del cumplimiento de las normas en el sistema. | |

Tabla 68 Tarea#15.

| Tarea | |
|---|---|
| Número: 15 | Número de HU: 4 |
| Nombre: Implementar la modificación de la evaluación del cumplimiento de las normas. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 23 de febrero de 2015 | Fecha fin: 25 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: Modificar evaluación del cumplimiento de las normas en el sistema. | |

Tabla 69 Tarea#16.

| Tarea | |
|--|---|
| Número: 16 | Número de HU: 4 |
| Nombre: Implementar la eliminación de la evaluación del cumplimiento de las normas. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 23 de febrero de 2015 | Fecha fin: 25 de febrero de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |

Descripción: Eliminar evaluación del cumplimiento de las normas del sistema.

Tabla 70 Tarea#21.

| Tarea | |
|--|--------------------------------------|
| Número: 21 | Número de HU: 6 |
| Nombre: Implementar el envío por correo al estudiante informando fecha. | |
| Tipo de tarea: Desarrollo | Estimación: 4 días |
| Fecha inicio: 3 de marzo de 2015 | Fecha fin: 6 de marzo de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá enviar un correo al estudiante informando la fecha planificada para realizar su cuartería. | |

Tabla 71 Tarea#22.

| Tarea | |
|--|---------------------------------------|
| Número: 22 | Número de HU: 7 |
| Nombre: Implementar el envío por correo al estudiante informando evaluación. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 10 de marzo de 2015 | Fecha fin: 11 de marzo de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá enviar un correo al estudiante informando su evaluación general del mes. | |

Tabla 72 Tarea#23.

| Tarea | |
|--|---------------------------------------|
| Número: 23 | Número de HU: 8 |
| Nombre: Implementar el filtrar por evaluación para ver los estudiantes reincidentes. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 12 de marzo de 2015 | Fecha fin: 14 de marzo de 2015 |
| Programador responsable: Michelle Pérez Acosta. | |
| Descripción: El sistema deberá contener un buscador que permita filtrar por meses y evaluación para ver los estudiantes reincidentes. | |

Tabla 73 Tarea#24.

| Tarea | |
|---|---------------------------------------|
| Número: 24 | Número de HU: 9 |
| Nombre: Implementar el filtrar por edificio y por paso de escalera para ver cuartería. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 16 de marzo de 2015 | Fecha fin: 18 de marzo de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá contener un buscador que permita filtrar por edificios y por pasos de escaleras para conocer la fecha de la cuartería de cada estudiante. | |

Tabla 74 Tarea#26.

| Tarea | |
|---|-------------------------|
| Número: 26 | Número de HU: 11 |
| Nombre: Implementar el filtrar por día para ver cuartería. | |

| | |
|--|---------------------------------------|
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 23 de marzo de 2015 | Fecha fin: 25 de marzo de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá contener un buscador que permita filtrar por días para ver las cuarterías. | |

Tabla 75 Tarea#33.

| Tarea | |
|--|--------------------------------------|
| Número: 33 | Número de HU: 14 |
| Nombre: Implementar la visualización de las bajas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 2 de abril de 2015 | Fecha fin: 4 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá mostrar las bajas por facultad. | |

Tabla 76 Tarea#34.

| Tarea | |
|---|--------------------------------------|
| Número: 34 | Número de HU: 14 |
| Nombre: Implementar la inserción las bajas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 2 de abril de 2015 | Fecha fin: 4 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir insertar las bajas por facultad. | |

Tabla 77 Tarea#35.

| Tarea | |
|--|--------------------------------------|
| Número: 35 | Número de HU: 14 |
| Nombre: Implementar la modificación de las bajas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 2 de abril de 2015 | Fecha fin: 4 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir modificar las bajas por facultad. | |

Tabla 78 Tarea#36.

| Tarea | |
|---|--------------------------------------|
| Número: 36 | Número de HU: 14 |
| Nombre: Implementar la eliminación de las bajas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 3 días |
| Fecha inicio: 2 de abril de 2015 | Fecha fin: 4 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir eliminar las bajas por facultad. | |

Tabla 79 Tarea#37.

| Tarea | |
|-------------------|-------------------------|
| Número: 37 | Número de HU: 15 |

| | |
|--|--------------------------------------|
| Nombre: Implementar la visualización de las altas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 6 de abril de 2015 | Fecha fin: 7 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá mostrar las altas por facultad. | |

Tabla 80 Tarea#38.

| Tarea | |
|---|--------------------------------------|
| Número: 38 | Número de HU: 15 |
| Nombre: Implementar la inserción de las altas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 6 de abril de 2015 | Fecha fin: 7 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir insertar las altas por facultad. | |

Tabla 81 Tarea#39.

| Tarea | |
|--|--------------------------------------|
| Número: 39 | Número de HU: 15 |
| Nombre: Implementar la modificación de las altas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 6 de abril de 2015 | Fecha fin: 7 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir modificar las altas por facultad. | |

Tabla 82 Tarea#40.

| Tarea | |
|---|--------------------------------------|
| Número: 40 | Número de HU: 15 |
| Nombre: Implementar la eliminación de las altas por facultad. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 6 de abril de 2015 | Fecha fin: 7 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir eliminar las altas por facultad. | |

Tabla 83 Tarea#41.

| Tarea | |
|---|--------------------------------------|
| Número: 41 | Número de HU: 16 |
| Nombre: Implementar la exportación de informes a PDF. | |
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 8 de abril de 2015 | Fecha fin: 9 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir exportar informes a PDF. | |

Tabla 84 Tarea#42.

| Tarea | |
|---|-------------------------|
| Número: 42 | Número de HU: 17 |
| Nombre: Implementar el envío de informes por correo. | |

| | |
|---|---------------------------------------|
| Tipo de tarea: Desarrollo | Estimación: 2 días |
| Fecha inicio: 10 de abril de 2015 | Fecha fin: 11 de abril de 2015 |
| Programador responsable: Claudia Milagros de León Martínez. | |
| Descripción: El sistema deberá permitir exportar informes a PDF. | |

Anexo 4: Casos de pruebas de aceptación

Tabla 85 Caso de prueba de aceptación #1.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU1_CP1 | Historia de usuario: 1 |
| Nombre: Visualizar planificación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario debe adicionar una cuartería y después tiene la opción de consultarlas. Pasos de ejecución: Para visualizar la planificación de la cuartería se debe ir a Vistas, Cuartería, donde se muestran todas las cuarterías registradas en el sistema. | |
| Resultado esperado: Se muestra la planificación de la cuartería que está en el sistema. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 86 Caso de prueba de aceptación #2.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU1_CP2 | Historia de usuario: 1 |
| Nombre: Insertar planificación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario debe adicionar una cuartería y después tiene la opción de consultarlas. Pasos de ejecución: Para visualizar la planificación de la cuartería se debe ir a Módulo Estudiante, Adicionar cuartería, una vez adicionada se guarda. | |
| Resultado esperado: Se muestra la planificación de la cuartería que está en el sistema. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 87 Caso de prueba de aceptación #3.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU1_CP3 | Historia de usuario: 1 |
| Nombre: Modificar planificación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con los roles Especialista General, Técnico B o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar en la planificación deseada. Pasos de ejecución: Para modificar la planificación de la cuartería se debe ir a Vistas, Cuartería, editar y luego Guardar. | |
| Resultado esperado: El sistema almacena los nuevos datos insertados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 88 Caso de prueba de aceptación #4.

| Caso de prueba de aceptación | |
|------------------------------|-------------------------------|
| Código: HU1_CP4 | Historia de usuario: 1 |

| |
|---|
| Nombre: Eliminar planificación de la cuartería. |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con los roles Especialista General, Técnico B o Director General. |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar señalando la planificación de la cuartería ya realizada. Pasos de ejecución: Para eliminar la planificación de la cuartería se debe ir a Vistas, Cuartería, eliminar y luego confirmar. |
| Resultado esperado: El sistema elimina los datos seleccionados. |
| Evaluación de la prueba: Prueba satisfactoria |

Tabla 89 Caso de prueba de aceptación #5.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU2_CP5 | Historia de usuario: 2 |
| Nombre: Visualizar evaluación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la pestaña donde se muestran las evaluaciones de la cuartería por estudiante. Pasos de ejecución: Para visualizar la evaluación de la cuartería se debe ir a Vistas, Evaluación y allí se muestra la evaluación. | |
| Resultado esperado: El sistema muestra las evaluaciones por estudiante registradas. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 90 Caso de prueba de aceptación #6.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU2_CP6 | Historia de usuario: 2 |
| Nombre: Insertar evaluación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Insertar evaluación. Pasos de ejecución: Para insertar la evaluación de la cuartería se debe ir a Vistas, Cuartería, luego se da clic encima del título que se le puso y se va a la pestaña de editar. Una vez en ella se le agrega la evaluación y finalmente se guarda. | |
| Resultado esperado: El sistema muestra las evaluaciones por estudiante registradas en la aplicación. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 91 Caso de prueba de aceptación #7.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU2_CP7 | Historia de usuario: 2 |
| Nombre: Modificar evaluación de la cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar en la evaluación deseada. Pasos de ejecución: Para modificar la evaluación de la cuartería se debe ir a Vistas, | |

Cuartelería, luego damos clic encima del título que se le puso y se va a la pestaña de editar. Una vez en ella se le modifica la evaluación y finalmente se guarda.

Resultado esperado: El sistema almacena la nueva evaluación modificada.

Evaluación de la prueba: Prueba satisfactoria

Tabla 92 Caso de prueba de aceptación #8.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU2_CP8 | Historia de usuario: 2 |
| Nombre: Eliminar evaluación de la cuarterería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Ninguna en la evaluación deseada. Pasos de ejecución: Para eliminar la evaluación de la cuarterería se debe ir a Vistas, Cuarterería, luego damos clic encima del título que se le puso y se va a la pestaña de editar. Una vez en ella se le modifica la evaluación a Ninguna y finalmente se guarda. | |
| Resultado esperado: El sistema elimina la evaluación seleccionada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 93 Caso de prueba de aceptación #9.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU3_CP9 | Historia de usuario: 3 |
| Nombre: Visualizar evaluación de la guardia estudiantil. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la pestaña donde se visualizan todas las evaluaciones de las guardias estudiantiles registradas en el sistema. Pasos de ejecución: Para visualizar la evaluación de la guardia estudiantil se debe ir a Vistas, Estudiante, luego damos clic encima del Nombre y Apellidos del estudiante. Una vez que se muestren los datos se da clic encima de la evaluación de la guardia. | |
| Resultado esperado: El sistema muestra todas las evaluaciones registradas. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 94 Caso de prueba de aceptación #10.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU3_CP10 | Historia de usuario: 3 |
| Nombre: Insertar evaluación de la guardia estudiantil. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al módulo Estudiante y escoge la opción Evaluar Guardia. Pasos de ejecución: Para insertar la evaluación de la guardia estudiantil se debe ir al Módulo Estudiante, Evaluar Guardia, una vez allí se introducen los datos del estudiante y se le da la evaluación de la guardia y finalmente se guarda. | |
| Resultado esperado: El sistema almacena la evaluación introducida por el usuario. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 95 Caso de prueba de aceptación #11.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU3_CP11 | Historia de usuario: 3 |
| Nombre: Modificar evaluación de la guardia estudiantil. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar en la evaluación deseada. Pasos de ejecución: Para modificar la evaluación de la guardia estudiantil se debe ir a Vistas, Estudiante, clic encima del Nombre y Apellidos del estudiante. Una vez allí se da clic encima de la evaluación de la guardia y se edita, finalmente se guarda. | |
| Resultado esperado: El sistema almacena la nueva evaluación introducida por el usuario. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 96 Caso de prueba de aceptación #12.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU3_CP12 | Historia de usuario: 3 |
| Nombre: Eliminar evaluación de la guardia estudiantil. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar en la evaluación deseada. Pasos de ejecución: Para eliminar la evaluación de la cuarterería se debe ir a Vistas, Estudiante, clic encima del Nombre y Apellidos del estudiante, clic encima de la evaluación de la guardia que se desea eliminar, luego presionar el botón Eliminar y confirmar. | |
| Resultado esperado: El sistema elimina la evaluación seleccionada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 97 Caso de prueba de aceptación #13.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU4_CP13 | Historia de usuario: 4 |
| Nombre: Visualizar evaluación del cumplimiento de las normas. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la pestaña donde se muestran las evaluaciones del cumplimiento de las normas por estudiante. Pasos de ejecución: Para visualizar la evaluación del cumplimiento de normas se debe ir a Vistas, Estudiante, se da clic encima del Nombre y Apellidos del estudiante, clic encima de la evaluación de las normas y se muestra la evaluación. | |
| Resultado esperado: El sistema muestra las evaluaciones por estudiante registradas. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 98 Caso de prueba de aceptación #14.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU4_CP14 | Historia de usuario: 4 |
| Nombre: Insertar evaluación del cumplimiento de las normas. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de | |

| |
|---|
| los roles Técnico B, Especialista General o Director General. |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante y escoge la opción Evaluar Normas, especificando el estudiante y las causas. Pasos de ejecución: Para insertar la evaluación del cumplimiento de normas se debe ir a Módulo Estudiante, Evaluar Normas, llenar los campos y finalmente guardar. |
| Resultado esperado: El sistema almacena la evaluación registrada. |
| Evaluación de la prueba: Prueba satisfactoria |

Tabla 99 Caso de prueba de aceptación #15.

| Caso de prueba de aceptación | |
|---|-------------------------------|
| Código: HU4_CP15 | Historia de usuario: 4 |
| Nombre: Modificar evaluación del cumplimiento de las normas. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar en la evaluación deseada. Pasos de ejecución: Para modificar la evaluación del cumplimiento de normas se debe ir a Vistas, Estudiante, clic encima de Nombre y Apellidos, clic encima de la evaluación del cumplimiento de normas, ir a la pestaña editar y finalmente guardar. | |
| Resultado esperado: El sistema almacena la nueva evaluación registrada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 100 Caso de prueba de aceptación #16.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU4_CP16 | Historia de usuario: 4 |
| Nombre: Eliminar evaluación del cumplimiento de las normas. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar en la evaluación deseada. Pasos de ejecución: Para eliminar la evaluación del cumplimiento de normas se debe ir a Vistas, Estudiante, clic encima de Nombre y Apellidos, clic encima de la evaluación del cumplimiento de normas, ir a la pestaña editar, presionar el botón Eliminar, finalmente confirmar. | |
| Resultado esperado: El sistema elimina la evaluación seleccionada. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 101 Caso de prueba de aceptación #21.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU6_CP21 | Historia de usuario: 6 |
| Nombre: Enviar correo al estudiante informando fecha. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante, adiciona una cuartería y la confirma. Pasos de ejecución: Para enviar correo al estudiante informando fecha se debe ir a Módulo Estudiante, Adicionar Cuartería, llenar los campos y guardar. | |

Resultado esperado: El sistema envía automáticamente al estudiante un correo una vez que esta sea planificada informando la fecha en que le corresponde la cuartería y los objetivos que debe cumplir.

Evaluación de la prueba: Prueba satisfactoria

Tabla 102 Caso de prueba de aceptación #22.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU7_CP22 | Historia de usuario: 7 |
| Nombre: Enviar correo al estudiante informando evaluación. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante y registra la evaluación mensual del estudiante. Pasos de ejecución: Para enviar correo al estudiante informando evaluación se debe ir a Módulo Estudiante, Crear evaluación mensual, llenar los campos y guardar. | |
| Resultado esperado: El sistema envía un correo automáticamente al estudiante informando la evaluación mensual y las causas. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 103 Caso de prueba de aceptación #23.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU8_CP23 | Historia de usuario: 8 |
| Nombre: Filtrar por evaluación para ver los estudiantes reincidentes. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y realiza una búsqueda en el sistema para ver la evaluación del estudiante. Pasos de ejecución: Para filtrar por mes y evaluación para ver reincidentes se debe ir a Vistas, Evaluación y una vez allí se filtra la búsqueda por el criterio que se desee. | |
| Resultado esperado: El sistema muestra los estudiantes que coinciden con los campos especificados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 104 Caso de prueba de aceptación #24.

| Caso de prueba de aceptación | |
|--|-------------------------------|
| Código: HU9_CP24 | Historia de usuario: 9 |
| Nombre: Filtrar por edificio y por paso de escalera para ver cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Técnico B, Especialista General o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y realiza una búsqueda en el sistema para ver la cuartería especificando los campos. Pasos de ejecución: Para filtrar por edificio y por paso de escalera para ver cuartería se debe ir a Vistas, Cuartería, una vez allí se llenan los campos que se deseen para filtrar la búsqueda. | |
| Resultado esperado: El sistema muestra los estudiantes que coinciden con los campos especificados. | |

Evaluación de la prueba: Prueba satisfactoria

Tabla 105 Caso de prueba de aceptación #26.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU11_CP26 | Historia de usuario: 11 |
| Nombre: Filtrar por día para ver cuartería. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles definidos. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y realiza una búsqueda en el sistema para ver la cuartería especificando los días. Pasos de ejecución: Para filtrar por día para ver cuartería se debe ir a Vistas, Cuartería por día y se señala la fecha deseada. | |
| Resultado esperado: El sistema muestra la cuartería coincidente con los datos entrados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 106 Caso de prueba de aceptación #33.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU14_CP33 | Historia de usuario: 14 |
| Nombre: Visualizar las bajas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Estado de permanencia. Pasos de ejecución: Para visualizar las bajas por facultad se debe ir a Vistas, Estado de permanencia y se selecciona la opción Baja. | |
| Resultado esperado: El sistema muestra la relación de bajas por facultad. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 107 Caso de prueba de aceptación #34.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU14_CP34 | Historia de usuario: 14 |
| Nombre: Insertar las bajas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante y escoge la opción Registrar permanencia. Pasos de ejecución: Para insertar las bajas por facultad se debe ir a Módulo Estudiante, Registrar permanencia y una vez que se llenen estos datos se guarda. | |
| Resultado esperado: El sistema almacena los datos registrados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 108 Caso de prueba de aceptación #35.

| Caso de prueba de aceptación | |
|--|--------------------------------|
| Código: HU14_CP35 | Historia de usuario: 14 |
| Nombre: Modificar las bajas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |

| |
|--|
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Estado de permanencia. Pasos de ejecución: Para modificar las bajas por facultad se debe ir a Vistas, Estado de permanencia, se seleccionan las bajas, luego se da clic sobre la facultad a la que pertenece el estudiante, se modifican los datos y finalmente se guardan. |
| Resultado esperado: El sistema almacena los nuevos datos registrados. |
| Evaluación de la prueba: Prueba satisfactoria |

Tabla 109 Caso de prueba de aceptación #36.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU14_CP36 | Historia de usuario: 14 |
| Nombre: Eliminar las bajas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar. Pasos de ejecución: Para eliminar las bajas por facultad se debe ir a Vistas, Estado de permanencia, se seleccionan las bajas, luego se da clic sobre la facultad a la que pertenece el estudiante, se presiona el botón Eliminar y se confirma. | |
| Resultado esperado: El sistema elimina los datos seleccionados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 110 Caso de prueba de aceptación #37.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU15_CP37 | Historia de usuario: 15 |
| Nombre: Visualizar las altas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Estado de permanencia. Pasos de ejecución: Para visualizar las altas por facultad se debe ir a Vistas, Estado de permanencia y se selecciona la opción Alta. | |
| Resultado esperado: El sistema muestra la relación de altas. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 111 Caso de prueba de aceptación #38.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU15_CP38 | Historia de usuario: 15 |
| Nombre: Insertar las altas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Estudiante y escoge la opción Registrar permanencia. Pasos de ejecución: Para insertar las altas por facultad se debe ir a Módulo Estudiante, Registrar permanencia y una vez que se llenen estos datos se guarda. | |
| Resultado esperado: El sistema almacena los datos insertados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 112 Caso de prueba de aceptación #39.

| Caso de prueba de aceptación | |
|------------------------------|--|
|------------------------------|--|

| | |
|---|--------------------------------|
| Código: HU15_CP39 | Historia de usuario: 15 |
| Nombre: Modificar las altas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Editar. Pasos de ejecución: Para modificar las altas por facultad se debe ir a Vistas, Estado de permanencia, se seleccionan las altas, luego se da clic sobre la facultad a la que pertenece el estudiante, se modifican los datos y finalmente se guardan. | |
| Resultado esperado: El sistema almacena los nuevos datos registrados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 113 Caso de prueba de aceptación #40.

| Caso de prueba de aceptación | |
|---|--------------------------------|
| Código: HU15_CP40 | Historia de usuario: 15 |
| Nombre: Eliminar las altas por facultad. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede al Módulo Vistas y escoge la opción Eliminar. Pasos de ejecución: Para eliminar las altas por facultad se debe ir a Vistas, Estado de permanencia, se seleccionan las altas, luego se da clic sobre la facultad a la que pertenece el estudiante, se presiona el botón Eliminar y se confirma. | |
| Resultado esperado: El elimina los datos seleccionados. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 114 Caso de prueba de aceptación #41.

| Caso de prueba de aceptación | |
|--|--------------------------------|
| Código: HU16_CP41 | Historia de usuario: 16 |
| Nombre: Exportar informes a PDF. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede a cualquiera de las opciones que desee. Pasos de ejecución: Para exportar alguna información a PDF el usuario debe seleccionar el informe que desea exportar y luego dar clic encima del ícono que exporta a PDF que siempre está en la página. | |
| Resultado esperado: Se exporta a PDF el informe seleccionado. | |
| Evaluación de la prueba: Prueba satisfactoria | |

Tabla 115 Caso de prueba de aceptación #42.

| Caso de prueba de aceptación | |
|--|--------------------------------|
| Código: HU17_CP42 | Historia de usuario: 17 |
| Nombre: Exportar informes a PDF. | |
| Condiciones de ejecución: El usuario debe estar autenticado en el sistema con alguno de los roles Especialista Superior o Director General. | |
| Entrada/ Pasos de ejecución: Entrada: El usuario accede a cualquiera de las opciones que desee. Pasos de ejecución: Para enviar por correo alguna información el usuario debe seleccionar | |

el informe que desea enviar y luego dar clic encima del ícono que envía por correo que siempre está en la página.

Resultado esperado: Se envía por correo el informe seleccionado.

Evaluación de la prueba: Prueba satisfactoria