



**Facultad 1**

**Sistema de gestión de información de estudiantes  
extranjeros en la Universidad de las Ciencias  
Informáticas**

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

Autor: Tonito Días Sumbelelo  
Tutores: MSc. Roberkis Terrero Galano  
Ing. Alexander Rodríguez Mompíe  
Ing. Leyriel Zurita González

**La Habana, 2020**

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2020.

Tonito Días Sumbelelo

---

Firma del Autor

---

Firma del Tutor

---

Firma del Tutor

## AGRADECIMIENTOS

A Dios (Jehová) quiero darle las gracias por la vida que me ha dado, su amor, fidelidad y la oportunidad de estudiar en la Universidad de las Ciencias Informáticas en Cuba.

A mis padres y abuela: Belo Elavo Sumbelelo, Filomena Nassinda y Isabel Tchissenda.

A mis queridos tíos y tías: Aurora Jiloi Paulo y Domingas Paulo Sumbelelo, que además de tíos son como padres y madres para mi vida, porque son parte de esta historia, por apoyarme todo el tiempo en mi formación académica y como hombre. A mi querida novia Manilsa y a toda la familia Sumbelelo y Ezequiel les doy las gracias por todo porque de una u otra forma se han preocupado muchísimo en verme formado.

Agradezco también a mis hermanos en la fe, de la UCI, han sido una bendición para mi vida son parte de este sueño como ingeniero, porque me han acogido con los dos brazos abiertos, como un miembro más de sus familias, jóvenes comprometidos y ejemplar dignos de imitar, fue un placer conocerlos les llevo en mi corazón. Gracias por todo hermanos.

Agradezco a todos mis queridos amigos colegas Angolanos que también son como mi familia, hemos compartidos tiempos buenos y aprendimos muchas cosas juntos, son también parte de mi vida, les doy las gracias por ayudarme en todo lo que necesitaba. A mis compañeros cubanos colegas de aulas les doy gracias por conocerles y compartir con ustedes, aprendí mucho sobre Cuba porque a cada día que pasaba me enseñaban cosas que era necesario saber cómo extranjeros, gracias a: Juan Pablo y Yudenis, David Mesa, David Heberto, Eduardo Joea, muchos otros.

Quiero agradecer también a todos los profesores de mi facultad la 1, los zorros y el personal directivo de la UCI que atienden los estudiantes extranjeros, por permitir estudiar en esa linda Universidad que fue mi casa durante cinco años, los instructores de residencia, nuestra mamita Daisy especialmente por cuidarnos todo el tiempo, más que instructora fue como madre.

Agradecer a los especialistas de la Dirección de Informatización (DIN), Néstor, Alejandro Calderón, Laritza, Roberkis, la profesora Mairelys que en poco tiempo me ha apoyado bastante en la revisión del documento.

A mis tutores, Alexander, Leyriel, Roberkis, que me han apoyado bastante para que esa tesis saliera con calidad, han quitado parte de su tiempo para apoyarme en todo, no hay palabras para describir lo importante que han sido para mí, muchas gracias tutores. A todos los que contribuyeron directa o indirectamente con esta tesis, les doy las gracias. Muchas Gracias.

**DEDICATORIA**

*A Dios, a mis padres y a mi familia.*

## RESUMEN

La Dirección de Relaciones Internacionales en la Universidad de las Ciencias (DRI-UCI) es el área encargada de centralizar la información que se gestiona de los estudiantes extranjeros que cursan estudios de pregrado en la Institución, y de confeccionar reportes relacionados con la misma. El trabajo de centralización, gestión de la información y confección de reportes en la Dirección, se realiza en su mayoría de forma manual, haciendo uso del correo electrónico y/o de algunas herramientas del paquete Office: como el Word y el Excel. Este proceder trae consigo demoras en dichas actividades, así como la posible introducción de errores en los consolidados que se realizan, es por ello, que para solventar los problemas existentes y para agilizar el trabajo en la DRI relacionado con esta rama; se decide desarrollar un sistema para la gestión de información de los estudiantes extranjeros de pregrado que estudian en la UCI. El sistema tendrá la capacidad de registrar toda la información del estudiante ya sean: datos personales, actividades extracurriculares en las que ha participado, salidas realizadas en tiempo vacacional, residencia estudiantil en la que vive, ingresos en caso de enfermedad, etc. Además, se determinó para guiar el desarrollo de la solución el uso de la metodología AUP-UCI, para la implementación el framework Symfony4 y el gestor de base de datos PostgreSQL 11.0.

**Palabras Claves:** DRI-UCI, Sistema de Gestión de Información, estudiantes extranjeros.

**ABSTRACT**

La Dirección de Relaciones Internacionales en la Universidad de las Ciencias (DRI-UCI) es el área encargada de centralizar la información que se gestiona de los estudiantes extranjeros que cursan estudios de pregrado en la Institución, y de confeccionar informes relacionados con la misma. El trabajo de centralización, gestión de la información y confección de informes en la Dirección, realización en su mayoría de forma manual, uso del correo electrónico y / o de algunas herramientas del paquete Oficina: como el Word y el Excel. Este procederá a obtener demoras en diversas actividades, así como la posible introducción de errores en los consolidados que se llevan a cabo, es por ello, que para solventar los problemas afectados y para agilizar el trabajo en el DRI relacionado con esta rama; se decide desarrollar un sistema para la gestión de información de los estudiantes extranjeros de pregrado que estudian en la UCI. The system will have the ability to record all the student's information, be it: personal data, extracurricular activities in which he / she has had, trips made during vacation time, student residence where he / she lives, income in case of illness, etc. In addition, the use of the AUP-UCI methodology was determined to guide the development of the solution, for the implementation of the Symfony4 framework and the PostgreSQL 11.0 database manager.

**Key Words:** DRI-UCI, Information Management System, Foreign students.

**INDICE**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 1

1.1 Introducción ..... 1

1.2 Descripción de la gestión de información de estudiantes extranjeros de pregrado en la UCI. .... 1

1.3 Estudio del estado del arte..... 3

1.4. Metodología de desarrollo de software..... 7

1.5 Tecnologías para el desarrollo de la solución ..... 9

1.6 Entorno de Desarrollo Integrado (IDE) ..... 15

1.7 Conclusiones del capítulo ..... 15

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN..... 17

2.1 Introducción ..... 17

2.2 Propuesta de solución..... 17

2.3 Modelo conceptual ..... 18

2.4 Requisitos del software ..... 23

2.5 Modelo de caso de uso del sistema ..... 31

2.6 Modelo de datos ..... 37

2.7 Modelo de diseño..... 39

2.8 Patrones de diseño ..... 43

2.9 Conclusiones del capítulo ..... 47

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN ..... 49

3.1 Implementación..... 49

3.2 Solución del Sistema de gestión de estudiantes extranjeros en la UCI. .... 53

3.3 Pruebas de software ..... 55

3.4 Conclusiones del capítulo ..... 63

CONCLUSIONES GENERALES..... 64

RECOMENDACIONES ..... 65

REFERENCIAS BIBLIOGRÁFICAS ..... 66

ANEXOS..... 70

Anexo 1: Diseño de entrevista ..... 70

Anexo 2: Descripción de conceptos del Modelo conceptual (MC). .... 71

Anexo 3: Diagrama de clases del diseño ..... 75

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Escenarios de la metodología AUP-UCI. ....	8
<b>Tabla 2:</b> Descripción del concepto Estudiante. ....	19
<b>Tabla 3:</b> Descripción de requisitos funcionales. ....	24
<b>Tabla 4:</b> Resumen de la cantidad de requisitos funcionales según complejidad y prioridad. ....	29
<b>Tabla 5:</b> Descripción de los actores del sistema. ....	31
<b>Tabla 6:</b> Descripción del CU 13. Gestionar Estudiante. ....	33
<b>Tabla 7:</b> Estrategia de pruebas de software. ....	55
<b>Tabla 8:</b> Caso de prueba de unidad para el CU25. Autenticar Usuario.....	57
<b>Tabla 9:</b> Caso de prueba funcional para el CU1. Carrera. (parte 1).....	58
<b>Tabla 10:</b> Caso de prueba funcional para el CU1. Carrera. (parte 2).....	60
<b>Tabla 11:</b> Resultados del método de caja negra mediante la técnica partición equivalencia al CU1: Gestionar Carrera. ....	61
<b>Tabla 12:</b> Resultados de las pruebas de seguridad. Fuente: Elaboración propia.....	62

**ÍNDICE DE FIGURAS**

**Figura 1:** Modelo conceptual del negocio. Fuente: Elaboración del autor..... 18

**Figura 2:** Diagrama de caso de uso del sistema. Fuente: Elaboración del autor. .... 32

**Figura 3:** Modelo Entidad Relación. Fuente: Elaboración del autor. .... 38

**Figura 4:** Diagrama de clases del diseño CU13. Gestionar Estudiante. Fuente: Elaboración del autor 40

**Figura 5:** Diagrama de paquetes. Fuente: Elaboración del autor..... 41

**Figura 6:** Diagrama de despliegue. Fuente: Elaboración del autor. .... 42

**Figura 7:** Funcionamiento de una aplicación con MVC. Fuente: Elaboración del autor. .... 43

**Figura 8:** Patrón Experto. Fuente: Elaboración propia..... 45

**Figura 9:** Patrón Creador. Fuente: Elaboración propia. .... 45

**Figura 10:** Patrón Controlador. Fuente: Elaboración propia. .... 46

**Figura 11:** Patrón Controlador. Fuente: Elaboración propia. .... 47

**Figura 12:** Diagrama de componentes del CUS Gestionar Usuario. Fuente: Elaboración del autor. ... 50

**Figura 13:** Ejemplo de la clase y funcione index en el controlador. Fuente: Elaboración del autor. .... 51

**Figura 14:** Definición de variables. Fuente: Elaboración del autor. .... 51

**Figura 15:** Asignación de variables y llamadas a funciones. Fuente: Elaboración propia. .... 52

**Figura 16:** Estructura de control if else y for. Fuente: Elaboración propia..... 52

**Figura 17:** Interfaz principal (rol administrador). Fuente: Elaboración propia..... 53

**Figura 18:** Interfaz principal del Módulo Estudiante. Fuente: Elaboración propia..... 54

**Figura 19:** Resultados de las pruebas realizadas al software. Fuente. Elaboración propia. .... 62

**Figura 20:** Gráfico de errores detectados y resueltos en las pruebas de seguridad. Fuente. Elaboración propia..... 63

## INTRODUCCIÓN

La evolución de la sociedad actualmente se apoya en gran medida en el desarrollo acelerado de las nuevas tecnologías de la información y las comunicaciones (TICs), impulsadas por los adelantos de la informática, que avizoran un cambio de paradigma en todos los ámbitos de la vida en sociedad.

Este apoyo se fundamenta principalmente en las mejoras considerables que la evolución de las TICs brinda a cualquier rama social donde se aplique. Un ejemplo de ello se evidencia, en la evolución de los sistemas de gestión información que se han utilizado principalmente para el procesamiento electrónico de datos. Datos que por muchos años fueron gestionados en soporte de papel, como es el ejemplo de: los libros de cuentas, las fichas de empleados, los informes anuales, los historiales clínicos, etc. La aparición de la informática permitió la evolución al soporte digital de esta información en papel, dando lugar a los primeros sistemas de gestión de información. (Hernandez Trasobares, 2017)

Las aplicaciones informáticas enfocadas a la gestión de información tienen como función, auxiliar a las organizaciones en sus tareas; y permiten almacenar, procesar y brindar resúmenes de información en tiempo real, con el propósito de que las organizaciones puedan lograr sus objetivos a largo, mediano o corto plazo.

Algunos autores como (Peralta, 2008), definen los sistemas de gestión de información como: el conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo en cuenta, el equipo computacional necesario y la designación adecuada de los recursos humanos para que el sistema de información opere correctamente.

También (Peña Ayala, 2006) establece que: un sistema de gestión de información es un conjunto de elementos interrelacionados, con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones.

Estos sistemas son desarrollados en las empresas para contribuir con el desempeño de las actividades que se planifiquen. Por eso se pueden encontrar: sistemas de registros médicos en un hospital, sistemas de automatización de oficinas, sistemas de registros criminales, sistemas de pago de nóminas en todas las empresas, sistemas de inventarios en los mercados, sistemas de automatización de bibliotecas, entre otros. Todos estos sistemas tienen en común funciones que pueden ser agrupadas en: funciones de captación de recolección de datos, funciones de almacenamiento, tratamiento de la información y distribución o diseminación de la misma. (Hamidian Fernández, et al., 2015).

Se consideran soluciones tecnológicas para dar soporte a la gestión de la información en cualquier tipo de organización; por eso han sido visualizadas por muchas organizaciones como un enfoque que contribuye a la elevación de la eficacia y eficiencia organizacional, así como al perfeccionamiento de la toma de decisiones. Lo que ha tenido una presencia significativa en muchos países desarrollados, y en instituciones que han contado con líderes capaces de visualizar la importancia de contar con la información que necesitan en el momento oportuno. (Vázquez, y otros, 2015)

Actualmente, la mayor parte de entidades, tanto empresas como instituciones, basan la gestión de sus procesos en herramientas de software, aunque, en ciertos casos, el soporte digital no se ha adoptado completamente y se sigue utilizando el papel; en este caso es donde intervienen los procesos de informatización, para dar solución a la problemática de las organizaciones de ir reemplazando el papel por sistemas tecnológicos para apoyar la toma de decisión. (Paloma Torres, 2017)

Cuba no se ha quedado fuera de esos cambios tecnológicos, ya que existen en el País muchas instituciones que trabajan día tras día en el proceso de informatización de la sociedad cubana. La Universidad de las Ciencias Informáticas (UCI) es un ejemplo de este trabajo, reflejándose claramente en su misión de: Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la informática, producir aplicaciones y servicios informáticos a partir del vínculo estudio – trabajo – producción, sirviendo de soporte a la industria cubana de la informática. La UCI está claramente alineada con el proceso de informatización cubano, donde se hace indispensable la creación de sistemas de gestión de información que apoyen los procesos de las diferentes instituciones cubanas para que la sociedad pueda insertarse en la era de la información y el conocimiento tecnológico.

La UCI, cuenta con una infraestructura tecnológica, donde el uso eficiente de la misma constituye un reto diario. Por esta razón, una de las tareas fundamentales que se llevan a cabo es la informatización de los procesos que forman parte de la vida universitaria. La forma de concepción del Centro y su funcionamiento, vinculan estrechamente la producción de software y la formación docente de los estudiantes que allí cursan estudios. Este enlace hace de ella uno de los centros del País que más aporta a la informatización de la sociedad cubana y a la Universidad en sí.

Desde sus inicios en el 2002, la UCI ha mantenido a través de los Planes Anuales de Informatización un escalonado trabajo para ir informatizando sus procesos, donde la integración armónica de los fundamentos administrativos y tecnológicos permite avanzar en el camino de la excelencia universitaria. En la Universidad se han informatizado procesos como: la gestión académica de pregrado, la gestión académica de postgrado, procesos de la residencia estudiantil y de trabajadores, el control de acceso, la transportación de residentes en la Universidad, procesos de cooperación internacional, etc.; pero aún

quedan algunos que pueden complementar el trabajo de las áreas que los relacionan. En este caso se tiene a la Dirección de Relaciones Internacionales (DRI), donde se han realizado trabajos de informatización para los procesos del control de viajeros en funciones de trabajo, la gestión y control de las actividades de cooperación internacional con la UCI, pero van quedando otros como: la gestión de los estudiantes extranjeros que cursan estudios en ella.

Dado el incremento de estudiantes extranjeros de pregrado en el Centro, la DRI tiene la necesidad de contar con una mejor centralización y disponibilidad de la información que genera este suceso; además de contar con un mejor soporte para ella, que le permita realizar búsquedas y reportes de una forma más simple y rápida. Estos resúmenes de información deben ser confeccionados siempre con fiabilidad y rapidez para que las instancias interesadas puedan tomar las decisiones correctas con relación a los estudiantes.

La DRI es la encargada de registrar, controlar, centralizar y brindar al Rector de la Universidad la información de los estudiantes extranjeros que se necesite. Para obtener la información requerida de estos estudiantes, la DRI recibe información de diferentes áreas internas de la UCI, y que son las que atienden directamente a los estudiantes. Dentro de estas áreas se encuentran las facultades, la residencia y el centro de salud, las cuales envían la información que se les solicita en diferentes formatos y por diferentes vías, haciendo que la organización de la información para realizar un consolidado sea un proceso engorroso, provocando en algunos casos incoherencias y demora en la confección de reportes, lo que puede evidenciar un trabajo poco profesional en alguna medida.

Entre los principales problemas se encuentran:

- ✓ Aunque en la DRI se encuentra confeccionado un expediente para cada estudiante, mantenerlos actualizados se hace trabajoso, ya que la información de los estudiantes que se recibe de las áreas que los atienden en ocasiones relaciona a más. Esto hace que el especialista de la DRI tenga que confeccionar e imprimir un documento por cada estudiante, separando la información para poder actualizar el expediente de cada uno.
- ✓ No se informa oportunamente por parte de las facultades y la residencia de la participación de los estudiantes extranjeros en actividades extracurriculares a la DRI, lo que trae consigo que no se tengan las correspondientes evidencias en el momento de evaluar o caracterizar a un estudiante.

- ✓ No se tiene un control centralizado de las entradas y salidas de los estudiantes a la Universidad por motivos de viajes en tiempos vacacionales, ya que en ocasiones la información la tienen las facultades y en otros la DRI.
- ✓ Se dificulta en ocasiones conocer si algún estudiante extranjero ha estado hospitalizado y los motivos, debido a que los responsables por el área de salud o la facultad que los atiende no informan a la DRI del suceso.
- ✓ Al estar la información de los estudiantes dispersa en las diferentes áreas de la UCI, provoca que no esté disponible para todos los interesados y no se pueda brindar la atención requerida.
- ✓ No existe una oportuna actualización de la información de los estudiantes extranjeros, lo que dificulta la realización de los diferentes reportes que se solicitan a la DRI relacionada con ellos.

Dada la situación problemática detallada anteriormente se plantea entonces el siguiente **problema de investigación**: ¿Cómo contribuir con la gestión de información de estudiantes extranjeros de pregrado en la Universidad de las Ciencias Informáticas?

Se definió como **objeto de estudio**: la gestión de información de estudiantes extranjeros y específicamente en el **campo de acción**: la gestión de información de estudiantes extranjeros de pregrado en la Universidad de las Ciencias Informáticas.

Se define como **objetivo general**: Desarrollar un sistema informático para la gestión de información y la obtención de reportes de los estudiantes extranjeros de pregrado en la Universidad de las Ciencias Informáticas haciendo uso de herramientas libres.

Derivando del objetivo general, los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación mediante el estudio de los referentes teóricos de la gestión de información de estudiantes extranjeros basado en evidencias; valorar y seleccionar una metodología de desarrollo, así como las herramientas y tecnologías de software necesarias para el desarrollo de la solución informática.
- Realizar el análisis y diseño un sistema informático que posibilite la gestión de información de estudiantes extranjeros de pregrado en la Universidad de las Ciencias Informáticas.
- Implementar un sistema informático que posibilite la gestión de información de estudiantes extranjeros de pregrado en la Universidad de las Ciencias Informáticas, haciendo uso de las herramientas y tecnologías seleccionadas.

- Validar la solución desarrollada mediante la aplicación de pruebas de software para dar cumplimiento a los requerimientos de software levantados.

Para cumplir los objetivos anteriores, se plantean como **Tareas de Investigación:**

1. Análisis del estado del arte de sistemas de gestión de información en Cuba y el mundo referente a los sistemas de gestión de información de estudiantes extranjeros.
2. Determinación de las herramientas, tecnologías y metodología de desarrollo de software a utilizar en el desarrollo de la solución.
3. Levantamiento de información relacionada con el negocio que engloba el problema de investigación.
4. Realización de la especificación y descripción de requerimientos funcionales del sistema.
5. Definición de los requerimientos no funcionales del sistema.
6. Definición de la arquitectura del software y los patrones de diseño a utilizar en el desarrollo del sistema.
7. Implementación de la solución de software haciendo uso de las herramientas, tecnologías y metodología de desarrollo de software seleccionadas.
8. Validación de la propuesta de solución aplicando las pruebas definidas.

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

➤ **Métodos teóricos**

- **Histórico-lógico:** Se utiliza para conocer los sistemas de gestión existentes tanto a nivel nacional como internacional, que tengan semejanza con el que se va implementar, así como la investigación realizada referente al tema.
- **Analítico-Sintético:** Se hace uso de este método para el análisis bibliográfico de la investigación y para definir las características de los sistemas estudiados.
- **Modelación:** La utilización de este método permite realizar los diagramas correspondientes a las diferentes etapas del desarrollo del software, con el fin de ir documentando el trabajo en la medida de su avance como aplicación de una buena práctica.

➤ **Métodos empíricos**

- **Entrevista:** Posibilita conocer y analizar el negocio sobre el cual se realiza la propuesta de solución, permitiendo hacer un levantamiento de requisitos ajustado al trabajo que se realiza con los estudiantes extranjeros en la Dirección de Relaciones Internacionales de la Universidad de las Ciencias informática (UCI), y facilitando la constatación de la información que recopila. El diseño de las entrevistas realizadas se encuentra en el Anexo 1.

El presente trabajo está estructurado en tres capítulos, los cuales recogen las temáticas que a continuación se describen:

**Capítulo 1:** Se expone la fundamentación teórica relacionada con la investigación, donde se realiza el estudio del estado del arte de soluciones informáticas encontradas ajustándose al campo de acción con alcance internacional y nacional. Se determina la metodología de desarrollo de software, los lenguajes de programación y herramientas a utilizar en la solución.

**Capítulo 2:** En este capítulo se exponen los elementos de análisis y diseño que definen la solución de software. Se describen la técnica de obtención de requisitos utilizada en el trabajo, se realiza la modelación conceptual del negocio y la descripción de los casos de usos, según el escenario de seleccionado de la metodología de desarrollo. Además, se describen la arquitectura, los patrones de diseño utilizados, el modelo de base de datos definido. Se describe la propuesta de solución enfocada al cumplimiento de los objetivos trazados y sus principales características.

**Capítulo 3:** En el capítulo se definen elementos para guiar la implementación de la solución propuesta, tales como: los estándares de codificación y se expone la estrategia definida para la realización de las pruebas al sistema, especificando los tipos de prueba a realizar en cada momento y los resultados obtenidos.

# **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

## **1.1 Introducción**

En el presente capítulo se describen sistemas relacionados con la gestión de la información, enfocados fundamentalmente a la temática de gestión de información de estudiantes extranjeros. Además, se conceptualizan los términos relacionados con el negocio. Se realiza una descripción y fundamentación de la metodología de desarrollo, lenguajes de programación y tecnologías a utilizar en el desarrollo de la propuesta de solución.

## **1.2 Descripción de la gestión de información de estudiantes extranjeros de pregrado en la UCI.**

En la gestión de la información relacionada con los estudiantes extranjeros en la Universidad de las Ciencias Informáticas, se utilizan varios términos que definen este trabajo, algunos de ellos son:

- **Becarios:** Estudiantes que tienen la asignación de una beca o carrera para estudiar en la Universidad de las Ciencias Informáticas.
- **Condición de becarios:** Condición en la que los estudiantes extranjeros pueden realizar estudios de formación completa en las universidades cubanas.
- **Actividad extracurricular:** Actividad que realiza un estudiante o becario que no está concebida dentro del plan de estudio de la carrera que cursa.
- **Carné:** Documento de identificación nacional que se emite a residentes en Cuba y en el caso de los estudiantes con carácter temporal mientras dure la carrera.
- **Ingreso:** Se refiere a la actividad de estar temporalmente en un centro hospitalario debido a la presencia de alguna enfermedad o padecimiento.
- **Graduación:** Se refiere al evento de culminar los estudios en la carrera cursada, termina la carrera.

La gestión de la información de los estudiantes extranjeros en la UCI está a cargo fundamentalmente de la Dirección de Relaciones Internacionales (DRI). Es esta Dirección la responsable de confeccionar para cada estudiante un expediente con el recorrido que tendrá durante el transcurso de su carrera. El expediente de la DRI es independiente al expediente docente que llevan las secretarías docentes de la facultad donde cursa la carrera el estudiante.

El expediente que la DRI crea para cada estudiante, se confecciona inicialmente solo con sus datos personales, y en el transcurso de la carrera se va incorporando información sobre su recorrido. El expediente recoge información sobre:

- La facultad, carrera, año y semestre que cursa dentro de la carrera, registrando información de cada semestre cursado.
- Los carnés de identidad que se le emiten al estudiante, tratando de controlar la vigencia de los mismos para que el estudiante este actualizado migratoriamente.
- Las actividades extracurriculares en las que participa el estudiante durante la carrera, para tener una fuente confiable y completa de información en el momento de emitir criterios.
- La entradas y salidas en tiempo vacacional a su país de origen.
- Ingresos hospitalarios que tenga el estudiante, si los tiene.
- La ubicación dentro de la residencia estudiantil de la Universidad.
- La graduación.

Los expedientes en la DRI están mayormente en soporte de papel, y su gestión generalmente es de forma manual. Se utiliza el paquete ofimático para la realización de documentos que son colocados en los expedientes de los estudiantes, y también para conformar los resúmenes de información que se necesiten.

De la emisión de la información relacionada con los estudiantes extranjeros no solo es responsable la DRI, varias áreas dentro de la UCI están involucradas, pero si es la DRI la responsable de centralizarla y emitir cualquier reporte relacionado con la misma. Las áreas involucradas son:

- Facultad donde cursa la carrera el estudiante y dentro de ella, el decano y la secretaria docente.
- Residencia donde vive el estudiante y dentro de ella, el vicerrector y directores de residencia.
- Hospital de la UCI y como responsable, el coordinador y el médico del consultorio del área de su edificio.

La DRI para mantener actualizados los expedientes de los estudiantes, solicita periódicamente información a las áreas involucradas. Esta información llega a la Dirección en diferentes formatos y por diferentes vías; entonces el especialista que se encarga de la actualización de los expedientes en ocasiones debe reorganizarla antes de ponerla en cada uno de los expedientes. También la persona encargada debe actualizar los correspondientes controles digitales con los que cuenta, con el fin de

poder confeccionar resúmenes de información una vez que se los soliciten de una forma más rápida. Estos controles digitales se llevan haciendo uso del paquete ofimático, específicamente el Excel.

Para contar con más información los estudiantes extranjeros en Cuba y lograr un mejor entendimiento del trabajo con ellos, se consultó la Resolución Ministerial 186/07 Reglamento para Estudiantes Extranjeros en los Centros de Educación Superior (CES).

### **1.3 Estudio del estado del arte**

Con el objetivo de conocer como las instituciones a nivel mundial y en Cuba llevan la gestión de la información de los estudiantes extranjeros en sistemas informáticos, se realizó un análisis de algunas aplicaciones encontradas. La búsqueda se enfocó en sistemas que cumplieran con los requerimientos que tiene la DRI en cuanto a la gestión de información de los estudiantes extranjeros y para su explicación se dividieron en dos grupos: encontradas fuera de Cuba y dentro del País.

#### **1.3.1 Internacional**

##### **➤ Plataforma Campus France**

Campus France es una plataforma que promociona y contribuye con la educación superior francesa, con la acogida de estudiantes e investigadores extranjeros en Francia, fomenta su movilidad internacional, gestiona programas de becas y dirige la Red Alumni mundial.

Entre sus objetivos Campus France tiene, acoger a los estudiantes e investigadores extranjeros en Francia, brindándole toda la información necesaria para su instalación y estancia en el país. De esta forma, ayuda a los centros de educación superior y a las autoridades locales a mejorar sus dispositivos de acogida e integración. (Campus France, 2017)

Además, la Plataforma a través de los centros de acogida permite dar seguimiento de estudio a los estudiantes durante toda su estancia en Francia. La actualización continua de su avance estudiantil posibilita:

- Comprobar la inscripción efectiva del becario para el año en curso.
- Conocer posibles dificultades surgidas con el estudiante.
- Recibir una copia de los resultados académicos al final del semestre y/o del curso universitario.

Esta información se recopila a lo largo de toda la estancia y se remite a los interesados:

- Al principio de cada curso universitario, a través de una ficha individual que recoge la trayectoria del becario desde su llegada a Francia.
- Al final del curso universitario, junto con las notas del becario certificadas por el centro anfitrión.
- Y en cualquier momento, si el becario tiene dificultades.

Campus France ofrece información que ayuda a los becarios a gestionar su alojamiento en una residencia estudiantil privada o en hoteles residenciales, pero no registra donde se encuentra alojado cada uno. Igual procedimiento se sigue con las actividades culturales, la Plataforma solo ofrece información sobre ellas.

### ➤ **Sistema informático de gestión de ayudas, becas y premios de España (SAB)**

SAB es un sistema informático para la tramitación y gestión de ayudas, becas y premios de la Secretaria de Estado de Cultura del Ministerio de Educación, Cultura y Deporte de España. El sistema permite la presentación de solicitudes por vía electrónica, y trata una gestión conjunta de las solicitudes de una determinada convocatoria de subvención, con independencia de la vía de entrada de la solicitud al sistema. Da soporte a todo el proceso, desde el punto de vista del ciudadano y del gestor, utilizando los servicios proporcionados por los Departamentos de la Administración del Gobierno para la validación de certificados electrónicos, notificaciones electrónicas y verificación de datos de identidad.

Además, facilita la consulta a la Agencia Tributaria y Seguridad Social de datos de situación de deuda de los solicitantes. Es un sistema web que cuenta con un servidor de aplicación encargado de gestionar toda la información relativa de becas, ayudas y premios.

Los objetivos principales perseguidos con SAB son:

- Dar cumplimiento a lo establecido para los procedimientos de ayudas, becas y premios del Ministerio de Educación, Cultura y Deporte de España.
- Normalizar lo máximo posible los formularios de las distintas convocatorias de subvenciones.
- Estandarizar el proceso de tramitación.
- Centralizar la información de subvenciones para facilitar su posterior explotación. (Juan Pablo Sanz Martín)

### ➤ **SIGMA SIS**

SIGMA Gestión Universitaria es una agrupación de interés económico integrada por varias Universidades españolas y otras instituciones privadas del ámbito de la enseñanza superior, a las que se unió la Universidad de Córdoba en enero de 2008.

Esta empresa centra su actividad en el desarrollo de aplicaciones de gestión académica para las Universidades. Entre estas aplicaciones se encuentra el Sistema de Gestión Académica denominado **SIGMA SIS**, que entre otros dispone de los siguientes módulos:

- Módulo que gestiona los estudios oficiales de grado, así como los estudios de máster y doctorado.
- Módulo para la gestión de la ordenación docente que permite la definición de oferta académica, planes de docencia, definición de grupos de docencia, asignación de profesorado.
- Módulo para la gestión de los estudios propios de la Universidad.

SIGMA SIS es de fácil configuración y totalmente parametrizable, por lo que se trata de una solución que se adapta a las necesidades concretas de cada universidad para los procesos que incorpora, consolida la información de manera automática y ofrece gran variedad de reportes de control. (Universidad de Córdoba, 2019)

### **1.3.2 Nacional**

#### ➤ **Sistema informático para la gestión de la información de Becarios Extranjeros del Ministerio de Comercio Exterior y la Inversión Extranjera de Cuba.**

El Departamento Asistencia Técnica y Becas forma parte de la distribución organizativa de la Dirección de Colaboración Económica del Ministerio del Comercio Exterior (MINCEX), y tiene como objetivo dar seguimiento a los becarios extranjeros residentes en Cuba y fuera del país. Este departamento recibe la información de varios organismos, entidades o empresas en formato duro, vía e-mail, telefónica o mediante una persona encargada de llevar personalmente la información para luego ser tramitada por especialistas. Mediante el uso de un sistema informático el Departamento registra y organiza la información relacionada con:

- Registro histórico de los programas de becarios extranjeros, tanto de las acciones en Cuba como en el exterior.

- Control de los principales procesos asociados a los becarios extranjeros desde su matrícula hasta la graduación, incluyendo la formación posgraduada.
  - Control de la información sobre los becarios extranjeros en Cuba, y los programas de formación en el exterior directamente tutelados por instituciones cubanas.
  - Emisión de informes estadísticos con frecuencia trimestral y anual, sobre el estado del programa de becas.
  - Registro de información que contribuya al seguimiento de los graduados.
- **Sistema informático para la gestión de los procesos de la Dirección de Relaciones Internacionales en la Universidad Central Marta Abreu de Las Villas.**

La Dirección de Relaciones Internacionales de la Universidad Central Marta Abreu (DRI-UCLV) cuenta con su sistema informático como apoyo a la gestión de sus procesos. Entre los procesos que se gestionan en el sistema se encuentran los del Departamento de Atención a Estudiantes Extranjeros (DAEE), quien se encarga de atender a todos los estudiantes extranjeros que ingresan al Centro.

Del trabajo que se realiza en el DAEE, el sistema tiene implementado los procesos de:

- Registro de estudiantes extranjeros en la UCLV de manera autofinanciada o financiada por el gobierno. El desarrollo de este proceso en el sistema consiste, en registrar los datos personales y académicos del estudiante para el inicio de su objetivo estudiantil, manteniéndolos actualizados para la generación de los reportes que se requieren.
- Pago de contrato por parte de los estudiantes autofinanciados. De este proceso el sistema permite registrar el pago de lo pactado en contratos con los estudiantes en condición de autofinanciados.
- Pago de contrato por parte de los estudiantes financiados por el gobierno. De este proceso el sistema permite registrar el pago de lo pactado en contratos con los estudiantes en condición de financiados.

Esta solución informática fue desarrollada utilizando el marco de trabajo Symfony acompañado del gestor de bases de datos PostgreSQL. (Espinosa, y otros, 2016) y que se ajusta a lo establecido en la Resolución Ministerial 186/07.

El sistema informático para la gestión de los procesos de la DRI-UCLV, aunque basa su funcionamiento en la Resolución Ministerial 186/07, y esta sirve de base para el desarrollo de la propuesta de solución, no gestiona todas las funcionalidades que el cliente necesita. Un ejemplo de ello se tiene en la gestión de salidas de estudiantes extranjeros, de las actividades extracurriculares, de los ingresos, de su recorrido estudiantil, etc.; información que la DRI de la UCI necesita gestionar imperantemente.

### 1.3.3 En la Universidad

#### ➤ **Módulo de Relaciones Internacionales del Sistema de Gestión: Akademos, de la UCI**

La UCI cuenta con un sistema informático que posibilita la gestión de varios procesos de diversas áreas de la Universidad, entre ellas se encuentra la Dirección de Relaciones Internacionales. El sistema nombrado Akademos cuenta con un módulo para la DRI, entre la información que este módulo gestiona se encuentra: el registro de convenios, redes internacionales, visitas y trámites de viajeros, pero no gestiona ninguna información relacionada con el seguimiento a estudiantes extranjeros que cursa estudios en la universidad y por tanto no cubre ninguna de las necesidades planteadas por el cliente.

### 1.3.4 Resultado del análisis de las soluciones

Durante el estudio de los diversos sistemas encontrados que tratan el registro y control de la información de estudiantes extranjeros, se pudo constatar que todos poseen una variedad de funcionalidades que se ajustan estrictamente a las área y procesos que desarrollan las instituciones o áreas de trabajo que los albergan, ninguna de ellas se ajusta completamente a las necesidades que tiene la DRI de la UCI. Es por eso, que se determina no utilizar ninguno de ellos como propuesta de solución a la presente investigación, pero si se tomaron algunas ideas que pueden ser de utilidad para el desarrollo del software.

## 1.4. Metodología de desarrollo de software

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Permite llevar a cabo un proceso de software detallado y completo; para controlar, gestionar y administrar un proyecto con altas posibilidades de éxito. (Izquierdo, 2016) En un proceso de desarrollo de software, la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo.

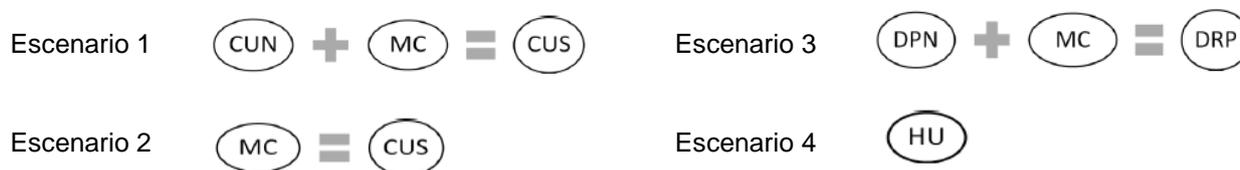
La UCI cuenta con más de seis centros productivos para la ejecución de proyectos de desarrollo de software fundamentalmente. Para gestionar estos proyectos ha definido una metodología de desarrollo híbrida, con el fin de lograr un mejor ajuste a las particularidades del desarrollo de software en la UCI, y a la vez tener una guía única para todos ellos. Esta metodología se nombra AUP-UCI y es una variación de la metodología AUP, Proceso Unificado Ágil (Agile Unified Process – AUP por sus siglas en inglés) con elementos de CMMI nivel 2.

Como el trabajo a desarrollar en la presente investigación, es una solución para ser desarrollada y utilizada en la UCI se asume guiar el proceso de desarrollo con la metodología AUP-UCI, la cual se ajusta al ambiente que presenta el negocio que se desea informatizar y da la posibilidad al cliente de acompañar el desarrollo.

Esta metodología cuenta con disciplinas ingenieriles y de gestión. Las fases son: Inicio, Ejecución y Cierre; y las disciplinas: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas, Gestión de la configuración, Planeación de proyecto y Monitoreo y control de proyecto; las tres últimas son las consideradas disciplinas de gestión. (UCI, Actividad productiva, 2015)

En la disciplina Modelado de negocio de la AUP-UCI se proponen tres variantes para el modelado, compresión y documentación del negocio, estas son: Casos de uso de negocio (CUN), Descripción de procesos de negocio (DPN) y Modelo conceptual (MC). Estas variantes condicionan la ejecución de la disciplina de Requisitos, definiéndose en ella cuatro escenarios para ejecutarla. Estos escenarios están formados por una variante de la disciplina de Modelado de negocio y una de las formas de encapsular los requisitos. Para el encapsulado de requisitos la metodología define las siguientes formas: Casos de uso del sistema (CUS), Historias de usuarios (HU) y Descripción de requisitos por procesos (DRP). En la Tabla 1 se muestran los diferentes escenarios que propone la metodología.

**Tabla 1:** Escenarios de la metodología AUP-UCI.



Luego de analizado el negocio a informatizar, las características de los escenarios que define la metodología y visionar la propuesta de solución, se decide seleccionar el escenario 2 para la modelación del sistema y por tanto la documentación del proyecto. Fue seleccionado particularmente porque se recomienda para proyectos donde el objetivo primario es la gestión y presentación de la información, y

donde no es relevante responsabilidad de las personas que ejecutan las actividades. En el negocio la responsabilidad de las actividades está centralizada en la DRI, lo que hace este elemento sencillo de trabajar en el momento del diseño de la solución.

### **1.5 Tecnologías para el desarrollo de la solución**

#### **1.5.1 Herramientas CASE**

El objetivo primordial de las herramientas CASE consiste en representar objetos de datos de negocios, sus relaciones, y ayuda a comprender mejor la forma en que fluyen estos objetos de datos entre distintas zonas de negocio en el seno de una entidad. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas no satisfacen las necesidades de la organización. (Asensio, 2018)

Dentro de las herramientas CASE, se encuentra el Visual Paradigm para dar soporte al ciclo de vida del desarrollo de software y fue la seleccionada para este trabajo, pues además de ser una de las más completas, existe experiencia positiva en su uso por los proyectos de la Universidad.

##### **- Visual Paradigm 15.1**

Se selecciona el Visual Paradigm en su versión 15.1, por ser una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: negocio, requisitos, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Su uso se ha extendido entre los analistas de todo el mundo y en la Universidad es la que se utiliza por excelencia. Dadas sus facilidades de integración con distintos entornos de desarrollo e incluye UML 2.0 como Lenguaje de Modelado. (Pressman, 2007)

#### **1.5.2 Servidor Web**

##### **- Apache 2.4.34**

Los servidores web apache son uno de los principales pilares de Internet actualmente, es mediante ellos que se visualizan las páginas web y material multimedia que se muestran día a día, también forman parte de la infraestructura de las aplicaciones móviles y servicios en la nube. (Álvaro De León, 2019)

Un servidor de páginas web es un programa que permite acceder a páginas web alojadas en una computadora. Apache es un programa que permite crear un servidor http de forma rápida y sencilla, su

configurabilidad, robustez y estabilidad hacen que cada vez más servidores reiteren su confianza en este programa.

Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable. El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. (Mesa, y otros, 2009).

Teniendo en cuenta los beneficios y la información analizada de este servidor, además de las características del sistema que se visiona, se decide utilizar Apache como servidor web para el desarrollo de la solución propuesta.

### **1.5.3 Sistema Gestor de Base de Datos**

Como sistema gestor de bases de datos a utilizar en la propuesta de solución se ha seleccionado a PostgreSQL en su versión 11.0. PostgreSQL es un potente sistema de base de datos relacional de código abierto, que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas.

PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de alto rendimiento. No sorprende que PostgreSQL se haya convertido en la base de datos relacional de código abierto elegida por muchas personas y organizaciones por sus características. (postgresql, 2019)

Las características más importantes de PostgreSQL son:

- El lenguaje SQL que usa es muy próximo al estándar ISO/IEC, lo que resulta relativamente sencillo portar consultas y scripts de otros sistemas de bases de datos, y así aprender fácilmente las variantes de este lenguaje.
- Cumple con atomicidad, consistencia, aislamiento y durabilidad para sus operaciones.
- Permite crear esquemas, tablas heredadas y triggers orientados a eventos que no poseen otros motores.

- Permite definir procedimientos, no solo en PostgreSQL, sino también en otros lenguajes como Pearl, TCL o Python. Incluso si el lenguaje que se quiere usar no está soportado, se puede definir con nuevas extensiones.
- Si se necesita algún tipo de dato que no está soportado de serie, también se puede definir.
- Tiene un soporte nativo de replicación maestro-esclavo, pero también es posible añadir otros tipos a través de productos de terceros, libres o de pago.
- También provee una excelente escalabilidad vertical. (Julio Gonzalez Gil , 2018)

### 1.5.4 Herramientas y lenguajes informáticos

Para desarrollar un software es necesario una serie de requisitos, entre ellos se encuentra el conjunto de símbolos y de reglas tanto sintácticas como semánticas que indican al ordenador que acciones desarrollar, a esta aglomeración de instrucciones se le llama lenguaje de programación.

#### - Lenguaje de Marcado de Hipertexto (HTML)

HTML es un lenguaje que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. (Herrera, 2015)

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Y aunque algunas Interfaces de Programación de Aplicaciones (APIs) y la especificación de las Hojas de Estilo en Cascada (CSS, por sus siglas en inglés) por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript de otras cosas. (Gauchat, 2012)

#### - Hojas de estilo en cascada (CSS3)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los

contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML bien definidos y con significado completo, también llamados documentos semánticos. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes, y sus ventajas son:

- Flexibilidad.
- Compatibilidad y continuidad.
- Separación del contenido y presentación.
- Unificación del diseño de las páginas del sitio.
- Optimización de los tiempos de carga y el tráfico del servidor (Pérez, 2009).

### - JavaScript

Es un lenguaje interpretado, que no requiere compilación, es multiplataforma y permite incluir macros en páginas web. Estos macros se ejecutan en el ordenador del cliente, y no en el servidor. Por su compatibilidad con la mayoría de los navegadores modernos, es uno de los lenguajes de programación del cliente más utilizado. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. (Mesa, y otros, 2009)

Las ventajas más destacadas de JavaScript:

- Es un lenguaje muy sencillo.
- Es rápido, por lo tanto, tiende a ejecutar las funciones inmediatamente.
- Cuenta con múltiples opciones de efectos visuales.
- Es soportado por los navegadores más populares y es compatible con los dispositivos más modernos, incluyendo iPhone, móviles y PS3.
- Es muy versátil, puesto que es muy útil para desarrollar páginas dinámicas y aplicaciones web.
- Es una buena solución para poner en práctica la validación de datos en un formulario.
- Es multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil.

### - PHP

Se utiliza PHP en su versión 7 como lenguaje de programación del lado del servidor. Es un lenguaje de código abierto y uno de los más utilizados para los desarrollos web en todo el mundo.

La ventaja de PHP frente a otros lenguajes de programación compatibles con el entorno web, es que la mayoría de servicios de hosting son compatibles con PHP y las principales herramientas para desplegar servicios web rápidamente como WordPress, Joomla o Drupal, están escritas en PHP.

PHP 7 es la nueva versión de PHP, lanzada en el año 2016 y trae consigo algunas novedades. A nivel general, se ejecuta más rápido que sus versiones anteriores, lo que trae una ventaja sustancial. A nivel técnico son algunos detalles lo que lo diferencian de sus versiones anteriores, tales como: definición de constantes tipo array, la posibilidad de usar clases anónimas, sesiones tipo array y otros que facilitan su uso. (Julian Bohorquez, 2016)

### 1.5.5 Marco de desarrollo web

Un marco de trabajo, es una estructura que establece como organizar el desarrollo de un software determinado.

Otros autores definen un marco de trabajo como: una estructura conceptual y tecnológica de soporte, definida normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado para ayudar al desarrollo y unión de los diferentes componentes de un proyecto. (Rivero, 2015)

#### - **Symfony4**

Symfony es un marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador (MVC). Separa la lógica de negocio, la lógica de servidor y la presentación de las interfaces web, además proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas nix (Unix, Linux, etc.) como en plataformas Windows. (Espinosa, y otros, 2016)

Se ha seleccionado el marco de trabajo Symfony, en su versión 4, para el desarrollo del lado del back-end, donde se hace la conexión con las bases de datos, y garantizar la seguridad del sistema

gestionando y optimizando al máximo los recursos para que las páginas sean más ligeras. Además, reúne las mejores prácticas de desarrollo web e integra varias herramientas informáticas que aportan fortaleza al producto final, y usa Doctrine para el Mapeo Relacional de Objeto (Object Relational Mapping – ORM por sus siglas en inglés) en caso de ser necesario su uso.

### - **Bootstrap 4**

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de la pantalla de una computadora, una tableta u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS3, pero es 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. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. (Bootstrap, 2019)

Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto le da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El framework trae varios elementos con estilos predefinidos fáciles de configurar: botones, menús desplegables, formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos. (Johanny Solis, 2014)

Características de Bootstrap:

- Soporte: ofrece un soporte extraordinario con HTML5 y CSS3. Sencillo de manejar.
- Imágenes: permite insertar imágenes responsive de modo simple, las cuales se adaptan de manera automática a la pantalla del dispositivo.
- Personalizable: brinda la posibilidad de diseñar un sitio web utilizando diferentes combinaciones de HTML, CSS y JavaScript.
- Plugins: incluye una inmensa cantidad gracias a la aparición de nuevos plugins de terceros.
- Documentación: contiene mucha información necesaria para resolver dudas a principiantes y a profesionales. (Johanny Solis, 2014)

Este framework se ha seleccionado para el diseño de las interfaces de la propuesta de solución, por sus bondades y fácil integración con las demás herramientas y tecnologías seleccionadas para el desarrollo.

### 1.6 Entorno de Desarrollo Integrado (IDE)

#### - PhpStorm

PhpStorm de JetBrains es un PHP IDE inteligente y profesional que proporciona a los desarrolladores una mezcla de herramientas inteligentes, hábitos útiles y mejores prácticas para el desarrollo de PHP. Todos orientados a ayudar a la productividad del desarrollador. Las características principales son:

- Asistencia de código inteligente: se realizan inspecciones para verificar que el código es correcto y para analizar el proyecto en su conjunto.
- Navegación de código inteligente: el IDE entiende a donde se quiere ir y le lleva allí instantáneamente.
- Refactorización rápida y segura: se puede refactorizar el código de manera segura para renombrar, mover, eliminar, extraer un método, etc.
- Fácil debug y test.
- Control de versiones: permite hacer uso de diferentes herramientas utilizadas con este fin.

Debido a que la aplicación a desarrollar es web y tiene varios archivos, bibliotecas y clases almacenadas en carpetas anidadas y por los beneficios anteriormente explicados, se escoge PhpStorm para facilitar el trabajo durante la disciplina de Implementación. Además, posibilita el desarrollo del patrón Modelo Vista Controlar (MVC) y presenta el proyecto organizado estructuralmente facilitando luego so soporte. (jetbrains, 2019)

### 1.7 Conclusiones del capítulo

El estudio realizado sobre la gestión de la información asociada a los estudiantes extranjeros en la Dirección de Relaciones Internacionales permitió conocer las características fundamentales de dicho proceso logrando así, determinar y fundamentar las bases teóricas de la investigación. El análisis de sistemas de similar propósito, posibilitó ultimar que, los sistemas estudiados no satisfacen las necesidades y condiciones requeridas por el cliente y el negocio; por lo que se hace necesario desarrollar una solución informática que contribuya con la gestión de información de los estudiantes extranjeros en la UCI. Y con el análisis de las herramientas, lenguajes y la selección del proceso de desarrollo de

software, permitió profundizar en los conocimientos necesarios para el desarrollo de una solución informática que se ajuste a las necesidades del cliente.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

### 2.1 Introducción

En el capítulo se exponen las características de la propuesta de solución, iniciándose con el análisis del modelo de negocio para una mejor comprensión del entorno de la investigación. Se presenta una descripción detallada de los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Se describe la arquitectura, el diseño del sistema y los patrones utilizados. Se muestra el diagrama lógico de datos, las principales vistas del sistema y el diagrama de despliegue.

### 2.2 Propuesta de solución

Teniendo en cuenta las necesidades identificadas durante el análisis preliminar del registro y control de la información de los estudiantes extranjeros por la Dirección de Relaciones Internacionales de la Universidad de la Ciencias Informáticas, se decide desarrollar una solución de software que permita la gestión de la información de estos estudiantes; así como, la obtención de reportes de forma rápida y ajustados a la información entregada a la Dirección.

El sistema permitirá de manera sencilla realizar el registro de la información relacionada con los estudiantes extranjeros durante los años que dure la carrera en la que están matriculados en la Universidad. Ofrecerá la posibilidad de acceder a la información almacenada de forma rápida, logrando la obtención de reportes según la necesidad del usuario que se autentique. Los reportes pondrán ser cualitativos y cuantitativos, según se requiera. Este sistema contará con una gestión de usuarios y de roles, y que a través de los roles que se definan se dará a los usuarios, acceso a las funcionalidades implementadas.

El sistema permitirá a la DRI mantener centralizada y accesible la información relacionada con el recorrido y vida de los estudiantes extranjeros en la Universidad, específicamente sobre los temas de:

- La carrera, año y semestre que cursa, incluyendo la facultad a la que pertenece.
- Los periodos de vacaciones en los que haya salido del Centro, incluyendo el lugar y la fecha de salida y entrada.
- Los ingresos en instituciones de salud que haya tenido, en caso que suceda.
- Las actividades extracurriculares en las que haya participado durante la carrera.
- La ubicación del estudiante en la residencia estudiantil de la UCI.

Al estar la información organizada, actualizada y centralizada será fácil de acceder lo que permitirá:



A continuación, se describe el concepto Estudiante relacionado en el MC, los demás se encuentran descritos en el Anexo 2.

**Tabla 2:** Descripción del concepto Estudiante.

Descripción		➤ Representa los estudiantes extranjeros que cursan o cursaron una carrera en la UCI.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>primer_nombre</b>	Primer nombre del estudiante.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros - Caracteres extraños (* - . /)	- Diferente a las clases válidas.
<b>segundo_nombre</b>	Segundo nombre del estudiante.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros - Caracteres extraños (* - . /)	- Diferente a las clases válidas.
<b>primer_apellido</b>	Primer apellido del estudiante.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros - Caracteres extraños (* - . /)	- Diferente a las clases válidas.
<b>segundo_apellido</b>	Segundo apellido del estudiante.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros - Caracteres extraños (* - . /)	- Diferente a las clases válidas.
<b>carne</b>	Representa el número del carné de identidad del estudiante.	Carne	Sí	Sí	- Números enteros	- Diferente a las clases válidas.
<b>pasaporte</b>	Representa el número del pasaporte activo del estudiante.	cadena de caracteres (20)	No	Sí	- Letras mayúsculas. - Números enteros.	- Diferente a las clases válidas.
<b>solapin</b>	Representa el solapin elaborado al estudiante e la Universidad	cadena de caracteres (10)	Sí	Sí	- Letras - Números enteros	- Diferente a las clases válidas.

<b>Descripción</b>	➤ Representa los estudiantes extranjeros que cursan o cursaron una carrera en la UCI.					
<b>Atributos</b>						
<b>Nombre</b>	<b>Descripción</b>	<b>Tipo</b>	<b>¿Puede ser nulo?</b>	<b>¿Es único?</b>	<b>Restricciones</b>	
<b>pais_procedencia</b>	País del cual procede el estudiante.	<b>Pais</b>	No	Sí	-	-
<b>provincia</b>	Provincia de la cual procede en estudiante en su país de origen.	cadena de caracteres (20)	No	Sí	- Letras - Numeros	- Diferente a las clases válidas.
<b>municipio</b>	Municipio del cual procede en estudiante en su país de origen.	cadena de caracteres (20)	No	Sí	- Letras - Numeros	- Diferente a las clases válidas.
<b>fecha_entrada</b>	Fecha en la que entró el estudiante a Cuba para inicio de la carrera.	fecha	No	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>fecha_salida</b>	Fecha en la que salió el estudiante a Cuba al finalizar estudios.	fecha	Sí	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>fecha_entrada_uci</b>	Fecha en la que entró el estudiante a UCI para inicio de la carrera.	fecha	No	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>fecha_salida_uci</b>	Fecha en la que salió el estudiante a Cuba al finalizar estudios.	fecha	Sí	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>sexo</b>	Sexo del estudiante.	cadena de caracteres (10)	No	Sí	- Femenino - Masculino	- Diferente a las clases válidas.
<b>usuario</b>	Usuario de correo que se le asigna al estudiante.	cadena de caracteres (10)	No	Sí	- Letras	- Diferente a las clases válidas.
<b>telefono</b>	Números de teléfono del estudiante.	numérico (20)	Sí	No	- Números enteros	- Diferente a las clases válidas.

<b>Descripción</b>	➤ Representa los estudiantes extranjeros que cursan o cursaron una carrera en la UCI.					
<b>Atributos</b>						
<b>Nombre</b>	<b>Descripción</b>	<b>Tipo</b>	<b>¿Puede ser nulo?</b>	<b>¿Es único?</b>	<b>Restricciones</b>	
<b>correo</b>	Correo electrónico del estudiante.	cadena de caracteres (50)	Sí	No	- Letras - Números enteros - Caracteres extraños (@ - .)	- Diferente a las clases válidas.
<b>apartamento</b>	Apartamento de la residencia estudiantil donde está ubicado el estudiante.	<b>Apartamento</b>	No	Sí	-	-
<b>graduado</b>	Representa si el estudiante se graduó de la carrera o no.	numérico	No	Sí	- Si - No	- Diferente a las clases válidas.

### 2.3.1 Reglas del negocio

Las reglas de negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización, y son de vital importancia para alcanzar los objetivos de la misma. Las organizaciones funcionan siguiendo múltiples reglas de negocio, que pueden estar embebidas en procesos, aplicaciones informáticas o documentos. Estas son en una organización el medio por el cual la estrategia de funcionamiento es implementada y especifican en un nivel de detalle adecuado lo que se debe hacer.

Para la construcción de un sistema informático que se ajuste a las necesidades de un cliente, es vital identificar e incluir las reglas de negocio que rigen los procesos en la organización, ya que solo así, el sistema se comportará ajustado a las necesidades del cliente.

Mediante las entrevistas realizadas y la documentación revisada para la modelación del negocio se identificaron un grupo de reglas de negocio que contribuyeron a la confección del MC y a una mejor comprensión del negocio, estas fueron agrupadas según lo establecido en el documento Reglas de negocio del Expediente de proyecto 5.0 definido para los proyectos de la UCI.

### ➤ Reglas textuales

- Los estudiantes durante los años que dure la carrera solo pueden tener un año de repitencia, solo en casos excepcionales (autorizados) pueden continuar estudios.

### ➤ Reglas del modelo de datos

- Un estudiante solo puede cursar estudios en condición: de becario, de autofinanciamiento o de estudiante extranjero con matrícula externa.

### ➤ Reglas de relación

- Un estudiante extranjero durante el curso de una carrera puede tener registro de participación en varias actividades extracurriculares dentro de un semestre.
- Solo se pueden asociar actividades extracurriculares a estudiantes extranjeros que estén cursando una carrera en la Universidad.
- En una misma actividad extracurricular pueden participar varios estudiantes.
- Un estudiante solo puede estar cursando al mismo tiempo una carrera.
- Un estudiante solo puede estar ubicado en un apartamento.
- La Universidad puede tener cursando estudios a estudiantes extranjeros de varios países.
- En una facultad solo se puede impartir una carrera.
- Una carrera está compuesta por varios años, y estos años a su vez, por semestres.
- Un apartamento solo puede pertenecer a un edificio, y este a su vez, a una manzana.
- Un estudiante puede ser ingresado varias veces dentro en un semestre mientras cursa una carrera.
- Un estudiante puede realizar varias salidas y entradas a la Universidad durante el transcurso de una carrera.

### ➤ Reglas de derivación

- El total de estudiantes que han cursado una carrera en la Universidad se puede calcular a partir de todos los estudiantes registrados como graduados.
- El total de estudiantes que cursan carrera en la Universidad se puede calcular a partir de los estudiantes que se mantienen cursando estudios.

- Para conocer si un estudiante ha repetido algún año de la carrera se puede determinar revisando el registro del recorrido del estudiante y si tiene algún año o semestre repetido se considera que repitió.

### 2.4 Requisitos del software

La obtención de requisitos es una de las etapas fundamentales del proceso de análisis, diseño y construcción de un software, pues contribuye a la realización de una primera descripción del problema que permite establecer las motivaciones para su solución mediante sistemas informáticos. (Zapata, et al., 2014)

El análisis de requisitos, también conocido como ingeniería de requisitos, es el proceso de definir las expectativas del usuario para un nuevo software que se está creando o modificando. En la ingeniería de software, a veces se hace referencia a ella por nombres como: la recopilación de requisitos o la captura de requisitos. El análisis de requisitos abarca aquellas tareas que entran en la determinación de las necesidades o condiciones a cumplir por un producto nuevo, teniendo en cuenta los requisitos de los diversos interesados. (visual-paradigm, 2019)

#### 2.4.1 Técnicas de captura de requisitos

Existen varias técnicas de recopilación de requisitos o captura de requisitos, estas permiten que el proceso de recopilación se haga de una forma más efectiva. Entre las técnicas se encuentran:

- Entrevista,
- Tormenta de ideas,
- Técnicas para facilitar las especificaciones de una aplicación (TFEA),
- Juegos de roles, y otras.

Para este trabajo se decidió el uso de la entrevista, ya que es la más tradicional de ellas; y consiste en reuniones entre el analista e interesado, donde a través de preguntas y repuestas se extrae el dominio de la aplicación. (Goguen, et al., 1993)

La aplicación de la entrevista permitió analizar en detalle las necesidades del cliente, y permitió conocer las expectativas del mismo respecto al trabajo a realizar. También se realizó la revisión de algunos reportes conformados por la DRI, que relacionan información de los estudiantes extranjeros, para conocer su formato y composición. Que el negocio no fuera tan complicado propició que la técnica aplicada cubriera todas las necesidades de información del analista, logrando un claro levantamiento de requisitos.

## 2.4.2 Requisitos funcionales

Los requisitos funcionales capturan el comportamiento previsto del sistema. Este comportamiento puede ser servicios, tareas o funciones expresadas que el sistema debe realizar. En el desarrollo de productos, es útil distinguir entre la funcionalidad de línea de base necesaria para que un sistema compita en ese dominio del producto, y las características que diferencian el sistema de los productos de la competencia, y de variantes en la línea, familia de productos de su propia organización. Las características pueden ser funcionalidades adicionales o diferir de la funcionalidad básica junto con algunos atributos de calidad como el rendimiento o la utilización de la memoria. (Malan, y otros, 2014)

Como resultado de haber aplicado las técnicas de captura de requerimientos abordadas en el párrafo anterior se obtuvo como un total de 80 requisitos funcionales, los cuales se exponen en la Tabla 3.

*Tabla 3: Descripción de requisitos funcionales.*

No.	Requisitos	Descripción	Complejidad	Prioridad
RF1	Autenticar usuario	Permite realizar el proceso de autenticación para acceder al sistema.	Media	Media
RF2	Registrar usuario	Permite registrar un nuevo usuario.	Baja	Baja
RF3	Eliminar usuario	Permite elimina un usuario.	Baja	Baja
RF4	Modificar usuario	Permite modificar usuarios.	Baja	Baja
RF5	Listar usuarios	Muestra el listado de los usuarios que registrados.	Baja	Baja
RF6	Visualizar trazas	Permite visualizar las trazas que se registran en el sistema son el manejo de la información.	Baja	Baja
RF7	Registrar rol	Permite registrar un nuevo rol.	Baja	Baja
RF8	Modificar rol	Permite modificar un rol registrado.	Baja	Baja
RF9	Eliminar rol	Permite eliminar un rol registrado.	Baja	Baja
RF10	Mostrar listado de roles	Permite mostrar los roles registrados.	Baja	Baja
RF11	Registrar estudiante	Permite el registro de un estudiante.	Media	Alta
RF12	Mostrar detalles de un estudiante activo	Permite mostrar detalles de un estudiante que cursa estudios.	Baja	Alta
RF13	Modificar estudiante activo	Permite que se le modifiquen datos a un estudiante que cursa estudios.	Media	Alta

<b>No.</b>	<b>Requisitos</b>	<b>Descripción</b>	<b>Complejidad</b>	<b>Prioridad</b>
<b>RF14</b>	Mostrar listado de estudiantes activos	Se muestra el listado de estudiantes que cursan estudios.	Media	Alta
<b>RF15</b>	Registrar estudiantes graduados	Permite asignar el dato de graduado a un grupo de estudiantes	Media	Alta
<b>RF16</b>	Registrar carné a un estudiante	Permite el registro de un carné a un estudiante.	Baja	Baja
<b>RF17</b>	Eliminar carné a un estudiante	Permite que un carné sea eliminado a un estudiante.	Baja	Baja
<b>RF18</b>	Modificar carné a un estudiante	Permite la modificación de los datos de un carné de un estudiante.	Baja	Baja
<b>RF19</b>	Mostrar carné de un estudiante	Permite que se muestren los carnés que ha tenido un estudiante.	Baja	Baja
<b>RF20</b>	Registrar carrera	Permite el registro de una carrera.	Media	Alta
<b>RF21</b>	Eliminar carrera	Permite eliminar una carrera que no tenga asociaciones con estudiantes.	Media	Alta
<b>RF22</b>	Modificar una carrera	Permite modificar una carrera que no tenga asociaciones con estudiantes.	Media	Alta
<b>RF23</b>	Mostrar listado de carreras	Permite que se muestre un listado con todas las carreras.	Media	Media
<b>RF24</b>	Mostrar detalles de una carrera	Permite que se muestren los datos que tiene una carrera.	Baja	Baja
<b>RF25</b>	Asignar semestres y años a una carrera	Permite la asociación de una carrera con sus años y semestres.	Media	Alta
<b>RF26</b>	Registrar año	Permite el registro de un año.	Baja	Alta
<b>RF27</b>	Eliminar año	Permite la eliminación de un año que no haya sido asociado a una carrera.	Baja	Media
<b>RF28</b>	Modificar año	Permite la modificación de un año.	Baja	Media
<b>RF29</b>	Mostrar listado de años registrados.	Muestra el listado de los años registrados.	Baja	Alta
<b>RF30</b>	Registrar semestre	Permite el registro de un semestre.	Baja	Alta
<b>RF31</b>	Eliminar semestre	Permite eliminar un semestre que no haya sido relacionado con una carrera.	Baja	Baja
<b>RF32</b>	Modificar semestre	Permite la modificación de un semestre.	Baja	Media

<b>No.</b>	<b>Requisitos</b>	<b>Descripción</b>	<b>Complejidad</b>	<b>Prioridad</b>
<b>RF33</b>	Mostrar listado de semestres registrados.	Muestra el listado de los semestres registrados.	Baja	Alta
<b>RF34</b>	Registrar facultad	Permite el registro de una facultad	Baja	Alta
<b>RF35</b>	Eliminar facultad	Permite la eliminación de una facultad si esta no ha sido relacionada.	Baja	Baja
<b>RF36</b>	Modificar facultad	Permite la modificación de una facultad.	Baja	Media
<b>RF37</b>	Mostrar listado de facultades	Permite que se muestre el listado de facultades registradas.	Baja	Alta
<b>RF38</b>	Asignar carreras a facultad	Permite que se asocien carreras a una facultad seleccionada.	Media	Alta
<b>RF39</b>	Registrar manzana	Permite el registro de una manzana.	Baja	Alta
<b>RF40</b>	Eliminar manzana	Permite la eliminación de una manzana que no tenga relación con edificios.	Baja	Baja
<b>RF41</b>	Modificar manzana	Permite la modificación de una manzana.	Baja	Media
<b>RF42</b>	Mostrar listado de manzanas	Permite que se muestre el listado de manzanas registradas.	Baja	Alta
<b>RF43</b>	Registrar apartamento	Permite el registro de un apartamento.	Baja	Alta
<b>RF44</b>	Eliminar apartamento	Permite la eliminación de un apartamento siempre que no esté relacionado.	Baja	Baja
<b>RF45</b>	Modificar apartamento	Permite modificar los datos de un apartamento.	Baja	Media
<b>RF46</b>	Mostrar detalles de un apartamento	Permite que se muestren los datos que tiene un apartamento.	Baja	Baja
<b>RF47</b>	Mostrar listado de apartamentos	Permite que se muestre el listado de los apartamentos registrados.	Baja	Alta
<b>RF48</b>	Registrar edificio	Permite el registro de edificios.	Baja	Alta
<b>RF49</b>	Eliminar edificio	Permite la eliminación de un edificio que no esté relacionado.	Baja	Baja
<b>RF50</b>	Modificar edificio	Permite modificar los datos para un edificio.	Baja	Media
<b>RF51</b>	Mostrar detalles de un edificio	Permite que se muestren los datos que tiene un edificio.	Baja	Baja
<b>RF52</b>	Mostrar listado de edificios	Permite que se muestre el listado de los edificios registrados.	Baja	Alta
<b>RF53</b>	Buscar estudiante activo	Permite que se busque a un estudiante activo (que este	Media	Alta

<b>No.</b>	<b>Requisitos</b>	<b>Descripción</b>	<b>Complejidad</b>	<b>Prioridad</b>
		cursando la carrera) dado parámetros de búsqueda.		
<b>RF54</b>	Registrar salida a un estudiante	Permite que se registre un evento de salida para un estudiante.	Media	Alta
<b>RF55</b>	Eliminar salida a un estudiante	Permite que se elimine un evento de salida registrado a un estudiante.	Baja	Baja
<b>RF56</b>	Mostrar salidas efectuadas por un estudiante	Permite que se muestre las salidas que tiene registradas un estudiante.	Media	Media
<b>RF58</b>	Modificar salida registrada a un estudiante	Permite la modificación de los datos de una salida registrada para un estudiante.	Media	Media
<b>RF59</b>	Registrar ingreso a un estudiante	Permite que se efectúe el registro de datos de un ingreso para un estudiante.	Media	Alta
<b>RF60</b>	Eliminar ingreso a un estudiante	Permite la eliminación de un ingreso para un estudiante	Baja	Baja
<b>RF61</b>	Modificar ingreso registrado a un estudiante	Permite la modificación de los datos de un ingreso que se ha registrado a un estudiante.	Media	Media
<b>RF62</b>	Mostrar los ingresos que ha tenido un estudiante	Se muestran todos los ingresos que ha tenido un estudiante.	Media	Media
<b>RF63</b>	Registrar actividad a un estudiante	Permite el registro de una actividad a un estudiante.	Media	Alta
<b>RF64</b>	Eliminar actividad registrada a un estudiante	Permite que se elimine una actividad registrada a un estudiante.	Baja	Baja
<b>RF65</b>	Modificar actividades registradas a un estudiante	Permite la modificación de los datos de una actividad registrada a un estudiante.	Media	Media
<b>RF66</b>	Mostrar actividades registradas a un estudiante	Permite mostrar las actividades registradas a un estudiante.	Media	Media
<b>RF67</b>	Registrar tipo de actividad	Permite el registro de un tipo de actividad.	Media	Alta
<b>RF68</b>	Eliminar tipo de actividad	Permite la eliminación de un tipo de actividad que no esté relacionada con un estudiante.	Baja	Baja
<b>RF69</b>	Modificar tipo de actividad	Permite la modificación de un tipo de actividad.	Media	Media
<b>RF70</b>	Mostrar listado de tipos de actividades	Permite que se muestre el listado de los tipos actividades registradas.	Media	Media

<b>No.</b>	<b>Requisitos</b>	<b>Descripción</b>	<b>Complejidad</b>	<b>Prioridad</b>
<b>RF71</b>	Mostrar estudiantes por carrera y facultad	Permite mostrar reporte con el listado de los estudiantes organizados por carrera, facultad, año y semestre que cursa, utilizando criterios de selección.	Media	Alta
<b>RF72</b>	Mostrar reporte cuantitativo de estudiantes por carreras registradas	Permite mostrar reporte cuantitativo con estudiantes que cursan carreras en la Universidad, utilizando criterios de selección.	Media	Alta
<b>RF73</b>	Mostrar estudiantes que culminaron la carrera	Permite mostrar reporte con el listado de los estudiantes que culminaron carrera en la Universidad, utilizando criterios de selección.	Media	Alta
<b>RF74</b>	Mostrar reporte cuantitativo de estudiantes graduados	Permite mostrar reporte cuantitativo con estudiantes que terminaron carreras en la Universidad, utilizando criterios de selección.	Media	Alta
<b>RF75</b>	Mostrar estudiantes con ingresos	Permite mostrar reporte con el listado de los estudiantes activos que han tenido ingresos, utilizando criterios de selección.	Media	Alta
<b>RF76</b>	Mostrar estudiantes por ubicación en residencia	Permite mostrar reporte con el listado de los estudiantes activos con su ubicación en residencia, utilizando criterios de selección.	Media	Alta
<b>RF77</b>	Mostrar completitud de capacidad de la residencia	Permite mostrar reporte con información sobre la completitud de llenado de la residencia estudiantil, utilizando criterios de selección.	Media	Alta
<b>RF78</b>	Mostrar estudiantes con salidas registradas	Permite mostrar reporte con el listado de los estudiantes con salidas registradas, utilizando criterios de selección.	Media	Alta
<b>RF79</b>	Mostrar estudiantes con actividades extracurriculares registradas	Permite mostrar reporte con el listado de los estudiantes con actividades extracurriculares registradas, utilizando criterios de selección.	Media	Alta
<b>RF80</b>	Exportar reportes a PDF	Permite la impresión de los reportes que se generen en el sistema	Media	Media

En la Tabla 4 se muestra el resumen de la evaluación de complejidad y prioridad asignada a los requisitos funcionales.

**Tabla 4:** Resumen de la cantidad de requisitos funcionales según complejidad y prioridad.

Resumen		
Cantidad de requisitos	Complejidad	Prioridad
Alta	0	37
Media	35	18
Baja	45	25
<b>Total</b>	<b>80</b>	<b>80</b>

### 2.4.3 Requisitos no funcionales

Son condiciones que se le imponen al sistema a desarrollar relacionadas con aspectos principalmente de calidad, usabilidad, rendimiento, disponibilidad, fiabilidad, seguridad, compatibilidad con hardware o software. Son propiedades o cualidades que el producto debe tener. (Pressman, 2007) A continuación se muestran los requisitos no funcionales definidos para la propuesta de solución.

#### ➤ Usabilidad

- **RNF1:** Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los iconos o textos de acciones deben estar representados por una imagen o texto acorde a la acción que se realiza.
- **RNF2:** El sistema debe presentar un menú lateral y una barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
- **RNF3:** El sistema debe permitir a través de un menú lateral el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de la arquitectura de información.

#### ➤ Seguridad

- **RNF4:** La información estará protegida contra accesos no autorizados utilizando mecanismo de validación que puedan garantizarlo, debe contarse con usuarios, contraseñas y nivel de acceso a la información gestionados a través de los roles del sistema.
- **RNF5:** Debe usarse mecanismo de encriptación de datos que por cuestiones de seguridad no deben viajar al servidor en texto plano, como es el caso de las contraseñas.
- **RNF6:** Antes de proceder a la eliminación de información en el sistema debe existir un mensaje de advertencia.

### ➤ Requisitos de interfaz de usuario

- **RNF7:** Los mensajes de advertencia que se muestren en todo el sistema deben tener texto estándar según la acción, ejemplo: si una información fue registrada correctamente, todos los mensajes que se ajusten a esta acción pueden decir: "La información fue registrada satisfactoriamente. "
- **RNF8:** La interfaz debe tener un diseño sencillo con pocos gráficos, con vista a acelerar la velocidad de respuesta del sistema.

### ➤ Requerimientos de Interfaz externa

- **RNF9:** Se debe utilizar para la comunicación entre el cliente y el servidor de aplicaciones se el protocolo HTTPS y entre el servidor de aplicaciones y la base de datos el protocolo TCP/IP.

### ➤ Rendimiento

- **RNF10:** El tiempo promedio de respuesta del sistema no excederá los 5 segundos de manera general.

### ➤ Hardware

- **RNF11:** Para la ejecución del sistema se requiere que la PC cliente tenga los siguientes componentes de hardware: Pentium 4 o superior, 512 MB RAM y 1 GB disco duro disponible como mínimo.

### ➤ Software

- **RNF12:** Para el uso del sistema se requiere una PC cliente con los siguientes sistemas operativos: Windows o GNU/Linux.
- **RNF13:** Para el despliegue se debe contar en el servidor de bases de datos con PostgreSQL 11.0.

#### 2.4.4 Validación de los requisitos funcionales

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos representa realmente al sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos.

Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias. (Escalona, y otros, 2017)

Aun así, existen algunas técnicas que pueden aplicarse con la intención de argumentar la pertinencia de las técnicas seleccionadas. En este trabajo se utilizó la técnica prototipado de interfaz de usuario, ya que es una técnica de representación aproximada de la interfaz de usuario de un software a clientes y usuarios y que les permite entender fácilmente la propuesta de los ingenieros para resolver su problema de negocio.

## 2.5 Modelo de caso de uso del sistema

El Modelo de Casos de Uso (MCU) es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Este modelo sirve para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. En los epígrafes siguientes se estarán describiendo los elementos del MCU confeccionados para la solución propuesta.

### 2.5.1 Definición de los actores del sistema

Un actor no es parte del sistema, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información; puede estar representado por un ser humano, un software o una máquina que interactúa con el sistema. Los actores del sistema son aquellas personas o sistemas que interactúan de manera directa con el. Para el trabajo se han definido los actores que se detallan en la Tabla 5.

*Tabla 5: Descripción de los actores del sistema.*

Actores del Sistema	Justificación
<b>Usuario</b>	Es una generalización que representa el rol de los usuarios para autenticarse en el sistema.
<b>Trabajador</b>	Es el rol encargado de la gestión de toda la información referente a los estudiantes extranjeros en la Universidad, las operaciones básicas a realizar son: registrar, modificar, eliminar y generar reportes de toda la información referente a los estudiantes.  Las responsabilidades de este actor pueden ser desagregadas en varios roles mediante la gestión de roles y posterior asignación de rol a los usuarios, lo que quiere decir, que el sistema puede tener un rol Trabajador para la gestión de toda la información o tantos usuarios con accesos limitados como el Administrador defina.
<b>Administrador</b>	El administrador es el rol designado para gestionar los accesos a las funcionalidades del sistema, es decir, la gestión de roles y usuarios con permiso de acceso al sistema.

### 2.5.2 Diagrama de caso de uso del sistema

Un Caso de Uso del Sistema (CUS) es un conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Un diagrama de casos de uso del sistema consiste

en relacionar los actores del sistema con los casos de usos del sistema, permitiendo que los desarrolladores de software y los clientes lleguen a un acuerdo sobre las condiciones y posibilidades que debe cumplir el sistema para cada tipo de usuario. (Mediavilla, 2010)

En la Figura 2 siguiente se muestra el diagrama de CUS donde se encapsulan todos los requisitos funcionales identificados.

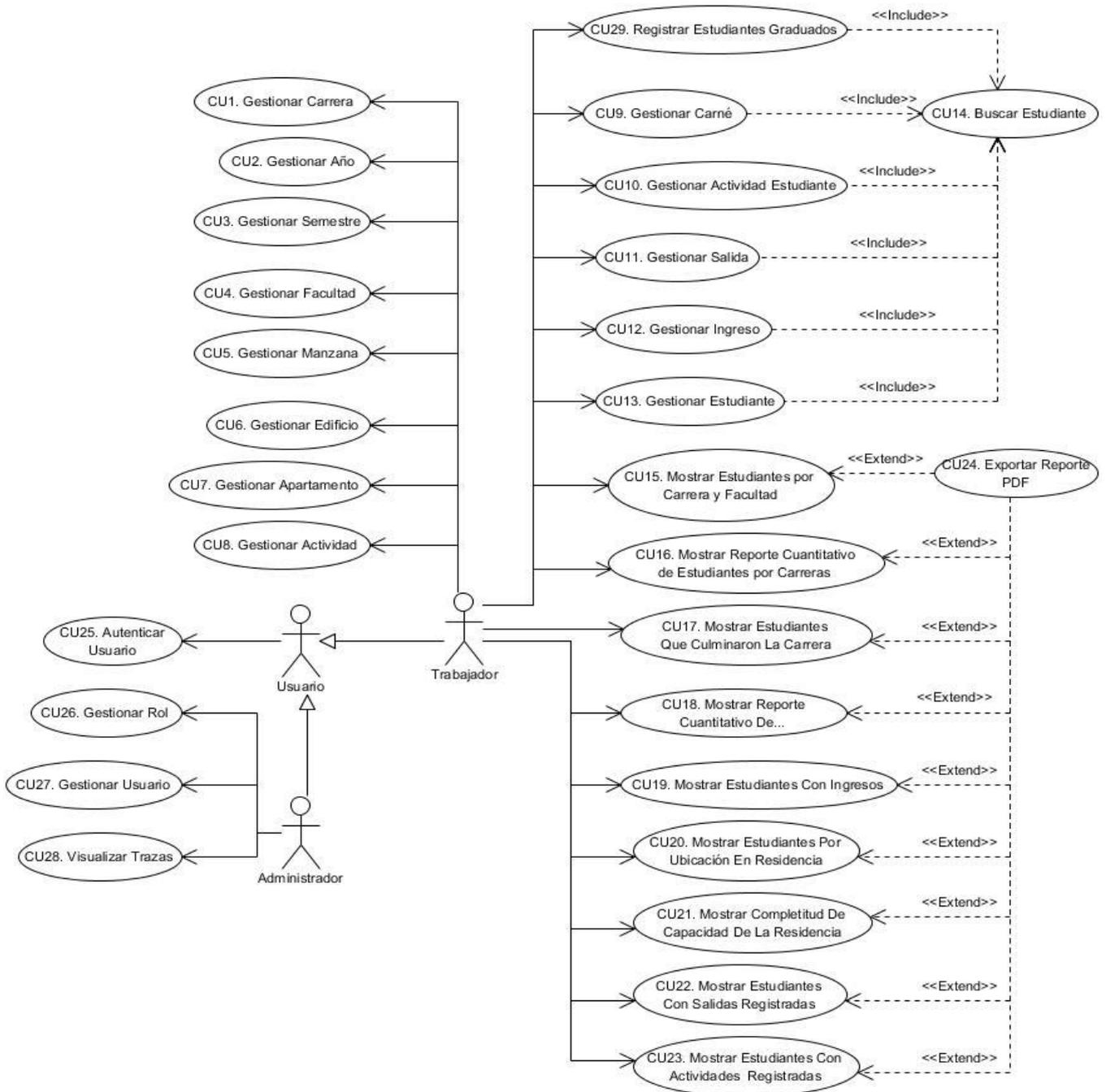


Figura 2: Diagrama de caso de uso del sistema. Fuente: Elaboración del autor.

### 2.5.3 Descripción de los casos de uso

La descripción de los Casos de Uso refleja paso a paso la forma en que interactúa el sistema y el usuario. Tienen como objetivo detallar su flujo, incluyendo como comienza y termina. Para dar cumplimiento a todos los requisitos identificados se modelaron un total de 29 CU. A continuación, se expone un ejemplo de descripción realizada para uno de los casos de uso modelados para la solución.

*Tabla 6: Descripción del CU 13. Gestionar Estudiante.*

<b>CU13. Gestionar Estudiante</b>			
<b>Objetivo</b>	Se podrá registrar, modificar, listar y mostrar detalles de estudiantes.		
<b>Actores</b>	Trabajador		
<b>Resumen</b>	Se podrá registrar un estudiante nuevo, y a través de la búsqueda de un estudiante se podrá modificar o mostrar sus datos registrados en el sistema.		
<b>Complejidad</b>	Media		
<b>Prioridad</b>	Alta		
<b>Precondiciones</b>	El usuario ha sido validado.		
<b>Postcondiciones</b>	Se registraron o se modificaron los datos a un estudiante.		
<b>Flujo de eventos</b>			
<b>Flujo básico Listar estudiantes</b>			
	<b>Actor</b>		<b>Sistema</b>
		1.	Muestra la IU sin elementos en Lista de estudiantes.
2.	El Trabajador selecciona la opción deseada: - Registrar nuevo estudiante. Ver Sección 1. - Buscar estudiante. Ver CU14. Buscar Estudiante y se pasa al paso siguiente.		
3.	El Trabajador selecciona la opción deseada en la Lista de estudiantes: - Ver detalles de estudiante. Ver Sección 2. - Modificar datos de estudiante. Sección 3. - Eliminar estudiante. Sección 4		
<b>Sección 1: "Registrar estudiante"</b>			
<b>Flujo básico Registrar estudiante</b>			
	<b>Actor</b>		<b>Sistema</b>
		1.	Muestra la IU para el registro de un nuevo estudiante.
2.	El Trabajador introduce los datos del estudiante y selecciona una de las siguientes opciones: - Aceptar. Se pasa al paso siguiente.	3.	Verifica que no haya otro estudiante registrado en el mismo número de pasaporte o nombre y apellidos:

	- Cancelar. Ver Flujo alternativo 2a de la Sección 1. Cancelar registro de estudiante.		<p>- Si no hay coincidencias, se verifica que se cumplan las validaciones y obligatoriedad de los campos registrados.</p> <ul style="list-style-type: none"> <li>• Si se cumple lo anterior se almacenan los datos y se muestra el mensaje: "Los datos fueron registrados satisfactoriamente". Se pasa para la IU del CU13. Listar Estudiantes y Termina el caso de uso.</li> <li>• Si no se cumple se muestran los mensajes de validación de los campos.</li> </ul> <p>- Si hay coincidencia, muestra el mensaje: "El elemento que trata de registrar ya existe."</p>
4.	El Trabajador selecciona la opción Aceptar. Se cierra el mensaje y se mantiene en la misma IU.		

**Flujos alternos**

**2a. Cancelar registro de estudiante.**

	Actor		Sistema
		1.	Muestra la IU del CU13. Listar Estudiantes

**Prototipo elemental de interfaz gráfica de usuario Sección 1**

**Registrar estudiante**

**Datos personales**

Primer nombre: *	Segundo nombre:	Primer apellido: *
<input type="text"/>	<input type="text"/>	<input type="text"/>
Segundo apellido: *	Sexo: *	usuario:
<input type="text"/>	--Seleccione-- ▼	<input type="text"/>
Número de pasaporte:	Número de solapín:	Pais de origen: *
<input type="text"/>	<input type="text"/>	--Seleccione-- ▼
Provincia de origen:	Municipio de origen:	Teléfono:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha de entrada a Cuba:	Fecha de salida de Cuba:	
--Seleccione--	--Seleccione--	
Fecha de entrada la Universidad:	Fecha de salida de la Universidad:	
--Seleccione--	--Seleccione--	

**Sección 2: "Mostrar detalles de estudiante"**

**Flujo básico Mostrar detalles de estudiante**

	Actor		Sistema
		1.	Muestra la IU con datos del estudiante seleccionado.
2.	El Trabajador selecciona cerrar la IU. Se muestra		

la IU CU13. Listar Estudiantes																																							
<b>Prototipo elemental de interfaz gráfica de usuario Sección 2</b>																																							
<b>Datos de estudiante</b>			<b>✕</b>																																				
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="text-align: left; padding: 5px;">Datos personales</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">Primer nombre:</td> <td style="padding: 5px;">Silvia</td> <td style="padding: 5px;">Número de carné:</td> <td style="padding: 5px;">9856321</td> </tr> <tr> <td style="padding: 5px;">Segundo nombre:</td> <td style="padding: 5px;"></td> <td style="padding: 5px;">Número de pasaporte:</td> <td style="padding: 5px;">EI98745</td> </tr> <tr> <td style="padding: 5px;">Primer apellido:</td> <td style="padding: 5px;">Lorenzo</td> <td style="padding: 5px;">Número de solapín:</td> <td style="padding: 5px;">EI98745</td> </tr> <tr> <td style="padding: 5px;">Segundo apellido:</td> <td style="padding: 5px;">Lorenzo</td> <td style="padding: 5px;">Sexo:</td> <td style="padding: 5px;">Femenino</td> </tr> <tr> <td style="padding: 5px;">usuario:</td> <td style="padding: 5px;">slorenzo</td> <td style="padding: 5px;">Pais de origen:</td> <td style="padding: 5px;">Angola</td> </tr> <tr> <td></td> <td></td> <td style="padding: 5px;">Provincia de origen:</td> <td style="padding: 5px;">Luanda</td> </tr> <tr> <td></td> <td></td> <td style="padding: 5px;">Municipio de origen:</td> <td style="padding: 5px;">Samba</td> </tr> <tr> <td></td> <td></td> <td style="padding: 5px;">Teléfono:</td> <td style="padding: 5px;">52326598</td> </tr> </tbody> </table>				Datos personales				Primer nombre:	Silvia	Número de carné:	9856321	Segundo nombre:		Número de pasaporte:	EI98745	Primer apellido:	Lorenzo	Número de solapín:	EI98745	Segundo apellido:	Lorenzo	Sexo:	Femenino	usuario:	slorenzo	Pais de origen:	Angola			Provincia de origen:	Luanda			Municipio de origen:	Samba			Teléfono:	52326598
Datos personales																																							
Primer nombre:	Silvia	Número de carné:	9856321																																				
Segundo nombre:		Número de pasaporte:	EI98745																																				
Primer apellido:	Lorenzo	Número de solapín:	EI98745																																				
Segundo apellido:	Lorenzo	Sexo:	Femenino																																				
usuario:	slorenzo	Pais de origen:	Angola																																				
		Provincia de origen:	Luanda																																				
		Municipio de origen:	Samba																																				
		Teléfono:	52326598																																				
<b>Sección 3: “Modificar estudiante”</b>																																							
<b>Flujo básico Modificar estudiante</b>																																							
	<b>Actor</b>		<b>Sistema</b>																																				
		1.	Muestra la IU para modificar el estudiante seleccionado mostrando todos los campos llenos con los datos previamente registrados.																																				
2.	El Trabajador modifica los datos deseados del estudiante y selecciona una de las siguientes opciones:  - Aceptar. Se pasa al paso siguiente. - Cancelar. Ver Flujo alternativo 2a de la Sección 3. Cancelar registro de estudiante.	3.	Verifica que no haya otro estudiante registrado en el mismo número de pasaporte o nombre y apellidos:  - Si no hay coincidencias, se verifica que se cumplan las validaciones y obligatoriedad de los campos registrados.  • Si se cumple lo anterior se almacenan los datos y se muestra el mensaje: “Los datos fueron registrados satisfactoriamente”. Se pasa para la IU del CU13. Listar Estudiantes y Termina el caso de uso.  • Si no se cumple se muestran los mensajes de validación de los campos.  - Si hay coincidencia, muestra el mensaje: “El elemento que trata de modificar ya existe.”																																				
4.	El Trabajador selecciona la opción Aceptar. Se cierra el mensaje y se mantiene en la misma IU.																																						
<b>Flujos alternos</b>																																							
<b>2a. Cancelar modificación de datos de estudiante.</b>																																							
	<b>Actor</b>		<b>Sistema</b>																																				
		1.	Muestra la IU del CU13. Listar Estudiantes																																				
<b>Prototipo elemental de interfaz gráfica de usuario Sección 3</b>																																							

Modificar estudiante

Datos personales

Primer nombre: <input type="text" value="Silvia"/>	Segundo nombre: <input type="text"/>	Primer apellido: <input type="text" value="Lorenzo"/>
Segundo apellido: <input type="text" value="Lorenzo"/>	Sexo: <input type="text" value="Femenino"/>	usuario: <input type="text" value="slorenzo"/>
Número de pasaporte: <input type="text" value="E198745"/>	Número de solapín: <input type="text" value="E198745"/>	Pais de origen: <input type="text" value="Angola"/>
Provincia de origen: <input type="text" value="Luanda"/>	Municipio de origen: <input type="text" value="Samba"/>	Teléfono: <input type="text" value="52326598"/>
Fecha de entrada a Cuba: <input type="text" value="21/08/2016"/>	Fecha de salida de Cuba: <input type="text" value="--Seleccione--"/>	
Fecha de entrada la Universidad: <input type="text" value="21/08/2016"/>	Fecha de salida de la Universidad: <input type="text" value="--Seleccione--"/>	

**Sección 4: "Eliminar estudiante"**

**Flujo básico Eliminar estudiante**

	Actor	Sistema
		1. Verifica que el estudiante no tenga registrados datos en el sistema en las entidades: Registro_Salidas, Registro_Recorrido_Estudiante, Registro_Actividades - Si no tiene registros, Se muestra un mensaje de confirmación de la acción: "Está seguro que desea eliminar el elemento". Se pasa al paso 2. - Si tiene registros, se muestra el mensaje: "El elemento que trata de eliminar tiene dependencias". Se pasa al paso 4.
2.	El Trabajador selecciona una de las acciones: - Aceptar. Se pasa al paso siguiente. - Cancelar. Se muestra la IU del CU13. Listar estudiantes.	3. Elimina el elemento seleccionado y se muestra el mensaje: "El elemento ha sido eliminado satisfactoriamente".
4.	El Trabajador selecciona la opción Aceptar. Se muestra la IU del CU13. Listar estudiantes.	

**Prototipo elemental de interfaz gráfica de usuario Sección 2**

<b>Relaciones</b>	<b>CU incluidos</b>	Buscar estudiante. Ver CU14. Buscar Estudiante
	<b>CU extendidos</b>	N/A

<b>Requisitos no funcionales</b>	N/A																																				
<b>Asuntos pendientes</b>	N/A																																				
<b>Prototipo principal de interfaz gráfica de usuario</b>																																					
<div style="border: 1px solid black; padding: 5px;"> <p><b>Buscar estudiante</b> <span style="float: right;">+</span></p> <hr/> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <input style="width: 300px; border: 1px solid #ccc;" type="text" value="nombres y apellidos"/> <input style="margin-left: 10px; border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Buscar"/> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <p style="font-size: small;">Lista de estudiantes <span style="float: right;">Cantidad: 5 ▼</span></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #f2f2f2;"> <th>No.</th> <th>Nombres y apellidos</th> <th>Pasaporte</th> <th>Pais</th> <th>Sexo</th> <th>Opciones</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Roberto López López</td> <td>IE8469</td> <td>Angola</td> <td>Masculino</td> <td> </td> </tr> <tr> <td>2.</td> <td>Luisa Pérez Pérez</td> <td>IE8469</td> <td>Angola</td> <td>Femenino</td> <td> </td> </tr> <tr> <td>3.</td> <td>Xiomara Gómez Gómez</td> <td>IE8469</td> <td>Angol</td> <td>Femenino</td> <td> </td> </tr> <tr> <td>4.</td> <td>Xiomara Gómez Gómez</td> <td>IE8469</td> <td>Angola</td> <td>Femenino</td> <td> </td> </tr> <tr> <td>5.</td> <td>Xiomara Gómez Gómez</td> <td>IE8469</td> <td>Angola</td> <td>Femenino</td> <td> </td> </tr> </tbody> </table> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span style="font-size: small;">◀◀ Página 1 de 2 ▶▶</span> <span style="font-size: x-small;">Mostrando 5 de 9</span> </div> </div> </div>		No.	Nombres y apellidos	Pasaporte	Pais	Sexo	Opciones	1.	Roberto López López	IE8469	Angola	Masculino		2.	Luisa Pérez Pérez	IE8469	Angola	Femenino		3.	Xiomara Gómez Gómez	IE8469	Angol	Femenino		4.	Xiomara Gómez Gómez	IE8469	Angola	Femenino		5.	Xiomara Gómez Gómez	IE8469	Angola	Femenino	
No.	Nombres y apellidos	Pasaporte	Pais	Sexo	Opciones																																
1.	Roberto López López	IE8469	Angola	Masculino																																	
2.	Luisa Pérez Pérez	IE8469	Angola	Femenino																																	
3.	Xiomara Gómez Gómez	IE8469	Angol	Femenino																																	
4.	Xiomara Gómez Gómez	IE8469	Angola	Femenino																																	
5.	Xiomara Gómez Gómez	IE8469	Angola	Femenino																																	

## 2.6 Modelo de datos

Un modelo de datos es el concepto de como los datos en la base de datos van a estar definidos, de tal forma que facilite el almacenamiento de información y brindar lo necesario en el momento solicitado.

Los modelos de datos definen cómo se modela la estructura lógica de una base de datos. Los modelos de datos son entidades fundamentales para introducir la abstracción en un Sistema Gestor de bases de Datos (DBMS por sus siglas en inglés). Los modelos de datos definen cómo se conectan los datos entre sí y cómo se procesan y almacenan dentro del sistema.

Un modelo de datos se refiere a las interrelaciones lógicas y el flujo de datos entre diferentes elementos de datos involucrados en el mundo de la información. También documenta la forma en que se almacenan y recuperan los datos. Los modelos de datos facilitan el desarrollo comercial y técnico de la comunicación al representar con precisión los requisitos del sistema de información y al diseñar las respuestas necesarias para esos requisitos. Los modelos de datos ayudan a representar qué datos se requieren y qué formato se utilizará para diferentes procesos comerciales. (techopedia., 2019)

En la Figura 3 se muestra el modelo de datos definido para la propuesta de solución, en su construcción se tuvo como base el MC realizado durante el modelado de negocio y la información recogida en el levantamiento de requisitos.

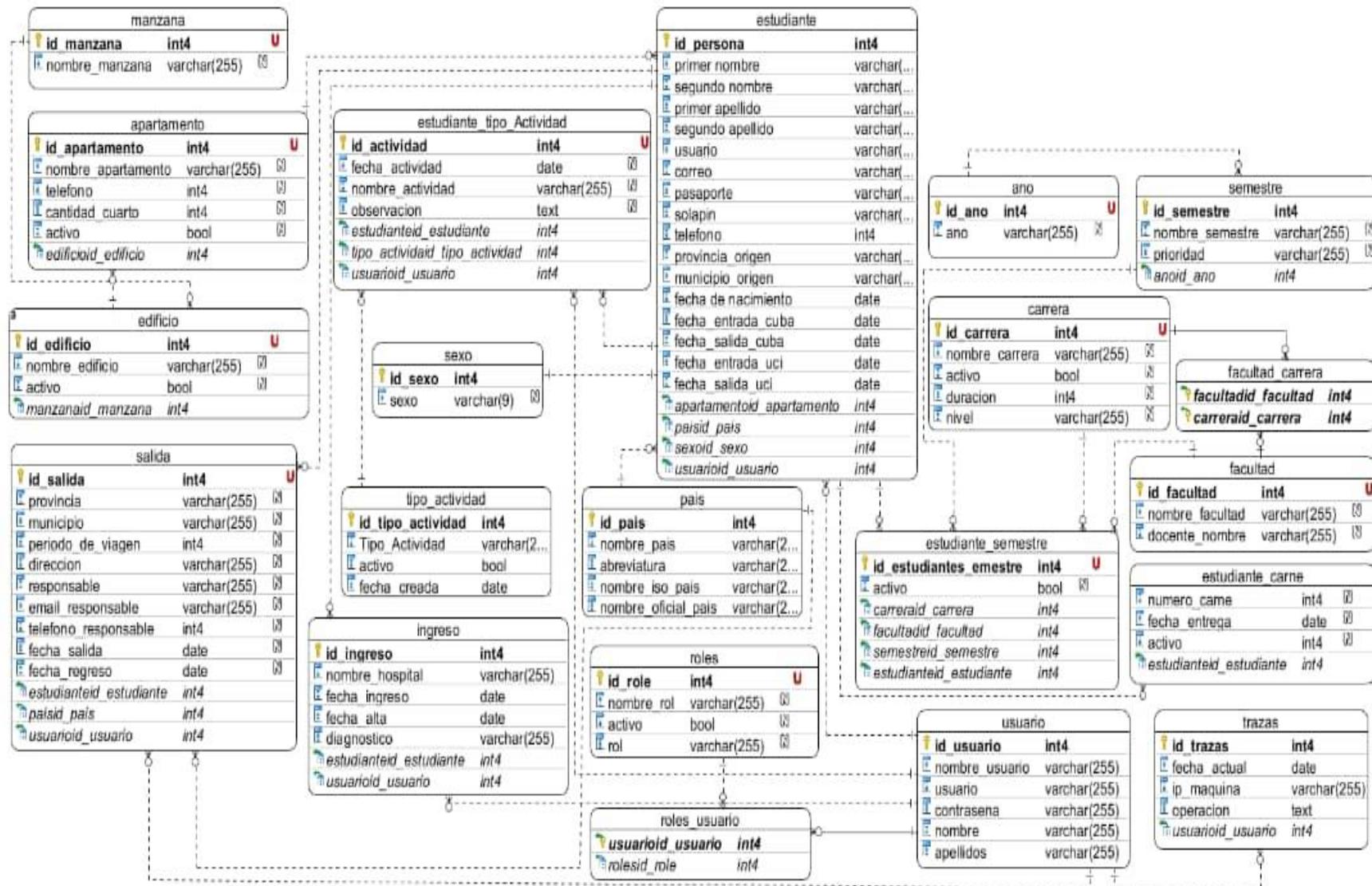


Figura 3: Modelo Entidad Relación. Fuente: Elaboración del autor.

## 2.7 Modelo de diseño

El diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software (Pressman, 2007)

### 2.7.1 Diagrama de clases del diseño

Un diagrama de clases de diseño muestra la especificación para las clases de software de una aplicación. Incluye la siguiente información:

- Clases, asociaciones y atributos
- Interfaces, con sus operaciones y constantes
- Métodos
- Navegabilidad
- Dependencias

A continuación, en la Figura 4 se muestra el diagrama de clases de diseño construido para el CU13. Gestionar Estudiante y en el

Anexo 3 se muestran los correspondientes a los demás casos de uso del sistema.

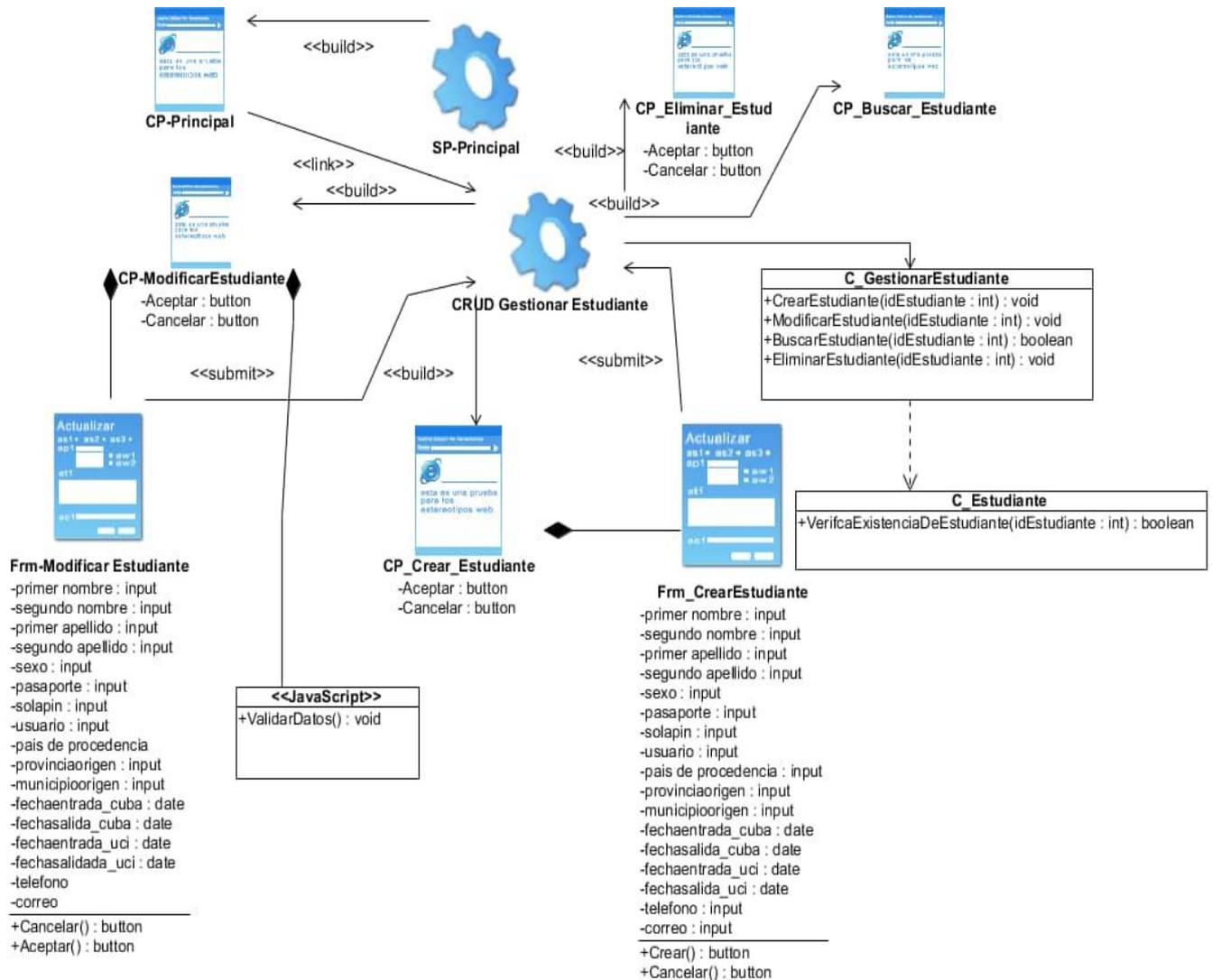


Figura 4: Diagrama de clases del diseño CU13. Gestionar Estudiante. Fuente: Elaboración del autor.

### 2.7.2 Diagrama de paquetes

Un diagrama de paquetes es la agrupación de los elementos que son modelados por UML, de manera que se facilite el entendimiento de modelos complejos, dentro de estos modelos existen elementos como clases, relaciones entre clases, diagramas de casos de uso, entre otros elementos de alto nivel. Los paquetes pueden estar anidados unos dentro de otros, y unos paquetes pueden depender de otros paquetes. Para la organización del código fuente de la aplicación, Symfony propone una estructura

predefinida de directorios. En la Figura 5 se muestra el diagrama de paquetes para el Sistema de gestión de registro de estudiantes extranjeros en la UCI. (Raquel Murillo Montesdeoca, 2015)

El directorio Controller contendrá las clases controladoras del sistema. Estas son de manera general las responsables de la gestión del sistema. Para el cumplimiento de dicha responsabilidad cuenta con el directorio Entity que alberga todas las entidades, las entidades son las clases encargadas de la abstracción de la lógica relacionada con los datos, proporcionando el uso de los mismos en dependencia del gestor de base de datos utilizado. El directorio Repository contiene las clases con las sentencias DQL (Doctrine Query Lenguaje) propio de Doctrine. El directorio Dependencia Injections en Symfony por defecto contiene los archivos necesarios para el correcto funcionamiento del Inyector de dependencias. Por último, está el directorio public que contiene todas las clases de la vista del sistema, como JavaScript y CSS.

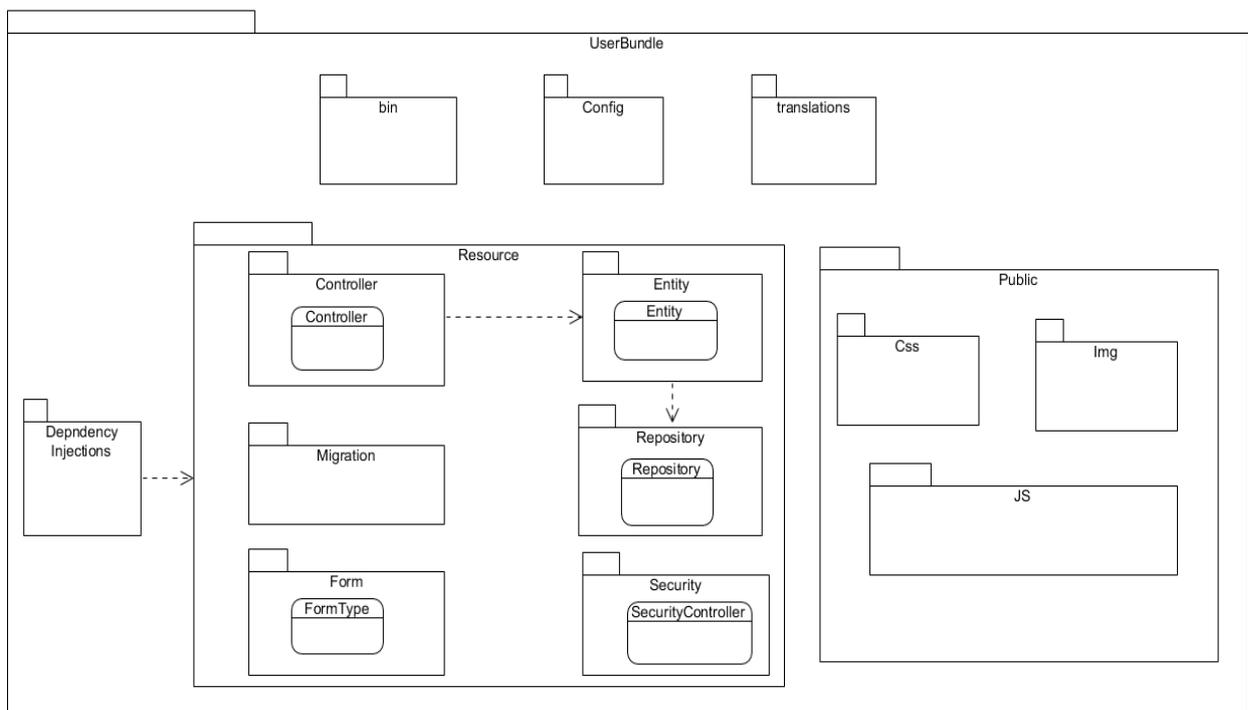
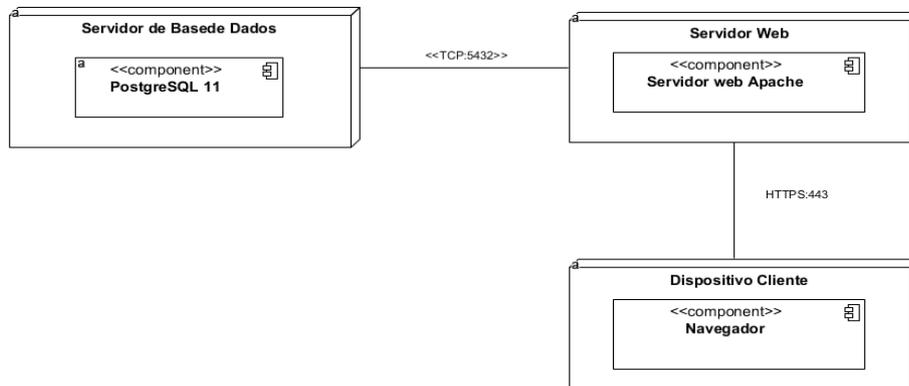


Figura 5: Diagrama de paquetes. Fuente: Elaboración del autor.

### 2.7.3 Diagrama de despliegue

El diagrama de despliegue permite modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes (nodos). Los nodos son objetos físicos que existen en tiempo de ejecución y que representan algún tipo de recurso computacional, también pueden ser dispositivos del sistema.

El diagrama de despliegue de la solución propuesta se representa por una computadora cliente que solicita peticiones mediante el protocolo HTTPS al servidor de aplicaciones: Apache, el cual se conecta al servidor de base de datos MySQL para extraer la información, como se muestra en la figura siguiente.



*Figura 6: Diagrama de despliegue. Fuente: Elaboración del autor.*

- **Servidor de Bases de Datos:** dispositivo físico que ejecuta al sistema gestor de base de datos PostgreSQL que almacena la información persistente del Sistema de gestión de registro de estudiantes extranjeros.
- **Servidor Web:** hardware donde se ejecuta el servidor encargado de dar soporte a la aplicación web.
- **Dispositivo cliente:** representa cualquier dispositivo que permita al usuario acceder a la aplicación web a través de un navegador.
- **HTTPS:** protocolo empleado para establecer a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor web.
- **TCP:** protocolo empleado para establecer la conexión entre el servidor web y el sistema gestor de base de datos usando el puerto 5432.

#### 2.7.4 Patrón arquitectónico

Los estilos arquitectónicos nos dicen, a grandes rasgos, cómo organizar nuestro código. Es el nivel más alto de granularidad y especifica capas, módulos de alto nivel de la aplicación, y cómo esos módulos y capas interactúan entre sí, las relaciones entre ellos. (Alvarez, 2019)

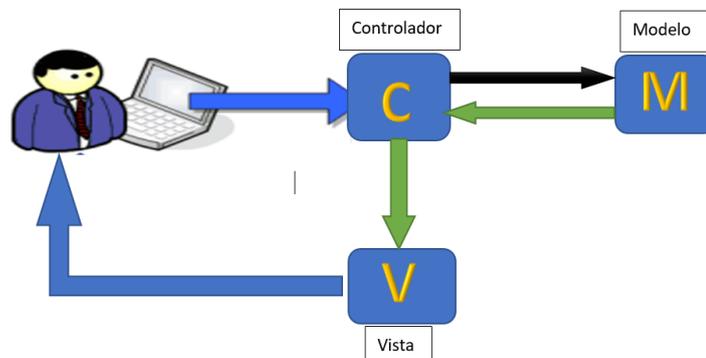
El patrón MVC (Modelo Vista Controlador) es un patrón de arquitectura de software que se ha utilizado en el desarrollo de la solución de software, este se encarga de separar la lógica de negocio de la interfaz del usuario, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla.

MVC permite dividir la aplicación en tres niveles de abstracción:

- **Modelo:** representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como intermediario con la base de datos.
- **Vista:** es la encargada de mostrar la información al usuario de forma gráfica.
- **Controlador:** es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma legible.

El funcionamiento del patrón MVC, puede resumirse en:

- El usuario realiza una petición.
- El controlador captura el evento.
- Hace la llamada al modelo o modelos correspondientes efectuando las modificaciones pertinentes sobre el modelo.
- El modelo será el encargado de interactuar con la base de datos, ya sea en forma directa, con una capa de abstracción o mediante un servicio web, y retornará esta información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista, procesa esta información creando una capa de abstracción para la lógica, quien se encargará de procesar los datos y mostrar la información a través de una interfaz gráfica.



*Figura 7: Funcionamiento de una aplicación con MVC. Fuente: Elaboración del autor.*

## 2.8 Patrones de diseño

Los patrones de diseño son el esqueleto de la solución a los problemas comunes en el desarrollo de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

En el desarrollo software, los patrones de diseño describen las clases y objetos que se comunicarán entre sí, de manera que puedan resolver un problema general de diseño en un contexto particular. En un contexto informático, un patrón de diseño es similar a conceptos como biblioteca de clases, frameworks, técnicas y herramientas de refactorización o programación extrema. (Tello, 2009)

La clasificación de los patrones de diseño en este contexto atiende a dos características como son:

- Atendiendo a su propósito
  - Creacionales: Resuelven problemas relativos a la creación de objetos.
  - Estructurales: Resuelven problemas relativos a la composición de objetos.
  - de Comportamiento: Resuelven problemas relativos a la interacción entre objetos.
- Atendiendo a su ámbito
  - Clases: Relaciones estáticas entre clases.
  - Objetos: Relaciones dinámicas entre objetos.

### 2.8.1 Patrones de diseño de asignación de responsabilidades (GRASP)

Los patrones Generales de Software para la Asignación de Responsabilidades (Grasp, por sus siglas en inglés) es un sistema orientado a objetos se compone de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones requeridas. Los diagramas de interacción describen gráficamente estas operaciones, a partir de los objetos en interacción, que se responsabilizan de una actividad determinada.

La arquitectura de Symfony está diseñada con el objetivo de que se utilicen los cinco patrones de diseño que componen esta clasificación (Controlador, Creador, Alta cohesión, Bajo acoplamiento y Experto) situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC. (Rosabal, 2019)

A continuación, se especifican los patrones que se evidencian en el sistema propuesto:

- **Experto:** su funcionamiento consiste en asignar la responsabilidad al experto en la información, es decir, a la clase que contiene la información necesaria para cumplir la responsabilidad. De esta forma se logra que las clases tengan un mejor comportamiento y hacen que las mismas sean más cohesivas con el fin de darle soporte.

Estudiante

```

getNombreEstudiante ()
setNombreEstudiante($nombre_estudiante)
getPasaporte ()
setPasaporte($pasaporte)
getCi($ci)
setCi($ci)
getProvinciaOrigen()
setProvinciaOrigen()
    
```

**Figura 8:** Patrón Experto. Fuente: Elaboración propia.

➤ **Creador:** Este patrón asigna la responsabilidad crear instancias de un objeto a una clase **A**, a una clase **B** si cumple una de estas condiciones:

- B agrega los objetos A.
- B contiene los objetos A.
- B registra las instancias de los objetos A.
- B utiliza específicamente los objetos A.
- B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).
- B es un creador de los objetos A.

Se utiliza en las clases del modelo para la creación de instancias que estén relacionadas mediante relaciones de uno a muchos, como se puede observar a continuación en la figura.

Actividad
<pre> getNombreActividad() setNombreActividad(\$nombre_actividad) getFechaActividad() setFechaActividad(\$fecha_actividad): getObservacion() setObservacion(\$observacion) getTipoActividadaidTipoActividad() setTipoActividadaidTipoActividad(\$tipo_actividadaid_tipo_actividad)                     </pre>

**Figura 9:** Patrón Creador. Fuente: Elaboración propia.

### 2.8.2 Patrones alta cohesión y bajo acoplamiento en Symfony

➤ **Alta Cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente

relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

- **Bajo Acoplamiento:** La clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.
- **Controlador:** es un objeto responsable del manejo de los eventos del sistema, que no pertenece a la interfaz de usuario, el controlador recibe la solicitud del servicio desde la capa GUI, coordina su realización y delega en otros objetos. En la siguiente figura se muestra un ejemplo de la utilización del patrón en la clase **EstudianteController** donde se maneja la información referente a la tabla **tb\_estudiante**.

```

1 /** @Route("/estudiante") ... */
2 class EstudianteController extends AbstractController
3 {
4     /** @Route("/", name="estudiante_index", methods={"GET"}) ... */
5     public function index(EstudianteRepository $estudianteRepository): Response {...}
6
7     /** @Route("/new", name="estudiante_new", methods={"GET","POST"}) ... */
8     public function new(Request $request): Response {...}
9
10    /** @Route("/{id}", name="estudiante_show", methods={"GET"}) ... */
11    public function show(Estudiante $estudiante): Response {...}
12
13    /** @Route("/{id}/edit", name="estudiante_edit", methods={"GET","POST"}) ... */
14    public function edit(Request $request, Estudiante $estudiante): Response {...}
15
16    /** @Route("/{id}", name="estudiante_delete", methods={"DELETE"}) ... */
17    public function delete(Request $request, Estudiante $estudiante): Response {...}

```

*Figura 10: Patrón Controlador. Fuente: Elaboración propia.*

### 2.8.3 Patrones de comportamiento (Gof)

Los patrones GOF (por sus siglas en inglés de The Gang of Four), describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. A continuación, se reflejan los patrones GoF emplados en el diseño del sistema:

- **Patrón Front Controller (Controlador Frontal):** Symfony posee una implementación de Controlador Frontal que despacha las peticiones hacia métodos o clases, que en la práctica son páginas controladoras. Antes del despacho, la petición es procesada por varios filtros.
- **Patrón Decorator (Decorador):** Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.
- **Patrón de diseño Singleton:** recibe su nombre debido a que sólo se puede tener una única instancia para toda la aplicación de una determinada clase, esto se logra restringiendo la libre creación de instancias de esta clase mediante el operador new e imponiendo un constructor privado y un método estático para poder obtener la instancia.
- La intención de este patrón es garantizar que solamente pueda existir una única instancia de una determinada clase y que exista una referencia global en toda la aplicación. (Guerrero, et al., 2013)

```
public function new(Request $request): Response
{
    $tipoActividad = new TipoActividad();
    $form = $this->createForm( type: TipoActividadType::class, $tipoActividad);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($tipoActividad);
        $entityManager->flush();

        return $this->redirectToRoute( route: 'tipo_actividad_index');
    }

    return $this->render( view: 'tipo_actividad/new.html.twig', [
        'tipo_actividad' => $tipoActividad,
        'form' => $form->createView(),
    ]);
}
```

*Figura 11: Patrón Controlador. Fuente: Elaboración propia.*

## 2.9 Conclusiones del capítulo

La descripción detallada del proceso de registro de estudiantes extranjeros en la UCI permitió una mejor comprensión del proceso y sus involucrados. Esto permitió la definición de los conceptos del negocio y sus relaciones, elemento base para el entendimiento del negocio y su validación con el cliente. El uso

de técnicas para la obtención de requisitos permitió comprender, identificar y describir las exigencias funcionales y no funcionales que debe cumplir la solución software. Y la confección de diagramas para las diferentes etapas del análisis y diseño, así como, la utilización de patrones de diseño y arquitectura permitió diseñar una solución ajustada a los requerimientos identificados.

## CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN

Un producto para ser entregado al cliente, necesita estar completamente desarrollado y debidamente validado según las funcionalidades acordadas con el cliente. En el actual capítulo se describe la construcción de la solución propuesta teniendo en cuenta las técnicas de programación y las estrategias de codificación. Además, se define la estrategia de pruebas a aplicar al producto una vez culminada su implementación.

### 3.1 Implementación

El objetivo fundamental de la implementación es llevar a la práctica el diseño y la arquitectura establecidos durante las fases anteriores del desarrollo de software, además, en la implementación se definen los estándares de codificación a emplear y se realizan las implementaciones de los requisitos de software.

#### 3.1.1 Diagrama de componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema de información, muestra la organización de los componentes de software, sus interfaces y las dependencias entre ellos. El diagrama de componentes describe la descomposición física del sistema de software en componentes, a efectos de construcción y funcionamiento. (Peña, et al., 2012)

En la Figura 12 se muestra el diagrama de componentes para el caso de uso “Gestionar Usuarios” el cual está conformado por 3 paquetes de implementación básicos, estos son:

- **Paquete de clases modelo:** reúne las clases que interactúan con la base de datos.
- **Paquete de clases vista:** es el encargado de agrupar los componentes que permiten la interacción directa del usuario final con el sistema.
- **Paquete de clases controlador:** contiene las clases que manipulan las acciones del usuario y se apoya en el subsistema del modelo para dar respuesta a las peticiones de la vista.

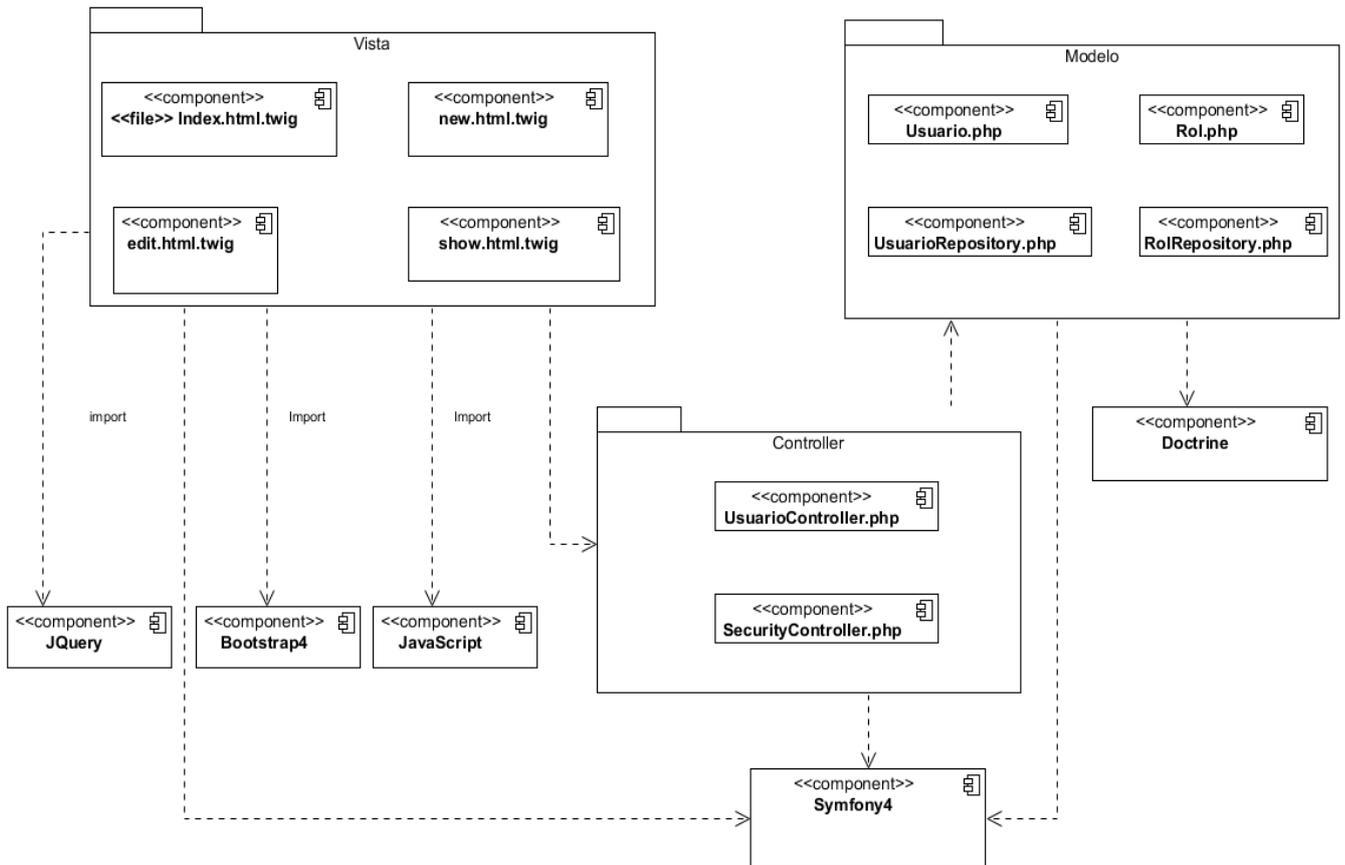


Figura 12: Diagrama de componentes del CUS Gestionar Usuario. Fuente: Elaboración del autor.

### 3.1.2 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Constituyen una guía para el desarrollo en las líneas de producción, desde el punto de vista arquitectónico, con el propósito de lograr una estandarización del código. Permiten una mejor integración entre las líneas de producción y se establecen las pautas que conllevan a lograr un código más legible y reutilizable, de tal forma que se pueda aumentar su mantenibilidad a lo largo del tiempo. (Calleja, y otros, 2016) .

Para la implementación del sistema el estándar de código utilizado es CamelCase. Se escoge CamelCase entre los muchos estándares existentes, porque es el más simples y fácil de comprender su filosofía y en las industrias de desarrollo del software es el que más se utiliza. Este es un estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos de estándares de CamelCase. (Rogers, 2014) :

- **UpperCamelCase**, cuando la primera letra de cada una de las palabras es mayúscula.

- Ejemplo: EjemploDeUpperCamelCase.
- **LowerCamelCase**, igual que la anterior con la excepción de que la primera letra es minúscula.
- Ejemplo: ejemploDeLowerCamelCase.

### 3.1.3 Nombres de clases y métodos

```

class ActividadController extends AbstractController
{
    /**
     * @Route("/", name="actividad_index", methods={"GET"})
     */
    public function index(ActividadRepository $actividadRepository): Response
    {
        return $this->render( view: 'actividad/index.html.twig', [
            'actividad' => $actividadRepository->findAll(),
        ]);
    }
}

```

Figura 13: Ejemplo de la clase y funcione index en el controlador. Fuente: Elaboración del autor.

### 3.1.4 Definición de las variables

El nombre de las variables se escribe con la primera letra en minúscula, si es un nombre compuesto se empleará la notación CamelCasing descrita anteriormente. Siempre las variables llevarán delante del nombre el símbolo de peso (“\$”) como establece el lenguaje de programación PHP, en el caso de la declaración de variables en el lenguaje JavaScript, se utiliza la palabra reservada (“var o let”) antes del nombre. Ver ejemplo en la Figura 14.

```

class Actividad
{
    /** @ORM\Id() ... */
    private $id;

    /** @ORM\Column(type="string", length=255) ... */
    private $nombre_actividad;

    /** @ORM\Column(type="date") ... */
    private $fecha_actividad;

    /** @ORM\Column(type="text") ... */
    private $observacion;
}

```

Figura 14: Definición de variables. Fuente: Elaboración del autor.

### 3.1.5 Asignación de variables y llamadas a funciones

Las asignaciones de variables se realizarán mediante el signo de igualdad (“=”) o también mediante el símbolo de flecha (“->”). Las llamadas a funciones se realizarán sin espacios entre el nombre de la función y los paréntesis, los atributos pasados por parámetro se separarán por una coma y un espacio en caso de existir más de un atributo. Ver ejemplo en la Figura 15.

```
public function new(Request $request): Response
{
    $estudiante = new Estudiante();
    $form = $this->createForm( type: EstudianteType::class, $estudiante);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {...}

    return $this->render( view: 'estudiante/new.html.twig', [...]);
}
```

Figura 15: Asignación de variables y llamadas a funciones. Fuente: Elaboración propia.

### 3.1.6 Estructuras de control

Las estructuras de control utilizadas en la implementación son: if, if else, for. Y se adicionan a la implementación los operadores binarios en casos de hacer comparaciones entre elementos necesarios para que se cumpla una determinada condición, (“==”, “!=”, “&&”, “||”, “<=”, “>=”, ...), con excepción del operador de concatenación punto (“.”). Ver ejemplo en la Figura 16.

```
<tbody>
{% for estudiante in estudiantes %}
    <tr>
        <td>{{ estudiante.id }}</td>
        <td>{{ estudiante.nombreEstudiante }}</td>
        <td>{{ estudiante.usuario }}</td>
        <td>{{ estudiante.sexoidSexo.sexo }}</td>
        <td>{{ estudiante.pasaporte }}</td>
        <td>{{ estudiante.ci }}</td>
        <td>{{ estudiante.solapin }}</td>
        {#<td>{{ estudiante.provinciaOrigen }}</td>
        <td>{{ estudiante.municipioOrigen }}</td>
        <td>{{ estudiante.fechaEntradaCuba ? estudiante.fechaEntradaCuba|date('Y-m-d') : '' }}</td>
        <td>{{ estudiante.fechaSalidaCuba ? estudiante.fechaSalidaCuba|date('Y-m-d') : '' }}</td>
        <td>{{ estudiante.fechaEntradaUci ? estudiante.fechaEntradaUci|date('Y-m-d') : '' }}</td>
        {#<td>{{ estudiante.fechaSalidaUci ? estudiante.fechaSalidaUci|date('Y-m-d') : '' }}</td>#}
        <td>{{ estudiante.getPaisidPais.getNombrePais}}</td>
        <td>
            <a href="{{ path('estudiante_show', {'id': estudiante.id}) }}"><i class="fa fa fa-eye" title="mostrar" ></i><span> &nbsp;   </span></a>
            <a href="{{ path('estudiante_edit', {'id': estudiante.id}) }}"><i class="fa fa fa-edit" title="modificar" ></i></a>
            {#<a href="{{ path('estudiante_edit', {'id': estudiante.id}) }}"><i class="fa fa fa-edit" title="editar" ></i></a>#}
        </td>
    </tr>
{% else %}
    <tr>
        <td colspan="14">no records found</td>
    </tr>
{% endfor %}
</tbody>
```

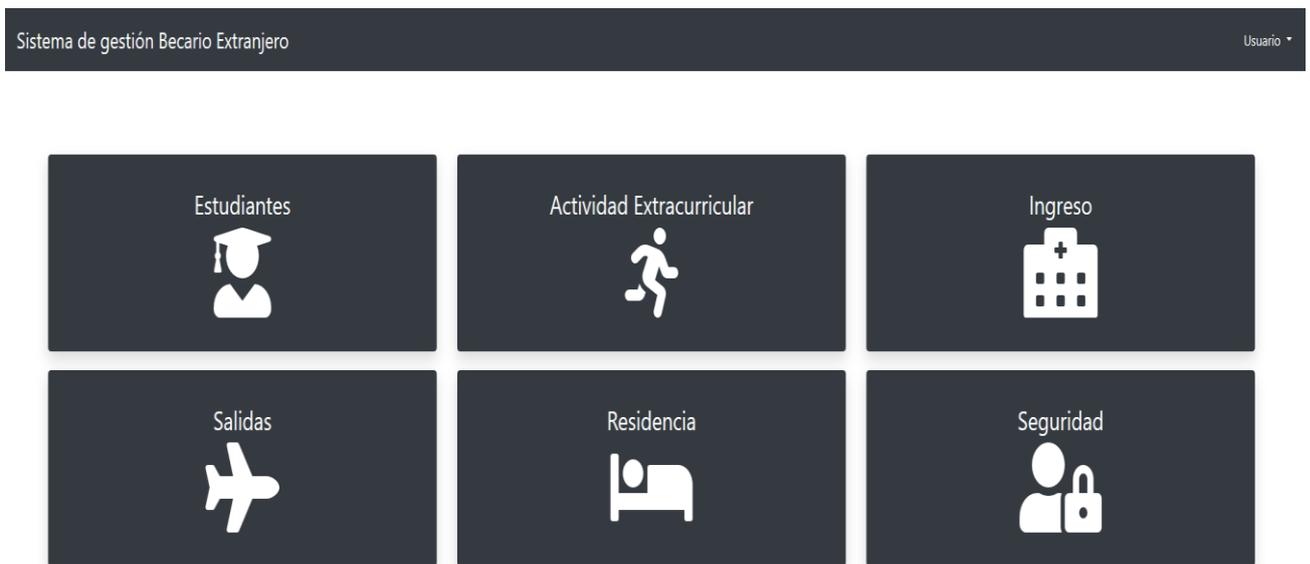
Figura 16: Estructura de control if else y for. Fuente: Elaboración propia.

### 3.2 Solución del Sistema de gestión de estudiantes extranjeros en la UCI.

Una vez analizada la información levantada, determinados y validados los requisitos de software, definida la arquitectura y modelada la base de datos, se procedió a estructurar visualmente la solución de software. En la definición de los casos de uso la información fue agrupada por momentos de registro, dando como resultado la definición de varios módulos para la aplicación, siempre teniendo en cuenta las los establecido sobre la información de los estudiantes extranjeros y dando cumplimiento a las necesidades de la DRI y dificultades que dieron lugar al problema de la investigación.

Al usuario autenticarse en la solución de software se mostrará en la interfaz principal solo aquellos módulos a los que tenga permiso, y desde donde podrá acceder a cada uno de ellos según la operación deseada. Esto es posible a través de la gestión de los roles que se les asignan a los usuarios. En la gestión de roles se incluyen las funcionalidades a las que tiene permisos el rol, así una vez que se le asigna este rol a una persona solo podrá ver esas funcionalidades.

La solución cuenta con un rol administrador del sistema que tiene los privilegios de acceso a cualquier funcionalidad, además con usuarios de nivel secundario para los que se establecen los privilegios según la accesibilidad definida para su rol.



**Figura 17:** Interfaz principal (rol administrador). Fuente: Elaboración propia.

Una vez seleccionado el acceso a alguno de los módulos se mostrará la vista principal del módulo, y un menú lateral en el cual se relacionan todas las funcionalidades del módulo. Ver la Figura 18.

The screenshot shows the 'Listado de Estudiante' (Student List) interface. At the top, there is a search bar labeled 'Buscar:'. Below it is a table with the following columns: No., Nombre Estudiante, Usuario, Sexo, Pasaporte, Cí, Solapín, País, and Opción. The table contains six rows of student data. To the right of the table is a sidebar with navigation options: Estudiante, Semestre, Año, Carrera, and Facultad. At the bottom of the table area, there is a 'Mostrar' dropdown set to '10' and 'registros', and a pagination control showing 'Anterior', '1', and 'Siguiente'.

No.	Nombre Estudiante	Usuario	Sexo	Pasaporte	Cí	Solapín	País	Opción
1	Tonito Dias Sumbelelo	tsumbelelo	m	N161624	94204000	E161616	Santa Lucia	
2	Cleusio De Oliveira Carlos	coliveira	f	N172737	945930	E161612	Angola	
3	Charar Morgan	Morgan	m	C346572	980749683	E8475992	Santa Lucia	
4	Olga João Eduardo	Ojoão	f	N234567	93766453	E124356	Angola	
5	Ludmila Carnot	Ccarnot	f	N125467	93457689	E234567	Santa Lucia	
6	Adilson Cardoso	Cardoso	m	N34568494	9234856	E346587	Angola	

Figura 18: Interfaz principal del Módulo Estudiante. Fuente: Elaboración propia.

### 3.2.1 Descripción de los módulos del sistema

La solución propuesta está compuesta por seis módulos donde se engloban todos los requisitos funcionales identificados. A continuación, se evidencia una breve descripción de cada uno de ellos:

- **Módulo Estudiante:** Permite la gestión de la información de todos los estudiantes extranjeros que cursan una carrera en la Universidad de las Ciencias Informáticas. Ver Figura 18.
- **Módulo de Actividad Extracurricular:** Permite gestionar las actividades extracurriculares en las que ha participado un estudiante extranjero y que puede formar parte de su caracterización estudiantil.
- **Módulo de Ingreso:** Permite gestionar para cada estudiante las enfermedades o padecimientos, así como ingresos por alguna causa.
- **Módulo de Salidas:** Permite gestionar las salidas de la Universidad efectuadas por un estudiante extranjero ya sean dentro o fuera del país en periodos vacacionales.
- **Módulo de Residencia:** Permite gestionar la residencia estudiantil para cada estudiante extranjero.
- **Módulo de Seguridad:** Se encarga de gestionar la seguridad del sistema, haciendo uso del control de acceso basado en roles. Esta forma de establecer la seguridad permite establecer las

funcionalidades específicas a las cuales puede acceder cada usuario; y enfoca su interés principalmente en determinar qué usuarios y qué grupos de usuarios, pueden ejecutar qué tipo de operación sobre qué tipo de recurso.

### 3.3 Pruebas de software

Las pruebas que se le aplican a un software son actividades donde un sistema o uno de sus componentes se ejecuta en circunstancias previamente evaluadas, organizadas y descritas. Las pruebas se ejecutan para tratar de descubrir que errores presenta la solución, con el fin de determinar el nivel de calidad y comprobar el grado de cumplimiento de las especificaciones iniciales del sistema. Es una buena práctica el registro de los resultados que se obtengan de cada prueba.

Según la metodología utilizada, la disciplina de pruebas se ejecuta en tres momentos:

- Pruebas internas: donde se verifica la construcción del software mediante la aplicación de pruebas al software por parte del equipo de proyecto, y se aplican las que el proyecto decida. Se pueden utilizar los diseños de casos de pruebas, las listas de chequeo y de ser posible componentes automatizados de pruebas. El principal objetivo del diseño de casos de prueba es obtener un conjunto de escenarios que tengan la mayor probabilidad de descubrir los defectos del software.
- Pruebas de liberación: estas pruebas las diseña y las aplica una entidad certificadora de la calidad externa al proyecto. Se aplica no solo al software sino a cualquier artefacto que se vaya a entregar al cliente.
- Pruebas de aceptación: Se realiza antes de comenzar el despliegue de la solución. Su objetivo es verificar que el software está listo y puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

En este trabajo se realizarán solo las pruebas para el primer momento de los antes descritos y para ellos se ha definido la siguiente estrategia de pruebas.

*Tabla 7: Estrategia de pruebas de software.*

Nivel	Tipo de prueba	Método	Técnica
Unidad	Funcional	Caja blanca	Prueba de camino básico
Sistema	Funcional	Caja negra	Particiones equivalentes

#### Métodos de prueba a utilizar en la estrategia de prueba

- **Pruebas de caja blanca:** suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, que han sido ejecutados por los casos de prueba. (Martínez, 2015)

Desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa. Mediante las pruebas de caja blanca se puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
  - Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
  - Ejecuten todos los bucles en sus límites operacionales.
  - Ejerciten las estructuras internas de datos para asegurar su validez.
- **Pruebas de caja negra:** suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como se llevan a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Para las pruebas de caja negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones, y se analizan los resultados que arroja el sistema para cada uno de los casos. En este método se verifica el sistema considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable. (Martínez, 2015)

### 3.3.1 Prueba de unidad

Las pruebas de unidad se centran en la verificación de los elementos más pequeños del software que se puedan probar examinando las estructuras de datos locales. Su objetivo es asegurar que las funcionalidades mantienen su integridad durante los pasos de ejecución de los algoritmos. Para la prueba se utiliza el método de caja blanca o estructural, descrito anteriormente, que se basa en un minucioso examen de los detalles procedimentales del código a evaluar.

Existen varias técnicas para el diseño de las pruebas de unidad que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba.

La técnica del camino básico se basa en obtener una medida de la complejidad del diseño procedimental de un programa. La medida de complejidad del procedimiento es la complejidad ciclomática y representa un límite inferior para el número de casos de pruebas que se deben realizar para asegurar que se ejecuta cada camino del programa. En la se muestra la plantilla utilizada para el diseño de los casos de prueba de unidad.

Tabla 8: Caso de prueba de unidad para el CU25. Autenticar Usuario

<b>Caso de prueba unitaria de la funcionalidad</b> Autenticar Usuario	
<b>Probador:</b> Tonito Dias Sumbelelo	
<b>Tipo de datos esperados:</b>	Booleano
<b>Código al que se aplica:</b>	
<pre> public function getUser(\$credentials, UserProviderInterface \$userProvider) {     \$token = new CsrfToken( id: 'authenticate', \$credentials['csrf_token']); //1     if (!\$this-&gt;csrfTokenManager-&gt;isTokenValid(\$token)) { //2         throw new InvalidCsrfTokenException(); //3     }      \$user = \$this-&gt;entityManager-&gt;getRepository( className: User::class)-&gt;findOneBy(['email' =&gt; \$credentials['email']]); //4      if (!\$user) { //5         // fail authentication with a custom error         throw new CustomUserMessageAuthenticationException( message: 'Email could not be found.'); //6     }      return \$user; //7 } </pre>	
<p><b>Complejidad ciclomática:</b></p> <p><b>Formula 1</b>  <math>V(G) = R</math>  <math>R = 3</math></p> <p><b>Formula 2</b>  <math>V(G) = A - N + 2</math>  <math>V(G) = 8 - 7 + 2 = 3</math></p> <p><b>Formula 3</b>  <math>V(G) = P + 1</math>  <math>V(G) = 2 + 1 \square V(G) = 3</math></p> <p><b>Caminos independientes:</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Camino Básico No 1: 1,2,3,4,6,7.</li> <li><input type="checkbox"/> Camino Básico No 2: 1,2,4,5,6,7.</li> <li><input type="checkbox"/> Camino Básico No 3: 1,2,3,4,5,7.</li> </ul>	<p><b>Representación del grafo:</b></p> <pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     2 --&gt; 4((4))     3 --&gt; 4     4 --&gt; 5((5))     5 --&gt; 6((6))     5 --&gt; 7((7))     6 --&gt; 7 </pre>

Caso de prueba para los caminos básicos	
<b>Tipo de dato esperado:</b>	Booleano
<b>Resultado de la prueba:</b>	
<b>Camino # 1</b>	
1. La contraseña es correcta el usuario tiene acceso al sistema.	
<b>Camino # 2</b>	
2. Se muestra un mensaje en pantalla especificando que hay una falla en la autenticación.	
<b>Camino # 3</b>	
3. Se muestra un mensaje en pantalla especificando que los datos están incompletos.	
<b>Evaluación del caso de prueba:</b>	Satisfactoria

### 3.3.2 Pruebas funcionales

Es un tipo de prueba de software que valida el sistema de software contra los requisitos o especificaciones funcionales. El propósito de las pruebas funcionales es probar cada función de la aplicación de software, proporcionando la entrada adecuada, verificando la salida con los requisitos funcionales. (Ares, 2019)

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplen con las funciones específicas para los que han sido creados, es decir, con los requisitos funcionales definidos. Para ello, se utiliza el método de caja negra ya que se enfocan en el funcionamiento de la interfaz del sistema a partir del estudio de sus entradas y salidas.

Para confeccionar los casos de prueba de caja negra existen distintas técnicas entre las que se encuentra la técnica de particiones equivalentes. Se basa en identificar las particiones para un sistema o componente. Las particiones de equivalencia son un conjunto de datos segregados por su validez en el sistema, donde cada uno de los miembros de los conjuntos debería ser procesado equivalentemente. Para este trabajo se define por cada requerimiento funcional del sistema un caso de prueba, donde se recogen los datos necesarios para probarlo. La Tabla 9 muestra uno de los casos de prueba diseñados para el sistema, en este caso solo para el CU1. Gestionar Carrera.

**Tabla 9:** Caso de prueba funcional para el CU1. Carrera. (parte 1)

Escenario	Descripción	Variables				
		Carrera	Duración	Nivel	Activo	Respuesta del sistema
EC 1.1 Registrar una carrera satisfactoriam ente.	Mediante este escenario el usuario puede registrar una carrera nueva en el sistema.	V	V	V	V	
		Ingeniero en Ciencias Informáticas	5	Superior	Si	Muestra el mensaje: "El elemento fue creado satisfactoriamente. "
		Alguna de las variables inválidas				

### CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN

Escenario	Descripción	Variables				
		Carrera	Duración	Nivel	Activo	Respuesta del sistema
		**	0	-	-	Muestra mensajes de validación de los campos.
<b>EC 1.2</b> Registrar una carrera nueva, pero ya existe.	Mediante este escenario el usuario trata de registrar una carrera nueva en el sistema, pero ya existe.	<b>I</b>	<b>V</b>	<b>V</b>	<b>V</b>	
		Ingeniero en Ciencias Informáticas	5	Superior	Si	Muestra el mensaje: "El elemento que trata de registrar ya existe".
<b>EC 1.3</b> Modificar una carrera satisfactoriamente	Mediante este escenario se pueden modificar los datos de una carrera previamente registrada en el sistema.	<b>V</b>	<b>V</b>	<b>V</b>	<b>V</b>	
		Ingeniero en Ciencias Informáticas	5	Superior	Si	Muestra el mensaje: "El elemento fue modificado satisfactoriamente."
		<b>Alguna de las variables inválidas</b>				
		**	0	-	-	Muestra mensajes de validación de los campos.
<b>EC 1.4</b> Modificar una carrera que existe.	Mediante este escenario el usuario trata de modificar una carrera existente en el sistema, pero coincide el nombre con el de otra ya registrada.	<b>I</b>	<b>V</b>	<b>V</b>	<b>V</b>	
		Ingeniero en Ciencias Informáticas	5	Superior	Si	Muestra el mensaje: "El elemento que trata de modificar ya existe".
<b>EC 1.5</b> Eliminar una carrera que existe con dependencias.	Mediante este escenario el usuario elimina una carrera existente que no tiene dependencias.	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	
		-	-	-	-	Muestra el mensaje: "El elemento fue eliminado satisfactoriamente".
<b>EC 1.6</b> Eliminar una carrera que existe sin dependencias.	Mediante este escenario el usuario trata eliminar una carrera existente que tiene dependencias.	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>	
		-	-	-	-	Muestra el mensaje: "El elemento que trata de eliminar tiene relaciones con otros".

**Tabla 10:** Caso de prueba funcional para el CU1. Carrera. (parte 2)

Escenario	Flujo central
<p><b>EC 1.1</b> Registrar una carrera satisfactoriamente.</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Registrar.</li> <li>- Llena los campos con los datos de la carrera.</li> <li>- Selecciona el botón Aceptar.</li> </ul>
<p><b>EC 1.2</b> Registrar una carrera nueva, pero ya existe.</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Registrar.</li> <li>- Llena los campos con los datos de la carrera, pero coloca el nombre de una carrera registrada previamente.</li> <li>- Selecciona el botón Aceptar.</li> </ul>
<p><b>EC 1.3</b> Modificar una carrera satisfactoriamente</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Modificar en la Carrera que desea modificar de la Lista de carreras, y se muestran los datos de la carrera seleccionada.</li> <li>- Cambia datos de la carrera.</li> <li>- Selecciona el botón Aceptar.</li> </ul>
<p><b>EC 1.4</b> Modificar una carrera que existe.</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Modificar en la Carrera que desea modificar de la Lista de carreras, y se muestran los datos de la carrera seleccionada.</li> <li>- Cambia datos de la carrera, pero coloca el nombre de una carrera registrada previamente.</li> <li>- Selecciona el botón Aceptar.</li> </ul>
<p><b>EC 1.5</b> Eliminar una carrera que existe sin dependencias</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Eliminar en la Carrera que desea eliminar de la Lista de carreras y la carrera no tiene relación interna con otra entidad.</li> </ul>
<p><b>EC 1.6</b> Eliminar una carrera que existe sin dependencias.</p>	<p>El usuario una vez autenticado y con permisos a la funcionalidad Carrera del módulo Estudiante.</p> <ul style="list-style-type: none"> <li>- Selecciona el módulo Estudiante y en el panel lateral derecho la funcionalidad Carrera.</li> <li>- Selecciona el botón Eliminar en la Carrera que desea eliminar de la Lista de carreras, y la carrera tiene relación interna con otra entidad.</li> </ul>

Las causas de las No Conformidades (NC) detectadas fueron:

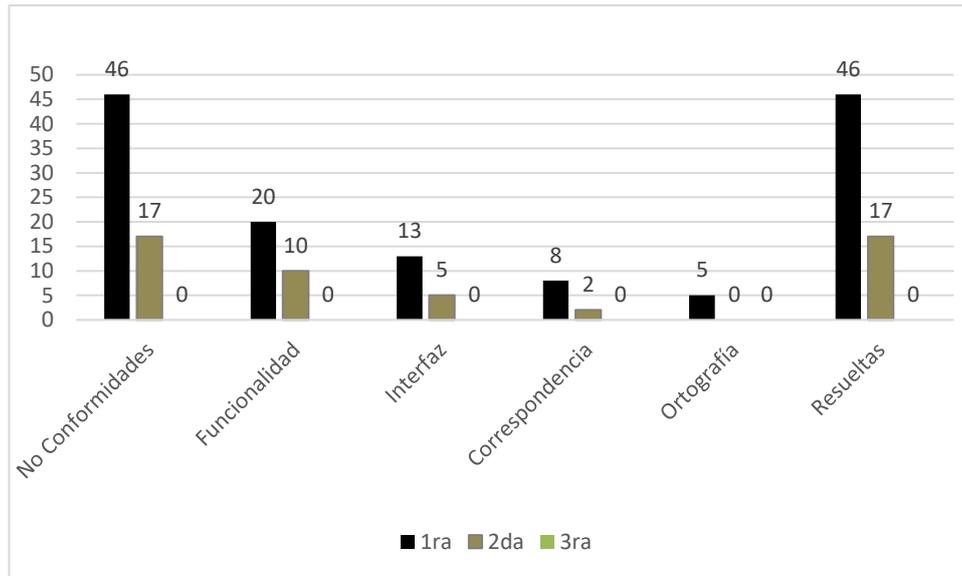
- Errores de interfaz: vistas que no cumplían con las pautas de diseño establecidas.
- Errores de ortografía: errores ortográficos en la vista y en los valores que se mostraban.
- Errores de correspondencia: el nombre de los campos no eran los más acertados para los valores que recoge, es decir, poco intuitivo.
- Errores de funcionalidad: componentes que no funcionaban correctamente.

En la tabla siguiente se muestra como trascurrieron las iteraciones de las pruebas para el caso de uso seleccionado.

**Tabla 11:** Resultados del método de caja negra mediante la técnica partición equivalencia al CU1: Gestionar Carrera.  
Fuente: Elaboración propia.

No. Iteración	NC	Funcionalidad	Interfaz	Correspondencia	Ortografía	Resueltas
<b>1ra</b>	11	3	2	3	3	<b>11</b>
<b>2da</b>	6	1	2	2	1	<b>6</b>
<b>3ra</b>	0	0	0	0	0	<b>0</b>
<b>Total</b>	<b>17</b>	<b>4</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>17</b>

Mediante la ejecución de las pruebas se logró el correcto y completo funcionamiento de las funcionalidades desarrolladas en la solución propuesta, solucionando la totalidad de las no conformidades identificadas en la 3ra iteración. En la Figura 19 se muestran un resumen de los defectos detectados y resueltos por iteración.



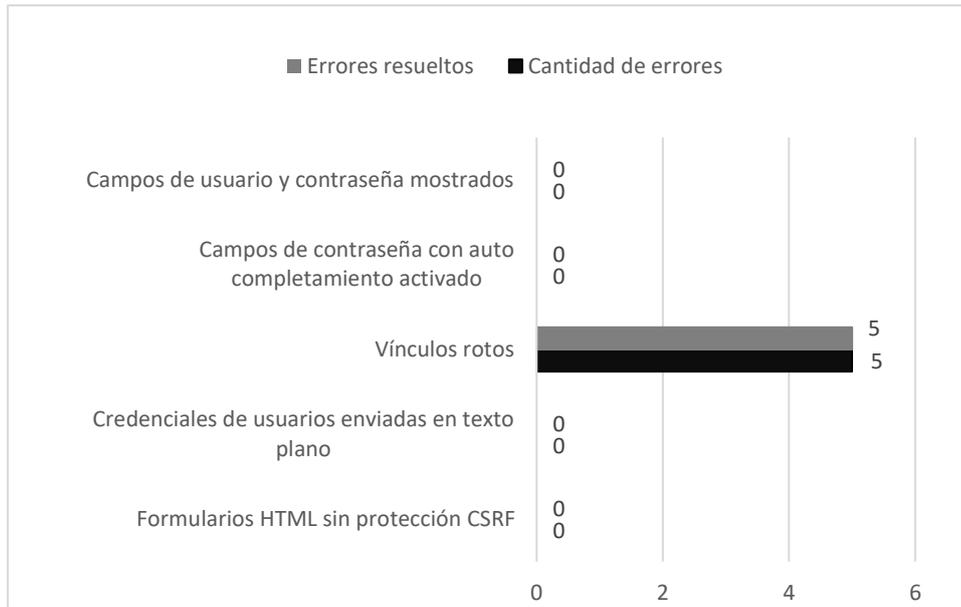
**Figura 19:** Resultados de las pruebas realizadas al software. Fuente. Elaboración propia.

### 3.3.3 Pruebas de seguridad

La prueba de seguridad se realiza para probar la seguridad de una aplicación y sus entornos cuando están sujetos a ataques por parte de un intruso externo o interno. Las pruebas de seguridad validan los servicios de seguridad de una aplicación e identifican posibles fallos y debilidades.

**Tabla 12:** Resultados de las pruebas de seguridad. Fuente: Elaboración propia.

Categorías de vulnerabilidades	Cantidad de errores	Resueltas
Formularios HTML sin protección CSRF	0	0
Credenciales de usuarios enviadas en texto plano	0	0
Vínculos rotos	5	5
Campos de contraseña con auto completamiento activado	0	0
Campos de usuario y contraseña mostrados	0	0
<b>Total</b>	<b>5</b>	<b>0</b>



**Figura 20:** Gráfico de errores detectados y resueltos en las pruebas de seguridad. Fuente. Elaboración propia.

**- Resultados de las pruebas de seguridad**

Una vez analizados los resultados obtenidos en las pruebas de seguridad se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de acciones que permitió reforzar la seguridad del Sistema. Luego se procedió a otra iteración de revisión para la cual no fueron detectados errores, quedando así la solución libre de defectos.

**3.4 Conclusiones del capítulo**

El uso de los estándares de codificación definidos y utilizados para el desarrollo del sistema, permitió mantener organizadas las actividades de programación y lograr un código fuente legible. Permitted trazar una guía para el desarrollo, organizado y sin pérdida de tiempo. Y tras la definición de la estrategia de prueba, se logró establecer una guía clara para el desarrollo organizado de las mismas; posibilitando la detección de defectos que fueron corregidos en su totalidad.

## **CONCLUSIONES GENERALES**

Con el desarrollo de la investigación se obtuvieron a las siguientes conclusiones:

- La búsqueda y análisis de características de sistemas informáticos similares al requerido, permitió determinar que era necesario el desarrollo de una solución nueva que se ajustara a las necesidades existentes en la DRI de la UCI, con relación a la gestión de la información de los estudiantes extranjeros que cursan estudio en la Institución.
- Mediante los métodos de la investigación utilizados se pudo realizar un buen levantamiento de información en la etapa inicial del trabajo, propiciando esto un correcto entendimiento y modelación del negocio, una rápida captura y validación de requisitos; así como un adecuado análisis y diseño de la solución.
- El uso de la metodología de desarrollo AUP-UCI, permitió que la ejecución del sistema se realizara de una forma organizada y que el trabajo quedara debidamente documentado.
- La selección de las herramientas y tecnologías adecuadas propició el desarrollo de una solución informática acorde a los requerimientos establecidos por el cliente y a las características del equipo de desarrollo.
- La definición de patrones y estándares de programación propició el establecimiento de un método por el cual guiarse para programar y organizar el desarrollo de la solución, brindándole al código calidad y buena legibilidad.
- La definición de una estrategia de pruebas dotó al trabajo de un camino organizado para la aplicación de las mismas y la obtención de una solución libre de defectos.

## RECOMENDACIONES

- Desarrollar la implementación de otros reportes para el sistema que sean de interés para la Dirección de Relaciones Internacionales, con el objetivo de ampliar el campo de respuesta ante la solicitud de información consolidada.
- Extender el análisis, diseño e implementación de la solución para la gestión de información de los estudiantes extranjeros de postgrado que cursan estudios en la Universidad de las Ciencias Informáticas, con el fin de seguir informatizando la Dirección de Relaciones Internacionales de la UCI.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Alvarez, Miguel Angel. 2019.** rjcodeadvance. *rjcodeadvance*. [Online] Tutoriales, junio 25, 2019. [Cited: febrero 27, 2020.] <https://rjcodeadvance.com/patrones-de-software-que-es-patron-de-arquitectura-parte-3/>.
2. **Álvaro De León. 2019.** infranetworking. *infranetworking*. [Online] Infranetworking Internacional , Juño 27, 2019. [Cited: Noviembre 28, 2019.] [blog.infranetworking.com](http://blog.infranetworking.com).
3. **Ares, Laura. 2019.** visual-engin.com. *visual-engin.com*. [Online] visual-engin.com, Mayo 8, 2019. [Cited: Mayo 13, 2020.] <http://visual-engin.com/2017/10/26/importancia-pruebas-de-software-testing/>.
4. **Asensio, Rafael Menéndez-Barzanallana. 2018.** barzana universidade Murcia. *barzana universidade Murcia*. [Online] universidade Murcia, Outubro 14, 2018. [Cited: Noviembre 24, 2019.] [www.um.es](http://www.um.es). Raf18.
5. **Bahit, Eugenia. 2011.** *El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC*. s.l.: Creative Commons AtribuciónNoComercial-SinDerivadas 3.0 Unported., 2011. 1107229727073.
6. **Bootstrap. 2019.** getbootstrap. *getbootstrap*. [Online] getbootstrap, 28 noviembre, 2019. [Cited: marzo 15, 2020.] <https://getbootstrap.com/>. get19.
7. **Calleja, Manuel Arias and González, Leyriel Zurita . 2016.** *Estándares de codificación Y Sistema de planificación de Actividades*. Havana : s.n., 2016.
8. **Campus France. 2017.** *Las misiones del Campus France*. 2017.
9. **Escalona, María José, et al. 2017.** *Ingeniería de Requisitos en Aplicaciones para la Web Un estudio comparativo Y PROCEDIMIENTO PARA ESPECIFICAR Y VALIDAR REQUISITOS DE SOFTWARE EN MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA, BASADO EN ESTUDIOS PREVIOS EN LA REGIÓN*. Sevilla ,Munich : Universidad de Sevilla Y Universidad Autónoma de Manizales, 2017.
10. **Espinosa, Alejandro Bermúdez and Camacho, Raúl Armando Alfonso. 2016.** *Sistema informático para la gestión de los procesos de la Dirección de Relaciones Internacionales en la Universidad Central Marta Abreu de Las Villas*. Villa Clara : SIRI-UCLV, 2016.
11. **Fabián, Coelho. 2017.** significados. *significados*. [Online] significados, Setembro 3, 2017. [Cited: Dezembro 4, 2019.] [www.significados.com](http://www.significados.com).
12. **Francisco José García Peñalvo. 2018.** tutoriales. *tutoriales*. [Online] tutoriales, mayo 15, 2018. [Cited: Dezembro 4, 2019.] [tutoriales.grial.eu](http://tutoriales.grial.eu). Fra19.
13. **GAIBOR, LUIS HERNAN BENAVIDES. 2011.** *GESTIÓN, LIDERAZGO Y VALORES EN LA ADMINISTRACIÓN DE LA UNIDAD EDUCATIVA SAN JUAN DE BUCAY DEL CANTON GENERAL ANTONIO ELIZALDE ( BUCAY). DURANTE PERIODO*. Loja : CENTRO UNIVERSITARIO GUAYAQUIL, 2011. CI. 0201033354.
14. **García, Diarleth. 2018.** tecnologías-información. *tecnologías-información*. [Online] IntelDig, Setembro 29, 2018. [Cited: Noviembre 13, 2019.] [www.tecnologias-informacion.com](http://www.tecnologias-informacion.com).
15. **Gauchat, Juan Diego. 2012.** *El gran libro de HTML5, CSS3 y Javascript*. Barcelona : Marcombo, 2012. 978-84-267-1 770-2.
16. **Goguen, J. and Linde, CH. 1993.** *Techniques for requirements elicitation. Proceedings of Requirements Engineering*. 1993. pp. 152-164.
17. **Guerrero, Carlos A., Suárez, Johanna M. and Gutiérrez, Luz E. 2013.** *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. Bucaramanga : Unidades Tecnológicas de Santander, 2013. S0718-07642013000300012.

18. **Gustavo. 2019.** tutoriales. *tutoriales*. [Online] hostinger, Mayo 13, 2019. [Cited: Noviembre 24, 2019.] [www.hostinger.es.tut19](http://www.hostinger.es.tut19).
19. **Hamidian Fernández, Benito Francisco and Ospino Sumoza, Germán Ramón. 2015.** *¿Por qué los sistemas de información son esenciales?* 2015.
20. **Hernandez Trasobares, Alejandro . 2017.** *Los sistemas de información: Evolución y desarrollo*. 2017.
21. **Herrera, javier Flores. 2015.** *codigofacilito*. *codigofacilito*. [Online] codigofacilito, Agosto 25, 2015. [Cited: Noviembre 24, 2019.] [codigofacilito.com](http://codigofacilito.com).
22. **Iván Gutiérrez. 2017.** *calticconsultores*. *calticconsultores*. [Online] calticc, Junio 5, 2017. [Cited: Noviembre 19, 2019.] [www.calticconsultores.com](http://www.calticconsultores.com). cal17.
23. **Izquierdo, Rachel Cristina Morales. 2016.** *Sistema web para la gestión de la información del Departamento*. Havana : UCI, 2016. Sis16.
24. **Jeremy Dick, Hull, Elisabeth and Jackson, Ken . 2017.** *Requirements Engineering*. Bern : Springer Nature, 2017. ISBN 9783319610733.
25. **jetbrains. 2019.** *jetbrains*. <https://www.jetbrains.com>. [Online] jetbrains, Diciembre 20, 2019. [Cited: Diciembre 20, 2019.] <https://www.jetbrains.com/es-es/phpstorm/features/>. jet19.
26. **Johanny Solis. 2014.** *www.arweb.com*. *arwebarweb*. [Online] arwebarweb, setiembre 26, 2014. [Cited: Marzo 15, 2020.] <https://www.arweb.com/blog/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/>. Joh14.
27. **Juan Pablo Sanz Martín.** *administracionelectronica*. *administracionelectronica*. [Online] administracionelectronica. [Cited: Dezembro 4, 2019.] [administracionelectronica.gob.es](http://administracionelectronica.gob.es).
28. **Julian Bohorquez. 2016.** *juarbo*. *juarbo*. [Online] Juarbo, Mayo 2, 2016. [Cited: Noviembre 24, 2019.] [www.juarbo.com](http://www.juarbo.com). Jul16.
29. **Julio Gonzalez Gil . 2018.** *openwebinars*. *openwebinars.net*. [Online] openwebinars, Agosto 17, 2018. [Cited: marzo 6, 2020.] <https://openwebinars.net/blog/caracteristicas-importantes-de-postgresql/>. Jul18.
30. **Kendall, K y Kendall. Mayo 2011.** *UML Diagrama de Secuencia*. Mérida : Pearson Education, Inc, Mayo 2011. 978-607-32-0577-1.
31. **Leyva, Osmar capote Vázquez y Raúl Vázquez. 2015.** *Sistema de Gestión de Información para el Vicedecanato de Investigación y Postgrado de la Facultad 2*. Havana : UCI, 2015. Osm151.
32. **Loidi, José Ramón Saborido and Yero, Denisse Pereira. 2018.** *REGULAMENTO PARA ESTUDIANTES EXTRANJEROS EN LOS CENTROS DE EDUCACIÓN CUBANOS*. Havana : Mes, 2018.
33. **Malan, Ruth, et al. 2014.** *Functional Requirements and Use Cases*. Ecuador : hp.com , EDILOJA Cía Ltda, 2014. SBN-978-9942-08-212-1.
34. **Martillo Mero, Sergio Alejandro. 2014.** *SISTEMA INFORMÁTICO DE CONTROL DE EJECUCIÓN DE CONVENIOS Y BECAS DE LA UNIDAD DE COOPERACIÓN INTERINSTITUCIONAL DE LA UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO*. Quevedo Ecuador : Universidad de Quevedo, 2014. 1 204309262.
35. **Martínez, Eduardo Salazar. 2015.** *InformáticaJuridica.com*. *InformáticaJuridica.com*. [Online] InformáticaJuridica.com, Enero 1, 2015. [Cited: Mayo 13, 2020.] [informaticaJuridica.com//Procedimiento para realizar pruebas de Caja Blanca](http://informaticaJuridica.com//Procedimiento para realizar pruebas de Caja Blanca).
36. **Mediavilla, Elena. 2010.** *Programacion orientada a objetos. Master de computacion.Modelado de casos de uso*. 2010.
37. **Mesa, Anileidy Basso and Almaguer, Dayaima Gómez. 2009.** *Sistema automatizado para el proceso de caracterización de los estudiantes*. Havana : UCI, 2009.

38. **nmp.** 2020. <https://www.npmjs.com>. *nmp*. [Online] npm, enero 3, 2020. [Cited: febrero 2, 2020.] <https://www.npmjs.com/package/datamodel>. htt20.
39. **Paloma Torres.** 2017. [www.dealerworld.es](http://www.dealerworld.es). [Online] Noviembre 17, 2017. [Cited: Novembor 11, 2019.]
40. **Peña Ayala, Alejandro.** 2006. *Ingeniería de Software. Una guía para crear Sistemas de Información*. 1ra. México : 2006, 2006. 970-94797-0-9.
41. **Peña, Onel Salinas and Pérez Bandomo, Dariel Ricardo.** 2012. *Modernización del proceso tramitación de carné de extranjeros por asuntos particulares de la Dirección de Inmigración y Extranjería de la República de Cuba*. havana : tesis, 2012.
42. **Pérez, Javier Eguíluz.** 2009. *Introducción a CSS*. 2009.
43. **Polo, Marbier Pérez.** 2017. *Módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de*. Havana : s.n., 2017.
44. **postgresql.** 2019. [www.postgresql.org](http://www.postgresql.org). *www.postgresql.org*. [Online] [www.postgresql.org](http://www.postgresql.org), 2019. [Cited: marzo 6, 2020.] <https://www.postgresql.org/about/>. pos20.
45. **Pressman, R.S.** 2007. *Ingeniería del software. Un enfoque práctico*. 2007.
46. **Rama, Teresa Estévez.** 2017. [itblogsogeti.com](http://itblogsogeti.com). *itblogsogeti.com*. [Online] [itblogsogeti.com](http://itblogsogeti.com), Marzo 23, 2017. [Cited: Mayo 13, 2020.] <https://itblogsogeti.com/2017/03/23/pruebas-de-carga-y-estres/>.
47. **Raquel Murillo Montesdeoca.** 2015. *exa. exa*. [Online] *exa*, julio 8, 2015. [Cited: abril 26, 2020.] [exa.unne.edu.ar](http://exa.unne.edu.ar).
48. **Ricoarango., Marialis Soto.** 2012. *Servidor para la gestión de la información de Becarios Extranjeros del Ministerio de Comercio Exterior y la Inversión Extranjera*. Artemisa : Facultad Regional Mártires de Artemisa , 2012.
49. **Rivero, Eblis Manuel Caro.** 2015. *Sistema para la gestión de información del Departamento de Reproducción de Documentos de la Universidad de las Ciencias*. Havana : UCI, 2015.
50. **Rocio Camargo Villa.** 2015. [es.slideshare.net](http://es.slideshare.net). *es.slideshare.net*. [Online] Mayo 13, 2015. [Cited: MAyo 13, 2020.] <https://es.slideshare.net/mrocio1/diseo-caso-de-pruebas-48099189>.
51. **Rogers, Jessica.** 2014. *File Naming Standards for Digital Collections*. 2014.
52. **Rosabal, Andy González.** 2019. *Módulo para recomendar al usuario cursos y contenidos en la Plataforma Educativa ZERA*. Havana : UCI, 2019. Ros19.
53. **Tello, Jesús Cáceres.** 2009. *Revista de Educación a Distancia. Revista de Educación a Distancia*. [Online] intercambio de información e ideas acerca de investigaciones , Enero 15, 2009. [Cited: Febrero 21, 2020.] <https://revistas.um.es/red/article/view/89351>. 1578-7680 .
54. **Tinoco, Richard Dios.** 2011. *Sistemas*. Llanos : UNIVERSIDAD POLITECNICA TERRITORIAL DE LOS LLANOS, 2011.
55. **Turrado, Jorge.** 2020. [www.campusmvp.es](http://www.campusmvp.es). *www.campusmvp.es*. [Online] [www.campusmvp.es](http://www.campusmvp.es), Marzo 10, 2020. [Cited: Mayo 13, 2020.]
56. **UCI, Actividad productiva.** 2015. *Metodología de desarrollo para la actividad productiva de la UCI. Programa de mejora*. Universidad de las Ciencias Informáticas. 2015.
57. **Universidad de Cordoba.** 2019. Universidad de Cordoba. *Universidad de Cordoba*. [Online] Universidad de Cordoba, enero 15, 2019. [Cited: Dezembro 4, 2019.] [www.uco.es](http://www.uco.es).
58. **Vargas, Mariela.** 2017. [losandestraining.com](http://losandestraining.com). *losandestraining.com*. [Online] Los Andes Training, Agosto 23, 2017. [Cited: Mayo 13, 2020.] <https://losandestraining.com/2017/08/23/que-son-las-pruebas-de-aceptacion/>.

59. **Vázquez, Osmar Capote and Leyva, Raúl Vázquez. 2015.** *Sistema de Gestión de Información para el Vicedecanato de Investigación y Postgrado de la Facultad 2.* dirección de formación post-graduada , Universidad de las ciencias Informaticas. Havana : UCI, 2015. Osm.
60. **visual-paradigm. 2019.** *www.visual-paradigm.com. www.visual-paradigm.com.* [Online] *www.visual-paradigm.com, 1 1, 2019.* [Cited: Enero 11, 2020.] <https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/>. vis19.
61. **Zapata, C. M. and Arango, F. 2014.** *Alineación entre metas organizacionales y elicitación de requisitos del software.* 2014. 143:101-110.

## ANEXOS

## Anexo 1: Diseño de entrevista

<b>Entrevistado:</b>	
<b>Cargo:</b>	
<b>Especialista funcional en:</b>	
<b>Responsabilidad en el proceso:</b>	
<b>Preguntas:</b>	
<ul style="list-style-type: none"> <li>• ¿Cómo se organiza el proceso de gestión de la información relacionada con los estudiantes extranjeros en la DRI?</li> <li>• ¿Quién es o son las personas responsables en la DRI de recibir la información de los estudiantes extranjeros?</li> <li>• ¿Quién es o son las personas responsables de entregar en las facultades la información de los estudiantes extranjeros a la DRI?</li> <li>• ¿Quién es o son las personas responsables por la residencia de entregar la información relacionada con los estudiantes extranjeros a la DRI?</li> <li>• ¿Quién es o son las personas que por parte de la atención de salud son las responsables de entregar la información a la DRI?</li> <li>• ¿Qué información consolidada entrega la DRI en los reportes de los estudiantes extranjeros?</li> <li>• ¿Qué información es necesaria registrar de los estudiantes extranjeros? <ul style="list-style-type: none"> <li>➤ En el registro del estudiante.</li> <li>➤ De las entradas y salidas.</li> <li>➤ De las actividades extracurriculares.</li> <li>➤ De los ingresos y enfermedades</li> </ul> </li> <li>• ¿Qué prioridad e importancia le concede a cada grupo de información?</li> <li>• ¿Pudiera darle un orden de prioridad a la información que recibe de los estudiantes extranjeros de pregrado?</li> <li>• ¿Qué formato tienen los informes que les solicitan con en consolidado de información referente a los estudiantes?</li> </ul>	

## Anexo 2: Descripción de conceptos del Modelo conceptual (MC).

- **Registro\_Recorrido\_Estudiente**

<b>Descripción</b>	Representa el recorrido por las facultades que ha tenido el estudiante durante el transcurso de la carrera.					
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>estudiante</b>	Estudiante que se relaciona con el recorrido.	<b>Estudiante</b>	No	Sí	-	-
<b>facultad</b>	Facultad en la que se encuentra cursando la carrera el estudiante.	<b>Facultad</b>	No	Sí	-	-
<b>carrera</b>	Carrera en la que se encuentra cursando estudios el estudiante.	<b>Carrera</b>	No	Sí	-	-
<b>semestre</b>	Semestre dentro de la carrera en el que se encuentra cursando estudios el estudiante.	<b>Semestre</b>	No	Sí	-	-
<b>activo</b>	Se refiere a si el estudiante esta cursando el semestre que se relaciona	numérico	No	Sí	- 0 (no está activo) - 1(está activo)	- Diferente a las clases válidas.

- **Facultad**

<b>Descripción</b>	Representa las unidades organizativas dentro de la Universidad donde los estudiantes reciben su formación docente.					
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>facultad</b>	Nombre de la facultad.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros.	- Diferente a las clases válidas.
<b>docente</b>	Docente en el que está la facultad.	cadena de caracteres (50)	No	Sí	- Letras - Números enteros.	- Diferente a las clases válidas.

- **Carrera**

Descripción		Representa las carreras que se imparten en la Universidad.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>carrera</b>	Nombre de la carrera.	cadena de caracteres (50)	No	Sí	- Letras. - Números enteros.	- Diferente a las clases válidas.
<b>duracion</b>	Duración en años	numérico	No	Sí	- Números relaes	- Diferente a las clases válidas.
<b>nivel</b>	Nivel para el que se gradúa el estudiante que cursa la carrera.	cadena de caracteres (50)	No	Sí	- Letras.	- Diferente a las clases válidas.
<b>activo</b>	Se refiere a si la carrera está activa o no, que si se puede usar por el negocio.	numérico	No	Sí	- 0 (no está activa) - 1(está activa)	- Diferente a las clases válidas.

- **Semestre**

Descripción		Representa los semestres que puede tener una carrera.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>semestre</b>	Nombre del semestre.	cadena de caracteres (20)	No	Sí	- Letras. - Números enteros.	- Diferente a las clases válidas.
<b>ano</b>	Año al que pertenece el semestre	<b>Año</b>	No	Sí	-	-

- **Año**

Descripción		Representa los años que puede tener una carrera, relacionando dentro de ellos los semestres.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>ano</b>	Nombre del año de una carrera.	cadena de caracteres (10)	No	Sí	- Letras. - Números enteros.	- Diferente a las clases válidas.
<b>carrera</b>	Nombre de la carrera.	<b>Carrera</b>	No	Sí	-	-
<b>semestre</b>	Semestres que la carrera tiene en el año que se relaciona.	<b>Semestre</b>	No	No	-	-

- **Manzana**

<b>Descripción</b>		Representa las manzanas en las que se divide la residencia estudiantil.				
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
manzana	Número de la manzana.	numérico (hasta 2 dígitos)	No	Sí	- Números enteros.	- Diferente a las clases válidas.

- Edificio

<b>Descripción</b>		Representa los edificios que pertenecen a la residencia de estudiantes.				
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
edificio	Número del edificio.	numérico (hasta 3 dígitos)	No	Sí	- Números enteros.	- Diferente a las clases válidas.
manzana	Número de la manzana.	<b>Manzana</b>	No	Sí	-	-
activo	Se refiere a si el edificio está activo, si se puede usar por el negocio.	numérico	No	Sí	- 0 (no está activo) - 1 (está activo)	- Diferente a las clases válidas.

- Apartamento

<b>Descripción</b>		Representa los apartamentos de la residencia estudiantil donde pueden estar ubicados los estudiantes.				
<b>Atributos</b>						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
apartamento	Número del apartamento.	numérico (hasta 6 dígitos)	No	Sí	- Números enteros.	- Diferente a las clases válidas.
edificio	Número del edificio al que pertenece el apartamento.	<b>Edificio</b>	No	Sí	-	-
telefono	Número de teléfono que tiene el apartamento.	numérico (hasta 4 dígitos)	No	Sí	- Números enteros.	- Diferente a las clases válidas.
cant_cuartos	Cantidad de cuartos	numérico (1 dígito)	No	Sí	- Números enteros.	- Diferente a las clases válidas.
activo	Se refiere a si el apartamento está activo, si se puede usar por el negocio.	numérico	No	Sí	- 0 (no está activo) - 1 (está activo)	- Diferente a las clases válidas.

- Ingreso

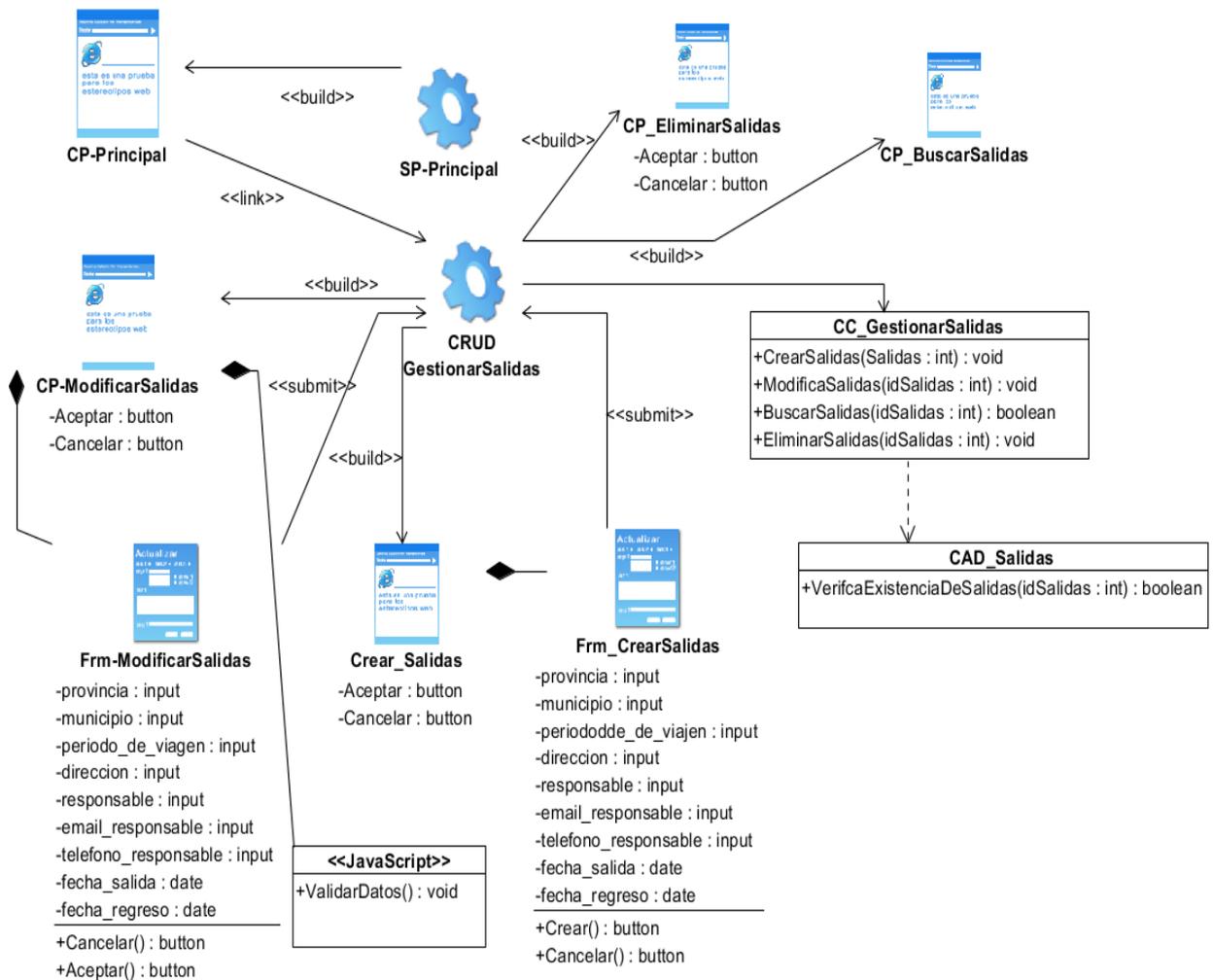
Descripción		Representa los ingresos que ha tenido un estudiante.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>estudiante</b>	Estudiante que se relaciona con el ingreso	<b>Estudiante</b>	No	Sí	-	-
<b>fecha_ingreso</b>	Fecha de inicio del ingreso.	fecha	No	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>fecha_alta</b>	Fecha de inicio del ingreso.	fecha	No	Sí	- Fecha (día, mes, año)	- Diferente a las clases válidas.
<b>diagnostico</b>	Motivo o causa por la que fue ingresado el estudiante.	cadena de caracteres (100)	No	Sí	- Letras. - Números enteros.	- Diferente a las clases válidas.
<b>hospitalizacion</b>	Nombre del lugar donde fue hospitalizado el estudiante.	cadena de caracteres (100)	No	Sí	- Letras. - Números enteros.	- Diferente a las clases válidas.
<b>semestre</b>	semestre dentro de la carrera en el que fue hospitalizado.	numérico (4 dígitos)	No	Sí	- Números enteros.	- Diferente a las clases válidas.

- **Actividad\_Extracurricular**

Descripción		Representa las actividades extracurriculares en las que puede participar un estudiante.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
<b>actividad</b>	Nombre de la actividad.	cadena de caracteres (50)	No	Sí	- Letras.	- Diferente a las clases válidas.
<b>tipo_actividad</b>	Tipo de la actividad.	cadena de caracteres (50)	No	Sí	- Letras.	- Diferente a las clases válidas.

### Anexo 3: Diagrama de clases del diseño

- Diagrama de clases del diseño CU11. Gestionar Salidas.



• Diagrama de clases del diseño CU8. Gestionar Actividad

