



**Facultad 1**

# **Sistema de Gestión Económica de la**

# **Facultad 1**

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

**Autor:** Rocío del Carmen Ortiz Vega

**Tutor:** Prof. Aux. MSc. Geidis Sánchez Michel.

La Habana, diciembre de 2022

Año 64 de la Revolución



**DECLARACIÓN DE AUTORÍA**

Yo Rocío del Carmen Ortiz Vega, con CI: 98122418177, con título de trabajo de diploma “*Sistema de gestión económica de la Facultad 1*”, concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año \_\_\_\_.

Rocío del Carmen Ortiz Vega

---

Firma del Autor

Geidis Sánchez Michel

---

Firma del Tutor

## **DATOS DE CONTACTO**

Aux.MSc. Geidis Sánchez Michel, profesora de Redes y Seguridad Informática del departamento de Sistemas Digitales de la Facultad 1, máster en Gestión de Proyectos Informáticos y Vicedecana de Economía y Administración.

## AGRADECIMIENTOS

*Gracias a mi novio, por su ayuda incondicional, por darme la certeza de que puedo contar siempre con él y por haberme aguantado en los momentos más tormentosos de este proceso. Gracias por tanto amor, dedicación y por ser un apoyo fundamental en mi vida.*

*A todas mis amigas Elizabeth, la Chuki, Brenda, Néida, Enmanuel, Gustavo, a la gente del 107 por siempre estar a mi lado, por apoyarme a lo largo de este camino y por cuidarme siempre como mi familia.*

*A Maydelis gracias por estar en mi vida, por acompañarme siempre, por compartir todo conmigo y siempre tenerme presente. Gracias por sus palabras de aliento, por sufrir con todos mis problemas, por siempre contar con su ayuda, por ser como una hermana para mí.*

*Gracias a toda mi familia, a mi abuela Elva, a mi papá, a mis hermanitas, a mi tía Gloria, a Dalia y a mi prima Clarita por formar parte importante de mi vida y darme tanto amor.*

## DEDICATORIA

*Todo en la vida te lo debo a ti Mamá, gracias por estar siempre en los momentos difíciles de mi vida y sufrirlos aún más que yo misma, por ser un ejemplo para mí, por tu fortaleza y por siempre darme lo mejor de ti. Esta tesis es el resultado de lo que me has enseñado en la vida, ya que siempre has sido una persona honesta y entregada a tu trabajo y a tu familia. Es por ello que hoy te dedico este trabajo de tesis. Gracias por confiar en mí y darme la oportunidad de culminar esta etapa de mi vida.*

*A mi Hermano mayor, por ser mi amigo y compañero que me ha ayudado a crecer, gracias por estar siempre conmigo en todo momento. Gracias por la paciencia que has tenido para enseñarme, por el amor que me das, por tus cuidados en el tiempo que hemos vivido juntos. Gracias por estar pendiente durante toda esta etapa. A ti también te dedico este trabajo de diploma.*

*También le dedico este trabajo a mi tutora Geidis, sin tus palabras de aliento y tu ayuda no hubiera podido lograrlo, gracias por ser más que mi tutora una mi amiga.*

## RESUMEN

En la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), en el Vicedecanato de Economía y Administración la gestión de los procesos (dieta, bonificación de pasaje, préstamo estudiantil, alumno ayudante y solicitudes de almacén) se realizan de forma manual, lo que provoca demoras en la ejecución del trabajo, pérdida y duplicación de la información. Teniendo en cuenta esta situación se define como objetivo general: desarrollar un sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1. Como base tecnológica fundamental para la propuesta de solución se utiliza Python 3.1.0.5 como lenguaje de programación, Django 4.01 como Framework , SQLite 3.40.0 como gestor de base de datos y Apache 2.4.46 como servidor web. La metodología de desarrollo de software empleada es AUP-UCI en el Escenario No.2. La arquitectura de software definida es Modelo – Vista - Controlador. Se muestra, además, el resultado de la estrategia de validación que fue aplicada a la propuesta de solución, evidenciándose un producto funcional.

**Palabras clave:** bonificación de pasaje, dietas, gestión económica, préstamo estudiantil.

## ABSTRACT

*In Faculty 1 of the University of Informatics Sciences (UCI), in the Vice Dean of Economics and Administration, the management of the processes (diet, ticket bonus, student loan, student assistant and warehouse) are carried out manually, which causes delays in the execution of work, loss and duplication of information. Taking this situation into account, the general objective is defined: to develop a system that contributes to the management of economic control in the Vice-dean of Economics and Administration of Faculty 1. Python 3.1 is used as the fundamental technological base for the proposed solution.05 as a programming language, Django 4.0.1 as a Python Framework, SQLite 3.40.0 as a database manager and Apache 2.4.46 as web server. The software development methodology used is AUP-UCI in Scenario No.2. The defined software architecture is Model- View – Controller. It also shows the result of the validation strategy that was applied to the solution proposal, evidencing a functional product.*

*Keywords: allowances, economic management, student loan, ticket bonus.*

**Índice**

<b>INTRODUCCIÓN .....</b>	<b>10</b>
<b>CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL SISTEMA DE GESTIÓN ECONÓMICO DE LA FACULTAD 1 .....</b>	<b>15</b>
I.1 Conceptos asociados a la investigación .....	15
I.2 Análisis de las soluciones existentes .....	19
1.2.1 Ámbito internacional .....	20
1.2.2 Ámbito nacional .....	20
I.3 Análisis de las metodologías, herramientas y tecnologías a utilizar para el desarrollo del sistema.....	22
1.3.2 Herramientas y tecnologías .....	23
Conclusiones del capítulo.....	27
<b>CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: SISTEMA DE GESTIÓN ECONÓMICA DE LA FACULTAD 1 .....</b>	<b>28</b>
II.1 Descripción de la propuesta de solución .....	28
II.1.1 Modelo Conceptual.....	28
II.2 Requisitos, Análisis y Diseño de la solución propuesta .....	30
II.2.1 Requisitos Funcionales.....	30
II.2.2 Requisitos No Funcionales .....	34
II.3 Modelo de caso de uso del sistema.....	35
II.3.1 Descripción de los Casos de Uso del Sistema.....	36
II.4 Diagrama de Secuencia .....	43
II.5 Modelo de Datos .....	44



II.4 Estilo Arquitectónico .....	45
II.4.1 Patrón Arquitectónico .....	46
II.4.2 Patrón de Diseño .....	47
Conclusiones del capítulo.....	49
<b>CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN ECONÓMICO DE LA FACULTAD 1.....</b>	<b>50</b>
III.1 Diagrama de componente .....	50
III.2 Estándares de codificación.....	55
III.3 Diagrama de Despliegue .....	58
III.4 Evaluación de la propuesta de solución.....	59
III.4.1 Pruebas de Caja Negra .....	60
III.4.2 Pruebas de Seguridad .....	66
III.4.3 Pruebas de Rendimiento .....	67
III.5. Pruebas de aceptación .....	69
III.5.1. Satisfacción de los usuarios .....	69
Conclusiones del capítulo:.....	72
<b>CONCLUSIONES FINALES .....</b>	<b>73</b>
<b>RECOMENDACIONES.....</b>	<b>74</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>75</b>
<b>ANEXOS.....</b>	<b>79</b>

**ÍNDICE DE TABLAS**

Tabla 1: Variable Independiente .....	12
Tabla 2: Variable Dependiente.....	12
Tabla 3: Comparación de sistemas homólogos.....	21
Tabla 4: Requisitos Funcionales .....	30
Tabla 5: Caso de uso autenticar usuario.....	36
Tabla 6: Caso de uso Gestionar Alumno Ayudante.....	37
Tabla 7: Descripción del Diagrama de Componente Gestionar Alumno Ayudante .....	51
Tabla 8: Descripción del Diagrama de Componentes Préstamo Estudiantil .....	52
Tabla 9: Descripción del Diagrama de Componentes Bonificación de Pasaje.....	54
Tabla 10: Tipos de Pruebas .....	59
Tabla 11: Descripción del caso de uso: Crear Alumno Ayudante .....	60
Tabla 12: Prueba de caja negra: Alumno Ayudante .....	61
Tabla 13: Descripción del caso de uso: Préstamo Estudiantil .....	63
Tabla 14: Prueba de caja negra: Préstamo Estudiantil.....	63
Tabla 15: Pruebas de rendimiento realizada con Apache JMeter.....	68
Tabla 16: Escala de Satisfacción .....	71
Tabla 17: Descripción del Diagrama de Componentes Dietas .....	81
Tabla 18: Descripción del Diagrama de Componentes Almacén.....	83
Tabla 19: Caso de uso Bonificación de Pasaje .....	84
Tabla 20: Descripción del Caso de uso Préstamo Estudiantil .....	90
Tabla 21: Descripción del Caso de uso Dieta.....	96

**ÍNDICE DE FIGURAS**

Figura 1: Modelo Conceptual .....	29
Figura 2: Caso de uso del sistema .....	35
Figura 3: Diagrama de secuencia Crear Alumno Ayudante .....	43
Figura 4: Diagrama de secuencia Modificar Alumno Ayudante .....	44
Figura 5: Diagrama de secuencias Eliminar Alumno Ayudante .....	44
Figura 6: Modelo de Datos .....	45
Figura 7: Arquitectura MTV .....	46
Figura 8: MVT del sistema .....	47
Figura 9: Diagrama de Componente Gestionar Alumno Ayudante .....	50
Figura 10: Diagrama de Componentes Gestionar Préstamo Estudiantil.....	52
Figura 11: Diagrama de Componentes Gestionar Bonificación de Pasaje .....	53
Figura 12: Diagrama de Despliegue.....	58
Figura 13:Pruebas Funcionales .....	65
Figura 14: Pruebas de Seguridad .....	67
Figura 15: Diagrama de Componentes Dietas .....	80
Figura 16: Diagrama de Componentes Almacén.....	83

## INTRODUCCIÓN

La economía es una ciencia social casi tan antigua como la humanidad, en los principios de la convivencia en sociedad se comenzaron a planificar y administrar los recursos en comunidad, permitiendo así la satisfacción de las necesidades y una mejor calidad de vida. Ya en la antigüedad en las primeras civilizaciones existían evidencias del estudio de esta ciencia. Se les concede a los griegos el mérito de ser los primeros en emplear el término economía que en aquel entonces solo lo utilizaron para referirse a la administración del hogar (Velazquez, 2016).

Estos primeros avances dieron paso al desarrollo de la humanidad y más tarde el nacimiento del mundo empresarial, lo cual impone la necesidad de gestionar la economía para mantener una correcta administración de los recursos de la organización, es decir, gastos e ingresos, pero esta sería imposible sin una buena gestión de la información, la que en sus inicios solo se recopilaba en manuscrito, los que aumentaron considerablemente durante la revolución industrial con el surgimiento de la imprenta y la formación de las universidades (Velazquez, 2016).

En Cuba, con la actualización del modelo económico, se exige un riguroso control de la situación financiera de las entidades, para obtener buenos resultados en la gestión económica, por lo que se debe tener en cuenta la implementación y aplicación de sistemas informáticos contables, capaces de llevar el control de los requerimientos necesarios en aras de aumentar el crecimiento de la economía.

Estas condiciones reclaman eficiencia, efectividad y eficacia en el ejercicio del gasto público, así como el empleo de la creatividad en el uso de las potencialidades humanas creadas y el uso de alternativas para captar recursos complementarios al presupuesto del Estado (Santana, 2021). En este proceso de perfeccionamiento de la economía el Ministerio de Educación Superior (MES) y su red de instituciones lleva a cabo un proceso de mejora continua, que alcanza estados de desarrollo cualitativa y cuantitativamente superiores en cada periodo transcurrido, y está caracterizado por un modelo de gestión económico-financiero diseñado, aprobado, implementado e institucionalizado en la organización (Santana M. A., 2018).

La Universidad de las Ciencias Informáticas como universidad adscripta al MES no está exenta del control de sus procesos económicos. Para gestionar y dar seguimiento al presupuesto del estado como parte de su aparato administrativo cuenta en las facultades con el Vicedecanato de Economía y Administración.

El Vicedecanato de Economía y Administración de la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), como parte de sus funciones lleva a **cabo la gestión económica. En la gestión económica** se realizan disímiles actividades entre la que destacan: gestión de las solicitudes de materiales a los almacenes, de dietas, de bonificación de pasaje, control de los alumnos ayudantes y de los préstamos estudiantiles, entre otros. Para llevar a cabo la ejecución de estas actividades la vicedecana gestiona de forma manual las informaciones. Esta situación provoca que se genere un gran cumulo de información en formato duro o a través de la herramienta Excel, lo cual ralentiza la gestión de los procesos, y da origen a la duplicidad y pérdida de información.

Como resultado de la situación problemática se plantea el **problema de investigación**: ¿Cómo contribuir a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1?

En la presente investigación se define como **objeto de estudio** el proceso de gestión económica, enmarcado en el **campo de acción**: gestión económica en el Vicedecanato de Economía y Administración de la Facultad 1.

Para dar solución al problema de investigación se define como **objetivo general**: Desarrollar un sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1. El objetivo general se desglosa en los siguientes **objetivos específicos**:

- Caracterizar los referentes teóricos relacionados con los procesos de gestión económicas de dietas, solicitud de materiales, bonificación de pasajes, préstamo estudiantil y de los alumnos ayudantes.
- Definir el ambiente de desarrollo para el sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.
- Diseñar el sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.
- Implementar el sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.
- Probar el sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.

La presente investigación será guiada por la **hipótesis**: El desarrollo de un sistema contribuye a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.

**Variable independiente:** sistema.

**Variable dependiente:** contribuye a la gestión.

A continuación, en la tabla 1 y 2 se definen la variable independiente y dependiente correspondiente a la hipótesis.

*Tabla 1: Variable Independiente*

VARIABLE INDEPENDIENTE	INDICADORES	UNIDAD DE MEDIDA
Sistema	Funcional	1: Funcionamiento completo. 2: Funcionamiento parcialmente completo. 3: Funcionamiento incompleto. 4: Funcionamiento fuera de servicio.
	Interfaz Gráfica	1: Errores ortográficos. 2: Botones coherentes. 3: Diseño adaptativo. 4: Uso de colores refrescante.
	Satisfacción del usuario	1: Clara satisfacción. 2 : Más satisfecho que insatisfecho. 3: No definida. 4: Más insatisfecho que satisfecho. 5: Clara insatisfacción. 6: Contradictoria.

*Tabla 2: Variable Dependiente*

VARIABLE DEPENDIENTE	DIMENSIONES	INDICADORES	UNIDAD DE MEDIDA
Contribuye a la gestión	Centralización	Información almacenada de	Diseño de la base de datos

		manera centralizada	
	Disponibilidad	Tiempo de consulta de la información	Bien: 1 a 3 segundos Regular: 4 a 5 segundos Mal: mayor a 5 segundos
	Calidad de la información	Duplicidad de la información	1: No existe duplicidad. 2: Existe duplicidad.

**Posibles resultados:** Sistema para la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1.

Para obtener los conocimientos necesarios, con la finalidad de hacer posible el cumplimiento del objetivo trazado en el trabajo, se utilizaron algunos de los **métodos científicos** existentes, tanto teóricos como empíricos (González, 2008).

**Histórico-Lógico:** se emplea para estudiar la evolución histórica y tendencias actuales de la gestión económica, para la construcción de los referentes teóricos que sustentan la investigación, el estudio de soluciones similares, así como de las metodologías de desarrollo, las tecnologías, herramientas, framework (marcos de trabajo), lenguajes de programación y de modelado utilizados.

**Análisis-Síntesis:** empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema. Así como, para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser utilizada en el desarrollo del mismo, además, en la identificación de los elementos del marco teórico de la investigación.

**Modelación:** para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales que den cumplimiento a los requisitos funcionales de la solución propuesta.

Los **métodos empíricos** representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a determinada elaboración racional; de ellos se emplearon los siguientes:

**Entrevista:** empleado en los encuentros con el cliente para obtener la información necesaria que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución. (Ver anexo 1).

**Observación:** se utilizó para identificar buenas prácticas y vulnerabilidades de los sistemas similares para la gestión económica en el Vicedecanato de Economía y Administración de la Facultad 1.

**Análisis documental:** se utilizó para la revisión bibliográfica y la consulta de las diferentes resoluciones de la Constitución de la República que sustentan los procesos económicos que se llevan a cabo en el Vicedecanato de Economía y Administración de la Facultad 1.

El presente trabajo de diploma está estructurado: introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos. A continuación, la descripción de la estructura de los capítulos:

**Capítulo 1:** Fundamentos y referentes teórico-metodológicos sobre el sistema de gestión económico de la Facultad 1: se realiza un estudio del arte sobre los principales aspectos de la investigación. Se describe el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías utilizadas en la propuesta de solución.

**Capítulo 2:** Características y diseño del sistema de gestión económica de la Facultad 1: se modelan los procesos del negocio, teniendo en cuenta la metodología seleccionada. Se identifican los requisitos funcionales y no funcionales del sistema, así como el diseño arquitectónico de la propuesta.

**Capítulo 3:** Implementación y pruebas del Sistema de Gestión Económica de la Facultad 1: en este capítulo se definen los estándares de codificación que sirven de guía para la implementación de la propuesta de solución, así como la estrategia de pruebas a aplicar para lograr un correcto funcionamiento del Sistema.



## **CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL SISTEMA DE GESTIÓN ECONÓMICO DE LA FACULTAD 1**

En este capítulo se abordan los principales elementos teóricos que sustentan la investigación. Se realiza un estudio de los sistemas homólogos y se describe el proceso de desarrollo de software, así como las técnicas, metodología y tecnologías utilizadas en la propuesta de solución.

### **I.1 Conceptos asociados a la investigación**

#### **Gestión Económica**

Según el sitio web GoCardless: la gestión económico-financiera consiste en administrar y analizar correctamente los recursos de la empresa con el fin de lograr, de manera eficiente, los objetivos establecidos (GoCardless C. , 2022).

Otros autores consideran que la gestión económica consiste en garantizar un aseguramiento material y financiero a las actividades sustantivas con adecuada eficiencia, eficacia y calidad, dentro de un marco de riguroso control y exigencia (Lizmary Ricardo Herrera, 2019).

Por su parte (Alizo, 2010) plantean que la gestión económica implica adaptarse al entorno económico, cuyos componentes son: mano de obra, tasa de interés, tipo de cambio, disponibilidad de recursos de la zona, adquisición tecnológica, financiamiento público, producción, inversión, innovación y mercado, elementos éstos presentes implícitamente dentro del contexto y proceso de toma de decisiones

La autora del presente trabajo considera que la gestión económica puede definirse como el proceso de administrar las actividades económicas dentro de un sistema dado, con eficiencia y eficacia, involucrando un equipo determinado de trabajadores, de allí la sintonía que tiene esta expresión con la dirección, ya que denota guiar y encaminar hacia un determinado objetivo.

#### **Dieta**

En la Gaceta Oficial de la República de Cuba del Ministerio de Justicia, en la Resolución No. 338 de 2020 se plantea: los jefes de las entidades garantizan las condiciones de alimentación y hospedaje de sus trabajadores cuando son enviados a labores fuera de su lugar habitual, siempre que sea posible, o en su lugar, se otorga la dieta para esos conceptos.

Los gastos por dietas son aquellos en los que incurren los cuadros, directivos, funcionarios y trabajadores, en lo adelante trabajadores, cuando son enviados eventual o temporalmente a realizar labores que implican una variación de su lugar habitual de alimentación y alojamiento, en el territorio nacional (Ministerio de Justicia, 2020).

Estos gastos comprenden los de alojamiento y de alimentación; este último en concepto de desayuno, almuerzo y comida.

SEXTO: Las dietas se entregan por anticipado a los receptores o se pueden pagar por las entidades directamente a las empresas o instalaciones que prestan los servicios de alojamiento y gastronomía, utilizando los instrumentos de pago vigentes; y se calculan tomando como base los días de duración del trabajo a realizar y el importe de la dieta diaria establecida por la presente.

NOVENO: Los receptores de los anticipos para dietas, liquidan dentro de los tres días hábiles siguientes a la fecha de regreso a la localidad donde radique su centro de trabajo y en ningún caso se entrega nuevo anticipo al receptor, sin que haya liquidado el anterior.

DÉCIMO: Cuando excepcionalmente, por razones justificadas y mediante la debida autorización, al trabajador le correspondiera la dieta y no se le asignara anticipadamente, se le paga la que corresponde a su regreso a la entidad, según los días utilizados (Ministerio de Justicia, 2020).

### **Bonificación de pasajes**

En la Gaceta Oficial de la República de Cuba del Ministerio de Justicia en la Instrucción No. 2/07 hace mención a la bonificación de pasajes para el uso del transporte urbano e interprovincial por estudiantes y cadetes de la Educación Superior. En el segundo artículo plantea que: recibirán el beneficio de la bonificación los estudiantes becados que estudian en Centros de Educación Superior fuera de su municipio de residencia, cuando hagan uso del transporte terrestre suburbano, a medias y largas distancias en servicios corrientes, regular y especial, por ómnibus o ferrocarril respectivamente, y en la transportación marítima desde y hacia la Isla de la Juventud (Ministerio de Justicia, 2007).

QUINTO: Los estudiantes recibirán la bonificación del 50% del pasaje solo una vez al mes y para los doce meses del año teniendo en cuenta los procedimientos establecidos en las Normas Cubanas de Información Financiera:

- El estudiante pagará en la taquilla de compra de boletines el 100% del importe del pasaje y

solicitará después de realizado el viaje el reintegro de la bonificación del 50% del mismo previa presentación del boletín o comprobante.

- El estudiante solicitará excepcionalmente a la Dirección de su Facultad un Anticipo del 50% del importe del pasaje que se bonifica, debiendo liquidar el mismo por los mecanismos establecidos con la aprobación del Decano, se presentará a la Dirección de Economía recibir el anticipo correspondiente.
- La entrega del efectivo por el concepto de bonificación será en las cajas de pago habilitadas en cada Centro de Educación Superior.

### **Alumnos Ayudantes**

En la Resolución No. 47/2022 en el Capítulo 8 Trabajo con Alumnos Ayudantes y estudiantes de alto aprovechamiento docente en el Artículo 386.1 plantea que: los Alumnos Ayudantes son aquellos estudiantes previamente seleccionados en las carreras que se distinguen por poseer una destacada trayectoria integral, y por mostrar aptitudes favorables para el aprendizaje y para el desempeño de la docencia (Ministerio de Educación Superior, 2021).

Estos estudiantes realizan tareas complementarias a su plan de estudio con el propósito de formarlos como profesores en las instituciones de educación superior para contribuir al trabajo docente y metodológico en las diferentes disciplinas y asignaturas.

En su desempeño como alumnos ayudantes pueden realizar tareas de investigación, en ramas afines a la carrera o relacionadas con el trabajo científico metodológico.

Artículo 387: Los alumnos ayudantes son seleccionados desde que finalizan, el primer año. Los requisitos mínimos que deben cumplir son los siguientes:

a) Buena evaluación de la conducta política y social, avalada por el colectivo de año académico y las organizaciones estudiantiles.

b) Poseer un índice académico igual o mayor que 4 puntos y haber obtenido como mínimo la calificación de Bien (4) en la asignatura a impartir.

C) Capacidad y disposición para trabajar en las tareas que se le asignen.

Excepcionalmente puede ser aprobado por el rector, si el estudiante no posee un índice académico igual o mayor que 4 puntos.

Si el estudiante pierde alguno de los requisitos anteriores no puede mantenerse como alumno ayudante.

Artículo 338.1 Los alumnos ayudantes que impartan asignaturas como profesores deben tener un período previo de preparación antes de participar directamente en la docencia, que incluye la realización de una clase de comprobación con evaluación satisfactoria.

Esta preparación puede estar dirigida a lograr su formación en aspectos de carácter metodológicos y en el contenido de la asignatura que va a impartir, de acuerdo con las necesidades institucionales y con las aptitudes y motivaciones que posea el estudiante. Pueden participar en tareas de carácter docente y educativo en las sedes centrales, en los centros universitarios municipales y filiales, fungiendo como profesores, profesores guías y tutores, entre otros.

### **Préstamo estudiantil**

En la RESOLUCIÓN No.108 /20 Capítulo 1 Disposiciones Generales en el Artículo 2 hace referencia al concepto de préstamo estudiantil que es aquel que recibe un estudiante, a solicitud propia, para cubrir necesidades que no puede alcanzar por proceder de una familia de bajos ingresos, así como a los que no tienen amparo familiar, con carácter reembolsable (Ministerio de Justicia de Cuba, 2020).

En dicha resolución antes mencionada en el Capítulo IV De los préstamos estudiantiles reembolsables en los siguientes artículos se hace alusión al funcionamiento del préstamo estudiantil donde este es fundamentado ante la Comisión de Estipendio presidida por el Decano.

ARTÍCULO 25. Los estudiantes de la Educación Superior del curso diurno que así lo requieran, puede otorgársele por un tiempo determinado (trimestre, semestre, curso) préstamos por pagos mensuales convenidos una vez concluidos cada curso.

ARTÍCULO 26. Los préstamos se fundamentarán ante la Comisión de la Facultad y el mismo no podrá exceder los trescientos veinte (320.00) pesos mensuales. Cuando con carácter excepcional se apruebe una cuantía superior a la establecida, esta no podrá exceder de los cuatrocientos (400.00) pesos mensuales, lo que será aprobado por el jefe máximo de la IES.

ARTÍCULO 27. Los estudiantes que causen baja del sistema de la educación superior por cualquiera

de las causales establecidas deberán reintegrar el adeudo acumulado por el préstamo recibido, para lo cual se podrán establecer convenios de pagos aplazados, o el reintegro total.

ARTÍCULO 28. Los estudiantes que al concluir sus estudios obtengan Título de Oro o sean Premio al Mérito Científico, el monto de la deuda le será condonado, y a los que obtengan una calificación en el rango de 4.50 y 4.74 puntos como promedio durante la carrera, serán beneficiados con un cincuenta por ciento (50%) de condonación de dicho adeudo.

### **Solicitud de materiales**

En la Resolución No. 11/2007 del Ministerio de Finanzas y Precios se establece el modelo SC-2-07: Solicitud de entrega, que tiene como objetivo: Formalizar el pedido de productos con destino al consumo de la unidad solicitante. Este modelo es de uso obligatorio para todas las entidades que se encuentran dentro del territorio nacional. Cada entidad definirá las áreas emisoras del documento, así como la distribución de copias, si procede, acorde a sus características organizativas y define como datos obligatorios (Ministerio de Finanzas y Precios, 2007):

1. Fecha de emisión del modelo.
2. Nombre y código de la entidad.
3. Nombre y código del almacén al que se efectúa la Solicitud.
4. Nombre y código del área, del centro de costo y número de la Orden de Producción o de Trabajo a la que se cargarán los productos.
5. Código, descripción, unidad de medida y cantidad de cada producto solicitado.
6. Nombre y firma del solicitante y fecha de la Solicitud.
7. Nombre y firma del funcionario que autoriza la Solicitud y fecha de la misma.
8. Nombre y firma del empleado que recibe la Solicitud en el almacén y fecha de la misma.
9. Número consecutivo del modelo.

### **I.2 Análisis de las soluciones existentes**

En esta sección se realiza un análisis de soluciones similares en el ámbito internacional y nacional, con el fin de obtener las características principales, que apoyen la propuesta de solución.

### 1.2.1 Ámbito internacional

#### **Anfix**

Es una herramienta integral de gestión empresarial en la nube. Ofrece diferentes tipos de planes: el “Profesional” cubre las necesidades básicas de cualquier autónomo con respecto a la facturación, el “Empresarial” se hace cargo de la gestión contable y tiene una opción orientada a asesorías.

Este software tiene una tecnología, denominada cleverscan, que permite escanear y contabilizar tickets y facturas. También cuenta con un sistema de conciliación bancaria automática e incorporan las notificaciones que envía la agencia tributaria directamente al programa. Se presenta en forma web y en aplicaciones de Android y iOS, lo que resulta ventajoso, pero es de licencia privativa y el pago se realiza anualmente (Cordero, 2019).

#### **Factorial**

Factorial es un Software de recursos humanos desarrollado para digitalizar empresas mediante la integración de procesos de gestión como el control de horarios, fichaje, control de gastos y comunicación interna, es de licencia privativa. Está disponible para ordenadores y dispositivos móviles en iOS y Android, a fin de brindar una experiencia de uso óptima, intuitiva y segura. Factorial está pensado tanto para un departamento de Recursos Humanos de una sola persona, como para equipos de 20, 50 o más de 100 empleados y actualmente es empleado por muchas compañías en diferentes partes del mundo (Romero, 2022).

### 1.2.2 Ámbito nacional

#### **KRONOX**

Este software es de creación extranjera, pero es utilizado en Cuba y cuenta con un conjunto de soluciones automatizadas, inteligentes e innovadoras centradas en la administración de capital humano o mano de obra que ayudan a las organizaciones a manejar el aspecto contable de la planificación y gestión de recursos empresariales. Los usuarios a nivel de administrador del sistema pueden adaptar los flujos de trabajo y las plantillas básicas de Kronox para que se ajusten a la forma en que su empresa administra su negocio. El sistema permite la creación personalizada de formularios de evaluación de empleados y puede almacenar dicha documentación de forma segura en línea, incorporar calendarios, documentos y hojas de cálculo. Permite convertir los datos de nómina en informes en tiempo real. Es una herramienta muy competente y completa, pero se necesita capacitación para utilizarlo y es de software privativo (Ain, 2022).

## RODAS XXI

Sistema multiempresa y multiusuario desarrollado en Cuba por CITMATEL para la automatización de la gestión empresarial. Rodas XXI, actualmente cuenta con siete módulos: finanzas, contabilidad, activos fijos, nóminas, inventario, facturación y recursos humanos. Estos módulos pueden ser empleados e integrados en su totalidad, y se puede formar cualquier subconjunto entre ellos, o cada uno de forma independiente. Esta herramienta ha sido diseñada para trabajar en el sistema operativo de Microsoft: Windows XP, por lo que no es multiplataforma ni de código abierto (Rodríguez, 2021).

A continuación, en la Tabla 3 se presenta una tabla resumen donde se recogen los datos más significativos de cada uno de los sistemas analizados:

*Tabla 3: Comparación de sistemas homólogos*

<b>Sistema</b>	<b>Entorno</b>	<b>Código Abierto</b>	<b>Multiplataforma</b>
Anfix	Web	NO	SÍ
Factorial	Web	NO	SÍ
KRONOX	Web	NO	NO
RODAS XXI	Web	NO	NO

El estudio realizado anteriormente sirvió de base para la identificación de características básicas para la implementación de la propuesta de solución, pero aun cuando son herramientas versátiles para la gestión empresarial, ninguna gestiona específicamente los procesos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1, por lo que no responden a las necesidades específicas del cliente, demostrando así la necesidad de la creación de sistema para la gestión económica en Vicedecanato de Economía y Administración de la Facultad 1.

El análisis de los sistemas homólogos permitió identificar las siguientes características básicas para la implementación de la propuesta de solución:

- Desarrollo del Sistema de Gestión Económica de la Facultad 1 (SGEF1) sobre un entorno web, ya que todos los sistemas estudiados fueron desarrollados de esta manera, haciéndolos más portables y aportando la ventaja de no requerir de instalación, solo de un navegador web presente en cualquier dispositivo.
- Desarrollar para la propuesta de solución un sistema multiplataforma, aunque solo Anfix y Factorial cuentan con esta característica, dado que los sistemas multiplataforma son compatibles con todo tipo de dispositivos y cualquier sistema operativo.
- Además, para darle cumplimiento a el paradigma de independencia tecnológica por el que hoy aboga el país, se determinó como característica primordial utilizar código abierto para su desarrollo a pesar de que ninguno de los sistemas anteriores lo tiene en cuenta.

### **I.3 Análisis de las metodologías, herramientas y tecnologías a utilizar para el desarrollo del sistema.**

A continuación, se especifican la metodología, herramientas y tecnologías que se utilizan en el desarrollo del sistema que contribuye a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1

#### **1.3.1 Metodología para el desarrollo de software**

La metodología no es más que un conjunto de elementos de tipo racional que se emplean para alcanzar objetivos referentes a una investigación, por ello, al término se le conoce como la metodología de investigación o, en su defecto, como la metodología de un proyecto. El término tiene su génesis en el griego meta, el cual significa ir más allá, camino y logos, lo cual significa estudio, razón o análisis. Científicamente la metodología es un procedimiento general para lograr de manera precisa el objetivo de la investigación, por lo cual se presentan los métodos y técnicas para la realización de la investigación (Tamayo y Tamayo, 2007).

Las metodologías de desarrollo de software se clasifican en tradicionales y ágiles. Según (Dingsøyr et al. 2012) las metodologías tradicionales se caracterizan por tener una estructura de desarrollo claramente establecida, lineal y poco flexible ante cambios de un entorno volátil. Presentan un alto costo al



ser implementadas. Tienen roles, actividades y artefactos claramente definidos. Requieren de una documentación exhaustiva, clara y detallada de todo el proyecto. Y centran su atención en cumplir un plan establecido desde el inicio del desarrollo, por todas las partes involucradas.

Por su parte (Serrador y Pinto, 2015) plantean que las metodologías ágiles son una nueva serie de metodologías que pretenden superar las limitaciones de los SDM tradicionales basados en planes. Su uso en el desarrollo de software se ha extendido en los últimos años, toda vez que permite contrarrestar los peligros de los métodos tradicionales de planificación inicial que a menudo conducen a patologías de desarrollo posteriores (Serrador y Pinto, 2015).

Para guiar el desarrollo de la propuesta de solución se selecciona la variante de *Proceso Unificado Ágil (AUP, por sus siglas en inglés)* para la producción de software en la UCI, por ser la metodología que se emplea en los proyectos productivos de la universidad. La metodología cuenta con tres etapas como son una etapa de Inicio en la cual se llevan a cabo las tareas relacionadas con la planeación del proyecto. En la Ejecución se tienen en cuenta las actividades requeridas para el desarrollo, los requisitos y la arquitectura. En el Cierre se analizan los resultados del proyecto. AUP UCI define 7 disciplinas elementales: Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación (Sánchez, 2015).

Esta metodología está certificada en el nivel 2 de *Capability Maturity Model Integration (CMMI, por sus siglas en inglés)*. Se seleccionó específicamente el escenario número dos (2) ya que aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información (Sánchez, 2015).

### **1.3.2 Herramientas y tecnologías**

#### **Lenguaje de Modelado Unificado 2.5**

El Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) es un lenguaje estándar para escribir diseños de software, permite visualizar, especificar, construir y documentar los artefactos de un sistema de software. Al igual que los arquitectos de edificios crean planos, los arquitectos de software usando como herramienta las prestaciones de UML, pueden crear diagramas para ayudar a los desarrolladores de software a su construcción. Además de permitir socializar y especificar el diseño de un sistema (Pressman, 2005).

#### **Visual Paradigm 8.0**

Es una suite de herramientas de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) para el desarrollo de aplicaciones usando UML, perfecto para la construcción de sistemas a gran escala. Aporta al proceso de desarrollo calidad, confiabilidad y estabilidad, así como soporte para la Programación Orientada a Objetos. Es perfecta para la construcción de artefactos, necesarios durante todo el proceso ingenieril del desarrollo de software (Aristigüi, 2017).

### **Microsoft Visual Studio Code 1.7.3**

Es un editor de código ligero y adaptable desarrollado por Microsoft, puede correr sobre Windows, macOS y Linux. Tiene soporte para JavaScript, TypeScript and Node.js, pero usando su rico ecosistema de extensiones se puede personalizar a niveles importantes, permitiendo construir un entorno de desarrollo ideal para cualquier programador, en casi cualquier lenguaje de programación, agregándole capacidades que rivalizan con cualquier Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) especializado (Microsoft, 2022).

### **JavaScript 8.0**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2019).

Este lenguaje fue creado con bajas expectativas, sin embargo, en poco tiempo superó a Java como el lenguaje principal para crear páginas web dinámicas. Aunque los primeros 20 años de JavaScript se caracterizaron por fallidos intentos de mejorarlo, rediseñarlo o reemplazarlo, al final de ese período se convirtió en el lenguaje de programación más difundido en el mundo no solo para páginas web, adicionándose la posibilidad de construir aplicaciones web usando Node.js, así como aplicaciones de escritorio y para teléfonos móviles, con el uso de React Native, esto sin dejar de hablar de su uso en robots y sistemas embebidos (WIRFS-BROCK, 2020).

### **Lenguaje de Marcación de Hipertexto 5.0**

HTML 5 es un **Lenguaje de Mercado de Hipertexto** (HyperText Markup Language) que funciona con las bases de HTML 4, pero que cuenta con etiquetas adicionales que permiten crear sitios webs más modernos y compatibles con las tecnologías digitales actuales. Cuenta con algo llamado “etiquetas

semánticas”, que permiten que el sitio sea indexado con mayor exactitud por los buscadores y definir mejor la estructura y componentes de una página web, ya que no sólo brindan instrucciones sobre la presentación de un elemento de texto, además brindan información sobre su contenido o relevancia. Por ejemplo, HTML5 permite catalogar un contenido como artículo, una sección, un encabezado, un cavas de diseño, entre otros.

Algunas de las ventajas de HTML5 son (Workana LLC , 2021):

- Mejora la experiencia del usuario en los navegadores web más comunes (Google, Safari, Firefox, entre otros) y entre distintos dispositivos (diseño responsivo).
- Para obtener un mejor posicionamiento en buscadores (de hecho, Google recomienda su uso).
- HTML5 puede ser interpretado por numerosos sistemas informáticos.
- Soporta funcionalidades multimedia e interactivas, como vídeos o formularios.
- Permite utilizar API´s de Javascript para incluir elementos complejos, como dibujos 3D o mapas directamente en el documento.
- Posibilidad de realizar consultas SQL.
- Sirve para crear aplicaciones web.
- Permite optimizar el tiempo de carga de un sitio.
- Protege mejor la información del sitio.

### **Lenguaje Cascading Style Sheets 3.0**

Hoja de Estilo en Cascada (CSS, por sus siglas en inglés) es un lenguaje de estilos estandarizado por W3C y utilizado para definir la presentación, el formato y la apariencia de un documento de marcado como por ejemplo HTML. Nacieron con la necesidad de diseñar la presentación de la información de tal manera que resulte más comprensible, además de facilitar dicho proceso y poderlo hacer para ofrecer distintas presentaciones de la información en dispositivos diferentes con características particulares a cada uno. En su última versión 3, ofrecen nuevas facilidades y herramientas que permiten al diseñador o programador presentar los contenidos de forma más vistosa, en menos tiempo y con menos esfuerzo, por lo que resulta una herramienta muy atractiva (PUIG, 2013).

### **Bootstrap 5.0**

Bootstrap es un framework desarrollado por Twitter, dirigido al front-end de una aplicación web, por lo que está orientado a la parte que se ejecuta en el cliente, en este caso el navegador web. Utiliza HTML5, CSS y JavaScript, los lenguajes por excelencia para contenido web, para hacer uso de sus librerías que permiten diseñar la web de una manera profesional creando al mismo tiempo un diseño adaptado para dispositivos móviles, con el uso de la filosofía "Mobile first" del inglés móvil primero, la cual consiste

en enfocarse primero en la vista de teléfonos móviles y luego ir escalando hasta llegar a las vistas de mayor resolución de pantalla. Es código abierto y uno de los proyectos más populares de todos los tiempos en GitHub (GÓMEZ, 2018) y (NISKA, 2014).

### **JQuery 3.6.0**

JQuery es una librería de JavaScript software libre y de código abierto, que permite simplificar el trabajo con JavaScript. Es rápida, sencilla, eficaz y poderosa, se usa para darle dinamismo y funcionalidades al frontend de las páginas web, mediante la manipulación del DOM y el manejo de eventos o acciones del usuario, facilita la aplicación de animaciones y la utilización de AJAX para simular el comportamiento de una aplicación de escritorio. El hecho de simplificar las funcionalidades del lenguaje garantiza lograr mejores resultados con menor tiempo y esfuerzo (VENTURA, 2021).

### **Python 10.5**

Es un lenguaje de programación de alto nivel y multiplataforma inventado por Guido Van Rossum en 1989. Python es uno de los lenguajes más usado en el mundo del software libre por lo que funciona sin costo es Windows, Unix, Linux, y otros sistemas operativos con una sintaxis más simple y elegante que la de otros lenguajes de programación. Es interpretado, de una sintaxis sencilla, propiedades que lo hacen muy adecuado para el proceso de enseñanza-aprendizaje de lenguaje de programación y su flexibilidad a la hora de proponer soluciones de programación ya que es un lenguaje de múltiples propósitos. Python también soporta características de computación moderna tales como grandes datos (Big Data en inglés), aprendizaje de máquina, y el desarrollo de aplicaciones web con poca programación como con el marco de trabajo Django (Python, 2022).

### **Django 4.0.1**

Es un es un marco de trabajo (framework) para el desarrollo de aplicaciones web usando Python. Considera algunas funcionalidades para facilitar el desarrollo de aplicaciones web. Como resultado, no es necesario escribir todo el código ni usar tiempo para buscar errores de código en el framework. Es decir, mediante Django, el desarrollo de sistemas de información web puede ser rápido, seguro, escalable y también fáciles de mantener. Django representa un marco de trabajo para el desarrollo rápido de sistemas de información web con Python. También implementa el concepto de Model-Template-View (MTV, por sus siglas en inglés) que es ligeramente diferente de Model-View-Controller (MVC, por sus siglas en inglés) (Django, 2022).

### **Sistema Gestor de Base de Datos:**

SQLite 3.40.0 implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. SQLite es el motor de base de datos más utilizado en el mundo. SQLite está integrado en todos los teléfonos móviles y en la mayoría de las computadoras y viene incluido dentro de innumerables otras aplicaciones que las personas utilizan todos los días. El formato de archivo SQLite es estable, multiplataforma y compatible con versiones anteriores y los desarrolladores se comprometen a mantenerlo así hasta el año 2050. Los archivos de base de datos SQLite se usan comúnmente como contenedores para transferir contenido enriquecido entre sistemas y como un formato de archivo a largo plazo para datos. El código fuente de SQLite es de dominio público y es gratuito para todos para cualquier propósito (SQLite, 2022).

### **Apache 2.4.46**

Es un servidor web que implementa el protocolo HTTP, es software libre y se puede instalar en los sistemas operativos Windows y Linux. Es de fácil configuración, pues su estructuración en módulos permite al usuario utilizar los servicios y funcionalidades que ofrece. Se define Apache como software para el servidor de aplicaciones en su versión 2.4.46, por su flexibilidad, rapidez, porque es gratuito, multiplataforma e interpreta varios lenguajes como Python y HTML.

Estas características promueven la eficiencia y rendimiento del sistema que debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño, teniendo en cuenta la concurrencia que pueda existir; debe prestar servicios sin que se amplíen los rangos de tiempo de repuesta (Apache, 2018).

### **Conclusiones del capítulo**

- ✓ El análisis de los principales conceptos asociados a la investigación, permitió fundamentar los procesos de gestión económica del Vicedecanato de Economía y Administración de la Facultad 1. Con el análisis de los sistemas homólogos se concluyó que, ninguno satisface la necesidad de la problemática, por lo que se propone implementar un nuevo sistema.
- ✓ Se definió como bases tecnológicas fundamentales: Python 3.10.5 como lenguaje de programación, Django 4.01 como Framework Python, SQLite 3.40.0 como gestor de base de datos y AUP-UCI en el Escenario No.2 como metodología de desarrollo, sentando las bases tecnológicas para el desarrollo de la solución.

## **CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: SISTEMA DE GESTIÓN ECONÓMICA DE LA FACULTAD 1**

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como el estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del sistema. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

### **II.1 Descripción de la propuesta de solución**

Para dar solución a la problemática planteada en la presente investigación, se propone implementar un sistema que contribuya a la gestión del control económico en el Vicedecanato de Economía y Administración de la Facultad 1. Este sistema debe gestionar los procesos de: Dietas, Bonificación de Pasaje, Préstamo Estudiantil, Gestión de Almacén y el control de los Alumnos Ayudantes.

#### **II.1.1 Modelo Conceptual**

Un modelo conceptual es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, presentado como uno o más diagramas de clases. Se pueden utilizar para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Los objetos del dominio o clases pueden obtenerse a partir de una especificación de requisitos (GARCERANT, 2008). A continuación, en la figura 1 se representa el modelo conceptual.

#### **Descripción de las clases del modelo:**

1. **Facultad 1:** Facultad de la Universidad de las Ciencias Informáticas.
2. **Decanato:** Oficina de la decana de la facultad.
3. **Vicedecanato de Economía y Administración:** Oficina de la Vicedecana de Economía y Administración de la facultad.
4. **Decana:** Dirige la facultad.

5. **Vicedecana de Economía y Administración:** Dirige la parte Económica y Administrativa de la facultad.
6. **Procesos:** Procesos que realiza la vicedecana.
7. **Gestionar Dietas:** Proceso de gestionar las dieta.
8. **Gestionar Préstamo Estudiantil:** Proceso el cual un estudiante solicita un préstamo.
9. **Gestionar Bonificación de Pasaje:** Proceso mediante el cual se le bonifica un pasaje al estudiante.
10. **Gestión de Almacén:** Proceso mediante el cual se controla la solicitud y el destino final de varios productos.
11. **Gestión Alumno Ayudante:** Proceso mediante el cual se realiza un informe de los alumnos ayudantes.

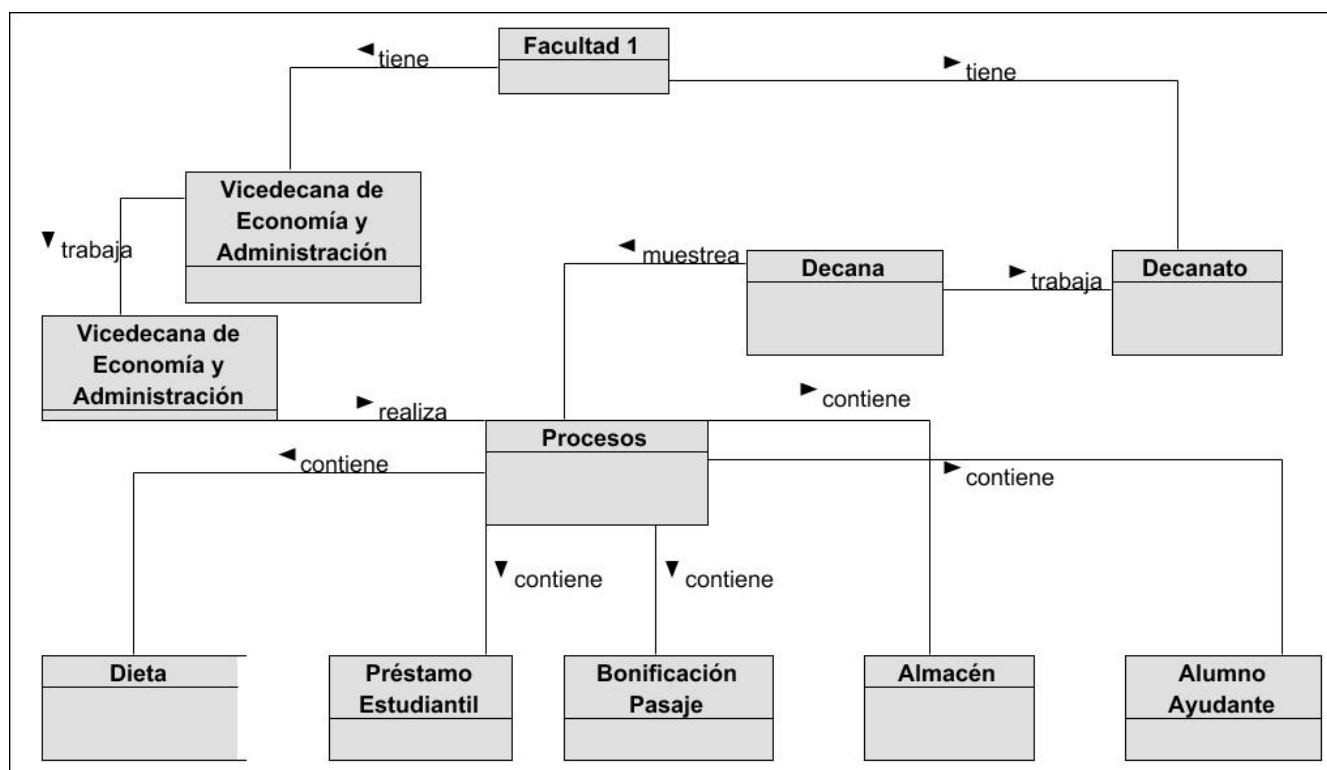


Figura 1: Modelo Conceptual

## II.2 Requisitos, Análisis y Diseño de la solución propuesta

Con la finalidad de que el módulo propuesto se desarrolle con las necesidades del cliente, a través del uso de la técnica de la entrevista, se llegaron a obtener los requisitos funcionales y no funcionales.

### II.2.1 Requisitos Funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

A continuación, en la tabla 4 se muestran los requisitos funcionales.

Tabla 4: Requisitos Funcionales

No.	Requisitos Funcionales	Descripción	Complejidad	Prioridad
RF1	Autenticar usuario	El usuario inicia sección en el sistema	Alta	Alta
RF2	Registrar usuario	El usuario es registrado en el sistema	Alta	Alta
RF3	Eliminar usuario	El usuario es eliminado del sistema	Baja	Media
RF4	Modificar usuario	El usuario es modificado	Media	Alta
RF5	Registrar Dieta	Permite al usuario registrar una dieta	Alta	Alta
RF6	Modificar Dieta	Permite al usuario modificar una dieta	Media	Alta
RF7	Eliminar Dieta	Permite al usuario eliminar una dieta	Baja	Media



RF8	Mostrar Dieta	Permite al usuario mostrar una dieta	Media	Media
RF9	Exportar a formato pdf una dieta	Permite al usuario exportar a formato pdf una dieta	Alta	Alta
RF10	Listar Dietas	Permite al usuario listar todas las dieta	Media	Media
RF11	Buscar Dieta	Permite al usuario buscar una dieta en específica	Baja	Media
RF12	Registrar Alumno Ayudante	Permite al usuario registrar un Alumno Ayudante	Alta	Alta
RF13	Eliminar Alumno Ayudante	Permite al usuario eliminar un Alumno Ayudante	Baja	Media
RF14	Modificar Alumno Ayudante	Permite al usuario modificar un Alumno Ayudante	Media	Alta
RF15	Listar los Alumnos Ayudantes	Permite al usuario listar los Alumnos Ayudantes	Media	Media
RF16	Buscar Alumno Ayudante	Permite al usuario buscar un Alumno Ayudante	Baja	Media
RF17	Registrar Bonificación de pasaje	Permite al usuario registrar una Bonificación de Pasaje	Alta	Alta
RF18	Mostrar Bonificación de pasaje	Permite al usuario mostrar una Bonificación de Pasaje	Media	Media
RF19	Exportar a formato pdf una Bonificación de pasaje	Permite al usuario exportar a formato pdf una Bonificación de Pasaje	Alta	Alta

RF20	Eliminar Bonificación de pasaje	Permite al usuario eliminar una Bonificación de Pasaje	Baja	Media
RF21	Listar Bonificación de pasaje	Permite al usuario listar una Bonificación de Pasaje	Media	Media
RF22	Modificar Bonificación de pasaje	Permite al usuario modificar una Bonificación de Pasaje	Media	Alta
RF23	Buscar Bonificación de Pasaje	Permite al usuario buscar una Bonificación de Pasaje en específica	Baja	Media
RF24	Registrar Préstamo Estudiantil	Permite al usuario registrar un Préstamo Estudiantil	Alta	Alta
RF25	Eliminar Préstamo Estudiantil	Permite al usuario eliminar un Préstamo Estudiantil	Baja	Media
RF26	Modificar Préstamo Estudiantil	Permite al usuario modificar un Préstamo Estudiantil	Media	Alta
RF27	Listar Préstamo Estudiantil	Permite al usuario listar los Préstamos Estudiantiles	Media	Media
RF28	Buscar Préstamo Estudiantil	Permite al usuario buscar un Préstamo Estudiantil en específico	Baja	Media
RF29	Registrar Solicitud de Almacén	Permite al usuario registrar una Solicitud de Almacén	Alta	Alta
RF30	Eliminar Solicitud de Almacén	Permite al usuario eliminar una Solicitud de Almacén	Baja	Media

RF31	Mostrar Solicitud de Almacén	Permite al usuario mostrar Solicitudes de Almacén	Media	Media
RF32	Exportar a pdf Solicitud de Almacén	Permite al usuario exportar a pdf una Solicitud de Almacén	Alta	Alta
RF33	Modificar Solicitud de Almacén	Permite al usuario modificar una Solicitud de Almacén	Media	Alta
RF34	Buscar Solicitud de Almacén	Permite al usuario buscar una Solicitud de Almacén en específico	Baja	Media
RF35	Listar Solicitud de Almacén	Permite al usuario listar las Solicitudes de Almacén	Media	Media
RF36	Registrar Destino Final	Permite al usuario registrar un Destino Final	Alta	Alta
RF37	Eliminar Destino Final	Permite al usuario eliminar un Destino Final	Baja	Media
RF38	Modificar Destino Final	Permite al usuario modificar un Destino Final	Media	Alta
RF39	Mostrar Destino Final	Permite al usuario mostrar Destinos Finales	Media	Media
RF40	Exportar a pdf Destino Final	Permite al usuario exportar a pdf un Destino Final	Alta	Alta
RF41	Listar Destino Final	Permite al usuario listar los Destinos Finales	Media	Media
RF42	Buscar Destino Final	Permite al usuario buscar un Destino Final en específico	Baja	Media

## II.2.2 Requisitos No Funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2005).

- **Usabilidad:**

RnF1. Tipo de aplicación informática: la herramienta debe ser web, y debe permitir adaptarse a dispositivos móviles.

RnF2. El sistema debe poseer una interfaz fácil de utilizar.

- **Software**

RnF3. Sistema Operativo: Podrá utilizar cualquier sistema.

RnF4. La propuesta de desarrollo debe ser capaz de ejecutarse en cualquier navegador.

- **Hardware:**

RnF5. La computadora que se utilice debe contar como mínimo con 2 GB de RAM y de almacenamiento 80 GB.

- **Seguridad:**

RnF8. El acceso a la información debe estar restringido por usuario y contraseña.

RnF9. Se podrá acceder a las funcionalidades del sistema solamente después de autenticarse.

- **Restricciones de diseño e implementación.**

RnF10. El marco de trabajo que se utilizará está basado en el uso de herramientas no privativas:

- Lenguaje de programación del lado del servidor Python v3.10.5.

- Lenguaje de programación del lado del cliente JavaScript 8.0.
- Lenguaje para el estilo del documento web CSS3.
- Lenguaje para la estructuración del documento web HTML5.
- Lenguaje para el modelado del proceso UML v2.5.1.
- Framework de desarrollo web Django v4.0.1.
- Herramienta para el modelado del proceso Visual Paradigm 8.0.
- Herramienta para la codificación Microsoft Visual Studio Code v1.73.1
- Bootstrap 5.1.3 como biblioteca para los estilos.
- SQLite 3.40.0 como gestor de bases de datos.

### II.3 Modelo de caso de uso del sistema

Un caso de uso de sistema es una secuencia de acciones que un sistema lleva a cabo que da lugar a un resultado de valor observable para un actor particular (alguien o algo fuera del sistema que interactúa con el sistema) (IBM, 2022). A continuación, en la figura 2 se representa el diagrama de casos de uso del sistema.

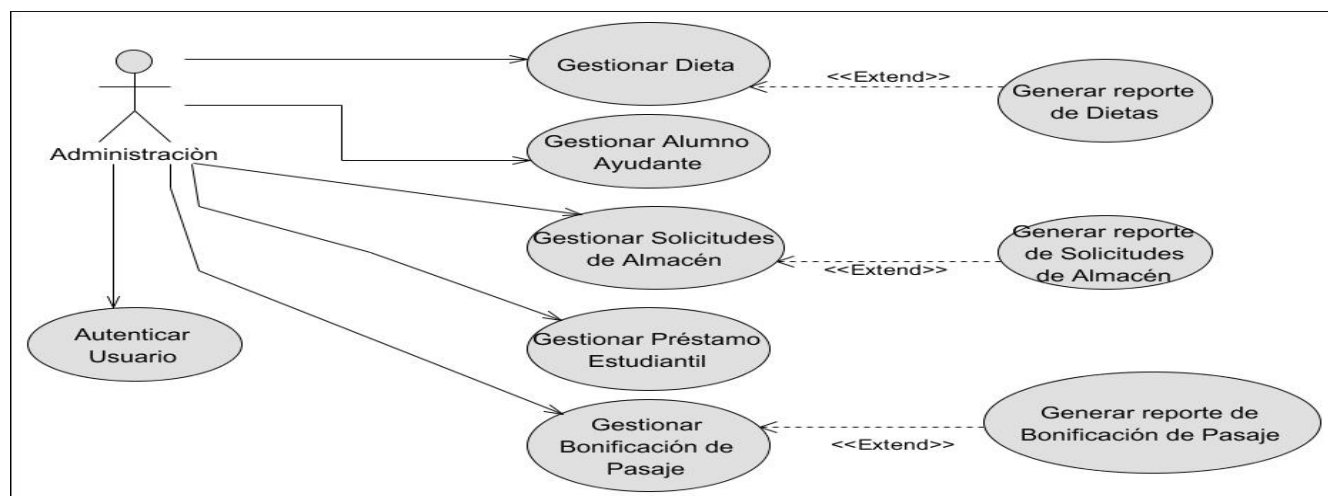


Figura 2: Caso de uso del sistema

### II.3.1 Descripción de los Casos de Uso del Sistema

A continuación, en las tablas 5 y 6 se muestran las descripciones de los casos de uso autenticar usuario y gestionar alumno ayudante.

Tabla 5: Caso de uso autenticar usuario

<b>Objetivo</b>	Permitir a los usuarios de sistema tener acceso a los recursos del sistema.	
<b>Actores</b>	Administración	
<b>Resumen</b>	El caso de uso inicia permitiendo a los usuarios acceder al sistema a través de un usuario y una contraseña.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario previamente registrado.	
<b>Postcondiciones</b>	El usuario se encuentra autenticado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Autenticar usuario &gt;</b>		
<b>Actor</b>	<b>Sistema</b>	
1.	El usuario debe acceder al sistema.	
2.		El sistema muestra una pantalla con los siguientes datos a introducir: - Usuario. - Contraseña. - El botón "Acceder". - El enlace "Crear una nueva cuenta".
3.	El usuario llena los campos para acceder al sistema.	
4.		Valida que los campos estén correctos.
5.		Verifica existencia del usuario.
6.		Permite acceder a las funcionalidades del sistema. Finalizando así el Caso de Uso.

Prototipo elemental de interfaz gráfica de usuario			
Flujos alternos			
4a. Campos vacíos.			
Actor		Sistema	
1.		Muestra mensaje como: “El campo usuario es obligatorio.” o “El campo contraseña es obligatorio.”	
5a. Existencia de usuario			
Actor		Sistema	
1.		Muestra un mensaje indicado: “Estas credenciales no coinciden con los registros”.	
Relaciones	CU Incluidos	No aplica	
	CU Extendidos	No aplica	
Requisitos funcionales	no	No aplica	
Asuntos pendientes		No aplica	

Tabla 6: Caso de uso Gestionar Alumno Ayudante

<b>Objetivo</b>	Permite insertar, modificar, eliminar y listar los datos de los Alumnos Ayudantes.
<b>Actores</b>	Administración.
<b>Resumen</b>	El caso de uso inicia cuando el usuario Administración inserta, modifica o elimina dichos datos.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	El usuario Administración se encuentra autenticado.
<b>Postcondiciones</b>	Se registró, modificó o eliminó los datos de un Alumno Ayudante.

Flujo de eventos		
Flujo básico < Gestionar datos de los Alumnos Ayudantes >		
Actor		Sistema
1.	Selecciona la opción de “Alumno Ayudante”.	
2.		Muestra en pantalla una lista con las opciones de Insertar datos de un Alumno Ayudante, modificar datos de un Alumno Ayudante, eliminar datos de un Alumno Ayudante.
3.	Desea registrar, eliminar o modificar datos de un Alumno Ayudante.	
4.		<p>Se muestra en pantalla las opciones para para gestionar los datos de un Alumno Ayudante.</p> <p>a) Para insertar datos de un Alumno Ayudante ir a la sección “Insertar datos de un Alumno Ayudante”</p> <p>b) Para modificar datos de un Alumno Ayudante ir a la sección “Modificar datos de un Alumno Ayudante” que se encuentra en la vista de los Alumnos Ayudantes.</p> <p>c) Para eliminar datos de un Alumno Ayudante ir a la sección “Eliminar datos de un Alumno Ayudante”</p> <p>d) Para listar datos de un Alumno Ayudante ir a la sección “Listar datos de un Alumno Ayudante”</p>
Sección 1: “Listar datos de los Alumnos Ayudantes”		
Flujo Básico.		



2a. Listar Alumnos Ayudantes.		
Actor		Sistema
1.		Muestra en pantalla el listado con los Alumnos Ayudantes
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 2: “Insertar datos del Alumno Ayudante”</b>		
<b>Flujo básico &lt; Insertar datos del Alumno Ayudante &gt;</b>		
Actor		Sistema
1.	Presiona la opción “ Insertar datos del Alumno Ayudante “.	
2.		Muestra un formulario con los siguientes campos a introducir: - Nombre(s) y Apellido(s) del Alumno Ayudante - Año que cursa - Baja - Solapín
3.	Introduce los datos del Alumno Ayudante en los campos y presiona la opción “Insertar datos del Alumno Ayudante” o “Cancelar”.	
4.		Verifica que los campos estén llenos.
5.		Verifica que los datos introducidos sean correctos.
6.		Verifica que el Alumno Ayudante no se encuentre ya insertado
7.		Guarda los datos de Alumno Ayudante y muestra un mensaje “Los datos fueron guardados con éxito”. Finalizando así el Caso de Uso.

Flujos alternos		
<b>4a. Campos vacíos.</b>		
Actor		Sistema
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
Flujos alternos		
<b>5a. Datos incorrectos.</b>		
Actor		Sistema
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
Flujos alternos		
<b>6a. Datos de Alumno Ayudante registrados.</b>		
Actor		Sistema
1.		Muestra un mensaje “Datos guardados con éxito”.
Prototipo elemental de interfaz gráfica de usuario		
Sección 3: “Modificar datos de Alumno Ayudante”		
Flujo básico <Modificar datos de Alumno Ayudante >		
Actor		Sistema
1.	Presiona la opción “Modificar datos del Alumno Ayudante”.	
2.		Muestra un formulario con los siguientes campos a modificar: <ul style="list-style-type: none"> <li>- Nombre(s) y Apellido(s) del Alumno Ayudante</li> <li>- Año que cursa</li> <li>-Baja</li> </ul>

		- Solapín Y el botón “Editar datos del Alumno Ayudante”. Y el botón “Cancelar”.
3.	Actualiza los datos de Alumno Ayudante en los campos y presiona la opción “Modificar datos de Alumno Ayudante” o “Cancelar”.	
4.		Verifica que los campos estén llenos.
5.		Verifica que los datos introducidos sean correctos.
6.		Actualiza los datos del Alumno Ayudante y muestra un mensaje “Datos del Alumno Ayudante modificados con éxito”. Finalizando así el Caso de Uso.

**Flujos alternos**

**4a. Campos vacíos.**

Actor		Sistema
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.

**Flujos alternos**

**5a. Datos incorrectos.**

Actor		Sistema
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.

**Prototipo elemental de interfaz gráfica de usuario**

**Flujos alternos**

**2a. Marca botón ‘Cancelar’**

Actor		Sistema
1.		Retornar a la vista inicio donde se encuentra el listado de Alumno Ayudante “Listado de Alumno Ayudante”

**Prototipo elemental de interfaz gráfica de usuario**

**Sección 4: “Eliminar datos de Alumno Ayudante”**

**Flujo básico <Eliminar datos de Alumno Ayudante >**

Actor		Sistema
1.	Presiona la opción “Eliminar datos del Alumno Ayudante”.	
2.		Muestra la lista de los Alumnos Ayudantes
3.	Selecciona el Alumno Ayudante que desea eliminar y presiona la opción “Eliminar Alumno Ayudante” o “Cancelar”.	
4.		Elimina el Alumno Ayudante y muestra un mensaje “Alumno Ayudante eliminado con éxito”. Finalizando así el Caso de Uso.

**Prototipo elemental de interfaz gráfica de usuario**

<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	No aplica
<b>Asuntos pendientes</b>		No aplica

## II.4 Diagrama de Secuencia

Los diagramas de secuencia (DS) en el UML se utilizan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí (Sommerville, 2011).

A continuación, en las figuras 3, 4 y 5 se reflejan los diagramas de secuencia del caso de uso gestionar Alumno Ayudante.

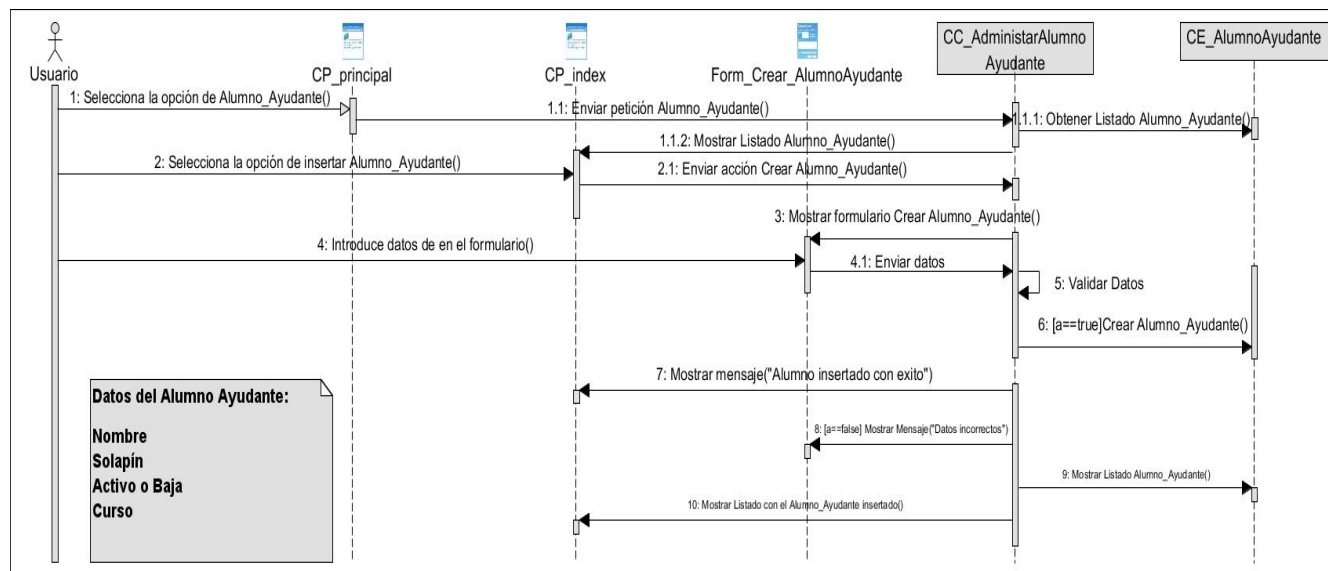


Figura 3: Diagrama de secuencia Crear Alumno Ayudante

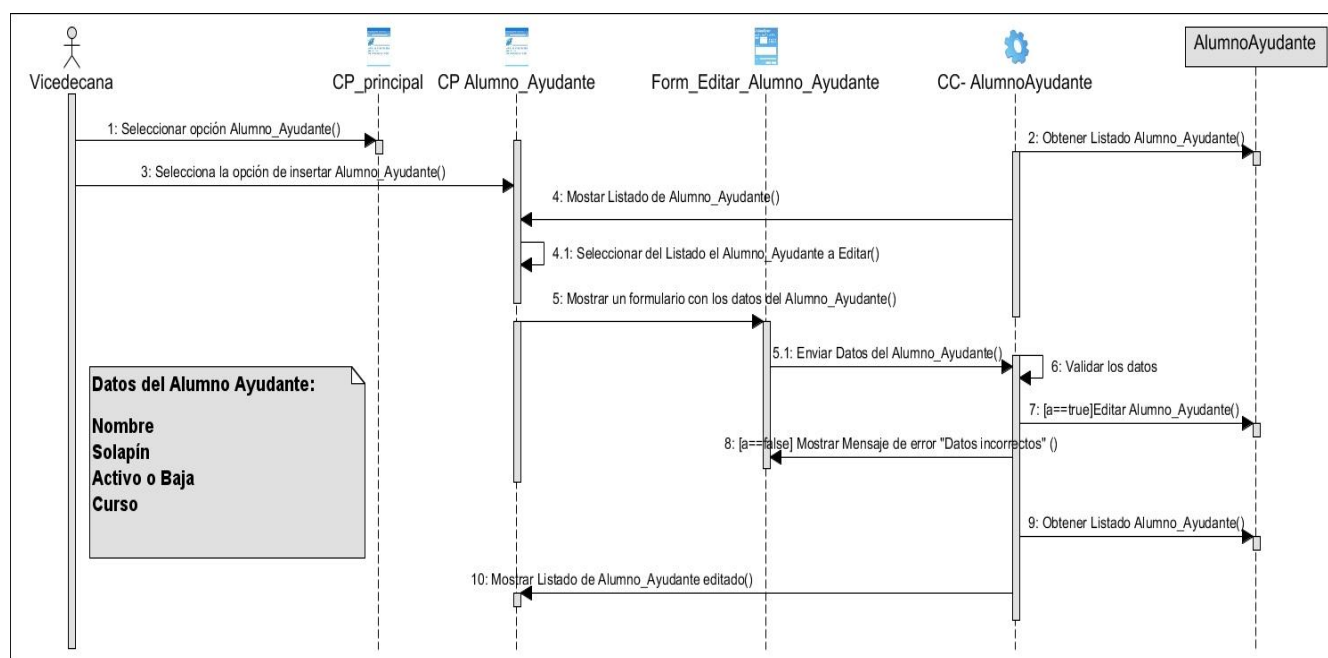


Figura 4: Diagrama de secuencia Modificar Alumno Ayudante

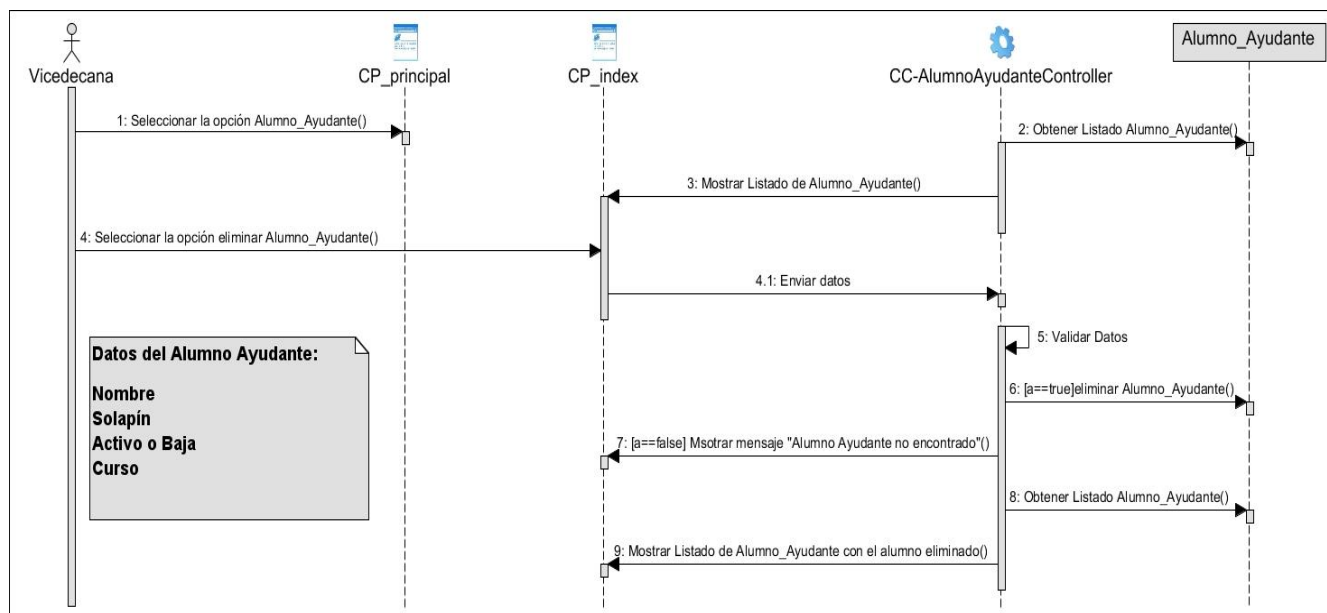


Figura 5: Diagrama de secuencias Eliminar Alumno Ayudante

### II.5 Modelo de Datos

El modelado de datos es el proceso de creación de una representación visual o esquema que define los sistemas de recopilación y administración de información de cualquier organización. Este esquema o modelo de datos ayuda a las diferentes partes interesadas, como analistas de datos, científicos e ingenieros, a crear una vista unificada de los datos de una organización. El modelo esboza los datos que recoge la empresa, la relación entre los distintos conjuntos de datos y los métodos que se usarán para almacenarlos y analizarlos (Amazon, 2022).

En la figura 6 se muestra el modelo de datos de la propuesta de solución.

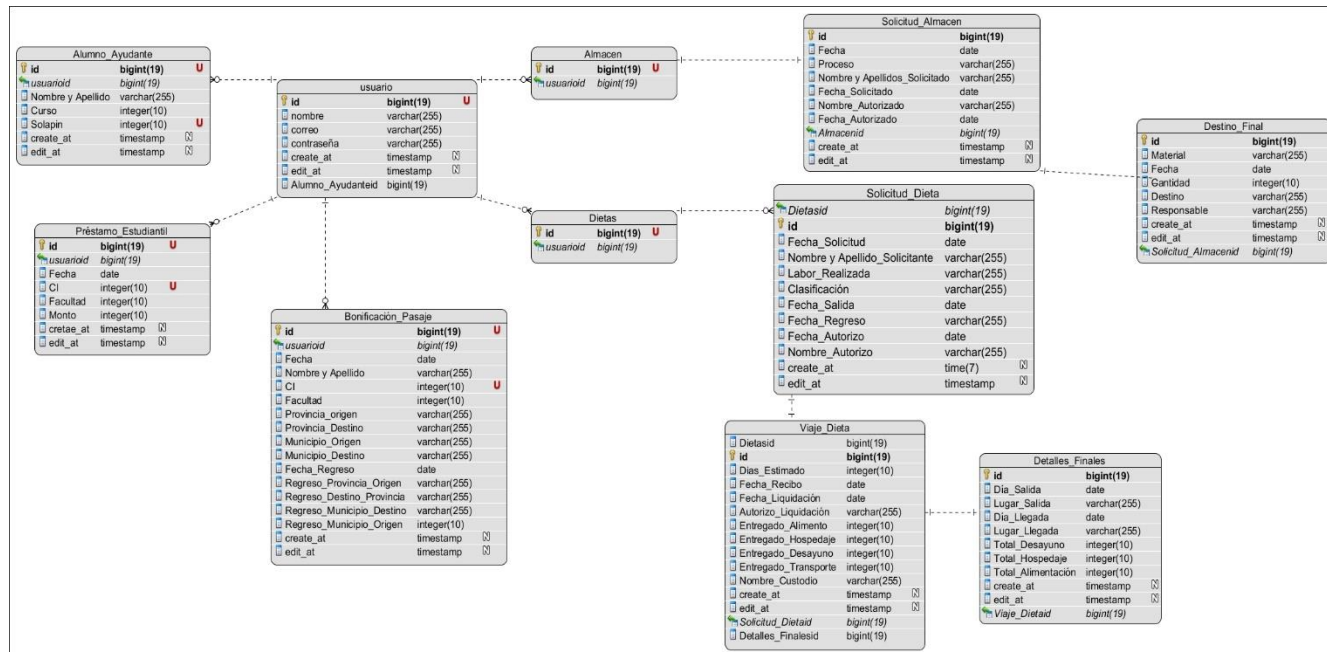


Figura 6: Modelo de Datos

### Descripción de las tablas del modelo:

- **Usuario:** Es la tabla encargada de almacenar los datos de un usuario en el sistema.
- **Dieta:** Tabla que almacena todo lo relacionado con las dietas.
- **Solicitud\_Dieta:** Tabla que almacena todo lo relacionado con las solicitudes de dietas.
- **Viaje\_Dieta:** Tabla que almacena todo lo relacionado con los viajes de las dietas.
- **Detalles\_Finales:** Tabla que almacena todo lo relacionado con los detalles finales de las dietas.
- **Almacén:** Tabla que almacena todo lo relacionado con la gestión de almacén.
- **Solicitud\_Almacén:** Tabla que almacena todas las solicitudes de almacén.
- **Destino\_Final:** Tabla que almacena los destinos finales de los materiales de Almacén.
- **Bonificación\_Pasaje:** Tabla que almacena los datos de las bonificaciones de pasaje.
- **Préstamo\_estudiantil:** Tabla que almacena los datos de los préstamos estudiantiles.
- **Alumno\_ayudante:** Tabla que almacena los datos de los Alumnos Ayudantes.

### II.4 Estilo Arquitectónico

Es un conjunto de principios que proveen un framework abstracto para una familia de sistemas que

promueve la reutilización de componentes y diseños mediante soluciones a problemas recurrentes. Es decir, un estilo arquitectónico es un conjunto de decisiones y principios que encajan como solución en determinados tipos de problemas que suelen repetirse en varios sistemas (BARRERA, 2018).

### II.4.1 Patrón Arquitectónico

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Permiten tener una estructura de código común, por lo que ahorran gran cantidad de tiempo en la construcción de software (Daudinot Hamiltón, 2015).

El framework de desarrollo Django está inspirado en la clásica arquitectura de programación MVC, pero en Django esta tendencia se aplica resumiendo una nueva estructura denominada MTV (*Template*: plantilla). Como resultado se obtiene lo que sería un “controlador” en MTC como “vista” en MTV y lo que tiene que ver con la “vista” se denomina “*template*” tal como se muestra en la Figura 7. Parte del trabajo del controlador lo maneja el mismo framework con sus funciones para manejar el acceso a la base de datos y además, por otro lado, el denominado “*template*” en los dos significan lo mismo exceptuando en el acceso a la base de datos que ya viene implementada en Django.

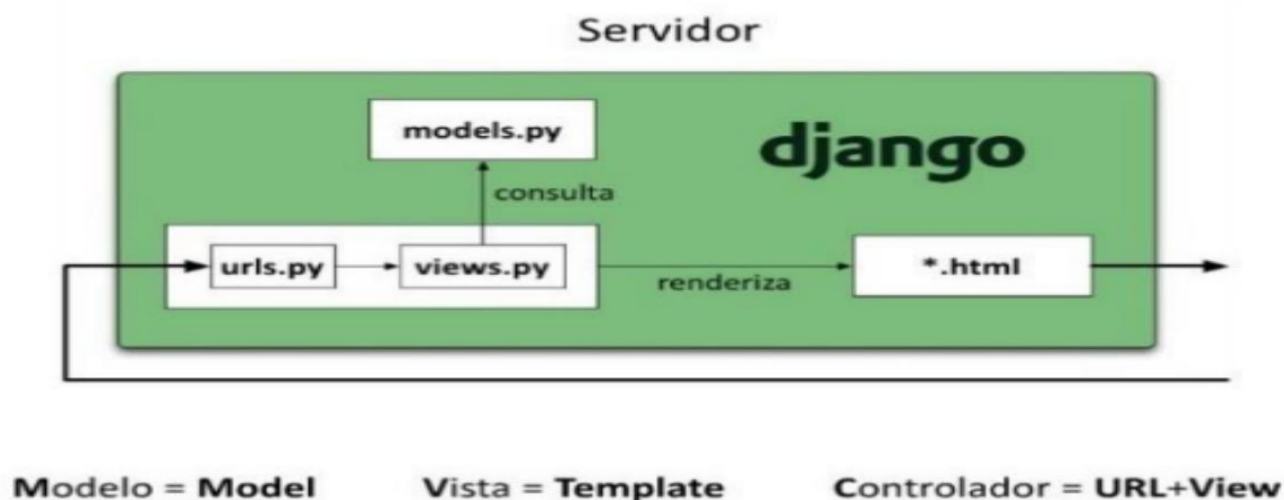


Figura 7: Arquitectura MTV

Se asumirá el estilo arquitectónico MTV de Django a partir de lo planteado anteriormente. El funcionamiento de este estilo comienza cuando el usuario a través del navegador envía la solicitud a la vista, que está contenida en el archivo views. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada, es como un puente entre los modelos y las plantillas que están contenidos en el archivo *models* y en la carpeta *templates*. Luego la vista para obtener los datos interactúa con el



modelo, la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos. En la figura 8, se muestra una representación de la arquitectura enfocada a la propuesta de solución.

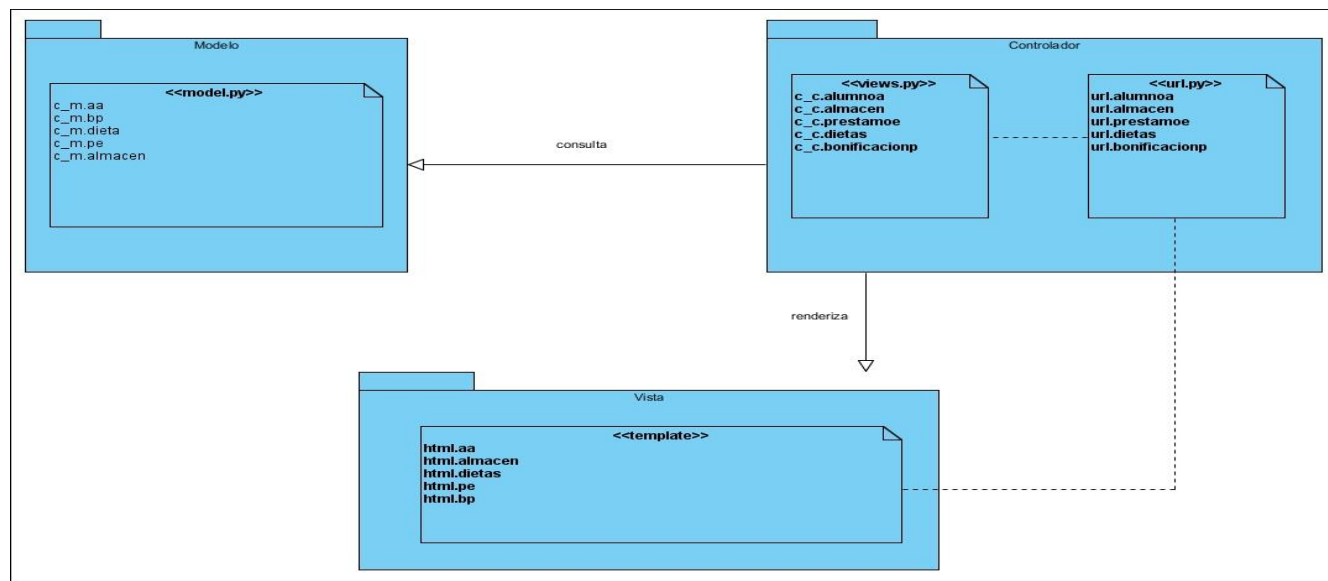


Figura 8: MVT del sistema

## II.4.2 Patrón de Diseño

Un patrón de diseño arquitectónico define una familia de sistemas en base a un esquema de organización estructural. Bajo esta perspectiva, los patrones arquitectónicos pueden ser vistos como los bloques constructivos básicos de las arquitecturas de software. Los patrones de diseño arquitectónico describen pares de elementos problema; solución con el objetivo de brindar soluciones abstractas a problemas recurrentes en un dominio específico. Las arquitecturas de sistemas de software complejos suelen basarse en la aplicación de más de un patrón de diseño. Este es el caso de las arquitecturas de aplicaciones web, en donde la construcción del diseño arquitectónico implica vincular múltiples componentes haciendo uso de diversos esquemas de organización (BLAS & LEONE, 2019).

### Patrones GRASP:

Los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar indicadamente el software orientados a objetos.

Los patrones GRASP utilizados en la propuesta de solución se enuncian a continuación :

**Creador:**

Este patrón es quien guía el proceso de asignación de responsabilidades relacionadas con la creación de objetos.

Este patrón se evidencia en:

- Archivo views.py contiene las funciones crearAA, crearBP, crearPE que se encarga de la creación de un Alumno Ayudante, de una Bonificación de Pasaje y de un Préstamo Estudiantil respectivamente.
- En el archivo viewsAlmacen.py en el que se encuentran las funciones crearSA y crearDF que se encargan de la creación de una Solicitud de Almacén y un Destino Final.
- También en el archivo viewsDieta.py se encuentran las funciones crear crearS, crearV, crearDF que se encargan de la creación de una Solicitud de Viaje, Detalles del Viaje y Detalles Finales respectivamente.

**Controlador:**

Responsable del manejo de los eventos del sistema, que no pertenece a la interfaz del usuario. Esto se evidencia en los archivos views.py, viewsDieta.py, viewsAlmacen.py que aíslan las capas internas del programa de las interfaces.

**Bajo Acoplamiento:**

El acoplamiento mide el grado en que una clase está conectada, tiene conocimiento o depende de otra. En el desarrollo de la solución se evidencia un patrón de bajo acoplamiento al realizar modificaciones en las clases Bonificacionp, PrestamoE, AlumnoA y que ninguna de estas se vea afectada, producto a que las mismas no están relacionadas, por tanto, no tiene repercusión sobre las demás.

**Alta Cohesión:**

Verifica que la información que almacena una clase debe ser coherente y debe estar, en la medida de

lo posible, relacionada con la clase, se encargan de realizar una función específica, para evitar la saturación. Además, la asignación de responsabilidades garantiza la alta cohesión en las clases existentes. Este patrón se pone de manifiesto en los archivos `modelsDieta.py` que relacionan las clases `SolicitudD`, `ViajeD` y `FinalD` con el objetivo de simplificar el trabajo.

### **Patrones GOF**

Los Patrones Pandilla de cuatro (GOF, por sus siglas en inglés), describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. Con este enfoque es sabido que todos los diseñadores de software usan patrones, no importa si trabajan desde el paradigma estructurado o desde el nuevo paradigma orientado a objetos (CRUZ, 2018).

### **Decorador:**

El patrón decorador en Python son mecanismos de encapsulación para establecer en las aplicaciones desarrolladas con este lenguaje una clara separación entre mecanismos funcionales y de confiabilidad. En otras palabras, los decoradores son, en esencia, mecanismos de separación de código en el sistema (DZICHKOVSKII, 2020). Este se evidencia en el archivo `base.html` del que heredan todos los archivos `html` contenidos en la carpeta `templates` añadiendo nuevas funcionalidades.

### **Conclusiones del capítulo**

En este capítulo fueron descritos los procesos que intervienen en el negocio y se arribó a las siguientes conclusiones:

- La definición de los requisitos funcionales y no funcionales, evidenciaron las características que el sistema debe cumplir, a partir de las exigencias del cliente.
- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

## CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN ECONÓMICO DE LA FACULTAD 1

En el presente capítulo se realiza la fase de implementación, además, se describen los estándares de codificación empleados para la misma. Posteriormente se realiza un plan de pruebas, para comprobar el cumplimiento de los requisitos funcionales y verificar la calidad del sistema.

### III.1 Diagrama de componente

El Diagrama de Componente describe cómo se organizan los componentes de acuerdo con el lenguaje de programación utilizado y al entorno de implementación, y cómo dependen los componentes entre sí. Un componente es el empaquetamiento físico de un elemento del diseño, como lo son las clases en el modelo de diseño. Según el lenguaje de programación utilizado, los componentes que se generan tienen sus características específicas, pues como se ha expresado anteriormente ellos no son más que el empaquetamiento físico de un elemento (Jimenez, 2018). A continuación, en las figuras 9, 10 y 11 se muestran los diagramas de Componente para los casos de uso Gestionar Alumno Ayudante, Gestionar Préstamo Estudiantil y Gestionar Bonificación de Pasaje.

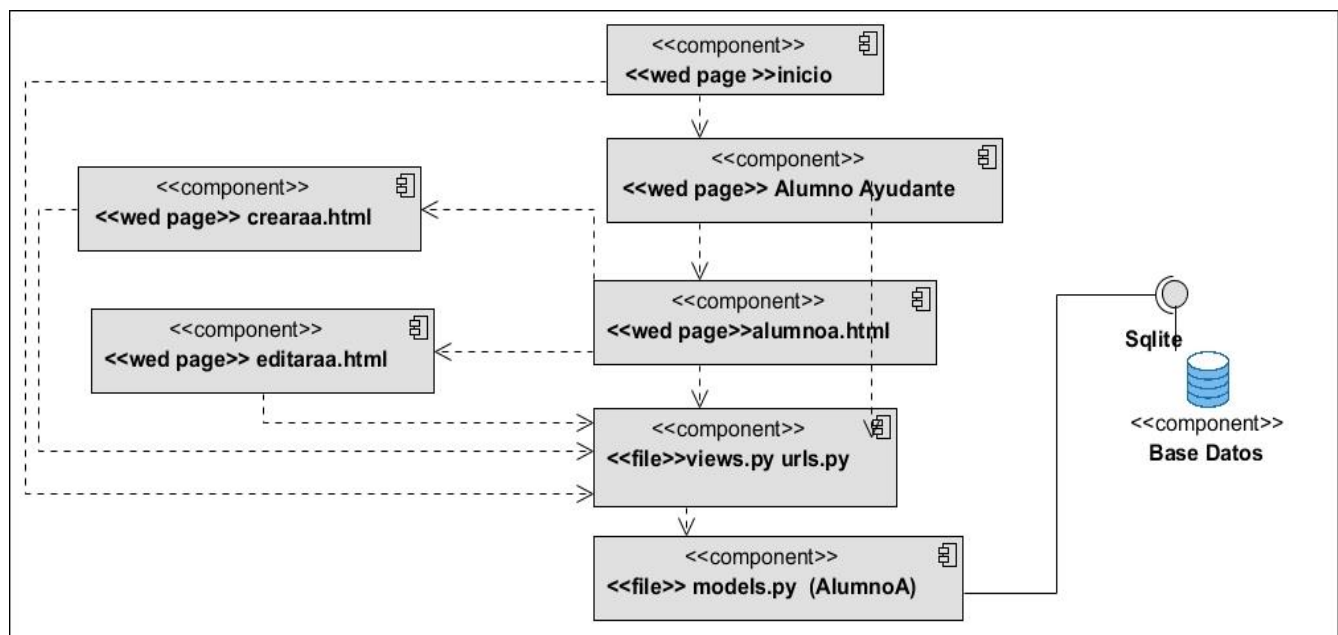


Figura 9: Diagrama de Componente Gestionar Alumno Ayudante

Tabla 7: Descripción del Diagrama de Componente Gestionar Alumno Ayudante

Componentes		Descripción
<b>Modelo</b>	models.py	El archivo models.py es el que contiene la Clase Alumna que almacena los Alumnos Ayudantes en el sistema.
<b>Vista</b>	Alumno Ayudante	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan los Alumnos Ayudantes.
	alumnaa.html	Este componente se encarga de mostrar un listado con los Alumnos Ayudantes insertados en la base de datos, así como mostrar las opciones necesarias para gestionar los Alumnos Ayudantes.
	creaeaa.html	Este componente se encarga de registrar un nuevo Alumno Ayudante en la base de datos con los datos necesarios del mismo.
	editaraa.html	Este componente se encarga de editar un Alumno Ayudante en la base de datos con los datos necesarios del mismo.
<b>Controlador</b>	views.py urls.py	Estos componentes se encargan de realizar todos los métodos relacionados a la gestión de los Alumnos Ayudantes y del proceso de renderización de todas las plantillas HTML del sistema.

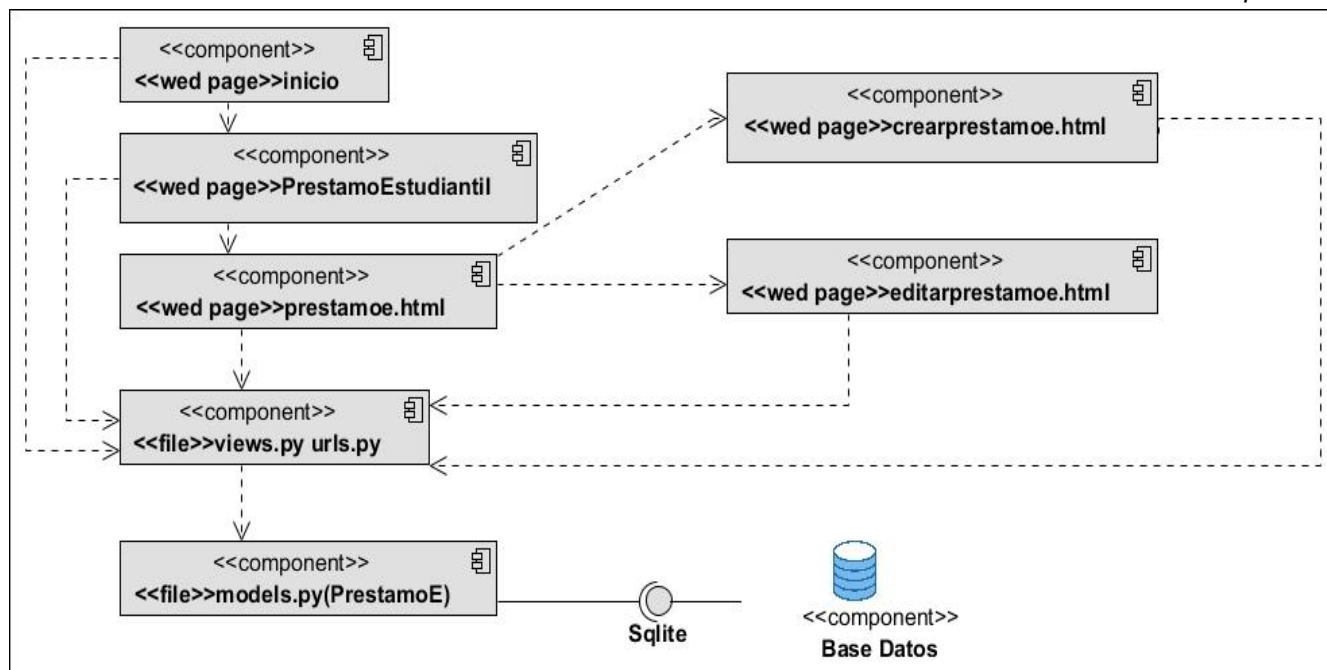


Figura 10: Diagrama de Componentes Gestionar Préstamo Estudiantil

Tabla 8: Descripción del Diagrama de Componentes Préstamo Estudiantil

Componentes		Descripción
<b>Modelo</b>	models.py	El archivo models.py es el que contiene la Clase Prestamoe que almacena los Préstamos Estudiantiles en el sistema.
<b>Vista</b>	Préstamo Estudiantil	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan los Préstamos Estudiantiles.
	prestamo. html	Este componente se encarga de mostrar un listado con los Préstamos Estudiantiles insertados en la base de datos, así como mostrar

		las opciones necesarias para gestionar los Préstamos Estudiantiles.
	creaepe. html	Este componente se encarga de registrar un nuevo Préstamo Estudiantil en la base de datos con los datos necesarios del mismo.
	editarpe. html	Este componente se encarga de editar un Préstamo Estudiantil en la base de datos con los datos necesarios del mismo.
<b>Controlador</b>	views.py urls.py	Estos componentes se encargan de realizar todos los métodos relacionados a la gestión de los Préstamos Estudiantiles y del proceso de renderización de todas las plantillas HTML del sistema.

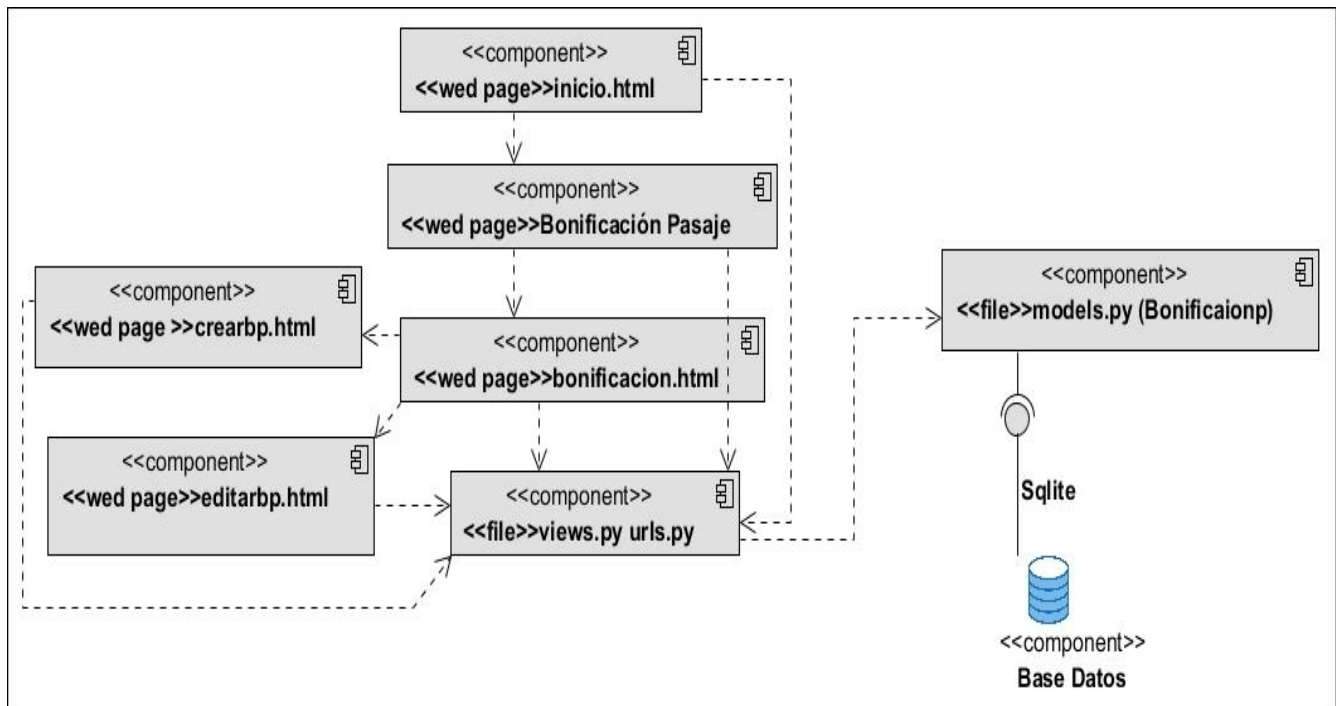


Figura 11: Diagrama de Componentes Gestionar Bonificación de Pasaje

Tabla 9: Descripción del Diagrama de Componentes Bonificación de Pasaje

	Componentes	Descripción
<b>Modelo</b>	models.py	El archivo models.py es el que contiene la Clase Bonificacionp que almacena las Bonificaciones de Pasaje en el sistema.
<b>Vista</b>	Bonificación de Pasaje	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las Bonificaciones de Pasaje.
	bonificacionp. html	Este componente se encarga de mostrar un listado con las Bonificaciones de Pasaje insertados en la base de datos, así como mostrar las opciones necesarias para gestionar las Bonificaciones de Pasaje.
	creaebp. html	Este componente se encarga de registrar una nueva Bonificación de Pasaje en la base de datos con los datos necesarios de la misma.
	editarbp. html	Este componente se encarga de editar una Bonificación de Pasaje en la base de datos con los datos necesarios de la misma.
<b>Controlador</b>	views.py urls.py	Estos componentes se encargan de realizar todos los métodos relacionados a la gestión de las Bonificaciones de Pasaje y del proceso



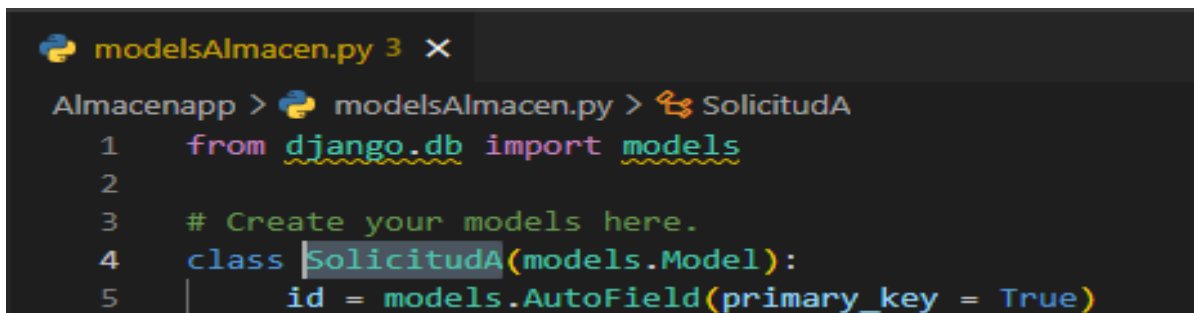
		de renderización de todas las plantillas HTML del sistema.
--	--	--

### III.2 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento (Guilarte Domínguez, 2019).

#### Normas a seguir para el trabajo con las clases del Sistema

- Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas. Los nombres serán simples y descriptivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL, HTTP, entre otras.).
- Los nombres de las clases se escriben con la primera letra de cada palabra que lo compone en mayúscula.
- Se utilizan los comentarios en caso de ser necesario para explicar la utilidad de la clase.



```

modelsAlmacen.py 3 x
Almacenapp > modelsAlmacen.py > SolicitudA
1  from django.db import models
2
3  # Create your models here.
4  class SolicitudA(models.Model):
5      id = models.AutoField(primary_key = True)

```

## Normas a seguir para nombrar las variables y métodos

- Los nombres deben ser descriptivos y concisos. No usar grandes frases y solo abreviaturas pequeñas.
- Al definir una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- El identificador para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto se divide cada palabra con mayúscula inicial.
- Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas.

```
def inicio(request):
    return render(request, "Sistemaapp/inicio.html")

def data(request):
    return render(request, "Sistemaapp/data.html")
```

## Normas a seguir para la declaración de clases / interfaces

- Durante el desarrollo de clases / interfaces se deben seguir las siguientes reglas de formateo: No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros.
- El carácter inicio de bloque ("{"") debe aparecer al final de la línea que contiene la sentencia de declaración.
- El carácter fin de bloque ("}") se sitúa en una nueva línea tabulada al mismo nivel que su correspondiente sentencia de inicio de bloque, excepto cuando la sentencia sea nula, en tal caso se situará detrás de "{".
- Los métodos se separarán entre sí mediante una línea en blanco.

```
<!-- Contenido Cambiante -->
{% block content %}

{% endblock %}
```

## Normas a seguir para los espacios en blanco en expresiones y sentencias

- Evitar utilizar espacios en blanco en las siguientes situaciones: dentro de paréntesis, corchetes y llaves, antes de una coma, un punto y coma o dos puntos, dentro del paréntesis que comienza la lista de argumentos en la llamada a una función. Deben rodearse con exactamente un espacio los siguientes operadores binarios:
- Asignación (=).
- Asignación de aumentación (+=, -=, entre otros).
- Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not).
- Expresiones lógicas (and, or, not).
- No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.

```
def login_view(request):  
    if request.method == 'POST':  
        username = request.POST.get('username')  
        password = request.POST.get('password')  
        user = authenticate(username=username, password=password)
```

### Norma a seguir para comentarios

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
- Si un comentario es corto el punto final puede omitirse.
- Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código Javascript.



A continuación, se muestra una descripción de los componentes del diagrama de despliegue propuesto para el SGEF1:

**Cliente PC/Móvil:** esta Pc de escritorio o móvil a través del protocolo HTTPS se conecta al servidor web para acceder a la página web a través de cualquier navegador.

**Servidor de aplicaciones web + BD :** este nodo representa el servidor de aplicaciones y servidor de Base de datos.

**Impresora:** consiste en una dispositivo que permite imprimir los documentos deseados por el usuario.

### III.4 Evaluación de la propuesta de solución

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponda con los requisitos del cliente. Durante la etapa de implementación, pueden introducir algunos errores y pueden pasarse por alto algunos elementos que son importantes para el adecuado funcionamiento del sistema. Por tal motivo, es esencial llevar a cabo la fase de validación, en la cual, a través de varios tipos y métodos de pruebas de software se logra evaluar la calidad y descubrir errores en el sistema (González Rodríguez, 2017). A continuación, en la tabla 10 se describen las pruebas realizadas al sistema.

Tabla 10: Tipos de Pruebas

Tipo de prueba	Método (técnica) de prueba	Validación
Funcional	Casos de prueba (Caja Negra)	Valida las funcionalidades diseñadas para el Sistema.
Seguridad	Software Acunetix	Valida la confidencialidad, integridad y disponibilidad de los datos en el sistema.

Carga y estrés	Software Apache JMeter	Valida el comportamiento del sistema con distintos niveles de usuarios concurrentes y el consumo excesivo de sus recursos.
----------------	------------------------	--

### III.4.1 Pruebas de Caja Negra

Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna del código, detalles de implementación o escenarios de ejecución internos en el software (Pressman, 2002b).

Con el propósito de realizar las pruebas de funcionales o también llamada pruebas de Caja Negra al SGEF1, se diseñó un conjunto de casos de pruebas, referentes a los casos de uso Crear Alumno Ayudante y Crear Préstamo Estudiantil, pertenecientes a los requisitos funcionales de prioridad alta, Registrar Alumno Ayudante y Registrar Préstamo Estudiantil. A continuación, en las tablas 11 y 13 se muestra los casos de prueba para evaluar el requisito Crear Alumno Ayudante y Crear Préstamo Estudiantil. En las celdas de las tablas 12 y 14 de los casos de pruebas se pueden encontrar los valores **V**, para datos válidos y **I**, para datos inválidos.

Tabla 11: Descripción del caso de uso: Crear Alumno Ayudante

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	NO	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
2	Curso	Campo de número	NO	Se debe escribir un número del 1 al 5.

3	Baja	Campo SI/No	SI	En caso de seleccionar la opción el estudiante se guardará como baja y en caso contrario como activo.
4	Solapín	Campo Alfanumérico	NO	Se deben escribir el código del Solapín que puede contener 7 caracteres de tipo numérico y alfabético.

Tabla 12: Prueba de caja negra: Alumno Ayudante

Caso de prueba 1: RF4_Crear Alumno Ayudante						
Condición de ejecución: el usuario debe tener el permiso de Administrador						
Escenario	Descripción	1	2	3	4	Respuesta del sistema
EC 1.1 Registrar un Alumno Ayudante de forma correcta	Interfaz con el formulario para llenar los datos de un Alumno Ayudante, si todos son correctos, se registra el Alumno Ayudante en el Sistema	V	V	V	V	Agrega los datos del Alumno Ayudante y muestra un mensaje de notificación.
		Rocío del Carmen Ortiz Vega	5	SI	E16186	
EC 1.2 Registrar un Alumno Ayudante	Interfaz con el formulario para llenar los datos del	I	I	V	V	Comprueba si alguno de los campos está vacío, si lo está, muestra un mensaje debajo de cada campo

<p>con campos vacíos</p>	<p>Alumno Ayudante, si existe algún campo vacío que sea requerido, se muestra un mensaje debajo de cada campo señalando que el campo es requerido</p>			<p>SI</p>	<p>E16186</p>	<p>vacío señalando que el campo es requerido y que debe de llenarlo</p>
<p>EC 1.3 Registrar un Alumno Ayudante con campos erróneos</p>	<p>Interfaz con el formulario para llenar los datos de un Alumno Ayudante, si existe algún campo erróneo, se muestra un mensaje debajo de cada campo señalando que el campo tiene un error.</p>	<p><b>V</b></p>	<p><b>I</b></p>	<p><b>V</b></p>	<p><b>I</b></p>	<p>Comprueba si los campos tienen errores, si tienen errores muestra un mensaje debajo de los campos diciendo que los campos tienen un error y tienen que ser rellenados correctamente.</p>
		<p>Rocio del Carmen Ortiz Vega</p>	<p>6</p>	<p>SI</p>	<p>123456 7</p>	



Tabla 13: Descripción del caso de uso: Préstamo Estudiantil

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Fecha	Campo de fecha	NO	Se deben escribir la fecha según el formato indicado.
2	Nombre	Campo de texto	NO	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
3	CI	Campo numérico	NO	Se deben escribir 11 números.
4	Facultad	Campo SI/No	SI	Se debe seleccionar un número correspondiente a la facultad.
5	Monto	Campo numérico	NO	Se deben escribir números.

Tabla 14: Prueba de caja negra: Préstamo Estudiantil

Caso de prueba 1: RF4_Crear Préstamo Estudiantil							
Condición de ejecución: el usuario debe tener el permiso de Administrador							
Escenario	Descripción	1	2	3	4	5	Respuesta del Sistema
EC 1.1 Registrar un Préstamo Estudiantil de forma correcta	Interfaz con el formulario para llenar los datos de un Préstamo Estudiantil, si todos son correctos, se registra el	V	V	V	V	V	Agrega los datos del Préstamo Estudiantil y muestra un mensaje de notificación.
		1/10/2022	Sofía Peña Rodríguez	00110618177	1	6000.0	

	Préstamo Estudiantil en el Sistema						
EC 1.2 Registrar un Préstamo Estudiantil con campos vacíos	Interfaz con el formulario para llenar los datos del Préstamo Estudiantil, si existe algún campo vacío que sea requerido, se muestra un mensaje debajo de cada campo señalando que el campo es requerido	<b>V</b>	<b>I</b>	<b>V</b>	<b>I</b>	<b>V</b>	Comprueba si alguno de los campos está vacío, si lo están, muestra un mensaje debajo de cada campo vacío señalando que el campo es requerido y que debe de llenarlo.
		1/10/2022		00110618177		6000.0	
EC 1.3 Registrar un Préstamo Estudiantil con campos erróneos	Interfaz con el formulario para llenar los datos de un Préstamo Estudiantil, si existe algún campo erróneo, se	<b>V</b>	<b>I</b>	<b>I</b>	<b>V</b>	<b>V</b>	Comprueba si los campos tienen errores, si tienen errores muestra un mensaje debajo de los campos diciendo que los campos tienen un error y tienen que ser rellenados correctamente.
		1/10/2022	sofí@PeñaRodríguez	dos10618177	1	6000.0	

	muestra un mensaje de- bajo de cada campo señalando que el campo tiene un error.									
--	--	--	--	--	--	--	--	--	--	--

En total, se planificaron tres iteraciones de prueba. Como resultado final de las pruebas funcionales, se obtuvo, en una primera iteración, un total de once (11) no conformidades, divididas en ocho (8) de ortografía, dos (2) de funcionalidad y una (1) de validación. De estas, se resolvieron en la segunda iteración ocho (8), y tres (3) quedaron pendientes. En una segunda iteración, no se identifican nuevas no conformidades y de las tres (3) pendientes, solo quedo una (1) que se mantuvo para la próxima iteración, donde fue resuelta y no se detectaron nuevas no conformidades, obteniendo, de esta manera, resultados satisfactorios. La siguiente gráfica, muestra los resultados antes descritos:

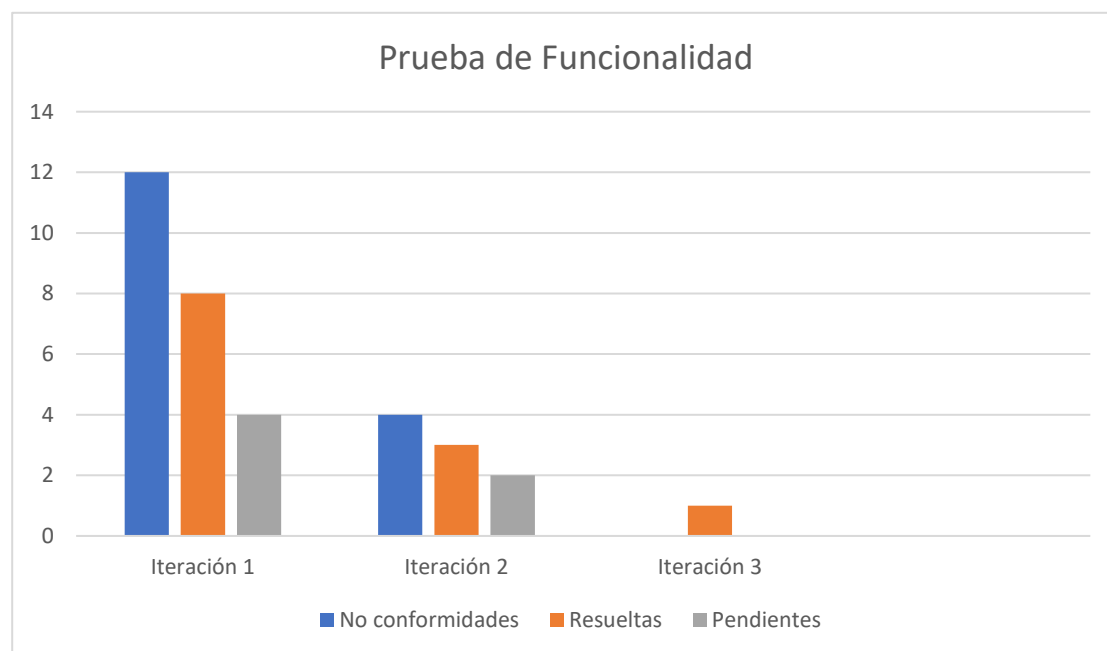


Figura 13: Pruebas Funcionales

Las no conformidades de validación estuvieron relacionadas en la primera iteración con guardar los datos de un Alumno Ayudante, el sistema no permitía registrar los datos sin que seleccionara opción de baja en el formulario de creación de un Alumno Ayudante, esta no conformidad fue resuelta al permitir que el campo baja, el cual estaba definido como campo requerido, pudiese tomar valor nulo. La segunda no conformidad estaba relacionada con el formulario encargado de la creación de un Préstamo Estudiantil, el cual permitía introducir caracteres extraños en el campo que correspondía a el nombre del estudiante, esto se resolvió validando mediante una expresión regular que en este campo solo se pudieran introducir caracteres alfabéticos.

Las no conformidades de funcionalidad estaban relacionadas a que no se permitía acceso a la interfaz Préstamo Estudiantil, esto se debía a que la URL se encontraba mal configurada, esto se solucionó arreglando correctamente esta ruta.

### **III.4.2 Pruebas de Seguridad**

Las pruebas de seguridad permiten realizar una evaluación de los sistemas desde el punto de vista externo y sin conocimiento previo de este. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación. Además, buscan medir la confidencialidad, integridad y disponibilidad de los datos, partiendo de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Una vez terminadas las pruebas es posible medir y cuantificar los riesgos a los cuales se ven expuestos los sistemas de la infraestructura interna y externa (Guilarte Domínguez, 2019).

Con el objetivo de evaluar la seguridad de la solución se emplea la herramienta *Acunetix WVS 9.5*. Durante las dos iteraciones realizadas, la aplicación detectó un total de 22 no conformidades, divididas en cuatro secciones principalmente: nivel alto (0 no conformidades), nivel medio (12 no conformidades), nivel bajo (4 no conformidades) y de carácter informal (6 no conformidades).

De las de nivel medio, destacó el uso de protocolo no seguro para el envío de datos, así como los mensajes de error que se muestra en el modo DEBUG de Django para el desarrollo. Para las de nivel bajo, se detectó problemas para la protección contra ataques de fuerza bruta a la página de autenticación, así como directorios que pueden ser accesibles directamente sin pasar la autenticación y la protección de las cookies y las sesiones en el navegador. De tipo Informativo se obtuvo varios errores, relacionados a los campos correspondientes a los detalles finales y los campos detalles del viaje de las Dietas. Todas estas deficiencias fueron corregidas en la primera iteración, y para una segunda, no se

identificó ninguna nueva, por lo cual se obtuvo finalmente una herramienta que cumple con los requisitos de seguridad definidos para la misma. Los resultados antes descritos, se muestran a continuación en la siguiente gráfica:

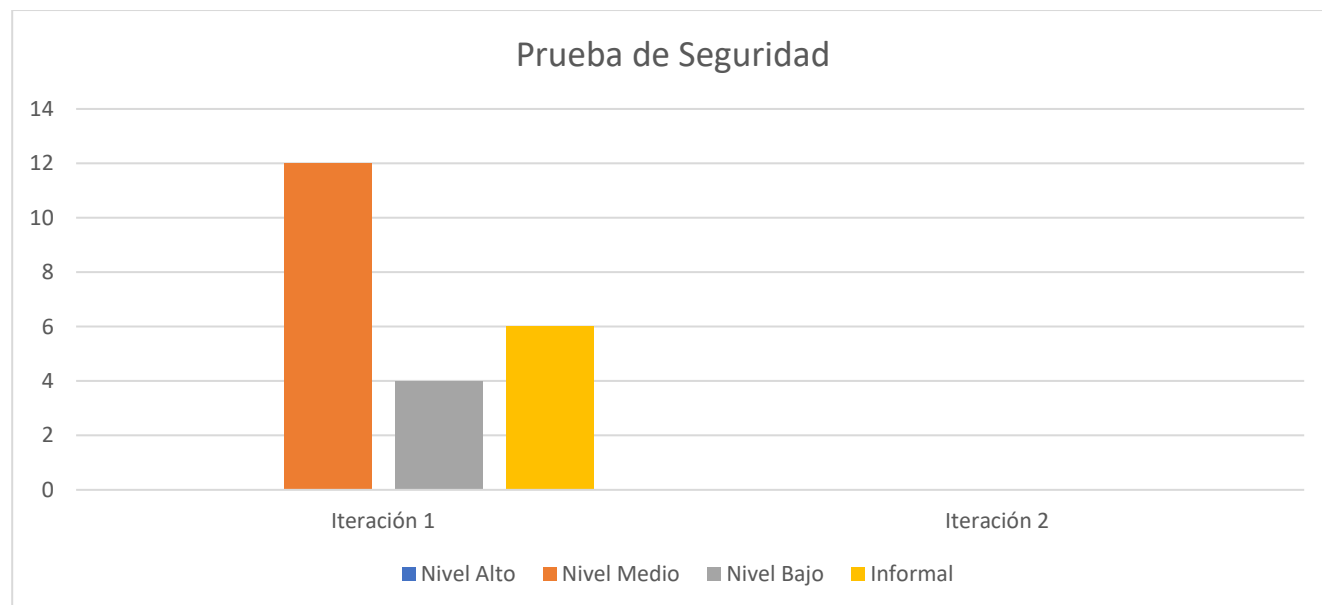


Figura 14: Pruebas de Seguridad

### III.4.3 Pruebas de Rendimiento

La prueba de carga y estrés se refiere, generalmente, a la práctica de comprobar el comportamiento de una aplicación mediante cargas o entradas pesadas. Las mismas se realizan con el fin de verificar si el sistema satisface los requisitos de rendimiento para situaciones críticas como pueden ser: la cantidad límite de usuarios accediendo de forma concurrente a los servicios brindados, documentos extremadamente grandes, cantidad de transacciones que se pueden procesar de forma concurrente cada minuto, tiempo de respuesta, entre otros (Rodríguez, 2017).

El tipo de prueba de rendimiento que se le realizó a la propuesta de solución fue las pruebas de carga y estrés, en la cual se utilizó como herramienta el Apache *JMeter* en su versión 2.10, diseñado para pruebas de carga de comportamientos funcionales y la medición del rendimiento. Las pruebas se realizaron desde un ordenador con 2GB de RAM, microprocesador Intel Core i3 con 2.00 GHz y sistema operativo GNU/Linux. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas a la propuesta de solución:

**Muestra:** Cantidad de peticiones realizadas.

**Media:** Tiempo promedio en milisegundos en el que se obtienen los resultados.

**Mediana:** Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

**Min:** Tiempo mínimo transcurrido en milisegundos para las muestras de la URL dada.

**Max:** Tiempo máximo transcurrido en milisegundos para las muestras de la URL dada.

**Línea 90 %:** Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

**% Error:** Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

**Rendimiento (Rend):** El rendimiento se mide en cantidad de solicitudes por segundo.

**Kb/s:** Velocidad de carga de las páginas.

Tabla 15: Pruebas de rendimiento realizada con Apache JMeter

Usua- rios	Mues- tras	Media	Me- diana	Min	Max	Línea 90%	%Error	Rend	Kb/s
5	10	78	75	8	199	152	0.00%	157.0	586.70
15	30	364	358	10	876	746	0.00%	152.7	570.58

Las pruebas realizadas muestran que la propuesta de solución es capaz de responder 10 peticiones iniciadas por 5 usuarios conectados simultáneamente en un tiempo promedio de 75 milisegundos (0.075 segundos) aproximadamente, con 0.00%de error, esto evidencia que el sistema puede procesar la carga esperada.

Luego con el objetivo de analizar el comportamiento de la propuesta de solución en condiciones mayores, se realizó una prueba de estrés para un conjunto de 15 usuarios conectados simultáneamente. En este caso, la propuesta de solución responde a las 30 peticiones en un tiempo promedio de 364 milisegundos (0.364 segundos) aproximadamente, con un porcentaje de error de 0.00%. Este resultado demostró nuevamente que el Sistema puede procesar con éxito la carga esperada.

### III.5. Pruebas de aceptación

Para la revisión final de las especificaciones del diseño y de la implementación se realizaron las pruebas de aceptación con el objetivo de determinar diferentes tipos de errores y resolverlos (Pressman, 2006), para ello se emplean las pruebas Alfa y Beta.

- Pruebas Alfa: Se desarrollan en conjunto, el desarrollador y los usuarios finales. Con el objetivo de registrar los errores y problemas del uso del software.
- Pruebas Beta: Se realiza en el sistema final donde se va a desplegar el software. A diferencia de la prueba alfa, la prueba beta es una aplicación del software en su ambiente final. El objetivo es registrar todos los problemas que se encuentran durante la prueba.

Para la aplicación de las pruebas de aceptación se realizó la prueba alfa, aplicada en un ambiente controlado, donde se probó por parte del cliente y asistida por el desarrollador, el cumplimiento de los requisitos requeridos para el sistema, obteniendo como resultado la aceptación del cliente de la propuesta de solución. (Fabre, Ortega, & Farrat, 2019)

#### III.5.1. Satisfacción de los usuarios

Con el objetivo de evaluar el sistema implementado se utiliza la técnica de ladov, esta técnica evalúa el nivel de satisfacción del usuario, permitiendo conocer si la solución propuesta cumple con las expectativas esperadas. Esta técnica constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas (preguntas 1, 2 y 3) que se intercalan dentro de un cuestionario (Ver Anexo). Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov”, el cual se muestra a continuación en la tabla:

3- ¿Satisface sus necesidades el sistema propuesto?	1- ¿Se pueden gestionar correctamente los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1 sin una adecuada organización, centralización y disponibilidad de los recursos?		
	NO	NO SÉ	SÍ
	2- ¿Utilizaría usted el sistema propuesto a la hora de gestionar los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la		

	Facultad 1?								
	SÍ	NO SÉ	NO	SÍ	NO SÉ	NO	SÍ	NO SÉ	NO
Me satisface mucho.	1	2	6	2	2	6	6	6	6
No me satisface tanto.	2	3	2	3	3	3	6	3	6
Me da lo mismo.	3	3	3	3	3	3	3	4	3
Me disgusta más de lo que me satisface.	6	3	6	3	4	4	3	4	4
No me satisface nada.	6	6	6	6	4	4	6	4	5
No sé qué decir.	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción. Escala de satisfacción:

- Clara satisfacción (1).
- Más satisfecho que insatisfecho (2).
- No definida (3).
- Más insatisfecho que satisfecho (4).
- Clara insatisfacción (5).
- Contradictoria (6).

Para medir el grado de satisfacción se tomó una muestra de 10 personas: dos con conocimiento de los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1 y el resto con experiencia en el desarrollo de sistemas web sobre el framework Django. La técnica de ladov permite conocer el Índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y -1 de la siguiente



forma:

Tabla 16: Escala de Satisfacción

INDICE	ESCALA
+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho.
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción.

La satisfacción grupal se calcula utilizando la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

Donde:

- A, B, C, D, E representan el número de sujetos con índice individual 1, 2, 3 o 6, 4, 5 respectivamente.
- N representa el número total de sujetos del grupo.

Esto permite reconocer las categorías grupales:

- Insatisfacción: desde (-1) hasta (-0,5)
- Contradictorio: desde (-0,49) hasta (+0,49)
- Satisfacción: desde (+0,5) hasta (1).

Luego de haber aplicado la técnica, calculando el ISG se obtiene como resultado 0.9 lo que significa una clara satisfacción con el uso del sistema de gestión de la Economía de la Facultad 1.

$$ISG = \frac{8(+1) + 2(+0,5) + 0(0) + 0(-0,5) + 0(-1)}{10} = 0.9$$

A partir de la operacionalización de las variables definidas en la investigación y luego de la aplicación de la técnica de ladov se puede demostrar que el sistema cumple con el nivel de satisfacción del usuario, representado por un valor ISG =0.9.

### **Conclusiones del capítulo:**

- Con la aplicación de los estándares de codificación se facilita la comprensión y la legibilidad del código, proporcionando un código seguro y mantenible.
- La realización de las pruebas de seguridad, funcionalidad y rendimiento, permitió la obtención de un sistema con calidad.
- La aplicación de la técnica ladov permitió validar la propuesta de solución, demostrando un índice de satisfacción general de 0.9, lo que demuestra un alto grado de satisfacción de los usuarios sobre la propuesta de solución.

## CONCLUSIONES FINALES

Una vez realizada la fundamentación teórica que sustentó este trabajo, definidas las características del sistema, efectuada la implementación y validación del mismo, se obtuvieron resultados que le permiten al autor presentar las siguientes conclusiones:

- ✓ El análisis de los principales conceptos asociados a la investigación, fundamentó los elementos teóricos que propiciaron un mejor entendimiento de los procesos de gestión económica del Vicedecanato de Economía y Administración de la Facultad 1.
- ✓ El análisis de los sistemas homólogos, permitió identificar la necesidad de desarrollar un nuevo sistema para la Gestión Económica de la Facultad 1.
- ✓ EL análisis de las herramientas y tecnologías existentes, aportó la base tecnológica para el desarrollo de la propuesta de solución.
- ✓ La aplicación de pruebas de software, permitió validar el correcto funcionamiento del sistema, demostrando a través de la técnica ladov que el sistema propuesto contribuye a la gestión económica de la Facultad 1.

## **RECOMENDACIONES**

Para dar continuidad a la presente investigación se recomienda:

- Adicionar una funcionalidad que permita a los directivos de las áreas solicitar a la Vicedecana de Economía y Administración las dietas mediante el sistema.

## REFERENCIAS BIBLIOGRÁFICAS

- Ain, M. S. (2022). *KRONOX*. Obtenido de <https://kronox.es/>
- Alizo, M. A. (Septiembre de 2010). *Gestión económica vinculada con la innovación y adquisición de tecnológica en los emprendimientos emergentes de negocio tipo PYME*. Obtenido de [http://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S1315-99842010000300007](http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1315-99842010000300007)
- Amazon. (13 de Septiembre de 2022). *AWS Amazon*. Obtenido de <https://aws.amazon.com/es/what-is/data-modeling/>
- Aristigüi, L. L. (2017). *Sistema experto basado en modelos de rasgos de contacto bilineales para la predicción de propiedades biológicas en proteínas*. Base Datos. Tesis.Universidad de las Ciencias Informáticas.Facultad 6.
- BARRERA, G. M. (2018). *Estilo arquitectónico para aplicaciones IoT*. Obtenido de <http://hdl.handle.net/10419/203805>
- BLAS, M. J., & LEONE, H. P. (2019). *Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube* .
- CONSORTIUM, W. W. (1999). *HTML 4.01. Specification*.
- Cordero, R. (2019). *Simulación de un mercado real a través del software de gestión de la empresa ANFIX para la asignatura "Dirección de operaciones en distribución comercial"*.
- CRUZ, V. H. (2018). *Identificación de patrones creacionales en diseño de software mediante métodos difusos*. UNIVERSIDAD POLITÉCNICA DE TULANCINGO .
- Cuba, M. d. (2020). *Artículo.No240*. La Habana: Gaceta Oficial de la República de Cuba.
- Cuba, M. d. (2020). *Resolución No. 2/2018 (GOC-2018-460-O25)* . La Habana: Gaceta Oficial de la República de Cuba.
- Daudinot Hamiltón, Y. (2015). *Perfil de usuario para el Sistema de Planificación de Actividades SIPAC*. Universidad de las Ciencias Informáticas.
- Django. (2022). *Django. El framework web para perfeccionistas con plazos*. Obtenido de <https://www.djangoproject.com/>
- DZICHKOVSKII, I. (2020). *Mecanismos reutilizables para mejorar la confiabilidad de aplicaciones desarrolladas con Python*.
- Fabre, A. F., Ortega, N. S., & Farrat, Y. R. (6 de diciembre de 2019). El proceso de validación mediante la Técnica de ladov en cursos por encuentros. *Revista Ingeniería Agrícola*.

- GARCERANT, I. (10 de Julio de 2008). *Modelo de Dominio. Tecnología y Synergix*. Obtenido de <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio>
- GoCardless. (Diciembre de 2021). *GoCardless*. Obtenido de *Cómo llevar la gestión económica empresarial*: <https://gocardless.com/es/guias/consejos/es-como-llevar-la-gestion-economica-empresarial/>
- GoCardless, C. (2022). *GoCardless*. Obtenido de <https://gocardless.com/es/guias/consejos/es-como-llevar-la-gestion-economica-empresarial/>
- GÓMEZ, R. (2018). *Aplicación web para la gestión y búsqueda de eventos*.
- González Rodríguez, A. (2017). *Subsistema de gestión de perfil de usuario para el buscador Orión*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/7963>
- González Rodríguez, A. (2017). *Subsistema de gestión de perfil de usuario para el buscador Orión*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/7963>
- Guilarte Domínguez, C. R. (2019). *Sistema para la gestión de procesos en la Dirección de Extensión Universitaria*. . Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/10152>
- IBM. (11 de Septiembre de 2022). *IBM*. Obtenido de <https://www.ibm.com>
- Jimenez, G. (2018). APLICACIÓN WEB PARA LA GESTIÓN DE LA INFORMACIÓN EN LA ESCUELA DE CAPACITACIÓN DE LA CONSTRUCCIÓN DE PINAR DEL RÍO, CUBA . *Revista Científica Multidisciplinaria*.
- Justicia, M. d. (2020). *Resolución No.338*. La Habana: Gaceta Oficial de la República de Cuba.
- Lizmary Ricardo Herrera, R. V. (enero de 2019). *La gestión económica financiera en las universidades. Impacto en la calidad*. Obtenido de <http://scielo.sld.cu/pdf/rus/v11n1/2218-3620-rus-11-01-56.pdf>
- Microsoft. (8 de Junio de 2022). Obtenido de <https://code.visualstudio.com/learn/overview>
- Ministerio de Educación Superior. (2021). *Resolución No 47/22*. Habana: Departamento Jurídico del Ministerio de Educación Superior.
- Ministerio de Finanzas y Precios. (2007). *Resolución No 11*. Habana: Gaceta Oficial de la República de Cuba.
- Ministerio de Justicia. (2007). *Instrucción No. 2/07*. La Habana: Gaceta Oficial de la República de Cuba.
- Ministerio de Justicia. (2020). *Resolución No.338*. La Habana: Gaceta Oficial de la República de Cuba.

- Ministerio de Justicia de Cuba. (2020). *RESOLUCIÓN No.108 /20 Capítulo 1*. La Habana: Gaceta Oficial de la República de Cuba.
- Mora, S. L. (2016). *Programación de aplicaciones web*.
- NISKA, C. (2014). *Extending Bootstrap*.
- Pérez, J. E. (2019). *Introducción a JavaScript*.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*.
- PUIG, J. C. (2013). *CSS3 y Javascript avanzado*. Obtenido de <https://openlibra.com/es/book/download/css3-y-javascript-avanzado>
- Python. (2022). *Python*. Obtenido de <https://www.python.org/>
- Rodríguez, A. G. (2017). *Subsistema de gestión de perfil de usuario para el buscador*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/7963>
- Rodríguez, L. (2021). *Los sistemas contables automatizados y su utilización en las entidades*. La Habana.
- Romero, R. P. (2022). *Factorial*. Obtenido de <https://factorialhr.es/>
- Sánchez, N. B. (2019). *Análisis y diseño de un sistema de gestión económica para el área de administración de la Facultad 6*. La Habana.
- Sánchez, T. R. (2015). *Metodología de desarrollo para la Actividad productiva de la UCI*. Habana.
- SÁNCHEZ, T. R. (3 de June 2015 ). *Metodología de desarrollo para la Actividad productiva de la UCI*.
- Santana, M. A. (2018). *La actividad económica en la red de instituciones del Ministerio de Educación*. Obtenido de <http://scielo.sld.cu/pdf/rces/v37n1/rces07118.pdf>
- Santana, M. I. (Diciembre de 2021). *Transformaciones en la gestión económico-financiera de las universidades cubanas*. Obtenido de [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2073-60612022000100011](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2073-60612022000100011)
- SOLÓRZANO, J. A. (2018). *Desarrollo de una aplicación web multiplataforma usando el framework Django, para publicitar eventos sociales, aplicado en el municipio del cantón Morona*. Escuela Superior Politécnica de Chimborazo.
- Sommerville. (2005). *Ingeniería de Software—Ian Sommerville 7a Edición*.
- SQLite. (2022). *SQLite*. Obtenido de <https://www.sqlite.org/index.html>
- SQLite, C. (13 de Septiembre de 2022). *SQLite*. Obtenido de <https://www.sqlite.org/index.html>

Tamara, R. (2015). *Metodología de desarrollo para el proceso productivo de la UCI*. La Habana.

Velazquez, C. D. (2016). *El proceso de gestión y la gestión económica en las empresas*. Obtenido de Observatorio de la Economía Latinoamericana: <https://www.eumed.net/coursecon/ecolat/cu/2011/dsvl.htm>

VENTURA, J. N. (2021). *JQUERY*.

WIRFS-BROCK, B. A. (2020). JavaScript: the first 20 years.

*Workana LLC* . (2021). Obtenido de Workana LLC : <https://i.workana.com/glosario/que-es-html-5/>



## ANEXOS

### **Anexo 1. Entrevista al cliente para conocer la necesidad del desarrollo de la propuesta de solución y definir los requisitos funcionales y no funcionales.**

Estimado profesor: Se necesita de su cooperación en una investigación para una tesis de pregrado. Por ello, sería de gran ayuda que respondiera lo siguiente:

1. ¿Cuáles son los procesos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1?
2. ¿Qué son las dietas, las bonificaciones de pasaje y los préstamos estudiantiles?
3. ¿Cuántas personas intervienen en estos procesos?
4. ¿Cuál es la frecuencia con la que se realizan estos procesos?
5. ¿Qué tipo de informe o reportes se generan después de haber realizado alguno de estos procesos?
6. ¿Por qué surge la necesidad de gestionar los procesos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1?

### **Anexo 2:**

Estimado(a): Lea con cuidado cada pregunta antes de responder. Le agradecemos su participación y franqueza al decirnos honestamente lo que piensa sobre el uso de un Sistema de Gestión Económica de la Facultad 1.

1- ¿Considera usted que se pueden gestionar correctamente los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1 sin una adecuada centralización, disponibilidad y calidad de la información?

No  No sé  Sí

2- ¿Considera usted que el sistema propuesto presenta una interfaz gráfica atractiva?

\_ No \_ No sé \_ Sí

---

3- ¿Satisface sus necesidades el sistema propuesto?

\_ Me satisface mucho. \_ No me satisface tanto. \_ Me da lo mismo. \_ Me disgusta más de lo que me satisface. \_ No me satisface nada. \_ No sé qué decir.

---

4- ¿Modificaría usted algún elemento del sistema propuesto para la gestión de los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1? Argumente.

---

5- ¿Considera usted que el sistema propuesto contribuye a la gestión de los procesos económicos que se realizan en el Vicedecanato de Economía y Administración de la Facultad 1? Argumente.

**Anexo 3:**

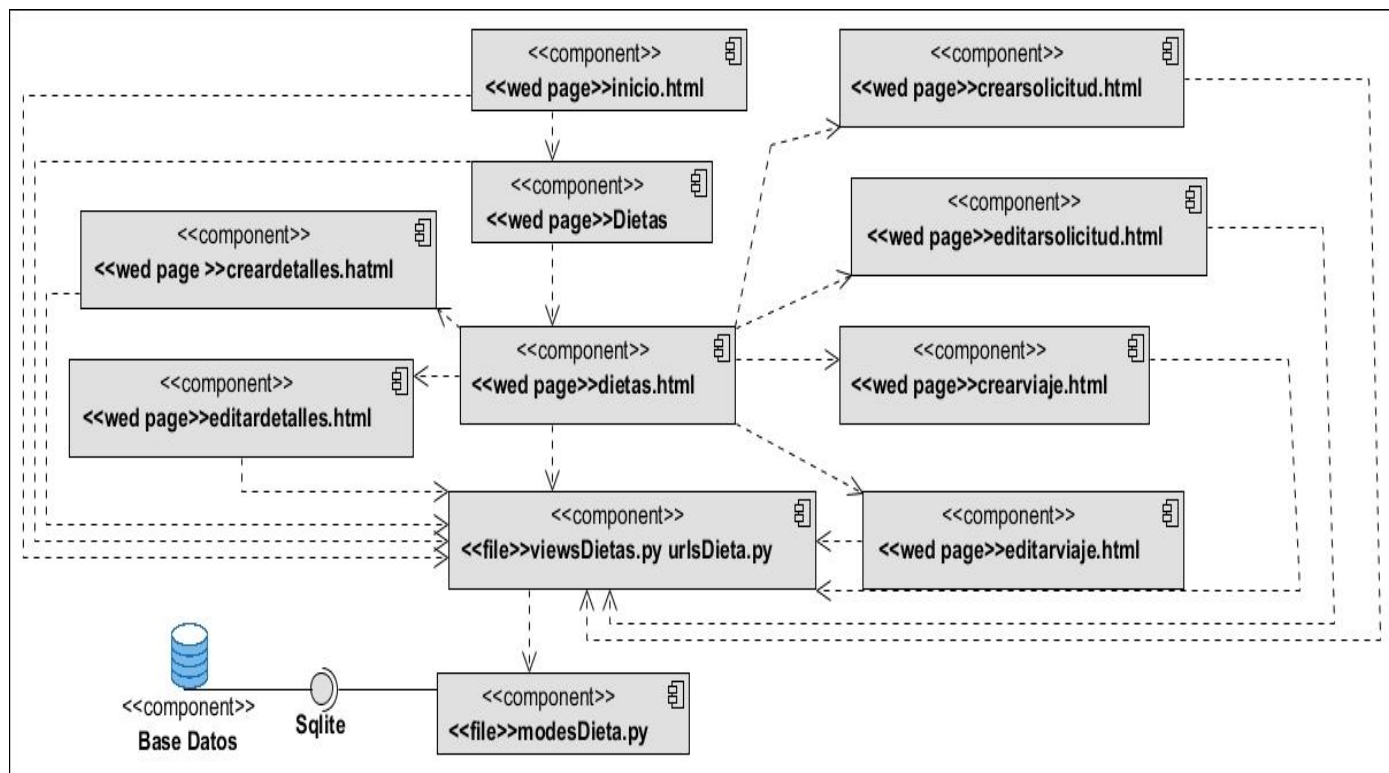


Figura 15: Diagrama de Componentes Dietas

Tabla 17: Descripción del Diagrama de Componentes Dietas

Componentes		Descripción
<b>Modelo</b>	modelsDietas.py	El archivo modelsDietas.py es el que contiene la Clase SolicitudD, ViajeD y FinalD que almacena las Dietas en el sistema.
	Dietas	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las Dietas.
	dietas.html	Este componente se encarga de mostrar un listado con las Dietas insertados en la base de datos, así como mostrar las opciones necesarias para gestionar las Dietas.
	crearsolicitud.html	Este componente se encarga de registrar una nueva Solicitud en la base de datos con los datos necesarios de la misma.
	editarsolicitud.html	Este componente se encarga de editar una Solicitud en la base de datos con los datos necesarios de la misma.
	crearviaje.html	Este componente se encarga de registrar un nuevo Viaje en la base de datos con los datos necesarios del mismo.

	editarviaje. html	Este componente se encarga de editar un Viaje en la base de datos con los datos necesarios del mismo.
	creardetalles.html	Este componente se encarga de registrar un nuevo Detalle en la base de datos con los datos necesarios del mismo.
<b>vista</b>	editardetalles. html	Este componente se encarga de editar un Detalle en la base de datos con los datos necesarios del mismo.
<b>Controlador</b>	viewsDietas.py urlsDietas.py	Estos componentes se encargan de realizar todos los métodos relacionados a la gestión de las Dietas y del proceso de renderización de todas las plantillas HTML del sistema.

**Anexo 4:**

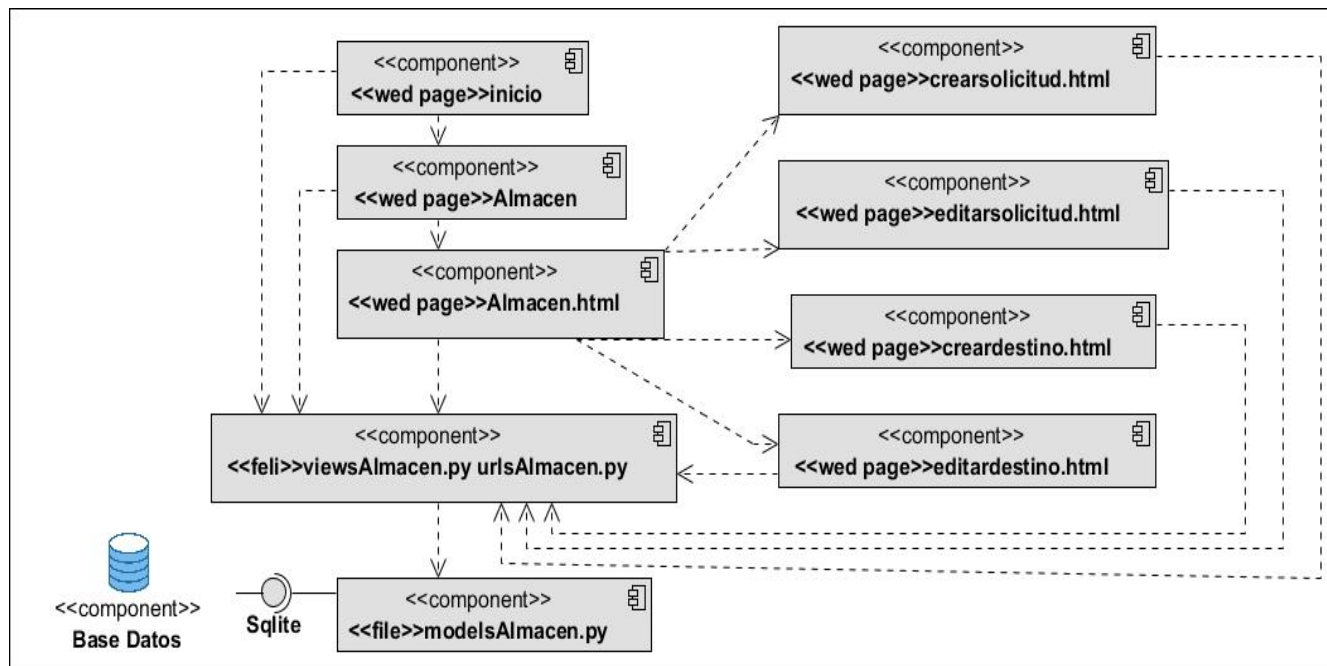


Figura 16: Diagrama de Componentes Almacén

Tabla 18: Descripción del Diagrama de Componentes Almacén

Componentes		Descripción
<b>Modelo</b>	modelsAlmacen.py	El archivo modelsAlmacen.py es el que contiene la Clase SolicitudA, y DestinoF que almacenan todo lo referente a los Almacenes en la Base Dato.
<b>Vista</b>	Almacén	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan los Almacenes.
	almacen.html	Este componente se encarga de mostrar un listado con los Almacenes insertados en la base de datos, así como mostrar las opciones necesarias para gestionar los Almacenes.
	crearsa.html	Este componente se encarga de registrar una nueva Solicitud en la base de datos con los datos necesarios de la misma.
	editarsa. html	Este componente se encarga de editar una Solicitud en la base de datos con los datos necesarios de la misma.
	creardf.html	Este componente se encarga de registrar un nuevo Destino Final en la base de datos con los datos necesarios del mismo.

	editardf. html	Este componente se encarga de editar un Destino Final en la base de datos con los datos necesarios del mismo.
<b>Controlador</b>	viewsAlmacen.py urlsAlmacen.py	Estos componentes se encargan de realizar todos los métodos relacionados a la gestión de los Almacenes y del proceso de renderización de todas las plantillas HTML del sistema.

**Anexo 5:***Tabla 19: Caso de uso Bonificación de Pasaje*

<b>Objetivo</b>	Permite insertar, modificar, eliminar y listar los datos de Bonificación de Pasaje.	
<b>Actores</b>	Administración.	
<b>Resumen</b>	El caso de uso inicia cuando el usuario Administración inserta, modifica o elimina dichos datos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario Administración se encuentra autenticado.	
<b>Postcondiciones</b>	Se registró, modificó o eliminó los datos de una Bonificación de Pasaje.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Gestionar datos de las Bonificaciones de Pasaje &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
1.	Selecciona la opción de “Bonificación de Pasaje”.	

2.		Muestra en pantalla una lista con las opciones de Insertar datos de una Bonificación de Pasaje, modificar datos de una Bonificación de Pasaje, eliminar datos de una Bonificación de Pasaje.
3.	Desea registrar, eliminar o modificar datos de una Bonificación de Pasaje.	
4.		<p>Se muestra en pantalla las opciones para para gestionar los datos de una Bonificación de Pasaje.</p> <p>a) Para insertar datos de una Bonificación de Pasaje ir a la sección “Insertar datos de una Bonificación de Pasaje”</p> <p>b) Para modificar datos de una Bonificación de Pasaje ir a la sección “Modificar datos de una Bonificación de Pasaje” que se encuentra en la vista de las Bonificaciones de Pasaje.</p> <p>c) Para eliminar datos de una Bonificación de Pasaje ir a la sección “Eliminar datos de una Bonificación de Pasaje”</p> <p>d) Para listar datos de una Bonificación de Pasaje ir a la sección “Listar datos de una Bonificación de Pasaje”</p>
<b>Sección 1: “Listar datos de las Bonificaciones de Pasaje”</b>		
<b>Flujo Básico.</b>		
<b>2a. Listar las Bonificaciones de Pasaje</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra en pantalla el listado con las Bonificaciones de Pasaje

Prototipo elemental de interfaz gráfica de usuario		
Sección 2: “Insertar datos de la Bonificación de Pasaje”		
Flujo básico < Insertar datos de la Bonificación de Pasaje >		
Actor	Sistema	
8.	Presiona la opción “ Insertar datos de la Bonificación de Pasaje “.	
9.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> <li>- Fecha</li> <li>- Nombre y Apellido</li> <li>- Carnet de Identidad</li> <li>- Facultad</li> <li>- Provincia Origen</li> <li>- Provincia Destino</li> <li>- Municipio Origen</li> <li>- Municipio Destino</li> <li>- Fecha Regreso</li> <li>- Regreso Provincia Origen</li> <li>- Regreso Provincia Destino</li> <li>- Regreso Municipio Origen</li> <li>- Regreso Municipio Destino</li> </ul>
10.	Introduce los datos de la Bonificación de Pasaje en los campos y presiona la opción “Insertar datos de la Bonificación de Pasaje” o “Cancelar”.	
11.		Verifica que los campos estén llenos.
12.		Verifica que los datos introducidos sean correctos.
13.		Verifica que el Bonificación de Pasaje no se encuentre ya insertado



14.		Guarda los datos de Bonificación de Pasaje y muestra un mensaje “Los datos fueron guardados con éxito”. Finalizando así el Caso de Uso.
<b>Flujos alternos</b>		
<b>4a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>6a. Datos de Bonificación de Pasaje registrados.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra un mensaje “Datos guardados con éxito”.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 3: “Modificar datos de Bonificación de Pasaje”</b>		
<b>Flujo básico &lt;Modificar datos de Bonificación de Pasaje &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
7.	Presiona la opción “Modificar datos de la Bonificación de Pasaje”.	
8.		Muestra un formulario con los siguientes campos a modificar: - Fecha

		<ul style="list-style-type: none"> <li>- Nombre y Apellido</li> <li>- Carnet de Identidad</li> <li>- Facultad</li> <li>- Provincia Origen</li> <li>- Provincia Destino</li> <li>- Municipio Origen</li> <li>- Municipio Destino</li> <li>- Fecha Regreso</li> <li>- Regreso Provincia Origen</li> <li>- Regreso Provincia Destino</li> <li>- Regreso Municipio Origen</li> <li>- Regreso Municipio Destino</li> </ul> <p>Y el botón “Editar datos de la Bonificación de Pasaje”.</p> <p>Y el botón “Cancelar”.</p>
9.	Actualiza los datos de Bonificación de Pasaje en los campos y presiona la opción “Modificar datos de Bonificación de Pasaje” o “Cancelar”.	
10.		Verifica que los campos estén llenos.
11.		Verifica que los datos introducidos sean correctos.
12.		Actualiza los datos de la Bonificación de Pasaje y muestra un mensaje “Datos de la Bonificación de Pasaje modificados con éxito”.
Finalizando así el Caso de Uso.		
<b>Flujos alternos</b>		
<b>4a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>

1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Flujos alternos</b>		
<b>2a. Marca botón 'Cancelar'</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Retornar a la vista inicio donde se encuentra el listado de Bonificación de Pasaje "Listado de Bonificación de Pasaje"
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 4: "Eliminar datos de la Bonificación de Pasaje"</b>		
<b>Flujo básico &lt;Eliminar datos de Bonificación de Pasaje &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
5.	Presiona la opción "Eliminar datos de la Bonificación de Pasaje".	
6.		Muestra la lista de las Bonificaciones de Pasaje
7.	Selecciona la Bonificación de Pasaje que desea eliminar y presiona la opción "Eliminar Bonificación de Pasaje" o "Cancelar".	

8.		Elimina la Bonificación de Pasaje y muestra un mensaje “Bonificación de Pasaje eliminado con éxito”. Finalizando así el Caso de Uso.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 5: “Exportar en formato pdf los datos de Bonificación de Pasaje”</b>		
<b>Flujo básico &lt; Exportar a formato pdf los datos de Bonificación de Pasaje &gt;</b>		
	Presiona la opción “Exportar a pdf los datos de las Bonificaciones de Pasaje”.	
		Muestra la lista de las Bonificaciones de Pasaje
	Selecciona la Bonificación de Pasaje que desea exportar y presiona la opción “Exportar a pdf los datos de la Bonificación de Pasaje.” o “Cancelar”.	
		Exportar a formato pdf pdf la Bonificación de Pasaje. Finalizando así el Caso de Uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	No aplica
<b>Asuntos pendientes</b>		No aplica

**Anexo 6:***Tabla 20: Descripción del Caso de uso Préstamo Estudiantil*

<b>Objetivo</b>	Permite insertar, modificar, eliminar y listar los datos de los Préstamos Estudiantiles.
-----------------	--

<b>Actores</b>	Administración.	
<b>Resumen</b>	El caso de uso inicia cuando el usuario Administración inserta, modifica o elimina dichos datos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario Administración se encuentra autenticado.	
<b>Postcondiciones</b>	Se registró, modificó o eliminó los datos de un Préstamo Estudiantil.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Gestionar datos de los Préstamos Estudiantiles &gt;</b>		
<b>Actor</b>	<b>Sistema</b>	
1.	Selecciona la opción de “Préstamo Estudiantil”.	
2.		Muestra en pantalla una lista con las opciones de Insertar datos de un Préstamo Estudiantil, modificar datos de un Préstamo Estudiantil, eliminar datos de un Préstamo Estudiantil.
3.	Desea registrar, eliminar o modificar datos de un Préstamo Estudiantil.	
4.		Se muestra en pantalla las opciones para para gestionar los datos de un Préstamo Estudiantil. a) Para insertar datos de un Préstamo Estudiantil ir a la sección “Insertar datos de un Préstamo Estudiantil” b) Para modificar datos de un Préstamo Estudiantil ir a la sección “Modificar datos de un Préstamo Estudiantil” que se encuentra en la vista de los Préstamos Estudiantiles.

		<p>c) Para eliminar datos de un Préstamo Estudiantil ir a la sección “Eliminar datos de un Préstamo Estudiantil”</p> <p>d) Para listar datos de un Préstamo Estudiantil ir a la sección “Listar datos de un Préstamo Estudiantil”</p>
<b>Sección 1: “Listar datos de los Préstamos Estudiantiles”</b>		
<b>Flujo Básico.</b>		
<b>2a. Listar Préstamos Estudiantiles</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra en pantalla el listado con los Préstamos Estudiantiles
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 2: “Insertar datos del Préstamo Estudiantil”</b>		
<b>Flujo básico &lt; Insertar datos del Préstamo Estudiantil &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
15.	Presiona la opción “ Insertar datos del Préstamo Estudiantil “.	
16.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> <li>- Nombre(s) y Apellido(s)</li> <li>- Carnet de identidad</li> <li>- Facultad</li> <li>- Monto</li> </ul>
17.	Introduce los datos del Préstamo Estudiantil en los campos y presiona la opción “Insertar datos del Préstamo Estudiantil” o “Cancelar”.	
18.		Verifica que los campos estén llenos.

19.		Verifica que los datos introducidos sean correctos.
20.		Verifica que el Préstamo Estudiantil no se encuentre ya insertado
21.		Guarda los datos de Préstamo Estudiantil y muestra un mensaje “Los datos fueron guardados con éxito”. Finalizando así el Caso de Uso.
<b>Flujos alternos</b>		
<b>4a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>6a. Datos de Préstamo Estudiantil registrados.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra un mensaje “Datos guardados con éxito”.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 3: “Modificar datos de Préstamo Estudiantil”</b>		
<b>Flujo básico &lt;Modificar datos de Préstamo Estudiantil &gt;</b>		
<b>Actor</b>		<b>Sistema</b>

13.	Presiona la opción “Modificar datos del Préstamo Estudiantil”.	
14.		Muestra un formulario con los siguientes campos a modificar: - Nombre(s) y Apellido(s) - Carnet de identidad - Facultad - Monto Y el botón “Editar datos del Préstamo Estudiantil”. Y el botón “Cancelar”.
15.	Actualiza los datos de Préstamo Estudiantil en los campos y presiona la opción “Modificar datos de Préstamo Estudiantil” o “Cancelar”.	
16.		Verifica que los campos estén llenos.
17.		Verifica que los datos introducidos sean correctos.
18.		Actualiza los datos del Préstamo Estudiantil y muestra un mensaje “Datos del Préstamo Estudiantil modificados con éxito”. Finalizando así el Caso de Uso.
<b>Flujos alternos</b>		
<b>4a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		



<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Flujos alternos</b>		
<b>2a. Marca botón 'Cancelar'</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Retornar a la vista inicio donde se encuentra el listado de Préstamo Estudiantil "Listado de Préstamo Estudiantil"
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 4: "Eliminar datos de Préstamo Estudiantil"</b>		
<b>Flujo básico &lt;Eliminar datos de Préstamo Estudiantil &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
9.	Presiona la opción "Eliminar datos del Préstamo Estudiantil".	
10.		Muestra la lista de los Préstamos Estudiantiles
11.	Selecciona el Préstamo Estudiantil que desea eliminar y presiona la opción "Eliminar Préstamo Estudiantil" o "Cancelar".	
12.		Elimina el Préstamo Estudiantil y muestra un mensaje "Préstamo Estudiantil eliminado con éxito". Finalizando así el Caso de Uso.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica

	<b>CU Extendidos</b>	No aplica
<b>Requisitos no funcionales</b>		No aplica
<b>Asuntos pendientes</b>		No aplica

**Anexo 7:**

*Tabla 21: Descripción del Caso de uso Dieta*

<b>Objetivo</b>	Permite insertar, modificar, eliminar y listar los datos de las Dietas.	
<b>Actores</b>	Administración.	
<b>Resumen</b>	El caso de uso inicia cuando el usuario Administración inserta, modifica o elimina dichos datos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario Administración se encuentra autenticado.	
<b>Postcondiciones</b>	Se registró, modificó o eliminó los datos de una Dieta.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Gestionar datos de las Dietas &gt;</b>		
<b>Actor</b>	<b>Sistema</b>	
1.	Selecciona la opción de "Dieta".	
2.		Muestra en pantalla una lista con las opciones de Insertar datos de una Dieta, modificar datos de una Dieta, eliminar datos de una Dieta.
3.	Desea registrar, eliminar o modificar datos de una Dieta.	
4.		Se muestra en pantalla las opciones para para gestionar los datos de una Dieta. a) Para insertar datos de una Dieta ir a la sección "Insertar datos de una Dieta" b) Para modificar datos de una Dieta ir a la sección "Modificar datos de una Dieta" que se encuentra en la vista de las Dietas. c) Para eliminar datos de una Dieta ir a la sección "Eliminar datos de una Dieta" d) Para listar datos de una Dieta ir a la sección "Listar datos de una Dieta"

<b>Sección 1: “Listar datos de las Dietas”</b>		
<b>Flujo Básico: “Listar datos de las Dietas”</b>		
<b>1a. Listar Dieta</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra en pantalla el listado con las Dietas
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 2: “Insertar datos de la Dieta”</b>		
<b>Flujo básico &lt; Insertar datos de la Dieta &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
1.	Presiona la opción “ Insertar datos de la Dieta “.	
2.		Muestra un formulario con los siguientes campos a introducir: <ul style="list-style-type: none"> <li>- Fecha</li> <li>- Nombre y Apellido</li> <li>- Labor a realizar</li> <li>- Clasificación</li> <li>- Carnet de Identidad</li> <li>- Fecha de Salida</li> <li>- Fecha Regreso</li> <li>- Fecha Autorizo</li> <li>- Nombre Autorizo</li> </ul>
3.	Introduce los datos de la Dieta en los campos y presiona la opción “Insertar datos de la Dieta” o “Cancelar”.	
4.		Verifica que los campos estén llenos.
5.		Verifica que los datos introducidos sean correctos.

6.		Verifica que la Dieta no se encuentre ya insertado
7.		Guarda los datos de la Dieta y muestra un mensaje “Los datos fueron guardados con éxito”. Finalizando así el Caso de Uso.
<b>Flujos alternos</b>		
<b>2a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Flujos alternos</b>		
<b>6a. Datos de Dieta registrados.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra un mensaje “Datos guardados con éxito”.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 3: “Modificar datos de Dieta”</b>		
<b>Flujo básico &lt;Modificar datos de Dieta &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
1	Presiona la opción “Modificar datos de la Dieta”.	

2		<p>Muestra un formulario con los siguientes campos a modificar:</p> <ul style="list-style-type: none"> <li>- Fecha</li> <li>- Nombre y Apellido</li> <li>- Labor a realizar</li> <li>- Clasificación</li> <li>- Carnet de Identidad</li> <li>- Fecha de Salida</li> <li>- Fecha Regreso</li> <li>- Fecha Autorizo</li> <li>- Nombre Autorizo</li> </ul> <p>Y el botón “Editar datos de la Dieta”.</p> <p>Y el botón “Cancelar”.</p>
3	Actualiza los datos de Dieta en los campos y presiona la opción “Modificar datos de Dieta” o “Cancelar”.	
4		Verifica que los campos estén llenos.
5		Verifica que los datos introducidos sean correctos.
6		<p>Actualiza los datos de la Dieta y muestra un mensaje “Datos de la Dieta modificados con éxito”.</p> <p>Finalizando así el Caso de Uso.</p>
<b>Flujos alternos</b>		
<b>4a. Campos vacíos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo que el campo es obligatorio según el campo que lo requiera.
<b>Flujos alternos</b>		

<b>5a. Datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Muestra varios mensajes diciendo el error que tenga el campo según el campo que lo requiera.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Flujos alternos</b>		
<b>2a. Marca botón 'Cancelar'</b>		
<b>Actor</b>		<b>Sistema</b>
1.		Retornar a la vista inicio donde se encuentra el listado de las Dietas "Listado de Dieta"
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 4: "Eliminar datos de la Dieta"</b>		
<b>Flujo básico &lt;Eliminar datos de Dieta &gt;</b>		
<b>Actor</b>		<b>Sistema</b>
1	Presiona la opción "Eliminar datos de la Dieta".	
2		Muestra la lista de las Dietas
3	Selecciona la Dieta que desea eliminar y presiona la opción "Eliminar Dieta o "Cancelar".	
4		Elimina la Dieta y muestra un mensaje "Dieta eliminada con éxito". Finalizando así el Caso de Uso.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
<b>Sección 5: "Exportar en formato pdf los datos de Dieta"</b>		
<b>Flujo básico &lt; Exportar a formato pdf los datos de Dieta &gt;</b>		
	Presiona la opción "Exportar en formato pdf los datos de las Dietas".	

		Muestra la lista de las Dietas
	Selecciona la Dieta que desea exportar y presiona la opción “Exportar en formato pdf los datos de las Dietas” o “Cancelar”.	
		Exporta la Dieta en formato pdf. Finalizando así el Caso de Uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	No aplica
<b>Asuntos pendientes</b>		No aplica