MÓDULO DE TRÁMITES INTERNOS DE LA PLATAFORMA DE TRÁMITES Y SERVICIOS DE LA UNIVERSIDAD DE HOLGUÍN

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

Autor:

Daniel Diéguez Saavedra

Tutor:

Ing. Leonardo González Guerrero

# 1. Introducción

En la era digital, la transformación tecnológica se ha convertido en un pilar fundamental para las instituciones de educación superior. Las universidades modernas han adoptado diversas herramientas y plataformas digitales para optimizar sus procesos académicos, administrativos y de servicios, buscando mejorar la eficiencia, la transparencia y la experiencia del usuario. Dentro de este contexto, las plataformas de trámites universitarios emergen como elementos clave para la gestión centralizada y automatizada de solicitudes y servicios. Estas plataformas ofrecen la posibilidad de simplificar procesos complejos, reducir tiempos de respuesta, minimizar errores y proporcionar un seguimiento transparente de cada trámite, contribuyendo así a una gestión más eficiente y a una mayor satisfacción de la comunidad universitaria.

La Universidad de Holguín (UHO) no es ajena a esta realidad. Si bien la UHO cuenta con una plataforma de trámites en línea, esta carece de módulos específicos para la gestión de trámites internos como Transporte, Mantenimiento, Alimentos y Hospedaje. Esta limitación representa una oportunidad para mejorar significativamente la eficiencia y la calidad de los servicios ofrecidos a la comunidad universitaria.

El desarrollo del módulo propuesto se basa en un enfoque centrado en el usuario, tomando en consideración las experiencias y necesidades tanto del personal administrativo como de los estudiantes. Para ello, se llevarán a cabo encuestas y entrevistas que permitan identificar los principales puntos de dolor y áreas de mejora. Adicionalmente, se analizarán plataformas similares implementadas en otras universidades para extraer mejores prácticas y adaptarlas a nuestro contexto. Este proceso de desarrollo iterativo asegurará que el módulo no solo cumpla con los requisitos funcionales, sino que también sea intuitivo y fácil de usar.

Abordar esta problemática en la Universidad de Holguín es de suma importancia, ya que la integración de los módulos de trámites internos en la plataforma existente permitiría optimizar la gestión de recursos, mejorar la calidad del servicio, aumentar la satisfacción de los profesores y personal administrativo, y apoyar la toma de decisiones basada en datos. Una vez analizada la situación actual de los procesos ligados a los trámites internos de la Universidad se encontraron las siguientes **deficiencias:**

1. Existe falta de comunicación entre las diferentes áreas que intervienen en la gestión trámites internos.
2. Dificultad en el almacenamiento de los datos asociados al proceso de solicitudes de trámites internos.
3. Errores a la hora de extraer información.
4. Pérdida de documentos asociados a los trámites internos.
5. Inconsistencias y falta de claridad al solicitar algunos trámites

Las deficiencias encontradas condicionan el siguiente **problema de investigación**: ¿Cómo favorecer la gestión de solicitudes de los trámites internos en la Universidad de Holguín? El problema planteado se enmarca dentro del **objeto de estudio**: el proceso de gestión de solicitudes de trámites internos en las Universidades. Para solucionar el problema se plantea como **objetivo de la investigación:** desarrollar un módulo de gestión tramites internos a la plataforma de gestión de solicitudes de los trámites y servicios en la Universidad de Holguín. El objetivo delimita el siguiente **campo de acción**: informatización del proceso de gestión de solicitudes de los trámites internos en la Universidad de Holguín.

Para guiar la presente investigación se definen las siguientes **preguntas científicas:**

* ¿Cuáles son fundamentos teóricos sustentan el proceso de gestión de solicitudes de trámites en las universidades cubanas?
* ¿Cómo ocurre el proceso de gestión de solicitudes de los trámites internos en la Universidad de Holguín?
* ¿Cómo valorar la sostenibilidad del sistema informático de gestión de solicitudes de los trámites internos en la Universidad de Holguín?
* ¿Cómo desarrollar un módulo que permita gestionar los trámites internos en la Plataforma de Trámites y Servicios de la Universidad de Holguín?
* ¿Cómo valorar que el Módulo de Gestión de Trámites Internos satisface las necesidades que lo originaron?

Para cumplir con el objetivo trazado y responder las preguntas científicas elaboradas se plantean las siguientes **tareas de investigación:**

1. Elaborar los fundamentos teóricos que sustentan la informatización de gestión de solicitudes de trámites en las universidades cubanas.
2. Analizar el proceso de gestión de solicitudes de los trámites internos de la Universidad de Holguín.
3. Valorar la sostenibilidad del módulo en las dimensiones socio-humanista, administrativa, tecnológica y ambiental.
4. Desarrollar un módulo que agregue la gestión de solicitudes de tramites internos a la plataforma existente
5. Comprobar el comportamiento del software y el cumplimiento de las necesidades del cliente a través de pruebas de aceptación.

**Estructura del documento**

El presente documento consta de dos epígrafes los cuales se describen de la siguiente manera:

**Materiales y métodos de la investigación:** se explican los fundamentos teóricos que sustentan la investigación, los métodos científicos empleados durante la investigación, se valoran las principales tendencias, herramientas y tecnologías para la construcción de la solución propuesta, así como la metodología de desarrollo de software empleada para ello.

**Resultados y discusión:** se exponen los principales resultados obtenidos.

A continuación de estos dos epígrafes el documento también consta de conclusiones, recomendaciones, referencias bibliográficas y anexos.

# 2. Materiales y métodos de la investigación

### 2.1 Fundamentos Teóricos

Para establecer una base sólida para esta investigación, se realiza una revisión profunda de los conceptos fundamentales y de las tendencias tecnológicas más relevantes. Esto permite justificar la elección de las herramientas, lenguajes y metodología que se utilizarán en el desarrollo del módulo asegurando que sea robusto y eficiente.

**Trámite:**

Un trámite es una actuación que implica una acción o conjunto de acciones para lograr obtener un beneficio o cumplir con una obligación. Se realizan trámites de todo tipo en ámbitos de la educación, legales, financieros, o de salud entre otros (Paula Nicole, 2018).

**Servicio:**

Un servicio se refiere a una acción o conjunto de acciones que se realizan con el propósito de satisfacer las necesidades o deseos de una persona o entidad. A diferencia de los bienes tangibles, los servicios son intangibles, lo que significa que no pueden ser tocados ni poseídos físicamente (Javier Sánchez, 2016).

**Gestión:**

La gestión se refiere al conjunto de procesos y acciones que permiten dirigir recursos humanos, financieros, materiales y de información dentro de una organización con el objetivo de lograr metas definidas. Este concepto es fundamental en el ámbito empresarial, donde se busca maximizar la productividad y optimizar el uso de los recursos disponibles (Guillermo Westreicher, 2020).

**Sistema Informático:**

Un sistema informático es una combinación de hardware y software diseñada para llevar a cabo funciones específicas de procesamiento de información, almacenamiento, y comunicación. Estos sistemas son cruciales para el funcionamiento eficiente de numerosas aplicaciones en diversos campos, como el comercio, la educación y la salud (Graciela Marker, 2018).

**Sistema de Gestión:**

Un sistema de gestión es un conjunto interrelacionado de políticas, procesos, procedimientos y recursos que permiten a las organizaciones alcanzar sus objetivos de manera eficiente y organizada (Rodrigo Ricardo, 2024).

**Módulo:**

Un módulo se refiere a una parte del código que se puede desarrollar, probar y mantener de manera independiente. Los módulos están diseñados para llevar a cabo tareas específicas y suelen agruparse en el contexto de programación modular o en el desarrollo de aplicaciones web, permitiendo así la organización y la escalabilidad del software (Helmut Corvo, 2020).

### 2.2 Métodos de investigación

Para realizar correctamente la investigación y cumplir con los requisitos del módulo se utilizaron los siguientes métodos investigativos:

**Métodos teóricos:**

1. Análisis y síntesis: se utilizó en la recopilación de los fundamentos teóricos, en el procesamiento de la información, y en la descomposición de cada uno de los requerimientos del sistema, así como del mismo en pequeños subsistemas para facilitar la comprensión del problema con más detalle.
2. Modelación: se empleó durante el desarrollo del sistema informático a través del uso de la metodología scrum.

Histórico-lógico: permitió estudiar el flujo de información que interviene en el proceso, así como comprender la lógica y las normas que rigen su funcionamiento. Posibilitó expresar en forma teórica la esencia del objeto y las necesidades existentes.

Enfoque Sistémico: se basa en la idea de que un sistema no puede ser comprendido completamente si se estudia de manera aislada, por lo que permitió definir como los distintos componentes interactúan entre si definiendo las relaciones que existen entre los elementos del sistema

**Métodos empíricos:**

1. Revisión de documentos: permitió recopilar información referente a los procesos que se realizan en las Universidades, y definir las tecnologías y herramientas a usar en el desarrollo del sistema informático.
2. Entrevista no estructurada: se utilizó para obtener información referente al proceso analizado y las deficiencias existentes, así como definir las necesidades y los requerimientos funcionales y no funcionales del sistema para desarrollar un producto informático que cumpla con las expectativas del cliente.
3. Observación científica: se empleó esencialmente para el análisis, diseño e implementación del sistema informático. Facilitó que se tuviera una mejor seguridad y exactitud en las decisiones tomadas y los resultados obtenidos acerca del producto, además, lograr la información directa sobre el proceso de desarrollo y objeto que está siendo investigado.

### 2.3 Valoración de Sostenibilidad

**Administrativa:** El desarrollo del módulo de gestión de trámites internos para la Universidad Holguín requerirá un total de 250 horas-hombre. Será necesario contar con recursos como servidores, computadoras y personal especializado para su implementación. Este módulo permitirá centralizar y automatizar los procesos de solicitud y gestión de trámites, lo que resultará en una reducción considerable del tiempo de procesamiento y en una mayor eficiencia operativa, la digitalización de los trámites reducirá la carga de trabajo manual y minimizará los errores humanos. Esto se traducirá en una mayor precisión y consistencia en la gestión de los trámites, lo que a su vez mejorará la calidad del servicio ofrecido a estudiantes y personal. El módulo es sostenible a largo plazo, ya que responde a las necesidades que motivaron su desarrollo y se ajusta a los recursos utilizados.

**Socio-Humanista:** El módulo contribuye a promover el conocimiento y la educación al facilitar la gestión eficiente de trámites académicos y administrativos. Al reducir el tiempo y el esfuerzo necesarios para completar estos trámites, los estudiantes y el personal pueden dedicar más tiempo a actividades académicas y de investigación, potenciando así el aprendizaje y el desarrollo intelectual. Mejora las condiciones laborales al optimizar los procesos administrativos y facilitar la interacción entre los trabajadores y los usuarios. Esta optimización permite un flujo de trabajo más ágil y eficiente, lo que se traduce en una mejor capacidad para registrar, responder y resolver solicitudes de manera oportuna. Esto no solo mejora la satisfacción de los usuarios, sino que también reduce el estrés y la carga de trabajo del personal administrativo. Favorece la competencia profesional en la gestión de proyectos informáticos sostenibles, ya que este tipo de sistemas son escasos en el país. Las pruebas de aceptación permiten verificar de forma crucial si el servicio cumple con los requisitos específicos de los clientes o usuarios finales. El módulo es sostenible a largo plazo, ya que responde a las necesidades que motivaron su desarrollo y se ajusta a los recursos utilizados.

**Ambiental:** La digitalización de los trámites reduce la dependencia del papel, disminuyendo así la cantidad de residuos y la deforestación asociada a la producción de papel. Este cambio hacia un entorno sin papel no solo conserva los recursos naturales, sino que también reduce las emisiones de carbono relacionadas con el transporte y la eliminación de documentos físicos. El módulo contribuye a una utilización más eficiente de los recursos energéticos. La centralización y automatización de las tareas permiten una reducción en el uso de equipos y dispositivos, lo que disminuye el consumo de energía eléctrica en la universidad. Esta eficiencia energética se traduce en un menor impacto ambiental y una reducción en los costos operativos Presenta interfaces amigables que facilitan la interacción natural y sencilla del usuario con el sistema, sin importar el tipo de dispositivo utilizado. El código del módulo es claro y comprensible, lo que permite su reutilización y mantenimiento futuro. El módulo es sostenible a largo plazo, ya que cumple con las necesidades que motivaron su desarrollo y se ajusta a los recursos utilizados.

**Tecnológica:** El módulo de gestión de trámites internos incorpora tecnologías avanzadas que garantizan su eficiencia y efectividad. Utiliza bases de datos robustas para el almacenamiento seguro de información y sistemas de encriptación para proteger la privacidad de los datos. La arquitectura modular del sistema permite su escalabilidad y adaptación a futuras necesidades y actualizaciones. Se implementan interfaces de usuario intuitivas y amigables que facilitan la interacción y mejoran la experiencia del usuario. Además, el módulo está diseñado para ser compatible con múltiples plataformas y dispositivos, asegurando su accesibilidad desde cualquier lugar y en cualquier momento. La sostenibilidad tecnológica del módulo radica en su capacidad de adaptarse a los avances tecnológicos y su facilidad de mantenimiento, asegurando su relevancia y funcionalidad a largo plazo. El módulo es sostenible a largo plazo, ya que responde a las necesidades que motivaron su desarrollo y se ajusta a los recursos utilizados.

### 2.4 Tecnologías, Herramientas y Metodologías utilizadas

Para desarrollar el módulo, es fundamental poseer los conocimientos necesarios en lenguajes de programación, frameworks y otras herramientas para crear aplicaciones web. Los lenguajes de programación son la base sobre la cual se construyen todas las aplicaciones web, y los frameworks permiten a los desarrolladores acelerar el proceso de creación de estas aplicaciones. Además, herramientas como los editores de código y los sistemas de control de versiones son indispensables para el desarrollo efectivo de cualquier aplicación.

**Hypertext Markup Language (HTML):**

HTML es el lenguaje estándar para crear y diseñar páginas web. Funciona mediante el uso de "etiquetas" que describen la estructura y el contenido de una página web. Estas etiquetas se utilizan para definir elementos como párrafos, encabezados, enlaces, imágenes y otros tipos de contenido multimedia. Proporciona la base sobre la cual se construyen las páginas web y, aunque no es un lenguaje de programación en sí mismo, es fundamental para el desarrollo web.

El propósito principal de HTML es estructurar el contenido de una página web de manera que los navegadores puedan interpretarlo y mostrarlo correctamente. Además, HTML se utiliza en conjunto con CSS (Cascading Style Sheets) y JavaScript para crear páginas web interactivas y visualmente atractivas. Mientras HTML define la estructura del contenido, CSS se encarga del diseño y la presentación, y JavaScript añade interactividad y funcionalidad dinámica.

**Cascading Style Sheets (CSS):**

CSS es un lenguaje de diseño utilizado para describir la presentación de un documento escrito en HTML o XML. Permite a los desarrolladores separar la estructura del diseño, controlando la apariencia de las páginas web mediante reglas de estilo. Estas reglas definen cómo se deben mostrar los elementos HTML en una página, especificando aspectos como colores, fuentes, márgenes, alineaciones y otros aspectos visuales.

El propósito principal de CSS es mejorar la estética y la usabilidad de las páginas web, permitiendo a los desarrolladores crear diseños consistentes y atractivos que sean fáciles de mantener y actualizar. CSS también facilita la adaptación de páginas web para diferentes dispositivos y tamaños de pantalla a través de técnicas como el diseño responsivo, donde las reglas de estilo se ajustan según el tamaño de la ventana del navegador (Gustavo B, 2019).

**JavaScript (JS):**

JavaScript es un lenguaje de programación de alto nivel y dinámico que se utiliza principalmente para crear contenido interactivo en las páginas web. Funciona en el lado del cliente, lo que significa que se ejecuta directamente en el navegador del usuario, permitiendo la creación de funcionalidades dinámicas sin necesidad de recargar la página. JavaScript se puede utilizar para una amplia variedad de tareas, como la manipulación del DOM (Document Object Model), la validación de formularios, la creación de animaciones y la gestión de eventos del usuario, como clics y movimientos del ratón.

El propósito de JavaScript es mejorar la experiencia del usuario al permitir una interacción más fluida y dinámica con las páginas web. Con la ayuda de bibliotecas y frameworks populares como jQuery, React, Angular y Vue.js, los desarrolladores pueden crear aplicaciones web avanzadas y responsivas. Además, JavaScript se ha expandido más allá del navegador con la introducción de Node.js, lo que permite a los desarrolladores utilizarlo también en el lado del servidor.

**Python:**

Python es un lenguaje de programación de alto nivel y de propósito general, conocido por su sintaxis clara y legible, lo que facilita su aprendizaje y uso. Funciona mediante la interpretación de código fuente en lugar de su compilación, lo que permite a los desarrolladores escribir y ejecutar código rápidamente. Python es compatible con múltiples paradigmas de programación, incluidos el orientado a objetos, el imperativo, el funcional y el procedimental. Una característica destacada de Python es su extensa biblioteca estándar, que proporciona módulos y funciones para realizar una amplia variedad de tareas, desde el procesamiento de cadenas y archivos hasta la manipulación de datos y redes.

Python se utiliza en una amplia gama de aplicaciones, como desarrollo web, análisis de datos, inteligencia artificial, automatización de procesos y desarrollo de software. Gracias a su versatilidad y a la gran cantidad de bibliotecas y frameworks disponibles, como Django para desarrollo web, NumPy y pandas para análisis de datos, y TensorFlow y PyTorch para aprendizaje automático, Python se ha convertido en uno de los lenguajes de programación más populares y demandados en la industria tecnológica. Su facilidad de uso y su capacidad para manejar tareas complejas lo hacen ideal tanto para principiantes como para desarrolladores experimentados (Leo Byte, 2024).

**Django:**

Django es un framework de alto nivel para el desarrollo de aplicaciones web, escrito en Python. Su principal objetivo es facilitar la creación de sitios web complejos y escalables de manera rápida y con un código limpio y eficiente. Django sigue el patrón de diseño Modelo-Vista-Controlador (MVC), aunque en su terminología se conoce como Modelo-Vista-Plantilla (MVT). Este framework proporciona una serie de componentes integrados, como un sistema de autenticación, un ORM (Object-Relational Mapping) para la gestión de bases de datos, formularios, enrutamiento URL y administración automática del sitio, lo que permite a los desarrolladores centrarse en la lógica de negocio en lugar de en la infraestructura subyacente.

Django es ampliamente utilizado en la industria debido a su enfoque en la reutilización de código, la seguridad y la escalabilidad. Es conocido por su "baterías incluidas", lo que significa que viene con una gran cantidad de funcionalidades listas para usar, lo que reduce la necesidad de instalar dependencias adicionales. Además, Django promueve las prácticas de desarrollo seguro, como la protección contra inyecciones SQL, la gestión adecuada de contraseñas y la prevención de ataques XSS (Cross-Site Scripting). Esto hace que Django sea una opción popular para desarrollar aplicaciones web robustas y seguras en diversos sectores, como el comercio electrónico, las redes sociales y los sistemas de gestión de contenidos.

**React:**

React es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.

Características:

JSX es una extensión de la sintaxis de JavaScript utilizada en la creación de elementos de React. Los desarrolladores la emplean para incrustar código HTML en objetos JavaScript. Ya que JSX acepta expresiones válidas de JavaScript e incrustación de funciones, puede simplificar las estructuras de código compleja

Virtual DOM: El Modelo de Objetos del Documento (DOM) presenta una página web en una estructura de árbol de datos. ReactJS almacena árboles DOM virtuales en la memoria. Al hacerlo, React puede aplicar actualizaciones a partes específicas del árbol de datos, lo que es más rápido que volver a renderizar la totalidad del árbol DOM.

Componentes y Props: ReactJS divide la interfaz de usuario en piezas aisladas y reutilizables de código conocidas como componentes. Los componentes de React funcionan de forma similar a las funciones de JavaScript, ya que aceptan entradas arbitrarias denominadas propiedades o props.

Gestión de Estado: Un estado es un objeto JavaScript que representa una parte de un componente. Cambia cada vez que un usuario interactúa con la aplicación, renderizando una nueva interfaz de cliente para reflejar las modificaciones.

La gestión de estados se refiere a la práctica de gestionar los estados de la aplicación React. Incluye el almacenamiento de datos en librerías de gestión de estados de terceros y la activación del proceso de re-renderización cada vez que los datos cambian (Stephan Miller, 2021).

**PostgreSQL:**

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez, escalabilidad y compatibilidad con los estándares SQL. Funciona como un motor de bases de datos que permite a los usuarios almacenar, consultar y manipular datos estructurados. PostgreSQL es compatible con una amplia gama de sistemas operativos, incluidos Linux, Windows y macOS, y se destaca por su capacidad para manejar grandes volúmenes de datos y realizar consultas complejas de manera eficiente. Una característica clave de PostgreSQL es su capacidad para soportar tipos de datos avanzados, índices personalizados y funciones definidas por el usuario, lo que lo convierte en una opción flexible y poderosa para una variedad de aplicaciones.

El propósito principal de PostgreSQL es proporcionar una plataforma confiable y segura para la gestión de datos en aplicaciones empresariales, científicas y web. Se utiliza en una amplia gama de industrias y aplicaciones, desde servicios financieros y comercio electrónico hasta análisis de datos y sistemas de información geográfica (GIS). Gracias a su arquitectura extensible y su enfoque en la conformidad con los estándares, PostgreSQL permite a los desarrolladores crear aplicaciones robustas y escalables que pueden crecer y adaptarse a medida que evolucionan las necesidades del negocio (Pete Scott, 2025).

**Git:**

Git es un sistema de control de versiones distribuido que permite a los desarrolladores rastrear y gestionar los cambios en el código fuente a lo largo del tiempo. Funciona mediante la creación de instantáneas (snapshots) del estado del código en momentos específicos, lo que permite a los usuarios revertir, comparar y fusionar diferentes versiones del código de manera eficiente. Git almacena los datos en un repositorio, y cada copia del repositorio es completa e independiente, lo que significa que los desarrolladores pueden trabajar de manera autónoma y sincronizar sus cambios con otros colaboradores cuando sea necesario.

El propósito principal de Git es facilitar la colaboración en proyectos de desarrollo de software, permitiendo a varios desarrolladores trabajar simultáneamente en el mismo proyecto sin conflictos. Git permite la creación de ramas (branches) para desarrollar características nuevas o realizar correcciones de errores sin afectar el código principal. Una vez que las modificaciones se han probado y están listas, se pueden fusionar (merge) de nuevo en la rama principal. Git también proporciona herramientas poderosas para revisar el historial de cambios, identificar quién realizó modificaciones específicas y resolver conflictos de manera eficiente.

**Visual Studio Code:**

Visual Studio Code, comúnmente conocido como VS Code, es un editor de código fuente desarrollado por Microsoft. Es un editor ligero pero poderoso que soporta una amplia gama de lenguajes de programación y está diseñado para ser extensible y personalizable. VS Code funciona en múltiples plataformas, incluyendo Windows, macOS y Linux, y se caracteriza por su velocidad y rendimiento. Su interfaz de usuario es intuitiva, con características como el resaltado de sintaxis, la autocompletación de código y la navegación rápida por archivos, lo que facilita la escritura y edición de código de manera eficiente.

VS Code se utiliza principalmente para el desarrollo de software, permitiendo a los desarrolladores escribir, depurar y desplegar código con facilidad. Una de sus principales ventajas es la integración de un terminal incorporado y un depurador, así como el soporte para Git, lo que facilita la gestión del código fuente y la colaboración en proyectos. Además, cuenta con un amplio ecosistema de extensiones que permiten a los desarrolladores añadir funcionalidades adicionales según sus necesidades, desde soporte para lenguajes específicos hasta herramientas de productividad y personalización del entorno de desarrollo (TecnoDigital, 2025).

**Scrum:**

Scrum es un marco de trabajo ágil utilizado en la gestión y desarrollo de proyectos, particularmente en el ámbito del desarrollo de software. Funciona mediante la división de un proyecto en iteraciones cortas y manejables llamadas "sprints," que generalmente duran entre una y cuatro semanas. Durante cada sprint, un equipo multifuncional trabaja en conjunto para completar una lista de tareas priorizadas, conocidas como el "backlog" del producto. Al final de cada sprint, se lleva a cabo una revisión y retrospectiva para evaluar el progreso, identificar áreas de mejora y planificar el próximo sprint. Este enfoque iterativo permite al equipo adaptarse rápidamente a los cambios y mejoras incrementales en el producto.

El propósito principal de Scrum es mejorar la productividad, la calidad del producto y la satisfacción del cliente mediante la entrega continua de incrementos funcionales del producto. Scrum promueve la colaboración, la transparencia y la adaptabilidad, lo que permite a los equipos responder eficazmente a los cambios en los requisitos y las prioridades del proyecto. Los roles clave en Scrum incluyen el Product Owner, responsable de gestionar el backlog y definir las prioridades; el Scrum Master, encargado de facilitar el proceso y eliminar obstáculos; y el equipo de desarrollo, que es responsable de entregar el incremento del producto al final de cada sprint. En resumen, Scrum es una metodología ágil que fomenta la colaboración y la mejora continua, ayudando a los equipos a entregar productos de alta calidad de manera eficiente y efectiva.

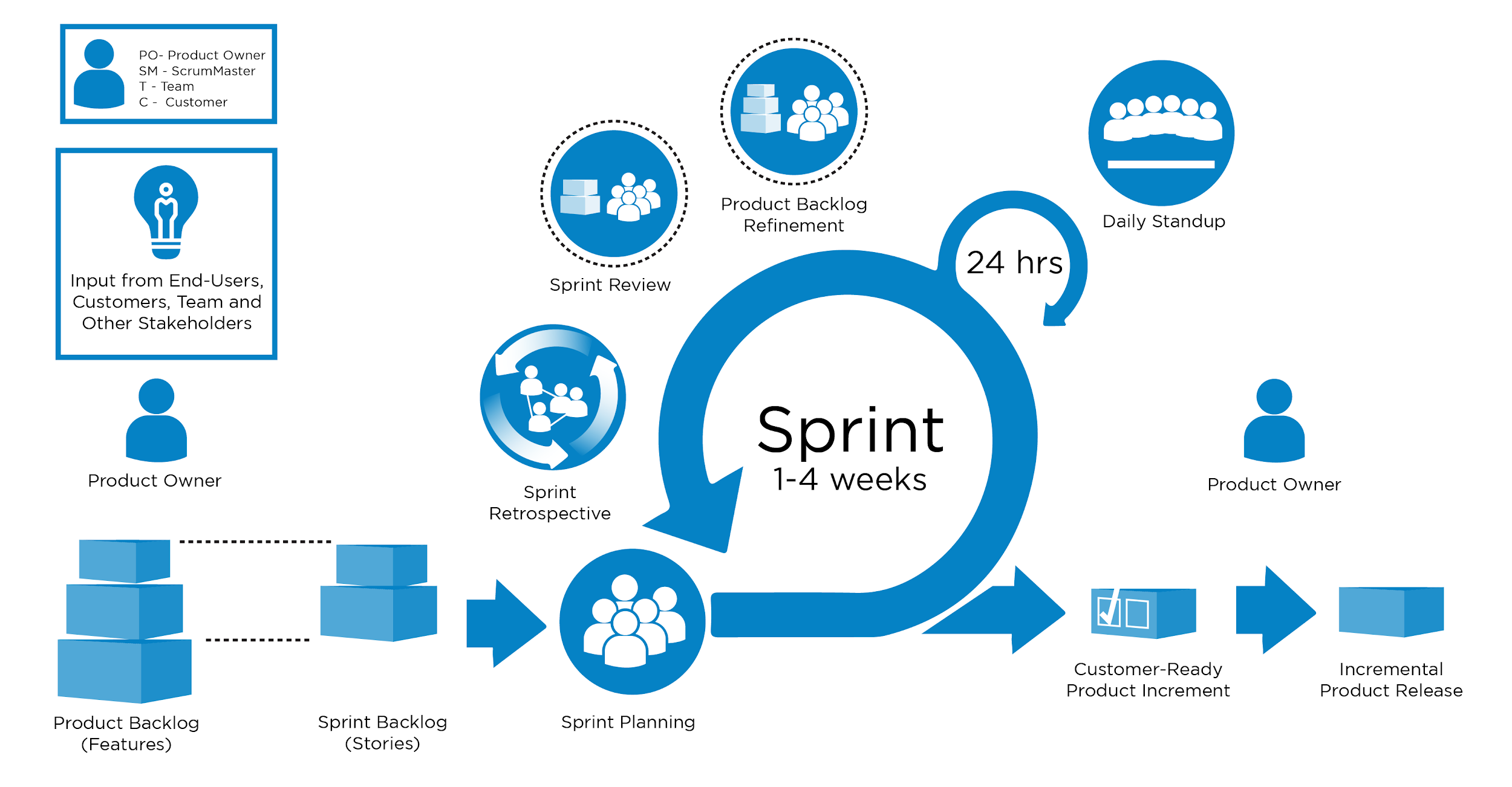


Figura 1: Scrum Process

**Draw.io**

Draw.io es una herramienta de diagramación en línea que permite a los usuarios crear una amplia variedad de diagramas, desde mapas conceptuales y diagramas de flujo hasta diagramas de redes, organigramas y modelos de procesos empresariales. El software está basado en la web, lo que significa que no requiere instalación y es accesible desde cualquier navegador moderno. Además, también ofrece versiones de escritorio para aquellos que prefieren trabajar fuera de línea. Lanzado en 2013 por JGraph Ltd., Draw.io ha evolucionado constantemente, añadiendo nuevas funcionalidades y mejorando su interfaz. Es conocido por ser una opción abierta y gratuita que no compromete en calidad, lo que le permite competir con herramientas de pago como Microsoft Visio o Lucidchart (Amalia Gómez, 2024).

# 3. Resultados y discusión

## 3.1 Requisitos funcionales

Los requerimientos funcionales son declaraciones que describen las características y funciones específicas que un sistema debe tener para satisfacer las necesidades de los interesados y del negocio. Estos requisitos se enfocan en lo que el sistema debe hacer, en lugar de en su calidad o comportamiento.

|  |  |  |
| --- | --- | --- |
| No | Requisito Funcional | Prioridad |
| RF-1 | Gestionar Trámites de Transporte como Usuario | Alta |
| RF-1.1 | Añadir Trámite | Alta |
| RF-1.2 | Editar Trámite | Alta |
| RF-1.3 | Visualizar Trámite | Alta |
| RF-2 | Gestionar Trámites de Transporte como Administrador | Alta |
| RF-2.1 | Visualizar Trámite | Alta |
| RF-2.2 | Eliminar Trámite | Alta |
| RF-2.3 | Validar Trámite | Alta |
| RF-3 | Imprimir Trámite de Transporte | Alta |
| RF-4 | Gestionar Trámites de Alimentación como Usuario | Alta |
| RF-4.1 | Añadir Trámite | Alta |
| RF-4.2 | Editar Trámite | Alta |
| RF-4.3 | Visualizar Trámite | Alta |
| RF-5 | Gestionar Trámites de Alimentación como Administrador | Alta |
| RF-5.1 | Visualizar Trámite | Alta |
| RF-5.2 | Eliminar Trámite | Alta |
| RF-5.3 | Validar Trámite | Alta |
| RF-6 | Imprimir Trámite de Alimentación | Alta |
| RF-7 | Gestionar Trámites de Hospedaje como Usuario | Alta |
| RF-7.1 | Añadir Trámite | Alta |
| RF-7.2 | Editar Trámite | Alta |
| RF-7.3 | Visualizar Trámite | Alta |
| RF-8 | Gestionar Trámites de Hospedaje como Administrador | Alta |
| RF-8.1 | Visualizar Trámite | Alta |
| RF-8.2 | Eliminar Trámite | Alta |
| RF-8.3 | Validar Trámite | Alta |
| RF-9 | Imprimir Trámite de Hospedaje | Alta |
| RF-10 | Gestionar Trámites de Mantenimiento como Usuario | Alta |
| RF-10.1 | Añadir Trámite | Alta |
| RF-10.2 | Editar Trámite | Alta |
| RF-10.3 | Visualizar Trámite | Alta |
| RF-11 | Gestionar Trámites de Mantenimiento como Administrador | Alta |
| RF-11.1 | Visualizar Trámite | Alta |
| RF-11.2 | Eliminar Trámite | Alta |
| RF-11.3 | Validar Trámite | Alta |
| RF-12 | Imprimir Trámite de Mantenimiento | Alta |
| RF-13 | Añadir Notas al Validar Trámite | Media |
| RF-14 | Visualizar Gráficas de Estados de los Trámites Internos | Media |
| RF-15 | Visualizar Gráficas de Estados de cada tipo de Trámite | Media |

## 

### 3.2 Modelo de Dominio

Cada trámite hereda de la clase abstracta *Procedure* para obtener los campos que comparten. Los modelos *FeedingProcedure* y *AccommodationProcedure* mantienen una relación ManyToMany con *FeedingDays* para permitir modificar la cantidad de personas que va a requerir el servicio cada día, igualmente para *Guest* que almacena la información de los invitados. *TransportProcedure* y *MaintanceProcedure* tienen relación ManyToOne con las clases *TransportProcedureType, MaintanceProcedureType y MaintancePriority*respectivamente.

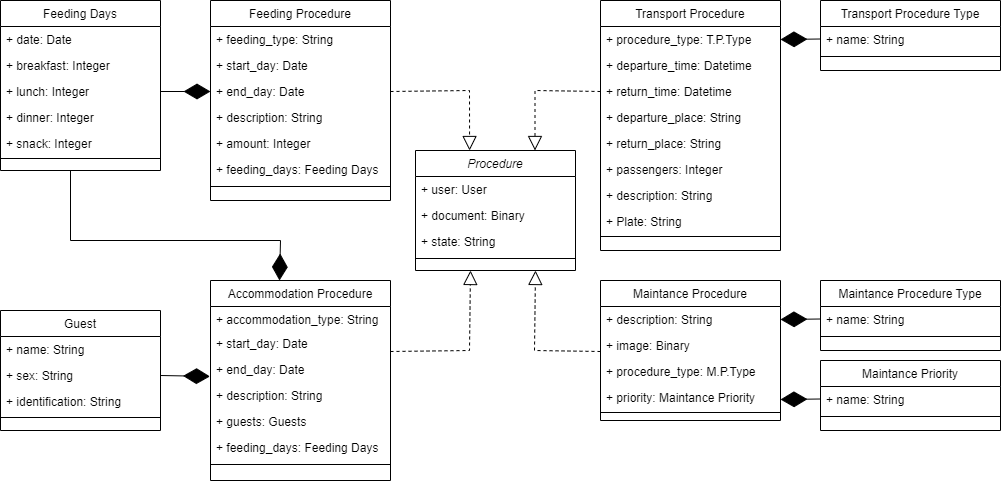


Figura 2: Diagrama de Clases

### 3.3 Arquitectura

La arquitectura Cliente-Servidor es un modelo de diseño en el que dos partes principales interactúan: el cliente, que solicita servicios, y el servidor, que responde a esas solicitudes. Es ampliamente utilizada en el desarrollo de software y en aplicaciones web modernas. Además, se utiliza una base de datos para gestionar y organizar los datos de manera eficiente.

Esta arquitectura se basa en el principio de la separación de responsabilidades, lo que permite una mayor modularidad y escalabilidad en el desarrollo de aplicaciones empresariales. Al dividir las tareas entre el cliente y el servidor, se logra un mejor rendimiento y una mayor capacidad de respuesta, lo que resulta en una experiencia de usuario mejorada y un aumento de la productividad en el entorno empresarial.

Las características principales son:

* Centralización de recursos
* Escalabilidad
* Mantenimiento sencillo
* Seguridad Mejorada
* Mientras el cliente espera la respuesta del servidor es conocido como pasivo o esclavo
* Aceptan conexiones de un gran número de clientes, aunque este se puede limitar.

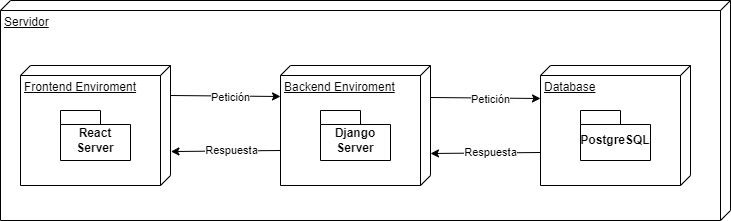


Figura 3: Arquitectura Cliente-Servidor

### 3.4Modelación de Despliegue

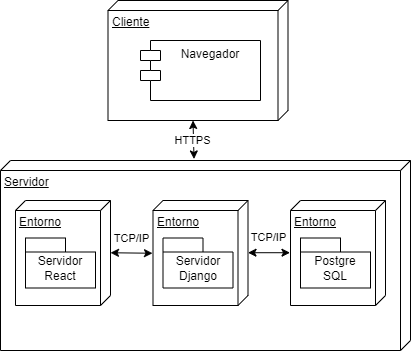


Figura 4: Diagrama de Despliegue

En primer lugar, el **Usuario** accede a la aplicación mediante un **Web Browser**, que actúa como la interfaz de interacción con el sistema. El navegador web se comunica con el **React Server**, el cual gestiona la lógica y el renderizado de la interfaz de usuario, enviando y recibiendo datos mediante el protocolo HTTPS. Luego, el **React Server** se conecta con el **Django Server**, responsable de manejar la lógica del backend y servir los datos de la aplicación. Este servidor Django, a su vez, accede a una base de datos **PostgreSQL** para almacenar y recuperar información de manera estructurada a través de TCP/IP. Esta arquitectura es común en aplicaciones web modernas, permitiendo una separación clara entre el frontend, backend y la capa de almacenamiento, lo que facilita la escalabilidad y el mantenimiento del sistema.

### 3.5Herramienta propuesta

**Características del sistema:**

Este módulo surge para abordar la necesidad de gestionar los trámites internos de la Universidad de Holguín. Su diseño basado en la web la hace accesible desde cualquier dispositivo, y su objetivo principal es favorecer la gestión de los procesos que se realizan en la Universidad de Holguín.

**Descripción del sistema:**

El módulo para gestionar trámites internos en la Universidad de Holguín cuenta con 3 roles de usuarios:

* El usuario Autenticado (podrá visualizar las interfaces Crear Trámite Interno (donde podrá crear cualquiera de los 4 tipos de trámites) y Mis Trámites (donde se encuentran todos los trámites realizados por el usuario)
* El usuario Administrador (podrá ver todo lo que realiza el usuario autenticado y tiene su Sitio Administrativo el cual puede ver todos los usuarios, sus trámites y las gráficas de los trámites realizados)
* El usuario Administrador de Trámites (tendrá su Sitio Administrativo el cual podrá ver todos los usuarios asociados a los trámites y las distintas tablas y gráficas referente a dichos trámites, además podrá crear, editar o eliminar trámites)

La interfaz de Formulario de Trámites muestra un formulario que permite crear trámites de cada tipo

La interfaz de Administrador de Trámites permite ver cada trámite por tipo y sus gráficas asociadas.

La interfaz de Administrador permite ver todos los trámites y sus gráficas asociadas.

La interfaz Mis Trámites permite ver los trámites en proceso de cada usuario.

3.6 Principales interfaces que componen la solución informática



Figura 5: Formulario de Trámite de Alimentación

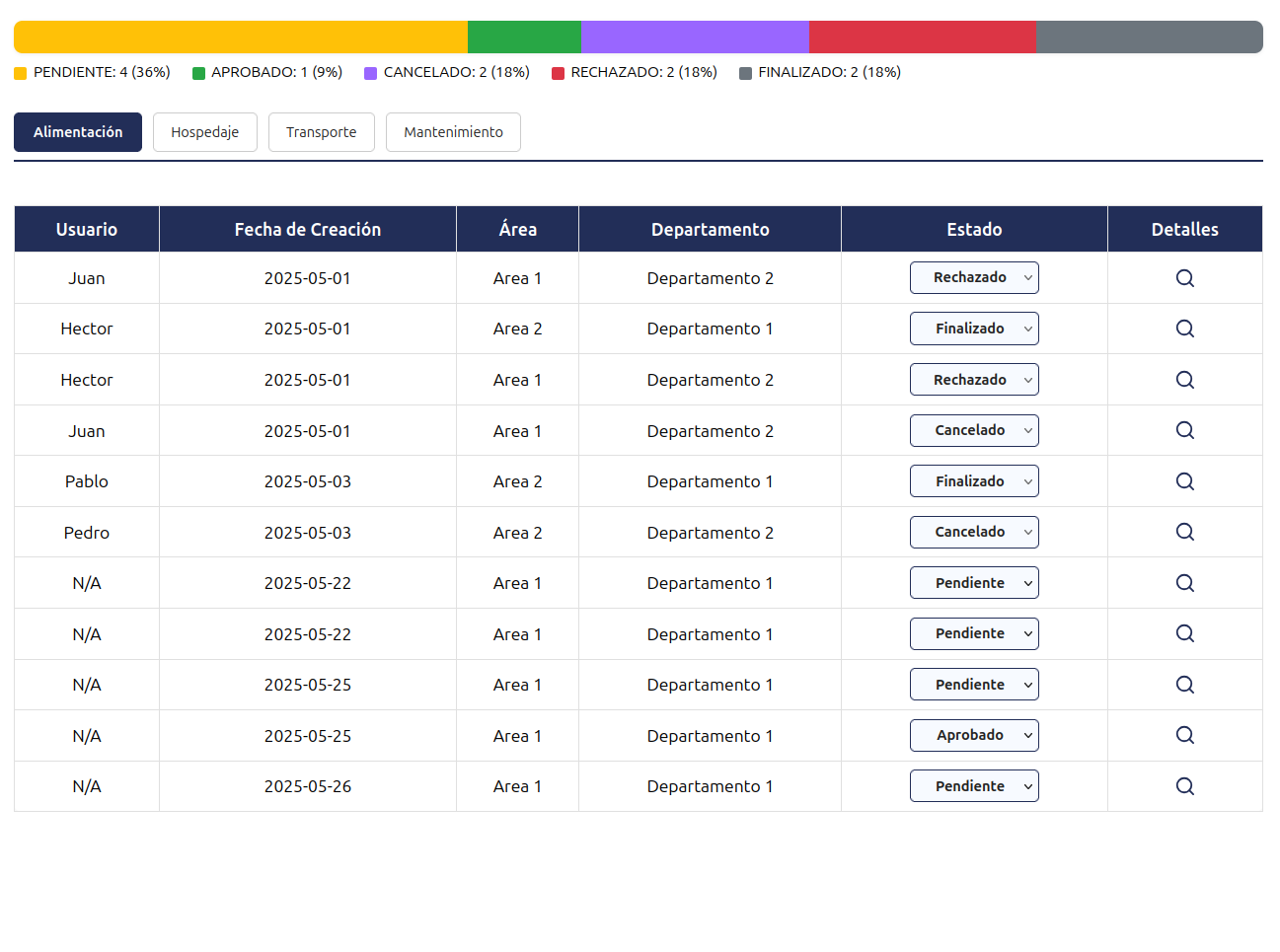


Figura 8: Administración de Trámites

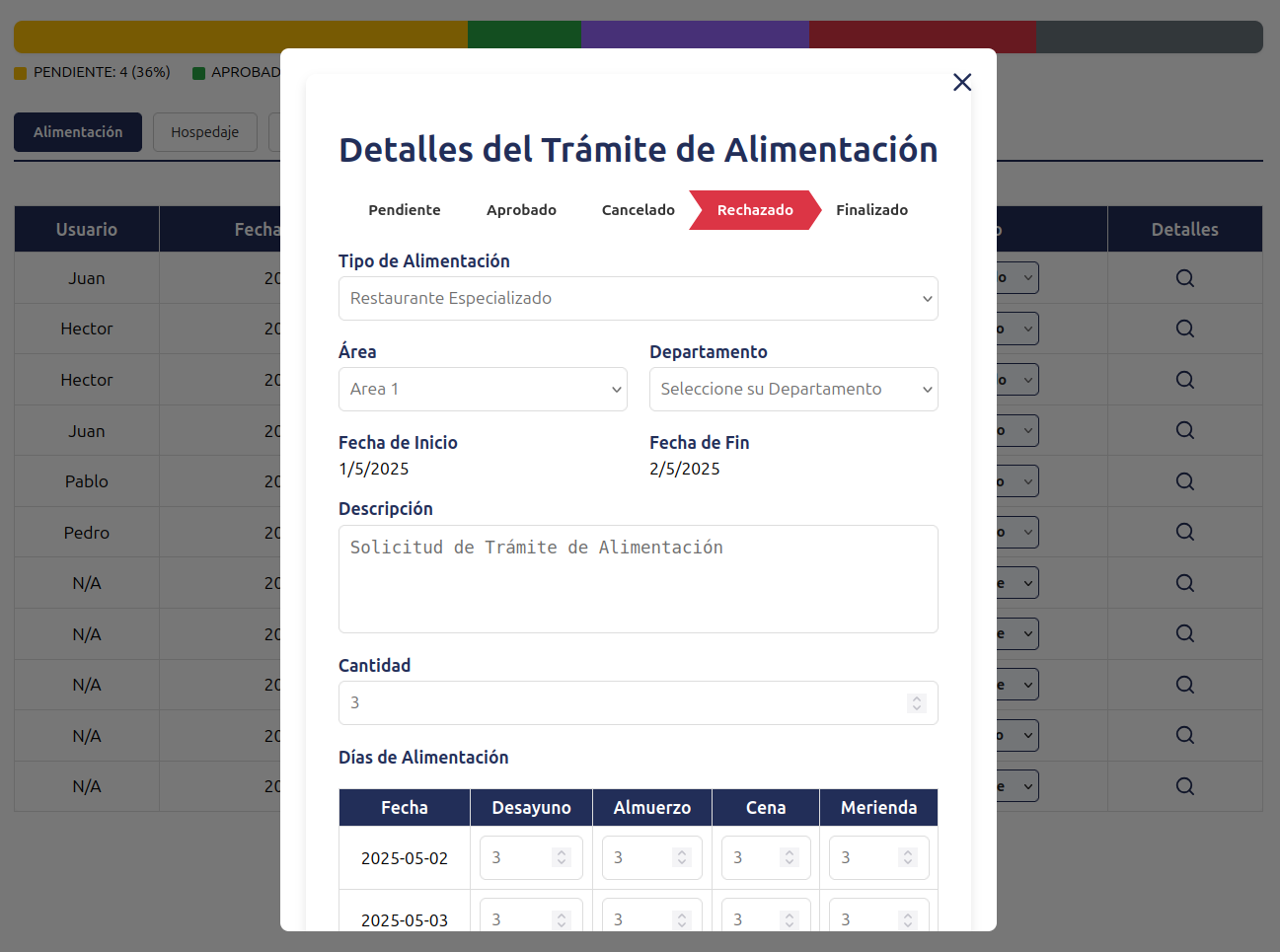


Figura 9: Administrar Trámite de Alimentación

# Conclusiones

El estudio realizado y los resultados obtenidos en la presente investigación permiten concluir que:

1. El estudio de los fundamentos teóricos que sustentan la investigación brindó los conocimientos necesarios para desarrollar el módulo de gestión de trámites internos.
2. El análisis de los procesos que intervienen en la gestión de solicitudes de trámites internos permitió informatizar correctamente dichos trámites.
3. Se valoró la sostenibilidad del sistema propuesto considerando las dimensiones socio-humanista, administrativa, tecnológica y ambiental, demostrando que el sistema tendrá un impacto positivo en la sociedad y será sostenible en el tiempo.
4. Se cumplió con el objetivo planteado en la presente investigación al desarrollar un módulo para la plataforma de trámites y servicios de la universidad de Holguín capaz de gestionar de forma eficiente los trámites internos de la Universidad.
5. Mediante pruebas de aceptación se comprobó que el módulo de gestión de trámites internos cumple con todos los requisitos iniciales y satisface las necesidades que lo originaron.

# Recomendaciones

* Agregar otros trámites internos que se empleen dentro de la Universidad de Holguín
* Incitar al personal universitario a realizar los trámites internos a través de la plataforma en lugar de hacerlos personalmente
* Garantizar la correcta capacitación del personal encargado de gestionar cada tipo de trámite dentro del módulo de gestión de tramites internos

# Referencias Bibliográficas

Roldán, Paula Nicole. «Trámite». Economipedia, 4 de noviembre de 2018, <https://economipedia.com/definiciones/tramite.html>.

Galán, Javier Sánchez. «¿Que es un servicio? Para qué sirve, características y ejemplos». Economipedia, 13 de febrero de 2016, <https://economipedia.com/definiciones/servicio.html>.

Westreicher, Guillermo. «¿Qué es la gestión? Para qué sirve, pasos a seguir y tipos». Economipedia, 7 de agosto de 2020, https://economipedia.com/definiciones/gestion.html.

Marker, Graciela. «Qué es un sistema informático: definición y concepto | T+I». Tecnología + Informática, 1 de diciembre de 2018, <https://www.tecnologia-informatica.com/que-es-sistema-informatico/>.

¿Qué es un Sistema de Gestión? Definición, tipos y ejemplos». Estudyando, 20 de febrero de 2024, <https://estudyando.com/sistemas-de-gestion-definicion-tipos-y-ejemplos/>.

«Programación modular: características, ejemplos, ventajas, aplicaciones». Lifeder, 8 de marzo de 2020, <https://www.lifeder.com/programacion-modular/>.

«HTML - Concepto, historia, cómo funciona y etiquetas». https://concepto.de/, https://concepto.de/html/. Accedido 5 de mayo de 2025.

B, Gustavo. «¿Qué Es CSS?» ES Tutoriales, 24 de enero de 2019, https://www.hostinger.com/es/tutoriales/que-es-css.

¿Qué Es JavaScript? - Aprende Desarrollo Web | MDN. 31 de marzo de 2025, <https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/What_is_JavaScript>.

Byte, Leo. «¿Qué es Python?: Una Introducción al Lenguaje». CodigoNautas, 12 de mayo de 2024, <https://codigonautas.com/que-es-python/>.

«What Is Django? (Definition, When to Use, How It Works)». Built In, https://builtin.com/software-engineering-perspectives/django. Accedido 5 de mayo de 2025.

Introduction to Django. https://www.w3schools.com/django/django\_intro.php. Accedido 5 de mayo de 2025.

«Django». Django Project, https://www.djangoproject.com/. Accedido 5 de mayo de 2025.

What Is React. https://www.w3schools.com/whatis/whatis\_react.asp. Accedido 5 de mayo de 2025.

Miller, Stephan. «What Is React?» Codecademy Blog, 13 de septiembre de 2021, <https://www.codecademy.com/resources/blog/what-is-react/>.

PostgreSQL: About. https://www.postgresql.org/about/. Accedido 5 de mayo de 2025.

Scott, Pete. «What Is PostgreSQL? Everything You Need to Know». Percona Database Performance Blog, 2 de febrero de 2025, <https://www.percona.com/blog/what-is-postgresql-used-for/>.

Git - What is Git? https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F. Accedido 5 de mayo de 2025.

«Git Guides». GitHub, 2025, <https://github.com/git-guides>.

¿Qué Es Scrum? | Scrum.Org. https://www.scrum.org/resources/blog/que-es-scrum. Accedido 5 de mayo de 2025.

TecnoDigital. «Qué es Visual Studio Code y para qué sirve: guía completa 2025». Informática y Tecnología Digital, 25 de abril de 2025, https://informatecdigital.com/que-es-visual-studio-code/.

Manager, Amalia Gómez-Senior Product. «¿Qué es Draw.io? Definición, características, funciones». amaliagomez.es, 7 de septiembre de 2024, <https://amaliagomez.es/draw-io-el-software-de-diagramas/>.

# Anexos

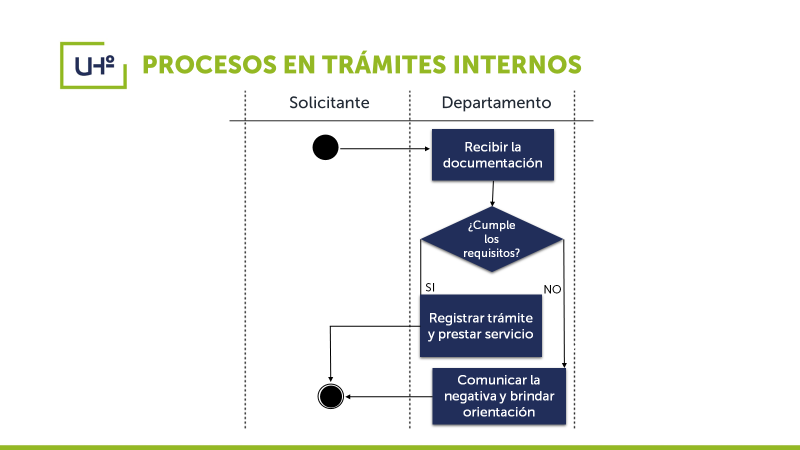


Figura 10: Procesos en los Trámites Internos



Figura 11: Formulario de Trámite de Hospedaje



Figura 12: Formulario de Trámite de Mantenimiento

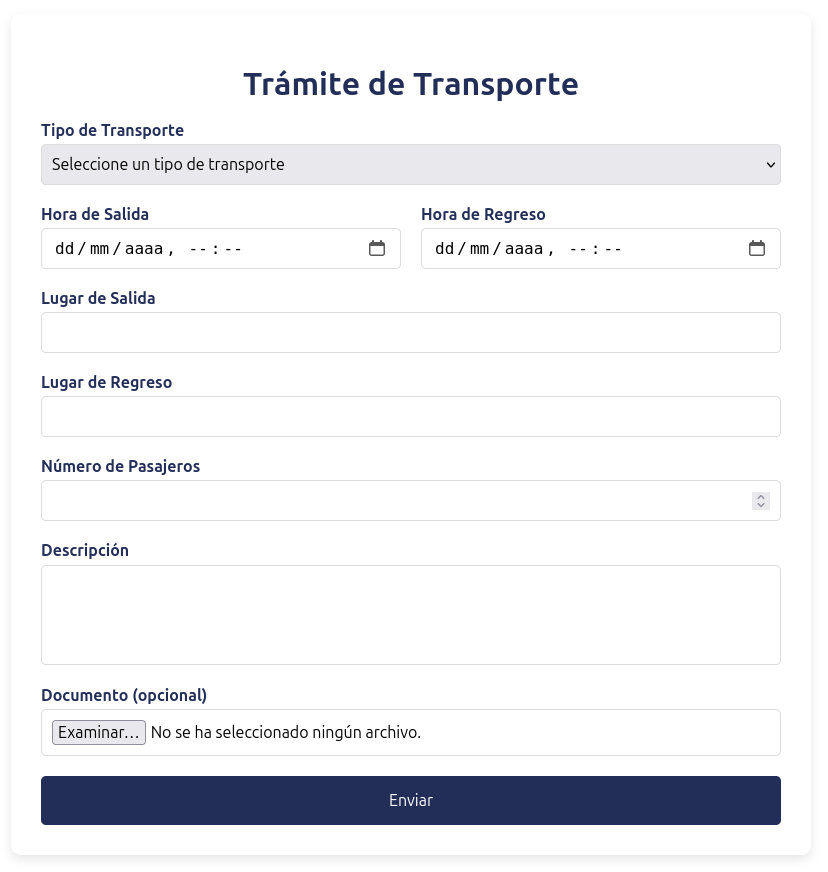


Figura 13: Formulario de Trámite de Transporte

NO