



UNIVERSIDAD  
**POLITÉCNICA**  
DE QUINTANA ROO

Formando Triunfadores

**Universidad Politécnica de Quintana Roo**

# **Desarrollo de Aplicación Web para Soporte Informático**

## **MEMORIA**

**Que para obtener el Título de:  
Ingeniería en Software**

**P R E S E N T A:**

**Rosario Novelo Joahan Emmanuel**

**202100173**

**Generación: 2021-2024**

**Asesor académico: Manuel Alejandro Flores Barrera**

**Empresa: Grupo DG**

**Asesor empresarial: Rubén Israel Rodríguez Villarreal**

**Cancún; Quintana Roo**

**Diciembre de 2024**

# ÍNDICE

DATOS GENERALES DE LA EMPRESA .....	v
ANTECEDENTES HISTÓRICOS DE LA EMPRESA.....	vi
DESCRIPCIÓN DE LA EMPRESA.....	vii
I. INTRODUCCIÓN.....	11
II. PLANTEAMIENTO DEL PROYECTO.....	12
Objetivo general.....	12
Objetivos específicos .....	12
Justificación .....	13
Viabilidad del proyecto.....	13
Impacto.....	14
III. MARCO TEÓRICO.....	15
Metodología SCRUM .....	15
Frameworks .....	15
API RESTful .....	16
Node.js .....	16
PostgreSQL .....	17
Gestión de Incidencias.....	17
Seguridad de la información .....	18
Tailwind CSS .....	19
IV. DESARROLLO DE ACTIVIDADES DEL PROYECTO.....	20
Alcance del proyecto.....	20
Metodología.....	20
Planificación del proyecto .....	21

Análisis y Documentación Inicial del proyecto.....	21
Casos de uso.....	22
Funciones Básicas de Usuario .....	22
Funciones avanzadas de Administración .....	27
Diseño de la Arquitectura del Sistema .....	43
Desarrollo de la normalización de Base de datos .....	43
Modelo Entidad-Relación.....	44
Diagrama Tercera Forma Normal (Base de datos normalizada).....	45
Diseño de Maquetados y Corrección y Refinamiento de Maquetados .....	46
Maquetados: Vista Usuario .....	47
Login .....	47
Mis tickets .....	48
Notificaciones.....	48
Creación de ticket.....	49
Maquetados: Vista Administrador .....	51
Dashboard Gestión de Tickets.....	51
Creación de nuevo usuario.....	52
V.    RESULTADOS O PRODUCTOS OBTENIDOS.....	53
VI.   CONCLUSIONES.....	54
FUENTES DE CONSULTA .....	55
ANEXOS.....	56
GLOSARIO .....	57



## **DATOS GENERALES DE LA EMPRESA**

### **Nombre de la empresa**

COMERCIALIZADORA INDUSTRIAL DG S. DE R.L. DE C.V.

### **Dirección**

Av. Sayil Supermanzana 7 entre Av. Tulum y Av. Bonampak, 77500 Cancún, Q.R.

### **Sector y giro empresarial**

Sector: Automotriz

Giro: Distribución y mantenimiento de vehículos comerciales

### **Servicios que ofrecen**

- Venta de vehículos comerciales: Distribuyen camiones y autobuses de marcas como Mercedes-Benz y Freightliner.
- Mantenimiento y postventa: Incluye revisiones periódicas, reparaciones y mantenimiento preventivo para vehículos comerciales.
- Venta de refacciones: Cuentan con repuestos originales para asegurar la compatibilidad y calidad en sus servicios.
- Servicio de rescate: Asistencia en carretera para vehículos pesados.

### **Nombre del Proyecto**

Desarrollo de Aplicación Web para Soporte Informático

### **Asesor de la empresa**

Rubén Israel Rodríguez Villarreal

### **Cargo del Asesor Empresarial**

Encargado de reclutamiento y selección

## **ANTECEDENTES HISTÓRICOS DE LA EMPRESA**

Desde principios de la década de los 70's GrupoDG se incursionó en el giro Automotriz, siendo el sector de mayor peso en la distribución de Automóviles y Camiones.

En 1991 Se obtiene la concesión de Mercedes Benz México S.A. de C.V. para Camiones Mercedes Benz y Freightliner en los Estados de Tabasco y Quintana Roo, constituyendo para tal efecto la empresa Autotab, S.A. de C.V.

En el marzo del año 2000 Autotab se expande en el sureste, abarcando el estado de Yucatán, inaugurando nuevas y modernas instalaciones en una superficie construida de 3,800 m<sup>2</sup> en tres niveles.

## **DESCRIPCIÓN DE LA EMPRESA**

### **Misión**

Satisfacer las demandas, necesidades, expectativas de nuestros clientes a través de productos y servicios de alta calidad, otorgándolos de manera oportuna, rápida y confiable. Garantizando la permanencia y superioridad competitiva en los segmentos donde participamos siendo líderes en el uso de tecnología, innovación y procesos.

### **Visión**

- Ser uno de los grupos empresariales más relevantes en el sureste de México
- Participar en segmentos de mercado rentables, con potencial de crecimiento y donde contemos con ventaja competitiva
- Contar con colaboradores capacitados y comprometidos con el desarrollo y valores del Grupo
- Estar a la vanguardia con el uso y desarrollo de tecnologías que nos permitan realizar nuestra labor con mayor eficacia y productividad
- Ser líderes en innovación

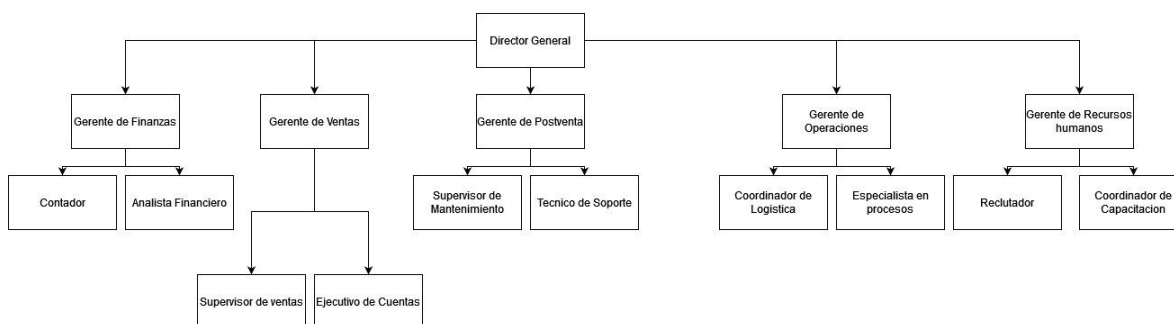
### **Políticas**

- Calidad: Compromiso con productos y servicios de alto estándar.
- Responsabilidad: Cumplimiento ético y profesional con clientes.
- Honestidad: Transparencia en todas las operaciones comerciales.
- Compromiso con el cliente: Satisfacción integral y atención personalizada.
- Innovación: Mejora continua en productos y procesos.

### **Objetivos**

- Consolidar el liderazgo en distribución y mantenimiento de vehículos comerciales en el sureste de México.
- Garantizar la satisfacción del cliente a través de servicios de alta calidad y atención personalizada.
- Innovar en soluciones de transporte y postventa para maximizar la eficiencia operativa y tecnológica.

## Estructura organizacional



*Ilustración 1. Organigrama: Empresa*



## **DESCRIPCIÓN DEL ÁREA DONDE SE REALIZARON LAS PRÁCTICAS PROFESIONALES**

### **Nombre del área**

Área Sistemas

### **Objetivos**

- Optimizar la Infraestructura Tecnológica. Mejorar y mantener la infraestructura tecnológica de la organización, asegurando que los servidores, redes y dispositivos sean eficientes, seguros y escalables para respaldar el crecimiento y las operaciones diarias.
- Implementar y Gestionar la Seguridad de la Información. Desarrollar políticas y procedimientos de seguridad de la información para proteger los datos y sistemas de la organización contra amenazas internas y externas, minimizando riesgos y garantizando la continuidad del negocio.
- Automatizar Procesos para Aumentar la Eficiencia Operativa. Identificar e implementar soluciones tecnológicas que permitan automatizar procesos repetitivos y mejorar la productividad de otras áreas, reduciendo el tiempo de respuesta y mejorando la experiencia de los usuarios internos.

### **Funciones generales del área**

#### **Soporte Técnico y Resolución de Incidencias**

Brindar asistencia a los usuarios internos para resolver problemas técnicos y consultas relacionadas con el uso de software, hardware y redes, asegurando la continuidad operativa en la organización.

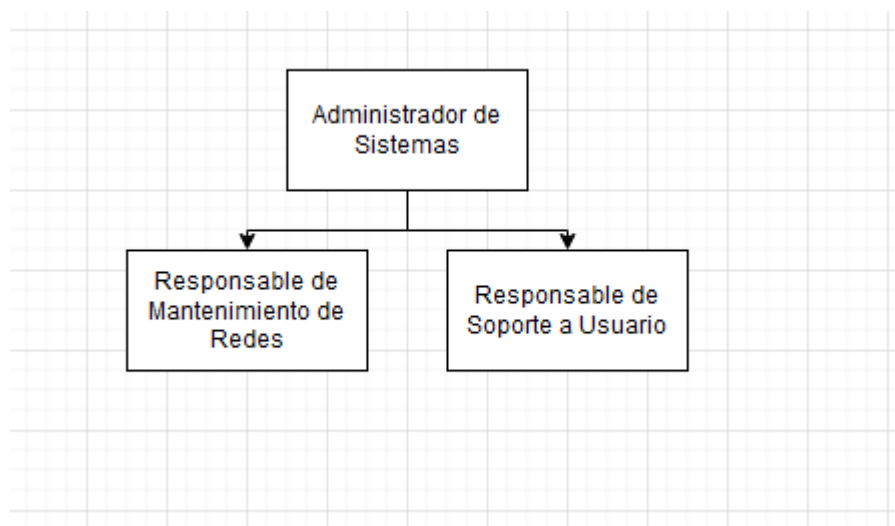
#### **Administración de Infraestructura y Redes**

Gestionar y mantener la infraestructura de TI, incluyendo servidores, redes, sistemas de almacenamiento y dispositivos, garantizando que los recursos estén disponibles, sean seguros y se mantengan actualizados.

## Desarrollo y Mantenimiento de Software

Desarrollar, implementar y mantener aplicaciones y sistemas internos que optimicen los procesos de la organización, adaptándose a los requisitos específicos y mejorando la eficiencia de las operaciones.

### Organigrama del área



*Ilustración 2. Organigrama: Área de Sistemas*

## **I. INTRODUCCIÓN**

Escribir:

Explica los motivos empresariales y personales que despertaron el interés por desarrollar el proyecto, aclarando en qué consiste, el porqué del mismo, dónde y en qué tiempo se llevó a cabo.

Describe en forma resumida el contenido de la memoria (se breve y conciso), de preferencia por cada uno de los capítulos, así como la metodología empleada y los resultados obtenidos.

Recuerda que, en este apartado, deberá quedar claro de qué se trata tu proyecto o cómo contribuye con el proyecto Institucional, qué partes lo componen, de qué manera lo desarrollaste y a qué resultados llegaste.

## II. PLANTEAMIENTO DEL PROYECTO

### Objetivo general

Desarrollar un sistema de Helpdesk centralizado para optimizar la gestión de tickets de soporte técnico en la empresa, mejorando la comunicación entre el Área de Sistemas y los usuarios internos, reduciendo la duplicidad de esfuerzos y facilitando un seguimiento eficiente de las incidencias.

### Objetivos específicos

- **Facilitar la creación y gestión de tickets.** Este objetivo es claro y enfocado. Expresa tanto la acción (implementar el sistema de tickets) como la finalidad (simplificar el reporte y gestión de incidencias). Es un buen objetivo específico porque aborda una necesidad fundamental del proyecto.
- **Mejorar la comunicación y transparencia.** Este objetivo es pertinente y relevante, ya que plantea establecer notificaciones y seguimiento en tiempo real para mejorar la interacción entre usuarios y administradores. Está bien definido y orientado hacia un propósito específico (transparencia en el proceso).
- **Optimizar la asignación de recursos.** Este objetivo es claro y práctico, pues busca reducir la duplicidad de esfuerzos mediante una asignación organizada de tareas. Su propósito es mejorar la eficiencia del equipo, un aspecto crítico para el éxito del proyecto.
- **Mantener un historial de tickets.** Este objetivo es adecuado porque permite identificar problemas recurrentes y contribuir a la mejora del mantenimiento preventivo. Además, es práctico y plantea una finalidad de análisis de patrones que puede ser muy útil para la empresa.
- **Garantizar la seguridad de los datos.** Es un objetivo esencial, dado que asegura que la información esté protegida, alineándose con los requisitos de seguridad de datos. Está bien formulado y es necesario para la credibilidad y el éxito del sistema.

- **Asegurar la escalabilidad del sistema.** Este objetivo es muy apropiado porque considera el crecimiento y adaptabilidad del sistema en el tiempo, lo cual es importante para cualquier sistema de soporte técnico centralizado. Está bien formulado y plantea una finalidad que beneficiará a largo plazo.

### **Justificación**

La implementación de un sistema de Helpdesk centralizado es crucial para optimizar la gestión de problemas técnicos en la empresa. Actualmente, el uso de canales informales como WhatsApp genera desorganización, duplicidad de esfuerzos y dificulta el seguimiento de incidencias, lo que causa insatisfacción en los usuarios y el equipo de soporte.

Este proyecto mejorará la eficiencia administrativa, al permitir una asignación clara y priorización de tareas, reduciendo tiempos de respuesta y costos operativos. En términos tecnológicos, brindará un entorno de soporte automatizado que no solo facilitará el seguimiento histórico de incidencias, sino que también permitirá identificar patrones para mejorar el mantenimiento preventivo.

El impacto humano es notable: al mejorar la comunicación y transparencia, se eleva la satisfacción y colaboración entre usuarios y técnicos. Además, el proyecto asegura la protección de datos, alineándose con principios éticos y de seguridad. En definitiva, esta solución contribuirá al desarrollo organizacional, optimizando la gestión de recursos y aumentando la productividad en la empresa.

### **Viabilidad del proyecto**

Para el desarrollo del proyecto, se utilizará una red pequeña donde un servidor virtual alojará la aplicación web y almacenará los datos localmente en la misma computadora. Este servidor virtual permitirá que tanto los usuarios como el equipo de Sistemas accedan a la plataforma desde dispositivos conectados en la red. La infraestructura local facilitará la ejecución de la plataforma y el acceso directo a los datos sin depender de servicios en la nube, optimizando recursos y garantizando una configuración rápida y controlada de la red y del almacenamiento de datos.

## **Impacto**

A corto plazo, mejorará la organización y el flujo de solicitudes, facilitando la administración de tickets y reduciendo los tiempos de respuesta. En el mediano plazo, permitirá al equipo de soporte gestionar los recursos de manera más eficiente, mejorando la productividad y satisfacción de los usuarios internos. A largo plazo, el sistema fomentará una mayor confiabilidad operativa, facilitará la identificación de patrones en los problemas y reducirá costos asociados a la gestión desorganizada, promoviendo un entorno de soporte más efectivo y adaptado a las necesidades de la empresa.

### **III. MARCO TEÓRICO**

#### **Metodología SCRUM**

Scrum es una metodología ágil para gestionar y completar proyectos complejos. Se centra en la colaboración, la flexibilidad y la entrega rápida de resultados valiosos.

La estructura de Scrum es beneficiosa para proyectos de software de soporte técnico, ya que permite ajustar rápidamente el desarrollo según las necesidades de los usuarios internos y las prioridades de negocio que puedan surgir. Cada sprint concluye con una revisión y una retrospectiva, lo cual permite evaluar el avance y realizar ajustes para el siguiente ciclo. La metodología Scrum se adapta bien a proyectos donde se requieren mejoras continuas y retroalimentación constante, asegurando que el sistema de Helpdesk evolucione de manera controlada y orientada a la satisfacción de los usuarios.

#### **Beneficios de Scrum**

- **Adaptabilidad:** Responde rápidamente a cambios y necesidades emergentes.
- **Transparencia:** Mejora la comunicación y visibilidad del progreso.
- **Eficiencia:** Facilita entregas rápidas y continuas de valor.

#### **Frameworks**

En el desarrollo de software, los frameworks son estructuras predefinidas que proporcionan herramientas y componentes reutilizables para construir aplicaciones de manera más rápida y organizada. Un framework establece una arquitectura base, definiendo patrones y buenas prácticas que facilitan el desarrollo y mantenimiento del proyecto, especialmente en aplicaciones complejas como un sistema de Helpdesk.

Los frameworks permiten al equipo de desarrollo enfocarse en las funcionalidades específicas del proyecto en lugar de tener que implementar desde cero estructuras comunes, como el manejo de bases de datos, autenticación y control de rutas. En este proyecto, frameworks como Express (para el backend) y React (para el Frontend) ofrecen módulos y funcionalidades integradas que permiten gestionar peticiones, definir interfaces de usuario y estructurar la lógica de negocio de manera modular y escalable.

## API RESTful

Una API RESTful (Representational State Transfer) es un estilo de arquitectura que facilita la comunicación entre diferentes sistemas mediante el uso de peticiones HTTP y principios REST. En el desarrollo de un sistema de Helpdesk, una API RESTful permite al Frontend y backend comunicarse de manera eficiente, gestionando datos de usuarios, tickets, y otras operaciones relacionadas con la gestión de incidencias.

Una API RESTful se organiza en endpoints, que son puntos de acceso definidos para manejar acciones específicas, como obtener, crear, actualizar o eliminar recursos. Estos endpoints siguen una estructura basada en recursos y utilizan métodos HTTP como GET (para obtener datos), POST (para crear datos), PUT (para actualizar datos) y DELETE (para eliminar datos). Esta organización permite que las interacciones con el sistema de Helpdesk sean consistentes y previsibles, simplificando la integración con interfaces de usuario y aplicaciones externas.

## Node.js

Node.js es un entorno de ejecución de JavaScript que permite desarrollar aplicaciones del lado del servidor. Aquí tienes una lista de las principales ventajas y características de Node.js que lo hacen adecuado para el desarrollo de un sistema:

- **Asincronía y no bloqueo:** Node.js utiliza un modelo basado en eventos y operaciones no bloqueantes, lo que permite manejar múltiples solicitudes de manera eficiente sin bloquear el procesamiento de otras. Esto es ideal para aplicaciones de tiempo real que requieren rapidez en la respuesta.
- **Escalabilidad:** Al ser modular, Node.js permite la creación de aplicaciones escalables, adecuadas para proyectos que pueden necesitar crecer en funcionalidad y capacidad a lo largo del tiempo.
- **Único lenguaje en Frontend y backend:** Como Node.js permite utilizar JavaScript tanto en el Frontend como en el backend, facilita la integración y el desarrollo continuo, especialmente en equipos que ya están familiarizados con JavaScript.
- **Gran ecosistema de paquetes:** Con el administrador de paquetes npm (Node Package Manager), Node.js ofrece acceso a una extensa colección de librerías y módulos que permiten agregar funcionalidades rápidamente sin



desarrollar desde cero, como autenticación, seguridad y manipulación de bases de datos.

- **Alto rendimiento:** Node.js utiliza el motor V8 de Google para ejecutar JavaScript, lo que le otorga velocidad y eficiencia en el procesamiento de operaciones, ideal para aplicaciones que gestionan una gran cantidad de solicitudes.
- **Comunidad activa:** La gran comunidad de desarrolladores de Node.js asegura una actualización constante de sus librerías, soporte continuo y acceso a mejoras de seguridad y rendimiento.

## PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) conocido por su estabilidad y robustez en el manejo de grandes volúmenes de datos y consultas complejas. En un sistema de Helpdesk, PostgreSQL permite almacenar de forma eficiente datos relacionados con los usuarios, tickets, actividades y cualquier información relevante para el seguimiento de incidencias. Algunas ventajas de PostgreSQL incluyen:

Soporte de transacciones: PostgreSQL asegura la integridad de los datos mediante transacciones, lo que permite realizar operaciones complejas sin comprometer la consistencia de la base de datos.

- **Escalabilidad:** PostgreSQL es capaz de manejar grandes volúmenes de datos y puede crecer en capacidad junto con las necesidades de la empresa.
- **Seguridad y control de acceso:** PostgreSQL permite gestionar permisos detallados de usuarios, asegurando que solo personal autorizado pueda acceder o modificar la información.
- **Soporte para JSON y consultas avanzadas:** PostgreSQL facilita la manipulación y almacenamiento de datos semiestructurados (como JSON), permitiendo consultas avanzadas y eficientes.

## Gestión de Incidencias

La gestión de incidencias es el proceso sistemático de identificar, registrar, y resolver problemas técnicos que pueden afectar el funcionamiento de una

organización. Este proceso es parte fundamental de la gestión de servicios de TI y se centra en restaurar el servicio normal de la organización lo más rápido posible, minimizando el impacto de los problemas en el negocio.

Los sistemas de gestión de incidencias suelen estructurarse para facilitar el registro y seguimiento de cada incidencia mediante un sistema de tickets. Estos tickets permiten organizar y priorizar los problemas técnicos de acuerdo con su urgencia e impacto en la organización. La gestión de incidencias también incluye la resolución y cierre de los tickets, garantizando que las soluciones se implementen de manera efectiva y que los problemas se documenten adecuadamente.

Uno de los beneficios clave de una gestión de incidencias organizada es la posibilidad de identificar patrones en los problemas reportados. Esto permite que las organizaciones implementen mejoras preventivas, disminuyendo la recurrencia de ciertos problemas y contribuyendo a un entorno de TI más estable. Además, al centralizar la información de incidencias y su resolución, se facilita la colaboración y la comunicación entre los equipos técnicos y los usuarios finales, mejorando la transparencia y la eficiencia en la atención de problemas técnicos.

### **Seguridad de la información**

La seguridad de la información es el conjunto de prácticas y medidas destinadas a proteger los datos y la infraestructura tecnológica de una organización frente a accesos no autorizados, alteraciones, pérdidas o ataques. Este campo abarca múltiples principios y técnicas que buscan garantizar la confidencialidad, integridad, y disponibilidad de la información, conocidos como la tríada CIA (Confidentiality, Integrity, Availability).

**Confidencialidad:** Este principio asegura que solo personas autorizadas puedan acceder a la información sensible. Las técnicas para mantener la confidencialidad incluyen controles de acceso, encriptación de datos y autenticación de usuarios.

**Integridad:** La integridad implica que los datos no sean alterados o manipulados de forma no autorizada. Esto se asegura mediante mecanismos como el control de

versiones, auditorías y el uso de firmas digitales que permiten verificar que los datos no han sido modificados.

**Disponibilidad:** La disponibilidad garantiza que la información esté accesible para los usuarios autorizados cuando la necesiten. Esto implica la implementación de sistemas de respaldo, redundancia en infraestructura y medidas de protección frente a ataques de denegación de servicio (DoS).

## **Tailwind CSS**

Tailwind CSS es un framework de utilidades CSS diseñado para facilitar la creación de interfaces de usuario personalizadas mediante un enfoque orientado a utilidades. En lugar de ofrecer componentes predefinidos, como botones o formularios ya estilizados, Tailwind proporciona una amplia variedad de clases CSS individuales que permiten aplicar estilos directamente en el HTML. Este enfoque "utility-first" permite a los desarrolladores construir rápidamente componentes personalizados sin tener que escribir CSS adicional, ya que cada clase representa una propiedad específica, como márgenes, colores, tamaños de fuente o alineación.

Una de las principales características de Tailwind es su alta personalización. Mediante un archivo de configuración (`tailwind.config.js`), los usuarios pueden ajustar el diseño para adaptarlo a las necesidades de su proyecto, definiendo colores, espaciados, fuentes y otros parámetros específicos. Esto facilita la coherencia visual a lo largo del proyecto y permite realizar cambios globales de manera rápida y sencilla.

## **IV. DESARROLLO DE ACTIVIDADES DEL PROYECTO**

### **Alcance del proyecto**

El sistema desarrollado implementa un mecanismo de autenticación seguro basado en el uso de correo electrónico y contraseña, donde todos los datos transmitidos son encriptados para proteger la información sensible. Esta funcionalidad garantiza la confidencialidad en el acceso de los usuarios, permitiendo solo a los administradores acceder a las herramientas de gestión.

La administración de usuarios es centralizada, brindando a los administradores la capacidad de crear, modificar y eliminar cuentas de usuario de manera controlada y sin comprometer el historial de tickets ya registrados. Este sistema asegura que cualquier cambio en las cuentas de usuario no afecte los datos previos de los tickets, preservando así la integridad de la información almacenada.

En el área de gestión de solicitudes y tareas, el sistema cuenta con un módulo de tickets que permite a los administradores visualizar los tickets activos y cambiar su estado según el progreso de cada solicitud. Los administradores pueden modificar el estado de los tickets en tiempo real, lo que facilita el seguimiento y resolución ágil de las tareas.

Se ha implementado un sistema de notificaciones que permite a los administradores recibir actualizaciones sobre los cambios de estado de los tickets. Estas notificaciones son visibles dentro del sistema, y también pueden ser enviadas por correo electrónico, proporcionando así una vía rápida y eficiente de acceso a la información relevante.

### **Metodología**

Para este proyecto, se ha adoptado la metodología ágil Scrum, adaptada al trabajo individual. El desarrollo se organiza en sprints, permitiendo un enfoque iterativo e incremental para avanzar en la creación del sistema de gestión de usuarios y tickets. Cada sprint incluye fases de planificación, desarrollo y revisión, lo que

facilita el seguimiento de los objetivos y la adaptación rápida a cualquier ajuste en los requisitos o hallazgos durante el proceso.

En cada ciclo, se evalúan las funcionalidades desarrolladas, como el módulo de autenticación o la interfaz de tickets, ajustándolas según sea necesario. Esto permite mantener una mejora continua y garantizar que el sistema evolucione de acuerdo con las necesidades del proyecto y los objetivos de negocio, validando al final de cada sprint el cumplimiento de los requisitos y asegurando que las funcionalidades cumplan con las expectativas de usabilidad y rendimiento.

### **Planificación del proyecto**

En la fase de planificación del proyecto, se hizo una visita a la empresa para entender mejor sus necesidades y lo que buscaban lograr con el proyecto. Este acercamiento permitió tener una idea general de los objetivos y analizar cómo se manejaban actualmente las actividades relacionadas con la atención de tickets, identificando posibles problemas y áreas de mejora.

También se participó en el proceso de atención de tickets para conocer de cerca cómo funciona el flujo de trabajo y los retos que enfrenta el equipo. Esto incluyó ver las formas de comunicación y colaboración que usan para atender a los usuarios, lo cual ayudó a entender mejor sus prácticas y necesidades.

También se evaluó la viabilidad del proyecto tomando en cuenta los criterios definidos por el área responsable. Además, se consideró la escalabilidad para asegurar que el proyecto pudiera adaptarse a futuras necesidades de crecimiento o cambio.

Se optó por utilizar la metodología Scrum, considerando los posibles retrasos que podrían surgir en algunas secciones del proyecto. Esta metodología también permite mantener una colaboración continua con el usuario final, facilitando su contribución y asegurando una retroalimentación efectiva durante el desarrollo.

[Imagen cronograma]

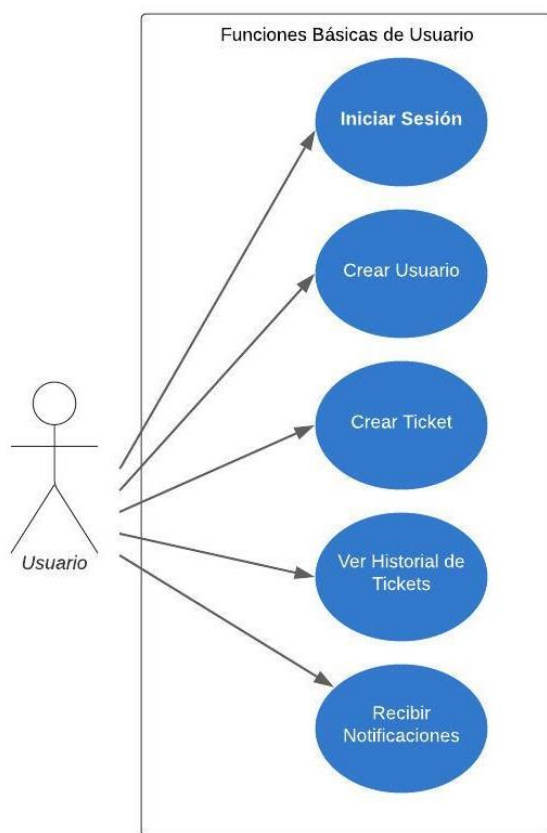
### **Análisis y Documentación Inicial del proyecto**

Se realizaron diversas entrevistas con los usuarios finales, en las que se resolvieron dudas como: ¿Cuáles son las etapas del proceso?, ¿Qué dispositivos

utilizan?, ¿Qué categorías manejan?, y ¿Cuáles son las áreas involucradas?, entre otras preguntas que ayudaron a mejorar el entendimiento de los requerimientos. Una vez respondidas la mayoría de las preguntas, se establecieron criterios para definir los requerimientos funcionales y no funcionales.

## Casos de uso

### Funciones Básicas de Usuario



*Ilustración 3. Diagrama Casos de Uso: Funciones Básicas de Usuario*

Casos de uso		No.1
Nombre	Registrar Usuario	
Actor	Cualquier Usuario	
Descripción	Proceso de registro de un nuevo usuario en el sistema, que puede ser iniciado por cualquier usuario, ya sea un administrador o un usuario nuevo. Incluye la verificación de duplicados en el correo electrónico y número de teléfono.	
Flujo principal	Eventos Actor	Eventos Sistema

	Paso 1. El usuario rellena el formulario de registro (Nombre, Apellido, Correo, Teléfono, Contraseña, Empresa, Departamento), y hace Click en Enviar Datos.	Paso 2. El sistema analiza los datos del formulario (Correo y Teléfono) buscando coincidencias.
		Paso 3. No existe ninguna coincidencia y se pasa a la creación de una llave de usuario
		Paso 4. Se encripta la contraseña previamente a su almacenado.
		Paso 5. Se almacenan todos los datos directamente en la base de datos.
	Paso 7. El usuario se le muestra la alerta en la vista de registro exitoso y es redirigido a el Login.	Paso 6. El sistema devuelve un código 200 al ser una petición exitosa y se le muestra una alerta al usuario.
Flujo alternativo	<p>Paso 3. Existe un correo previo al guardado de datos y se regresa un mensaje de error.</p> <p>Paso 3. Existe un número de teléfono previamente registrado se regresa un mensaje de error.</p>	
Precondición	El usuario está en la pantalla de creación de usuario y ha ingresado los datos requeridos.	
Postcondición	Se crea un nuevo usuario en la base de datos, o el sistema informa de un error si hay datos duplicados.	
Presunción	Cualquier usuario puede iniciar el proceso de registro sin necesidad de intervención adicional.	

Casos de uso		No.2
Nombre	Iniciar sesión	
Actor	Cualquier usuario registrado previamente	
Descripción	Proceso de autenticación de un usuario para acceder al sistema, verificando sus credenciales, el estado de su cuenta y su rol asignado. Si las credenciales son válidas y la cuenta está activa, el sistema genera un token de autenticación.	
Flujo principal	Eventos Actor	Eventos Sistema

	Paso 1. El usuario abre la plataforma e inserta sus datos (Correo, Contraseña).	Paso 2. Los datos son buscados usando el correo con la base de datos.
		Paso 3. Si el correo es encontrado se busca si la cuenta se encuentra en estado activo.
		Paso 4. Si lo anterior esta activa, si la contraseña es correcta comparando la encriptada
	Paso 6. El token se almacena, y el usuario es redirigido a la vista de su rol.	Paso 5. Si lo anterior es válido generar un token que pueda autenticar al usuario
Flujo alternativo	<p>Paso 3. No se encuentra ningún correo, y envía una alerta al usuario con credenciales incorrectas.</p> <p>Paso 4. La cuenta no se encuentra activa, y se envía una alerta al usuario de cuenta inactiva.</p> <p>Paso 5. La contraseña es incorrecta, y se envía una alerta al usuario con credenciales incorrectas.</p>	
Precondición	El usuario está en la pantalla de inicio de sesión y ha ingresado sus credenciales.	
Postcondición	El usuario recibe un token de autenticación válido y tiene acceso al sistema.	
Presunción	El usuario tiene una cuenta previamente registrada en el sistema.	

Casos de uso		No.3
Nombre	Crear Ticket	
Actor	Cualquier usuario previamente registrado	
Descripción	Proceso de autenticación de un usuario para acceder al sistema, verificando sus credenciales, el estado de su cuenta y su rol asignado. Si las credenciales son válidas y la cuenta está activa, el sistema genera un token de autenticación.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario abre la plataforma e inserta sus datos (Correo, Contraseña).	Paso 2. Los datos son buscados usando el correo con la base de datos.



		Paso 3. Si el correo es encontrado se busca si la cuenta se encuentra en estado activo.
		Paso 4. Si lo anterior esta activa, si la contraseña es correcta comparando la encriptada
	Paso 6. El token se almacena, y el usuario es redirigido a la vista de su rol.	Paso 5. Si lo anterior es válido generar un token que pueda autenticar al usuario
Flujo alternativo	<p>Paso 3. No se encuentra ningún correo, y envía una alerta al usuario con credenciales incorrectas.</p> <p>Paso 4. La cuenta no se encuentra activa, y se envía una alerta al usuario de cuenta inactiva.</p> <p>Paso 5. La contraseña es incorrecta, y se envía una alerta al usuario con credenciales incorrectas.</p>	
Precondición	El usuario está en la pantalla de inicio de sesión y ha ingresado sus credenciales.	
Postcondición	El usuario recibe un token de autenticación válido y tiene acceso al sistema.	
Presunción	El usuario tiene una cuenta previamente registrada en el sistema.	

Casos de uso		No.4
Nombre	Ver historial de tickets	
Actor	Cualquier usuario previamente registrado	
Descripción	El usuario accede al sistema para visualizar el historial de sus tickets. Puede ver detalles de cada ticket, incluyendo el estado, fecha de creación y comentarios asociados, permitiéndole realizar un seguimiento de sus solicitudes anteriores.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede a la vista de "Mis tickets".	Paso 2. Se realiza una petición automática, primero analiza el token si es válido.
		Paso 3. Si el token fue valido, entonces hace la búsqueda en la base de datos.

	Paso 5. El usuario se le muestra una tabla que contiene los datos de la respuesta.	Paso 4. Se devuelve una respuesta 200 al usuario, junto a los datos.
Flujo alternativo	Paso 3. El token es invalido, y no se retorna ningún dato, solo una alerta.	
Precondición	El usuario está autenticado en el sistema.	
Postcondición	El usuario visualiza el historial de sus tickets o un mensaje indicando que no hay tickets registrados.	
Presunción	El sistema mantiene un registro actualizado de todos los tickets asociados al usuario.	

Casos de uso		No.5
Nombre	Recibir Notificaciones	
Actor	Cualquier usuario previamente registrado	
Descripción	El usuario recibe notificaciones en el sistema sobre actualizaciones relevantes, como cambios en el estado de sus tickets, nuevos comentarios o cualquier otra acción relacionada con sus solicitudes. Esto permite al usuario mantenerse informado sobre el progreso de sus tickets.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede al panel de notificaciones	Paso 2. Se realiza una petición automática, primero analiza el token si es válido.
		Paso 3. Se hace la búsqueda de todas las notificaciones pertenecientes al usuario
	Paso 5. El panel de notificaciones, muestra todas las notificaciones al usuario.	Paso 4. Se devuelve una respuesta 200 al usuario, junto a los datos.
Flujo alternativo	Paso 3. El token es invalido, por lo que no se regresa ningún dato, y se envía una alerta al usuario.	
Precondición	El usuario tiene una cuenta en el sistema y tickets previos asociados que pueden recibir actualizaciones.	
Postcondición	El usuario visualiza las notificaciones y está informado sobre las actualizaciones de sus tickets.	
Presunción	El sistema genera notificaciones automáticamente cuando hay cambios en los tickets asociados al usuario.	

## Funciones avanzadas de Administración

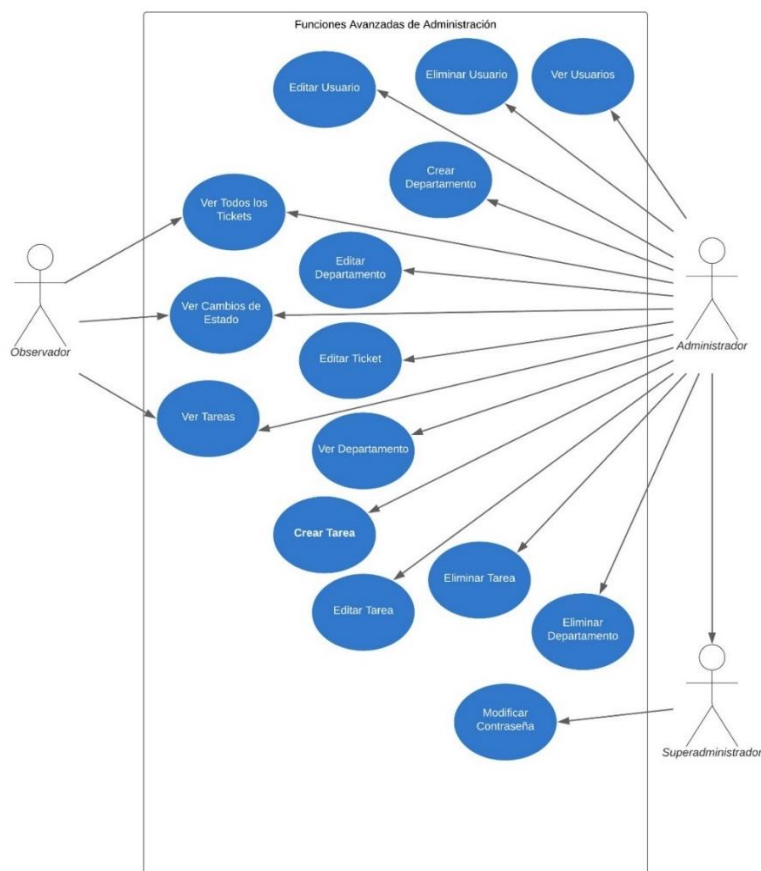


Ilustración 4. Diagrama Casos de Uso: Funciones avanzadas de Administración

Casos de uso		No.6
Nombre	Ver todos los Tickets	
Actor	Observador, Administrador o Superadministrador	
Descripción	Los usuarios con roles de observador, administrador o superadministrador pueden visualizar todos los tickets registrados en el sistema, permitiéndoles revisar el estado, detalles y actividad de cada ticket sin restricciones de usuario.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Accede a la pestaña principal o a la pestaña de tickets,	Paso 2. El sistema responde y manda una petición con el token.

		Paso 3. Si el token es válido, pasa a una comparativa de roles para ver si es posible enviar los datos.
		Paso 4. Si los roles concuerdan, se realiza una petición a la base de datos.
	Paso 6. Los datos son mostrados en la ventana de tickets.	Paso 5. Se realiza un envío de los datos, con una respuesta 200.
Flujo alternativo	<p>Paso 4. El token no es válido y se envía una alerta.</p> <p>Paso 5. Si el usuario no tiene el rol adecuado, el sistema muestra un mensaje de error y niega el acceso a la vista de todos los tickets.</p>	
Precondición	El usuario ha iniciado sesión con un rol de observador, administrador o superadministrador.	
Postcondición	El usuario visualiza la lista completa de tickets o recibe un mensaje de error si no tiene el rol adecuado.	
Presunción	Solo los roles de observador, administrador y superadministrador tienen acceso completo para visualizar todos los tickets en el sistema.	

Casos de uso		No.7
Nombre	Ver Cambios de estado de un Ticket	
Actor	Observador, Administrador o Superadministrador	
Descripción	Un administrador o superadministrador puede cambiar el estado de un ticket específico para reflejar su progreso o resolución. Esto permite actualizar el ticket a estados como "Abierto", "En proceso", "Pendiente", "Cerrado", entre otros.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede a un ticket en particular, y accede al componente de cambios de estado.	Paso 2. Se envía una solicitud, con el identificador del ticket y el token del usuario.
		Paso 3. Se revisa si el token es válido, para continuar.
		Paso 4. Si el token es válido, se revisa el rol del usuario que solicita la petición.

		Paso 5. Si el rol coincide, se realiza la búsqueda de todas las coincidencias del ticket.
	Paso 7. El componente recibe todos los datos y se los muestra al usuario.	Paso 6. Un único registro debe encontrarse y se debe enviar todos sus cambios de estado junto a una respuesta 200.
Flujo alternativo	<p>Paso 4. El token no es válido, por lo que se retorna una alerta.</p> <p>Paso 5. El rol no concuerda con lo esperado y se retorna una alerta.</p> <p>Paso 6. Si no hay notificaciones nuevas, el sistema muestra un mensaje indicando que no hay actualizaciones.</p>	
Precondición	El usuario tiene una cuenta en el sistema y tickets previos asociados que pueden recibir actualizaciones.	
Postcondición	El usuario visualiza las notificaciones y está informado sobre las actualizaciones de sus tickets.	
Presunción	El sistema genera notificaciones automáticamente cuando hay cambios en los tickets asociados al usuario.	

Casos de uso		No.8
Nombre	Ver tareas de un ticket	
Actor	Observadores, Administradores, Superadministradores	
Descripción	El usuario puede consultar las tareas asignadas a un ticket específico para obtener un desglose de las acciones planificadas o realizadas en relación con el ticket. Esto permite al usuario entender los pasos que se han tomado o que están pendientes para la resolución de su solicitud.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede al ticket, y se va a un apartado de Ver tareas.	Paso 2. Se envía una solicitud, con el identificador del ticket y el token del usuario.
		Paso 3. Se revisa si el token es válido, para continuar.
		Paso 4. Si el token es válido, se revisa el rol del usuario que solicita la petición.

		Paso 5. Si el rol coincide, se realiza la búsqueda de todas las coincidencias del ticket.
	Paso 7. Los datos son mostrados en la ventana de tickets.	Paso 6. Un único registro debe encontrarse y se debe enviar todos sus cambios de estado junto a una respuesta 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 4. El token no es válido y se envía una alerta.</li> <li>• Paso 5. Si el usuario no tiene el rol adecuado, el sistema muestra un mensaje de error y niega el acceso a la vista de las tareas de ese ticket.</li> <li>• Paso 6. Si el ticket no tiene tareas asociadas, el sistema muestra un mensaje indicando que no hay tareas registradas para ese ticket.</li> </ul>	
Precondición	El usuario tiene acceso al ticket seleccionado y puede visualizar las tareas asociadas.	
Postcondición	El usuario visualiza las tareas planificadas para el ticket o un mensaje indicando que no hay tareas registradas.	
Presunción	El sistema mantiene un registro de todas las tareas planeadas y en curso para cada ticket.	

Casos de uso		No.9
Nombre	Ver usuarios	
Actor	Administradores y Superadministradores	
Descripción		
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede al ticket, y se va a un apartado de Usuarios	Paso 2. Se envía una solicitud, el token del usuario.
		Paso 3. Se revisa si el token es válido, para continuar.
		Paso 4. Si el token es válido, se revisa el rol del usuario que solicita la petición.
		Paso 5. Si el rol coincide, se realiza la búsqueda de todos los usuarios filtrando datos que exponen su seguridad.
	Paso 7. Los datos son mostrados en la ventana de usuarios.	Paso 6. Se envían todos los datos de usuarios junto a una respuesta 200

Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 4. El token no es válido y se envía una alerta.</li> <li>• Paso 5. Si el usuario no tiene el rol adecuado, el sistema muestra un mensaje de error y niega la información a la vista.</li> <li>• Paso 6. Se devuelve vacío por lo tanto se muestra que no hay usuarios.</li> </ul>
Precondición	<p>El usuario debe estar autenticado y tener un rol de Administrador o Superadministrador para poder acceder a la lista de usuarios.</p> <p>El usuario tiene acceso al sistema y un token válido que permita la consulta de usuarios.</p>
Postcondición	El usuario visualiza una lista de usuarios filtrada según su rol, permitiéndole ver solo los datos que le corresponden de acuerdo con su nivel de acceso.
Presunción	<p>El sistema mantiene actualizada la lista de usuarios en función de los cambios recientes (altas, bajas o modificaciones de usuarios).</p> <p>Solo los usuarios con los permisos necesarios pueden acceder a esta información, garantizando la seguridad de los datos de los usuarios en la lista.</p>

Casos de uso		No.10
Nombre	Editar Usuario	
Actor	Administrador, superadministrador	
Descripción	Permite a un Administrador o superadministrador modificar la información de un usuario existente en el sistema, como sus datos personales o rol asignado, según los permisos del administrador.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede a la pestaña "Usuarios".	
	Paso 2. Selecciona la opción editar usuario y modifica los datos con un formulario.	Paso 3. Se envía una petición con los datos del formulario, el token y la clave única de usuario de ese registro.
		Paso 4. Se revisa si el token es válido.

		Paso 5. El token es válido y se revisan permisos.
	Paso 7. La vista muestra una alerta que dice "Datos correctamente guardados".	Paso 6. Si los permisos son válidos, se hace la modificación de los datos y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta que dice, token invalido.</li> <li>• Paso 6. Los permisos no son correctos y se manda una alerta, que dice "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta "Los datos no se guardaron correctamente".</li> </ul>	
Precondición	<p>El Administrador está autenticado y tiene permisos para editar usuarios.</p> <p>El usuario a editar existe en el sistema.</p>	
Postcondición	<p>La información del usuario ha sido actualizada en el sistema.</p> <p>Se guarda un registro del cambio en el historial de modificaciones de usuarios, si es necesario.</p>	
Presunción	<p>Solo los roles autorizados pueden editar información de usuarios.</p> <p>La información del usuario es visible solo para aquellos con permisos adecuados, protegiendo datos sensibles.</p>	

Casos de uso		No.11
Nombre	Eliminar Usuario	
Actor	Administrador, superadministrador	
Descripción	Permite a un Administrador o Superadministrador eliminar a un usuario del sistema, lo que implica deshabilitar su cuenta para evitar su acceso en el futuro. La eliminación se realiza mediante un "soft delete" para mantener los datos históricos de sus acciones en el sistema.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede a la pestaña "Usuarios".	
	Paso 2. Selecciona la opción "Eliminar Usuario".	Paso 3. Se envía una petición que lleva el token, y la clave de usuario



		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. Se envía una alerta “El registro se ha eliminado correctamente”.	Paso 6. Si los permisos son válidos, el registro es ubicado y eliminado, luego
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta “El token ha expirado o es invalido”.</li> <li>• Paso 6. Si los permisos no son válidos, y se envía una alerta “Permisos no suficientes”.</li> <li>• Paso 7. Si no se elimina el usuario, se muestra la alerta “La acción no se ha eliminado correctamente”.</li> </ul>	
Precondición	<p>El Administrador está autenticado y tiene permisos para eliminar usuarios.</p> <p>El usuario a eliminar existe en el sistema y no es un superadministrador.</p>	
Postcondición	<p>El usuario ha sido desactivado en el sistema y ya no tiene acceso.</p> <p>Se guarda un registro de la eliminación en el historial de actividades del sistema, si es necesario.</p>	
Presunción	<p>Solo los roles autorizados pueden eliminar usuarios.</p> <p>La eliminación es un proceso de “soft delete”, permitiendo conservar el historial de actividades del usuario.</p>	

Casos de uso		No.12
Nombre	Crear Departamento	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador crear un nuevo departamento dentro del sistema, asignando un nombre y otros detalles relevantes para su identificación y gestión.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Se accede a la pestaña de Departamentos	
	Paso 2. Se selecciona el botón de departamento, y	Paso 3. Se envía una petición con los datos del formulario, el

	escribes el nuevo nombre del departamento.	token y la clave única de usuario de ese registro.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. Se muestra una alerta "Se ha creado un nuevo departamento"	Paso 6. Si los permisos son válidos, se hace crea el nuevo registro y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta "El token ha expirado o es invalido".</li> <li>• Paso 6. Si los permisos no son válidos, y se envía una alerta "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta donde se ha creado un nuevo registro.</li> </ul>	
Precondición	<p>El Administrador está autenticado y tiene permisos para crear departamentos.</p> <p>El sistema permite la creación de un nuevo departamento sin duplicar nombres.</p>	
Postcondición	<p>El nuevo departamento ha sido creado y está disponible en el sistema para asignar usuarios o realizar otras operaciones.</p> <p>Se guarda un registro de la creación en el historial de actividades del sistema, si es necesario.</p>	
Presunción	<p>Solo los roles autorizados pueden crear departamentos.</p> <p>El sistema realiza verificaciones para asegurar la unicidad del nombre del departamento y la integridad de los datos ingresados.</p>	

Casos de uso		No.13
Nombre	Editar Departamento	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador editar un departamento dentro del sistema, asignando un nuevo nombre.	
Flujo principal	Eventos Actor	Eventos Sistema

	Paso 1. Acceder a la vista Departamentos.	
	Paso 2. Selecciona la opción editar departamento y modifica los datos con un formulario.	Paso 3. Se envía una petición con los datos del formulario, el token y la clave única de departamento de ese registro.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. La vista muestra una alerta que dice "Datos correctamente guardados".	Paso 6. Si los permisos son válidos, se hace la modificación de los datos y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta que dice, token invalido.</li> <li>• Paso 6. Los permisos no son correctos y se manda una alerta, que dice "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta "Los datos no se guardaron correctamente".</li> </ul>	
Precondición	<p>El Administrador está autenticado y tiene permisos para editar departamentos.</p> <p>El departamento a editar existe en el sistema.</p>	
Postcondición	<p>La información del departamento ha sido actualizada en el sistema.</p> <p>Se guarda un registro de la modificación en el historial de actividades del sistema, si es necesario.</p>	
Presunción	<p>Solo los roles autorizados pueden editar departamentos.</p> <p>La integridad de los datos del departamento es verificada antes de guardar los cambios.</p>	

Casos de uso		No.14
Nombre	Eliminar Departamento	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o superadministrador eliminar un departamento del sistema. Esta acción inactiva el	

	departamento, pero mantiene su información para fines históricos y de auditoría mediante un "soft delete".	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. El usuario accede a la pestaña "Departamentos".	
	Paso 2. Selecciona la opción "Eliminar Departamento".	Paso 3. Se envía una petición que lleva el token, y la clave de usuario
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. Se envía una alerta "El registro se ha eliminado correctamente".	Paso 6. Si los permisos son válidos, el registro es ubicado y eliminado, luego
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta "El token ha expirado o es invalido".</li> <li>• Paso 6. Si los permisos no son válidos, y se envía una alerta "Permisos no suficientes".</li> <li>• Paso 7. Si no se elimina el usuario, se muestra la alerta "La acción no se ha eliminado correctamente".</li> </ul>	
Precondición	<p>El Administrador está autenticado y tiene permisos para eliminar departamentos.</p> <p>El departamento a eliminar existe en el sistema y no es un departamento crítico que no pueda ser eliminado.</p>	
Postcondición	<p>El departamento ha sido marcado como inactivo y no es visible en las operaciones activas del sistema.</p> <p>Se guarda un registro de la eliminación en el historial de actividades del sistema, si es necesario.</p>	
Presunción	<p>Solo los roles autorizados pueden eliminar departamentos.</p> <p>La eliminación es un proceso de soft delete, permitiendo conservar la información para auditorías y referencias futuras.</p>	

Casos de uso		No.15
Nombre	Editar Ticket	
Actor	Administrador, Superadministrador	

Descripción	Permite a un Administrador o Superadministrador modificar la información de un ticket existente, como su estado, prioridad, o asignación, para mantener actualizados los detalles del ticket de acuerdo con el progreso o las necesidades del sistema.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Acceder a la vista Ticket.	
	Paso 2. Selecciona la opción editar ticket y modifica los datos con un formulario.	Paso 3. Se envía una petición con los datos del formulario, el token y la clave única de departamento de ese registro.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. La vista muestra una alerta que dice "Datos correctamente guardados".	Paso 6. Si los permisos son válidos, se hace la modificación de los datos y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta que dice, token invalido.</li> <li>• Paso 6. Los permisos no son correctos y se manda una alerta, que dice "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta "Los datos no se guardaron correctamente".</li> </ul>	
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para editar tickets.</li> <li>• El ticket a editar existe en el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La información del ticket ha sido actualizada en el sistema.</li> <li>• Se guarda un registro de la modificación en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden editar tickets.</li> <li>• La integridad de los datos del ticket es verificada antes de guardar los cambios.</li> </ul>	

Casos de uso		No.16
Nombre	Ver Departamento	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador visualizar la información detallada de un departamento específico, como su nombre, y la cantidad de usuarios asociados, para facilitar la administración y gestión de recursos dentro del sistema.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1.	
Flujo alternativo		
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para editar tickets.</li> <li>• El ticket a editar existe en el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La información del ticket ha sido actualizada en el sistema.</li> <li>• Se guarda un registro de la modificación en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden editar tickets.</li> <li>• La integridad de los datos del ticket es verificada antes de guardar los cambios.</li> </ul>	

Casos de uso		No.17
Nombre	Crear Tarea	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador crear una nueva tarea dentro del sistema, asignándole un nombre, y si esta completada para darle seguimiento.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Se accede a la publicación de un ticket.	
	Paso 2. Se selecciona el botón de tareas, y escribes	Paso 3. Se envía una petición con los datos del formulario, el

	la nueva tarea y haces Click, nueva tarea.	token y la clave única de usuario de ese registro.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. Se muestra una alerta "Se ha creado un nuevo tarea"	Paso 6. Si los permisos son válidos, se hace crea el nuevo registro y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta "El token ha expirado o es invalido".</li> <li>• Paso 6. Si los permisos no son válidos, y se envía una alerta "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta donde no se ha creado un nuevo registro.</li> </ul>	
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para crear tareas.</li> <li>• La información ingresada para la tarea cumple con los requisitos establecidos por el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La nueva tarea ha sido creada y está disponible en el sistema para asignación y seguimiento.</li> <li>• Se guarda un registro de la creación en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden crear tareas.</li> <li>• La integridad de los datos de la tarea es verificada antes de guardar los cambios.</li> </ul>	

Casos de uso		No.18
Nombre	Editar Tarea	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador modificar la información de una tarea existente, como el nombre, y si está completa conforme se avanza en su gestión.	
Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Acceder a la vista Ticket.	

	Paso 2. Selecciona la opción editar tarea y modifica los datos con un formulario.	Paso 3. Se envía una petición con los datos del formulario, el token y la clave única de departamento de ese registro.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. La vista muestra una alerta que dice "Datos correctamente guardados".	Paso 6. Si los permisos son válidos, se hace la modificación de los datos y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta que dice, token invalido.</li> <li>• Paso 6. Los permisos no son correctos y se manda una alerta, que dice "Permisos no suficientes".</li> <li>• Paso 7. Se muestra una alerta "Los datos no se guardaron correctamente".</li> </ul>	
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para editar tareas.</li> <li>• La tarea a editar existe en el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La información de la tarea ha sido actualizada en el sistema.</li> <li>• Se guarda un registro de la modificación en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden editar tareas.</li> <li>• La integridad de los datos de la tarea es verificada antes de guardar los cambios.</li> </ul>	

Casos de uso		No.19
Nombre	Eliminar Tarea	
Actor	Administrador, Superadministrador	
Descripción	Permite a un Administrador o Superadministrador eliminar una tarea del sistema. Esta acción inactiva la tarea, pero mantiene sus datos para fines de auditoría mediante un "soft delete".	
Flujo principal	Eventos Actor	Eventos Sistema



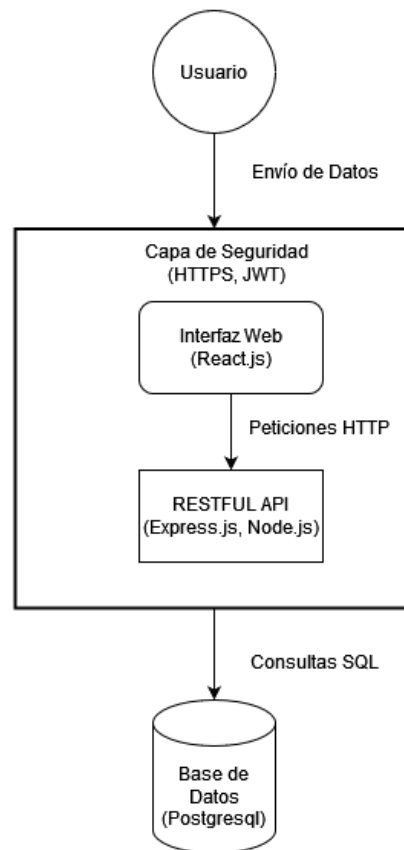
	Paso 1. El usuario accede a la publicación de un ticket.	
	Paso 2. Selecciona la opción "Eliminar Tarea".	Paso 3. Se envía una petición que lleva el token, y la clave de usuario
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. Se envía una alerta "El registro se ha eliminado correctamente".	Paso 6. Si los permisos son válidos, el registro es ubicado y eliminado, luego
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta "El token ha expirado o es invalido".</li> <li>• Paso 6. Si los permisos no son válidos, y se envía una alerta "Permisos no suficientes".</li> <li>• Paso 7. Si no se elimina el usuario, se muestra la alerta "La acción no se ha eliminado correctamente".</li> </ul>	
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para eliminar tareas.</li> <li>• La tarea a eliminar existe en el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La tarea ha sido marcada como inactiva y no es visible en las operaciones activas del sistema.</li> <li>• Se guarda un registro de la eliminación en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden eliminar tareas.</li> <li>• La eliminación es un proceso de soft delete, permitiendo conservar la información para auditorías y referencias futuras.</li> </ul>	

Casos de uso		No.20
Nombre	Modificar Contraseña	
Actor	Superadministrador	
Descripción	Permite a un Administrador o Superadministrador cambiar la contraseña de un usuario en caso de olvido o como medida de seguridad. Esta acción es útil para mantener la integridad de las cuentas y garantizar el acceso seguro al sistema.	

Flujo principal	Eventos Actor	Eventos Sistema
	Paso 1. Acceder a una vista de Usuarios.	
	Paso 2. Seleccionar un usuario y hacer Click en “Reestablecer contraseña”, escribirla confirmación y nueva contraseña.	Paso 3. Se envía la petición junto al token, la clave de usuario, y los nuevos datos.
		Paso 4. Se revisa si el token es válido.
		Paso 5. El token es válido y se revisan permisos.
	Paso 7. La vista muestra una alerta que dice “Contraseña actualizada”.	Paso 6. Si los permisos son válidos, se hace la modificación de los datos y se devuelve un status 200.
Flujo alternativo	<ul style="list-style-type: none"> <li>• Paso 5. El token no es válido, y se envía una alerta que dice, token invalido.</li> <li>• Paso 6. Los permisos no son correctos y se manda una alerta, que dice “Permisos no suficientes”.</li> <li>• Paso 7. Se muestra una alerta “Los datos no se guardaron correctamente”.</li> </ul>	
Precondición	<ul style="list-style-type: none"> <li>• El Administrador está autenticado y tiene permisos para modificar contraseñas de usuarios.</li> <li>• El usuario cuya contraseña se va a modificar existe en el sistema.</li> </ul>	
Postcondición	<ul style="list-style-type: none"> <li>• La contraseña del usuario ha sido actualizada en el sistema.</li> <li>• Se registra la actividad de modificación de contraseña en el historial de actividades del sistema, si es necesario.</li> </ul>	
Presunción	<ul style="list-style-type: none"> <li>• Solo los roles autorizados pueden modificar contraseñas.</li> <li>• La nueva contraseña se almacena de manera segura para proteger la información de acceso del usuario.</li> </ul>	

## Diseño de la Arquitectura del Sistema

Para el desarrollo de la arquitectura del sistema, se realizó un análisis exhaustivo del entorno donde se implementará la solución, incluyendo una evaluación de los sistemas operativos con los que el equipo de Sistemas está familiarizado. Este análisis permite considerar factores clave como el intercambio de información, el alojamiento y la adopción de una arquitectura cliente-servidor adecuada para optimizar el rendimiento y la interoperabilidad.



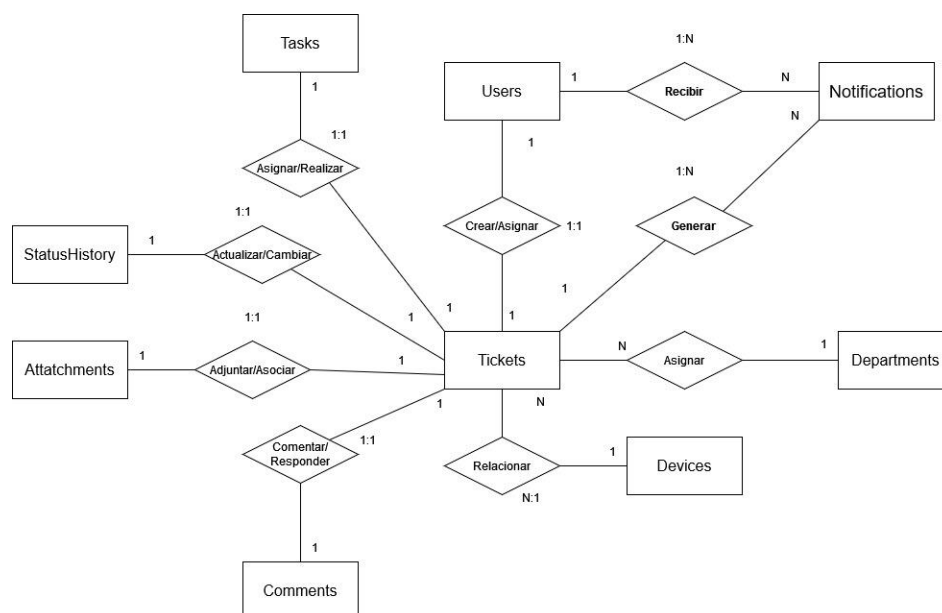
*Ilustración 5. Diagrama de Arquitectura*

## Desarrollo de la normalización de Base de datos

La base de datos diseñada tiene como objetivo proporcionar un sistema de gestión integral, que permite controlar, rastrear y organizar diversas interacciones y procesos dentro de la organización. Este sistema centralizado facilita la comunicación entre diferentes módulos, asegurando que la información sea

accesible y esté siempre actualizada. Al estructurar la base de datos, se priorizó la integridad y consistencia de los datos, permitiendo que cada registro tenga relaciones bien definidas que fortalecen el flujo de trabajo. Además, el modelo de datos apoya el seguimiento detallado de la actividad de cada entidad operativa, asegurando que todos los cambios sean registrados y se puedan auditar adecuadamente.

### Modelo Entidad-Relación



*Ilustración 6. Diagrama Entidad-Relación*

Este diagrama entidad-relación muestra cómo se estructuran las principales entidades y relaciones de un sistema de gestión de tickets. En el centro, los Tickets actúan como la entidad principal, conectándose con otras entidades que aportan contexto y funcionalidad al flujo de trabajo.

Las Tareas asociadas a cada ticket pueden ser asignadas o reasignadas según sea necesario, y el estado de cada tarea puede evaluarse o cambiarse conforme avanza el proceso. Los Usuarios se relacionan con los tickets mediante la creación o asignación, permitiendo que cada caso tenga un responsable específico. Además, los usuarios pueden recibir Notificaciones generadas en el sistema,



estructurado y altamente eficiente. En el centro, se encuentra la tabla de Tickets, que actúa como el núcleo de operaciones del sistema y conecta de manera estratégica con otras entidades que complementan sus funciones y amplían sus capacidades.

Cada entidad en el diagrama tiene un rol específico que contribuye a la gestión integral del sistema. Las relaciones establecidas con Usuarios permiten que los tickets sean asignados y gestionados por personas responsables, mientras que la vinculación con Departamentos organiza los tickets según áreas específicas de trabajo, asegurando un flujo ordenado de responsabilidades. La conexión con Tareas establece un registro detallado de acciones y progresos, y el StatusHistory documenta cada cambio en el estado de un ticket, ofreciendo una línea temporal de eventos que respalda la trazabilidad y el control de cada caso.

Las Notificaciones dentro del sistema garantizan que los usuarios involucrados estén informados de actualizaciones relevantes, lo que permite mantener una comunicación fluida y una coordinación efectiva. Los Comentarios asociados a cada ticket facilitan la adición de detalles y observaciones, mientras que los Attachments sirven como soporte documental para almacenar archivos importantes, lo que añade valor y contexto sin duplicar información.

El sistema también considera la administración de Dispositivos que puedan estar asociados a los tickets, creando un vínculo directo con los recursos físicos necesarios para la resolución de cada caso. Este diseño altamente normalizado prioriza la integridad y consistencia de los datos, permitiendo que cada elemento cumpla su propósito sin redundancias, manteniendo la estructura clara y adaptable a futuras necesidades.

### **Diseño de Maquetados y Corrección y Refinamiento de Maquetados**

En esta etapa del desarrollo del sistema de Helpdesk, se han definido y diseñado las vistas y funcionalidades específicas tanto para usuarios como para administradores. Esta fase se centra en la creación de interfaces claras y eficientes que se adapten a las necesidades y permisos de cada rol, garantizando

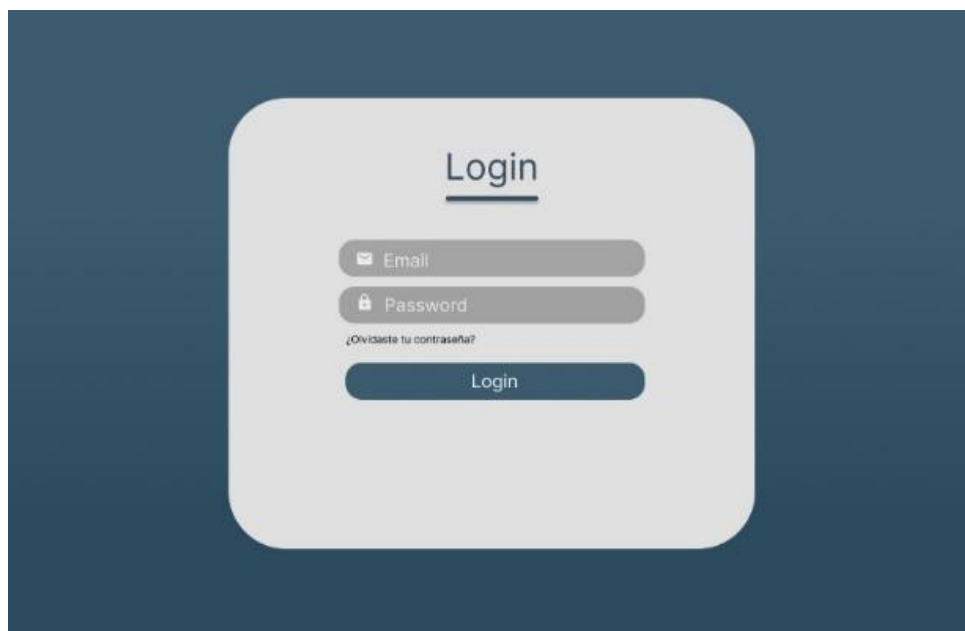
que cada usuario tenga acceso únicamente a las herramientas y opciones que le corresponden.

### **Maquetados: Vista Usuario**

Las vistas de usuario están basadas en los estados de ticket, las notificaciones ejemplificadas, inicio de sesión, y su forma de generación de ticket, que en su mayoría es lo que realizará el usuario dentro de sus funciones básicas

#### **Login**

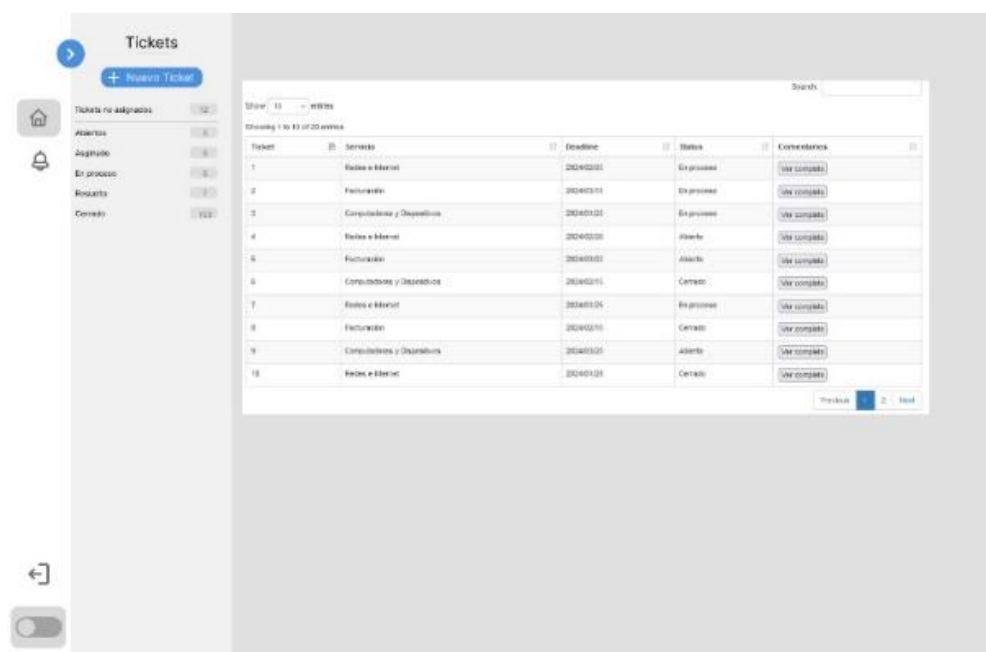
En la interfaz de inicio de sesión, se despliega un diseño sencillo y funcional que invita al usuario a ingresar sus credenciales con facilidad. Un contenedor en tonos claros resalta sobre el fondo azul oscuro, logrando un equilibrio visual que guía la atención hacia el centro de la pantalla. Dos campos, uno para el correo electrónico y otro para la contraseña, están dispuestos de manera ordenada, acompañados de pequeños iconos que facilitan su identificación. Al final, el botón de inicio de sesión, prominente y accesible, parece invitar al usuario a continuar. Justo debajo, una nota sutil ofrece la opción de recuperar la contraseña en caso de ser necesario.



*Ilustración 7. Maquetado de Interfaz de Login*

## Mis tickets

En la pantalla, se despliega una interfaz organizada para la gestión de tickets. A la izquierda, un menú lateral ofrece opciones de navegación rápidas, clasificando los tickets por estado y facilitando el acceso a las distintas categorías. En el centro, una tabla detallada muestra cada ticket con información clave, como el número de identificación, remitente, fecha de creación, estado actual y un botón de comentarios, que permite acceder rápidamente a las observaciones relacionadas. En la parte superior, un campo de búsqueda permite filtrar los tickets, optimizando la localización de información específica. El diseño promueve una experiencia intuitiva y eficiente para el usuario, permitiéndole gestionar y revisar tickets con facilidad.



Ticket	Servicio	Fecha de creación	Estado	Comentarios
1	Redes e Internet	2024/02/01	En proceso	<a href="#">Ver comentarios</a>
2	Facturación	2024/02/01	En proceso	<a href="#">Ver comentarios</a>
3	Computadores y Dispositivos	2024/02/01	En proceso	<a href="#">Ver comentarios</a>
4	Redes e Internet	2024/02/01	Atendido	<a href="#">Ver comentarios</a>
5	Facturación	2024/02/01	Atendido	<a href="#">Ver comentarios</a>
6	Computadores y Dispositivos	2024/02/01	Cerrado	<a href="#">Ver comentarios</a>
7	Redes e Internet	2024/02/01	En proceso	<a href="#">Ver comentarios</a>
8	Facturación	2024/02/01	Cerrado	<a href="#">Ver comentarios</a>
9	Computadores y Dispositivos	2024/02/01	Atendido	<a href="#">Ver comentarios</a>
10	Redes e Internet	2024/02/01	Cerrado	<a href="#">Ver comentarios</a>

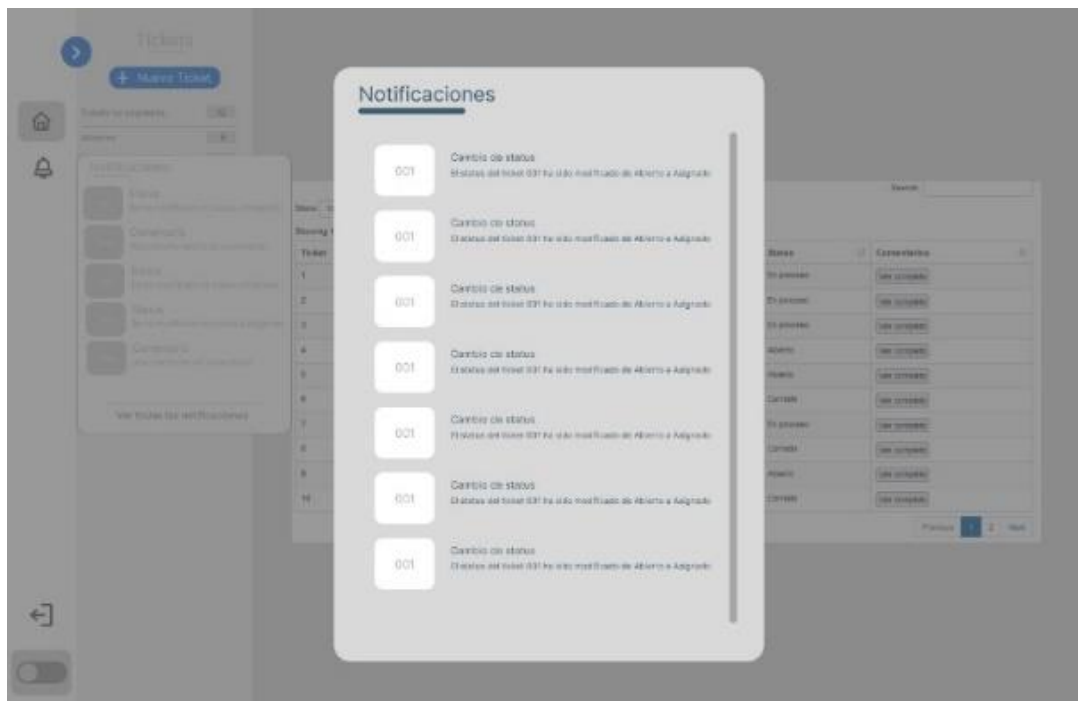
*Ilustración 8. Maquetado de Tabla de Tickets*

## Notificaciones

En esta interfaz, se presenta una ventana emergente de notificaciones diseñada para mantener al usuario informado sobre cambios y actualizaciones relevantes. La ventana, centrada sobre el fondo difuminado de la pantalla principal, destaca claramente las notificaciones recientes en una lista vertical. Cada notificación



incluye un breve mensaje descriptivo y un icono que permite identificar rápidamente el tipo de actualización. A la derecha, una barra de desplazamiento facilita la navegación por las notificaciones, permitiendo al usuario revisar múltiples mensajes sin desordenar el espacio visual. Este diseño garantiza que las alertas importantes se muestren de manera accesible y sin interrumpir las tareas en curso.

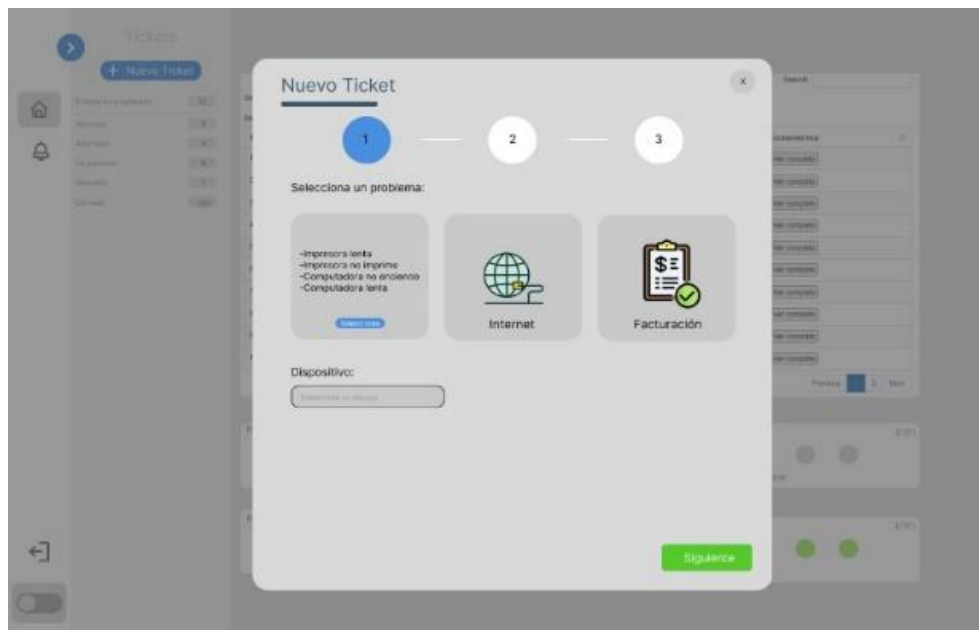


*Ilustración 9. Maquetado de Componente de Notificaciones*

## **Creación de ticket**

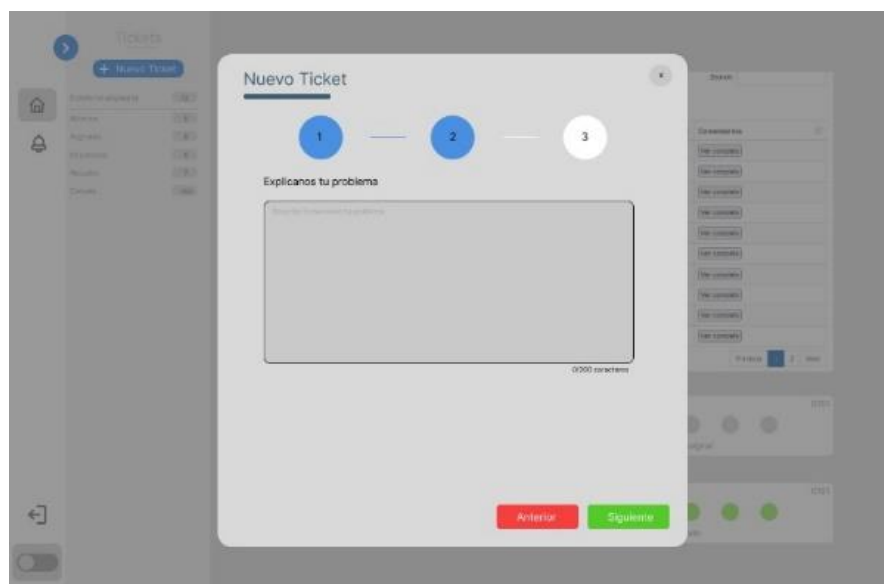
En esta interfaz de creación de tickets, el usuario es guiado paso a paso para registrar un nuevo caso. La pantalla muestra el primer paso de un proceso de tres etapas, representado visualmente por una línea de progreso en la parte superior. En esta fase, se pide al usuario seleccionar el área o problema relacionado con el ticket. Las opciones están claramente presentadas mediante iconos intuitivos y breves descripciones, facilitando la identificación rápida de la categoría adecuada. Debajo, un campo permite especificar el dispositivo relacionado, proporcionando

mayor contexto al ticket.



*Ilustración 10. Maquetado Crear Ticket 1*

El botón de Siguiente, resaltado en verde, invita al usuario a avanzar al siguiente paso, manteniendo una navegación simple y eficiente en el proceso de creación del ticket.



*Ilustración 11. Maquetado Crear Ticket 2*

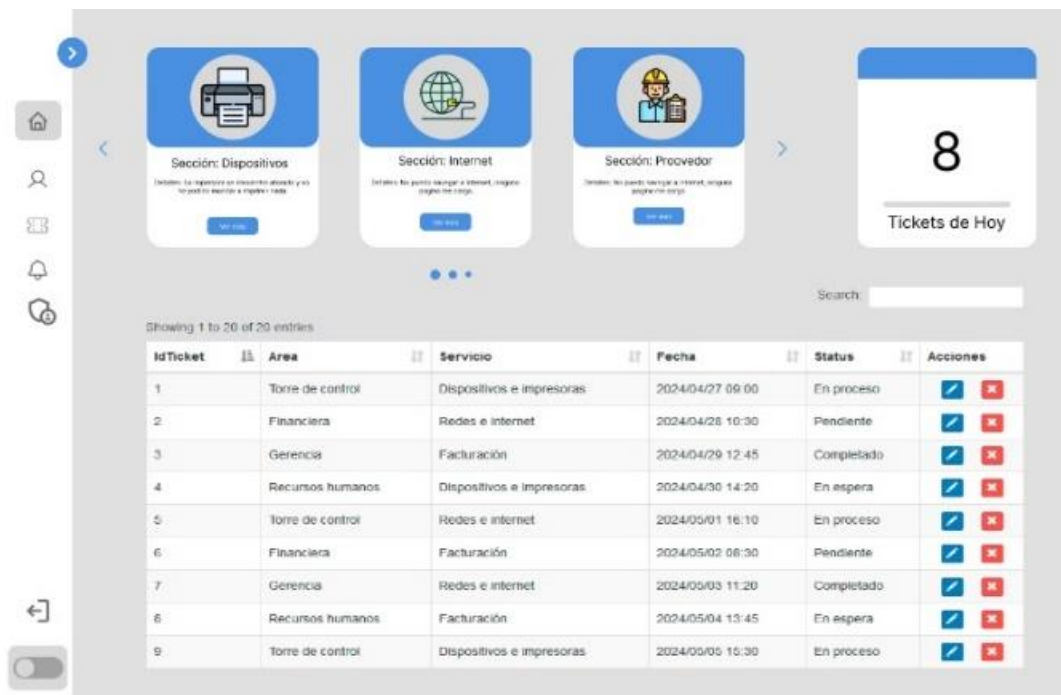
## **Maquetados: Vista Administrador**

Las vistas de administrador están orientadas a proporcionar un control completo sobre la gestión de tickets, permitiendo la creación, edición, eliminación y comentario en cada uno de ellos. Además, el administrador tiene acceso a las notificaciones relacionadas y puede gestionar usuarios, ajustando permisos y asignando responsabilidades. Estas funcionalidades avanzadas aseguran que el administrador tenga las herramientas necesarias para supervisar y optimizar el flujo de trabajo dentro del sistema de Helpdesk.

### **Dashboard Gestión de Tickets**

La interfaz ofrece una vista completa y organizada del sistema de tickets, diseñada para optimizar la supervisión y gestión de solicitudes. En la parte superior, se destacan secciones principales con tarjetas interactivas, cada una representando un área específica como Dispositivos, Internet y Procesador. Cada tarjeta permite acceder rápidamente a los tickets asociados a esas categorías, facilitando la navegación.

A la derecha, un contador muestra la cantidad de tickets generados en el día, proporcionando una visión rápida de la carga de trabajo diaria. Debajo de las tarjetas y el contador, una tabla muestra los detalles de cada ticket activo, incluyendo su número, área, servicio, fecha, estado y acciones disponibles. Los botones de acción permiten editar, ver o cerrar cada ticket de forma rápida. Este diseño intuitivo facilita al usuario el acceso a la información esencial y la gestión de cada caso de manera eficiente y ordenada.



*Ilustración 12. Dashboard Principal Administrativo*

### Creación de nuevo usuario

La pantalla de registro de un nuevo usuario se presenta en un formulario emergente, superpuesto sobre la vista principal. Los campos de entrada permiten ingresar datos esenciales como nombre, apellidos, correo, contraseña y departamento, organizando la información de forma clara y accesible. Opciones adicionales, como la selección de agencia y roles específicos, permiten ajustar los detalles según los requisitos de cada usuario. En la parte inferior, botones de acción permiten guardar, cancelar o guardar y continuar, brindando flexibilidad en el proceso de registro. El diseño es intuitivo y facilita una creación de usuario rápida y eficiente.

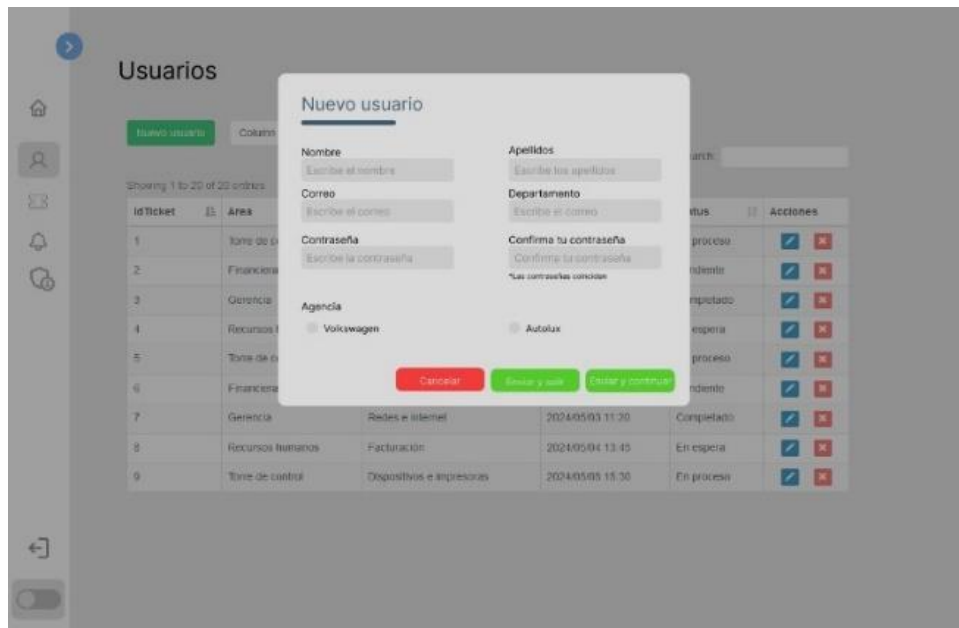


Ilustración 13. Maquetado: Creación de Nuevo Usuario

## V. RESULTADOS O PRODUCTOS OBTENIDOS

## **VI. CONCLUSIONES.**

Escribir:

- Conclusión general.
- Aportaciones de mejora a la empresa.

## FUENTES DE CONSULTA

Escribir:

Utilizar el formato APA (<http://normasapa.com/>) para las citas y las referencias. Ordenarlas de forma alfabética y emplear sangría francesa.

## **ANEXOS**

Escribir:

En este rubro se podrán presentar: los manuales, procedimientos, formatos y/o documentos mayores de dos cuartillas a los que se hace referencia en el documento, los cuales para no distraer la atención del lector se considera conveniente enviarlos al final del trabajo.



## **GLOSARIO**

Escribir:

En este apartado se incluyen las palabras técnicas empleadas en el desarrollo del proyecto, en orden alfabético. Mínimo 40 sin viñeta.