

Sistema de Tickets y Registro

Andrey Stteven Mantilla León & Daniel Esteban Pinzón Cárdenas

Unidades Tecnológicas de Santander

Nuevas tecnologías de desarrollo

Docente: M.Pm. Jesus David Arias Estupiñán

19 de noviembre de 2025

Documentación Sistema de Gestión de Tickets

Introducción

El Sistema de Gestión de Tickets es una aplicación web diseñada para registrar, gestionar y hacer seguimiento a incidencias o solicitudes de soporte dentro de una organización.

El sistema permite a los usuarios crear tickets, asignarlos, actualizarlos y cerrarlos, manteniendo un historial completo de interacciones.

El objetivo principal es optimizar la gestión de incidencias y mejorar la comunicación entre los usuarios y el equipo técnico mediante un sistema centralizado, seguro y escalable.

Arquitectura del Sistema

Descripción General

El sistema se basa en una arquitectura cliente-servidor desacoplada, donde el frontend (React) consume una API REST desarrollada en Flask (Python).

Ambos componentes se ejecutan dentro de contenedores Docker y se comunican mediante una red interna definida en Docker Compose.









El almacenamiento de datos se gestiona a través de PostgreSQL, con soporte para migraciones y configuración flexible.

Arquitectura Lógica

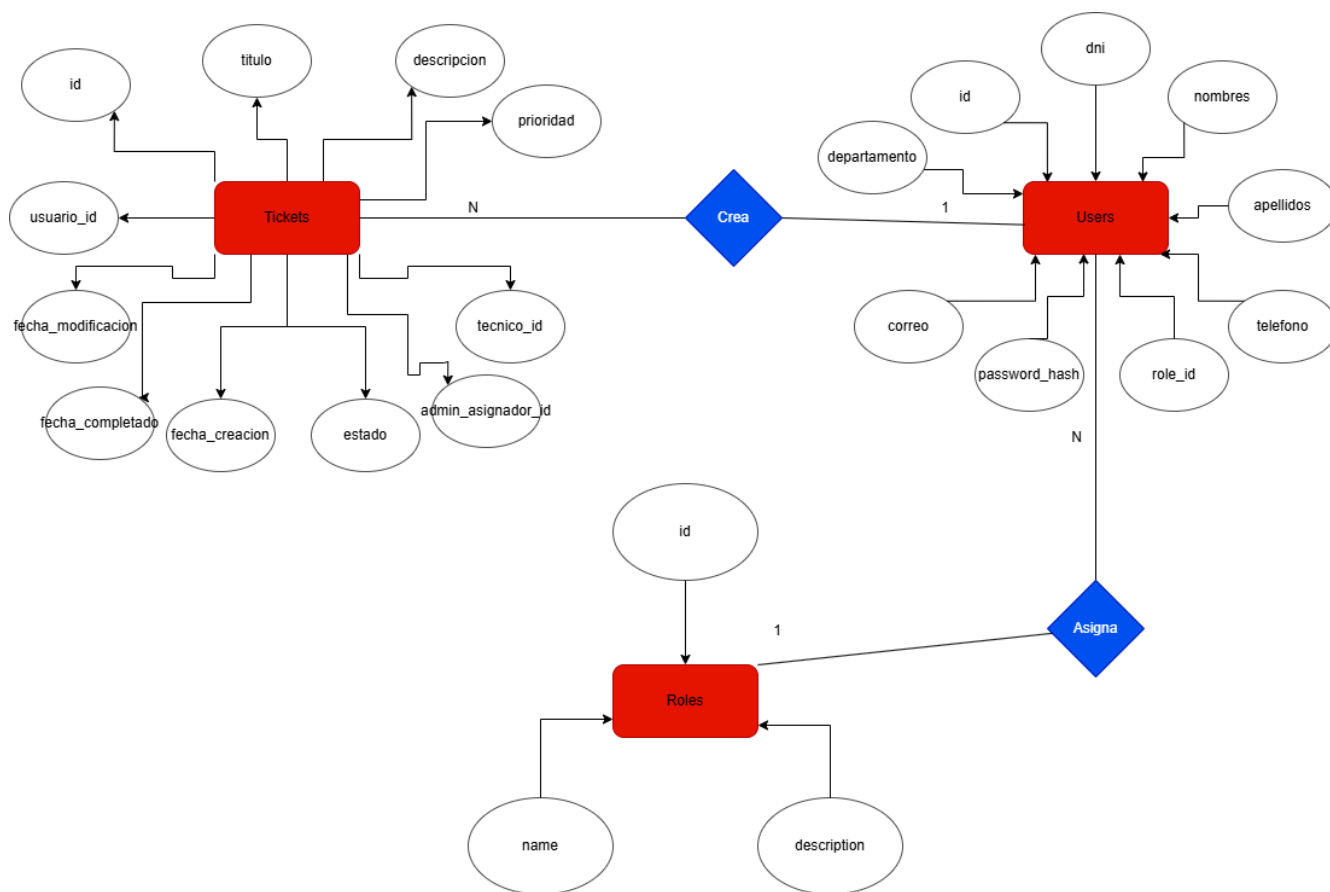
- **Frontend (React + TypeScript + Vite):**

¡Lo hacemos posible! Interfaz dinámica y responsiva.

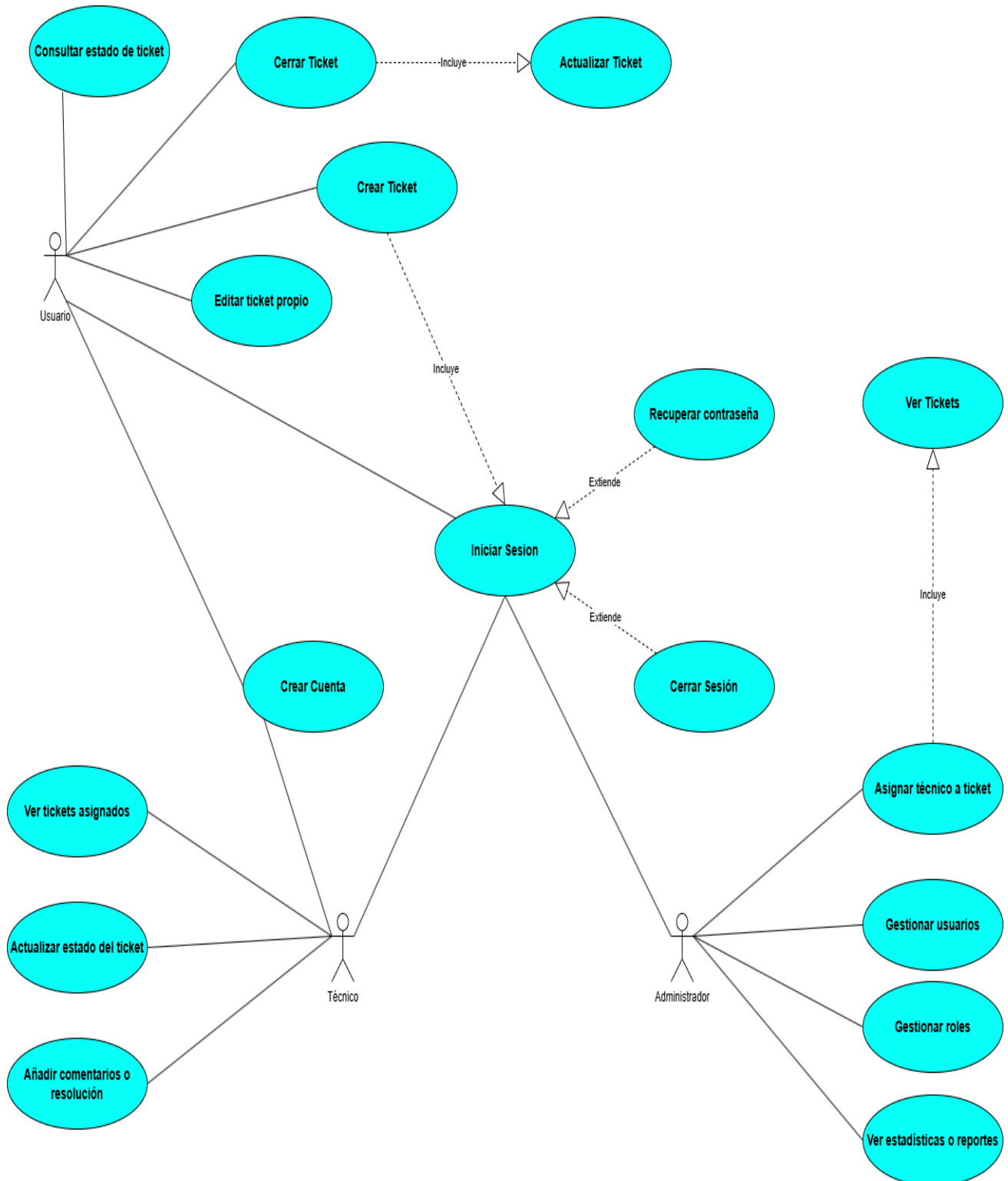
- Comunicación con la API mediante Axios.
 - Gestión de autenticación y sesión con JWT.
 - Componentes reutilizables y tipados con TypeScript.
- **Backend (Flask + Python):**
 - Implementa la API REST principal.
 - Controladores para usuarios, autenticación y tickets (MVC).
 - Uso de SQLAlchemy para modelos.
 - Manejo de seguridad con JWT y validación de dato.
- **Base de Datos (MySQL):**
 - Almacenamiento relacional de usuarios, roles, tickets y comentarios.
 - Despliegue con Clever Cloud de manera rápida y sencilla.
- **Infraestructura:**
 - Contenedores Docker para cada servicio.
 - Variables de entorno gestionadas desde .env.

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	angry_fermat	2d17c946c49e	tickets-front	8080:80	N/A	30 minutes ago	   
<input type="checkbox"/>	xenodochial_lichterman	4200bdde25f6	sistema-tickets	8000:8000	N/A	30 minutes ago	   

Diagramas UML



Caso de uso

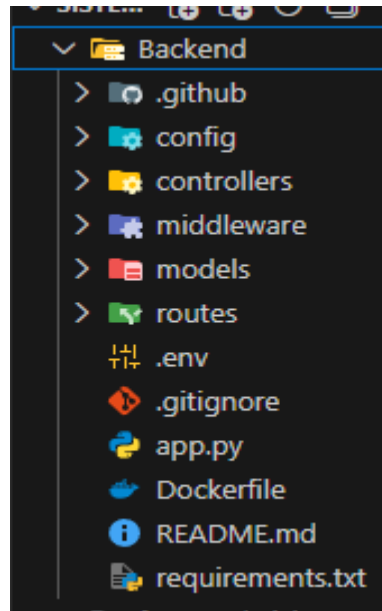


Justificación de Tecnologías Utilizadas

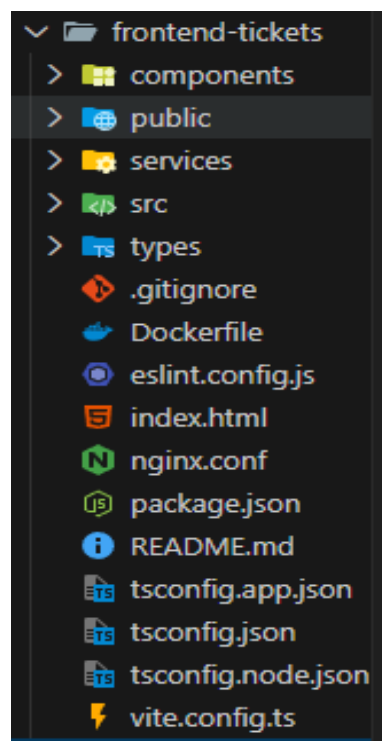
Componente	Tecnología	Justificación
Backend	Python + Flask	Framework ligero y modular para construir APIs REST. Facilita integración con SQLAlchemy, JWT y testing.
Frontend	React + TypeScript + Vite	React permite una UI dinámica y modular. TypeScript mejora la calidad del código y Vite acelera el desarrollo.
Base de Datos	PostgreSQL	Base relacional robusta, soporta transacciones, JSONB, índices avanzados y escalabilidad.
Servidor Web	Render + Vercel	Alto rendimiento y eficiencia para servir contenido de manera sencilla y por medio de GitHub.
Contenedores	Docker + Docker Compose	Facilita la portabilidad, aislamiento y despliegue del sistema en cualquier entorno.
Autenticación	JWT (JSON Web Tokens)	Proporciona un método seguro, stateless y escalable de autenticación.
ORM	SQLAlchemy	Permite manejar la base de datos de forma abstracta, segura y mantenible.
Versionamiento	Git + GitHub	Control de versiones y colaboración entre desarrolladores.
CI/CD	GitHub Actions	Automatiza pruebas, construcción de imágenes y despliegue.

Estructura del proyecto

Backend



Frontend



Manual de Instalación y Uso

Instalación y uso

- Url del sistema: <https://ticketssystem.vercel.app/>
- API: <https://api-tickets-lynz.onrender.com/>

Requisitos Previos

- Docker y Docker Compose instalados (para uso local).
- Git instalado.
- Puerto 8080 (frontend) y 8000 (backend) disponibles.
- Buena Conexión a internet.

Uso del Sistema

Inicio de sesión:

- Accede a la pantalla principal e ingresa con tu usuario y contraseña.
- El sistema generará un token JWT y mostrará el panel de tickets.
- En la parte superior derecha se encuentran los botones de Iniciar sesión para login y registro para crear los usuarios que por defecto es usuario normal.

Bienvenido al Sistema de Tickets

Gestiona y da seguimiento a tus tickets de soporte técnico

Para Usuarios

Crea tickets de soporte y da seguimiento a tus solicitudes

Para Técnicos

Gestiona tickets asignados y actualiza su estado

Para Administradores

Gestiona usuarios y asigna tickets a técnicos

Registrarse



DNI:

Teléfono:

Nombres:

Apellidos:

Departamento:

Correo Electrónico:

Contraseña:

Creación de Ticket:

- Haz clic en “Nuevo Ticket”.
- Completa los campos: título, descripción, prioridad.
- Envía y verifica que aparezca en la lista.

The screenshot shows a modal window titled "Nuevo Ticket" with a close button (X) in the top right corner. The form contains three input fields: "Título:" (Title), "Descripción:" (Description), and "Prioridad:" (Priority). The "Prioridad:" field is a dropdown menu currently set to "Media". At the bottom right of the modal are two buttons: "Crear Ticket" (Create Ticket) and "Cancelar" (Cancel). A red arrow points to the "Crear Ticket" button. In the background, a blurred view of the "Mis Tickets" page is visible, showing a list of tickets and a "Nuevo Ticket" button in the top right corner.

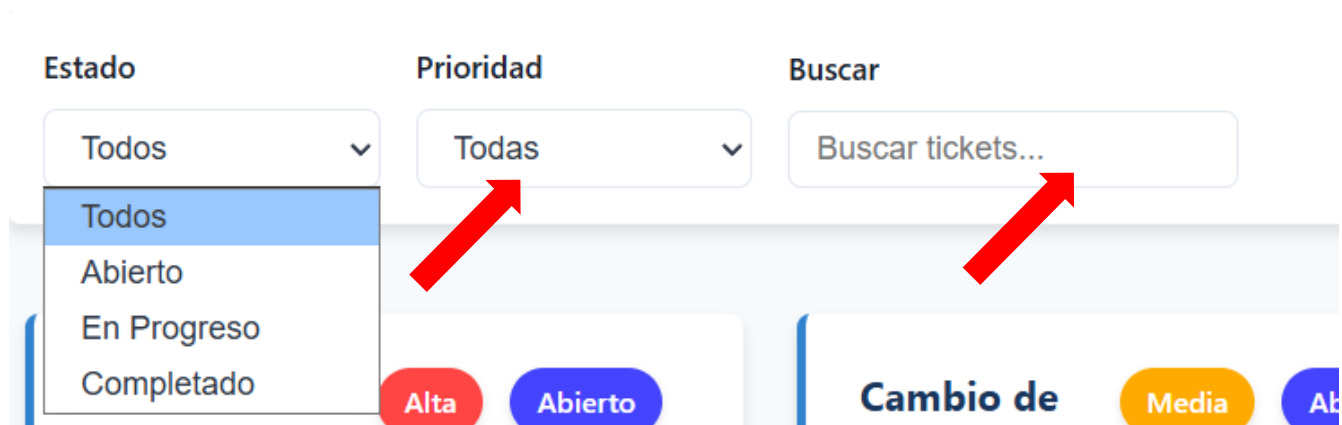
The screenshot shows the "Mis Tickets" page. At the top right, there is a blue button labeled "Nuevo Ticket". Below this, there is a search bar with the text "Buscar tickets...". The main content area displays two tickets. The first ticket is titled "Daño en el PC" with a status of "Alta" (High) and "Abierto" (Open). The second ticket is titled "Cambio de contraseña" with a status of "Media" (Medium) and "Abierto" (Open). Each ticket card includes the creator's name "Juan Diego Rodriguez" and the date "30/10/2025". At the bottom of each card are "Editar" (Edit) and "Eliminar" (Delete) buttons. A red arrow points to the "Nuevo Ticket" button in the top right corner.

Gestión de Tickets:

- Los usuarios pueden filtrar por estado, prioridad o fecha.
- Los administradores pueden asignar o cerrar tickets.

Comentarios y seguimiento:

- Cada ticket admite comentarios que se registran cronológicamente.



Conclusiones

- El Sistema de Gestión de Tickets ofrece una arquitectura moderna, modular y escalable.
- La integración entre Flask y React garantiza una separación clara entre la lógica de negocio y la interfaz de usuario.
- Gracias a Docker, la aplicación puede desplegarse fácilmente en distintos entornos sin complicaciones de configuración.
- Las tecnologías seleccionadas ofrecen un equilibrio entre simplicidad, seguridad y rendimiento, haciendo que el sistema sea adaptable a distintos escenarios empresariales.