



Universidad Internacional San Isidro Labrador

Proyecto final programado

Facultad de Ingeniería de Sistemas

Estudiante:

Michel Vargas Lira

Nombre del proyecto:

Sistema de Solicitudes de Soporte para la empresa Coopesana

RL

Contenido

Tabla de Figuras.....	3
Tabla de Diagramas	3
Justificación.....	4
Objetivo principal.....	4
Objetivos específicos.....	4
Requerimientos.....	4
Alcance	5
Diseño del Sistema	6
Funcionalidades clave:	6
Vistas principales de la aplicación web.....	8
1. Página principal:.....	8
2. Login:.....	9
3. Dashboard usuario standard:.....	9
4. Pagina con formulario para solicitud nueva:	10
5. Gestión de solicitudes para administrador.	11
6. Página de contacto:.....	11
Desarrollo del proyecto.....	12
Tecnologías Utilizadas	12
Motor y gestor de base de datos	12
Justificación Técnica	12
Diagrama de la Base de Datos (DER).....	13
Relación entre entidades	14
Requerimientos Implementados y Pendientes.....	14
Código de Conexión a la Base de Datos	15
Conclusión del Sistema <i>ServiciosTI</i>	16
Link Sistema Funcional.....	16

Tabla de Figuras

Figura no. 1: Vista principal.....	8
Figura no. 2: Vista de Login	9
Figura no. 3: Vista Dashboard	9
Figura no. 4: Vista de formulario de solicitudes	10
Figura no. 5: Vista de gestión de solicitudes.....	11
Figura no. 6: Vista página de contacto	11
Figura no. 7: Diagrama de entidad relación.....	14

Tabla de Diagramas

Diagrama no. 1: Estructura del proyecto ServiciosTI/	6
--	---

Justificación

La empresa Coopesana es una cogestionaria de servicios de salud contratada por la CCSS para brindar el servicio de ebais en 3 áreas, Santa Ana, Escazú, y San Francisco de 2 Ríos. En la empresa se cuenta con un departamento de TI el cual se encarga de brindar soporte a todo el equipo de las 3 áreas. Actualmente las solicitudes se reciben por medio del chat interno de la empresa, pero este método no lleva un control adecuado y no se cuenta con un historial del trabajo realizado. Anteriormente se probó la utilización de una etiquetera llamada Itop pero fue rechazada por gran parte del personal por ser demasiado compleja, teniendo demasiadas opciones y causando grandes dificultades tomando en cuenta que se trabaja con citas de pacientes y se debe trabajar con rapidez.

Objetivo principal

Creación de una aplicación web que cumpla con la función de llevar el control de las peticiones de soporte de los usuarios de una manera ordenada y simple.

Objetivos específicos

- Diseñar una base de datos relacional que permita almacenar y gestionar eficientemente las peticiones de soporte, usuarios, estados de solicitud y respuestas.
- Implementar un sistema de autenticación de usuarios, diferenciando entre usuarios comunes y personal de soporte para garantizar el acceso seguro y controlado a las funcionalidades del sistema.
- Desarrollar una interfaz web intuitiva y responsiva, que facilite a los usuarios registrar, consultar y hacer seguimiento de sus solicitudes de soporte.
- Incorporar funcionalidades para la gestión de solicitudes, como creación, edición, cierre y cambio de estado de cada petición por parte del personal autorizado.

Requerimientos

Requerimientos funcionales

- **Registro y autenticación:**
 - La responsabilidad de crear o modificar usuarios se reserva exclusivamente a los miembros del equipo de TI debido a las necesidades de seguridad de la empresa.
 - El sistema debe permitir a los usuarios autenticarse con un nombre de usuario y contraseña el cual se le entregara su jefatura.
 - El sistema debe diferenciar entre tipo de usuario estándar y personal de TI.
- **Gestión de solicitudes:**
 - El usuario debe poder crear solicitudes con los siguientes datos, Título, descripción, Departamento del usuario y nivel de urgencia.
 - El usuario debe poder ver el estado en el que se encuentra su solicitud.
 - Los técnicos informáticos deben poder cambiar los estados de las solicitudes y cerrarlos si es el caso.
 - Los técnicos informáticos podrán agregar comentarios y respuestas a las solicitudes.
- **Búsqueda de solicitudes:**
 - Los técnicos informáticos deben poder hacer búsqueda de las solicitudes por nivel de urgencia, palabras clave, fecha u otros.

Requerimientos no funcionales

- **Usabilidad:**
 - La interfaz debe ser clara y sencilla para evitar confusión en el usuario.
 - La aplicación web tiene que ser accesible desde cualquier computadora adentro del dominio de la empresa.
- **Rendimiento:**
 - Se busca un rendimiento optimo con la premisa de que en la empresa se trabaja con tiempos estrictos de ejecución.
- **Seguridad:**
 - Las contraseñas deben ser almacenadas cifradas.
 - Solo el personal autorizado puede ver o cerrar las solicitudes de otros usuarios.

Alcance

En el desarrollo de este proyecto, se aplicarán los conocimientos adquiridos a lo largo de mi formación académica, especialmente en áreas clave como bases de datos relacionales, utilizando PostgreSQL para el diseño y gestión de la información; desarrollo backend con Java en conjunto con el framework Spring Boot para la construcción de una API robusta y segura; y desarrollo frontend empleando HTML, CSS y JavaScript para crear una interfaz web clara e intuitiva.

Adicionalmente, se hará uso de herramientas complementarias como Postman, para realizar pruebas y validaciones de los servicios web, y Git, para el control de versiones y seguimiento del avance del proyecto.

Es importante destacar que, si bien este proyecto se desarrolla con fines educativos y será considerado como una versión beta, existe la posibilidad de que sea implementado en un entorno real dentro de la empresa, siempre y cuando reciba una aceptación favorable por parte de los usuarios y cumpla con los requerimientos funcionales establecidos.

Diseño del Sistema

Funcionalidades clave:

Para usuarios standard:

1. Página principal donde se puede tener acceso a, Iniciar sesión, contactar al equipo de TI, ver información del departamento de TI, ver noticias o eventos importantes de la empresa.
2. Página de Login donde puede ingresar su usuario y contraseña (Debe ser administrada previamente por el equipo de TI.)
3. Dashboard de usuario donde además de las opciones que teníamos en la página principal tiene la posibilidad de ver el estado de las solicitudes realizadas y un botón para agregar nuevas.
4. Pagina con formulario para crear una nueva solicitud de soporte.

Para Administradores:

1. Página principal donde se puede tener acceso a, Iniciar sesión, contactar al equipo de TI, ver información del departamento de TI, ver noticias o eventos importantes de la empresa.
2. Página de Login donde puede ingresar su usuario y contraseña.
3. Dashboard de administrador donde además de las opciones de la página principal se cuenta con un botón para administrar usuarios, para crear informes de TI, ver todas las solicitudes de soporte.
4. Página de administración de solicitudes donde puede crear, editar o dar por concluidas las solicitudes,
5. Página de administración de usuarios donde puede crear, editar, o eliminar usuarios de la plataforma.

```
└─ src/
  │ └─ main/
    │ │ └─ java/
      │ │ │ └─ com/
        │ │ │ │ └─ ServiciosTI/
          │ │ │ │ │ └─ ServiciosTI/
            │ │ │ │ │ │ └─ ServicioTIApplication.java
              │ │ │ │ │ │ └─ controller/
                │ │ │ │ │ │ └─ model/
                  │ │ │ │ │ │ └─ repository/
                    │ │ │ │ │ │ └─ service/
                      │ │ │ │ │ │ └─ resources/
                        │ │ │ │ │ │ └─ static/
                          │ │ │ │ │ │ │ └─ css/
                            │ │ │ │ │ │ │ │ └─ estilos.css
                              │ │ │ │ │ │ │ │ └─ img/
                                │ │ │ │ │ │ │ │ │ └─ Coopesana.png
                                  │ │ │ │ │ │ │ │ │ └─ index.html
                                    │ │ │ │ │ │ │ │ │ └─ views/
                                      │ │ │ │ │ │ │ │ │ │ └─ login.html
                                        │ │ │ │ │ │ │ │ │ │ └─ dashboardtecnico.html
                                          │ │ │ │ │ │ │ │ │ │ └─ dashboardusuario.html
                                            │ │ │ │ │ │ │ │ │ │ └─ mantenimientousuario.html
                                              │ │ │ │ │ │ │ │ │ │ └─ solicitudes.html
                                                │ │ │ │ │ │ │ │ │ │ └─ nuevasolicitud.html
```

```

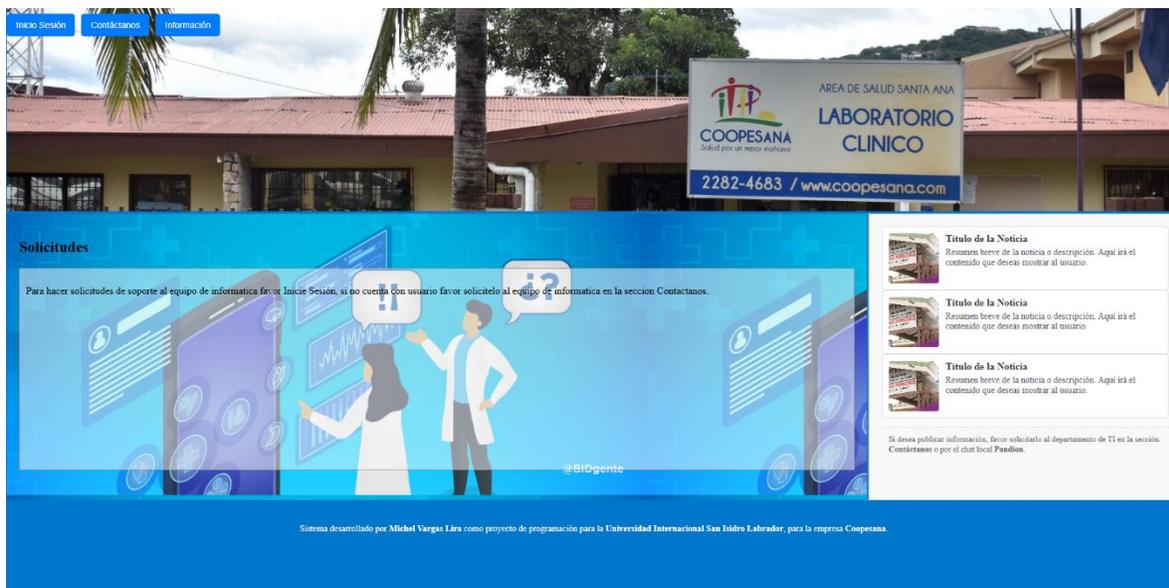
| | | | └─ contacto.html
| | | | └─ registro.html
| | | └─ application.properties
| └─ test/
|   └─ java/
|     └─ com/
|       └─ ServiciosTI /
|         └─ ServiciosTI /
|           └─ ServiciosTI ApplicationTests.java

```

Vistas principales de la aplicación web

1. Página principal:

Figura no. 1: Vista principal



La página da la Bienvenida al sistema, muestra algunas noticias relevantes y notifica a el usuario que debe iniciar sesión si necesita abrir una solicitud de soporte.

2. Login:

Figura no. 2: Vista de Login



Servicios TI Coopesana
Iniciar Sesión

Usuario

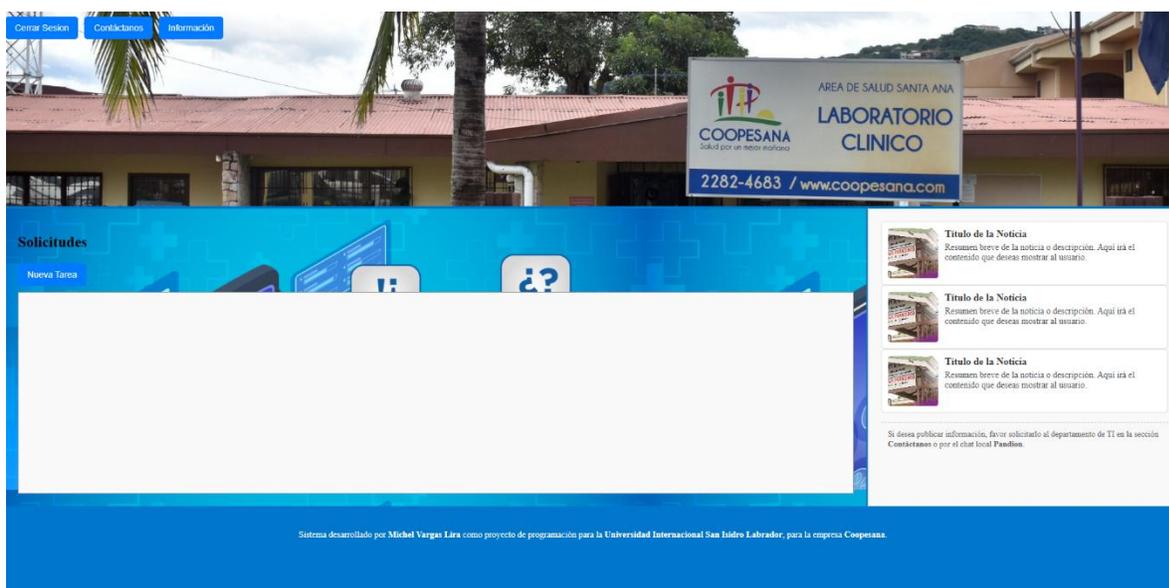
Contraseña

Ingresar

Es una página sencilla donde se solicita usuario y contraseña para ingresar al sistema. Desde esta página se va redirigir a 2 diferentes dashboards dependiendo del tipo de usuario que ingresa, administrador o usuario Standard.

3. Dashboard usuario standard:

Figura no. 3: Vista Dashboard



Inicio Sesión Configuración Información

AREA DE SALUD SANTA ANA
LABORATORIO CLINICO
COOPESANA Salud por un mejor mañana
2282-4683 / www.coopesana.com

Solicitudes
Nueva Tarea

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desea mostrar al usuario.

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desea mostrar al usuario.

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desea mostrar al usuario.

Si desea publicar información, favor solicitarlo al departamento de TI en la sección Contactanos o por el chat local Pandion.

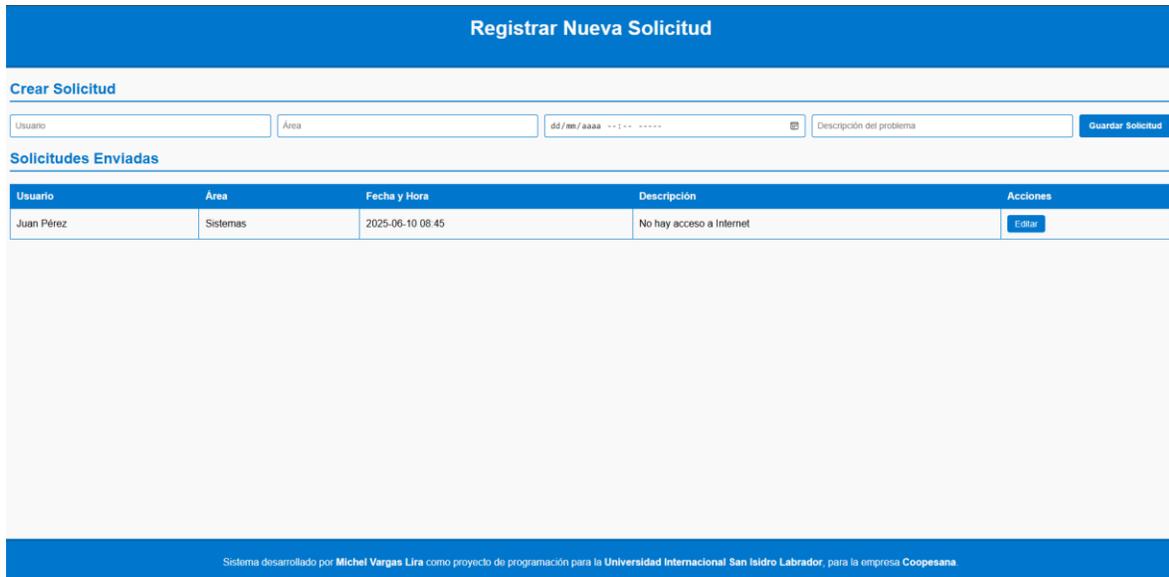
Sistema desarrollado por Michel Vargas Lira como proyecto de programación para la Universidad Internacional San Eloy Labrador, para la empresa Coopesana.

En esta página ya se le permite al usuario ver las solicitudes hechas con anterioridad, ver en qué estado se encuentran y sus soluciones, o también agregar una nueva solicitud.

El botón nueva tarea nos va redirigir a la página con el formulario para crear una nueva tarea o solicitud de soporte.

4. Pagina con formulario para solicitud nueva:

Figura no. 4: Vista de formulario de solicitudes



Usuario	Área	Fecha y Hora	Descripción	Acciones
Juan Pérez	Sistemas	2025-06-10 08:45	No hay acceso a Internet	Editar

En esta página el usuario puede ingresar los diferentes datos para crear la tarea nueva.

5. Gestión de solicitudes para administrador.

Figura no. 5: Vista de gestión de solicitudes

Gestión de Solicitudes de Soporte TI

Crear Solicitud

Usuario Área dd/mm/aaaa --:--:-- Descripción del problema Técnico asignado (opcional) Solución (si aplica)

Solicitudes Registradas

Usuario	Área	Fecha y Hora	Descripción	Técnico	Solución	Acciones
Maria López	Contabilidad	2025-06-10 10:30	No imprime la impresora	Pedro Sánchez	Se reinstaló el driver	<input type="button" value="Editar"/> <input type="button" value="Cerrar"/>

Sistema desarrollado por Michel Vargas Lira como proyecto de programación para la Universidad Internacional San Isidro Labrador, para la empresa Coopesana

En esta página el administrador o técnico puede revisar todas las solicitudes registradas y además puede eliminarlas, editarlas, cerrarlas o crear nuevas.

6. Página de contacto:

Figura no. 6: Vista página de contacto

Inicio Sesión



Contacto

Sede Santa Ana
Correo: coopesainf@coopesana.com
Tel: 22222222

Sede Escazú
Correo: coopesainf@coopesana.com
Tel: 22222222

Sede San Francisco-San Antonio
Correo: coopesainf@coopesana.com
Tel: 22222222

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desees mostrar al usuario.

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desees mostrar al usuario.

Título de la Noticia
Resumen breve de la noticia o descripción. Aquí irá el contenido que desees mostrar al usuario.

Si desea publicar información, favor solicitarlo al departamento de TI en la sección, Costarricense o por el chat local Preadmin.

Sistema desarrollado por Michel Vargas Lira como proyecto de programación para la Universidad Internacional San Isidro Labrador, para la empresa Coopesana.

En esta página aparece la información de contacto de los departamentos de informática de las 3 áreas de la empresa de manera independiente.

Desarrollo del proyecto

Tecnologías Utilizadas

Para el desarrollo del sistema de gestión de solicitudes del departamento de TI se emplearon tecnologías modernas que garantizan robustez, seguridad, mantenibilidad y escalabilidad. A continuación, se detallan las principales herramientas utilizadas:

Lenguaje de programación y entorno

- Java 17: Lenguaje principal para la lógica del backend, elegido por su robustez, seguridad y amplia adopción en entornos empresariales.
- Spring Boot: Framework que simplifica la creación de aplicaciones Java, promoviendo una arquitectura basada en REST y principios como la inyección de dependencias.

Librerías y herramientas adicionales

- Spring Security + JWT: Implementación de seguridad mediante autenticación basada en tokens (JWT), evitando el uso de sesiones tradicionales.
- Hibernate (JPA): Capa de persistencia para interactuar con la base de datos mediante mapeo objeto-relacional (ORM).
- Lombok: Herramienta que reduce el código repetitivo al generar automáticamente métodos como getters, setters y constructores.

Frontend

- HTML, CSS y JavaScript puro: Lenguajes estándar para construir una interfaz ligera, fácilmente mantenible y compatible con la mayoría de los navegadores modernos.

Motor y gestor de base de datos

- PostgreSQL: Sistema de gestión de bases de datos relacional, confiable y potente, con soporte para integridad referencial, concurrencia y consultas complejas.

Justificación Técnica

La selección tecnológica del sistema responde a criterios de estabilidad, escalabilidad, buenas prácticas y experiencia en entornos de producción:

- Java + Spring Boot: Su popularidad y madurez en el entorno empresarial garantiza soporte a largo plazo, buena documentación y comunidad activa. Spring Boot facilita la creación de APIs REST escalables y organizadas.
- JWT con Spring Security: Brinda autenticación robusta sin depender de sesiones de servidor, siguiendo principios modernos de seguridad en aplicaciones web.

- Hibernate (JPA): Permite interactuar con la base de datos sin escribir SQL directamente, gracias a su integración con Spring.
- PostgreSQL: Elegido por ser una base de datos libre, altamente confiable, con excelentes capacidades de escalabilidad y rendimiento.
- Frontend simple: Se optó por HTML, CSS y JS puro para evitar dependencias innecesarias, manteniendo el control total del código y asegurando compatibilidad.

Diagrama de la Base de Datos (DER)

A continuación, se presenta el diagrama entidad-relación que representa la estructura lógica del sistema. Se incluyen las entidades, atributos, claves primarias, claves foráneas y tipo de relación entre ellas.

Entidades principales

Usuarios

- **Descripción:** Contiene información de los usuarios del sistema, tanto administradores como usuarios estándar.
- **Campos:**
 - id (PK): Identificador único.
 - nombre: Nombre completo del usuario.
 - username: Nombre de usuario único.
 - password: Contraseña cifrada.
 - email: Correo electrónico del usuario.
 - rol: Tipo de usuario (ADMIN, USUARIO).
 - departamento: Área a la que pertenece.

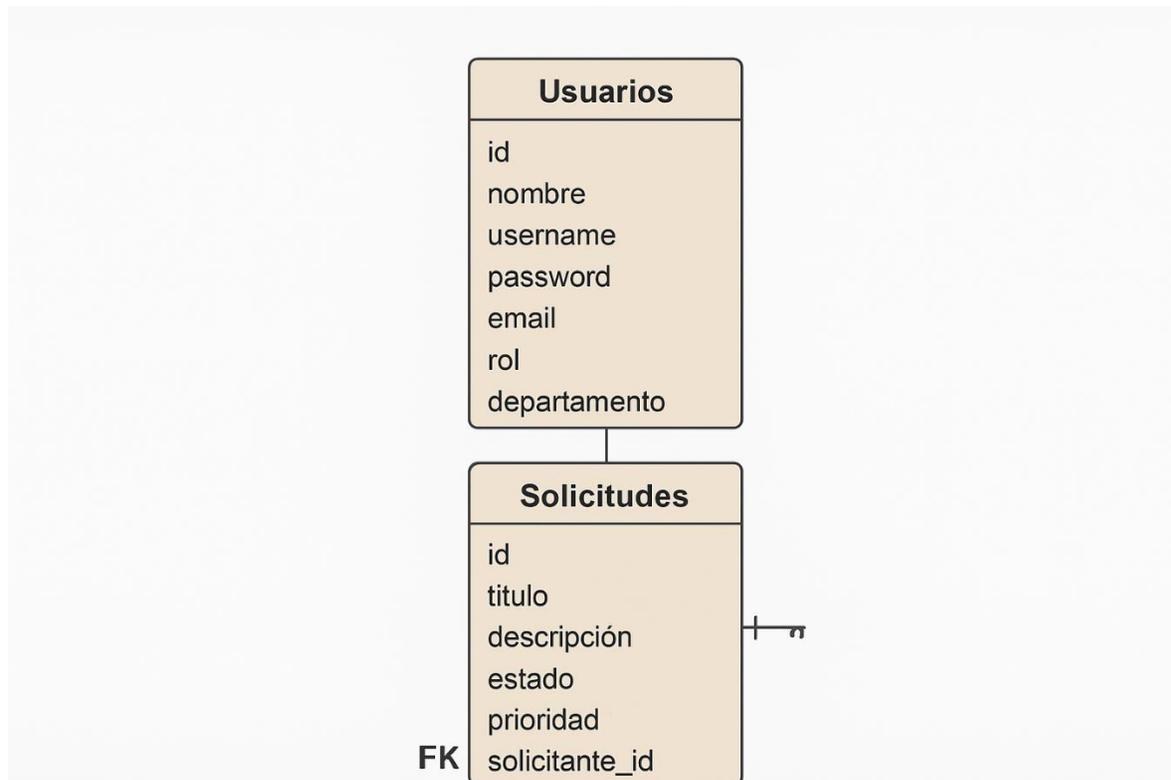
Solicitudes

- **Descripción:** Representa las solicitudes realizadas por los usuarios al departamento de TI.
- **Campos:**
 - id (PK): Identificador único de la solicitud.
 - titulo: Breve descripción del requerimiento.
 - descripcion: Detalle del problema.
 - estado: Estado actual (NUEVO, EN PROCESO, SOLUCIONADO).
 - prioridad: Nivel de importancia (ALTA, MEDIA, BAJA).
 - creadoEn: Fecha y hora de creación.
 - finalizadoEn: Fecha y hora de cierre (puede ser nulo).
 - solicitante_id (FK): Usuario que generó la solicitud.

Relación entre entidades

- **Usuarios (1) --- (N) Solicitudes**
 - Tipo: Relación uno a muchos.
 - Un usuario puede realizar múltiples solicitudes, pero cada solicitud pertenece a un único usuario.

Figura no. 7: Diagrama de entidad relación



Requerimientos Implementados y Pendientes

Para usuarios estándar

Implementados:

- Página principal con acceso a login, contacto, información del departamento y noticias.
- Página de inicio de sesión.
- Dashboard de usuario para ver solicitudes y crear nuevas.
- Formulario para crear solicitudes de soporte.

Pendientes:

- Mejoras visuales y de diseño.

- Mejorar el orden de visualización de solicitudes.

Para administradores

Implementados:

- Página principal con las mismas opciones que un usuario estándar.
- Dashboard de administrador con acceso a gestión de usuarios y solicitudes.
- Página de administración de solicitudes con funciones de creación, edición y finalización.
- Página de gestión de usuarios con opción para crear nuevos usuarios.

Pendientes:

- Mejoras visuales y de diseño.
- Implementar funciones de edición y eliminación de usuarios.

Código de Conexión a la Base de Datos

La conexión con PostgreSQL se define en el archivo `application.properties`, ubicado en `src/main/resources`. Este archivo contiene los parámetros necesarios para que Spring Boot realice la conexión automáticamente mediante inyección de dependencias.

Fragmento de `application.properties`

```
spring.application.name=ServicioTi

spring.datasource.url=jdbc:postgresql://localhost:5432/tidb
spring.datasource.username=postgres
spring.datasource.password=230993
spring.datasource.driver-class-name=org.postgresql.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
server.port=8080
logging.level.org.springframework.security=DEBUG
```

Explicación

- `spring.datasource.url`: Ruta de conexión al servidor PostgreSQL.
- `spring.datasource.username` y `password`: Credenciales de acceso.
- `ddl-auto=update`: Crea o actualiza las tablas según las entidades Java.

- `show-sql=true`: Muestra las consultas SQL en consola.
- `hibernate.dialect`: Especifica el dialecto propio de PostgreSQL.

Gracias a esta configuración, no es necesario escribir código adicional para establecer la conexión, ya que Spring Boot gestiona internamente el acceso y el pool de conexiones.

Conclusión del Sistema *ServiciosTI*

El sistema *ServiciosTI* constituye una plataforma orientada a la gestión integral de servicios tecnológicos, desarrollada con un enfoque modular que permite administrar usuarios, solicitudes y recursos de forma eficiente. A lo largo de su construcción, se implementaron tanto componentes de frontend como de backend, asegurando una comunicación fluida entre la interfaz de usuario y la lógica del sistema.

En el **backend**, se desarrollaron funcionalidades clave como el registro de usuarios, la autenticación segura mediante validaciones, la gestión de solicitudes de servicios y el almacenamiento estructurado de la información en una base de datos. Estas implementaciones garantizan que los procesos internos se ejecuten de manera ordenada, con persistencia de datos y soporte para futuras ampliaciones del sistema.

En el **frontend**, se creó una interfaz accesible y clara que facilita la interacción del usuario con las funcionalidades principales. Esto incluye formularios de registro y gestión de servicios, así como vistas que presentan la información de manera organizada y coherente con las necesidades del proyecto.

En conclusión, el proyecto *ServiciosTI* no solo cumple con los objetivos iniciales planteados, sino que sienta las bases para futuras mejoras, como la incorporación de reportes avanzados, paneles administrativos más robustos y la integración con herramientas externas. Gracias a su arquitectura, el sistema se convierte en una solución escalable y adaptable a diferentes contextos de gestión de servicios tecnológicos.

Link Sistema Funcional

Link a video explicativo: https://drive.google.com/file/d/1h-bsaRDno6X6C2k71pdbkO_pNk5G15tl/view?usp=sharing