

# REDVITAL

---

TU TIENES LA PIEZA QUE FALTA

Pilar Bourg, Olivia Gallego, Elena Morales,  
Alejandra O'Shea, Marta Sánchez y Josefina Silva

# 1 ¿Qué problema resolvemos?

La donación de sangre necesita:

- Mejor **coordinación** entre donantes y hospitales
- Información clara sobre **disponibilidad** y **centros cercanos**
- Un **sistema seguro** de citas
- **Comunicación inmediata** ante urgencias de sangre

# 2 Objetivos del Proyecto

- 01 • Registro e inicio de sesión seguro
- 02 • Área de donante con información personalizada
- 03 • Gestión de notificaciones
- 04 • Reserva de citas
- 05 • Localización de centros cercanos
- 06 • Gestión por parte de hospitales y administradores

# 3 Arquitectura del Sistema

## 01 • Interaccion del usuario

El usuario navega y realiza acciones desde la interfaz web (frontend).

## 02 • Lógica del frontend

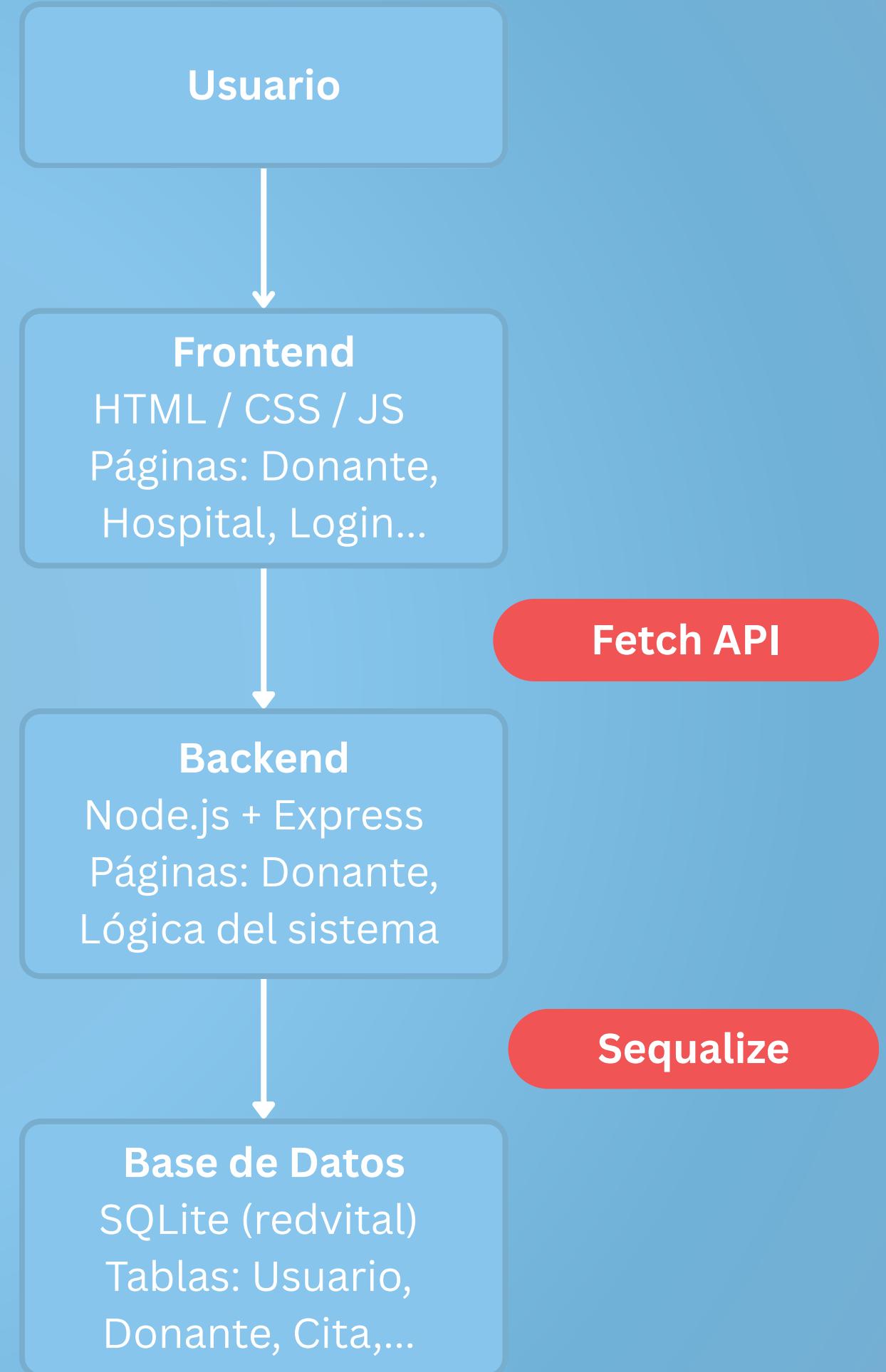
Las páginas usan JS + Fetch API para comunicarse con el backend sin recargar.

## 03 • Backend y API REST

Node.js + Express gestionan toda la lógica: login, citas, donantes, hospitales...

## 04 • Persistencia de Datos

Sequelize maneja tablas y relaciones: Usuario, Donante, Cita, Hospital...



# 4 Base de Datos

## Modelos

- **Usuario** – login, email, contraseña, rol, dirección, teléfono
- **Donante** – datos personales y clínicos, condiciones, grupo sanguíneo
- **Hospital** – información del centro, dirección
- **Cita** – fecha, hospital, donante
- **Donación** – registro de una donación realizada
- **Notificación** – alertas enviadas al donante
- **Solicitud** – solicitudes de sangre emitidas por hospitales
- **Inventario de sangre** – cantidades por tipo sanguíneo en cada hospital

## Relaciones

### **Usuario 1–1 Donante**

→ Cada usuario puede tener un único perfil de donante.

### **Hospital 1–n Solicitudes**

→ Un hospital puede pedir muchas solicitudes de sangre.

### **Solicitud 1–n Donaciones**

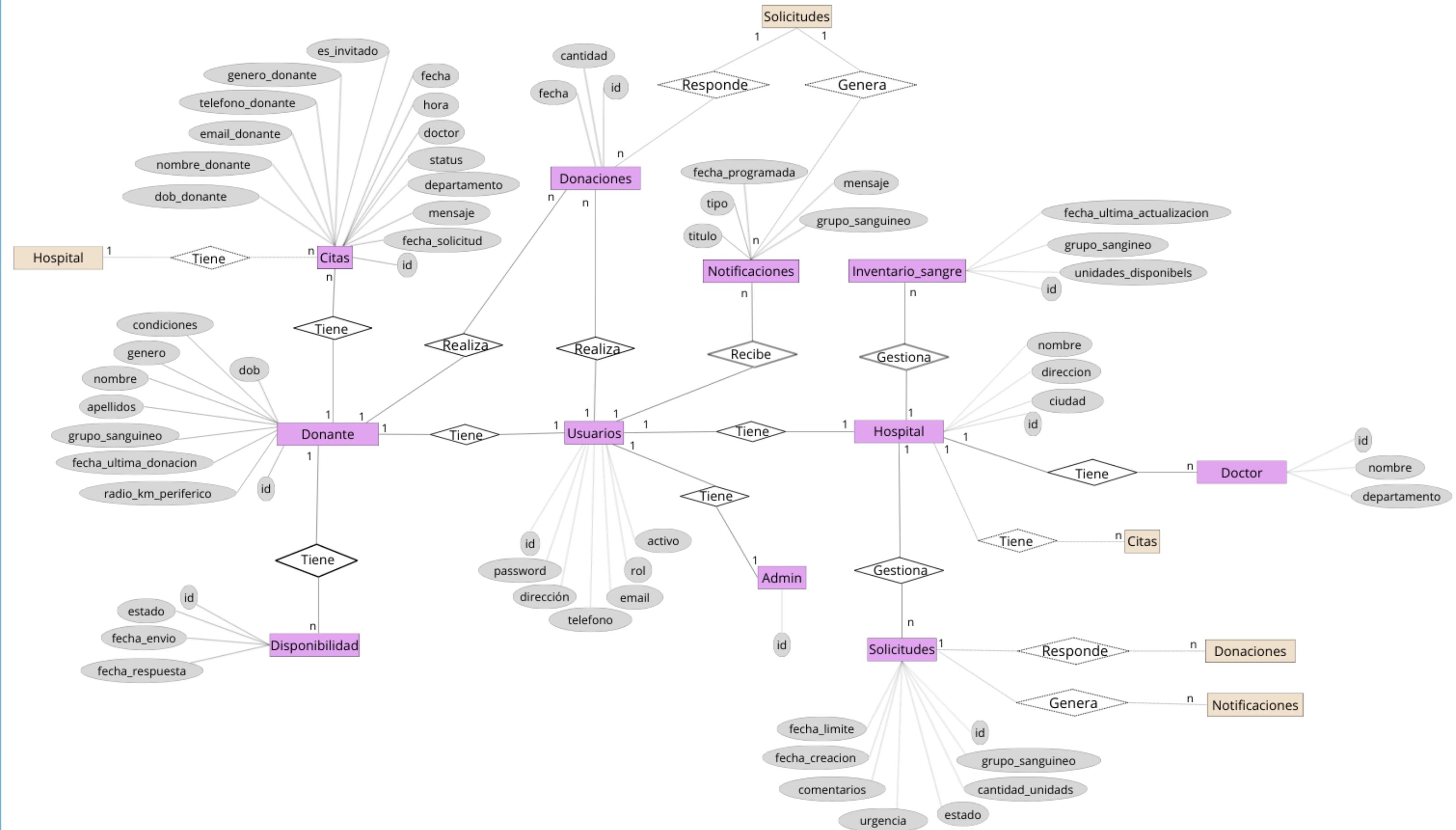
→ Una solicitud puede ser satisfecha con varias donaciones.

### **Usuario 1–n Notificaciones**

→ Un donante recibe múltiples alertas de urgencia.

### **Hospital 1–n InventarioSangre**

→ Cada hospital mantiene varios tipos de sangre en stock.



# 5 Funcionalidades

## Frontend

- Login y registro
- Página de donante con:
  - Perfil
  - Notificaciones
  - Historial
  - Centros cercanos  
(geolocalización)
  - Configuración
- Sistema de navegación suave entre secciones

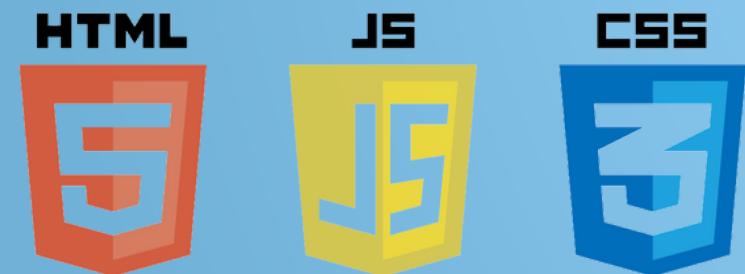
## Backend

- Rutas para donantes, citas, hospitales, notificaciones...
- Cálculo de distancia a hospitales con API Nominatim
- Autenticación con hash (bcrypt)
- Respuesta JSON limpia y consistente

# 6 Tecnologías Utilizadas

## Frontend

- HTML5
- CSS3
- JavaScript



## Backend

- Node.js
- Express.js
- Sequelize ORM
- SQLite
- Bcrypt
- Fetch API externa para geocodificación



express



# 7 Mejoras Futuras

**Sistema Nacional / Global de Interconexión de Bancos de Sangre.** Conectar hospitales y centros de donación en tiempo real, permitiendo:

- Compartir stock entre ciudades/países
- Evitar escasez regional
- Optimizar rutas de envío mediante IA

*Esto lo usan países como Japón o Canadá en sus redes nacionales de donación.*

---

## IA para Predicción de Necesidades de Sangre

- Un sistema de Machine Learning que prediga:
- Picos de demanda (accidentes, cirugías, festividades...)
- Tendencias estacionales
- Riesgo de escasez para cada tipo sanguíneo

*Similar a lo que usa la Cruz Roja Americana.*

---

## Algoritmos de optimización logística para transporte de sangre

- Integración con:
- Sistemas de drones médicos
- Gestión de frío en transporte
- Selección de rutas más rápidas según tráfico

*Muy usado ya en Suiza y Ruanda para entrega de sangre mediante drones.*

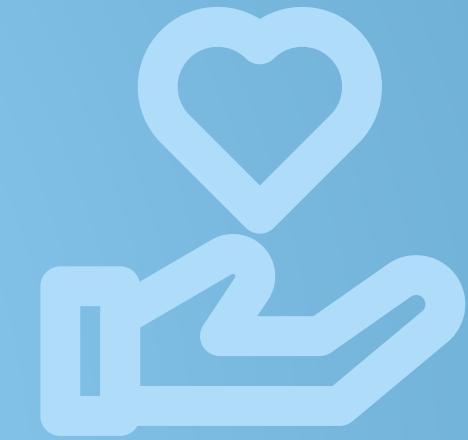
# 8 Conclusiones



Mejora la  
comunicación  
donante-  
hospital



Centraliza citas,  
notificaciones y  
datos médicos



Automatiza  
procesos

# Gracias

---

REDVITAL

Pilar Bourg, Olivia Gallego, Elena Morales,  
Alejandra O'Shea, Marta Sánchez y Josefina Silva