

Plan de Trabajo: Primeras 4 Semanas

Objetivo general al final de la semana 4

Tener un sistema funcional con:

- Autenticación básica (login/logout)
- CRUD de estudiantes
- Base de datos PostgreSQL configurada
- Interfaz web con listado y formulario de estudiantes
- Pruebas automatizadas básicas

Semana 1: Fundamentos y entorno

Objetivo: Configurar el entorno, definir la arquitectura y crear el modelo de datos.

Día	Actividades
Lunes	<div>- Crear repositorio en GitHub</div> <div>- Definir estructura de carpetas (`backend/`, `frontend/`, `db/`, etc.)</div> <div>- Crear `README.md` y `docs/CONTRIBUTING.md`</div>
Martes	<div>- Diseñar modelo de base de datos (DER → SQL)</div> <div>- Crear script `db/init.sql` con tablas: `estudiantes`, `países`, `localidades`, `generos`, etc.</div>
Miércoles	<div>- Configurar backend básico con Node.js + Express + TypeScript</div> <div>- Configurar conexión a PostgreSQL (con Prisma o TypeORM)</div>
Jueves	<div>- Configurar frontend básico con Vite + React + TypeScript + Tailwind CSS</div>

Día	Actividades
	- Probar conexión entre frontend y backend local
Viernes	- Definir endpoints iniciales de API (ej. <code>GET /api/health</code>) - Reunión de equipo: validar arquitectura y repartir tareas semana 2

✓ **Entregable semana 1:** Repositorio funcional, BD creada, servidores backend/frontend levantados.

Semana 2: Autenticación y gestión de usuarios

Objetivo: Implementar login, roles y protección de rutas.

Actividad	Responsable	Detalles
Diseñar modelo de usuario y roles	Backend	Tabla <code>usuarios</code> con <code>email</code> , <code>password_hash</code> , <code>rol</code>
Implementar registro/login (JWT)	Backend	Usar <code>bcrypt</code> y <code>jsonwebtoken</code>
Crear formulario de login	Frontend	Validación de campos, manejo de errores
Proteger rutas en backend	Backend	Middleware de autenticación
Proteger rutas en frontend	Frontend	Redirección si no está autenticado
Escribir pruebas de autenticación	QA	Login válido/inválido, token expirado

✓ **Entregable semana 2:** Sistema con login funcional y rutas protegidas.

Semana 3: CRUD de Estudiantes (Backend)

Objetivo: API completa para gestionar estudiantes.

Actividad	Responsable
Implementar endpoints CRUD (GET, POST, PUT, DELETE /estudiantes)	Backend
Validar entradas (DNI, email, teléfono, etc.)	Backend
Manejo de errores centralizado	Backend
Escribir tests unitarios e integración	QA + Backend
Documentar API (en docs/ o con Swagger)	Backend

✓ **Entregable semana 3:** API REST funcional y testeada para estudiantes.

Semana 4: Interfaz de Estudiantes (Frontend + Integración)

Objetivo: Pantallas funcionales conectadas a la API.

Actividad	Responsable
Diseñar layout base (navbar, sidebar, footer)	Frontend
Crear página de listado de estudiantes	Frontend
Crear formulario de alta/edición de estudiante	Frontend
Consumir API de estudiantes (Axios/fetch)	Frontend
Manejar estados de carga y errores	Frontend
Pruebas manuales de flujo completo	QA
Revisión de código y merge a main	Todo el equipo

✓ **Entregable semana 4:** Aplicación web funcional para gestionar estudiantes.