

Librería	Uso
next	Framework principal React/SSR
react / react-dom	Base de React
@prisma/client	Cliente ORM
prisma	CLI para migraciones
next-auth	Autenticación y sesiones
@auth/prisma-adapter	Adaptador NextAuth ↔ Prisma
bcrypt	Hasdeo de contraseñas
react-hot-toast	Notificaciones visuales
tailwindcss / postcss / autoprefixer	Estilos modernos
typescript	Tipado

Ejecuta una vez:

```
npm install next react react-dom @prisma/client prisma next-auth
@auth/prisma-adapter bcrypt react-hot-toast tailwindcss postcss
autoprefixer typescript
```

Avances

Entregable	Descripción	Estado
1. Acta de Constitución del Proyecto	Documento base con objetivos, justificación, alcance, cronograma e hitos.	Entregado
2. Documento de Requerimientos de Software (DRS)	Identificación de requerimientos funcionales, no funcionales, restricciones y riesgos iniciales.	Entregado
3. Modelo 4 + 1 (Despliegue, Física y Lógica)	Diagramas que describen la arquitectura técnica y lógica del sistema UniPost.	En progreso (3 vistas listas)
4. Configuración de entorno de desarrollo	Instalación y conexión de Next.js, Prisma, PostgreSQL (Supabase) y TailwindCSS.	Completado
5. Integración de base de datos (Supabase + Prisma)	Configuración y conexión estable del <i>Session Pooler</i> en entorno local, con migraciones funcionales.	Completado
6. Backend API funcional	Endpoints <code>/api/posts</code> , <code>/api/variants/[id]/publish</code> y <code>/api/auth</code> operativos.	Completado
7. Frontend MVP (Composer)	Interfaz en Next.js que permite crear, listar y publicar variantes de posts.	Completado
8. Mejoras UI/UX	Rediseño del panel de publicaciones con estética moderna y colores del branding UniPost.	Implementado
9. Control de versiones (Git + GitHub)	Repositorio remoto activo con commits y control de versiones.	En uso
10. Guía de despliegue técnico	Documento de instalación, dependencias y despliegue en Vercel.	Redactada

Logros técnicos

- Base de datos Supabase configurada con *Session Pooler* IPv4.

- **Prisma ORM** implementado con migraciones automáticas y modelo relacional completo ([User](#), [Organization](#), [Post](#), [Variant](#), [Schedule](#)).
- **API REST** construida en el App Router de Next.js 14.
- **Interfaz React funcional** con carga dinámica de publicaciones desde la BD.
- **Notificaciones visuales** implementadas con [react-hot-toast](#).
- **Diseño visual** modernizado con degradados, vidrio translúcido y componentes accesibles.
- **Repositorio GitHub** sincronizado y conectado a flujo CI/CD en Vercel (pendiente despliegue final).