



Documento de Arquitectura

Plataforma UniPost
Fecha: 19-10-2025

Tabla de contenido

Tabla de contenido.....	2
Historial de Versiones.....	3
Información del Proyecto.....	3
Vista Despliegue:.....	4
Vista de Procesos:.....	5
Vista Física:.....	7
Vista Lógica:.....	8
Vista de Escenarios:.....	9

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
21/10/2025	1	Víctor Nadal	Duoc UC	Definición de proyecto inicial

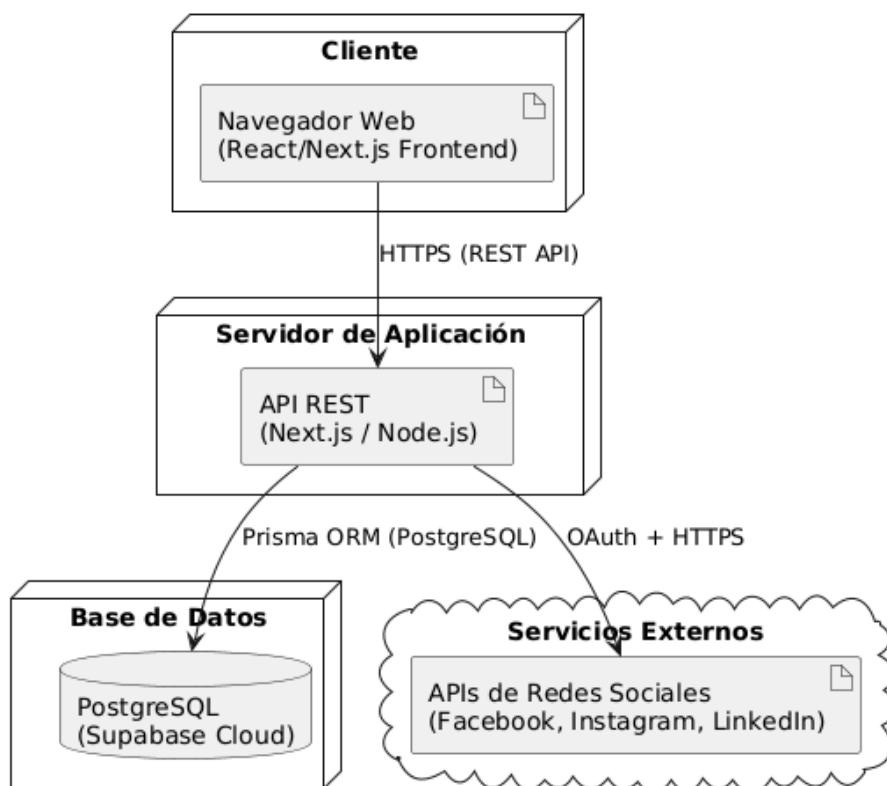
Información del Proyecto

Empresa / Organización	DuocUC
Proyecto	UniPost
Fecha de preparación	20/08/2025
Cliente	Público General
Patrocinador principal	-
Gerente / Líder de Proyecto	Víctor Nadal
Gerente / Líder de Análisis de negocio y requerimientos	Douglas Ramírez, Víctor Nadal

Vista Despliegue:

La vista de despliegue de *UniPost* muestra la distribución física del sistema en entornos de desarrollo y producción. El sistema se basa en una arquitectura **cliente–servidor**, donde la interfaz web (frontend) se comunica con una API alojada en un servidor Node.js/Next.js, que a su vez interactúa con una base de datos PostgreSQL gestionada en Supabase.

El entorno de desarrollo se ejecuta localmente, mientras que el entorno de producción se despliega en una plataforma en la nube (por ejemplo, **Vercel** o **Railway** para el backend). Los usuarios acceden mediante navegadores web desde dispositivos conectados a Internet.

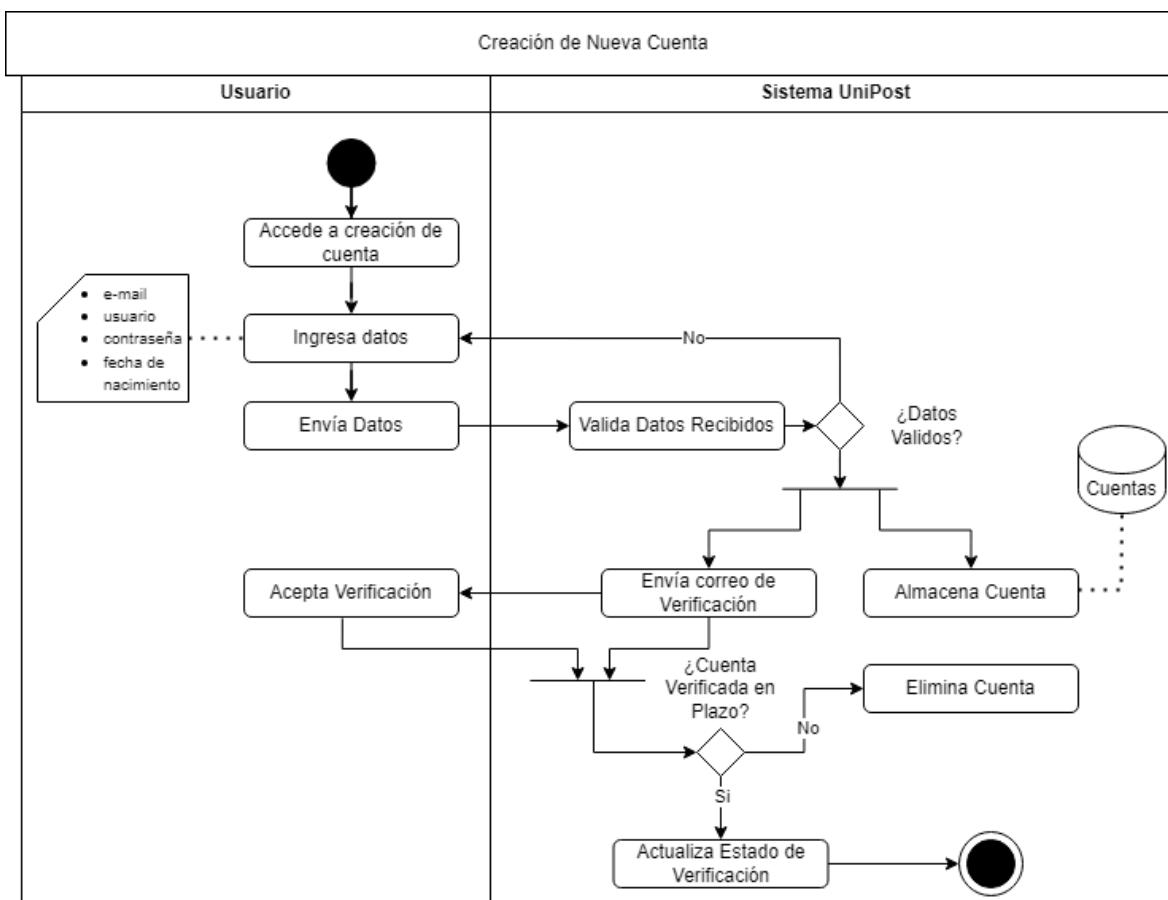


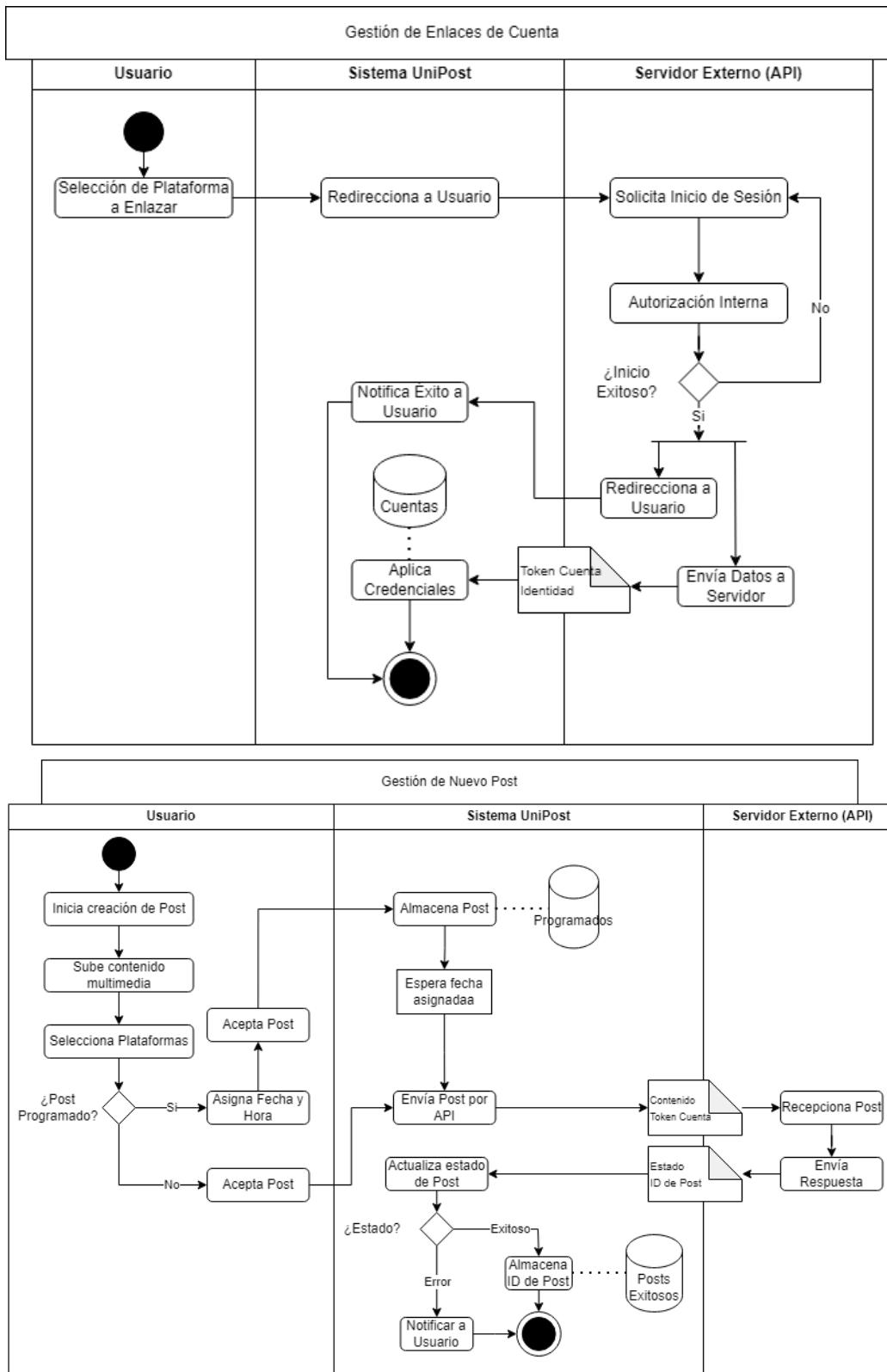
Vista de Procesos:

Los tres procesos esenciales del proyecto corresponden a:

- **Creación de Cuentas:** Diferenciación de roles y permisos.
- **Enlace de Cuentas con RRSS:** Utilizado para el envío de posts a plataformas asociado a sistema de cuentas.
- **Creación y Gestión de Posts:** Característica principal, manipulación de contenido multimedia.

El flujo de cada actividad es descrito por los siguientes diagramas:



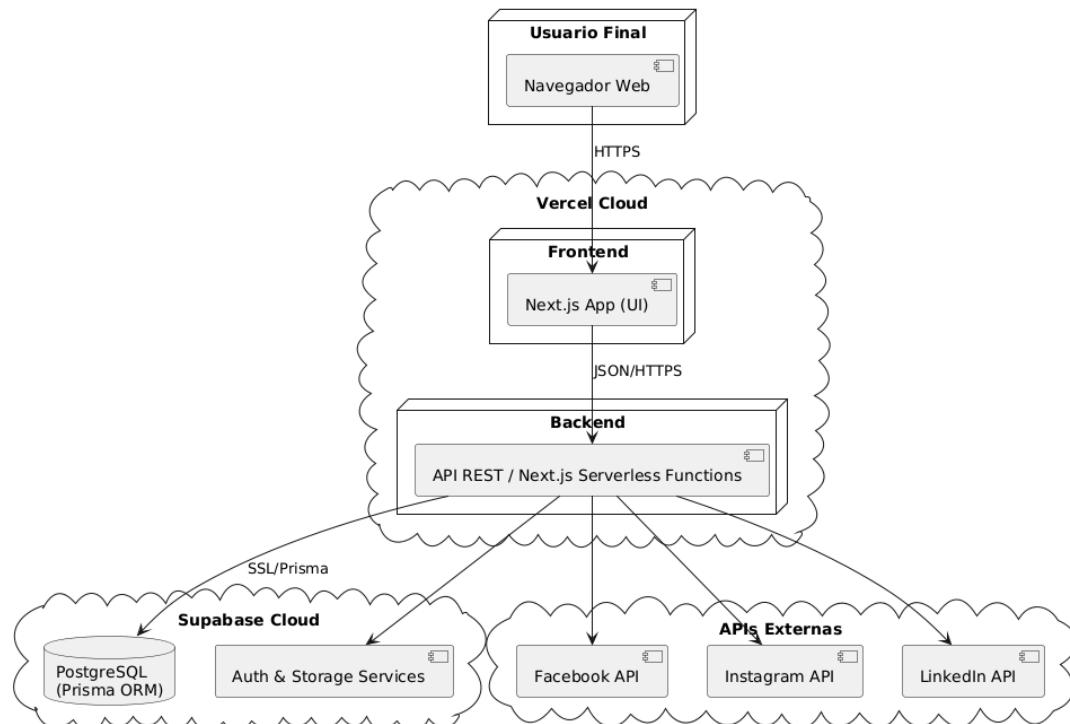


Vista Física:

En el entorno de producción, el sistema se despliega en una arquitectura basada en la nube:

- **Frontend** alojado en **Vercel**, que entrega contenido estático generado por Next.js.
- **Backend API** desplegado en el mismo entorno de Vercel (serverless functions).
- **Base de datos PostgreSQL** administrada por **Supabase**, alojada en la nube con conexiones seguras SSL.
- **Servicios externos de autenticación** y APIs de redes sociales mediante OAuth 2.0.

La comunicación entre los componentes se realiza mediante HTTPS y conexiones seguras cifradas.



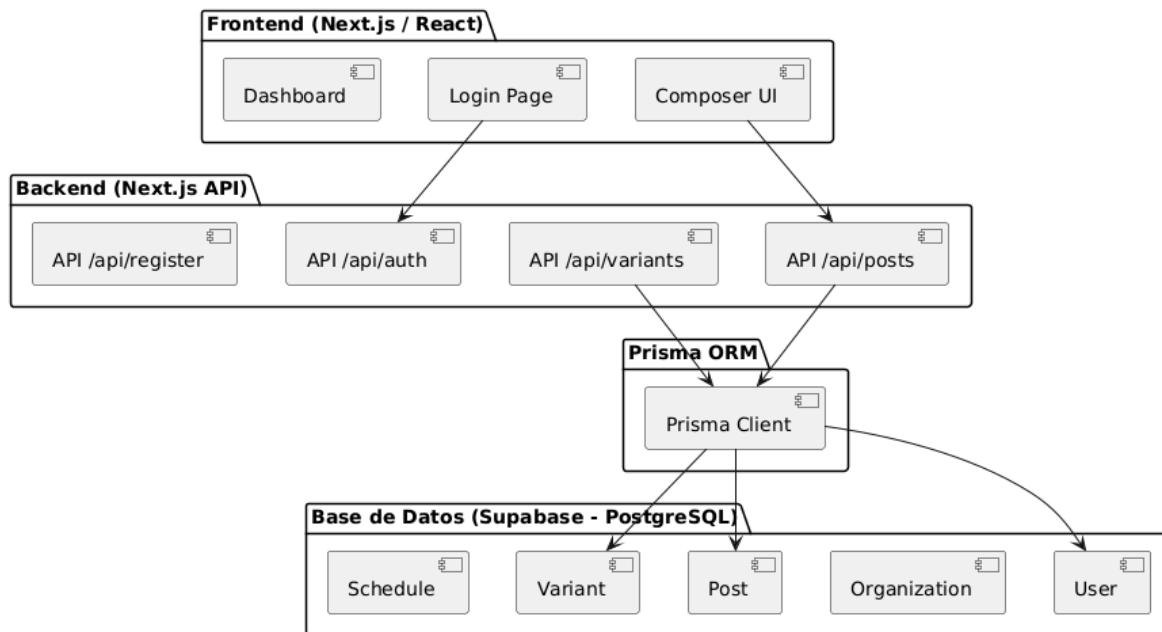
Vista Lógica:

La vista lógica de *UniPost* describe la organización de los componentes de software en capas interdependientes.

En la capa de **presentación**, se encuentra la interfaz desarrollada con *Next.js* y *React*, que se comunica con la **capa de aplicación** a través de endpoints REST implementados mediante el App Router de *Next.js*.

La **capa de aplicación** maneja la lógica de negocio, validación de datos y flujo de publicaciones, delegando la persistencia al ORM *Prisma*, que interactúa con la **base de datos PostgreSQL** alojada en *Supabase*.

Este modelo modular garantiza **mantenibilidad, escalabilidad y separación de responsabilidades**, permitiendo a *UniPost* evolucionar hacia una plataforma SaaS de gestión de contenidos multiplataforma.



Vista de Escenarios:

Al interior de la plataforma, como se contempla la implementación de roles, es necesario comprender el cómo se distribuyen.

La estructura de funciones por rol se presenta de una forma jerárquica, donde cada rol hereda las funciones y permisos de su inferior.

Las implementaciones del sistema puede ser visualizada a través del siguiente diagrama:

