

2. Tecnologías

En este capítulo se detalla las tecnologías principales utilizadas en el proyecto y la razón por la que fueron elegidas.

2.1 Backend — Laravel

Tecnologías empleadas:

- **Laravel 12**
- **PHP 8.3**
- **MySQL**
- **Laravel Sanctum** para autenticación
- **Mailtrap** para pruebas de correo

¿Por qué Laravel?

Lo elegí desde el principio porque ya tenía algo de experiencia con este framework. Conocía su flujo de trabajo, las migraciones, Eloquent y el sistema de rutas.

Esta familiaridad permitió:

- Acelerar notablemente el desarrollo de la API (endpoints funcionales en la primera semana).
- Reutilizar patrones ya conocidos (Controladores, Traits, Mailables, colas).

Durante el proyecto se confirmó que Laravel era la opción más adecuada técnicamente para los requisitos funcionales (gestión de usuarios inactivos, envío programado de correos, control de contribución por palabras, etc.).

2.2 Frontend — Vue 3 + Vite

Tecnologías empleadas

- **Vue 3 (Composition API)** — Framework principal para la construcción de la interfaz.
- **Vite** — Herramienta de desarrollo y build ultrarrápida.
- **Tailwind CSS** — Framework utilitario para estilos rápidos y responsivos.
- **Vue Router** — Navegación entre páginas sin recarga.
- **Pinia** — Gestión global del estado (sesión, usuario, autenticación).
- **Fetch API** — Utilizado para todas las peticiones al backend.

¿Por qué Vue 3?

Quería una interfaz limpia, rápida y sencilla de mantener, con una herramienta que tuviera una curva de aprendizaje moderada.

Vue 3 destaca porque:

- Tiene curva de aprendizaje suave
- La Composition API permite organizar mejor la lógica
- Reactividad muy sencilla sin necesidad de librerías adicionales
- Ecosistema muy ligero comparado con React

Además, **Vite** aporta:

- Recarga ultrarrápida en desarrollo
- Builds muy optimizadas

Y **Tailwind CSS** fue elegido porque:

- Ya lo había usado antes y me resulta muy sencillo e intuitivo
- Acelera mucho el maquetado
- Permite un estilo consistente sin escribir toneladas de CSS
- Es ideal para interfaces modernas y limpias (justo lo que buscaba para Agatha)

2.3 Herramientas de Desarrollo

Principales herramientas utilizadas:

- **Visual Studio Code** como editor principal
- **Postman** para probar la API durante el desarrollo
- **Git + GitLab** para control de versiones
- **Laragon** como servidor local rápido
- **Chrome DevTools** para depuración de frontend

2.4 Gestión de Base de Datos

Tecnología empleada:

MySQL 8

¿Por qué MySQL?

- Es estable y compatible con Laravel.
- Facilidad para definir una estructura y relaciones entre usuarios, historias, palabras y lugares.

- Perfecta para estructuras relacionales y consultas sencillas.
- Herramientas GUI como phpMyAdmin facilitan el mantenimiento.

MySQL me encajaba con el proyecto por ser este pensado para una estructura clara y estricta, donde las relaciones entre entidades son necesarias.