

# Razonamiento del Uso de Librerías en el Proyecto

## Composer

### Dependencias Principales

1. **php**: Se especifica la versión 8.2 para aprovechar la estabilidad que aporta y mejoras de rendimiento respecto a versiones anteriores, además que es la última versión disponible para usar laravel 11.
2. **inertiajs/inertia-laravel**: Facilita la creación de aplicaciones SPA con Laravel, permitiendo una mejor gestión de las rutas y estados.
3. **laravel/framework**: La base del proyecto.
4. **laravel/sanctum**: Proporciona autenticación simple para aplicaciones SPA, ideal para gestionar la seguridad de la API.
5. **tightenco/ziggy**: Permite usar las rutas de Laravel en JavaScript, facilitando la interacción entre frontend y backend.

### Dependencias de Desarrollo

1. **fakerphp/faker**: Generador de datos ficticios, útil para pruebas y desarrollo sin depender de datos reales.
2. **laravel-lang/lang**: Proporciona traducciones para Laravel, esencial para aplicaciones multilingües.
3. **laravel/breeze**: Ofrece un inicio rápido para la autenticación en Laravel, simplificando la configuración inicial.
4. **mockery/mockery**: Utilizada para crear mocks en pruebas, mejorando la calidad y efectividad de las pruebas unitarias.
5. **nunomaduro/collision**: Mejora la visualización de errores en la consola, para facilitar la depuración.
6. **pestphp/pest** y **pestphp/pest-plugin-laravel**: Proporcionan un framework de pruebas más elegante y fácil de usar para Laravel.

## NPM

### Dependencias de Desarrollo

1. **@iconify/react**: Permite integrar iconos fácilmente en la aplicación, mejorando la estética y usabilidad.
2. **@inertiajs/react**: Complemento para React que facilita la integración con Inertia.js, permitiendo una experiencia fluida y reactiva junto a laravel.

3. **@tailwindcss/forms**: Estilos predefinidos para formularios, optimizando la apariencia de la interfaz.
4. **@types/\***: Proporcionan definiciones de tipos para TypeScript, mejorando la experiencia de desarrollo y detección de errores.
5. **@vitejs/plugin-react**: Plugin que permite la integración de React en el entorno de Vite, mejorando el rendimiento de la compilación.
6. **autoprefixer, postcss**: Herramientas que optimizan y generan prefijos CSS automáticamente, asegurando compatibilidad con diferentes navegadores.
7. **laravel-vite-plugin**: Integración de Vite con Laravel, optimizando el proceso de construcción y carga de activos.
8. **react, react-dom**: Librerías fundamentales para construir interfaces de usuario con React.
9. **tailwindcss**: Framework CSS utility-first, permite un diseño altamente personalizable y responsivo.
10. **typescript**: Proporciona tipado estático en JavaScript, mejorando la calidad del código y la experiencia de desarrollo.
11. **vite**: Herramienta de construcción y desarrollo rápida, optimizando el flujo de trabajo.

## Dependencias

1. **@emotion/react**: Permite utilizar estilos en línea en componentes React, mejorando la gestión del CSS.
2. **@table-library/react-table-library y react-table**: Facilita la creación de tablas complejas en React, mejorando la gestión de datos.
3. **chart.js y react-chartjs-2**: Proporcionan herramientas para crear gráficos interactivos y visualizaciones de datos en la aplicación.

Este documento ofrece una visión general de por qué se han elegido estas librerías y cómo contribuyen al desarrollo eficiente y efectivo del proyecto. Si necesitas realizar ajustes o añadir más detalles, házmelo saber.