SOLUCIÓN A PRUEBA TÉCNICA PARA VACANTE DE DESARROLLADOR FULLSTACK.

Cristian Jiménez.
Mendoza, Argentina.

TECNOLOGÍAS USADAS:

Back End:

- PHP (7.x o posterior).
- Laravel – Breeze – Sanctum.
- MySQL.

Front End:

- Typescript
- React – React Router – Zustand, Axios.
- TailwindCSS, Shadcn UI.

FUNCIONALIDAD GENERAL:

Es una plataforma en la que se listan propiedades (inmuebles) para la venta y renta.
Con los siguientes detalles:

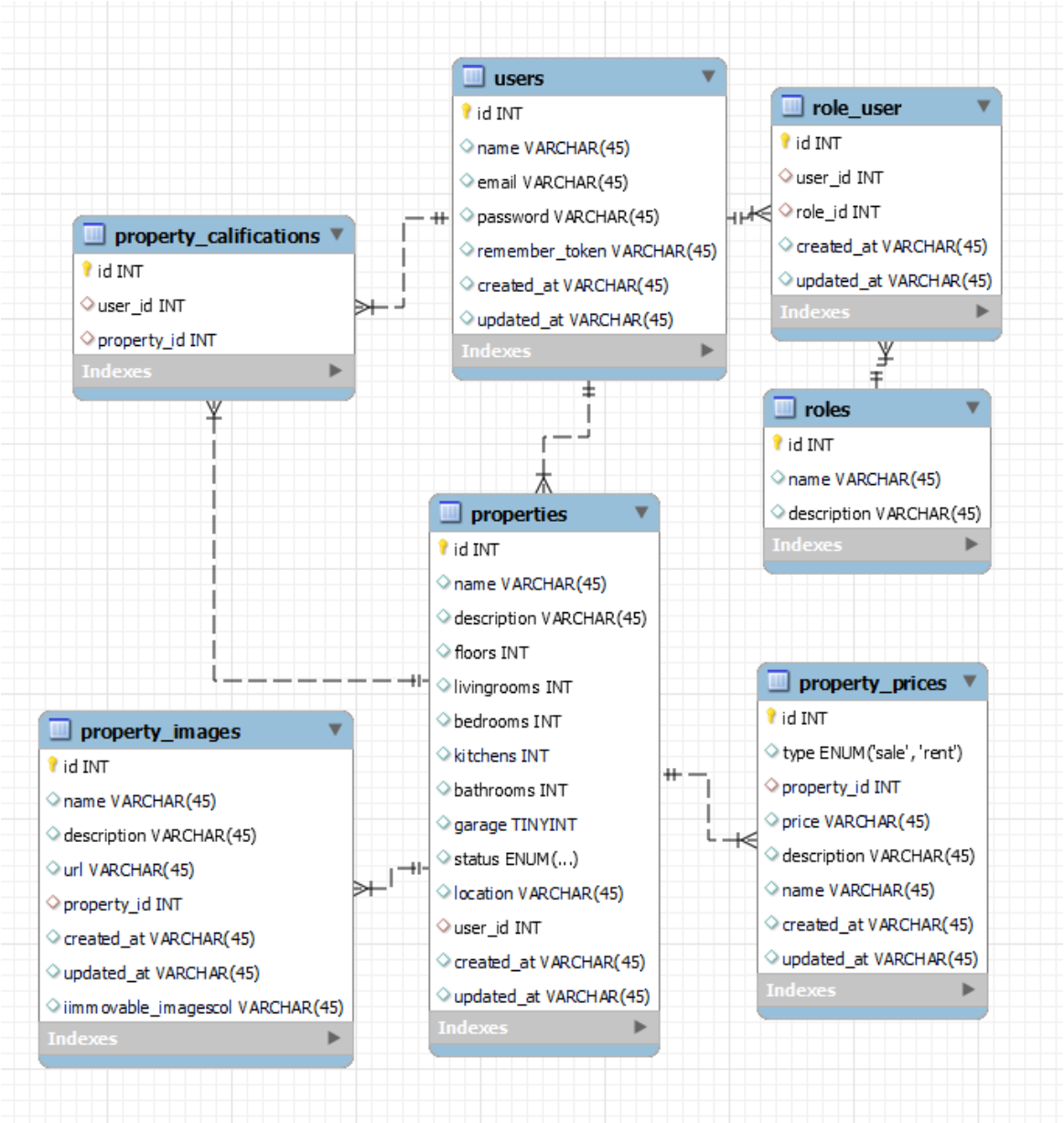
- Registro de usuario (inmobiliaria), inicio de sesión y administración de la información de la cuenta.
- Sesión de administración de inmuebles publicados (con sus correspondientes operaciones CRUD), con soft-deletes y la opción de eliminar permanentemente la propiedad publicada o restablecerla
- Sistema de calificación, cada usuario puede calificar una propiedad (mediante un sistema de estrellas).
- Se pueden agregar diferentes precios y éstos se dividen en dos categorías (venta y renta).
- Filtros de búsqueda en la home page o en la tabla de administración.
- Paginación de lado de Back End
- Se pueden cargar diversas imágenes para cada propiedad publicada.
- Estados : (vendida, disponible, rentada).
- Envío de correo de bienvenida al registrar usuario.

ESPECIFICACIONES TÉCNICAS:

BACK END:

Para el Back End utilicé Laravel 8 haciendo uso del paquete Sanctum para autenticación mediante sesiones y cookies, y también mediante tokens.

Hice el modelo entidad relación en MySQL Workbench
Y luego delegué la creación del mismo en las migraciones de Laravel:



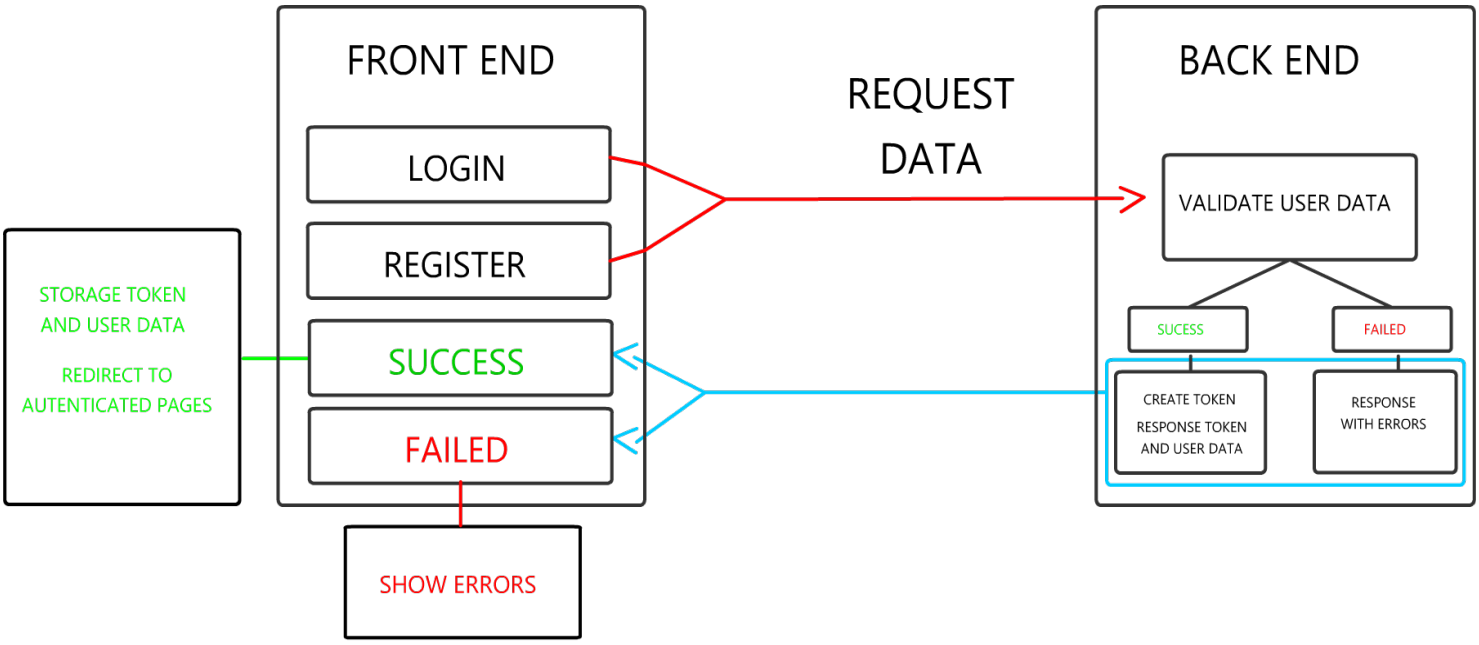
Utilizo el flujo de trabajo básico de Laravel (MVC):
Request → ruta → controlador → modelo → controlador → vista (respuesta json en este caso).

FRON TEND:

Para construir la interfaz de usuario e interactividad utilizo React y Typescript con Vite como empaquetador y TailwindCSS con Shadcn ui para los estilos.
También utilizo otros componentes para funcionalidad especifica como: Masonry para que se respete la coherencia de alineación de la donde se listan las propiedades y Rating para el componente de calificación de estrellas (que modifiqué para que al pasar el mouse se cambiara el background y al hacer click se mandara la petición al back end).

Hago uso de React Router para las rutas, Zustand para manejar el estado global y Axios para manejar las peticiones con un interceptor (para colocar el token si se encuentra en el locaStorage).

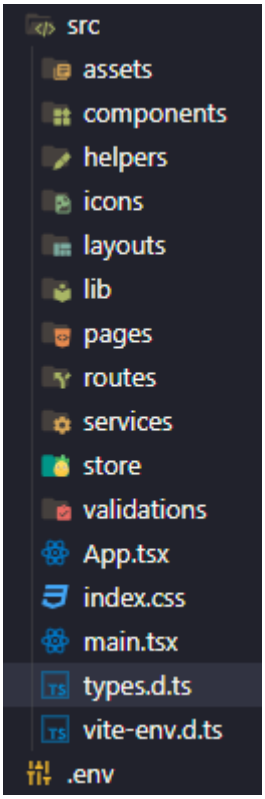
Flujo de autenticación:

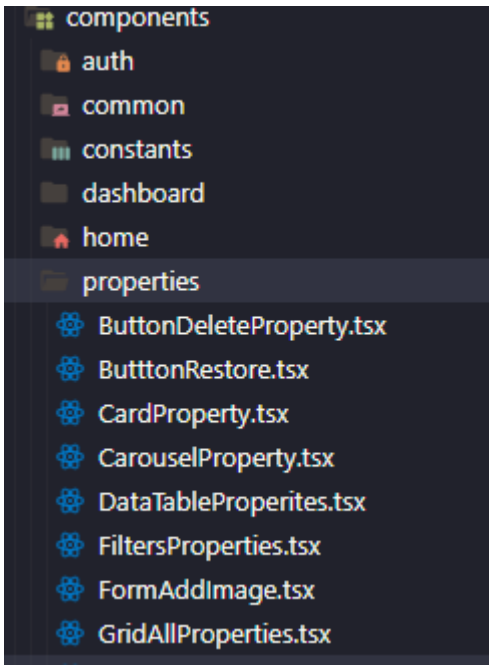


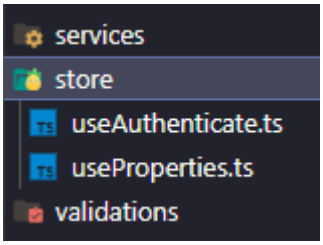
La **ESTRUCTURA** general de proyecto es básica:

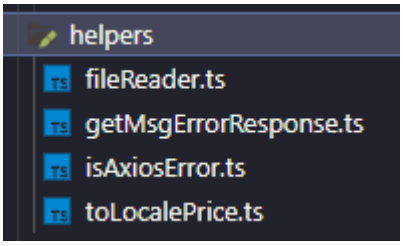
Main → Router → Página → Componentes.

Y se utilizan Helpers – Tipos – Layouts – Constantes entre páginas o donde hagan faltan.









FUNCIONALIDADES ESPECÍFICAS:

-**Envío de correo**: Al registrar un usuario se envía un correo de bienvenida.
Utilicé Mailtrap para el envío de correo (en el entorno local, solo haría falta cambiar en el archivo .env las credenciales de una cuenta pro para el envío de correo en producción).

Cuando se registra un usuario registro un evento y luego, un listener procesa el evento de manera asíncrona (mediante queues).

-**Sistema de calificación**: cada propiedad (inmueble) tiene una calificación basada en puntos del 1 al 5 representada mediante estrellas.

Cada vez que un usuario califica a una propiedad, se guarda la información en la base de datos, luego simplemente saco un promedio para la calificación general de cada propiedad.

-**Autenticación por tokens**: Excluyo el middleware VerifyCsrfToken para las rutas /api/* y utilizo tokens para validar a los usuarios.

De esta manera me fue más fácil hacer el despliegue (el Back End y el Front End en servidores diferentes) sin tener problemas con los Cors.

Sin embargo, también está desarrollada (en otra rama) para hacer la autenticación con sesiones y cookies cuando se va a desplegar en un mismo servidor.

-**Despliegue:** Para esta prueba desplegué el Back End en Hostgator haciendo uso de Cpanel para la configuración y el flujo que ofrece para CI/CD.

Para el Front, desplegué la aplicación en vercel, haciendo uso de la simpleza que ofrecen para desplegar aplicaciones web.