



# ESTRUCTURA DE PROYECTOS EN VUE

## INTEGRANTES:

- Jordan Ramírez Gallego
- Sergio Andrés Ordoñez Díaz
- Ángel Gustavo Solano Trujillo
- David Santiago Gómez Gutiérrez

# AGENDA DE CONTACTOS

Agenda de Contactos es una app móvil básica que permite guardar nombres, correos electrónicos, teléfonos y empresas de manera eficiente. Puedes buscar o editar cualquier contacto.



page 02



# TECNOLOGÍAS

## FRONTEND

- Framework: Ionic Vue
- Estado: Pinia (Gestión centralizada)
- HTTP: Axios (Conexión con backend)

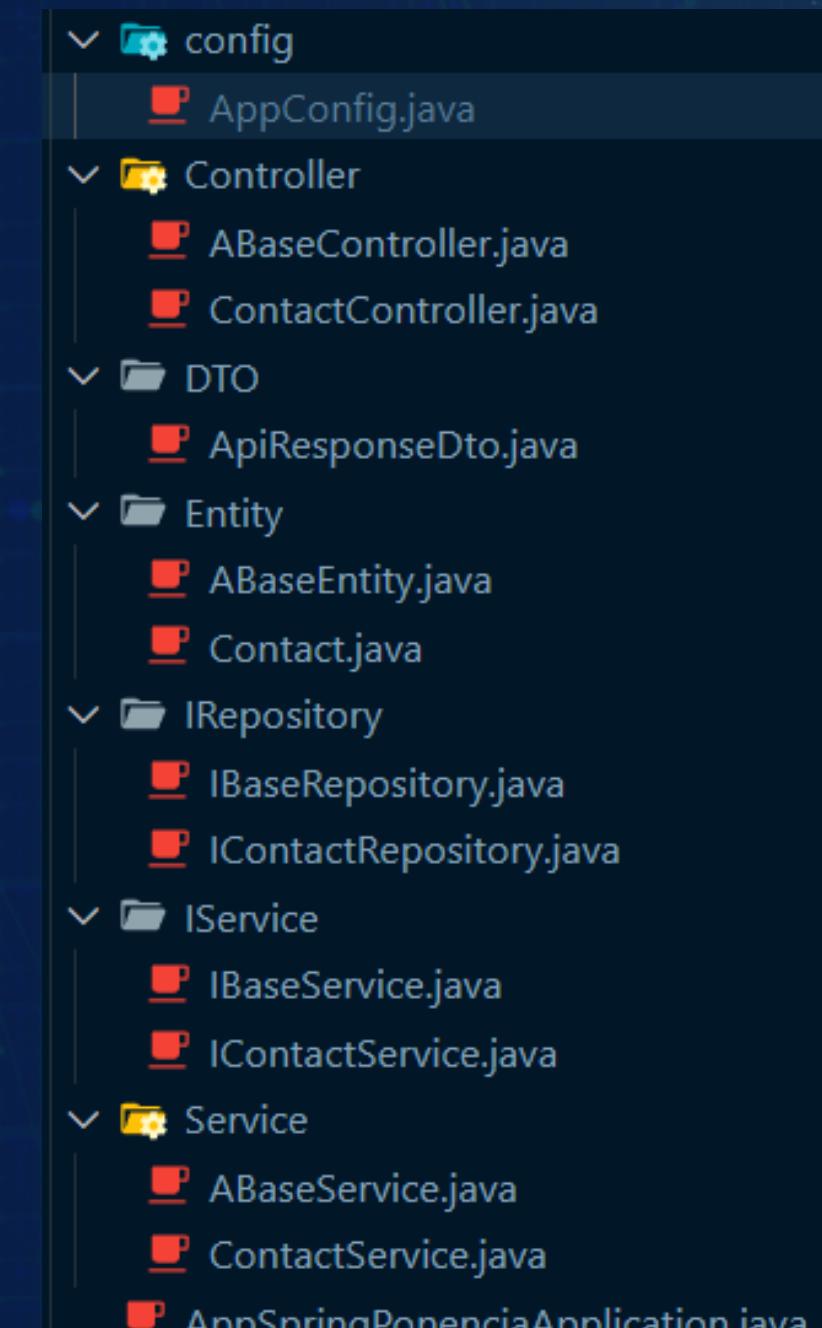


## BACKEND

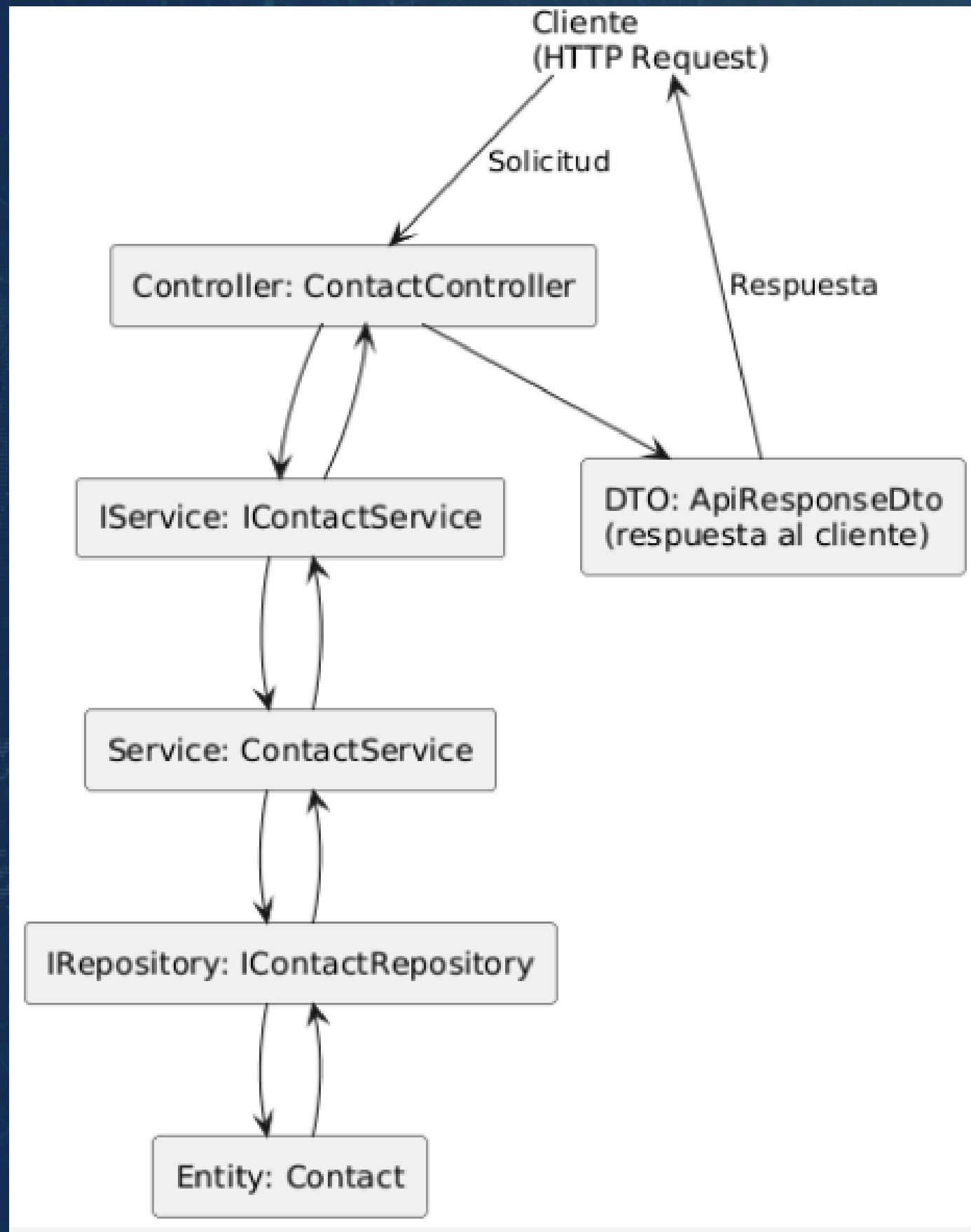
- Java 17 + Spring Boot
- MySQL(Base de datos)



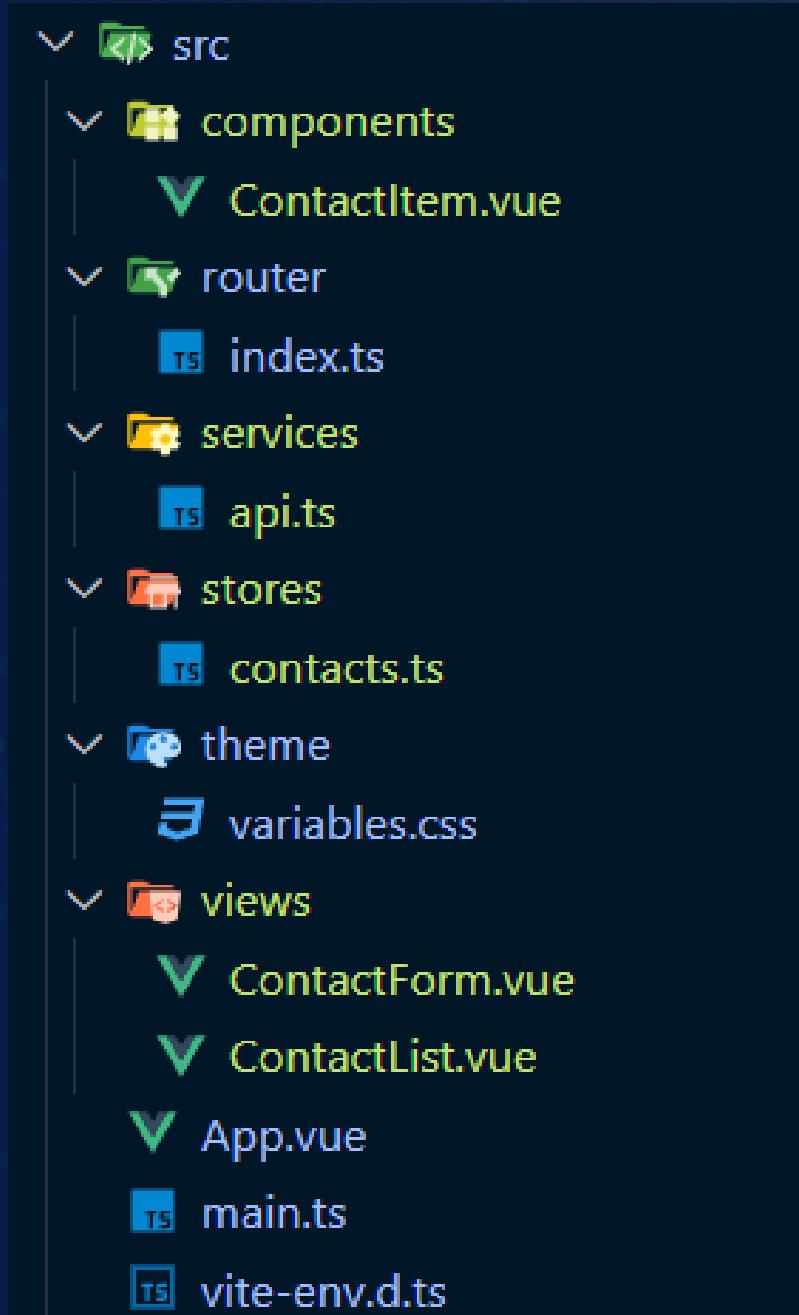
# BACKEND (SPRING BOOT)



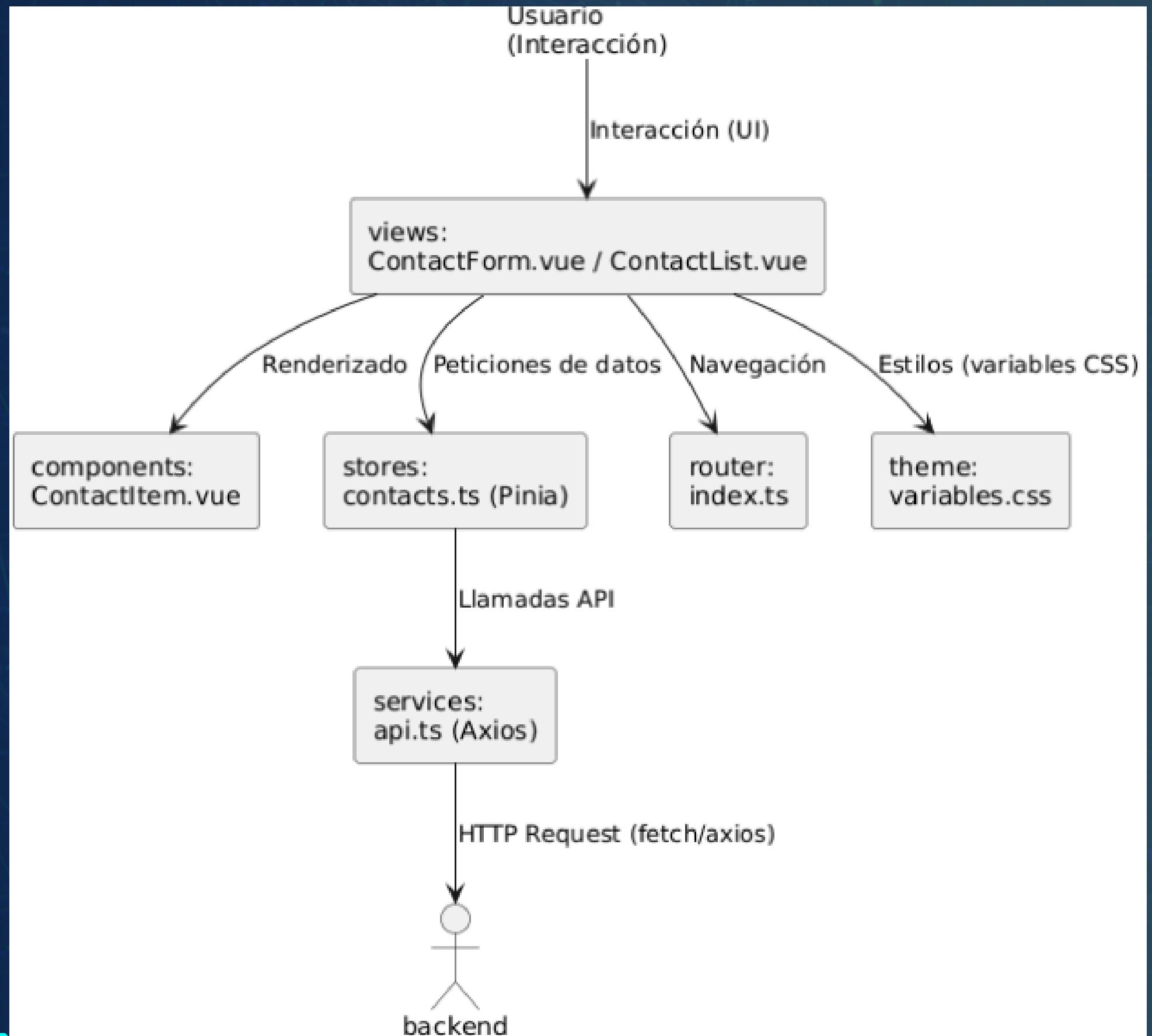
Es una API REST desarrollada que sigue una arquitectura en capas (controladores, servicios, repositorios y entidades) para gestionar operaciones CRUD de contactos. Utiliza un diseño basado en interfaces y clases abstractas (como ABaseEntity y ABaseService) para reutilizar lógica común, como auditoría. Configurado con CORS para integración con Ionic.



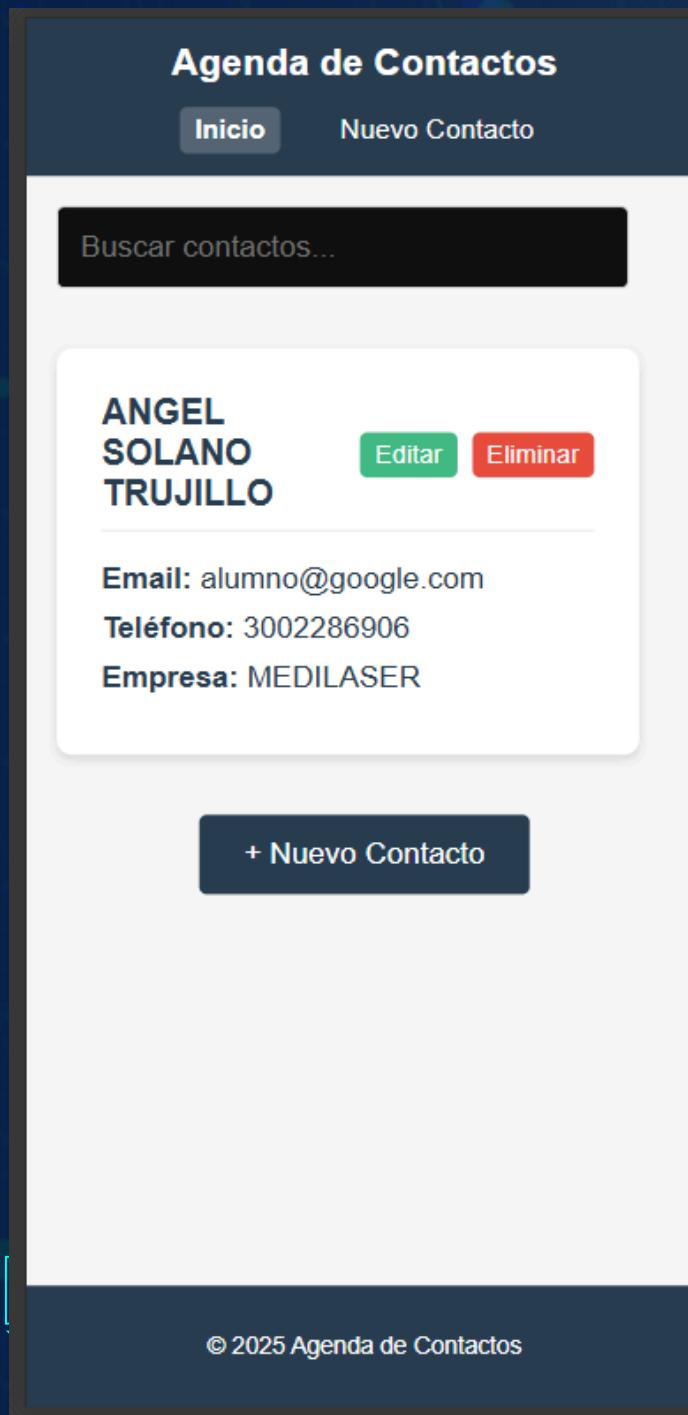
# FRONTEND (IONIC VUE)



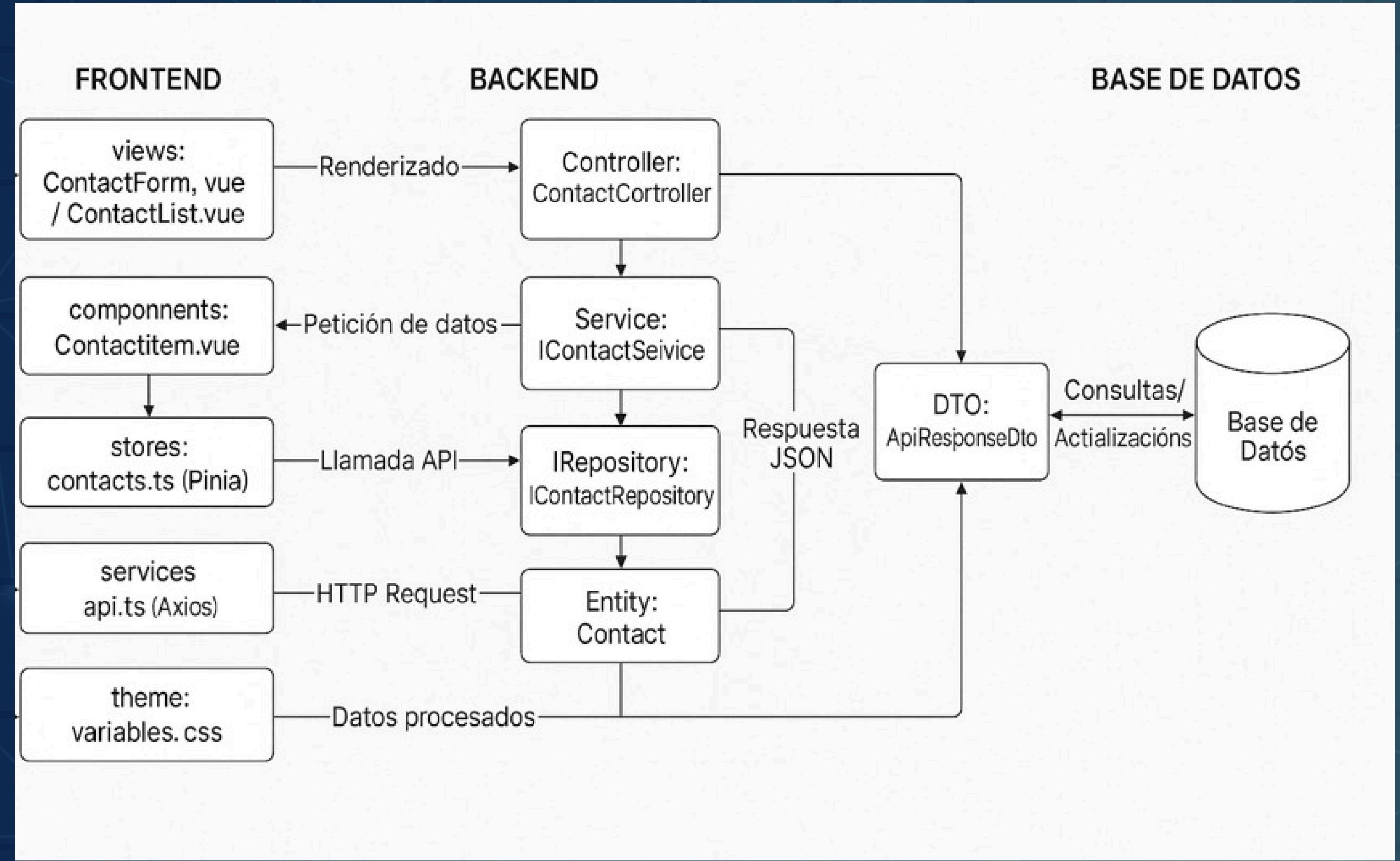
Ofrece una interfaz intuitiva y responsive para gestionar contactos. Utiliza Pinia para el manejo centralizado del estado, asegurando que los datos de contactos se mantengan sincronizados entre componentes. La aplicación consume la API REST del backend mediante Axios, implementando un CRUD completo con búsqueda en tiempo real.



# CONEXIÓN BACKEND Y FRONTEND



La conexión entre el frontend y el backend se realiza mediante una API RESTful, donde el frontend consume los endpoints del backend a través de llamadas HTTP con Axios. El backend procesa las solicitudes, interactúa con la base de datos y devuelve respuestas estandarizadas en formato JSON, que el frontend recibe y maneja.



# SERVICIO DE CONEXIÓN FRONTEND



```
api.ts  U ●
src > services > api.ts > ...
1 import axios, { AxiosInstance } from 'axios';
2
3 const api: AxiosInstance = axios.create({
4   baseURL: 'http://localhost:9000/api', // Ruta asignada al backend
5   timeout: 10000,
6   headers: {
7     'Content-Type': 'application/json',
8   },
9 });
10
11 // Respuesta de mensaje para manejar errores de respuesta
12 api.interceptors.response.use(
13   (response) => {
14     return response.data;
15   },
16   (error) => {
17     const errorMessage = error.response?.data?.message ||
18       error.message ||
19       'Error de conexión con el servidor';
20
21     console.error("Error API:", {
22       url: error.config?.url,
23       status: error.response?.status,
24       message: errorMessage
25     });
26
27     return Promise.reject(errorMessage);
28   }
29 );
30
31 export default api;
```

# SERVICIO DE CONEXIÓN BACKEND

```
AppConfig.java X
src > main > java > com > corhuila > app_spring_ponencia > config > AppConfig.java > ...
1 package com.corhuila.app_spring_ponencia.config;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5
6 import java.util.Arrays;
7 import org.springframework.web.cors.CorsConfiguration;
8 import org.springframework.web.cors.CorsConfigurationSource;
9 import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
10
11 @Configuration
12 public class AppConfig {
13     @Bean
14     public CorsConfigurationSource corsConfigurationSource() {
15         CorsConfiguration config = new CorsConfiguration();
16         config.setAllowedOrigins(Arrays.asList(
17             ...a:"http://localhost:8100", // Frontend Ionic Vue en desarrollo
18             "capacitor://localhost", // Para aplicaciones móviles Ionic
19             "http://localhost" // Para emuladores
20         ));
21         config.setAllowedMethods(Arrays.asList(...a:"GET", "POST", "PUT", "DELETE", "OPTIONS"));
22         config.setAllowedHeaders(Arrays.asList(...a:"Content-Type", "Authorization", "X-Requested-With", "Accept"));
23         config.setAllowCredentials(allowCredentials:true);
24         config.setMaxAge(maxAge:3600L);
25
26         UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
27         source.registerCorsConfiguration(pattern:"/**", config);
28         return source;
29     }
30 }
```



page 11



0



# THANK YOU!