**Documentación de la solución**

La solución de la prueba técnica sigue el patrón de diseño de arquitectura de software de Separación de intereses. La aplicación está hecha en Vue JS (lado del cliente) y una API REST con Codeigniter (lado del servidor).

Codeigniter es un framework PHP de código abierto que acelera y optimiza el desarrollo de aplicaciones web gracias a un compacto diseño de software.

Vue.js es un framework de Javascript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página.

**PROS**

* Al ser desarrollos independientes esto hace que los recursos se encuentren por separado logrando un desarrollo más ligero, escalable y flexible.
* La escalabilidad hace que los desarrollos pueden ir evolucionando según el cliente lo necesite y de manera independiente.
* El desarrollo de una aplicación basado en APIs da un mejor resultado que la utilización de un desarrollo tradicional MVC (Modelo Vista Controlador).
* Si alguna de las funciones del backend falla, el frontend seguirá activo y funcionando con las demás funciones del backend.
* Se pueden crear diferentes aplicaciones frontend y tenerlas conectadas a un solo backend en este caso a la API REST.

**CONTRAS**

* Si las pruebas no son exhaustivas del lado del backend algún error puede realizar un comportamiento inadecuado del lado del frontend y el error puede no ser visible del lado del cliente.
* Riesgos en la seguridad, si no se implementa medidas de autenticación en las peticiones la información puede estar en riesgo.

**Limitaciones**

* Si no se generan los endpoints adecuados para el tratamiento de la información, esto puede generar limitaciones de lado del cliente.
* La separación en las capas de desarrollo genera dispersión en el uso de las tecnologías y lenguajes de programación y esto puede representar una limitación al mantenimiento de los desarrollos.

**Mejoras**

Las mejores en la solución son las mencionadas en las contras y limitaciones:

* Validación de cada uno de los métodos de petición para evitar errores no visibles, esto hace ampliar las pruebas en el backend.
* Seguridad en los endpoints.