**Questionario**

**¿Por qué deberíamos utilizar VueJS?**

Vue.js es flexible y escalable.

En la práctica, esto significa que se puede utilizar para un enorme SPA modular (aplicaciones de una sola página), así como para construir pequeñas partes interactivas que se integrarán utilizando una tecnología diferente.

En otras palabras, puede ser cualquier cosa que necesite: solo una biblioteca en su proyecto o un marco con todas las funciones que se utiliza para crear un producto completo.

Convenciones prácticas.

Escribir código repetitivo es un sumidero de tiempo y recursos. Vue.js lo ayuda a evitar este tipo de trabajo ocupado al proporcionar muchas soluciones integradas y hacer cumplir las convenciones que ahorran esfuerzo.

Algunos de estos incluyen soporte nativo para cosas como administración de estado, animaciones y componentes de composición. Si bien esto puede parecer muy técnico, la conclusión es simple: ¿por qué perder el tiempo reinventando la rueda cuando solo puedes usar Vue?

Prueba del futuro.

El mantenimiento es una parte del desarrollo de software que a menudo se olvida.

Una vez que haya creado e implementado su aplicación, también debe mantenerse actualizado con las correcciones de errores, las nuevas funciones y otras mejoras. Vue.js facilita la actualización. Los nuevos lanzamientos que incluyen grandes actualizaciones se realizan de tal manera que se mantenga la mayor compatibilidad con versiones anteriores posible (90% entre las versiones 1 y 2).

Además, un proyecto basado en Vue no necesitará una refactorización rápidamente, liberando sus valiosos recursos para el desarrollo de funciones en lugar de lo que es básicamente una reelaboración desde el punto de vista comercial.

Aprende de los errores de los demás.

Another thing that makes Vue a solid technological choice is the fact that since it’s newer, its creators had the opportunity to learn from their predecessors’ mistakes, especially the issues present in AngularJS 1.x.

The clunky, illogical and hacky bits of other major frameworks are nowhere to be seen in Vue.

Progresivo.

Si tiene un proyecto que necesita un impulso de rendimiento, reescribirlo en Vue puede ser la solución milagrosa.

Gracias a su naturaleza progresiva, el marco se puede introducir en su base de código de forma gradual; no es necesario volver a trabajar todo de una sola vez. Vaya componente por componente para que todo el esfuerzo sea más manejable.

**¿Cuál es el ciclo de vida de un componente en VueJS?**

El ciclo de vida de un componente en Vue son las diferentes fases que atraviesa desde el momento de creación hasta el momento en que es destruido.

Hooks de lifecycle.

son métodos especiales o mirillas para darnos ideas sobre cómo funcionan las cosas detrás de escena de una biblioteca de framework (Vue). Estos métodos le permiten saber cuándo se crea un componente, se agrega al DOM, se actualiza o se destruye en la instancia de Vue

**¿Qué es un “estado”?**

El “estado” es simplemente un objeto que contiene las propiedades que deben compartirse dentro de la aplicación: este objeto de estado solo tiene una matriz de números. Las mutaciones son funciones responsables de la mutación directa del estado de almacenamiento.

**¿Qué son los “props”?**

En Vue, los “props” (o propiedades) son la forma en que pasamos los datos de un componente principal a sus componentes secundarios. Cuando construimos nuestras aplicaciones a partir de componentes, terminamos construyendo una estructura de datos llamada árbol.