1. **¿Qué es React?**

React es una librería Javascript focalizada en el desarrollo de interfaces de usuario, creada originalmente por Facebook. Es software libre y a partir de su liberación acapara una creciente comunidad de desarrolladores y entusiastas. Su creación se realizó en base a unas necesidades concretas, derivadas del desarrollo de la web de la popular red social. Además de facilitar el desarrollo ágil de componentes de interfaces de usuario, el requisito principal con el que nació React era ofrecer un elevado rendimiento, mayor que otras alternativas existentes en el mercado.

1. **¿Cuáles son las ventajas de utilizar React en un proyecto?**
2. Desarrollo rentable: Vía económica para crear aplicaciones multiplataforma, por ejemplo, en lugar de crear dos aplicaciones diferentes para Android y iOS, el desarrollador puede implementar el mismo código para ambas plataformas. Esto implicaría reducir los costos de desarrollo en un 50% aproximadamente.

### Entrega más rápida de proyectos de aplicaciones: Los desarrolladores pueden aprovechar varios componentes listos para crear funciones de aplicaciones más rápido que nunca. Además, se necesitan menos esfuerzos, ya que se requiere menos código en comparación con otras plataformas de desarrollo.

### Aprovecha JavaScript: JavaScript es el lenguaje de programación más utilizado en el mundo, esto facilita a los desarrolladores de JavaScript el uso de ReactJS, ya que el marco está escrito en JavaScript.

### Ventaja del código abierto: Brinda a los desarrolladores acceso para usar bibliotecas y marcos de forma gratuita. Impone algunas restricciones sobre la reutilización del software, pero también proporciona protección legal para los desarrolladores.

### Función de recarga activa (Hot Reloading): La función de recarga activa permite al desarrollador implementar cambios en un código y ver el efecto en la aplicación en tiempo real. Por lo tanto, permite actualizar una aplicación que ya está activa.

### Excelente rendimiento de la aplicación: Las aplicaciones nativas tienen el mejor rendimiento general, pero las aplicaciones.

### El diseño modular: ReactJS aprovecha la tecnología de programación modular para ofrecer funciones en forma de bloques libres llamados módulos. Esto permite la fácil creación e integración de actualizaciones de aplicaciones a través de módulos fáciles de usar, que son reutilizables para API web y móviles.

1. **¿Qué son las props?**

Los Props se refieren a las propiedades las cuales cumplen un rol importante en el proceso de desarrollo de una aplicación o página web. Los componentes son los bloques de construcción de React y estos componentes usan Props para mejorar su funcionalidad y poder reusar el código.

1. **¿Cuál es la diferencia entre una librería y un framework?**

Las principales diferencias entre librerías y frameworks son que un framework nos proporciona un marco de trabajo para desarrollar aplicaciones, mientras que las librerías únicamente solucionan un problema concreto, haciendo el código más fácil de leer.

1. **¿Qué es el Estado = STATE ?**

Son los valores internos que manejan la lógica y los datos de un componente, permite que éste reaccione a cualquier cambio lo que hará que se vuelva a renderizar en la interfaz.

Algunas de las características son:

-Es inmutables

-Es asíncrono.

-No se puede modificar directamente.

Para hacer cambios hay que hacer uso del método setState().

1. **¿Cuándo fue creada React, y por quién?**

React fue lanzada en el año 2013 por Jordan Walke, un ingeniero de Facebook y el primer prototipo se llamó "FaxJS"

1. **¿Qué son los componentes?**

Los componentes de React son elementos autónomos que pueden reutilizarse en una página.

1. **¿Qué entiendes por “Componentes Reutilizables”?**

Los componentes reutilizables son piezas de código que pueden implementarse una y otra vez, a medida que una aplicación se amplía. De esta forma se asegura la escalabilidad.

1. **¿Por qué React no ha trascendido de librería a Framework?**

React permite organizar el código en base a componentes reutilizables. Esos componentes tienen una vista, expresada en código JSX, que es capaz de reaccionar a cambios en los datos que manejan esos componentes. Nada más que eso, por tanto hasta aquí está claro que React no es un framework.

1. **¿Qué se entiende como renderizar?**

El renderizado es el proceso de mostrar tu [página web](https://smartsi.co/#paginasweb) en un navegador, debido a que transforma los documentos HTML y otros recursos de una página web en una representación visual interactiva en el dispositivo del usuario. Por lo que renderizar tiene que ver con una serie de acciones como: analizar, procesar y visualizar una web, las cuales son realizadas en cuestión de segundos.

1. **¿Qué se conoce o entiende como stack de desarrollo?**

Stacks de desarrollo web es el que compone de un sistema operativo, un servidor web, una base de datos y un intérprete de lenguaje de programación.

1. **¿Qué diferencia hay entre la extensión .js y la extensión .jsx?**

La extensión de JS es de javascript, mientras que JSX es una combinación de html y js.