

Historia de React.

React es una biblioteca de JavaScript desarrollada por Facebook, lanzada por primera vez en 2013. Su principal objetivo es facilitar la creación de interfaces de usuario (UI) interactivas y eficientes mediante un enfoque basado en componentes. React permite a los desarrolladores construir aplicaciones web de una manera más modular y reutilizable, lo que mejora la mantenibilidad y escalabilidad del código.

Hitos importantes en su desarrollo:

1. **Lanzamiento de React (2013):** La primera versión de React fue lanzada en mayo de 2013. Este lanzamiento marcó un cambio significativo en la forma en que las aplicaciones web fueron desarrolladas. React introdujo el concepto de un "DOM virtual", lo que permitió actualizaciones más eficientes de la UI al comparar y solo actualizar las partes que cambiaron, en lugar de renderizar todo el DOM.
2. **React Native (2015):** En 2015, Facebook lanzó React Native, una extensión de React para el desarrollo de aplicaciones móviles nativas. React Native permite a los desarrolladores crear aplicaciones móviles para iOS y Android utilizando JavaScript y los principios de React, pero con acceso a componentes nativos. Esto amplió el alcance de React más allá del desarrollo web, haciendo posible escribir una sola base de código para múltiples plataformas.
3. **Fiber Reconciliation Algorithm (2017):** En 2017, React introdujo React Fiber, una reescritura interna del algoritmo de reconciliación que mejoró significativamente el rendimiento y la capacidad de manejar animaciones, gestos, y tareas que requieren alta prioridad. Fiber permitió que React dividiera el trabajo en unidades más pequeñas, lo que mejoró la capacidad de respuesta de las aplicaciones durante actualizaciones complejas.
4. **React Hooks (2019):** En 2019, React introdujo los "Hooks", una característica que permitió el uso de estado y otras funcionalidades de React en componentes funcionales, eliminando la necesidad de clases para manejar el estado. Esto simplificó la lógica del componente y promovió un enfoque más funcional en el desarrollo de aplicaciones React. Hooks, como

useState y useEffect, se convirtieron rápidamente en estándares dentro de la comunidad de desarrollo de React.

5. **React Server Components (2020):** Presentado en 2020, React Server Components es una nueva característica que permite renderizar componentes en el servidor y enviar solo el HTML necesario al cliente. Esto reduce la cantidad de JavaScript que los navegadores necesitan descargar, mejorando el rendimiento de las aplicaciones.

3. Escribe una breve explicación sobre por qué Facebook decidió crear React.

Básicamente Facebook decidió crear react para poder crear aplicaciones complejas y dinámicas, manteniendo interfaces de historia eficientes y escalables, antes de React actualizar el DOM (Document Object Model) hoy era ineficiente y con más facilidad a ocurrir errores especialmente en aplicaciones que eran muy concurridas y con actualizaciones frecuentes.

Hola a lo que el equipo del Facebook requería de una solución que permitiera construir componentes reutilizables y que se hicieran más eficientes a la actualización de la interfaz, por lo que crearon React optimizando de esta manera la manera en la que las aplicaciones se actualicen y haciendo de las aplicaciones más rápidas y el código más fácil de mantener.

4. Menciona tres ventajas de usar React en el desarrollo de aplicaciones web.

- **Componentes reutilizables:** React permite construir aplicaciones con componentes modulares que pueden ser utilizados y reutilizados en diferentes partes de la aplicación. Esto mejora la consistencia y reduce el tiempo de desarrollo.
- **Comunidad y ecosistema amplio:** React cuenta con una gran Comunidad de desarrolladores y un extenso ecosistema de bibliotecas y herramientas. Esto facilita la resolución de problemas y la integración con otras tecnologías y la disponibilidad de recursos y de documentación.
- **Actualizaciones y mantenimiento continuo:** React es mantenido por Facebook, lo que garantiza actualizaciones regulares, mejoras en el rendimiento y soporte para las últimas tecnologías.

5. Explica cómo el Virtual DOM mejora el rendimiento de una aplicación.

El virtual DOM es una representación en memoria del DOM real, lo que hace es que en lugar de interactuar directamente con el DOM real React primero hace cambios en el virtual DOM.

Primero compara cuando se realiza un cambio en un componente React, en lugar de actualizar inmediatamente el DOM real, React primero actualiza el virtual DOM. Luego, compara la nueva versión del virtual DOM con la versión anterior para identificar que las partes de la interfaz que han cambiado a lo que se llama “reconciliación”.

Después de comparar, React genera un “diff” que contiene solo los cambios necesarios entre el nuevo y el antiguo virtual DOM, y con esta información React solo actualiza las partes que se han cambiado en lugar de volver a renderizar toda la página o componentes innecesarios.

Por **ejemplo** imaginemos una lista de empleados a los cuales ingresa un nuevo empleado, Sin el virtual DOM cada vez que cambia o ingresa a un nuevo elemento el navegador tiene que volver a renderizado toda la lista. con el virtual DOM, React solo se actualiza en ese único elemento dejando el resto del listado intacto mejorando de esa manera la eficiencia de la aplicación.

6. Define qué es una Single Page Application (SPA).

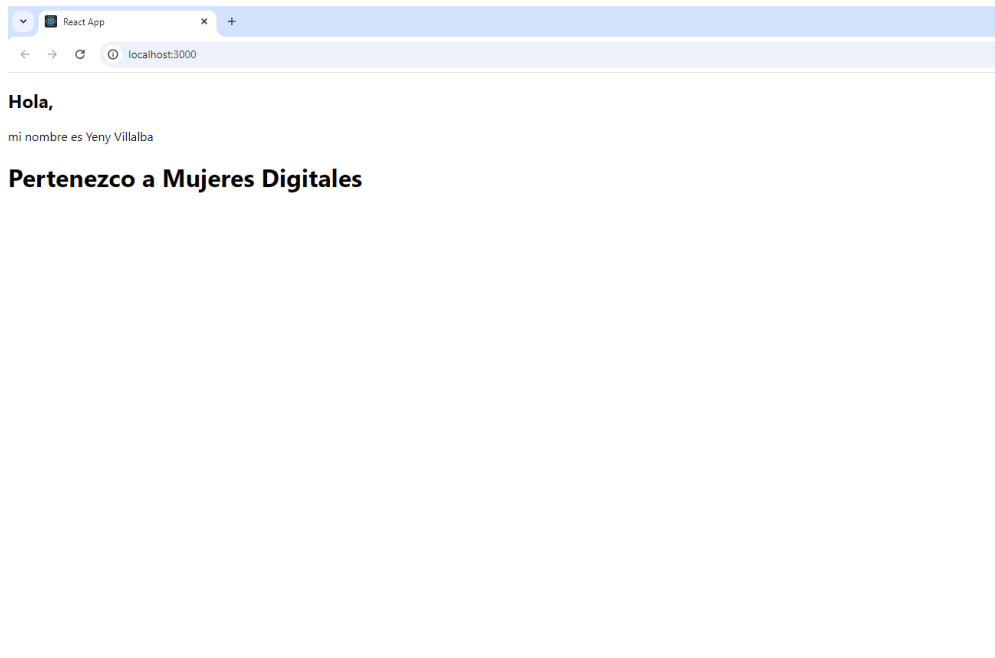
Una Single Page Application (SPA) es un tipo de aplicación web que interactúa con el usuario cargando una única página HTML, actualizando dinámicamente el contenido a medida que el usuario interactúa con la aplicación a diferencia de otras aplicaciones web donde cada interacción del usuario podría desencadenar la carga de una nueva página del del servidor una SPA carga una vez y después gestiona todas las interacciones mediante java script.

7. Explica cómo React facilita la creación de una SPA.

React facilita la creación de spa al proporcionar un enfoque basado en componentes, flujo de datos unidireccional , Los que permite a los desarrolladores construir interfaces de usuario dinámicas que pueden actualizarse y responder a las interacciones del usuario sin necesidad de recargar la pag.

Con componentes reutilizables que representan diferentes partes de la interfaz, estos componentes gestionan su propio estado y se actualizan de manera independiente cuando le cambian los datos sin afectar el resto de la aplicación.

9. Inicia el servidor de desarrollo y comparte una captura de pantalla de tu proyecto en ejecución.



10. Explica brevemente el propósito de las carpetas src y public en un proyecto React.

Src: contiene el código fuente principal de la aplicación. Aquí dónde se desarrolla la mayor parte de la lógica de la aplicación, estructura de componentes y estilos.

Public: contiene archivos estáticos que sirven directamente al navegador. Estos archivos no son procesados por el bundler como webpack sino que sirven tal como están.

11. Explica cómo JSX se diferencia del HTML tradicional.

JSX es una extensión de JavaScript que permite escribir código con una sintaxis similar a HTML dentro de archivos JavaScript a diferencia de HTM, JSX:

- permite incluir expresiones JavaScript dentro de las llaves {}.
- utiliza camelCase para atributos como por ejemplo 'className' en lugar de 'class'

- soporta componentes personalizados en lugar de solo elementos de HTML estándar.
- Permite lógica condicional directamente dentro del JSX.

12. Define los roles principales en un equipo SCRUM.

Algunos de los roles principales son:

- **propietario del producto:** es el responsable de definir y priorizar las características del asegurando que el equipo de desarrollo trabajar en las tareas actúa como enlace entre los interesados y el equipo.
- **Scrum Master:** facilita el proceso asegurando que el equipo siga las prácticas y principios de scrum. Elimina obstáculos que puedan impedir el progreso al equipo protege al equipo de distracciones externas y asegura que todos atienden y adopten estudio incorrecto no es un jefe sino un guía.
- **equipo de desarrollo:** es el equipo encargado de construir el producto, compuesto por profesionales de múltiples trina cómo desarrolladores diseñadores testers que colaboran para enseñar incrementos funcionales del producto en cada sprint. Donde el equipo es autoorganizado y decide cómo abordar las tareas para cumplir objetivos.

13. Explica qué es un sprint y cómo se planifica.

Un sprint es una unidad de tiempo fija en la que el equipo scrum trabaja en conjunto de tareas para entregar un incremento del producto, se planifica cuidadosamente en la reunión de sprint planning donde se seleccionan y organizan tareas en función de las Prioridades del dueño del producto y la capacidad del equipo.

Se planifica de la siguiente manera:

- **Reunión de planificación:** el sprint comienza con una reunión de planificación, dónde el propietario del producto, Scrum Master y el equipo de desarrollo participan.
- **Selección de tareas:** en esta reunión el propietario del producto, presenta los elementos del producto priorizados que son más importantes para el próximo sprint , También el equipo de desarrollo revisa los elementos que selecciona aquellos que se comprometerá a completar durante el Sprint.
- **Definición del Sprint Goal:** el equipo y el propietario del producto acuerdan el objetivo del sprint que define el propósito y el enfoque del trabajo durante el sprint.

- **División de tareas:** El equipo divide los elementos seleccionados en tareas más pequeñas y manejables, estimando el esfuerzo necesario para completarlas.