

Taller 2 React

1.

Hitos



2013 Open source

Se lanzo al público como un proyecto de código abierto (su código fuente se puso a disposición para su uso o modificación según los usuarios) en mayo de 2013, permitiendo a la comunidad de desarrolladores usar y mejorar la biblioteca. Esto marco un hito importante, ya que estableció una nueva forma de desarrollar aplicaciones interactivas con un enfoque basado en componentes.

2015 introducción de React Native

Un gran hito en la historia de React fue la introducción de React Native en 2015. Permitió a los desarrolladores crear aplicaciones móviles nativas para iOS y Android utilizando la misma base de código de React. Esto revoluciono el desarrollo móvil al proporcionar una forma mas eficiente de crear aplicaciones multiplataforma de alto rendimiento.



2. ¿Por qué facebook decidio crear React?

Debido a problemas de rendimiento y mantenibilidad, en cosas como noticias de facebook o instagram. Debido a eso crearon React enfocados en mejorar la eficiencia, poder mostrar varios componentes sin necesidad de recargar toda la pagina.

3. Ventajas de React.

- Es fácil de aprender, es simple en términos de sintaxis. Con saber HTML es suficiente, no es necesario aprender TypeScript con Angular.
- Gracias a si DOM virtual que usa XHTML o XML en una estructura de árbol más manejable para los navegadores mientras analiza los diferentes elementos de la aplicación.

- Debido a que es Open Source, cuenta con una comunidad que continuamente mejora la herramienta, además cuenta con varias librerías complementarias.

4. Virtual DOM

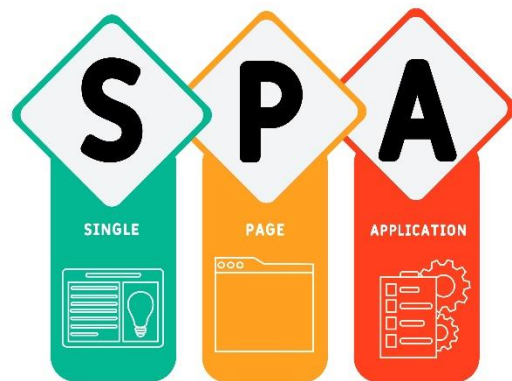
Cuando ocurre un cambio en la aplicación web, el virtual DOM interpreta dichos cambios y calcula la manera mas eficiente de actualizar el DOM para que renderice la menor cantidad de cambios posibles.

Así que al ser un intermediario permite interceptar cambios en la interfaz, calcular que elementos son en realidad necesarios, aplicarlos solamente en los nodos relevantes y enviar los cambios optimizados al DOM para la fase de renderizado, ahorrando recursos de procesamiento y brindando una experiencia de usuario más fluida en las aplicaciones web.

5. Single Page Application (SPA).

Es un modelo de aplicación que fue creado como una solución al modelo de construcción de software front-end convencional, que se denomina MPA (Múltiple Page Application).

Dentro de un MPA, cada vista o página de la aplicación es un archivo HTML separado que se carga desde el servidor cuando el usuario navega hasta él. Así que como resultado la navegación se vuelve lenta y la experiencia de usuario poco fluida, ya que hay que esperar a que se recargue la página completa cada vez que se carga una diferente.



Así que SPA son aplicaciones cuya funcionalidad se concentra en una sola página, solo utiliza una parte de la pantalla, sin necesidad de recargar todo. Por lo tanto, de forma asíncrona, solo se actualiza el contenido principal, así envía y recibe menos datos, lo que resulta en una operación mucho más eficiente. Logrando mejorar la experiencia de usuario. Plataformas como Outlook, Gmail y YouTube la usan.

En una SPA, todo el código necesario se carga en un solo archivo HTML y la navegación entre vistas y paginas se manejan dinámicamente del lado del cliente sin recargar la página completa.

6. React y SPA

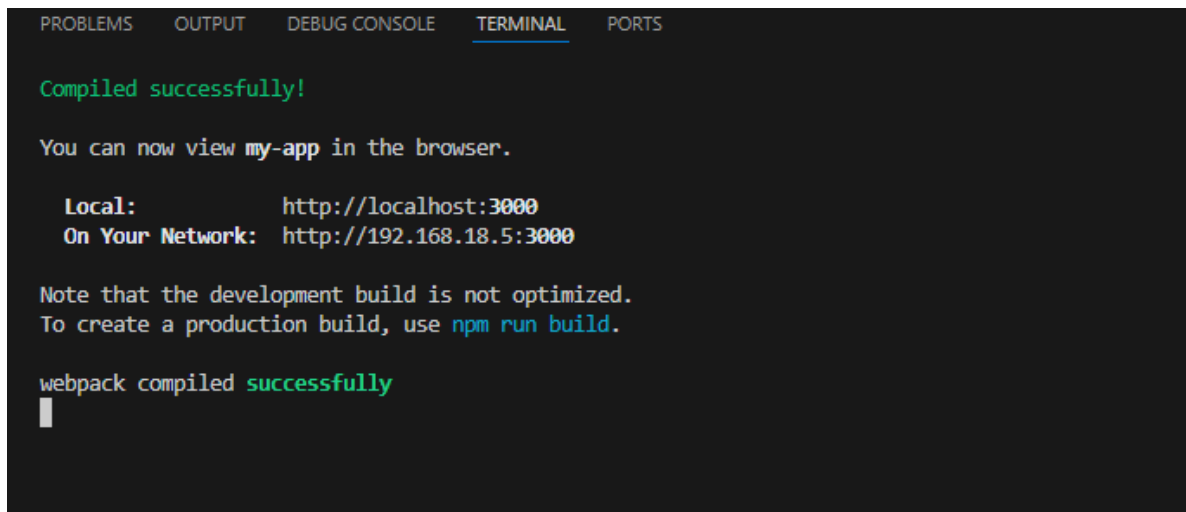
React facilita la creación de SPAs al manejar actualizaciones dinámicas de la interfaz grafica sin recargar la pagina completa. Por ejemplo, al cambiar el estado de un componente, React vuelve a renderizar solo ese componente, sin afectar al resto de la página.

Nota: Es el ejemplo usado en clase.

```
1  import React from "react";
2
3  class Contador extends React.Component{
4    constructor(props){
5      super(props);
6
7      this.state = {contador: 100}
8    }
9
10   incrementar = () =>{
11     //así incremento el contador siempre que se active la función al dar click
12     this.setState({contador : this.state.contador-1});
13   }
14
15   render(){
16     return(
17       <div>
18         <p>El contador va en : {this.state.contador}</p>
19         <button onClick={this.incrementar}>Decrementar</button>
20       </div>
21     )
22   }
23 }
24
25 export default Contador;
```

React actualiza el contador que en este caso lo decreuenta. Todo esto sin recargar la página.

7. Inicio de my-app



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Compiled successfully!

You can now view my-app in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.18.5:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

8. Carpetas src y public

- **src:** Contiene todo el código fuente de la aplicación, como los componentes React, archivos css, y otros recursos como imágenes. Aquí es donde se escribe y organiza la lógica de la aplicación.

- **public:** Almacena los archivos estáticos que no son procesados por WebPack, como el archivo index.html y otros recursos que se necesitan tal cual. Los archivos en esta carpeta se sirven directamente al navegador.

9.Diferencia entre JSX Y HTML

JSX es una extensión de JavaScript que permite escribir código similar al HTML dentro de JavaScript. Pero no es ni toca HTML como tal.

En JSX se permite el uso de expresiones JavaScript dentro del código HTML. Por ejemplo, `class` se convierte en `className` porque `class` es una palabra reservada de JavaScript.

JSX usa JavaScript para crear elementos HTML de una manera más dinámica y poderosa.

10. SCRUM

Roles:

1. **Product Owner:** Representa los intereses del cliente y define las prioridades en la lista de tareas(backlog).
2. **Scrum Master:** Facilita las ceremonias de SCRUM y asegura que el equipo siga las prácticas ágiles.
3. **Development Team:** El equipo encargado de desarrollar el producto.

Sprint:

Un sprint es un periodo de tiempo corto, generalmente de 1 a 4 semanas, durante el cual el equipo de desarrollo trabaja en una lista de tareas previamente planificadas. El sprint comienza con una planificación de sprint, donde el equipo selecciona las tareas del backlog que pueden completarse durante el sprint.

