

**TALLER REACT**

**WENDY ROSERO**

**INSTRUCTOR**

**SANTIAGO GIRALDO**

**SENA/MUJERES DIGITALES**

## DESARROLLO

**Historia de React:** React fue desarrollado por Facebook y lanzado en 2013 como una biblioteca de JavaScript para construir interfaces de usuario. Inicialmente, fue creado para resolver problemas relacionados con la actualización eficiente de la interfaz en grandes aplicaciones web. Jordan Walke, un ingeniero de Facebook, fue el responsable principal del desarrollo inicial.

### Dos hitos importantes en el desarrollo de React:

1. **Introducción de JSX (2013):** JSX es una extensión de JavaScript que permite escribir código que se parece mucho a HTML dentro de archivos JavaScript. Esto facilita la creación de componentes y hace que el desarrollo con React sea más intuitivo.
2. **Lanzamiento de React Hooks (2019):** React introdujo los Hooks en la versión 16.8, lo que permitió a los desarrolladores manejar el estado y otros efectos secundarios en componentes funcionales, eliminando la necesidad de componentes de clase para estas funcionalidades.

### ¿Por qué Facebook decidió crear React?

Facebook decidió crear React debido a la complejidad creciente de sus aplicaciones web, como el manejo de interacciones en tiempo real (comentarios, "likes", etc.), que generaban problemas de rendimiento y mantenibilidad. El enfoque principal de React era hacer que la interfaz de usuario se pudiera actualizar de manera eficiente y coherente sin afectar el rendimiento de la aplicación.

---

### Tres ventajas de usar React en el desarrollo de aplicaciones web:

1. **Reutilización de componentes:** React permite crear componentes reutilizables, lo que reduce la duplicación de código y mejora la mantenibilidad.
2. **Actualización eficiente de la interfaz:** Gracias al Virtual DOM, React realiza actualizaciones eficientes en la interfaz sin recargar toda la página.
3. **Comunidad y ecosistema amplios:** React cuenta con una gran comunidad de desarrolladores, lo que facilita la resolución de problemas y el acceso a bibliotecas y herramientas complementarias.

### ¿Cómo mejora el Virtual DOM el rendimiento de una aplicación?

El **Virtual DOM** es una representación ligera del DOM real. React usa este Virtual DOM para detectar cambios en la interfaz de manera más eficiente. Cuando se realiza un cambio en el estado de un componente, React compara el Virtual DOM

con la versión anterior para identificar las diferencias (reconciliación). Solo las partes del DOM real que han cambiado se actualizan, en lugar de renderizar todo el DOM, lo que mejora significativamente el rendimiento de la aplicación.

---

### **Definición de Single Page Application (SPA):**

Una **Single Page Application (SPA)** es una aplicación web que se carga en una sola página HTML. En lugar de cargar nuevas páginas del servidor cada vez que el usuario interactúa, las SPA actualizan dinámicamente la interfaz sin recargar la página. Esto proporciona una experiencia más fluida y rápida, similar a la de una aplicación nativa.

### **¿Cómo facilita React la creación de una SPA?**

React facilita la creación de SPA mediante su capacidad para renderizar y actualizar componentes de forma dinámica sin recargar toda la página. Su arquitectura basada en componentes y el uso del Virtual DOM permiten que las aplicaciones respondan de manera rápida y eficiente a las interacciones del usuario.

### **Propósito de las carpetas src y public en un proyecto React:**

- **src (source):** Esta carpeta contiene todo el código fuente de la aplicación React. Dentro de src, encontrarás componentes, archivos JavaScript (o TypeScript), hojas de estilo, imágenes, y otros recursos relacionados directamente con la lógica y estructura de la aplicación. En resumen, es donde ocurre el desarrollo principal.
- **public:** La carpeta public contiene archivos estáticos que no necesitan ser procesados por Webpack ni otro sistema de empaquetado. Aquí suelen estar el archivo index.html, imágenes u otros recursos públicos. Los archivos dentro de public se copian tal cual en la carpeta de salida al momento de la compilación, y el contenido dentro de src es inyectado en el archivo index.html.

---

### **¿Cómo se diferencia JSX del HTML tradicional?**

**JSX** es una extensión de JavaScript que permite escribir código que parece HTML, pero tiene algunas diferencias clave con el HTML tradicional:

- En JSX, se usa **camelCase** para atributos como `className` en lugar de `class` (que es una palabra reservada en JavaScript).
- JSX permite incluir **expresiones de JavaScript** entre llaves `{}`, mientras que en HTML solo se permiten valores estáticos.
- JSX es estrictamente **sintáctico**, lo que significa que los elementos deben estar correctamente cerrados, incluso los que no requieren cierre en HTML, como `<br />`.

Ejemplo:

jsx

Copiar código

```
const element = <h1 className="title">Hello, World!</h1>;
```

En HTML tradicional, sería algo como:

html

Copiar código

```
<h1 class="title">Hello, World!</h1>
```

---

## Roles principales en un equipo SCRUM:

1. **Product Owner (Dueño del Producto):** Responsable de maximizar el valor del producto y gestionar el Product Backlog. Define las prioridades de las tareas y se asegura de que el equipo trabaje en las funcionalidades que aporten mayor valor al negocio.
  2. **Scrum Master:** Facilita y asegura que el equipo siga las prácticas de Scrum. El Scrum Master actúa como un coach, eliminando obstáculos que impidan el progreso del equipo y asegurándose de que el equipo siga los principios de Scrum.
  3. **Development Team (Equipo de Desarrollo):** Es el grupo encargado de convertir las ideas y tareas del Product Backlog en incrementos funcionales del producto. Son autónomos y autogestionados, responsables del trabajo técnico.
- 

## ¿Qué es un sprint y cómo se planifica?

Un **sprint** es un ciclo de trabajo en Scrum, que dura típicamente entre una y cuatro semanas. Durante un sprint, el equipo trabaja en un conjunto específico de tareas que han sido seleccionadas del Product Backlog, con el objetivo de entregar un incremento del producto que esté potencialmente listo para ser lanzado.

## Planificación del sprint:

1. **Sprint Planning:** Se inicia el sprint con una reunión donde el equipo, junto con el Product Owner, define qué elementos del Product Backlog se van a trabajar durante el sprint.
2. **Definir el Objetivo del Sprint:** Se establece un objetivo claro para el sprint, que guía al equipo sobre lo que debe lograr.
3. **Selección de Tareas:** El equipo selecciona tareas basándose en la prioridad establecida por el Product Owner y en su capacidad de entrega. Las tareas seleccionadas se colocan en el Sprint Backlog.
4. **División de Tareas:** Las tareas seleccionadas se descomponen en subtareas más pequeñas y manejables para asegurar que se completen durante el sprint.