

# TALLER REACT 1

## 1. Ejercicio:

**2. Investiga la historia de React y menciona dos hitos importantes en su desarrollo.**

**Lanzamiento en 2013:** React fue lanzado por Facebook en marzo de 2013. Este evento marcó el inicio de una nueva forma de construir interfaces de usuario, introduciendo el concepto de componentes reutilizables y el Virtual DOM.

**Introducción de React Native (2015):** En marzo de 2015, Facebook presentó React Native, una extensión de React para el desarrollo de aplicaciones móviles nativas usando la misma biblioteca de React. Esto amplió el alcance de React al desarrollo móvil y ayudó a consolidar su popularidad.

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

Facebook creó React para abordar problemas en el desarrollo de sus interfaces de usuario, especialmente la complejidad y el bajo rendimiento asociados con la manipulación del DOM tradicional. React introdujo una arquitectura basada en componentes y el Virtual DOM para mejorar el rendimiento, facilitar el desarrollo de interfaces dinámicas y mantener el código más organizado y modular.

## 2. Ejercicio:

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

**Reutilización de Componentes:** React permite construir aplicaciones usando componentes reutilizables, lo que facilita el mantenimiento y la escalabilidad del código, promoviendo una estructura modular.

**Virtual DOM:** Utiliza un Virtual DOM para actualizar el DOM real de manera eficiente. Al comparar versiones del Virtual DOM y aplicar solo las diferencias, React minimiza el costo de las actualizaciones y mejora el rendimiento general de la aplicación.

**Ecosistema y Comunidad:** React cuenta con una extensa comunidad de desarrolladores y un ecosistema rico en herramientas y bibliotecas, lo que proporciona soporte, recursos y soluciones para problemas comunes, acelerando el desarrollo.

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

El Virtual DOM mejora el rendimiento al reducir la cantidad de operaciones necesarias en el DOM real. React mantiene una representación en memoria del DOM (el Virtual DOM). Cuando hay un cambio, React actualiza el Virtual DOM y compara esta nueva versión con la anterior. Solo las diferencias entre el Virtual DOM y el DOM real se aplican al DOM real, reduciendo el trabajo necesario y mejorando la eficiencia de las actualizaciones.

## **3. Ejercicio:**

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

Una Single Page Application (SPA) es una aplicación web que carga una sola página HTML en el navegador y actualiza dinámicamente el contenido de esa página en respuesta a las interacciones del usuario sin recargar la página completa. Esto se logra usando JavaScript para gestionar la navegación y el contenido, lo que proporciona una experiencia de usuario más rápida y fluida.

### **7. Explica cómo React facilita la creación de una SPA. Proporciona un ejemplo de cómo un componente de React puede actualizar la interfaz sin recargar la página.**

React facilita la creación de SPA al permitir la actualización dinámica de la interfaz mediante el manejo del estado y el enrutamiento sin necesidad de recargar la página completa. Los componentes de React se actualizan de manera eficiente en respuesta a cambios en el estado o las propiedades.

#### **Ejemplo de un componente React:**

```
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Has hecho clic {count} veces</p>
      <button onClick={() => setCount(count + 1)}>Haz clic aquí</button>
    </div>
  );
}

export default Counter;
```

En este ejemplo, el componente Counter muestra un contador que se incrementa cada vez que el usuario hace clic en el botón. La interfaz se actualiza dinámicamente sin necesidad de recargar la página, gracias a la gestión del estado (useState) y la función de actualización (setCount).

## **4. Ejercicio:**

### **8. Crea un nuevo proyecto React utilizando Create React App.**

Para crear un nuevo proyecto React, abre tu terminal y ejecuta:

**npx create-react-app my-app**

Reemplaza my-app con el nombre que elijas para tu proyecto. Este comando configura una nueva aplicación React con una estructura básica.

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

Para iniciar el servidor de desarrollo, navega al directorio del proyecto y ejecuta:

```
bash
```

```
Copiar código
```

```
cd my-app
```

```
npm start
```

Esto abrirá la aplicación en tu navegador en <http://localhost:3000>.

## 5. Ejercicio:

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

- **src:** Contiene el código fuente de la aplicación, incluyendo componentes de React, hojas de estilo y archivos JavaScript. Es el lugar principal donde se desarrolla la lógica y la estructura de la aplicación.
- **public:** Contiene archivos estáticos que se sirven directamente al navegador, como `index.html` y otros recursos estáticos (imágenes, favicons, etc.). El archivo `index.html` es el punto de entrada donde React monta la aplicación.

## 6. Ejercicio:

### 11. Explica cómo JSX se diferencia del HTML tradicional.

JSX (JavaScript XML) se diferencia del HTML tradicional en que es una extensión de la sintaxis de JavaScript que permite escribir HTML dentro de JavaScript. Las principales diferencias son:

- **Expresiones JavaScript:** JSX permite incluir expresiones JavaScript dentro de llaves `{}` para insertar variables y resultados de funciones.

- **Atributos:** En JSX, algunos atributos HTML tienen nombres diferentes, por ejemplo, `class` se convierte en `className`, y `for` se convierte en `htmlFor`.
- **Elementos auto-cerrados:** JSX requiere que los elementos auto-cerrados como `<img />` o `<input />` estén correctamente cerrados.

## **7. Ejercicio:**

### **12. Define los roles principales en un equipo SCRUM.**

- **Product Owner:** Define y prioriza los requisitos del producto en el Product Backlog, y actúa como el enlace principal entre el equipo de desarrollo y los stakeholders.
- **Scrum Master:** Facilita el proceso Scrum, ayuda al equipo a seguir las prácticas ágiles, elimina impedimentos y asegura un entorno de trabajo efectivo.
- **Development Team:** Un equipo autoorganizado que trabaja para entregar incrementos de producto en cada Sprint. Son responsables de diseñar, desarrollar, probar y entregar el producto.

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

Un Sprint es un ciclo de trabajo fijo, típicamente de 1 a 4 semanas, durante el cual el equipo de desarrollo trabaja para completar un conjunto específico de tareas del Product Backlog.

#### **Planificación del Sprint:**

- **Revisión del Product Backlog:** El Product Owner presenta los ítems priorizados para el próximo Sprint.
- **Definición del Sprint Goal:** Se establece un objetivo claro para el Sprint.
- **Selección de ítems:** El equipo selecciona los ítems del Product Backlog que pueden ser completados durante el Sprint, basándose en la capacidad del equipo.

- **Creación del Sprint Backlog:** Se elabora una lista detallada de tareas que el equipo se compromete a completar para alcanzar el Sprint Goal.