

ACTIVIDAD N-º1 REACT

MARIA ISABEL ROMERO MEJIA

INSTRUCTOR: SANTIAGO GIRALDO

MUJERES DIGITALES

2024

1.EJERCICIO

HISTORIA DEL REACT Y DOS HITOS IMPORTANTES

React fue creado por Jordan Walker ingeniero de Software en Facebook, se lanzó por primera vez en 2013 como una librería de código abierto para construir interfaces de usuarios. Desde su creación React ha revolucionado el desarrollo de aplicaciones web al introducir un enfoque de usuario basado en componentes y una arquitectura declarativa para crear interfaces de dinámicas y eficientes.

DOS HITOS IMPORTANTES EN SU DESARROLLO

1.Introducción del virtual DOM (2013): Uno de los hitos claves en la evolución del react fue la introducción virtual DOM, este concepto permitió a react actualizar de manera eficiente solo los elementos que han cambiado en la interfaz en lugar de volver a renderizar toda la página. Esto mejoro enormemente el rendimiento y la experiencia del usuario en aplicaciones dinámicas.

2.React hooks (2019): Los hooks permitieron a los desarrolladores gestionar el estado y los efectos secundarios en componentes funcionales, lo que simplifico la estructura del código y permitió un enfoque más flexible y limpio para la gestión del estado de aplicaciones en react.

2.EJERCICIO

TRES VENTAJAS DE USAR REACT EN EL DESARROLLO DE APLICACIONES WEB

1.Componentización y reutilización: React permite dividir la interfaz de usuario en componentes independientes, reutilizables y aislados, esto hace que el código sea más organizado, mantenible y fácil de escalar.

2.Actualizaciones eficientes con el DOM virtual: React lo utiliza para actualizar solo las partes de la interfaz que han cambiado, en lugar de volver a renderizar toda la página, esto mejora significativamente el rendimiento.

3.Amplia comunidad y ecosistema: React cuenta con una comunidad grande y activa, lo que proporciona acceso a muchas bibliotecas, herramientas y soluciones que facilitan el desarrollo rápido y eficiente de aplicaciones.

¿COMO VIRTUAL DOM MEJORA EL RENDIMIENTO DE UNA APLICACIÓN?

Virtual DOM es una representación ligera del dom real que react mantiene en memoria, cuando llega a ocurrir un cambio de los datos react crea una nueva versión del dom virtual y la compara con la versión anterior, luego solo aplica las diferencias necesarias al dom real en lugar de actualizar todo el árbol de elementos, esto reduce la cantidad de operaciones del dom real, que tienden a hcer costosas en términos de rendimiento y resulta en actualizaciones más rápidas y eficientes

3 EJERCICIO

¿Qué ES UNA SINGLE PAGE APPLICATION(SPA)?

Es un tipo de aplicación web que carga una sola pagina html estática al principio y a medida que el usuario interactúa con la aplicación, las actualizaciones y cambios en el contenido se realizan dinámicamente sin necesidad de recargar toda la página, esto permite una experiencia de usuario más rápida y fluida ya que solo las partes necesarias del contenido se actualizan

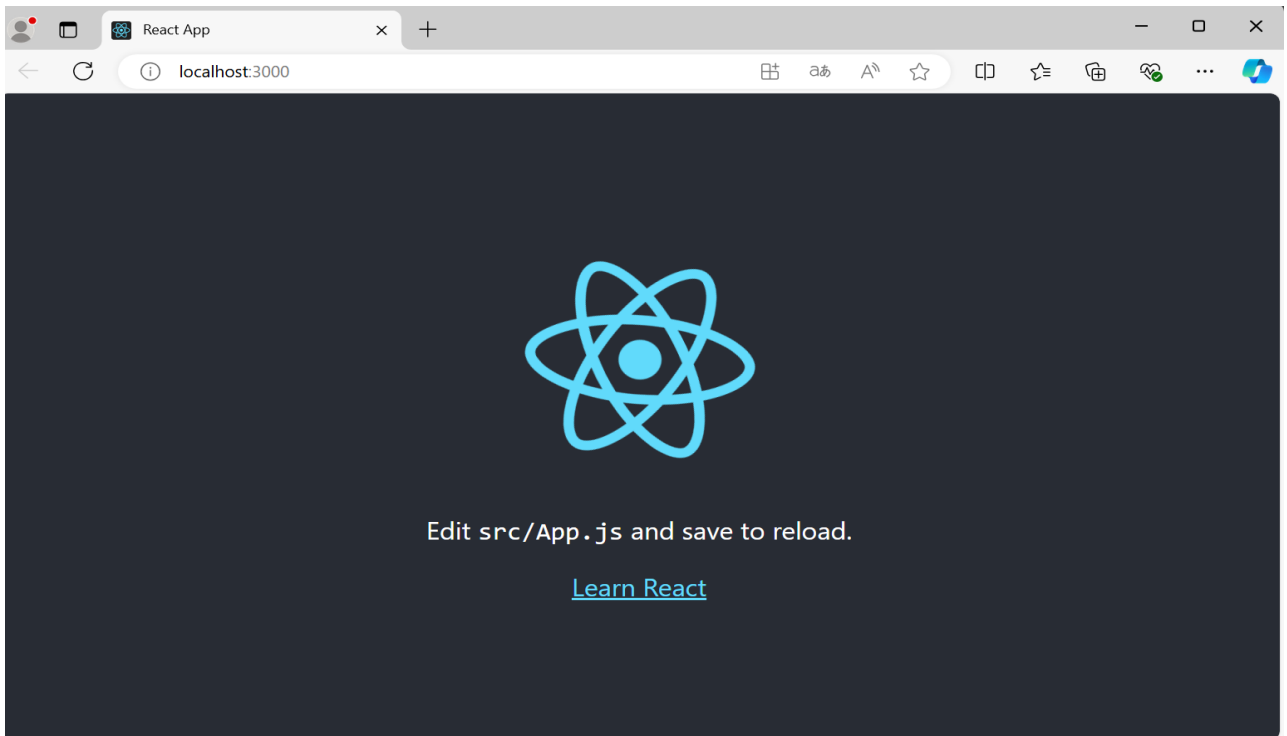
¿COMO REACT FACILITA LA CREACION DE UNA SPA?

La facilita a través de su enfoque de componentes y la actualización eficiente de la interfaz de usuario mediante virtual dom.

- **Rendimiento optimizado:** Gracias al Virtual DOM, React puede hacer actualizaciones parciales y eficientes en el contenido visible sin recargar toda la página.
- **Enrutamiento dinámico:** Con librerías como React Router, es posible gestionar la navegación dentro de la aplicación sin realizar peticiones al servidor para recargar páginas, lo que es clave en una SPA.
- **Componentes reutilizables:** React permite dividir la interfaz en componentes pequeños y reutilizables que gestionan su propio estado.

PROFESOR EL EJEMPLO DE ESTE PUNTO VA ESTAR EN APP.JS

4 EJERCICIO



5 EJERCICIO

En un proyecto React, las carpetas `src` y `public` tienen propósitos específicos:

- `src` (source): Esta carpeta contiene el código fuente de la aplicación. Aquí es donde se ubican los archivos y componentes de React que componen la lógica de la aplicación, como los archivos JavaScript/JSX, estilos, imágenes y otros recursos que se procesan y empaquetan al construir la aplicación. En esta carpeta suele estar el archivo `index.js` o `App.js`, que son puntos de entrada clave.
- `public`: La carpeta `public` contiene archivos estáticos que no necesitan ser procesados por React, como el archivo `index.html`, que es el contenedor principal donde React renderiza la aplicación. También se pueden almacenar aquí imágenes, iconos y otros archivos que no se modifican durante la ejecución de la aplicación. Cualquier archivo en esta carpeta es servido directamente sin modificaciones.

6 EJERCICIO

JSX (JavaScript XML) es una extensión de JavaScript que permite escribir elementos de la interfaz de usuario de forma similar a HTML, tiene algunas diferencias importantes con respecto al HTML tradicional :

Sintaxis dentro de JavaScript : JSX está integrado dentro de archivos JavaScript. Puedes escribir etiquetas similares a HTML directamente dentro del código JavaScript, lo que permite una interacción fluida entre la lógica y la presentación. En HTML tradicional, el código está separado en archivos HTML, CSS y JavaScript, lo que puede hacer que la interacción entre lógica y presentación sea menos directa.

Diferencias en la sintaxis : En JSX, las etiquetas deben estar cerradas, incluso las etiquetas vacías o de autocierre. Por ejemplo, `` o `
`, mientras que en HTML, algunas etiquetas pueden no requerir cierre explícito.

JSX usa `className` en lugar de `class`, porque `"class"` es una palabra reservada en JavaScript. Los atributos en JSX usan `camelCase` para las propiedades de estilo y eventos, por ejemplo, `onClick` en lugar de `onclick` y `style={{backgroundColor: "red"}}` en lugar de `style="background-color: red"`.

Expresiones JavaScript : En JSX, puedes incrustar expresiones de JavaScript dentro de llaves `{}`. Esto te permite añadir lógica directamente en las etiquetas JSX. En HTML tradicional, no puedes usar JavaScript de forma directa dentro de las etiquetas HTML, necesitas usar `<script>` o escribirlo en archivos externos.

7 EJERCICIO

Roles principales en un equipo SCRUM:

Product Owner (Propietario del Producto): Es responsable de maximizar el valor del producto y de gestionar el Backlog del Producto (lista priorizada de características o tareas). Define las prioridades, asegura que el equipo trabaje en las tareas más valiosas y sirve de enlace con los interesados (stakeholders).

Scrum Master: Es el facilitador del proceso Scrum. Asegura que el equipo siga las prácticas ágiles, elimina obstáculos que puedan interferir con el progreso del equipo y protege al equipo de distracciones. También ayuda al equipo a mejorar continuamente a través de retrospectivas.

Development Team (Equipo de Desarrollo): Es un equipo multidisciplinario encargado de desarrollar el producto. Está compuesto por desarrolladores, testers, diseñadores, etc. El equipo es autónomo y decide cómo abordar las tareas asignadas. Colaboran para completar las tareas comprometidas durante un sprint

¿Qué es un sprint y cómo se planifica?

Un sprint es un ciclo de trabajo dentro de Scrum que normalmente dura entre 1 y 4 semanas. Durante este período, el equipo de desarrollo se enfoca en completar un conjunto de tareas (o historias de usuario) que forman parte del Backlog del Producto. Al final de cada sprint, se entrega un incremento del producto que debe estar completamente terminado y potencialmente entregable.

Planificación del sprint: Revisión del Backlog del Producto: Durante la reunión de planificación del sprint, el equipo revisa las tareas o historias de usuario que se encuentran en el Backlog del Producto, priorizadas por el Product Owner.

Selección de tareas: El equipo de desarrollo selecciona las tareas más prioritarias que puede completar durante el sprint. Esto se basa en la capacidad del equipo y la complejidad de las tareas.

Definición del Sprint Goal: Se establece un objetivo del sprint (Sprint Goal), que describe el resultado deseado o el valor que se pretende lograr al final del sprint.

Creación del Sprint Backlog: Una vez que se seleccionan las tareas, se crea el Sprint Backlog, que contiene todas las tareas o historias de usuario que el equipo se compromete a completar durante el sprint. Estas tareas se dividen en sub tareas más pequeñas para facilitar su seguimiento.

Estimación y planificación: El equipo de desarrollo estima el esfuerzo necesario para cada tarea, ya sea en puntos de historia o en horas, para asegurarse de que las tareas elegidas son realistas para el tiempo disponible.