

Preguntas Programación III

1) Para que sirve Git?

Git es un sistema de control de versiones distribuido que se utiliza principalmente en el desarrollo de software. Sirve para realizar un seguimiento de los cambios en los archivos de un proyecto a lo largo del tiempo, lo que permite a los desarrolladores colaborar de manera eficiente en el desarrollo de software.

2) Diferencia entre GitHub y Git?

Git es un sistema de control de versiones distribuido. Es un software de línea de comandos que permite a los desarrolladores rastrear los cambios en los archivos de un proyecto a lo largo del tiempo.

GitHub es una plataforma de alojamiento de repositorios de código basada en la web que utiliza Git como su sistema de control de versiones subyacente. Proporciona una interfaz gráfica de usuario y servicios adicionales para facilitar la colaboración en proyectos de desarrollo de software.

3) Para que sirve el comando git commit -m "mensaje"?

Este comando sirve para confirmar los cambios que hemos realizado en los archivos y guardarlos definitivamente (a los cambios). También nos permite dejar un mensaje para detallar los cambios que se realizaron.

4) Para que sirve el comando git add .?

El comando 'git add .' es una forma abreviada de agregar todos los cambios realizados en el directorio actual y sus subdirectorios al área de preparación o staging area. Los cambios pueden incluir modificaciones, eliminaciones o adiciones de nuevos archivos.

5) ¿Que hace git checkout -b nombredelarama?

Nos crea una nueva rama y cambia a esta al mismo tiempo.

6) ¿Que hace git branch?

Nos muestra en que rama nos encontramos actualmente y también las demás ramas. También podemos crear una rama agregando un nombre luego de branch. Pero no nos moverá a esa nueva rama. Además podemos eliminar ramas con git branch -d nombredelarama y podemos renombrar ramas con git branch -m nombreakiguo nombrenuevo.

7) ¿Cual es la diferencia entre git reset --hard y git reset --soft.

Git reset --hard: Este comando realiza un reset "duro" al commit especificado. Restablece el HEAD (puntero de la rama actual) y la rama actual al commit especificado y descarta todos los cambios en el área de preparación y el directorio de trabajo. Los cambios no confirmados se pierden permanentemente y no se pueden recuperar.

En contraste git reset --soft, este comando realiza un reset "suave" al commit especificado. Restablece el HEAD y la rama actual al commit especificado, pero conserva los cambios en el área de preparación y el directorio de trabajo. Los cambios realizados después del commit especificado se mueven al área de preparación, lo que te permite realizar nuevos commits con los cambios.

8) ¿Que hace git merge? El comando git merge se utiliza en Git para fusionar cambios de una rama en otra rama. Permite combinar los cambios realizados en una rama secundaria

con una rama principal.

9) Para que sirve React?

React es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas y reutilizables. Su enfoque en los componentes, el rendimiento eficiente, el flujo de datos unidireccional y su compatibilidad con el desarrollo móvil a través de React Native lo convierten en una elección popular para el desarrollo de aplicaciones web modernas.

Aquí hay algunas de las principales características y usos de React:

Creación de componentes reutilizables:

Virtual DOM eficiente:

Flujo de datos unidireccional

10) Como se realiza un comentario dentro de un archivo JSX?

Se utilizan primero las llaves: {}, luego se utiliza /* */

11)Ventajas del DOM virtual

Mejor rendimiento y eficiencia.

Actualización eficiente

Simplifica la lógica de actualización

Facilita la creación de interfaces de usuarios reactivas

Compatibilidad multiplataforma.

12)Que es el renderizado en react

se refiere al proceso de generar la representación visual de los componentes en la interfaz de usuario. Cuando se desarrolla una aplicación con React, se definen componentes que contienen lógica y estructura para construir la interfaz de usuario. El renderizado es el mecanismo que toma estos componentes y los convierte en elementos visuales que se muestran en el navegador o en la aplicación móvil.

13) Que es un hook?

Un hook en React es una función especial que te permite agregar características de React a componentes funcionales. Los hooks permiten a los componentes funcionales tener un estado interno y utilizar características previamente reservadas para los componentes de clase, como el ciclo de vida y los contextos.

14) Que hace useState?

useState: Permite agregar estado local a un componente funcional. Con useState, puedes declarar una variable de estado y una función para actualizarla. Cada vez que actualices el estado, React volverá a renderizar el componente para reflejar los cambios

15)- como se obtiene el valor de un input o entrada de datos en react?

Se obtiene con el hook onChange

16)que es una props?

son un mecanismo para pasar datos de un componente padre a un componente hijo.

17) puedo pasar una props a un componente hermano?

No de manera directa.

18) como hago para subir a un componente padre una props?

A traves de una funcion callback.

19) useState devuelve un vector con cuantas posiciones?

Dos posiciones.}

20) que hay en la 1er posicion, q hay en la 2da posicion??

En la primera posicion, el estado actual y en la segunda posicion una funcion para cambiar ese estado.

21) - describa 5 nombres de eventos normalmente utilizados en react

onClick, onChange, onMouseOver, onSubmit, onKeyDown.

22) componentes cuantos tipos de componentes podemos crear

23) diferencia entre componente funcional y componente clase

La principal diferencia entre un componente funcional y un componente de clase en React es la forma en que están definidos y cómo manejan el estado y los ciclos de vida.

24) como es el comando para crear un proyecto en React?

Npx create-react-app nombreproyecto

25) Que es jsx? Explique brevemente

JSX es una extensión de sintaxis utilizada en React que permite escribir componentes de interfaz de usuario de forma declarativa utilizando una sintaxis similar a HTML.

28) para que sirve el evento onSubmit?

Para enviar los datos de un formulario.

29) para que sirve e.preventDefault()?

Para evitar el comportamiento predeterminado asociado a un evento. Por ejemplo, que al enviar un formulario, no se actualice la pagina.

30) para que sirve react-router-dom

Para crear rutas y enlaces dentro de una aplicacion.

31)que es BrowserRouter?

es un componente proporcionado por la biblioteca react-router-dom que se utiliza para envolver la aplicación y habilitar el enrutamiento en una aplicación de React. Actúa como el contenedor principal que maneja las rutas y la navegación dentro de la aplicación.

32)que hace useEffect()?

este hook nos permite definir efectos. Los efectos nos permiten ejecutar un trozo de código según el momento en el que se encuentre el ciclo de vida de nuestro componente.

33)Montaje actualizacion y desmontaje de un componente

Montaje: Esta fase ocurre cuando se crea un componente y se agrega al DOM por primera vez.

Actualización: Esta fase ocurre cuando un componente se actualiza debido a cambios en sus propiedades o en su estado.

Desmontaje: Esta fase ocurre cuando un componente se elimina del DOM. Durante el desmontaje.

34)Diferencias entre Routes y Route

El componente Route se utiliza para definir una ruta específica en la aplicación y asociarla a un componente de React.

El componente Routes se utiliza para agrupar múltiples rutas dentro de una estructura jerárquica.

36) Que hace useNavigate

Nos permite navegar entre rutas sin utilizar la etiqueta link, a traves de codigo.

37)Que hace useParams?

El hook useParams es una función proporcionada por la biblioteca react-router-dom. Se utiliza para obtener los parámetros de la URL en un componente de React.

38) cual es la diferencia entre bootstrap y react-bootstrap

bootstrap es una biblioteca de clases para CSS. React-bootstrap trabaja con componentes.