**CURSO DE REACT**

* -React es una biblioteca/framework osea un poco de los dos xd
* -React es declarativo.(queremos esto,pero no que hacer con eso)
* -JSX es como un html en javascript
* -Componentes son pequenos bloques
* -Aprende una vez y escribelo donde sea (multi-plataformas)

Herramientas:

[navegador](https://www.google.com/intl/es/chrome/)

[react developer tools](https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi/related?hl=es)

[editor de codigo](https://code.visualstudio.com/)

**Para hacer React** es necesario importar **React** y **ReactDOM** .

**React** es el análogo de **createElement** .

**ReactDOM** es el análogo de **appendChild** .

ReactDOM.render Toma 2 argumnentos:

ReactDOM.render(\_\_qué\_\_,\_\_dónde\_\_);

* El **qué** . Es qué queremos renderizar.
* El **dónde** . Es en dónde lo queremos renderizar.

Jsx = azúcar sintetica.

Jsx= es mucho mas legible y mas interpretativo.

Expresión= se interpreta y se evalua en javscript

React = javascript

Componentes: la clase con la que se hacen los elementos.

Se identifican los componentes con las preguntas: que elementos se repiten?, que elementos cumplen una función muy espeficica?.

Un componente funciona= pequenos bloques de legos, tienen vida nacen,crecen y desaparecen.

Mejor es llamarlo className.

Props argumentos de una función

State es importante en react .

Cuando se levanta estado se pone en el lugar mas cercano de todos los componentes que quieren compartir esa información.

React router:

React Router tiene 4 componentes principales:

Browser Router. Es necesario colocar el componente en la parte superior de la aplicación, lo que esté dentro de este va a servir como una Single Page App y les va a permitir al resto de las herramientas funcionar. Ya que fuera de él NO Funciona.

Route. Representa una dirección de Internet, el path es donde va a renderizar el componente y el componente es el componente que va a renderizar, usualmente es un a página.

Switch. Componente que nos sirve para representar 1 sola ruta de varias que podemos poner dentro del componente.

Link. Toma el lugar del elemento ancla, evita que se recargue completamente la página, actualiza la URL sin recargar la página completa.

Con reac router en lugar de tener una clase se puede tener una función

Los componentes cuando react los renderiza entran en escena

Cuando su estado cambia o recibe props diferentes se acrualizan.

Cuando cambian de pagina se desmontan.

Función timeout retorna un TimuotId

Para la aipi el login es importante para el usuario,error , y datos

El mejorlugar para inicializar una api es en el compinedidMount

Md5 le damos texto y te da hasgg de md5

Poling cada determinado tiempo buscan datos y se actualizan automáticamente

Los portales son para decirle que queremos renderizar y donde, es fuera de la aplicación

Hooks permite que las funciones tengan fitches, cunado se escribe un hook se empieza con use, no se puede hacer condiciones.

React.useState retorna dos argumentos.(un arreglo)

PREGUNTAS DE EXAMEN

1. componentDidUpdate recibe dos argumentos. ¿Cuáles son y en qué orden? prevProps, prevState
2. ReactDOM.createPortal nos permite hacer render de Un elemento fuera de la aplicación principal
3. Create-react-app nos ayuda a configurarlas siguientes herramientas: A
4. ¿Cuál de estas sería un uso correcto de React.useState? B
5. En pseudocódigo, decimos que ReactDOM.createPortal recibe los siguientes argumentos: C
6. El hecho de que React sea declarativo quiere decir que: Especificamos qué queremos, no cómo lo hacemos.
7. Cuando tenemos un setInterval o un setTimeout en nuestro componente ¿en dónde debemos removerlo? componentWillUnmount
8. ¿Por qué usamos un Portal para asegurar que los modales queden en un nodo fuera de la aplicación principal? C A
9. ¿Cuál de estos momentos presenta la mejor oportunidad para crear un componente? D
10. ¿Cuál es el método que React utiliza como señal de que ocurrió una actualización? B
11. ¿Cuál es el último método que se llama cuando se desmonta un componente? componentWillUnMount()
12. ¿Cuál prop es importante cuando renderizamos una lista de componentes? key
13. ¿Cuál es el feature principal de un Switch? A
14. ¿Cuál de las siguientes no es JSX válido? D, B
15. ¿Qué evento se dispara cuando un usuario escribe en una caja de texto? A
16. En pseudocódigo, decimos que ReactDOM.render recibe los siguientes argumentos: D
17. En React hay un prop que representa el contenido entre las etiquetas de un elemento. ¿Cómo se llama? props.children
18. ¿Cuál es la diferencia entre <Navbar /> y Navbar? A B
19. ¿Cuál es la diferencia entre <Link> y <a >? <Link> evita que se cargue completamente la página, mientras que <a> sí
20. En algunas ocasiones podemos definir dos sub-estados para los datos que obtenemos de una peticion. Esto son: con datos, vacío
21. Una regla de los hooks es que el su nombre debe comenzar con esta palabra: C
22. ¿Cuál es el único momento en el que podemos escribir directo a this.state? D
23. Hay ocasiones en las cuales hay un estado que tenemos que compartir con dos o más componentes. ¿A dónde debemos levantar ese estado? Al componente en común más cercano a los componentes que lo usan
24. Hay ocasiones en las cuales hay un estado que tenemos que compartir con dos o más componentes. ¿A dónde debemos levantar ese estado? B
25. JSX es syntactic sugar para: A
26. ¿Cuáles son los tres estados de una petición? A
27. ¿Cuándo es buen momento para dividir un componente en contenedor y componente presentacional? A
28. ¿Cuál de estos es el mejor lugar para iniciar una llamada HTTP? C
29. ¿Cuál de estos htt evento válido? B