Apuntes de React M2

-Encadenamiento Opcional /Optional Chaining (?.)-

es una característica poderosa de React que puede facilitar mucho el trabajo con objetos profundamente anidados. Cuando se trata de objetos de datos de gran tamaño procedentes de API externas, puede resultar complicado acceder a las propiedades de los objetos sin comprobaciones explícitas de valores nulos o indefinidos. Afortunadamente, ECMAScript 2020 introdujo el encadenamiento opcional, que le permite acceder de forma segura a estas propiedades sin riesgo de errores.

-REACT FORMS-

.Crear formulario funcional con React
.Diferenciar formularios controlados y no controlados
.Herramientas claves para construir formularios

Formularios:
Hay diferentes tipos de formularios en la web
1.De contacto,
2.Inicio de sesión "Log in",
3.De compra,
4.De comentarios

Existen 2 clases de formularios:
1.Controlled / Controlados & No controlados

"Estaremos trabajando con el tipo de formulario Controlado por las siguientes ventajas"

.El valor del input tiene un bind al estado del componente
.Recomendado por los creadores de React es decir Facebook o Meta
.Sigue los patrones de React
.Es predecible

-Key-

React tiene un atributo llamado key
Este le permite identificar de manera única a los elementos renderizados de una lista.

Riesgo de no utilizar dicho atributo key en formularios en React

.Rendimiento deficiente
.Errores de interfaz
.Problemas de estado y lógica

Ejemplo en código usando el atributo key

```
funtion Form ({language}){
   if (language === en){
         return (
                  <form>
        <input
          key = 'nombre'
          type = 'text'
          name = 'name'
        </input>

          </form>
   );
  }
};
```

Resumen

.Formulario --> Existen dos tipos de formularios los controlados y los no controlados el cual los controlados tiene numerosas ventajas para la gestión de información de un usuario

.Key --> Es un atributo que le permite a React diferenciar elementos renderizados de una misma lista

¿Cuál de estos NO es
un selector del DOM?

1. getElementById
2. getElementByClassName
3. querySelectorAll

¿Qué parámetros recibe el método addEventListener() ?

1. Una función de callback
2. Un evento y un selector
3. Un evento y una función

¿Cuál de los siguientes es un preprocesador CSS?

1. LESS
2.Bootstrap
3.Material UI

¿Qué variable es block scoping
y reasignable?

1.var
2.let
3.const

¿Cuál de estos NO corresponde
a un método HTTP?

1. GET
2. PATCH
3. CREATE

¿Cuáles son las dos etapas
de un bundler?

1. Verificación de enrutamiento y compilación
2. Resolución de dependencias y encapsulamiento
3. Creación del bundler.js y el deploy

¿Cuál de los siguientes permite transformar código en formato JSX a JS?

1. React
2. Webpack
3. Babel

_____ NO es una característica de React

1. bidireccional
2. declarativo
3. componentizado

¿Cuál de los siguientes permite compartir información entre un componente padre y su hijo?
1. props
2. createElement
3. state

___ permite a React identificar qué elementos han cambiado

1. props
2. key
3. state

Para controlar formularios en React, usamos...

1. No hay que usar nada, todos son controlados
2. El state del componente
3 EL DOM

¿Cuál de los siguientes hace parte del ciclo de vida de un componente?

1. componentDidMount
2. componentUpdate
3. componentDidUnmount

Este hook nos permite obtener el pathname donde estamos ubicados...

1. useParams
2. useLocation
3. useHistory

Qué componente proporciona el acceso al store de Redux?

1. Provider
2. BrowserRouter
3. App

¿Cuáles corresponde al flujo de Redux?

1. Eventos, Actions, Reducer, Store, State
2. Actions, Eventos, Reducer, Store, State
2. Eventos, Store, State, Reducer, Actions

mapStateToProps() permite traer información del estado global como props del componente

1. Falso
2. Verdadero

¿Qué parámetro recibe mapDispatchToProps()?

1. dispatch
2. state
3. Ninguno

El hook _____ es equivalente a mapStateToProps()

1. UseSelector
2. UseDispacth
3. UseReducer