

Resumen de Temas en Desarrollo Web y Móvil

Diego Muñoz

27 de noviembre de 2024

En este curso vimos varios conceptos clave en desarrollo web y móvil, desde **React** hasta cómo llevar una app de la web a **React Native**.

1. React y Hooks

React

- **React:** Es una librería para hacer interfaces de usuario usando componentes.
- **JSX:** Es una forma de escribir HTML dentro de JavaScript.
- **Virtual DOM:** React usa un sistema eficiente para actualizar solo lo que cambia en la interfaz.

Sin JSX

```
import React from 'react';

const Greeting = () => {
  return React.createElement(
    'div',
    null,
    React.createElement('h1', null, 'Hello, welcome to my app!'),
    React.createElement('p', null, 'Enjoy your stay.')
  );
}

export default Greeting;
```

Con JSX

```
import React from 'react';

const Greeting = () => {
  return (
    <div>
      <h1>Hello, welcome to my app!</h1>
      <p>Enjoy your stay.</p>
    </div>
  );
}

export default Greeting;
```

Hooks

- **useState:** Para manejar el estado en los componentes funcionales.
- **useEffect:** Para manejar efectos secundarios como fetch o suscripciones.
- **Custom Hooks:** Funciones reutilizables para lógica que puede ser compartida entre varios componentes.

useState

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

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

  return (
    <div>
      <h1>Contador: {count}</h1>
      <button onClick={() => setCount(count + 1)}>Incrementar</button>
    </div>
  );
};

export default Counter;
```

2. Gestión de Estado con Context

- **Context API:** Es una forma de compartir datos entre componentes sin tener que pasar props por todos lados.
- **useContext:** Un hook que nos ayuda a acceder a los valores de un contexto de forma fácil.

Context API

```
import { createContext, useState, useContext } from 'react';

const CountContext = createContext();

const CountProvider = ({ children }) => {
  const [count, setCount] = useState(0);
  const increment = () => setCount(count + 1);

  return (
    <CountContext.Provider value={{ count, increment }}>
      {children}
    </CountContext.Provider>
  );
};
```

useContext

```
const Counter = () => {  
  const { count, increment } = useContext(CountContext);  
  
  return (  
    <div>  
      <h1>Contador: {count}</h1>  
      <button onClick={increment}>Incrementar</button>  
    </div>  
  );  
};
```

3. Navegación en React

React Router

- **React Router:** Es la librería para manejar rutas en aplicaciones de una sola página (SPA).
- **Rutas dinámicas:** Creamos rutas con parámetros para hacerlas más flexibles.

Expo Router (React Native)

- **Expo Router:** Es la misma idea que React Router, pero adaptada para apps móviles con Expo.

4. APIs y Autenticación

Consumo de APIs con fetch

- **fetch**: Es la forma más común de hacer solicitudes HTTP en JavaScript.
- **Manejo de respuestas**: Usamos promesas (`.then()`) para trabajar con los datos de las APIs.

Autenticación con OAuth2 y JWT

- **OAuth2**: Es un protocolo que permite autenticar sin compartir las credenciales de usuario.
- **JWT**: Es un token que usamos para verificar la identidad del usuario de forma segura.

5. React Native

- **React Native:** Nos permite crear aplicaciones móviles nativas usando React.
- **Expo:** Es una herramienta que hace más fácil el desarrollo y despliegue de apps con React Native.

6. Despliegue

Despliegue en Web

- **Optimización para producción:** Preparamos el build para que la app cargue rápido y esté optimizada.
- **Deploying React Apps:** Subimos nuestras aplicaciones React a plataformas como netlify.

Despliegue en Móvil

- **Expo y React Native CLI:** Creación de APKs e IPAs para Android y iOS.
- **Publicación en Play Store/App Store:** Se mencionó que se puede publicar una app en las tiendas de aplicaciones.

- **React:** Librería para interfaces de usuario.
- **Hooks:** Nueva forma de manejar estado y efectos secundarios.
- **Context:** Para compartir datos entre componentes.
- **React Router:** Para manejar rutas en React.
- **Expo:** Herramienta para desarrollo de apps móviles con React Native.
- **React Native:** Desarrollo de apps móviles nativas.
- **Despliegue:** Cómo subir nuestras apps a producción.
- **APIs y Autenticación:** Consumo de APIs y autenticación segura.