Crear una aplicación web corporativa en ReactJS utilizando componentes, rutas y eventos:

Paso 1: Configuración del entorno

1. Asegúrate de tener Node.js y npm (Node Package Manager) instalados en tu sistema.
2. Abre una terminal y crea una nueva carpeta para tu proyecto.
3. Navega hasta la carpeta del proyecto en la terminal y ejecuta el siguiente comando para crear un nuevo proyecto de React:

```
npx create-react-app nombre-de-tu-proyecto
```

4. Una vez que se complete la instalación, ingresa al directorio del proyecto:

```
cd nombre-de-tu-proyecto
```

5. Ejecuta el siguiente comando para iniciar la aplicación:

```
npm start
```

Esto abrirá la aplicación en tu navegador en la dirección http://localhost:3000.

Paso 2: Creación de componentes

1. Dentro del directorio del proyecto, navega hasta la carpeta src.
2. Crea una nueva carpeta llamada components.
3. Dentro de la carpeta components, crea un nuevo archivo llamado Header.js y añade el siguiente código:

```
import React from 'react';

const Header = () => {
  return (
    <header>
      <h1>Web Corporativa</h1>
    </header>
  );
};

export default Header;
```

4. Crea otro archivo llamado Footer.js en la misma carpeta y añade el siguiente código:

```
import React from 'react';

const Footer = () => {
  return (
    <footer>
      <p>Derechos de autor &copy; {new Date().getFullYear()}</p>
    </footer>
  );
};

export default Footer;
```

Paso 3: Configuración de rutas

1. Instala la dependencia react-router-dom ejecutando el siguiente comando:

```
npm install react-router-dom
```

2. Abre el archivo src/App.js y reemplaza su contenido con el siguiente código:

```
import React from 'react';
import { BrowserRouter as Router, Switch, Route } from 'react-router-dom';import
Header from './components/Header';
import Footer from './components/Footer'; import
Home from './components/Home'; import About
from './components/About'; import Contact from
'./components/Contact';

const App = () => {
  return (
    <Router>
      <Header />
      <Switch>
        <Route exact path="/" component={Home} />
        <Route path="/about" component={About} />
        <Route path="/contact" component={Contact} />
      </Switch>
      <Footer />
    </Router>
  );
};

export default App;
```

3. Crea tres nuevos archivos en la carpeta components: Home.js, About.js y Contact.js.
4. En cada uno de estos archivos, puedes añadir el contenido que desees para las páginas correspondientes.

Paso 4: Añadir eventos

1. Abre el archivo src/components/Header.js y modifícalo de la siguiente manera:

```
import React from 'react';

const Header = () => { const
  handleClick = () => {
    console.log('¡Haz hecho clic en el botón!');
  };

  return (
    <header>
      <h1>Web Corporativa</h1>
      <button onClick={handleClick}>Haz clic aquí</button>
    </header>
  );
};

export default Header;
```

2. Ahora, cuando hagas clic en el botón en el encabezado, verás el mensaje en la consola del navegador.

Has creado una aplicación web corporativa en ReactJS con componentes, rutas y eventos. Puedes continuar añadiendo más funcionalidades y estilos según tus necesidades. Recuerda que este es soloun ejemplo básico para mostrarte los conceptos fundamentales.

Añadir un componente formulario

Agregar un componente de formulario a tu aplicación web corporativa en ReactJS:

Paso 1: Creación del componente de formulario

1. Dentro de la carpeta components, crea un nuevo archivo llamado Form.js.
2. Abre el archivo Form.js y añade el siguiente código:

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

const Form = () => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [message, setMessage] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    // Lógica para manejar el envío del formulario
    console.log('Formulario enviado');
  };

  return (
    <form onSubmit={handleSubmit}>
      <h2>Formulario de contacto</h2>
      <div>
        <label htmlFor="name">Nombre:</label>
        <input
          type="text"
          id="name"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
      </div>
      <div>
        <label htmlFor="email">Email:</label>
        <input
          type="email"
          id="email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
      </div>
```

```
    <div>
      <label htmlFor="message">Mensaje:</label>
      <textarea
        id="message"
        value={message}
        onChange={(e) => setMessage(e.target.value)}
      ></textarea>
    </div>
    <button type="submit">Enviar</button>
  </form>
 );
};

export default Form;
```

Paso 2: Incorporación del componente de formulario

1. Abre el archivo src/App.js.
2. Importa el componente de formulario añadiendo la siguiente línea al comienzo del archivo:

```
import Form from './components/Form';
```

3. Agrega el componente de formulario en la sección de rutas dentro del componente App de la siguiente manera:

```
<Route path="/contact" component={Form} />
```

Paso 3: Visualización del formulario

1. Guarda los cambios en el archivo src/App.js.
2. En tu navegador, asegúrate de que la aplicación esté en funcionamiento (ejecutando npm start en la terminal).
3. Navega hasta la página de contacto en la aplicación web (por ejemplo, http://localhost:3000/contact).
4. Verás el formulario de contacto que acabamos de crear. Prueba a ingresar algunos datos y envía el formulario. En la consola del navegador, deberías ver el mensaje "Formulario enviado".

Ahora tienes un componente de formulario funcional en tu aplicación web corporativa en ReactJS. Puedes personalizar el formulario y agregar la lógica necesaria para procesar los datos del formulariosegún tus requisitos.

Añadir un componente con props

Ejemplo de cómo añadir un componente con props a tu aplicación web corporativa en ReactJS:

Paso 1: Creación del componente con props

1. Dentro de la carpeta components, crea un nuevo archivo llamado WelcomeMessage.js.
2. Abre el archivo WelcomeMessage.js y añade el siguiente código:

```
import React from 'react';

const WelcomeMessage = (props) => {
  return (
    <div>
      <h2>Bienvenido, {props.name}!</h2>
      <p>{props.message}</p>
    </div>
  );
};

export default WelcomeMessage;
```

Paso 2: Uso del componente con props

1. Abre el archivo src/App.js.
2. Importa el componente WelcomeMessage añadiendo la siguiente línea al comienzo del archivo:

```
import WelcomeMessage from './components/WelcomeMessage';
```

3. Agrega el componente WelcomeMessage en la sección de rutas dentro del componente App de la siguiente manera:

```
<Route exact path="/" render={() => <WelcomeMessage name="Usuario" message="Bienvenido anuestra web corporativa" />} />
```

En este ejemplo, el componente WelcomeMessage se renderizará en la ruta raíz ("/") y pasará dos props: name con el valor "Usuario" y message con el valor "Bienvenido a nuestra web corporativa".

4. Puedes utilizar el componente WelcomeMessage en cualquier otra ruta o componente que desees, siguiendo la misma sintaxis de paso 3.

Paso 3: Visualización del componente con props

1. Guarda los cambios en el archivo src/App.js.
2. Asegúrate de que la aplicación esté en funcionamiento en tu navegador (ejecutando npm starten la terminal).
3. Navega hasta la ruta raíz de la aplicación (por ejemplo, http://localhost:3000/).
4. Verás el mensaje de bienvenida que hemos configurado en el componente WelcomeMessage,utilizando los valores de props que le pasamos.

Ahora tienes un componente con props en tu aplicación web corporativa en ReactJS. Puedes personalizar y utilizar props adicionales según tus necesidades, lo que te permitirá reutilizar componentes en diferentes partes de tu aplicación.