ThePower business school

🇪🇸 Componentes ReactJS

Introducción

Los componentes de ReactJS permiten separar nuestra interfaz de usuario en piezas independientes reutilizables. Para entenderlo mejor se puede extrapolar su uso al que tienen las funciones de JavaScript, siendo estas la forma más sencilla de declarar un componente:

```
import React from 'react';

const Hello = () => {
  return <h1>Hello there!</h1>;
}
```

Este componente se podrá utilizar a través de la etiqueta `<Hello/>`, y creará un `<h1>` con el texto "Hello there!" cada vez que lo utilicemos en nuestra aplicación.

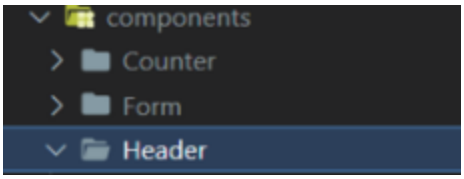
Carpeta components

En el interior de nuestra carpeta `src` crearemos una carpeta llamada `components` en la cual almacenar cada componente que creemos para renderizarlos desde `App.jsx`.

Además de esto podemos refactorizar nuestro flujo de construcción y el archivo `App.jsx` por defecto para tener un espacio de trabajo más claro y limpio eliminando elementos que, en nuestro caso, no vamos a utilizar.

Dentro de la carpeta `components` crearemos subcarpetas con los componentes generados y sus archivos necesarios, lo cual conformará una estructura clara de todos ellos y una manera limpia de exportarlos a el componente `App`.

Tal y como vemos en la imagen hemos creado 3 componentes: `Counter`, `Form`, `Header` , los cuales incluyen un archivo `.jsx` .



La estructura interna de cada componente es idéntica a la anteriormente vista en `App`, quedando nuestro código de la siguiente forma en este caso:

```
import React from "react";

const Header = () => {

  return (
    <header className="header">
           <h1>My name</h1>
    </header>
  );
};

export default Header;
```

Una vez creado nuestro componente tendremos que exportarlo para poder importarlo en el principal a través del `export default`, pero al tener diversos componentes sería conveniente volcarlos en un mismo archivo para poder reutilizar una misma ruta para todos ellos. Para ello utilizaremos un archivo `index.js` situado en la carpeta `components` de la siguiente forma:

```
import Header from "./Header/Header";
import Counter from "./Counter/Counter";
import Form from "./Form/Form";
export { Header, Counter, Form };
```

Tal como indica el nombre del archivo, en este caso nuestro `index.js` servirá de índice para los componentes, importando todos los que tengamos en el mismo y exportándolos en una misma línea. Esto nos será muy útil a la hora de importarlos en nuestro componente principal e insertarlos:

```
import React from "react";
import { Header, Counter, Form } from "./components";

const App = () => {
  return (
    <div className="App">
      <Header />
      <Form />
      <Counter />
    </div>
  );
};

export default App;
```

De esta forma ya tendríamos una estructura básica de una aplicación de ReactJS con distintos componentes.

Mini-ejercicio

Ha llegado el momento de ponerse a trabajar con ReactJS, para ello os proponemos una pequeña práctica que os ayude afianzar el concepto de componentes.

1. Crea una aplicación de ReactJS con vite → name: components-basics.
2. Crea tu carpeta de `components` dentro de `src`.
3. Realizamos algunos componentes de ReactJS:
    a. Componente Title ⇒ Crea un componente Title que retorne un `<h1>` con el texto “Welcome to Components ReactJS”.
    b. Componente SubTitle ⇒ Crea un componente que retorne un `<h2>` con el texto “This is a example components with ReactJS”.

c. Componente Image ⇒ Crea un componente que retorne un `<image>` y en su interior podéis seleccionar cualquier imagen de ReactJS.

d. Componente Paragraph ⇒Crea un componente que retorne un `<p>` con el texto que quieras en su interior.

4. Estila cada uno de ellos haciendo uso de CSS Modules → hoja de estilo asociada al componente.
5. Exporta los componentes en un `index.js` e importalos en `App.jsx`.
6. Comprueba que la visualización es correcta.