

Componentes

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. Componentes
2. ¿Para qué sirven?
3. Características
4. Componentes de clase vs. funcionales

1 | Componentes



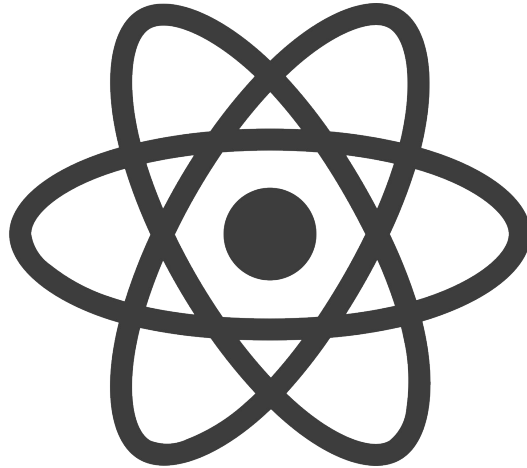
Un componente es una clase o una función que devuelve HTML a través de sintaxis JSX. Una vez definido, este se inscribe dentro del árbol de componentes de la aplicación. Aunque React tiene un fuerte enfoque sobre los componentes como una biblioteca, el ecosistema circundante lo convierte en un marco flexible.



2 | ¿Para qué sirven?

¿Para qué sirven?

Los componentes permiten separar la interfaz de usuario en piezas independientes, reutilizables y pensar en cada pieza de forma aislada.



3 | Características

Características de los componentes

Anidación

1

Un componente puede ser mostrado dentro de otro.

Reusabilidad

2

Un componente bien construido puede reutilizarse en cualquier aplicación.

Customización

3

Podemos customizar su contenido según donde lo usemos.

Ciclo de vida

4

Desde su registro montaje hasta que es “desmontado”.

4 | Componentes de clase vs. funcionales

Componentes de clase

La primera forma de declarar un componente es creando el mismo como una clase de Javascript. Podemos ver un ejemplo en el siguiente código:

```
JS index.js M X
src > JS index.js > MyComponentClass > render
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3
4  class MyComponentClass extends React.Component {
5    render() {
6      return <h1>Hola mundo de las Clases</h1>;
7    }
8  };
9
10 ReactDOM.render(
11   <MyComponentClass />,
12   document.getElementById('root')
13 );
14
```

Componentes de clase

Para crear un componente de clase debemos extender el mismo de **React.Component**, que es la clase “padre” que React nos proporciona, la que actúa como una especie de “molde” para que nuestros componentes puedan registrarse y utilizarse dentro de nuestra aplicación.

Por su parte, nuestro componente contará con un método “render”, dentro del cual deberemos retornar el JSX que deseamos renderizar cada vez que utilicemos dicho componente. Si bien existen otras particularidades asociadas a este tipo de componentes, la anterior es la sintaxis básica que deberemos emplear para crear un componente de clase.

Componentes funcionales

Tal y como su nombre indica, un componente funcional no es otra cosa que una función que retorna el JSX que deseamos renderizar para cada componente, tal como se muestra en el siguiente ejemplo:

```
JS index.js M X
src > JS index.js > [🔗] MyComponentFunctional
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3
4  const MyComponentFunctional = () => <h1>Hola mundo de las Funciones</h1>;
5
6  ReactDOM.render(
7    <MyComponentFunctional />,
8    document.getElementById('root')
9  );
10
```

En este caso, no necesitamos hacer uso de ninguna 'super clase' o método para crear nuestro componente. Con solo declarar nuestra función y retornar lo que deseamos renderizar es suficiente.

DigitalHouse>
Coding School