



Universidad Politécnica de Madrid

React

Componentes

Enrique Barra Arias
Álvaro Alonso González

- Conceptualmente, los componentes son elementos que aceptan argumentos de entrada arbitrarios (llamados "props") y devuelven elementos de React que describen lo que debería aparecer en la pantalla
- La forma más simple de definir un componente es escribir una **función de JavaScript**

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

```
const Welcome = (props) => {  
  return <h1>Hello, {props.name}</h1>;  
}
```

- También se puede usar una clase ES6 para definir un componente

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

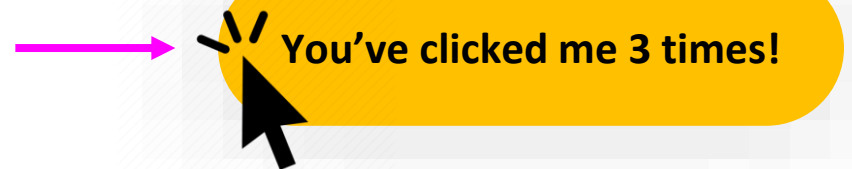
- **Alta cohesividad**, los elementos que forman un componente están correlacionados
- **Bajo acoplamiento**, si modificamos un componente, como tiene una API, no tenemos que modificar el resto.
- **Fácil de probar**. ¡Bienvenidos a las pruebas unitarias!
- **Declarativos**, los usamos y les decimos los datos con los que tienen que trabajar pero no les decimos cómo trabajar con ellos
- **Se definen con JSX** (aunque podríamos usar sólo JS)

- En resumen, un componente es sólo una función que devuelve un HTML para renderizar o una clase que extiende la clase `React.Component` con al menos un método de renderización (pintado).
- El elemento de React a renderizar puede ser una o varias etiquetas HTML (por ejemplo un `div` o un `h1`) u otro componente de React.
- No son sólo plantillas (es más similar a las vistas “ejs”), además tienen una API, y pueden tener múltiples métodos y distintas funcionalidades.

```
class ButtonCounter extends React.Component {
  constructor() {
    super()
    this.state = { clicks: 0 }
    this.handleClick = this.handleClick.bind(this)
  }

  handleClick() {
    this.setState({ clicks: ++this.state.clicks })
  }

  render() {
    return (
      <Button
        onClick={this.handleClick}
        text={`You've clicked me ${this.state.clicks} times!`}
      />
    )
  }
}
```



```
export default function SaludoSimple(props){  
  return(<div>  
    <h1>Esto es un titular</h1>  
    <p>Hola Enrique, qué tal</p>  
  </div>)  
}
```

- Lo que se devuelve en el return será lo que se pinte en la pantalla

- Se renderiza así en index.js:

```
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
  <React.StrictMode>  
    <SaludoSimple />  
  </React.StrictMode>  
);
```

- Se renderiza así en en otro componente “padre”, en su return:

```
return (<div>  
  <h1>Hola</h1>  
  <SaludoSimple/>  
  <h2>Adios...</h2>  
</div>)
```

```
import React from "react";

export default class SaludoSimpleClase extends React.Component {
  render() {
    return(<div>
      <h1>Esto es un titular</h1>
      <p>Hola Enrique, qué tal</p>
    </div>)
  }
}
```

- Tiene que tener un método render() que es el que devuelve lo que se pintará por pantalla
- Se renderiza igual que los componentes función, usando su nombre

[https://github.com/REACT-UPM/ejemplos simples](https://github.com/REACT-UPM/ejemplos_simples)