



Elementos



Elementos - ¿Qué es?

Los elementos son los bloques más pequeños de las aplicaciones de React.

Un elemento describe lo que quieres ver en la pantalla.

Son representaciones de las etiquetas de HTML que ya conocemos, **pero ojo, no es HTML!** se escriben utilizando una nueva sintaxis denominada JSX que veremos más adelante.



```
const element = <h1>Hello, world</h1>;
```

JSX ⚡

JSX - ¿Qué es?

JSX: una forma de “mezclar” HTML y JS

```
const element = <h1>Hola Senpai!</h1>;
```

JavaScript?

HTML?

JSX nos va a permitir **mezclar lógica con HTML**, lo que **simplifica la manipulación del DOM**.

El siguiente código crea un elemento div con el mensaje adecuado para el producto a comprar.

¿Cómo lo harían con lo que sabemos de JS hasta el momento?

¿Utilizando `querySelector` y la propiedad `.textContent`?

JSX - Ejemplo



```
const producto = {  
  nombre: 'Camiseta Uruguay',  
  cantidad: 2,  
  precio: 1000  
};
```

// Podemos insertar JavaScript entre llaves {}, por ejemplo para hacer un cálculo:

```
const mensaje = <div>El total de la compra sería {producto.cantidad * producto.precio}</div>;
```

Componente ⚡

Componente - ¿Qué es?

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

Conceptualmente, **los componentes son como las funciones de JavaScript**. Aceptan entradas arbitrarias (llamadas “props”) y retornan elementos de React que describen lo que debe aparecer en la pantalla.

Los componentes pueden estar **compuestos** por elementos o **otros componentes**.

Componente - Creando nuestro primer componente

Cosas a tener en cuenta al momento de crear nuestro componente:

- Los componentes **se definen con funciones**, si bien hay otras formas, las funciones es la más utilizada y la recomendada hoy en día.
- El **nombre del componente debe comenzar con Mayúscula**, para evitar pisar los “elementos”, que si están predefinidos con minúscula.
- **La función siempre debe retornar un componente o un elemento**, no puede retornar varios nodos, en el caso que queramos retornar varios debemos englobar con un padre.

Componente - Creando nuestro primer componente



```
function Header() {  
  return (  
    <div>  
      <h1>Mi APP de react</h1>  
    </div>  
  );  
}
```

Componente - Usando nuestro componente



```
ReactDOM.render(  
  <Header titulo="Mi app"></Header>,  
  document.getElementById('root')  
)
```

Props ⚡

Props - ¿Qué es?

Las **props** son simplemente la forma que tenemos para **pasarle información al componente**, y lograr que el componente logre mejor su cometido de ser una pieza reutilizable.

Las props **se indican como si fueran atributos HTML**, y react luego arma **un objeto** con estos valores y **se lo pasa a la función del componente**.

Props - Ejemplo

```
function Header(props) {  
  return (  
    <div>  
      <h1>{props.titulo}</h1>  
    </div>  
  );  
}  
  
const ejemplo = <Header titulo="Mi APP" />  
  
//Output  
//<div>  
//  <h1>Mi APP</h1>  
//</div>
```

LINKS



LINKS

- **React**

- <https://es.reactjs.org/docs/components-and-props.html>
- <https://es.reactjs.org/docs/rendering-elements.html>
- <https://es.reactjs.org/docs/introducing-jsx.html>



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE