

# REACT

**Componentes (Components):** Pieza encapsulada de la Interfaz del Usuario que es reutilizable y contiene su propia lógica. En react se crean usando funciones o clases y las funciones en React se declaran con la nomenclatura PascalCase.

```
function MiComponente() {  
  return <h1>Hola, mundo!</h1>;  
}
```

**Anidación de componentes (Nest Components):** Insertar componentes dentro de otros usando keywords como [export](#), [export default](#) e [import](#).

Declaramos la función

```
// En el archivo MiComponente.js  
  
import React from 'react';  
  
function MiComponente() {  
  return <h1>Hola, mundo!</h1>;  
}  
  
export default MiComponente;
```

La anidamos a otro componente

```
// En el archivo App.js  
  
import React from 'react';  
import MiComponente from './MiComponente';  
  
function App() {  
  return (  
    <div>  
      <MiComponente />  
    </div>  
  );  
}  
  
export default App;
```

**JSX (JavaScript + XML):** Sintaxis usada junto con React, da mayor legibilidad al código. [Ejemplo](#).

Fragment (<Fragment>, </Fragment>): Todo lo que devuelva una función de react debe estar dentro de una etiqueta padre, React no permite devolver varias etiquetas o elementos sin tener otro por encima.

La [versión abreviada](#): (<>, </>).

[Diferencias entre Fragment y Div.](#)