

¿Qué es React?

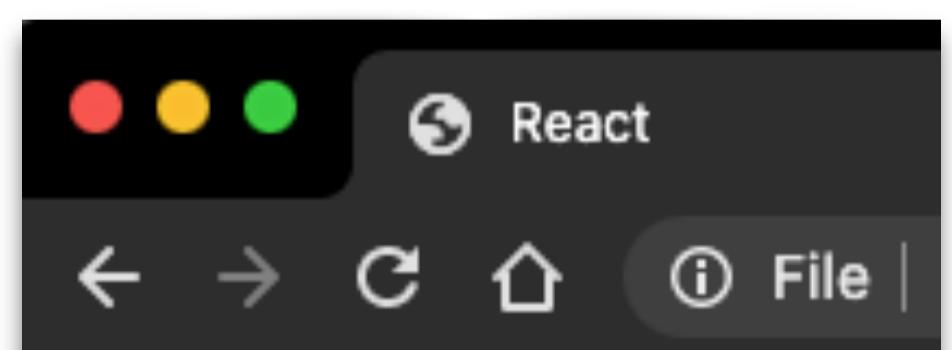


- **Librería**
- **Declarativa**
- **Eficiente**
- **Predecible**
- **Componentes**
- **Server-side con Node**
- **Aplicaciones móviles con React Native**

¿Cómo luce el código de React?

```
<body>  
  
  <div id="root"></div>
```

```
const divRoot = document.querySelector('#root');  
  
ReactDOM.render( <h1>Hola Mundo</h1> , divRoot );
```

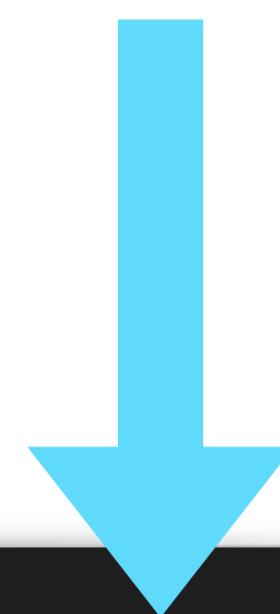
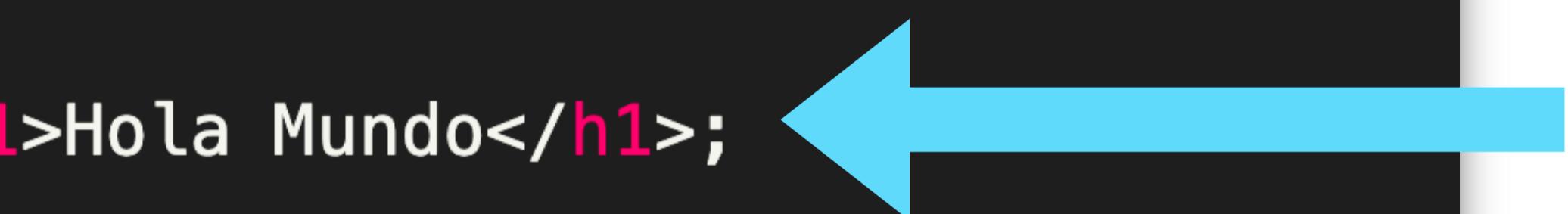
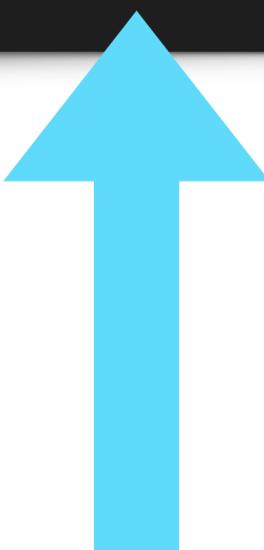


Hola Mundo

Nota:

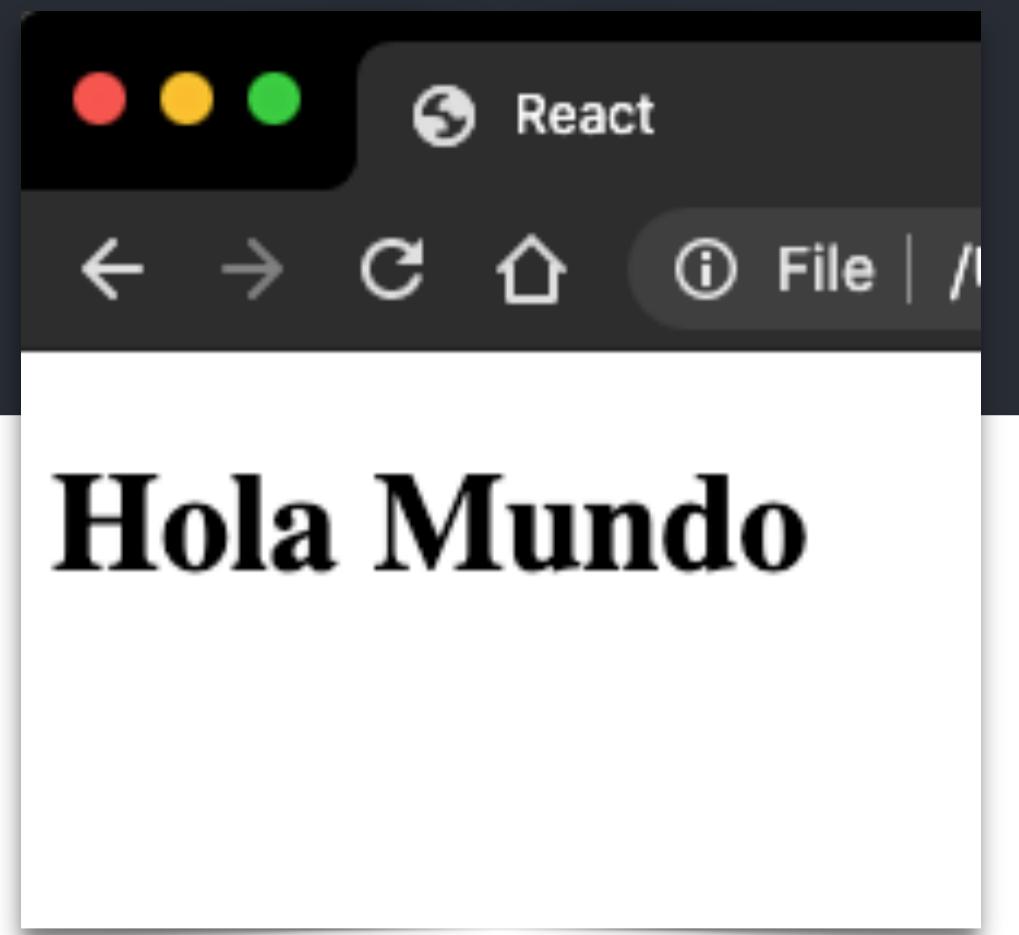
Referencia al elemento HTML

```
const divRoot = document.querySelector('#root');
const h1Tag = <h1>Hola Mundo</h1>;
ReactDOM.render( h1Tag , divRoot );
```



¿ Es una variable/constante que contiene la etiqueta?

Llama una función de React, para mostrar “algo” en el HTML



JSX = JS + XML

```
const h1Tag = <h1>Hola Mundo</h1>;
```

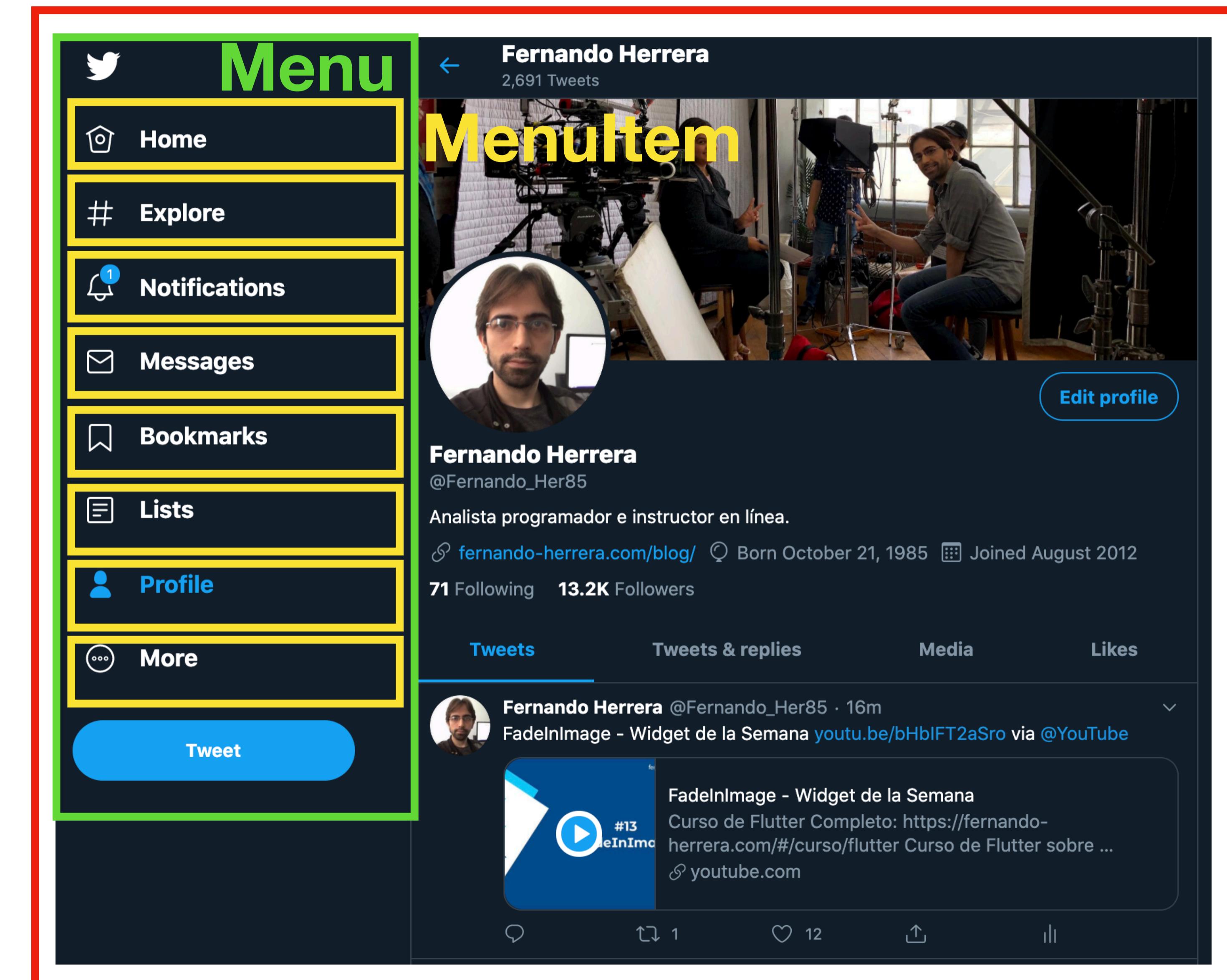
```
const h1Tag = document.createElement('h1', null, `Hola, soy ${ nombre }`);
```

.js .jsx

¿Qué es un componente React?

**Pequeña pieza de código encapsulada re-utilizable
que puede tener estado o no.**

Componentes



TwitterApp

Componentes

TwitterApp > Router > Screen/Página > Menú > MenuItem



Componentes

Formulario

Nombre: Fernando

Apellido: Herrera

Edad: 34

Submit

Introducción a las pruebas unitarias y de integración



¿Qué son?

¿Para qué sirven?

Características

Lineamientos

Mitos

No son una pérdida de tiempo



Unitarias y de Integración

**Unitarias: Enfocadas en pequeñas
funcionalidades.**

**Integración: Enfocadas en cómo reaccionan varias
piezas en conjunto**

Unitarias



Integración



Características de las pruebas

- 1. Fáciles de escribir**
- 2. Fáciles de leer**
- 3. Confiables**
- 4. Rápidas**
- 5. Principalmente unitarias**

A A A

- Arrange (Arregar)
- Act (Actuar)
- Assert (Afirmar)

A A A

Preparamos el estado inicial

- Inicializamos variables
- Importaciones necesarias

Arrange - Arreglar

Aplicamos acciones o estímulos

- Llamar métodos
- Simular clicks
- Realizar acciones sobre el paso anterior

Act - Actuar

Observar el comportamiento resultante

- Son los resultados esperados
- Ej: Que algo cambie, algo incremente o bien que nada suceda

Assert - Afirmar

Mitos

- **Hacen que mi aplicación no tenga errores**
- **Las pruebas no pueden fallar**
- **Hacen más lenta mi aplicación**
- **Es una pérdida de tiempo**
- **Hay que probar todo**

¿Qué es un Reducer?

1. Es una función común y corriente
2. Debe de ser una función pura
3. Debe de retornar un nuevo estado
4. Usualmente sólo recibe dos argumentos
 - El valor inicial (`initialState`) y la acción a ejecutar

Es una función común y corriente

```
const miFuncion = () => {  
}
```

```
const todoReducer = () => {  
}
```

Tareas por hacer, que llamaré “todos”

Debe de ser una función pura

1. No debe de tener efectos secundarios
2. No debe de realizar tareas asíncronas
3. Debe de retornar siempre un estado nuevo
4. No debe de llamar localStorage o sessionStorage
5. No debe de requerir más que una acción que puede tener un argumento.

Debe de ser una función pura

```
const initialTodos = [  
  id: 1,  
  todo: 'Comprar pan',  
  done: false  
];
```

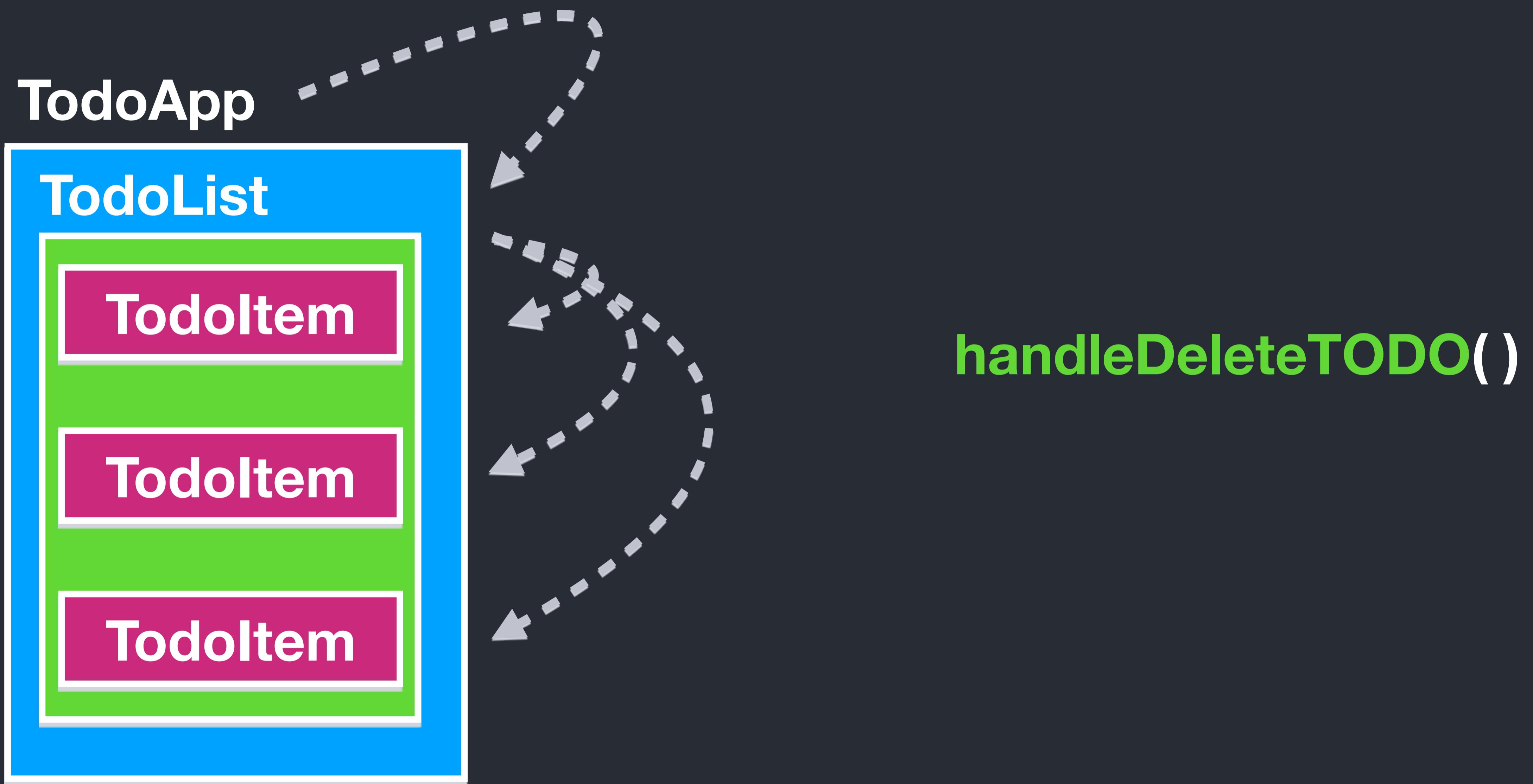
```
const todoReducer = ( state = initialTodos, action ) => {
```

```
  return state;  
}
```

```
let todos = todoReducer();  
  
console.log(todos);
```

```
▼ [{}...] i  
  ► 0: {id: 1, todo: "Comprar pan", done: false}  
    length: 1
```

¿Para qué hacer todo esto?



HomePage



Context

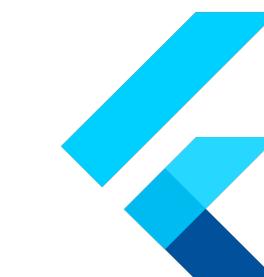
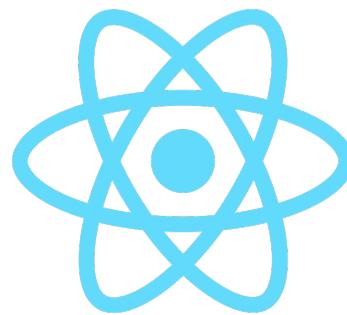


Login



¿Qué es Redux?

Es un contenedor predecible del estado de nuestra aplicación



Redux

Iniciar en Redux puede ser complicado

**Ahí es donde está la información
que mis componentes consumirán**



Store

Fuente única de la verdad

Reducer

Action

Agregar un nuevo TODO

Página o vista

agregar

borrar

actualizar

State

Reducer

```
[{  
  id: 1,  
  todo: 'Comprar pan',  
  done: false  
}];
```

