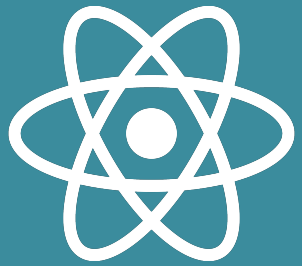
www.cesarguerra.mx



React Funcional

Estados en React

CONTENIDO

01

HOOKS

HOOKS

Los **Hooks (ganchos)** son una nueva API de la librería de React que nos permite **tener estado**, y otras características de React, en los componentes creados con una **function**.

Esto, antes, no era posible y nos obligaba a crear un componente con **class** para poder acceder a todas las posibilidades de React.

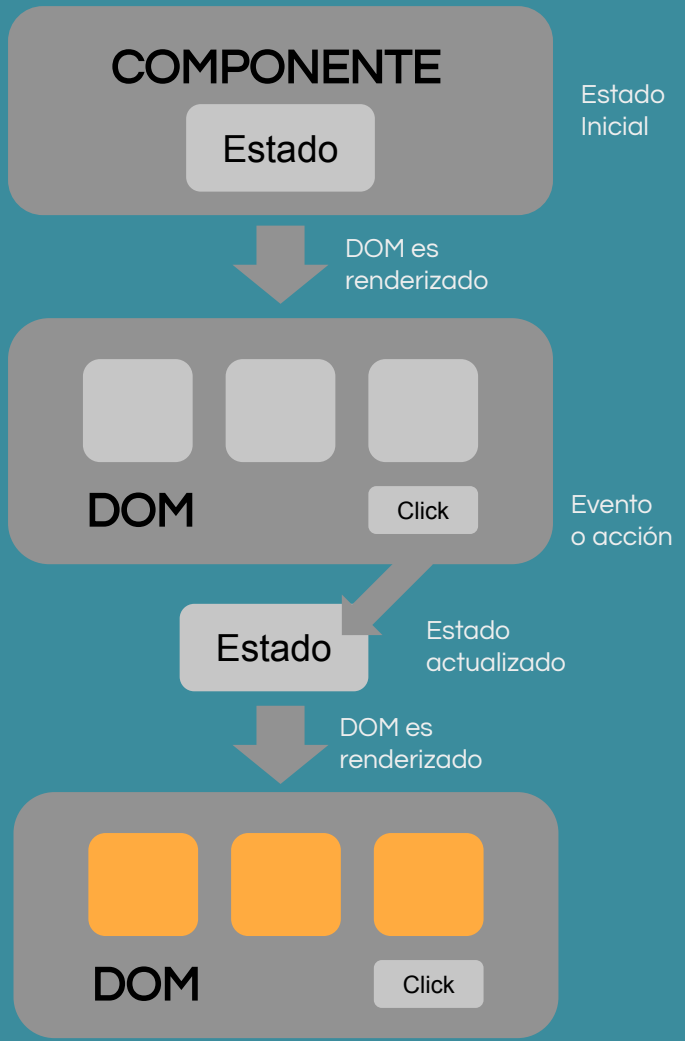
02

Estados en React

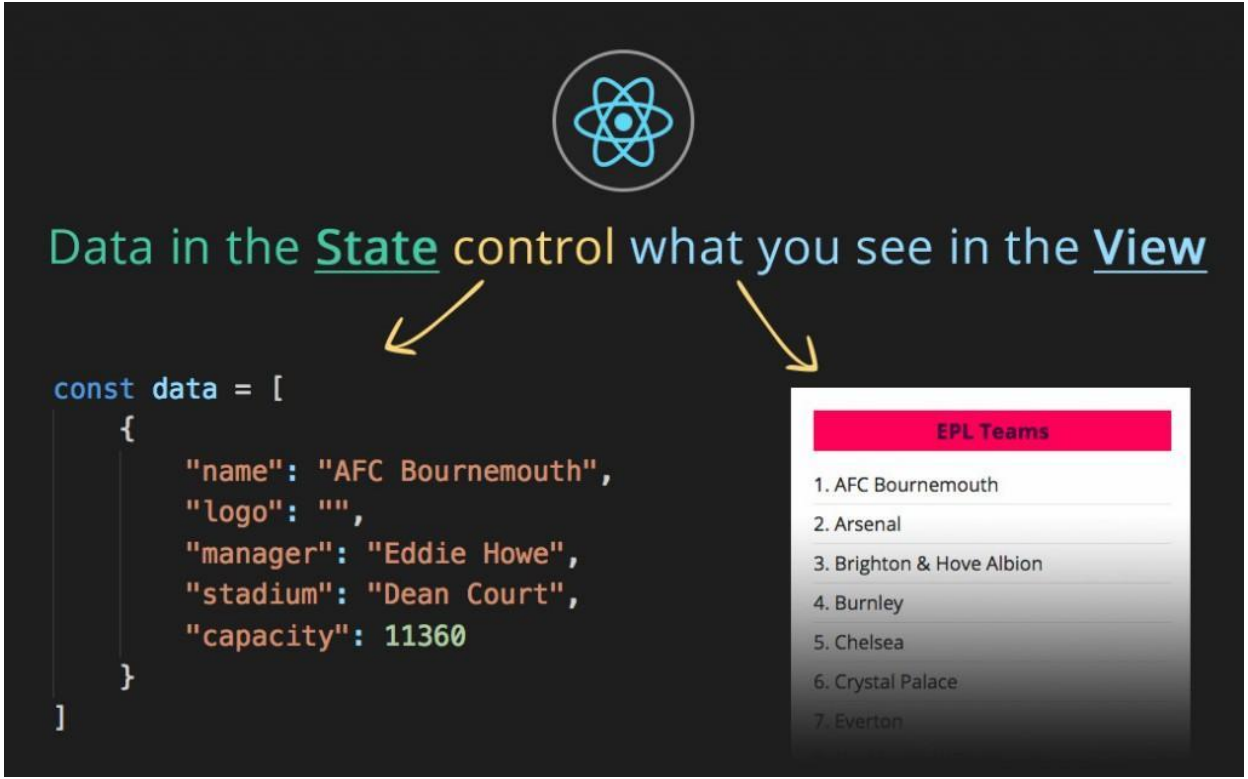
Estados en React

El estado (state) es un concepto que se refiere a cómo cambian los datos de nuestra aplicación a lo largo del tiempo.

Para cambiar nuestra aplicación de elementos HTML estáticos a una dinámica con la que el usuario pueda interactuar, necesitamos estados.



STATE


Data in the State control what you see in the View
const data = [
{
"name": "AFC Bournemouth",
"logo": "",
"manager": "Eddie Howe",
"stadium": "Dean Court",
"capacity": 11360
}
]
EPL Teams
1. AFC Bournemouth
2. Arsenal
3. Brighton & Hove Albion
4. Burnley
5. Chelsea
6. Crystal Palace
7. Everton

Estados en React

El manejo de estados en functional components se realiza por medio del **hook useState.**

Cada vez que se ejecuta el método set de un useState se reflejan los cambios en la interfaz de usuario.

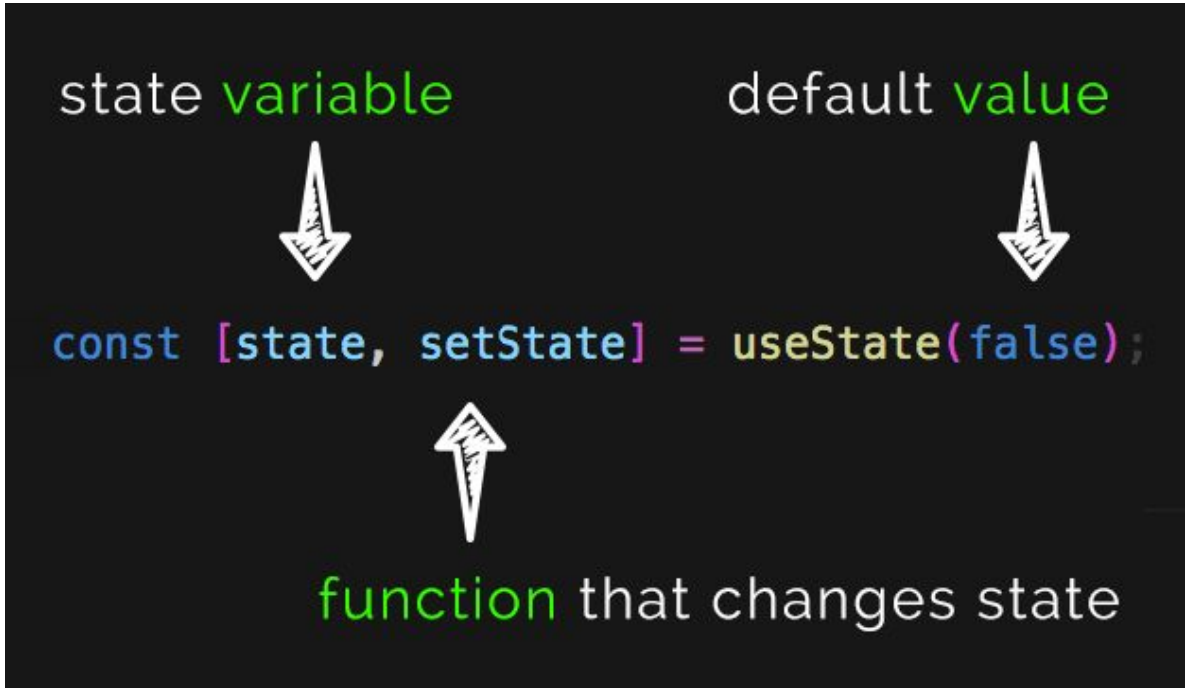
Para poder usarlo es importante importarlo:
`import { useState } from 'react';`

```
import React, { useState } from 'react';

function Example() {
  // Declaración de una variable de estado que llamaremos "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Uso de useState



https://ihatetomatoes.net/react-hooks-tutorial-for-beginners/

03

EVENTOS

¿QUÉ SON LOS EVENTOS?

Los eventos son formas de obtener datos sobre una determinada acción que un usuario ha realizado en nuestra app.

Los eventos más comunes son:
onClick: Cuando un usuario hace click.
onChange: Cuando un usuario escribe en una entrada (ejemplo: input).
onSubmit: Cuando se envía un formulario.

```
function ActionLink() {
  function handleClick(e) {
    e.preventDefault();
    console.log('The link was clicked.');
  }

  return (
    <a href="#" onClick={handleClick}>
      Click me
    </a>
  );
}
```

Pasar Eventos, No Llamarlos (1)

```
function handleClick() {
  alert('¡Me cliqueaste!');
}
```

Atención

Las funciones que se pasan a los manejadores de eventos deben ser pasadas, no llamadas. Por ejemplo:

pasar una función (correcto)	llamar una función (incorrecto)
`<button onClick={handleClick}>`	`<button onClick={handleClick()}>`

La diferencia es sutil. En el primer ejemplo, la función `handleClick` es pasada como un manejador de evento `onClick`. Esto le dice a React que lo recuerde y solo llama la función cuando el usuario hace clic en el botón.

En el segundo ejemplo, los `()` al final del `handleClick()` ejecutan la función *inmediatamente* mientras se renderiza, sin ningún clic. Esto es porque el JavaScript dentro de `{` y `}` en JSX se ejecuta de inmediato.

Referencia: https://es.react.dev/learn/responding-to-events

Pasar Eventos, No Llamarlos (2)

```
function handleClick() {
  alert('¡Me cliqueaste!');
```

Cuando escribes código en línea, la misma trampa se presenta de otra manera:

pasar una función (correcto)	llamar una función (incorrecto)
`<button onClick={() => alert('...')}>`	`<button onClick={alert('...')}>`

Pasar código en línea así no lo ejecutará al hacer clic; lo ejecutará cada vez que el componente se renderice:

```
// Esta alerta se ejecuta cuando el componente se renderiza, ¡no cuando se hace clic!
<button onClick={alert('¡Me cliqueaste!')}>
```

Si quieres definir un manejador de evento en línea, envuélvelo en una función anónima de esta forma:

```
<button onClick={() => alert('¡Me cliqueaste!')}>
```

Referencia: https://es.react.dev/learn/responding-to-events

04

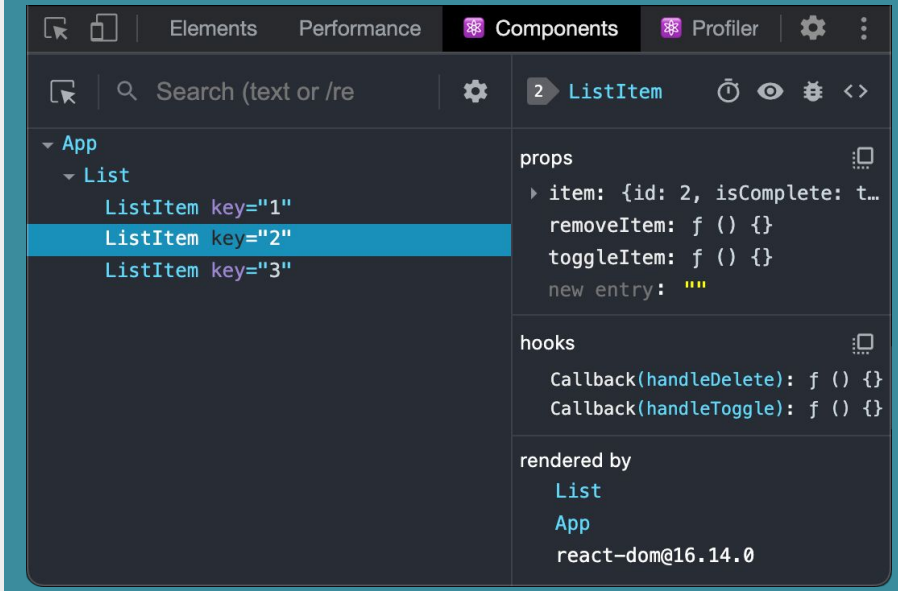
React Developer Tools

¿Qué es React Developer Tools?

Las Herramientas de Desarrollo de React (**React Developer Tools**) **es una extensión para navegadores web** que sirve para inspeccionar componentes de React, editar props y estados, e identificar problemas de rendimiento.

Cuando visites un sitio web construido con React, estas herramientas aparecerán como pestañas adicionales cuando inspeccionas el sitio web.

Referencia: https://es.react.dev/learn/react-developer-tools



Disponible para:

Google Chrome

Mozilla Firefox

Microsoft Edge

DEMOSTRACIÓN