



Maneja el estado de una SPA usando React

Jefferson A. Peña Torres

jefferson.amado.pena@correounivalle.edu.co

- Componentes con estado
- state vs props
- ¿Que es un Hook de estado?

Maneja el estado de una SPA usando React

Repasemos, una aplicación básica usando React

1. Componentes con estado

2. state vs props

3. ¿Que es un Hook de estado?

Una de las formas para realizar la configuración e instalación de los paquetes necesarios para iniciar el desarrollo de una aplicación SPA con React es utilizando `create-react-app`. Este paquete crea todo lo necesario e incluso la configuración para que ejecutando `npm start` en la terminal se pueda visualizar una App plantilla.

Con la extensión JSX, React puede crear componentes con etiquetas similares a las de HTML. Los componentes pueden ser considerados funciones simples que toman información (accesorios) y algunas de estas son funciones puras (paradigma funcional).

Las propiedades (**props**) son los valores que se pasan a los componentes, teniendo en cuenta su relación padre-hijo y los componentes envoltorios (**wrappers**) sirven para crear un estilo estándar para cualquier componente nuevo.



Maneja el estado de una SPA usando React

Componentes con estado

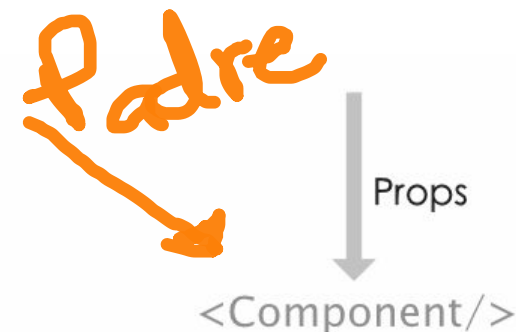
1

A diferencia de los componentes sin estado estos posee un método que les permite mantener ciertos datos. Cada componente tiene su propio estado. Ejemplo: `<Button />` es un componente y la información sobre el *focus*, *click* u otros eventos le conciernen a ese componente.

2

3

En React o cualquier *framework* basado en componentes utilizan el estado para lograr la interacción con el usuario.



Hijos



Maneja el estado de una SPA usando React

Componentes con estado

1



A diferencia de los componentes sin estado estos posee un método que les permite mantener ciertos datos. **Cada componente tiene su propio estado.** Ejemplo: `<Button />` es un componente y la información sobre el *focus*, *click* u otros eventos le conciernen a ese componente.

2

3

En React o cualquier *framework* basado en componentes utilizan el estado para lograr la interacción con el usuario. **El estado debe inicializarse en el constructor.**

```
import React, { Component } from 'react';  
class Button extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      value: 10  
    };  
    // This is to bind context  
    //when passing onClick as a callback  
    this.onClick = this.onClick.bind(this);  
  }  
  onClick() . . .  
  . . .  
};
```





Maneja el estado de una SPA usando React

Componentes con estado

1

A diferencia de los componentes sin estado estos posee un método que les permite mantener ciertos datos. Cada componente tiene su propio estado. Ejemplo: `<Button />` es un componente y la información sobre el *focus*, *click* u otros eventos le conciernen a ese componente.

2

3

En React o cualquier *framework* basado en componentes utilizan el estado para lograr la interacción con el usuario. El estado debe inicializarse en el constructor

```
import React, { Component } from 'react';  
class Button extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      value: 10  
    };  
    // This is to bind context  
    //when passing onClick as a callback  
    this.onClick = this.onClick.bind(this);  
  }  
  onClick() . . .  
  render() { . . . this.state.value . . . }  
};
```



El futuro digital
es de todos

MinTIC



Maneja el estado de una SPA usando React

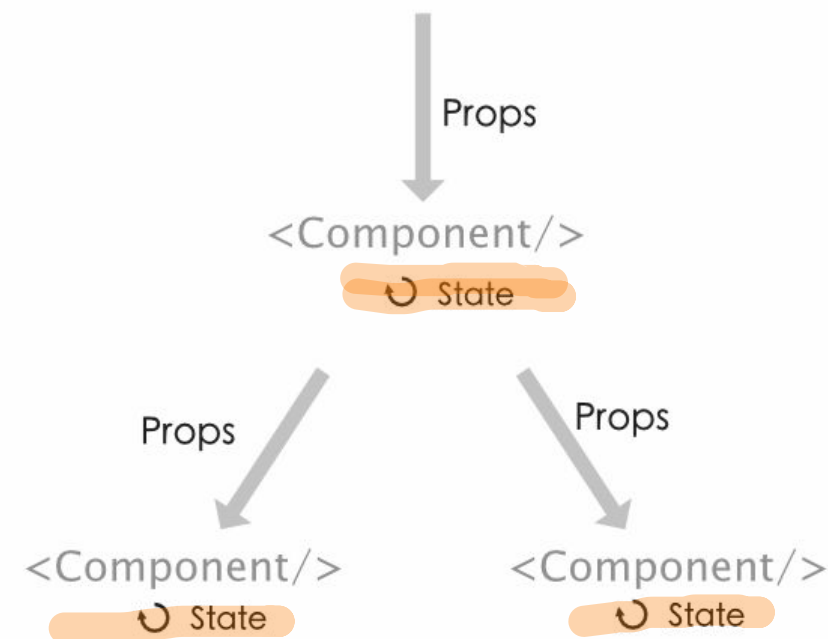
Estado vs. Propiedades

1

2

3

Las propiedades (*Props*) son inmutables, son de solo lectura y son pasadas de un componente padre a un componente hijo. Facilitan la transferencia de una "parte" del estado o funcionalidad del componente.



Maneja el estado de una SPA usando React

Estado vs. Propiedades

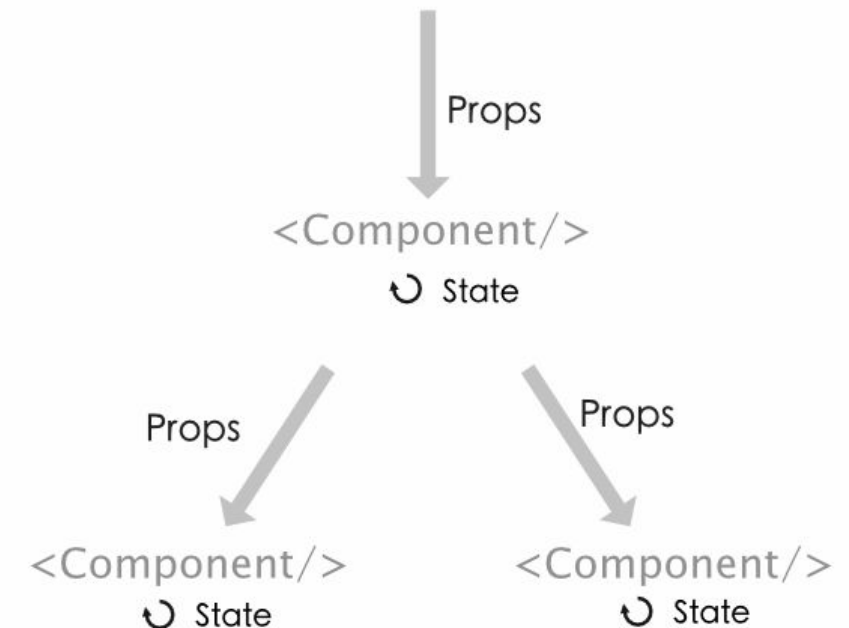
1

Las propiedades (*Props*) son inmutables, son de solo lectura y son pasadas de un componente padre a un componente hijo. Facilitan la transferencia de una "parte" del estado o funcionalidad del componente.

2

3

El estado es propio de cada componente y puede ser modificado o alterado de acuerdo al comportamiento que se programe.





El futuro digital
es de todos

MinTIC



Maneja el estado de una SPA usando React

Hook de estado

Un hook es una característica que tiene los componentes para acceder a su estado sin escribir una clase o un método constructor.



```
const [count, setCount] = useState(0)
```




El futuro digital
es de todos

MinTIC



Maneja el estado de una SPA usando React

Hook de estado

Un hook es una característica que tiene los componentes para acceder a su estado sin escribir una clase o un método constructor.

```
import { useState } from 'react'

const Counter = ( ) => {
  const [count, setCount] = useState(0)
  return (<div>
    <p>You clicked {count} times</p>
    <button onClick={() => setCount(count + 1)}>
    <div>)
}
ReactDOM.render(<Counter />, document.getElementById("main"))
```




Maneja el estado de una SPA usando React

Hook de efecto

Un hook es una característica que tiene los componentes para acceder a su estado sin escribir una clase o un método constructor.

El hook de efecto puede llevar a cabo tareas secundarias al cambiar el estado, como cambiar el título, cambiar el color, cambiar el texto de otro componente, etc

```
import { useState } from 'react'
```

```
const Counter = ( ) => {  
  const [count, setCount] = useState(0)  
  useEffect(() => {  
    document.title = `You clicked ${count} times`;   
  });  
  return (<div>  
    <p>You clicked {count} times</p>  
    <button onClick={() => setCount(count + 1)}>  
    </div>  
  )  
}  
ReactDOM.render(<Counter />, document.getElementById("main"))
```



Maneja el estado de una SPA usando React

Hook propios

Para algunas tareas que debe realizar el componente es necesario extraer la lógica del componente y convertirlas en operaciones reutilizables.

```
import { useState } from 'react'
```

```
const Switch = ( ) => {
```

```
  const [isOn, setIsOn] = useState(0)
```

```
  useEffect(() => {  
    (status) => setIsOn(status);
```

```
  });
```

```
  return (<div>
```

```
    isOn ? 'On': 'Off';
```

```
  <div>)
```

```
}
```

```
ReactDOM.render(<Switch />, document.getElementById("main"))
```

Maneja el estado de una SPA usando React

Resumen

1. Componentes con estado

En React o cualquier *framework* basado en componentes utilizan el estado para lograr la interacción con el usuario y cada componente tiene su estado.

2. state vs props

Mientras que las propiedades (*Props*) son inmutables y son pasadas de un componente padre a un componente hijo. El estado puede ser modificado o alterado de acuerdo al comportamiento que se programe.

3. ¿Que es un Hook de estado?

Un hook es una característica que tiene los componentes para acceder a su estado sin escribir una clase o un método constructor.



www.upb.edu.co/es/mision-tic
#MisiónTICSomosTodos