

# Integración con APIs

DigitalHouse >  
Coding School

“

Con React, podemos usar cualquier biblioteca AJAX. Algunas de las más populares son: **Axios**, **jQuery AJAX**, y **window.fetch**, la cual es soportada de manera nativa en la mayoría de los navegadores modernos.

”



# Índice

1. [componentDidMount\(\)](#)
2. [render\(\)](#)

# 1 | `componentDidMount()`

# componentDidMount()

Dentro de este método que identifica la fase del ciclo de vida del componente, **justo después de la primera vez que fue renderizado**, es donde debemos ir a buscar la información que queremos de la API.

Veamos a continuación cómo se vería esto en el código.

# componentDidMount()

```
class Gif extends Component {
```

```
  constructor(props){
```

```
    super(props)
```

```
    this.state = {
```

```
      gif: ""
```

```
    }
```

```
  }
```

```
  componentDidMount(){
```

```
    fetch('https://api.giphy.com/v1/gifs/random?api_key=FSHnai0lYd2NXPdn06Qdh64qxmGRVM69&tag=rat ing=g')
```

```
    .then(response => {response.json()})
```

```
    .then(data => {this.setState({gif: data.data.image_url})})
```

```
    .catch(e => {console.log(e)})
```

```
  }
```

```
  render(){
```

```
    //codigo...
```

```
  }
```

```
}
```

```
export default Gif;
```

Configuramos el estado del componente para que pueda almacenar la información de la API luego de hacer el fetch.

```
{}
```

# componentDidMount()

{}

```
class Gif extends Component {  
  constructor(props){  
    super(props)  
    this.state = {  
      gif: ""  
    }  
  }  
  
  componentDidMount(){  
    fetch('https://api.giphy.com/v1/gifs/random?api_key=FSHnai01Yd  
2NXPdn06Qdh64qxmGRVM69&tag=&rating=g')  
    .then(response => {response.json})  
    .then(data => {this.setState({gif: data.data.image_url})})  
    .catch(e => {console.log(e)})  
  }  
  
  render(){  
    //codigo...  
  }  
}  
  
export default Gif;
```

Ejecutamos el **fetch()** hacia el endpoint donde está la información. Una vez procesada la data con **.json**, la misma está lista para ser utilizada. Por eso, en el segundo **.then()**, asignamos esa información al estado del componente que habíamos declarado previamente.

# 2 | render()



# render()

Dentro del método **render()** es donde vamos a imprimir esa información que fuimos a buscar a la API. Sin embargo, en este método debemos contemplar un escenario donde la información quizás todavía no fue buscada, ya que el **fetch()** lo ejecutamos una vez que el componente fue renderizado.

Por esta razón, el código se vería algo así...

# render()

{}

```
class Gif extends Component {  
  constructor(props){  
    //codigo...  
  }  
  componentDidMount(){  
    //codigo...  
  }  
  
  render(){  
    let contenido  
    if(this.state.gif == ""){  
      contenido = <h3>Cargando...</h3>  
    } else {  
      contenido = <img src={this.state.gif}></img>  
    }  
    <div>  
      {contenido}  
      <button>Hace click acá para ver un Gif!</button>  
    </div>  
  }  
}  
  
export default Gif;
```

Dentro del método render, creamos la variable **contenido**, para poder decidir qué mostrar en base a si la información de la API está disponible o no.

# render()

{}

```
class Gif extends Component {
  constructor(props){
    //codigo...
  }
  componentDidMount(){
    //codigo...
  }

  render(){
    let contenido
    if(this.state.gif == ""){
      contenido = <h3>Cargando...</h3>
    } else {
      contenido = <img src={this.state.gif}></img>
    }
    <div>
      {contenido}
      <button>Hace click acá para ver un Gif!</button>
    </div>
  }
}

export default Gif;
```

Preguntamos si la propiedad **gif** del estado del componente, tiene información o sigue vacía. Si está vacía, mostramos una etiqueta que indique que la información se está cargando. De lo contrario, mostramos la información que almacena.

# render()

{}

```
class Gif extends Component {
  constructor(props){
    //codigo...
  }
  componentDidMount(){
    //codigo...
  }
  render(){
    let contenido
    if(this.state.gif == ""){
      contenido = <h3>Cargando...</h3>
    } else {
      contenido = <img src={this.state.gif}></img>
    }

    <div>
      {contenido}
      <button>Hace click acá para ver un Gif!</button>
    </div>
  }
}

export default Gif;
```

Por último, lo que vamos a terminar renderizando es lo que almacene nuestra variable **contenido**, ya que en ella estará la información correcta dependiendo de la disponibilidad de los datos de la API.

# Documentación



Para saber más podemos acceder a la documentación oficial de React haciendo clic en el siguiente [link](#).

DigitalHouse>  
Coding School