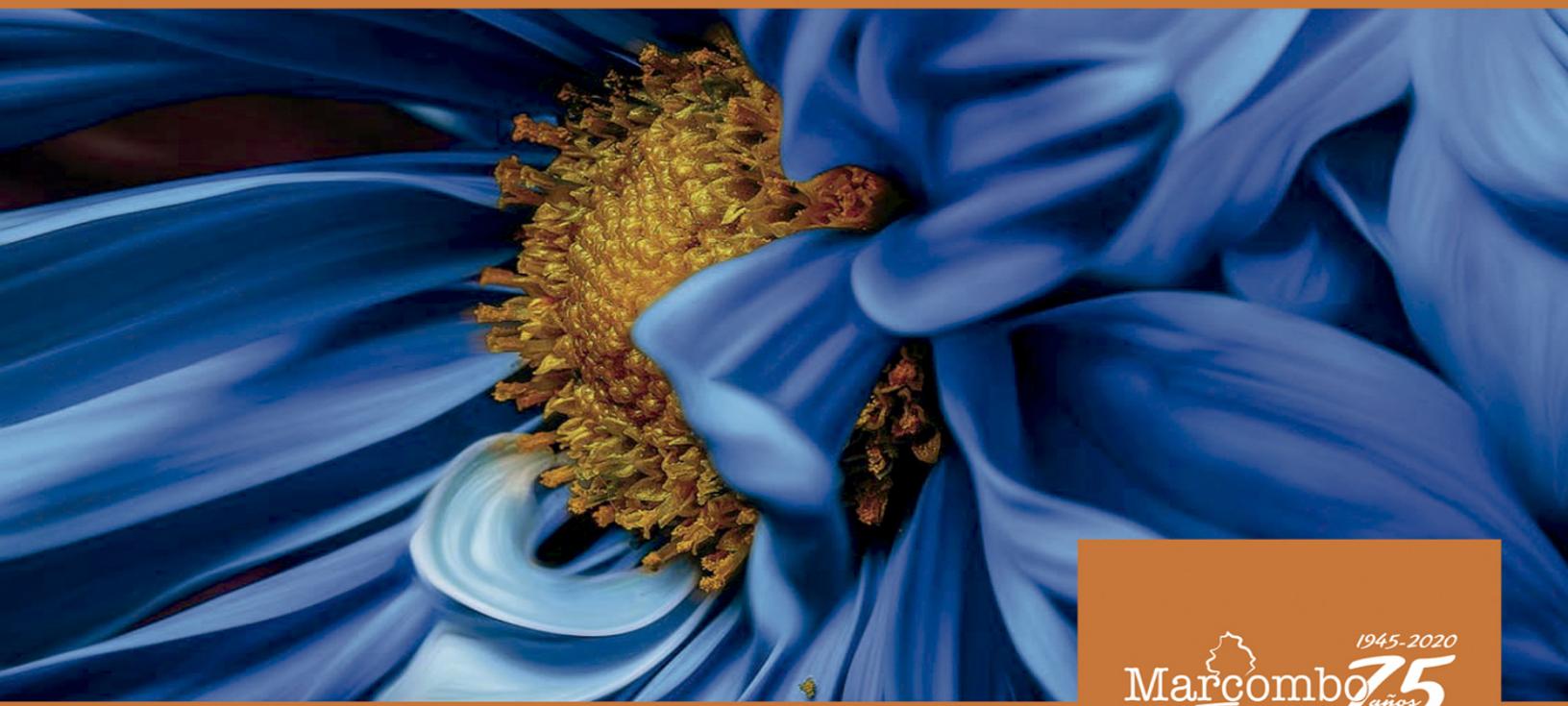


# Ejercicios prácticos con React

Carlos Santana Roldán

Cree aplicaciones web dinámicas con React desde Redux, Webpack, Node.js y GraphQL



1945-2020  
**Marcombo** 75 años

# Ejercicios prácticos con React

Cree aplicaciones web dinámicas con React  
desde Redux, Webpack, Node.js y GraphQL

Carlos Santana Roldán

Acceda a [www.marcombo.info](http://www.marcombo.info)  
para descargar gratis  
*contenido adicional*  
complemento imprescindible de este libro

Código: REACT1

# Ejercicios prácticos con React

Cree aplicaciones web dinámicas con React  
desde Redux, Webpack, Node.js y GraphQL

Carlos Santana Roldán



**Nota aclaratoria:** los códigos de ejemplo que se pueden encontrar en el libro se han desarrollado usando las versiones de software y librerías más actuales en el momento de la publicación del mismo, pero, como sabe, el software es algo que está en constante evolución y esto puede haber variado desde su fecha de publicación. Por este motivo existe la posibilidad que, de manera puntual, la sintaxis de algún ejemplo pueda variar.

Copyright © Packt Publishing Ltd. 2018. First published in the English language under the title *React Cookbook* (9781783980727).

Primera edición original publicada en inglés por Packt Publishing Ltd. con el título:  
*React Cookbook*, © 2018 Carlos Santana Roldán

Título de la edición en español: *Ejercicios prácticos con React*

Primera edición en español, año 2020

© 2020 MARCOMBO, S.L.  
[www.marcombo.com](http://www.marcombo.com)

Traducción: Francisco Martínez Carreno  
Correctora: Nuria Barroso  
Directora de producción: M.a Rosa Castillo

Los códigos de ejemplo que se pueden encontrar en el libro se han desarrollado usando las versiones de software y librerías más actuales en el momento de la publicación del mismo, pero, como sabe, el software es algo que está en constante evolución, y estas pueden haber variado desde su fecha de publicación. Por este motivo existe la posibilidad que, de manera puntual, la sintaxis de algún ejemplo pueda variar.

Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, [www.cedro.org](http://www.cedro.org)) si necesita fotocopiar o escanear algún fragmento de esta obra.

ISBN: 978-84-267-2867-8

Producción del ebook: [boqlab.com](http://boqlab.com)

A la memoria de mis abuelas, María Concepción Carrillo y Ana María Ochoa, a mi madre, Thelma C. Roldán, y a mi padre Francisco Santana, por sus sacrificios y por dar ejemplo de la importancia de la determinación y la dedicación.

*Carlos Santana Roldán*

## Sobre el autor

**Carlos Santana Roldán** es desarrollador web senior con más de 11 años de experiencia. Actualmente, ocupa la jefatura técnica de React Technical Lead en Disney ABC Television Group. Es el fundador de Codejobs.com, una de las comunidades de desarrolladores más populares de América Latina, y se ocupa de la capacitación de personal en tecnologías web tales como React, Node.js y JavaScript.

*Me gustaría expresar mi más profundo aprecio y especial gratitud a todas las personas del equipo editorial de Packt que me han ayudado en la realización de este libro, especialmente a Francis Savio Carneiro, Larissa Pinto y Diksha Wakode.*

*También me gustaría agradecer a Adrian Aguirre y Tony Guerrero la ayuda que me han prestado con el libro. Finalmente, quiero expresar mi más profundo agradecimiento a mi esposa, Cristina Rojas, por su apoyo; a mis padres, Francisco Santana y Thelma Roldán por su estímulo en mi desarrollo profesional y a mis amigos y familiares que me han ayudado a dar a conocer el libro.*

## Sobre el experto en revisión

**Mayur Tanna** es consultor senior de TI y es empleado de CIGNEX Datamatics. Ha trabajado en varios proyectos importantes con clientes

internacionales como el Banco Mundial y Naciones Unidas, y ha desempeñado un papel clave en la creación de la arquitectura de los citados proyectos utilizando las últimas tecnologías, entre las que se incluyen React, Angular, Node.js, MongoDB, Spring Boot, Firebase, Amazon Web Services y Google Cloud Platform. Mayur es coautor del libro *Serverless Web Applications with React and Firebase*. Tiene el título de maestría en aplicaciones informáticas y ha entrenado a muchos estudiantes de ingeniería por medio de talleres tecnológicos. En su tiempo libre, juega al tenis de mesa y al críquet.

*Quiero agradecer a mi esposa, la doctora Purna, a mis padres, Ratilal y Nirmala, a mi hijo pequeño, Dhyey, y al resto de mi familia, que me han apoyado y animado a pesar de todo el tiempo en el que no he podido disfrutar de su compañía. Sin su apoyo, la revisión de este libro no habría sido posible. También me gustaría agradecer al equipo de Packt que me haya dado la oportunidad de revisar este libro.*

# Contenido

## Prefacio

### Capítulo 1: Uso de React

Introducción

Uso de React con las últimas características de JS

Cómo hacerlo...

¿Qué novedades hay en React?

Cómo hacerlo...

Utilización de React en Windows

Cómo hacerlo...

### Capítulo 2: Cómo dominar los componentes y JSX

Introducción

Creación del primer componente de React

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

La organización de aplicaciones de React

Cómo hacerlo...

Cómo funciona...

Estilización de componentes con clases CSS y estilos en línea

Cómo hacerlo...

Cómo funciona...

Hay más...

Paso de props a componentes y su validación con PropTypes

Cómo hacerlo...

Cómo funciona...

Hay más...

Uso del estado local en componentes

Cómo hacerlo...

Cómo funciona...

Hay más...

Creación de componentes funcionales o apátridas

Cómo hacerlo...

Cómo funciona...

Hay más...

Asimilación de los métodos de ciclo de vida de React

Cómo hacerlo...

Lista de tareas, implementación de ComponentWillMount

El temporizador Pomodoro, implementación del constructor y de componentDidMount

Cambiador de criptomonedas, implementación de shouldComponentUpdate

Notas, implementación de componentWillReceiveProps y de componentWillUnmount

Gráficos C3.js, implementación de componentDidUpdate

Animación básica, implementación de componentWillUpdate

Cómo funciona...

Asimilación de los componentes puros de React

Preparación

Cómo hacerlo...

Cómo funciona...

Prevención de vulnerabilidades de XSS en React

Cómo hacerlo...

Cómo funciona...

Hay más...

### **Capítulo 3: Tratamiento de eventos, vinculación y paquetes de React útiles**

Introducción

Métodos vinculantes usando constructores en lugar de funciones flecha

Cómo hacerlo...

Cómo funciona...

Creación de elementos de formularios con eventos

Cómo hacerlo...

Cómo funciona...

Hay más...

Visualización de información en una ventana modal con react-popup

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Implementación con la guía de estilo de Airbnb React/JSX

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Actualización de títulos y etiquetas meta con React Helmet

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

## **Capítulo 4: Adición de rutas a aplicaciones con React Router**

Introducción

Implementación de React Router v4

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Adición de parámetros a las rutas

Cómo hacerlo...

Cómo funciona...

## **Capítulo 5: Dominio de Redux**

Introducción

Creación de Store en Redux

Preparación  
Cómo hacerlo...  
Cómo funciona...

Elaboración de creadores de acciones y envío de acciones

Preparación  
Cómo hacerlo...  
Cómo funciona...

Implementación de Firebase con Redux

Preparación  
Cómo hacerlo...  
Cómo funciona...

## **Capítulo 6: Creación de formularios con Redux Form**

Introducción

Creación de formularios controlados mediante el estado local

Preparación  
Cómo hacerlo...  
Cómo funciona...

Creación de formularios utilizando Redux Form

Preparación  
Cómo hacerlo...  
Cómo funciona...

Implementación de la validación en formularios

Cómo hacerlo...  
Cómo funciona...

## **Capítulo 7: Animaciones con React**

Introducción

Animación de una lista de tareas con ReactCSSTransitionGroup

Preparación  
Cómo hacerlo...  
Cómo funciona...

Uso de la biblioteca react-animations

Preparación

Cómo hacerlo...

Hay más...

Creación de la primera animación con React Pose

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

## **Capítulo 8: Creación de API con Node.js haciendo uso de MongoDB y MySQL**

Introducción

Creación de API básicas con Express

Preparación

Cómo hacerlo...

Cómo funciona...

Elaboración de bases de datos con MongoDB

Preparación

Instalación de MongoDB Community Edition manualmente (opción difícil)

Instalación de MongoDB Community Edition con Homebrew (opción fácil)

Ejecución de MongoDB

Cómo hacerlo...

Cómo funciona...

Puntos finales del método GET

Puntos finales del método POST

Puntos finales del método DELETE

Puntos finales del método PUT

Elaboración de una base de datos con MySQL

Preparación

Cómo hacerlo...

Cómo funciona...

Puntos finales del método POST

Puntos finales del método GET

Puntos finales del método DELETE

Puntos finales del método PUT

Adición de tokens de acceso para proteger las API

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

## **Capítulo 9: Apollo y GraphQL**

Introducción

Creación del primer servidor GraphQL

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Creación de líneas de tiempo de Twitter con Apollo y GraphQL

Preparación

Creación de un servidor back-end GraphQL

Cómo hacerlo...

Cómo funciona...

## **Capítulo 10: Dominio de Webpack 4.x**

Introducción

Configuración de Webpack 4 Zero

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Adición de React a Webpack

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Adición de Webpack Dev Server y Sass, Stylus o LessCSS a

*React*

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Optimización de Webpack 4, división de paquetes

Preparación  
Cómo hacerlo...  
Cómo funciona...  
Implementación de Node.js con React/Redux y Webpack 4  
Preparación  
Cómo hacerlo...  
Cómo funciona...  
Hay más...

## **Capítulo 11:Implementación de renderizado del lado del servidor**

Introducción  
Implementación del renderizado del lado del servidor  
Preparación  
Cómo hacerlo...  
Cómo funciona...  
Hay más...

Implementación de promises con renderizado del lado del servidor  
Preparación  
Cómo hacerlo...  
Cómo funciona...

Implementación de Next.js  
Preparación  
Cómo hacerlo...  
Cómo funciona...

## **Capítulo 12:Pruebas y depuración**

Introducción  
Prueba del primer componente con Jest y Enzyme  
Preparación  
Cómo hacerlo...  
Como funciona...  
Hay más...

Prueba de un contenedor Redux, acciones y reductores

Preparación

Cómo hacerlo...

Depuración de aplicaciones de React utilizando Dev Tools de React y Redux

Preparación

Cómo hacerlo...

Simulación de eventos

Cómo hacerlo...

Cómo funciona...

## **Capítulo 13:Despliegue en la fase de producción**

Introducción

Implementación en producción con Digital Ocean

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Configuración de Nginx, PM2 y un dominio en Droplet

Preparación

Cómo hacerlo...

Cómo funciona...

Hay más...

Implementación de Jenkins (integración continua)

Preparación

Cómo hacerlo...

Cómo funciona...

## **Capítulo 14:Uso de React Native**

Introducción

Creación de la primera aplicación de React Native

Preparación

Cómo hacerlo...

Cómo funciona...

Creación de una lista de tareas con React Native

Cómo hacerlo...

Cómo funciona...

Hay más...

Implementación de React Navigation V2

Preparación

Como hacerlo...

Cómo funciona...

## **Preguntas más frecuentes sobre React**

# Prefacio

Hoy en día existen toneladas de bibliotecas JavaScript, frameworks y herramientas para el desarrollo web. Sin embargo, debe evaluar cada tecnología para saber si se ajusta a los requisitos de nuestro proyecto. Este es el motivo de presentar React, una de las bibliotecas más capaces para la creación de interfaces de usuario dinámicas. Actualmente es la biblioteca más extendida (no es un framework) hecha por Facebook. He trabajado con otros frameworks JS como AngularJS (también con las nuevas versiones Angular 2, 4, 5), Backbone.js, Ember, y Vue.js en diferentes proyectos y puedo afirmar que React es la biblioteca con la que más disfruto a la hora de desarrollar nuevas aplicaciones web.

React ha cambiado la forma de desarrollar aplicaciones web y, en combinación con Redux, se consigue una potente arquitectura front-end que tiene sentido no solo para los desarrolladores experimentados, sino también para aquellos que están iniciando su viaje por el front-end.

El libro presenta todas las herramientas y las mejores prácticas de React en recetas sencillas y fáciles de seguir. Todas las recetas del libro son 100 % prácticas y cada una tiene el código necesario para entender lo más importante.

Bienvenido a un futuro mejor; diviértase y aprenda con la lectura de este libro.

## A quién va dirigido este libro

Este libro lo pueden utilizar desarrolladores que tengan conocimientos básicos sobre la creación de aplicaciones web. Pero está especialmente

indicado para desarrolladores de JavaScript, aunque no pone límite a cualesquiera otros tipos de desarrollos.

## De lo que trata este libro

Capítulo 1, *Uso de React*. React es una biblioteca JavaScript (con licencia del MIT) hecha por Facebook para desarrollar interfaces de usuario interactivas. Se utiliza para crear componentes dinámicos y reutilizables. Lo más importante de React es que se puede utilizar como cliente, como servidor, en aplicaciones móviles e incluso en aplicaciones RV.

Capítulo 2, *Cómo dominar los componentes y JSX*. Este capítulo contiene recetas relacionadas con la creación de componentes en React. Va a aprender a crear componentes de React (componentes de clase, componentes puros y componentes funcionales) y a organizar la estructura de su proyecto.

Capítulo 3, *Gestión de eventos, vinculación y paquetes de React útiles*. Este capítulo contiene recetas relacionadas con la gestión de eventos, métodos vinculantes en React, y también va a implementar algunos de los paquetes más útiles de React.

Capítulo 4, *Adición de rutas a aplicaciones con React Router*. En este capítulo, va a aprender a añadir rutas dinámicas a los proyectos usando React Router v4.

Capítulo 5, *Dominio de Redux*. Redux es un contenedor de estados predecibles para aplicaciones JavaScript. Esto significa que Redux se puede utilizar con Vanilla JavaScript o con frameworks/bibliotecas como Angular y jQuery. Redux es principalmente una biblioteca encargada de emitir actualizaciones de estados y respuestas a acciones.

Capítulo 6, *Creación de formularios con Redux Form*. Los formularios son una parte fundamental de cualquier aplicación web, y en las siguientes

recetas va a aprender a usar formularios con y sin Redux Form.

**Capítulo 7, Animaciones con React.** Las animaciones son muy frecuentes en cualquier aplicación web. Desde la aparición de CSS3, las animaciones se han generalizado y se implementan con facilidad. El uso más habitual de las animaciones es el de las transiciones, en las que se pueden cambiar las propiedades CSS y definir la duración o el retardo.

**Capítulo 8, Creación de API con Node.js utilizando MongoDB y MySQL.** Node.js se utiliza ampliamente como back-end para aplicaciones web porque facilita la creación de API, y su rendimiento es superior al de tecnologías como Java, PHP y Ruby. Habitualmente, la forma más extendida de usar Node.js es utilizar un framework llamado Express.

**Capítulo 9, Apollo y GraphQL.** GraphQL es un lenguaje de consulta de la capa de aplicación que se puede utilizar con cualquier base de datos, también es de código abierto (licencia del MIT) y desarrollado por Facebook. La principal diferencia con REST es que GraphQL no utiliza puntos finales sino consultas, y es compatible con la mayoría de los lenguajes de servidor como JavaScript (Node.js), Go, Ruby, PHP, Java, Python, etc.

**Capítulo 10, Dominio de Webpack 4.x.** Webpack 4 no necesita un archivo de configuración por defecto. Antes, en las versiones más antiguas, había que tener un archivo de configuración, pero por supuesto, si necesita personalizar Webpack 4 por necesidades del proyecto, puede crear un archivo de configuración y será mucho más fácil de programar.

**Capítulo 11, Implementación para la renderización del lado del servidor.** Probablemente, si no le interesa demasiado el SEO, no tiene que preocuparse por la renderización del lado del servidor (SSR). Actualmente, Googlebot soporta Client Side Rendering (CSR), y puede indexar su sitio en Google, pero si le interesa el SEO y le preocupa mejorarlo en otros

buscadores como Yahoo, Bing o DuckDuckGo, entonces la mejor opción es usar la renderización del lado del servidor (SSR).

Capítulo 12, *Pruebas y depuración*. Las pruebas y la depuración son muy importantes para cualquier proyecto que quiera ofrecer una calidad óptima. Desgraciadamente, muchos desarrolladores no se preocupan por las pruebas (pruebas unitarias) porque piensan que reducirán la velocidad del desarrollo y algunos de ellos lo dejan para el final del proyecto. En lo que respecta a mi experiencia personal, puedo decir que las pruebas desde el principio del proyecto ahorrarán tiempo porque al final tendrá que corregir un menor número de errores.

Capítulo 13, *Despliegue para la fase de producción*. Ahora es el momento de hacer el despliegue de su aplicación para producción y mostrarla al mundo. En este capítulo, aprenderá cómo implementar la aplicación de React utilizando uno de los mejores servicios de la nube: Digital Ocean.

Capítulo 14, *Uso de React Native*. React Native es un framework para crear aplicaciones para móviles usando JavaScript y React. Muchas personas piensan que con React Native se desarrolla una "aplicación web para móviles" o una "aplicación híbrida" (como Ionic, PhoneGap o Sencha) pero lo que se crea es una aplicación nativa porque React Native convierte el código de React a Java para aplicaciones Android y a Objective-C para aplicaciones iOS.

## Cómo sacar el máximo partido a este libro

Para dominar React, es necesario tener un conocimiento fundamental de JavaScript y Node.js. Este libro se dirige principalmente a desarrolladores web, y en el momento de escribirlo, se consideraron las siguientes suposiciones en relación con el lector:

- El lector sabe cómo instalar la última versión de Node.js.

- Es un desarrollador de tipo medio que puede entender la sintaxis de JavaScript ES6.
- Tiene poca experiencia con las herramientas CLI y la sintaxis de Node.js.

El libro también incluye contenido para desarrolladores de móviles (iOS y Android) y usa React Native. Si usted es un principiante debería aprender a instalar Android SDK o el simulador de iOS con Xcode.

## Descarga de los archivos con ejemplos de código

Para la descarga de los archivos siga los siguientes pasos:

1. Entre en [www.marcombo.info](http://www.marcombo.info).
2. Introduzca el código **REACT1**.
3. Rellene las casillas con su nombre, email y acepte la política de privacidad.
4. Busque el código **REACT1** en el listado.

Una vez descargado el archivo, asegúrese de descomprimir o extraer la carpeta utilizando la última versión de:

- WinRAR/7-Zip para Windows
- Ziipeg/iZip/UnRarX para Mac
- 7-Zip/PeaZip para Linux

Los códigos que se utilizan en el libro están también alojados en GitHub, en <https://github.com/PacktPublishing/React-Cookbook>. Si se produce una actualización de los códigos, estos se actualizarán en el repositorio existente en GitHub.



Las advertencias o notas importantes se indican así.



Los consejos y trucos se indican así.

## Contacte con nosotros

Los comentarios de nuestros lectores son siempre bien recibidos.

**Comentarios generales.** Si quiere preguntar sobre algún aspecto del libro, por favor, entre en [www.marcombo.com/contacto](http://www.marcombo.com/contacto).

**Erratas.** A pesar de que se han tomado todas las precauciones para asegurar la fiabilidad del contenido, puede haber errores. Si ha encontrado algún error en el libro, le agradeceríamos que nos lo comunique. Por favor, envíe un correo a [editorial@marcombo.com](mailto:editorial@marcombo.com).

**Piratería.** Si se encuentra con copias ilegales de nuestros trabajos en cualquiera de sus formas en Internet, le agradeceríamos que nos proporcione la dirección de la ubicación o el nombre del sitio web.

Por favor, contacte con nosotros en [pedidos@marcombo.com](mailto:pedidos@marcombo.com) indicando el enlace al material pirateado.

**Si está interesado en convertirse en autor.** Si hay un tema en el que es experto y está interesado en escribir o colaborar en la elaboración de un libro, por favor visite [www.marcombo.com](http://www.marcombo.com).

## Reseñas

Por favor, escriba una reseña. Una vez que haya leído y utilizado este libro, ¿por qué no dejar una reseña en el sitio en el que lo compró? Los lectores potenciales podrán leer su opinión imparcial y tenerla en cuenta a la hora de decidir sobre la compra del libro. A nosotros nos permite saber lo que piensa de nuestros productos, y nuestros autores pueden ver sus comentarios sobre su libro. ¡Gracias!

# Uso de React

En este capítulo se tratarán las siguientes recetas:

- Introducción
- Uso de React con las últimas características de JS
- ¿Qué hay nuevo en React?
- Uso de React en Windows

## Introducción

React es una biblioteca JavaScript (con licencia del MIT), desarrollada por Facebook, para crear interfaces de usuario interactivas. Se utiliza para crear componentes dinámicos y reutilizables. Lo mejor de React es que se puede usar en aplicaciones del lado de cliente, del lado de servidor, en aplicaciones para móviles e incluso en aplicaciones de RV.

En las páginas web modernas, es necesario manipular el DOM constantemente. El problema es que hacerlo a menudo puede afectar seriamente al rendimiento de la aplicación. React utiliza un DOM virtual, lo que significa que todas las actualizaciones ocurren en la memoria (esta operación es más rápida que manipular el DOM real directamente). La curva de aprendizaje de React es corta en comparación con otros frameworks de JavaScript como Angular, Vue o Backbone, principalmente porque el código de React está escrito en su mayoría en JavaScript moderno (clases, funciones flecha, plantillas de cadenas de texto, etc.) y no tiene demasiados patrones para escribir código, como son la inyección de dependencias, o un sistema de plantillas, como en Angular.

Empresas como Airbnb, Microsoft, Netflix, Disney, Dropbox, Twitter, PayPal, Salesforce, Tesla y Uber utilizan React de forma generalizada en sus proyectos. En este libro, aprenderá a desarrollar sus aplicaciones de React de la misma forma que la utilizan estas empresas, haciendo uso de las mejores prácticas.

## Uso de React empleando las últimas características de JS

Como dije en la introducción, React está escrito principalmente en JavaScript moderno (ES6, ES7 y ES8). Si desea aprovechar las ventajas de React, hay nuevas características de JS que debe dominar para obtener los mejores resultados de las aplicaciones de React. En esta primera receta, va a tratar las características esenciales de JS para que esté preparado y pueda empezar a trabajar en su primera aplicación de React.

### Cómo hacerlo...

En esta sección, verá cómo utilizar en React las características más importantes de JS:

1. `let` y `const`. La nueva forma de declarar variables es mediante `let` o `const`. Puede utilizar `let` para declarar variables que pueden cambiar sus valores pero solo en el ámbito de un bloque del programa. La diferencia entre `let` y `var` es que `let` es una variable en el ámbito de bloque, por lo que no puede ser global, y con `var` puede declarar una variable global. Por ejemplo:

```
var name = 'Carlos Santana';
let age = 30;

console.log(window.name); // Carlos Santana
console.log(window.age); // no definido
```

2. Para comprender de forma clara el “ámbito de bloque” va a declarar un bucle `for` con `var` y `let`. En primer lugar, va a utilizar `var` y verá su comportamiento:

```
for (var i = 1 ; i <= 10;  
     i++) { console.log(i); //  
             1, 2, 3, 4... 10  
}  
  
console.log(i); // Imprimirá el último valor de i: 10
```

3. Si escribe el mismo código, pero con `let`, ocurrirá lo siguiente:

```
for (let i = 1 ; i <= 10; i++)  
{ console.log(i); // 1, 2,  
            3, 4... 10  
}  
  
console.log(i); // ReferenceError no capturado: i no está  
definido
```

4. Con `const`, puede declarar constantes, lo que significa que su valor es inalterable (excepto para matrices y objetos):

```
const pi = 3.1416;  
pi = 5; // TypeError no capturado: Asignación de una variable  
        constante
```

5. Si declara una matriz con `const`, puede manipular los elementos de la misma (añadirlos, eliminarlos, o modificarlos):

```
const cryptoCurrencies = ['BTC', 'ETH', 'XRP'];  
  
// Adición de ERT: ['BTC', 'ETH', 'XRP', 'ERT'];  
cryptoCurrencies.push('ERT');  
  
// Eliminará el primer elemento: ['ETH', 'XRP', 'ERT'];  
cryptoCurrencies.shift();  
  
// Modificación de un elemento.  
cryptoCurrencies[1] = 'LTC'; // ['ETH', 'LTC', 'ERT'];
```

6. También, al utilizar objetos, puede añadir, eliminar o modificar nodos:

```
const person = {
    name: 'Carlos
    Santana', age:
    30,
    email: 'carlos@milkzoft.com'
};
// Adición de un nuevo nodo...
person.website = 'https://www.codejobs.com';

// Eliminación de un nodo...
delete person.email;

// Actualización de un nodo...
person.age = 29;
```

7. **Operador de propagación.** El operador de propagación (...) separa un objeto iterable en valores individuales. En React, se puede utilizar para añadir valores a una matriz, por ejemplo cuando quiere añadir un nuevo elemento a una lista de tareas utilizando `setState` (esto se explicará en el siguiente capítulo):

```
this.setState({
  items: [
    ...this.state.items, // Propaga los elementos en uso
    {
      task: 'My new task', // Una nueva tarea en la lista
    }
  ]
});
```

8. El operador de propagación se puede utilizar también en React para propagar atributos (props) en JSX:

```

render() {
  const props = {};
  props.name = 'Carlos Santana';
  props.age = 30;
  props.email = 'carlos@milkzoft.com';
  return <Person {...props} />;
}

```

9. **El parámetro rest.** El parámetro rest se representa también por .... Al último parámetro de una función prefijado con ... se le llama parámetro rest. El parámetro rest es una matriz que contendrá el resto de los parámetros de una función cuando el número de argumentos es superior al número de parámetros especificado:

```

function setNumbers(param1, param2, ...args) {
  // param1 = 1
  // param2 = 2
  // args = [3, 4, 5, 6];
  console.log(param1, param2, ...args); // Log: 1, 2, 3, 4, 5, 6
}
setNumbers(1, 2, 3, 4, 5, 6);

```

10. **Desestructuración.** La función de asignación desestructurada es la que más se utiliza en React. Es una expresión que nos permite asignar los valores o propiedades de un objeto iterable a variables. Generalmente, con esta función puede convertir sus componentes props en variables (o constantes):

```

// Imagine que está en el componente <Person> y que recibe
// los props (en this.props): name, age y email.
render() {
  // Los props son:
  // { name: 'Carlos Santana', age: 30, email:
  'carlos@milkzoft.com' }
  console.log(this.props);
  const { name, age, email } = this.props;
  // Ahora puede usar los nodos como constantes...
  console.log(name, age, email);

  return (
    <ul>
      <li>Name: {name}</li>
      <li>Age: {age}</li>
      <li>Email: {email}</li>
    </ul>
  );
}

// También se puede utilizar la desestructuración en
// parámetros de una función

const Person = ({ name, age, email }) => (
  <ul>
    <li>Name: {name}</li>
    <li>Age: {age}</li>
    <li>Email: {email}</li>
  </ul>
);

```

11. **Funciones flecha.** ES6 proporciona una nueva forma de crear funciones utilizando el operador `=>`. A estas funciones se les llama funciones flecha. Este nuevo método tiene una sintaxis más corta, y las funciones flecha son anónimas. En React, las funciones flecha se utilizan para enlazar el objeto `this` a los métodos en lugar de enlazarlo al constructor:

```

class Person extends Component {
  showProps = () => {
    console.log(this.props); // { name, age, email... }
  }

  render() {
    return (
      <div>
        Consoling props: {this.showProps()}
      </div>
    );
  }
}

```

12. **Plantillas literales.** Las plantillas literales son una nueva forma de crear una cadena de caracteres utilizando comillas invertidas (` `) en lugar de comillas simples (' ') o dobles (" "). React utiliza las plantillas literales para concatenar nombres de clases o para renderizar (representar) una cadena usando un operador ternario:

```

render()
{
  const { theme } = this.props;

  return (
    <div
      className=`base ${theme === 'dark' ? 'darkMode' :
      'lightMode'}`}
    >
      Some content here...
    </div>
  );
}

```

13. **Map.** El método `map()` devuelve una nueva matriz con los resultados de la llamada a una función que actúa sobre cada elemento de la matriz que hace la llamada. Map se utiliza frecuentemente en React, y se emplea fundamentalmente para