

## REACT DE JAVASCRIPT

React se utiliza ampliamente en el desarrollo web para construir interfaces de usuario interactivas, aplicaciones de una sola página (SPA) y sitios web dinámicos. Además de Facebook, su creador, muchas otras empresas y aplicaciones reconocidas utilizan React, como Netflix, Airbnb, Instagram, y plataformas de comercio electrónico. También es popular para el desarrollo de aplicaciones móviles con React Native, y se emplea en redes sociales, plataformas de entretenimiento, y aplicaciones de mensajería.

**ReactJS** es una de las librerías más populares de **JavaScript** para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.

Es importante señalar que ReactJS no es un framework de JavaScript. Esto porque sólo es responsable de renderizar los componentes de la capa de vista de una aplicación. React es una alternativa a frameworks como **Angular** y **Vue**, que permiten crear funciones complejas.

React te permite construir interfaces de usuario a partir de piezas individuales llamadas componentes. Crea tus propios componentes de React como Thumbnail, LikeButton, y Video. Luego combínalos para formar pantallas, páginas y aplicaciones.

Los componentes de React son funciones de JavaScript. ¿Quieres mostrar algún contenido condicionalmente? Utiliza una sentencia if. ¿Mostrar una lista? Prueba la función map() de un array. Aprender React es aprender a programar.

JSX. Es una extensión de la sintaxis de JavaScript popularizada por React. Al poner marcado JSX cerca de la lógica de renderizado relacionada hace que los componentes de React sean fáciles de crear, mantener y eliminar.

Los componentes de React reciben datos y devuelven lo que debe aparecer en la pantalla. Puedes pasarles nuevos datos en respuesta a una interacción, como cuando el usuario escribe en un

campo de texto. React actualizará entonces la pantalla para que se corresponda con los nuevos datos.

No tienes que construir toda tu página con React. Añade React a tu página HTML existente y renderiza componentes interactivos de React en cualquier lugar de ella.

React también es una arquitectura. Los frameworks que la implementan te permiten obtener datos en componentes asíncronos que se ejecutan en el servidor o incluso durante la compilación. Lee datos de un archivo o una base de datos y pásalos hacia abajo a tus componentes interactivos.

## ¿Qué es React?

React es una biblioteca de JavaScript front-end.

React fue desarrollado por el ingeniero de software de Facebook Jordan Walke.

React también se conoce como React.js o ReactJS.

React es una herramienta para crear componentes de UI.

# ¿Cómo funciona React?

React crea un DOM VIRTUAL en la memoria.

En lugar de manipular directamente el DOM del navegador, React crea un DOM virtual en la memoria, donde realiza toda la manipulación necesaria, antes de realizar los cambios en el DOM del navegador.

¡React sólo cambia lo que necesita ser cambiado!

React descubre qué cambios se han realizado y cambia **solo** lo que es necesario cambiar.

## Historia de React.JS

La última versión de React.JS es 19.0.0 (diciembre de 2024).

El lanzamiento inicial al público (versión 0.3.0) fue en julio de 2013.

React.JS se utilizó por primera vez en 2011 para la función Newsfeed de Facebook.

El ingeniero de software de Facebook, Jordan Walke, lo creó.

### Importar y exportar componentes

La magia de los componentes reside en su reusabilidad: puedes crear componentes que se componen a su vez de otros componentes. Pero mientras anidas más y más componentes,

a menudo tiene sentido comenzar a separarlos en diferentes archivos. Esto permite que tus archivos se mantengan fáciles de localizar y puedas reutilizar componentes en más lugares.

## Escribir marcado con JSX

JSX es una extensión de sintaxis para JavaScript que permite escribir marcado similar a HTML dentro de un archivo JavaScript. Aunque hay otras formas de escribir componentes, la mayoría de los desarrolladores de React prefieren la concisión de JSX, y la mayoría de las bases de código lo usan.

## JSX: Poniendo marcado dentro de JavaScript

La Web se ha construido sobre HTML, CSS, y JavaScript. Durante muchos años, los desarrolladores web mantuvieron el contenido en HTML, el diseño en CSS, y la lógica en JavaScript, ¡a menudo en archivos separados!. El contenido se marcó dentro del HTML mientras que la lógica de la página vivía por separado en JavaScript:

Pero, a medida que la Web se volvió más interactiva, la lógica determinó cada vez más el contenido. ¡JavaScript estaba a cargo del HTML! Esto es la razón por la que en React, la lógica de renderizado y el marcado viven juntos en el mismo lugar: componentes

Mantener juntas la lógica de renderizado y el marcado de un botón, garantiza que permanezcan sincronizados entre sí en cada edición. Por el contrario, los detalles que no están relacionados, como el marcado de un botón y el marcado de una barra lateral, están aislados entre sí, haciendo que sea más seguro cambiar cualquiera de ellos por su cuenta.

Cada componente de React es una función de JavaScript que puede contener algún marcado que React muestra en el navegador. Los componentes de React usan una extensión de sintaxis llamada JSX para representar el marcado. JSX se parece mucho a HTML, pero es un poco más estricto y puede mostrar información dinámica.

## JavaScript en JSX con llaves

JSX te permite escribir marcas similares a HTML dentro de un archivo JavaScript, manteniendo la lógica de renderizado y el contenido en el mismo lugar. A veces vas a querer agregar un poco de lógica JavaScript o hacer referencia a una propiedad dinámica dentro de ese marcado. En esta situación, puedes usar llaves en tu JSX para abrir una ventana a JavaScript.

### Usando llaves: Una ventana al mundo de JavaScript

JSX es una forma especial de escribir JavaScript. Eso significa que es posible utilizar JavaScript dentro de él, con llaves `{ }`. El ejemplo siguiente declara primero un nombre para el científico, `name`, y luego lo inserta con llaves dentro de `<h1>`:

### Usando "llaves dobles": CSS y otros objetos en JSX

Además de strings, números, y otras expresiones de JavaScript, incluso puedes pasar objetos en JSX. Los objetos también se indican con llaves, como `{ name: "Hedy Lamarr", inventions: 5 }`. Por lo tanto, para pasar un objeto de JavaScript en JSX, debes envolver el objeto en otro par de llaves: `person={{ name: "Hedy Lamarr", inventions: 5 }}`.

Puedes ver esto con estilos en línea CSS, en JSX. React no requiere que uses estilos en línea (las clases CSS funcionan muy bien para la mayoría de los casos). Pero cuando necesitas un estilo en línea, pasas un objeto al atributo `style`:

### Pasar props a un componente

Los componentes de React utilizan props para comunicarse entre sí. Cada componente padre puede enviar información a sus componentes hijos mediante el uso de props. Las props pueden parecerse similares a los atributos HTML, pero permiten pasar cualquier valor de JavaScript a través de ellas, como objetos, arrays y funciones.

Las props que puedes utilizar con una etiqueta `<img>` están predefinidas (ReactDOM se ajusta al estándar HTML). Sin embargo, puedes pasar cualquier prop a tus propios componentes, como `<Avatar>`, para personalizarlos

Puedes acceder a estas props especificando sus nombres `person`, `size` separados por comas dentro de `{ }` justo después de `function Avatar`. Esto te permitirá utilizarlas dentro del código de `Avatar` como si fueran variables.

Las props te permiten considerar de forma independiente los componentes padre e hijo. Por ejemplo, puedes modificar las props `person` o `size` dentro del componente `Profile` sin preocuparte por cómo serán utilizadas por el componente `Avatar`. De manera similar, puedes cambiar la forma en que `Avatar` utiliza estas props sin necesidad de revisar el componente `Profile`.

Considera las props como “controles” que puedes ajustar. Cumplen el mismo papel que los argumentos de una función—de hecho, ¡las props son el único argumento de tu componente! Las funciones de los componentes de React aceptan un único argumento, un objeto props:

Si quieres asignar un valor predeterminado para una prop en caso de que no se especifique ningún valor, puedes hacerlo mediante la desestructuración colocando `=` seguido del valor predeterminado justo después del parámetro:

## Bibliografía

<https://www.w3schools.com/react/default.asp>

<https://es.react.dev/>

<https://www.hostinger.com/es/tutoriales/que-es-react>

