

# Definición e Importancia de React JS

React es una biblioteca de JavaScript de código abierto y gratuita usada para construir interfaces de usuario basadas en componentes. Es mantenida por Meta (anteriormente Facebook) y una comunidad de desarrolladores individuales y empresas. Se puede utilizar para desarrollar aplicaciones de una sola página o móviles.

Debido a que React se preocupa únicamente por la interfaz de usuario y la renderización de componentes en el DOM, las aplicaciones de React a menudo dependen de bibliotecas para otras funcionalidades del lado del cliente.

La principal característica de React es su enfoque en la construcción de interfaces de usuario mediante la creación de componentes reutilizables.

# Definición e Importancia de React JS

Los componentes en React son bloques de construcción independientes que encapsulan la lógica y la interfaz de usuario de una parte específica de la aplicación. Estos componentes pueden ser combinados y reutilizados para construir interfaces complejas.

React utiliza un enfoque basado en el DOM virtual para optimizar el rendimiento de las aplicaciones. En lugar de actualizar directamente el DOM cada vez que cambia el estado de la aplicación, React utiliza un DOM virtual para realizar cambios eficientes y minimizar la manipulación directa del DOM.

En resumen, React es una biblioteca de JavaScript que permite construir interfaces de usuario interactivas y reutilizables mediante la creación de componentes. Su enfoque en la eficiencia y la reutilización de código lo ha convertido en una herramienta popular para el desarrollo de aplicaciones web modernas.

# Historia de React JS

**2011**



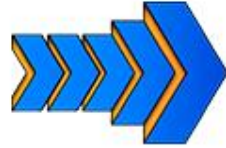
Creado por Facebook para construir su interfaz de usuario

**2012**



Instagram construye su sitio web usando React

**2013**



React fue presentado como Código Abierto

**2014**



Hay conferencias a nivel mundial proponiendo el uso de React



**2022**



React 18.2.0 es lanzado el 14 de Junio de 2022



**2016**



Se publica el sistema de Códigos de Error para React



**2015**



Netflix y Airbnb comienzan a usar React



**2014**



React Developer Tools es agregado como una extensión para Google Chrome

# Conceptos clave para usar React JS

## Lo que sí es React:

- Es una Librería para el desarrollo de Frontend moderno
- Se basa en Componentes
- Es declarativo, no nos indica que hacer si no al contrario, debemos indicarle qué hacer
- Es rápido, eficiente y fácil de aprender

## Lo que no es React:

- No es un Framework
- No es una herramienta de Backend
- No es un reemplazo de JavaScript
- No es una solución completa



# Sitios web más Populares que usan React



**Facebook**



**Instagram**



**Netflix**



**Airbnb**



**Pinterest**



**Dropbox**



**WhatsApp**

# Razones para Aprender a usar React

- **Popularidad:** React es una de las bibliotecas más populares y ampliamente utilizadas en el desarrollo web, lo que significa que hay una gran demanda de desarrolladores con experiencia en React en el mercado laboral.
- **Comunidad Activa:** React cuenta con una comunidad de desarrolladores muy activa y comprometida. Esto significa que siempre hay recursos, tutoriales y ejemplos disponibles para ayudarte a aprender y resolver problemas.
- **Reutilización de componentes:** React se basa en el concepto de componentes reutilizables, lo que te permite crear piezas de código independientes y reutilizables. Esto facilita el mantenimiento y la escalabilidad de tus proyectos.

# Razones para Aprender a usar React

- **Virtual DOM:** React utiliza un Virtual DOM (DOM virtual) para actualizar de manera eficiente solo las partes de la interfaz de usuario que han cambiado. Esto mejora el rendimiento y la velocidad de las aplicaciones.
- **React Native:** React también se puede utilizar para desarrollar aplicaciones móviles utilizando React Native. Esto te permite crear aplicaciones nativas para iOS y Android utilizando el mismo código base de React.
- **Flexibilidad:** React es muy flexible y se puede integrar fácilmente con otras bibliotecas o frameworks. Puedes utilizar React en combinación con herramientas como Redux, GraphQL, Next.js, entre otras.

# Razones para Aprender a usar React

- **Curva de aprendizaje suave:** Aprender React no es tan complicado como puede parecer. La sintaxis es fácil de entender, especialmente si ya tienes conocimientos de JavaScript.
- **Documentación completa:** React cuenta con una documentación oficial muy completa y bien organizada. Encontrarás guías, tutoriales y ejemplos que te ayudarán a aprender.
- **Oportunidades de carrera:** La demanda de desarrolladores con experiencia en React es alta. Aprender React te abrirá muchas oportunidades de trabajo en el campo del desarrollo web.



# Uso del DOM y del Virtual DOM en React

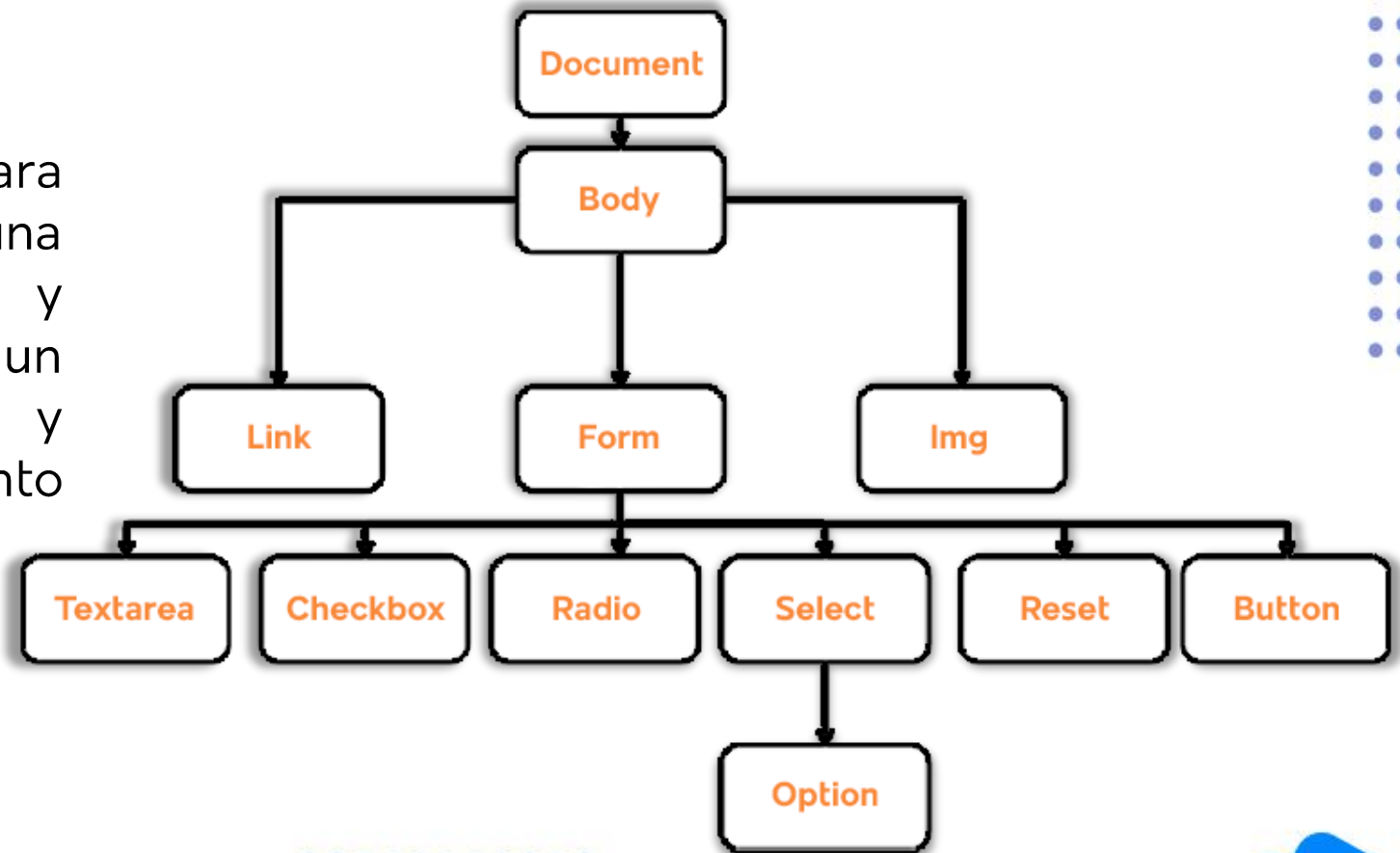
Trabajando en React, se crean componentes que representan partes de la interfaz de usuario. Estos componentes están compuestos por elementos JSX, que son similares a las etiquetas HTML pero en JavaScript.

Cuando se renderiza un componente de React, React genera una representación virtual de la interfaz de usuario llamada Virtual DOM. El Virtual DOM es una copia ligera y eficiente del DOM real.

# Uso del DOM y del Virtual DOM en React

## El DOM (Document Object Model):

Es una interfaz de programación para documentos HTML y XML. Es una representación estructurada y jerárquica del contenido de un documento, que permite acceder y manipular los elementos del documento de manera programática.



# Uso del DOM y del Virtual DOM en React

**Aquí está el proceso paso a paso de cómo trabaja React con el DOM y el Virtual DOM:**

- 1. Renderizado Inicial:** Cuando se renderiza un componente de React por primera vez, React crea una estructura de Virtual DOM que representa la interfaz de usuario deseada.
- 2. Comparación:** Cuando se actualiza el estado de un componente o se reciben nuevas propiedades, React compara la nueva estructura del Virtual DOM con la anterior.
- 3. Detección de Cambios:** React detecta las diferencias entre la nueva y la anterior estructura del Virtual DOM para determinar qué partes de la interfaz de usuario deben actualizarse.

# Uso del DOM y del Virtual DOM en React

- 4. Actualización del Virtual DOM:** React actualiza el Virtual DOM con las diferencias detectadas, creando un nuevo Virtual DOM.
- 5. Reconciliación:** React compara el nuevo Virtual DOM con el DOM real y determina los cambios mínimos necesarios para actualizar el DOM real.
- 6. Actualización del DOM:** React aplica los cambios necesarios al DOM real de manera eficiente y actualiza solo las partes que han cambiado.

# Programar en React es Pensar en Componentes

React puede cambiar tu forma de pensar en los diseños que miras y las aplicaciones que construyes. Cuando haces una interfaz de usuario con React, primero es necesario separarla en piezas denominadas componentes.

**Los Componentes** en React son bloques de construcción que encapsulan la lógica y la interfaz de usuario de una parte específica de la aplicación. Al dividir la interfaz de usuario en componentes más pequeños y reutilizables, se facilita el desarrollo, el mantenimiento y la escalabilidad del código.

Por ello, a continuación, separaremos en componentes una tabla de datos de productos con una funcionalidad de búsqueda:

☐ Only show products in stock

Name	Price
<b>Fruits</b>	
Apple	\$1
Dragonfruit	\$1
Passionfruit	\$2
<b>Vegetables</b>	
Spinach	\$2
Pumpkin	\$4
Peas	\$1



# Programar en React es Pensar en Componentes

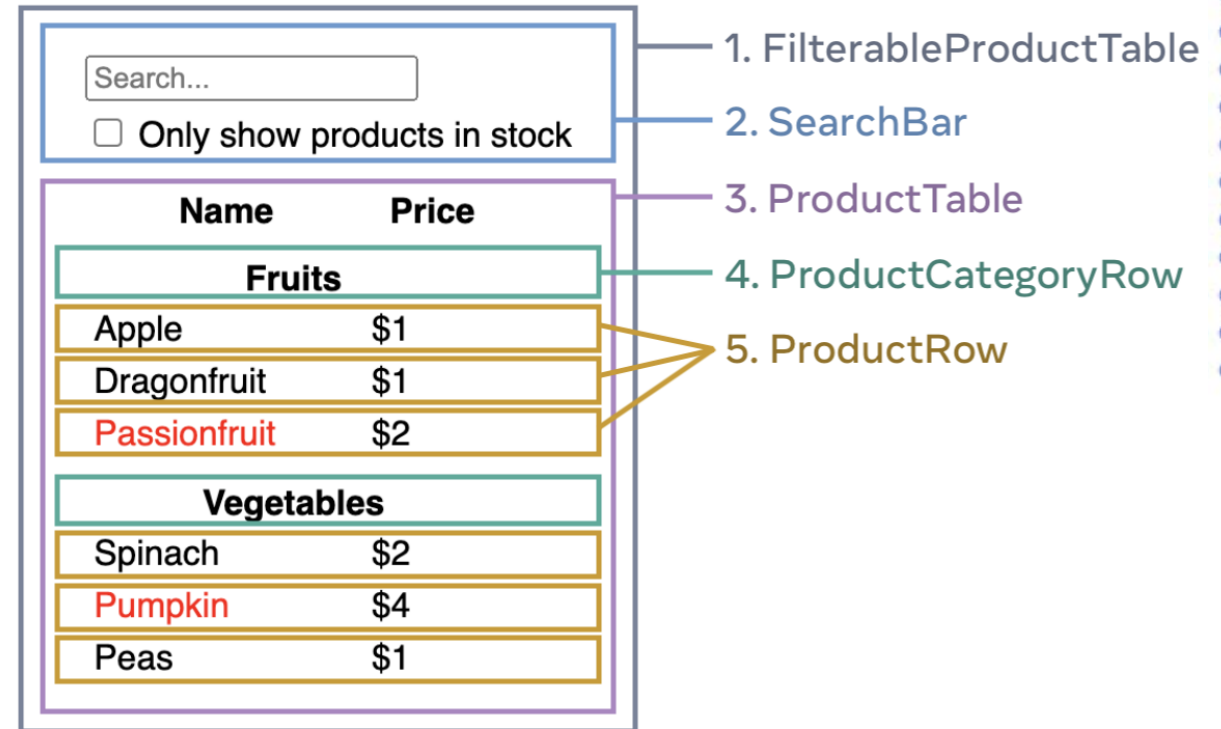
Dibujaremos cuadros alrededor de cada componente y subcomponente en el boceto y los nombraremos. Es probable pensar en dividir un diseño en componentes de distintas maneras:

- **Programación:** Lo ideal es que un componente sólo haga una cosa. Si termina creciendo, debería descomponerse en subcomponentes más pequeños.
- **CSS:** Analiza para qué harías selectores de clase.
- **Diseño:** Considera cómo organizarías las capas del diseño.

# Programar en React es Pensar en Componentes

Hay cinco (5) componentes en esta pantalla:

1. **FilterableProductTable:** Contiene toda la aplicación.
2. **SearchBar:** Recibe la entrada del usuario.
3. **ProductTable:** Muestra y filtra la lista de acuerdo a la entrada del usuario.
4. **ProductCategoryRow:** Muestra un encabezado para cada categoría.
5. **ProductRow:** Muestra una fila para cada producto.



Programar en React implica pensar en componentes porque se basa en la idea de construir interfaces de usuario mediante la composición de componentes reutilizables.

# Software y Herramientas a usar para programar en React

- **Un Navegador:** Se recomienda que sea Mozilla Firefox o Google Chrome.
- **React Developer Tools:** Es una extensión que puedes instalar en los navegadores anteriormente recomendados.
- **Git:** Su consola es mucho más amigable y cómoda para trabajar con React. Al momento de instalarla debemos seleccionar la opción de "Use Git from de Windows Command Prompt" para poder usarla dentro de nuestro editor de código en caso de ser necesario.
- **Editor de Código:** Preferiblemente que sea Visual Studio Code
- **Node JS:** Es un entorno de ejecución de JavaScript que te permitirá ejecutar y administrar tus proyectos de React.