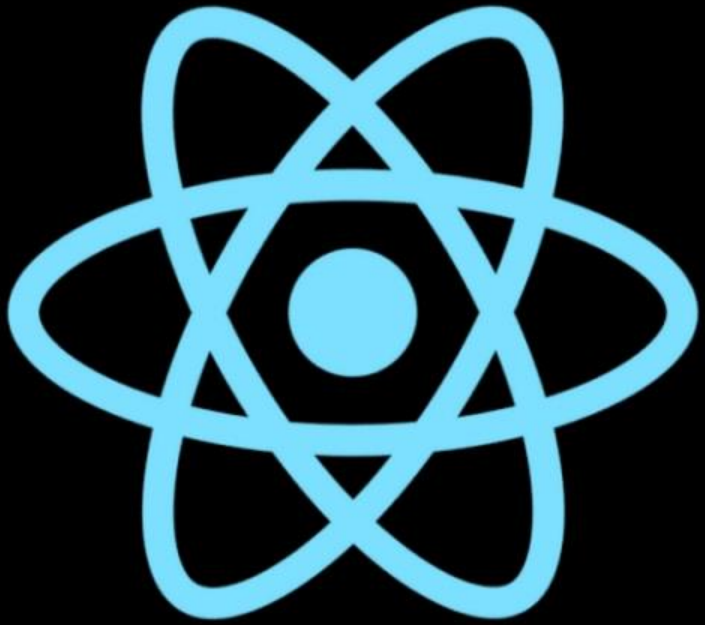
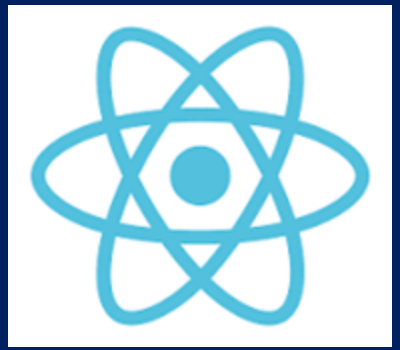


React JS

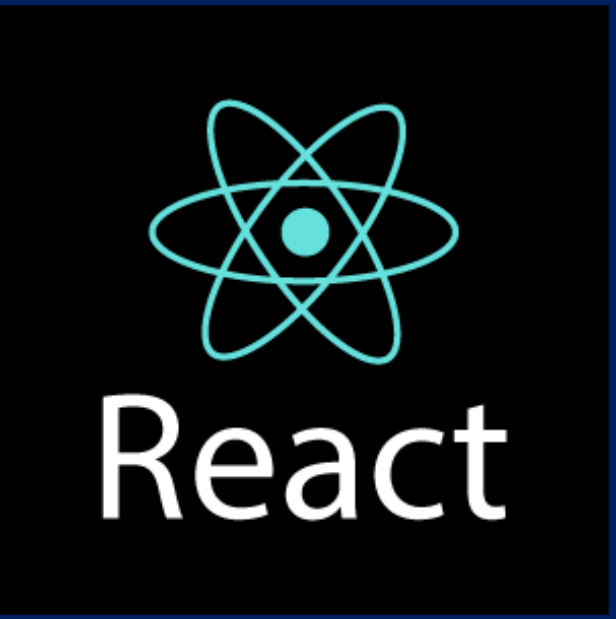


React es una Libreria de JavaScript de código abierto creada por Jordan Walke en Facebook para solucionar problemas de rendimiento al manipular el DOM, y se lanzó al público en 2013. Su enfoque en componentes y un DOM virtual simplificó la creación de interfaces de usuario (UI) dinámicas, y desde entonces ha crecido gracias a actualizaciones continuas, un ecosistema robusto y una comunidad global.

Librería o Framework?

https://react.dev/ (web oficial)
https://react.dev/learn/react-developer-tools
(herramienta de desarrollo)
https://nodejs.org/es (nodejs - ya lo tenemos)

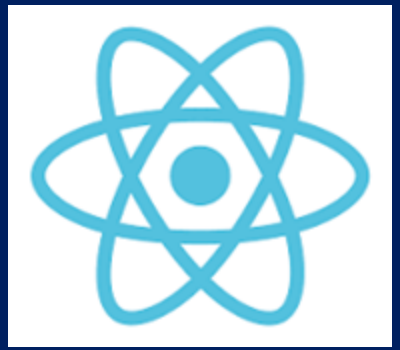
Línea de tiempo de React



- **2011** — Es creado porFacebook
- **2012** — Es usado por Instagram
- **2013** — Es liberado como código abierto
- **2014** — Es adoptado por muchas compañias
- **2015** — React Native es liberado
- **2016** — React 15 es liberado
- **2018** — React Hooks es liberado
- **Hoy** — Más de **50k** componentes desarrollados
  Facebook tiene un equipo completo dedicado
  Tiene una comunidad extensa y activa

@suabochica

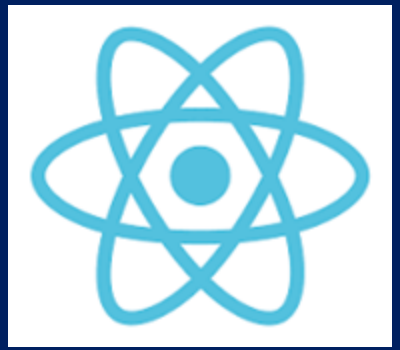
FUNCIONAMIENTO DE REACT JS



¿Cómo llega React a la performance que tanta fama le trae?

Hablemos de tres conceptos:

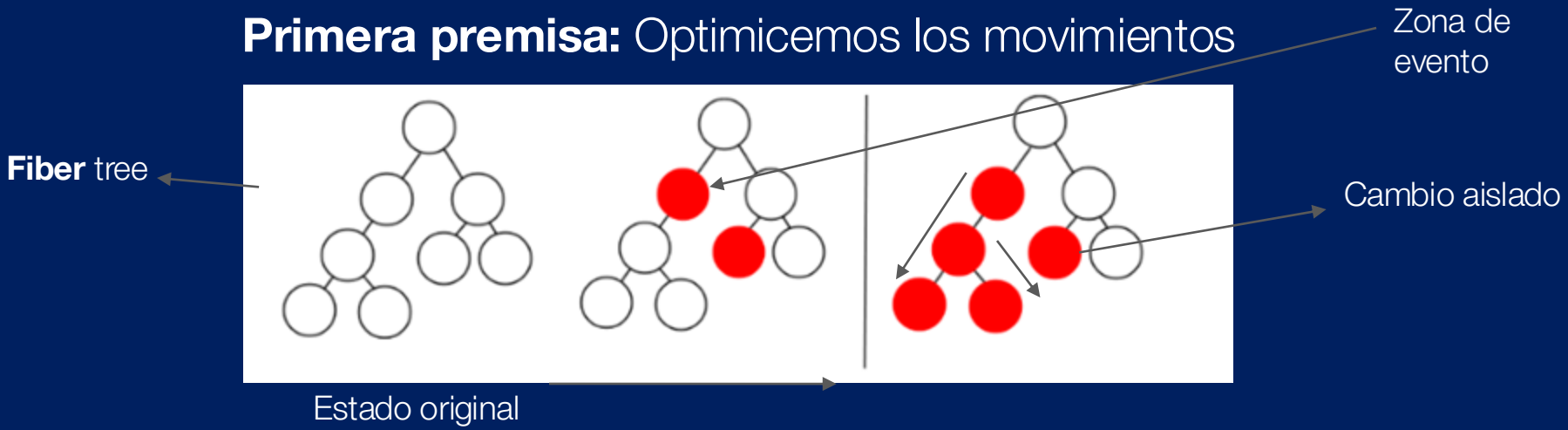
Virtual DOM,Diffing y Reconciliación

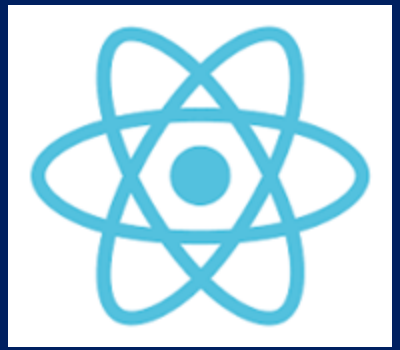


Primera premisa

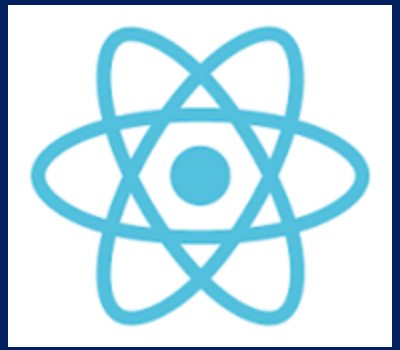
El acceso indiscriminado al DOM **es caro**, entonces se requirió encontrar una manera de realizarlo de la manera **más óptima** posible.

Primera premisa: Optimicemos los movimientos





En vez de aplicar **uno a uno** los cambios en los cinco nodos, **React** procesa este resultado en una memoria. Calcula el área de impacto y determina la menor cantidad de movimientos de modo **heurístico**, por lo que también sabe donde **no pueden haber ocurrido** cambios.



Segunda premisa: Flujo de datos unidireccional

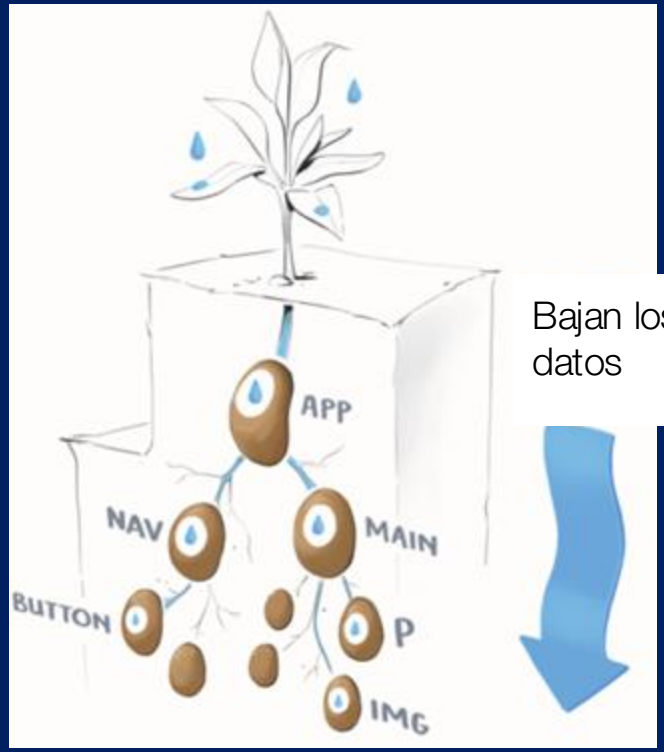
Para establecer esa seguridad, requiere que los datos y los cambios idealmente se provoquen de una manera específica con dos características:

Unidireccionalidad / De arriba **hacia abajo**

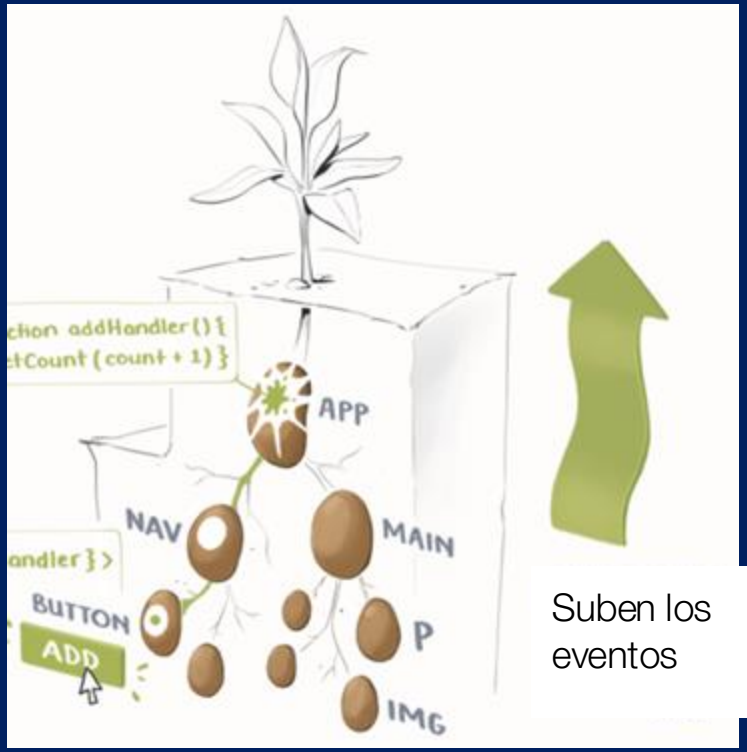
FLUJO DE DATOS

Reacción

Acción



Bajan los datos



Suben los eventos

Ilustradora: Maggie Appleton @ Woman of React 2020

RESUMIENDO:
¿QUÉ ES EL VIRTUAL DOM?

VIRTUAL DOM

Es un patrón de comportamiento, y **React** lo implementa con una tecnología llamada “**Fiber**”.

En sí resulta ser todo lo que React sabe de tu aplicación y cada nodo o **fibra.**

Esto es básicamente lo que React hace con el Virtual DOM: **una representación virtual de la UI que se mantiene en memoria y en sincronía “reconciliado” con el DOM “real”.**

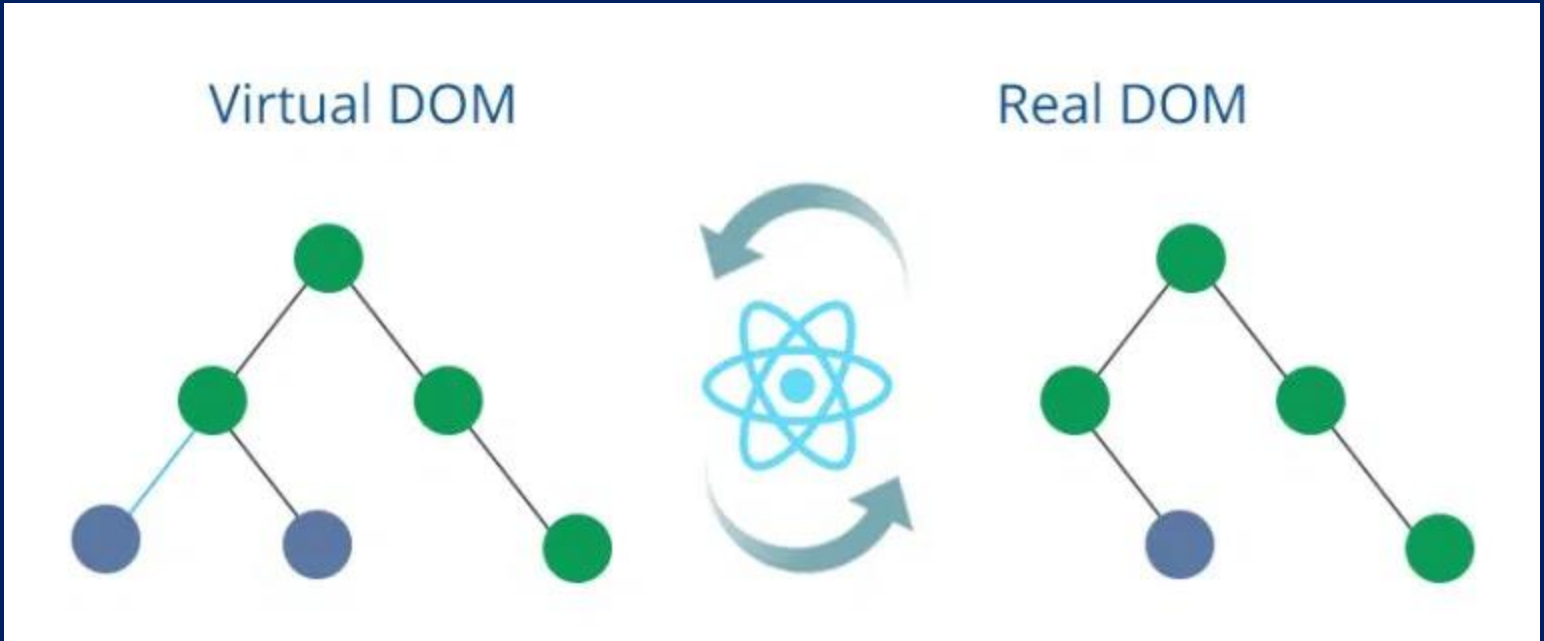
VIRTUAL DOM

Resumiendo el proceso

- En primer lugar, React ejecuta un algoritmo de “**diffing**” que identifica lo que ha cambiado.
- El segundo paso es la **reconciliación**, donde se actualiza el **DOM** con los resultados de **diff**.

React se encarga de todo esto, nosotros solo aprenderemos a ayudarlo

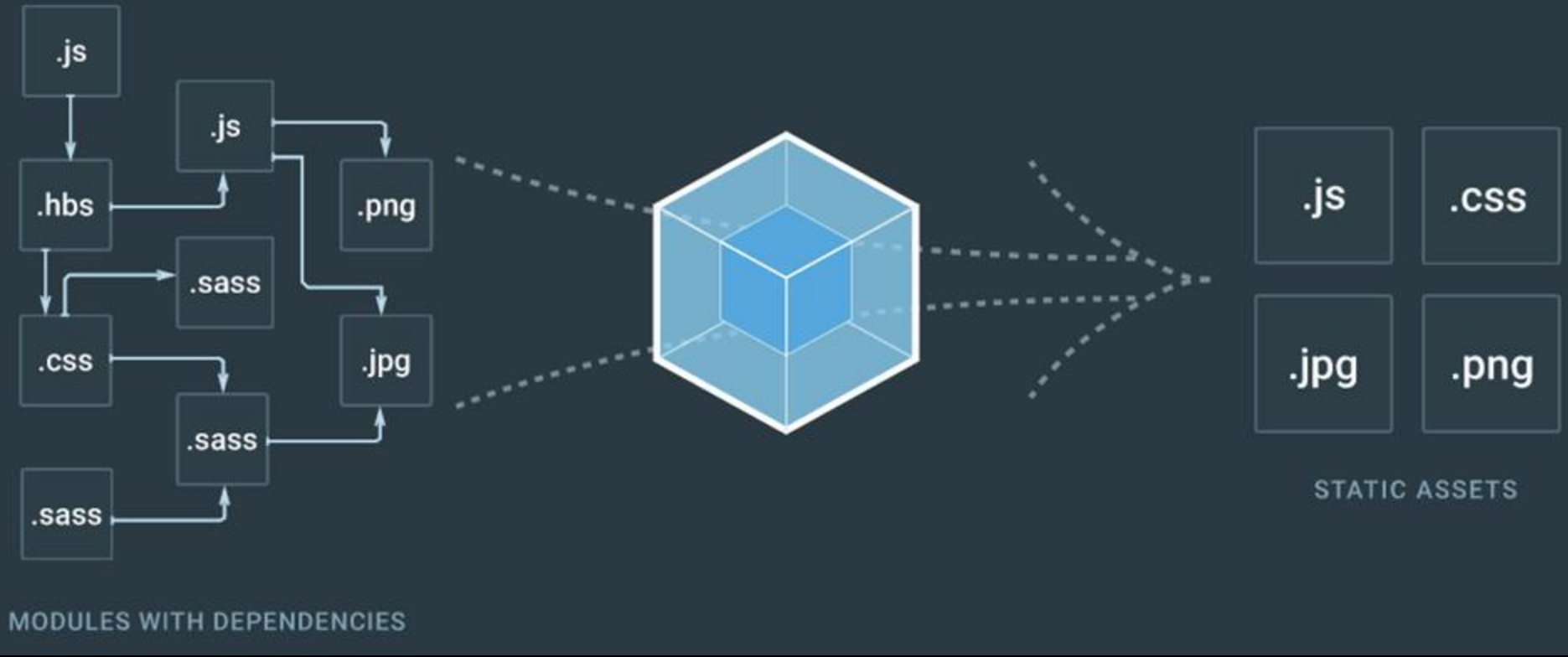
VIRTUAL DOM



webpack

Webpack es un *module bundler* o empaquetador de módulos que nació a finales de 2012, y en la actualidad es utilizado por miles de proyectos de desarrollo web Front-End.

Incluido desde React o Angular hasta en el desarrollo de aplicaciones conocidas como Twitter, Instagram, PayPal, o la versión web de Whatsapp.



Transformación de los módulos en Webpack.

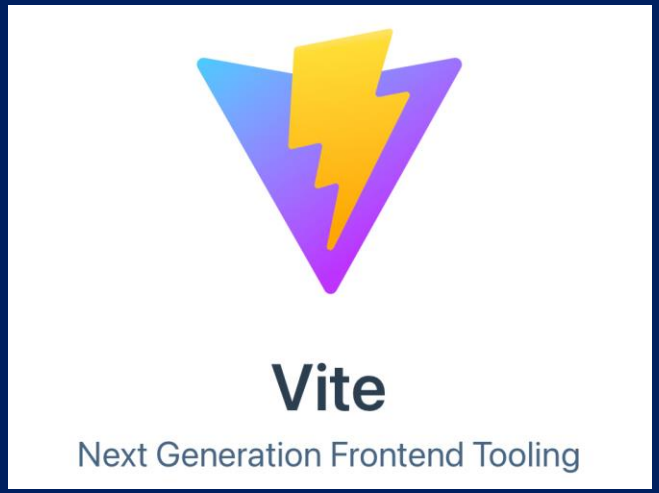
¿CÓMO FUNCIONA?

Podemos tener, por ejemplo, un módulo JS que vaya a depender de otros módulos .js, con imágenes en diferentes formatos como JPG o PNG. O estar utilizando algún preprocesador de CSS, como puede ser SASS, Less y Stylus.

Webpack recoge todos estos módulos y los transforma a assets que puede entender el navegador, como por ejemplo archivos JS, CSS, imágenes, videos, etc.

NO USAREMOS WEBPACK

En su lugar…. Usaremos:



https://vite.dev/

¿QUÉ ES EL TRANSPILING?

TRANSPILING

Es el proceso de **convertir código** escrito en un lenguaje, **a su representación en otro lenguaje**. Usualmente extienden o simplifican la escritura del lenguaje, o representación original.

- Implementan un proceso similar conceptualmente al **pollyfilling.**
- Logran niveles de simetricidad y simbiosis con el lenguaje original.

JSX

¿QUÉ ES Y POR QUÉ LO USAMOS?

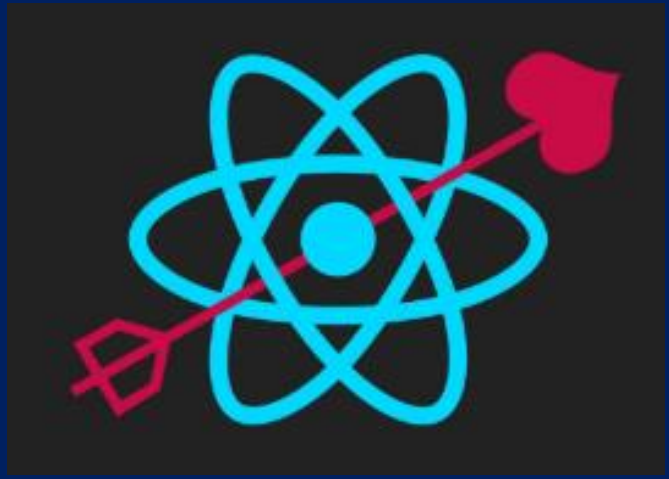
java**s**cript **x**ml

JSX es una extensión de sintaxis de Javascript que se parece a HTML

Oficialmente, es una **extensión que permite hacer llamadas a funciones y a construcción de objetos**. No es ni una cadena de caracteres, ni HTML.

JSX es una extensión de Javascript, no de React.

Esto significa que **no hay obligación de utilizarlo**, pero **es recomendado** en el sitio web oficial de React.



FUNCIONAMIENTO Y CARACTERÍSTICAS

¿CÓMO FUNCIONA?

JSX se transforma en código JavaScript.

https://babeljs.io/repl

Esto nos da algunas ventajas, como ver errores en tiempo de compilación, asignar variables, retornar métodos, etc.

STYLING EN JSX

Es posible definir y utilizar estilos inline en JSX, solo necesitamos convertirlos por convención:

border-color => borderColor
padding-top => paddingTop
'10px' => 10 **(no es necesario el px)**

INLINE STYLES EN JSX

Los mismos estilos se pueden configurar **inline** en JSX, solo necesitamos usar doble llave {{ }},

- La **primera** llave para avisar que se agregará **un objeto** en **js.**
- La **segunda** llave para empezar a escribir el objeto en sí.

```
const Salute = () => <p style={{ marginLeft: 15}}>Hello</p>
```

REGLAS GENERALES

- **Los elementos deben ser balanceados.** Por cada apertura debe haber un cierre.

```
<img src=""> Mal
<img src=""></img> Es mejorable
```

- **Si el elemento no tiene hijos, debe idealmente ser auto-cerrado**

```
<img src="" /> Ideal
```

REGLAS GENERALES

Class es palabra reservada, en su lugar usar className.

```
<img src="" class="my-class" /> Mal
<img src="" className="my-class" /> Ok
```

```
return (
  <h1 className='large'>Hello World</h1>
);
```

¡NO OLVIDEMOS!

En JSX se utilizan tanto los estilos como los eventos estándar del DOM, como **`onclick`**, **`onchange`**, **`onkeydown`**, etc. pero utilizando **camelCase**: `onClick`, `onChange`, `onKeyDown` / `marginTop`, `paddingBottom`, etc.

A medida que nuestra aplicación va creciendo y tenemos componentes más grandes que manejan distintos eventos, JSX nos va a ayudar mucho a agilizar y organizar nuestros desarrollo de componentes.

Iniciemos Nuestro primer Proyecto!!