

# Bootcamp de Full Stack F2

Bienvenidos a la clase N°41

- ¿Qué es React?
- Componentes de React
- ¿Qué es JSX & Babel?
- ¿Qué es el Virtual DOM?
- ¿Qué es Vite?
- ¿Qué es NPM?
- ¿Qué es un linter & ESLint?
- Mi Primer Proyecto

# Definiciones

## React

- **React** es una biblioteca para construir interfaces de usuario.
- React no es un framework ni siquiera se limita a la web.
- Se utiliza con otras bibliotecas para renderizar en ciertos entornos.

El objetivo principal de React es minimizar los errores que ocurren cuando los desarrolladores construyen interfaces de usuario. Esto lo hace mediante el uso de componentes piezas de código lógicas y auto-contenidas que describen una parte de la interfaz del usuario. Estos componentes se pueden juntar para crear una interfaz de usuario completa que permite enfocarse en el diseño de la interfaz. Desde [aquí](#), puedes acceder a la documentación oficial de React.

# Definiciones

## Componentes de React JS

En React existen dos tipos principales de componentes:

- **Componentes Funcionales:** Son funciones de JavaScript que reciben props (propiedades) como argumentos y devuelven elementos de React que describen cómo debería verse la interfaz de usuario.
- **Componentes de Clase:** Son clases de JavaScript que extienden la clase `React.Component`. Estos componentes tienen un estado interno y pueden tener los métodos del ciclo de vida de React.



# Definiciones

## JSX & Babel

The logo for JSX, consisting of the letters 'J', 'S', and 'X' in a stylized font. The 'J' and 'S' are light blue, and the 'X' is red.

- React acepta el hecho de que la lógica de renderizado está intrínsecamente unida a la lógica de la interfaz de usuario: cómo se manejan los eventos, cómo cambia el estado con el tiempo y cómo se preparan los datos para su visualización. En lugar de separar artificialmente tecnologías poniendo el maquetado y la lógica en archivos separados, React separa intereses con unidades ligeramente acopladas llamadas “componentes” que contienen ambas. Dado que JSX es más cercano a JavaScript que a HTML, React DOM usa la convención de nomenclatura camelCase en vez de nombres de atributos HTML. Más info [aquí](#).

The logo for Babel, featuring the word 'BABEL' in a stylized, handwritten yellow font.

- **Babel** es un "transpilador" para JavaScript. Básicamente, permite transformar código escrito con las últimas y novedosas características de JavaScript y que lo convierte en un código que sea comprendido por los navegadores más antiguos. Más info [aquí](#).

# Definiciones

## Virtual DOM

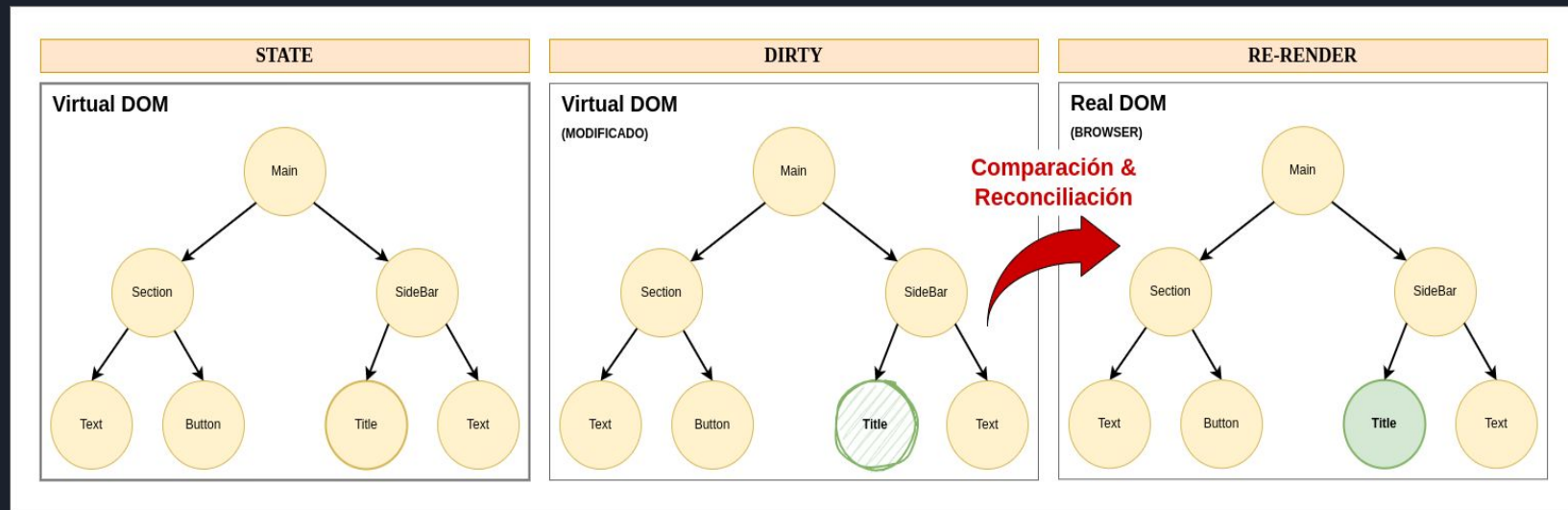
**React** guarda una réplica del **DOM** en memoria, llamada **virtual DOM**. Este es el mecanismo de actualización cuando hay un cambio de estado:

1. Se genera un cambio de estado (state).
2. Se ejecuta un algoritmo diferencial para ver qué ha cambiado (dirty).
3. Se ejecuta un proceso de reconciliación, que actualiza la réplica del DOM con los resultados del dirty.
4. Por último, se ejecuta un proceso de renderizado que consiste en que cuando han terminado las actualizaciones en el Virtual DOM, se actualiza el DOM real sólo con los cambios que se hayan producido entre ambos.

El proceso de reconciliación se da cuando algún nodo sufre uno o más cambios de estado.

# Definiciones

## Virtual DOM



# Definiciones

## Vite



Vite es una herramienta para Frontend, creada por Evan You, mismo creador de Vuejs, esta herramienta nos permitirá crear aplicaciones de Javascript usando algún framework o librería como ReactJs, VueJs, Angular ya sea con algún framework específico o ya sea Vanillajs. Más info [aquí](#).

### Características

- Inicio Instantáneo del Server: Nos brinda los archivos que necesitamos en el momento que necesitamos, a través de módulos de javascript nativos (EcmaScript Module), con lo cual no tendremos que lidiar con el building.
- Rápido HMR: Sin importar el tamaño de la aplicación, podremos ver reflejados los cambios en nuestro codebase de manera instantánea.
- Build Optimizado: De manera interna ya trae pre-configurado Rollup con soporte a múltiples páginas y soporte para librerías.
- Plugins Universales: Al hacer uso de Rollup de manera interna, este provee una interfaz compartida entre desarrollo y build.

# Definiciones

## NPM

Es el gestor de dependencias oficial de NodeJS y sirve para mantener el software del que dependen las aplicaciones que se desarrollan con Javascript o Node. La palabra npm es la sigla de "Node Package Manager" y básicamente consiste en una herramienta de línea de comandos que se usa para instalar y actualizar dependencias en proyectos Javascript o NodeJS, así como publicar packages que se podrán usar en otros proyectos. Más info [aquí](#).





# Definiciones

## Linter & Eslint

Un **linter** o también conocido como analizador de código estático, es una herramienta de software que verifica y analiza el código fuente de un programa en busca de posibles errores, problemas de estilo y violaciones de las convenciones de codificación.

**ESLint** es una popular herramienta de linting para el lenguaje de programación JavaScript. Proporciona análisis estático de código para identificar y corregir errores, así como para aplicar reglas y convenciones de codificación en proyectos de desarrollo de JavaScript.

**ESLint** utiliza el archivo `.eslintrc.cjs` para la configuración principal. Este es un archivo de texto que especifica las reglas y las configuraciones personalizadas que se deben aplicar al analizar el código.



# BREAK

Descansemos 15 minutos



# React

## Mi Primer Proyecto

- Instalar las extensiones de VSC (archivo FullStack.code-profile)
- Crear proyecto: `npm create vite@latest mi_primer_proyecto`
- Instalar dependencias:
  - `npm install axios --save`
  - `npm install prop-types --save`
  - `npm install sass --save-dev`
- Establecer las reglas para Eslint (archivo .eslintrc.cjs)
- Personalizar proyecto
- Codear "Hola Mundo"
- Aplicar estilos al "Hola Mundo"
- Levantar servidor de React: `npm run dev`



# CIERRE DE CLASE

Continuaremos en la próxima clase

