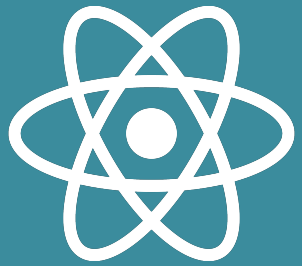
www.cesarguerra.mx



Introducción a React JS

Fundamentos y Creación de Proyectos con React

CONTENIDO

01 LIBRERÍAS Y FRAMEWORK
¿Cuál es la diferencia?

02 INTRO A REACT
¿Qué es y cómo funciona?

03 COMENZAR EN REACT
Creación de proyecto y estructura

04 USANDO REACT
JSX, Fragments y Props

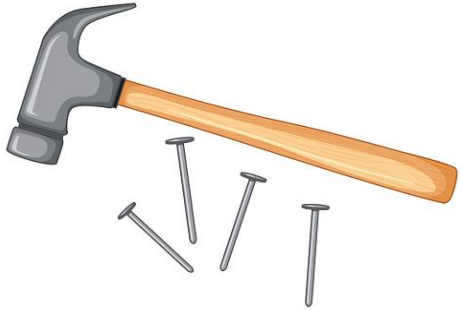
01
LIBRERÍAS Y FRAMEWORKS

¿CÓMO TRABAJABAMOS EN DESARROLLO?

Cuando nos adentramos en el mundo del desarrollo de software queremos construir aplicaciones mucho más rápido y las **librerías y frameworks** son herramientas que nos ayudan.

Pero,
¿Qué es una librería y que es un framework?

LIBRERÍAS Y MARCOS DE TRABAJO



Una librería es un conjunto de funcionalidades que resuelven necesidades específicas del proyecto, empaquetadas y reutilizables.

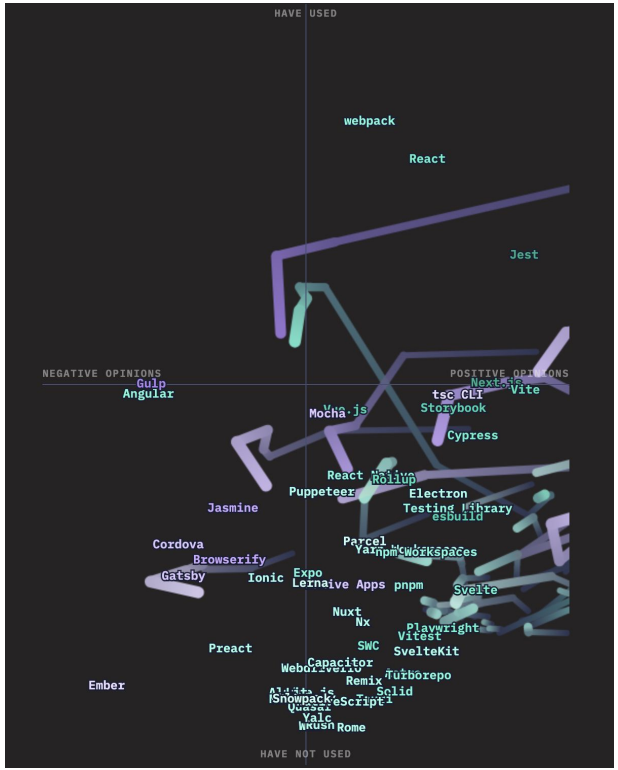
Puedes utilizar múltiples librerías en un proyecto.

Un framework te da toda la estructura para un proyecto completo, desde el inicio hasta el final, integrando múltiples funcionalidades de fábrica, asegurando así la compatibilidad entre sus componentes y asegurando un proyecto de estructura homogénea.

EN RESUMEN....

Referencia: EDTeam (s.f.). Framework vs Libreria

¿CUALES SON LAS MÁS USADAS?



RETENTION VS USAGE

Get JSON Data | Get GraphQL Query

Label	User Count	Retention %	Interest %
react	27289	82.95%	10.46%
vuejs	15379	77.33%	58.51%
angular	16257	42.62%	21.25%
preact	4252	74.2%	176.15%
ember	2483	17.04%	124.24%
svelte	7025	89.62%	239.16%
alpinejs	2029	76.1%	242.98%

https://2022.stateofjs.com/en-US/libraries/

02

REACT JS

¿QUÉ ES REACT?

Es una biblioteca JavaScript de código abierto (licencia MIT), diseñada con el **objetivo de crear interfaces de usuario** para facilitar el desarrollo de aplicaciones en una sola página (**SPA - Single Page Application**).

Es mantenido por Meta y la comunidad de software libre.

Creador: Jordan Walke - 2011

Soportado: Meta (Facebook)

Última Versión: React 18+ (29 Marzo 2022)

Charla de lanzamiento de React (2013):
https://www.youtube.com/watch?v=GW0rj4sNH2w

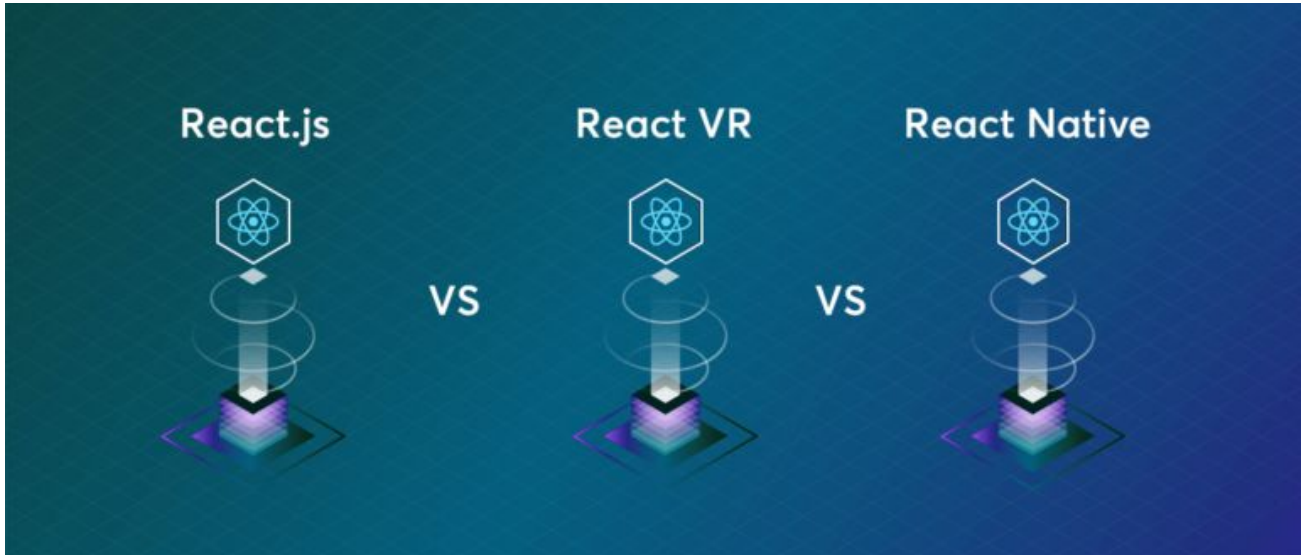
¿Por qué aprender React?

React es una librería o biblioteca que se destaca por ser simple y por tener una API muy amigable que la podemos entender como si fuera JavaScript.

Actualmente tiene una alta demanda laboral, empresas y startups como: **Airbnb, Facebook, Instagram, Uber utilizan React**.

¿POR QUE APRENDER REACT?

Normalmente cuando hablamos de React, nos referimos a React.js o ReactJS, pero de dentro del mismo troncal (de React) salen tres ramas diferentes: ReactJS, React Native y React VR.

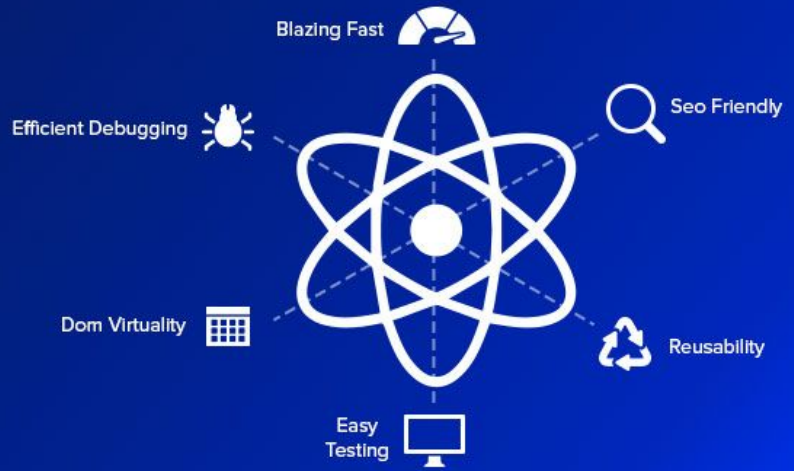


Beneficios de React

Veloz y reutilizable: Por su enfoque basado en SPA, Virtual DOM y en componentes que pueden ser reutilizados.

Respaldado por Meta: Asegurando su uso, evolución y respaldando a la comunidad.

Estabilidad: React es ampliamente usado en la industria y ha sido probado en múltiples escenarios.



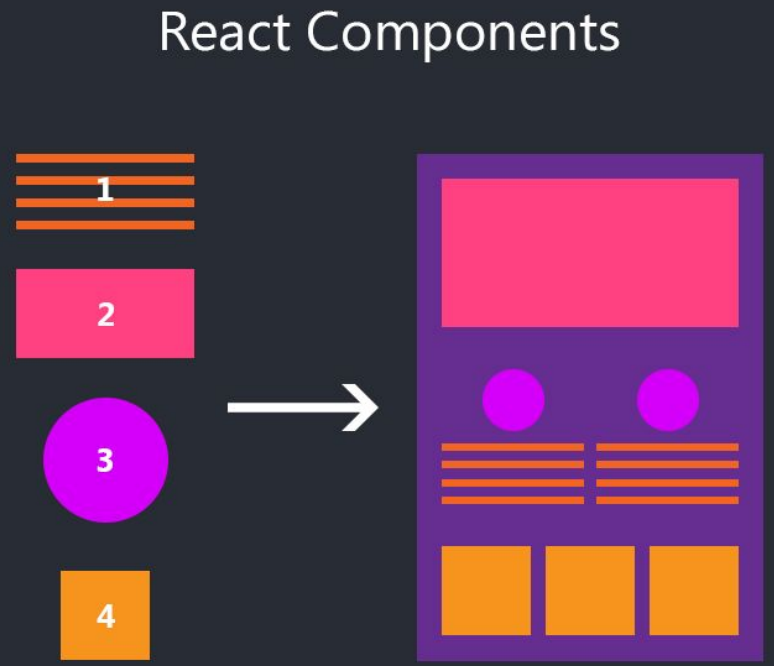
03

¿CÓMO FUNCIONA REACT JS?

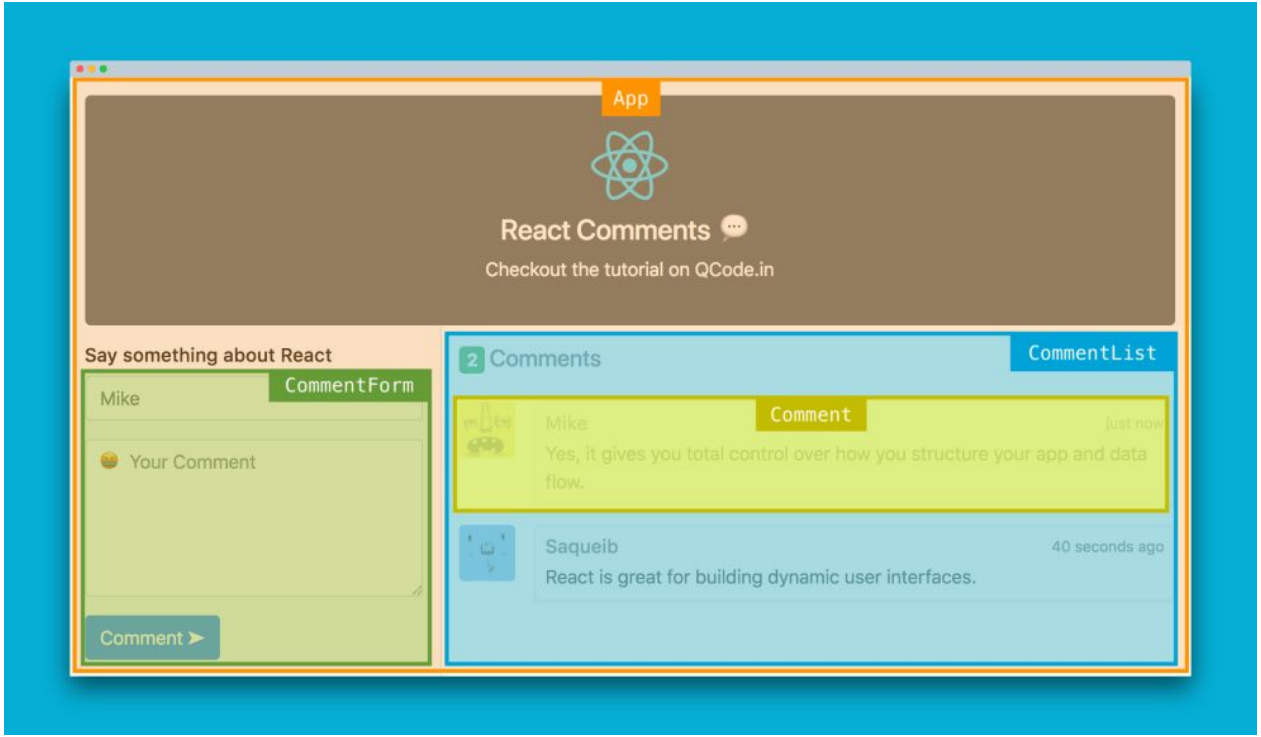
COMPONENTES

El objetivo principal de React es organizar los elementos sobre una interfaz dividiendo dicha interfaz de usuario (UI) en una colección/conjunto qué llamaremos componentes.

Los componentes nos permiten dividir una aplicación web en pequeñas partes donde cada parte es independiente y aunque estas partes sean independientes las podemos trabajar juntas o unirlas para realizar una aplicación web.



Componentes



Cada componente tiene un solo trabajo en específico.

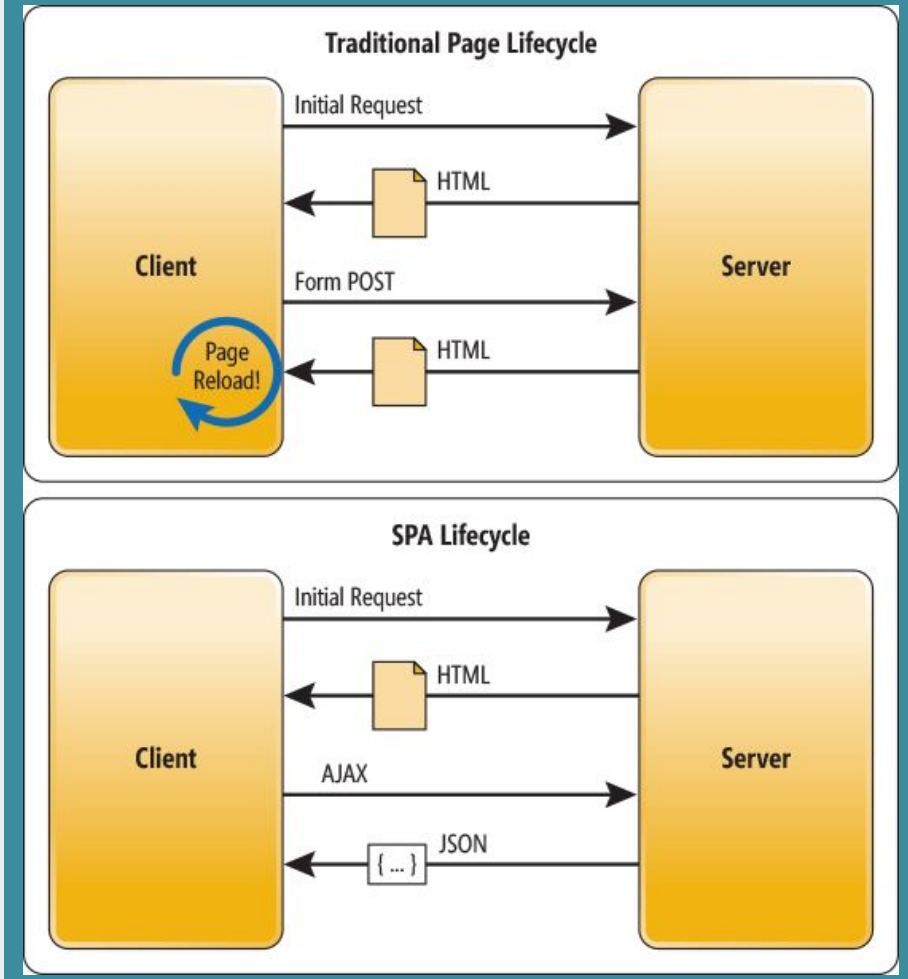
React está pensando basado en la tecnología de **atomic design**.

Un componente puede tener dentro más componentes.
Esto se llama nested components.

Single Page Application (SPA)

Una Single-Page Application (SPA) es un tipo de aplicación web que ejecuta todo su contenido en una sola página.

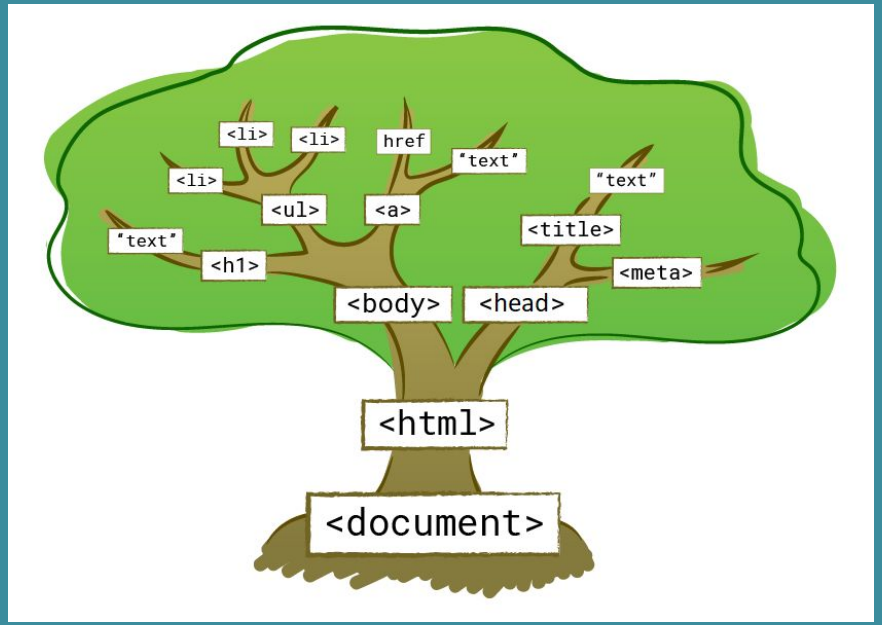
Funciona cargando el contenido HTML, CSS y JavaScript por completo al abrir la web. Al ir pasando de una sección a otra, solo necesita cargar el contenido nuevo de forma dinámica si este lo requiere, pero no hace falta cargar la página por completo. Esto mejora los tiempos de respuesta y agiliza mucho la navegación, favoreciendo así a la experiencia de usuario.



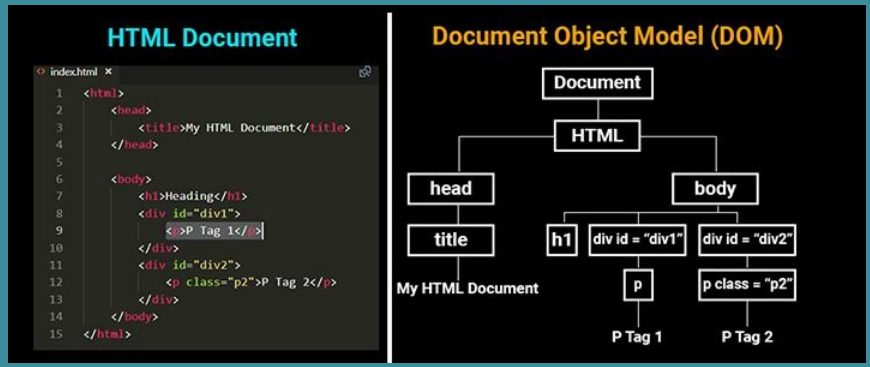
DOM
Document Object Model

Una página HTML está formada por múltiples etiquetas HTML, anidadas una dentro de otra, formando un árbol de etiquetas relacionadas entre sí, que se denomina árbol DOM (o simplemente DOM).

El DOM define la manera en que objetos y elementos se relacionan entre sí en el navegador y en el documento, permitiendo de esta forma a un lenguaje de programación (como JavaScript) manipularlo. Cabe señalar que **la manipulación de elementos en el DOM es un proceso LENTO.**



`document.getElementById("banner")`

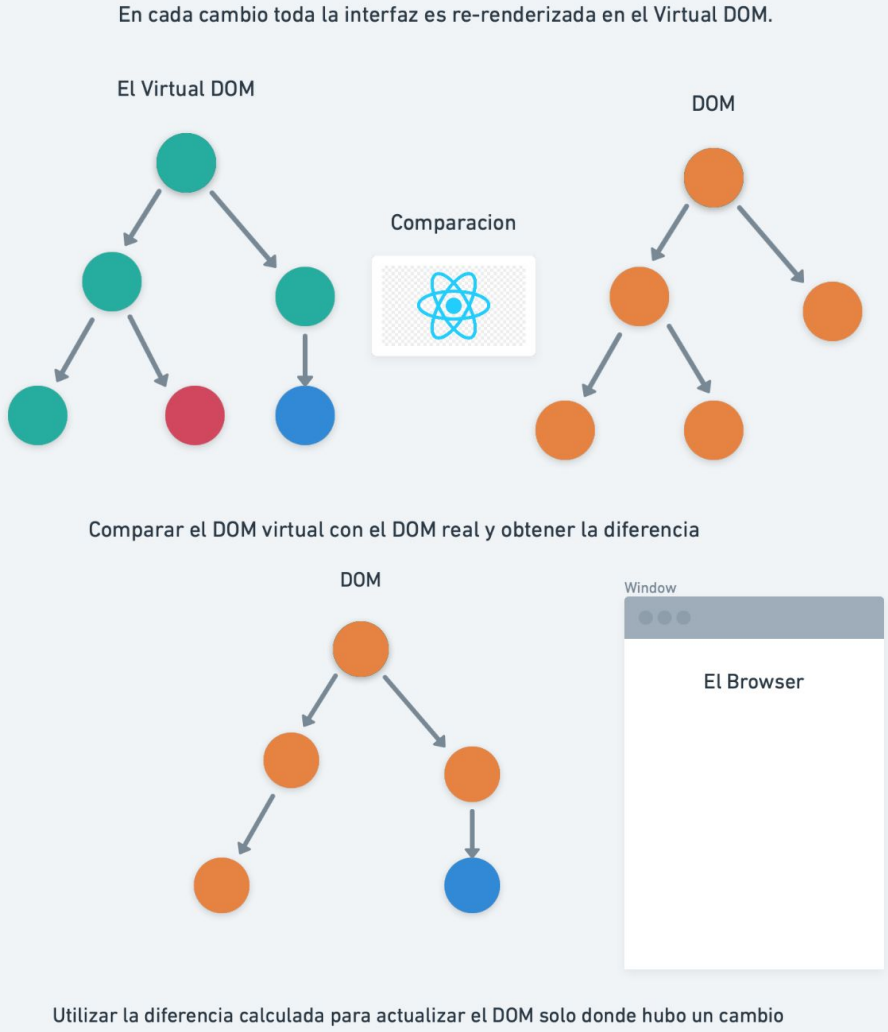


VIRTUAL DOM

React usa algo conocido como Virtual DOM para superar el lento problema de manipulación del DOM.

Un virtual DOM en realidad tiene las mismas propiedades que un objeto DOM real, pero a diferencia del DOM real, no afecta directamente nada en la pantalla.

Lo que significa que **el DOM virtual es solo un plano o una representación virtual del DOM real y no se ocupa de volver a renderizar la interfaz de usuario**.



Recursos Adicionales

What is React: A Visual Introduction for Beginners:

https://learnreact.design/posts/what-is-react

04

USANDO REACT

Creación de un proyecto de React

```
npm create vite@latest
```

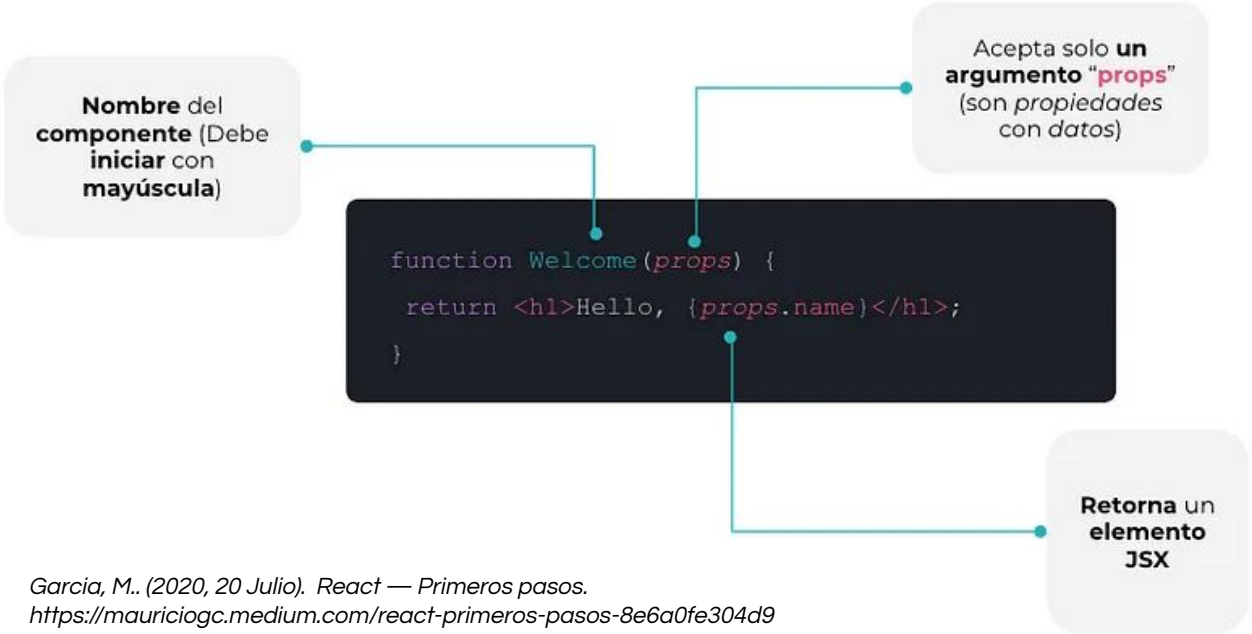
Crear directamente el proyecto de react

```
npm create vite@latest mi-proyecto -- --template react
```

Análisis del funcionamiento de React con Vite

- index.html
- src/main.js
- src/App.js

Estructura de Componentes en React



Garcia, M.. (2020, 20 Julio). React — Primeros pasos. https://mauriciogc.medium.com/react-primeros-pasos-8e6a0fe304d9

En React moderno los componentes se escriben por medio de funciones.

Estos pueden recibir datos llamados props y siempre deben de retornar una vista en formato JSX.

JSX

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.

```
const HeaderHero = () => {
  return (
    <header className="hero">
    <div className="textos-hero">
        <h1>Bienvenido a Website</h1>
        <p>Creamos el mejor sitio web para ti</p>
        <a href="#contacto">Contactame</a>
    </div>
    </header>
  )
}
export default HeaderHero
```

JSX = HTML + JS

Fragmentos de React

Un patrón común en React es que un componente devuelva múltiples elementos. Los Fragmentos te permiten agrupar una lista de hijos sin agregar nodos extra al DOM.

<React.Fragment>
...
</React.Fragment>

Hay una sintaxis nueva, más corta que puedes usar para declarar fragmentos. Parecen etiquetas vacías:

<>
...
</>

React (2023. <Fragment> (<>...</>)
https://es.react.dev/reference/react/Fragment

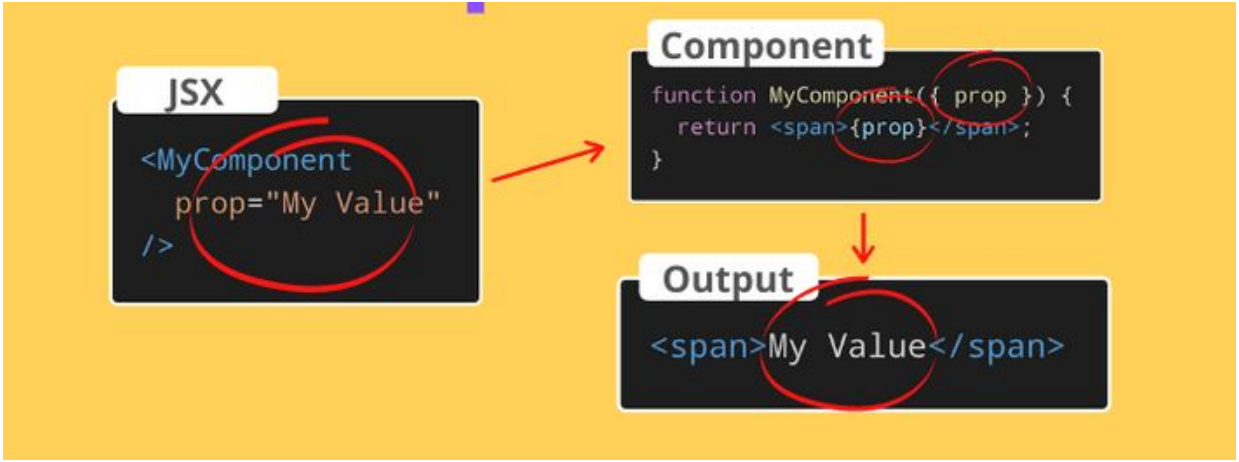
```
function myList() {
  return (
    <React.Fragment>
      <ChildA />
      <ChildB />
      <ChildC />
    </React.Fragment>
  );
}
```

Sintaxis completa

```
function myList() {
  return (
    <>
      <ChildA />
      <ChildB />
      <ChildC />
    </>
  );
}
```

Sintaxis corta

Props en React

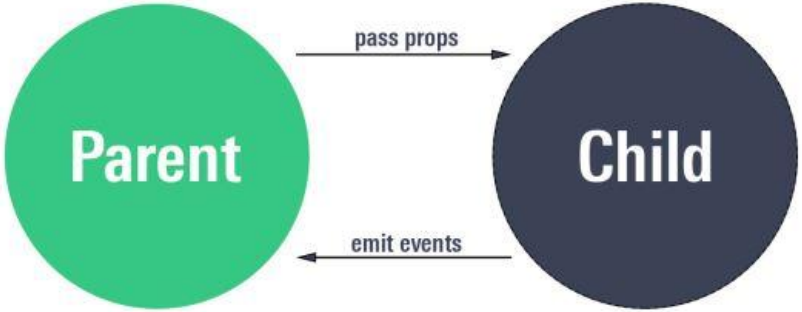


Los datos que se pasan a los componentes en React se llaman props.

Los props están disponibles en los parámetros del componente al que se le pasan. Los props siempre se incluyen como propiedades de un objeto.

CésarGuerra

Props en React



Los Props recibidos desde un componente Padre son sólo de lectura.

Destructurando Props

Una práctica común consiste en desestructurar las props al recibirlas en la función, para evitarnos tener que escribir **{props.nombre}**, así solo escribiriamos **{nombre}** para poder usarla.

Recursos Adicionales

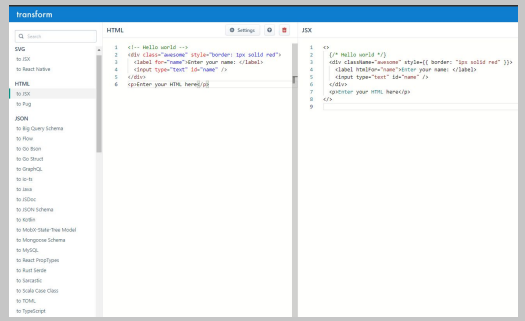
Documentación de React

https://es.react.dev/

DEMOSTRACIÓN

https://alexcgdesign.com/blog/crea-una-pagina-web-desde-cero-con-html-css/

Herramienta HTML -> JSX



Transform Tools:

https://transform.tools/html-to-jsx

CésarGuerra