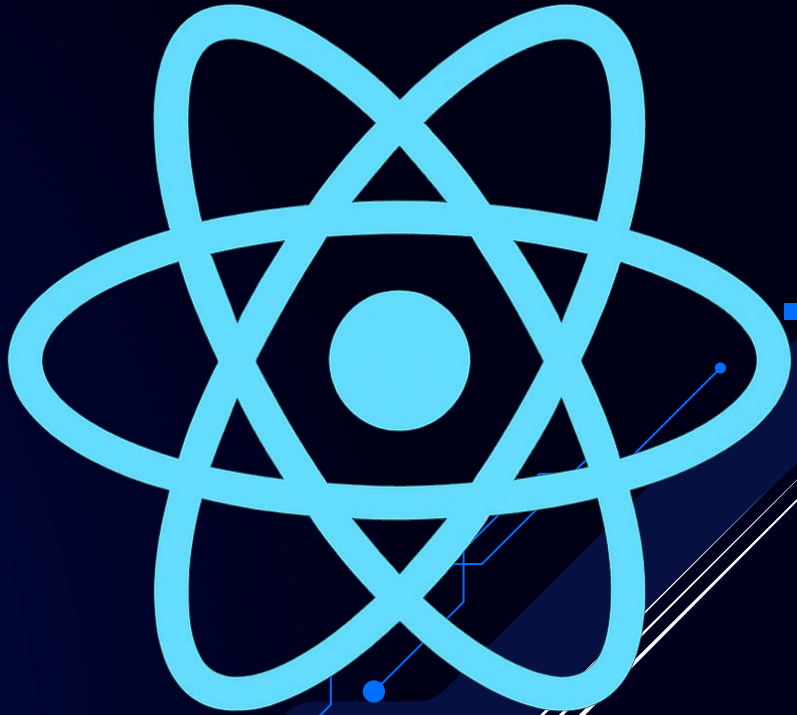
REACT



OUR TEAM

JAVIER
GÓMEZ
ALAYÓN
gomez.alayon.22@ull.edu.es

sara
darias
sánchez
sara.darias.19@ull.edu.es

Index

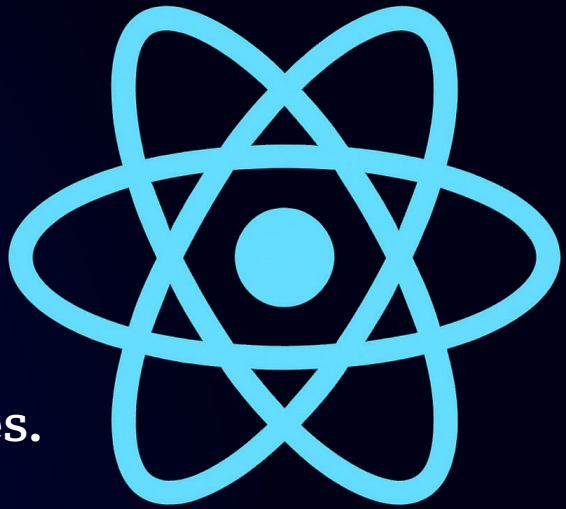
- What is React?
- Installation
- Characteristics
- Projects

01 WHAT IS REACT?

WHAT IS REACT?

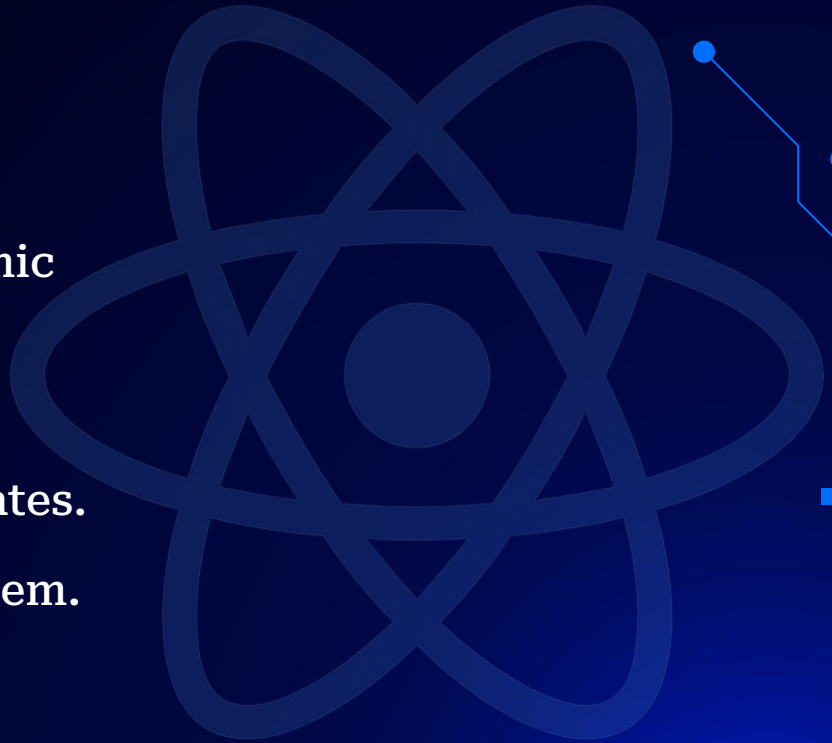
React Open-source JavaScript library created by Facebook in 2013.

- Builds dynamic, interactive UIs.
- Based on reusable components.
- Responds efficiently to data changes.



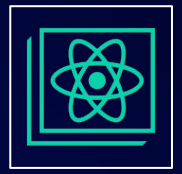
WHY REACT?

- Makes it easier to build dynamic user interfaces.
- Reusable components.
- Virtual DOM for efficient updates.
- Large community and ecosystem.
- High job market demand.
- Cross-platform compatibility.



02 INSTALLATION

Installation



Create react app

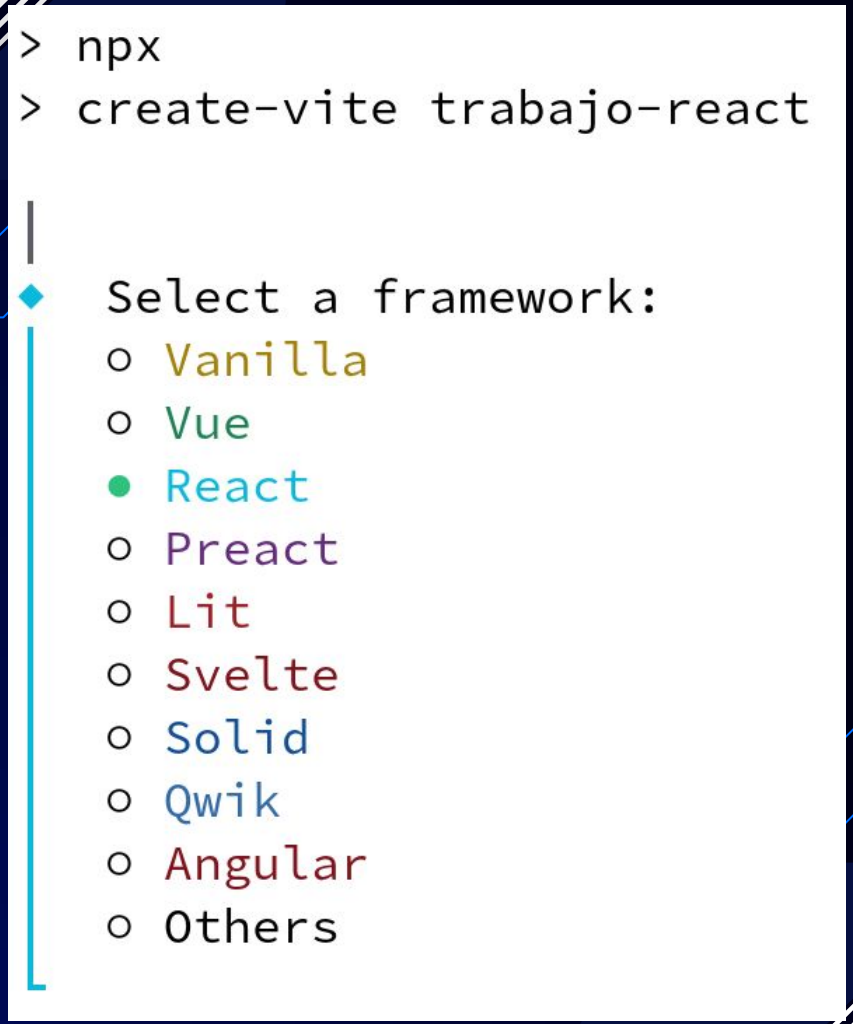
was an officially supported way to create single-page React applications.

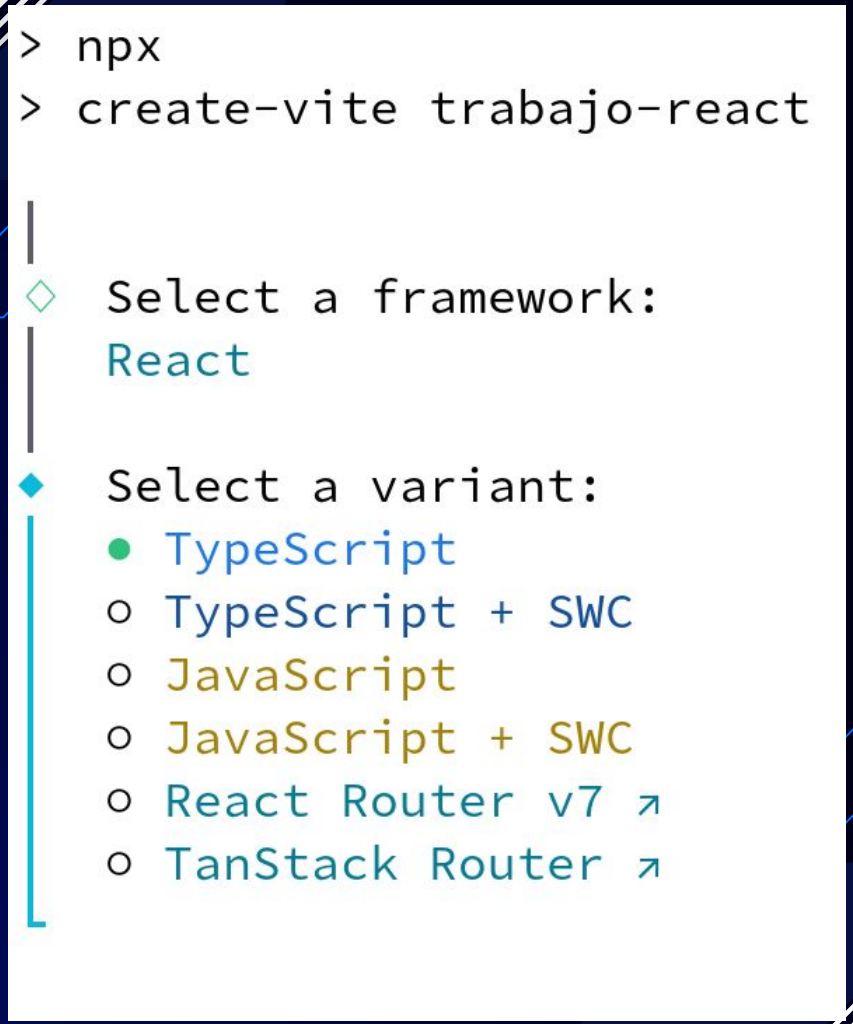
Vite

is a build tool that aims to provide a faster and leaner development.

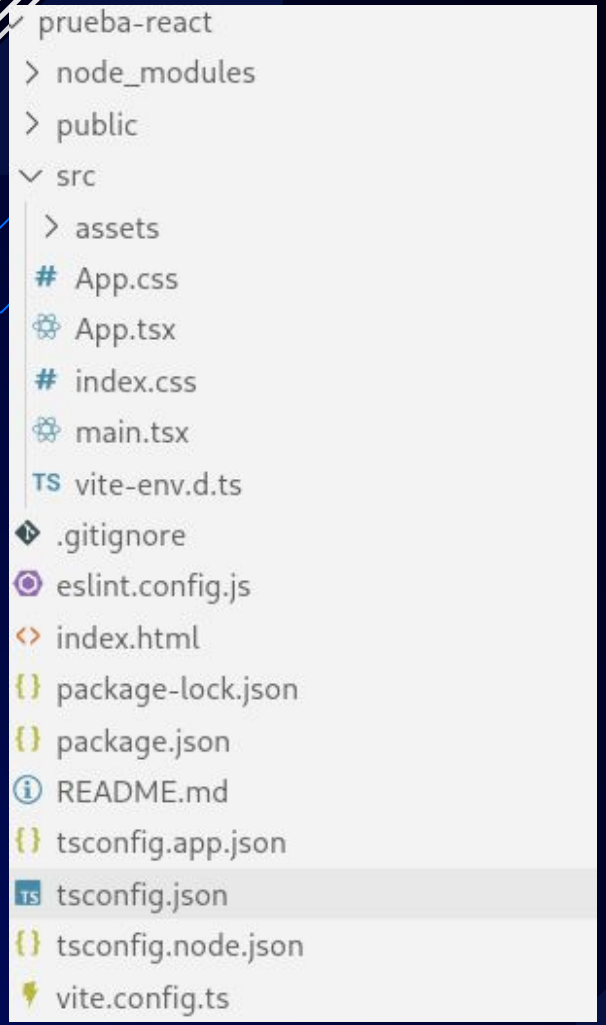
Installation with vite

```
npm create vite@latest [your project name]
```


> npx
> create-vite trabajo-react
Select a framework:
Vanilla
Vue
React
Preact
Lit
Svelte
Solid
Qwik
Angular
Others


> npx
> create-vite trabajo-react
◇ Select a framework:
React
◆ Select a variant:
● TypeScript
○ TypeScript + SWC
○ JavaScript
○ JavaScript + SWC
○ React Router v7 ↗
○ TanStack Router ↗

```
> npx
> create-vite trabajo-react

│
◇  Select a framework:
│  React
│
◇  Select a variant:
│  TypeScript
│
◇  Scaffolding project in /home/usuario/Trabajo/trabajo-react...
│
└  Done. Now run:

  cd trabajo-react
  npm install
  npm run dev
```


prueba-react
node_modules
public
src
assets
App.css
App.tsx
index.css
main.tsx
vite-env.d.ts
.gitignore
eslint.config.js
index.html
package-lock.json
package.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.node.json
vite.config.ts

03 Characteristics of React

VIRTUAL DOM

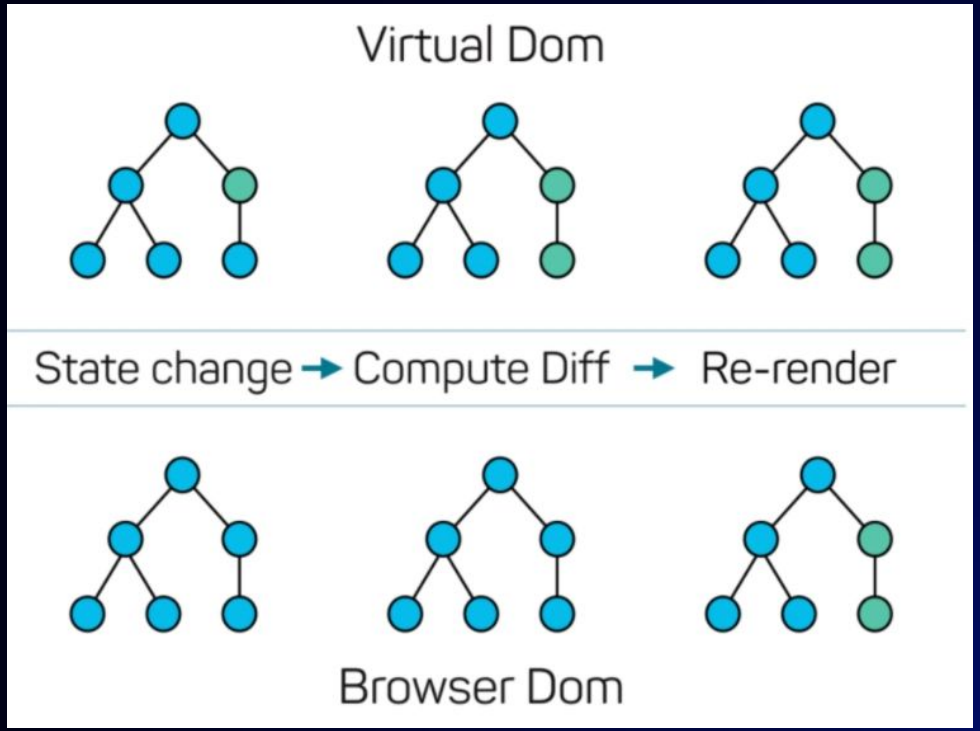
What is the virtual DOM?

- Is a programming concept where an virtual representation of a UI is kept in memory and synced with the “real” DOM by a library such as ReactDOM
- ReactDOM uses different phases to handle this purpose.

Why does React use the Virtual DOM?

- It has a major impact on the rendering of web applications, especially when the application has to render many times per minute.
- This improvement is completely transparent to the user

Phases of the Virtual DOM



JSX

What is JSX?

JSX (JavaScript XML) is a **syntax extension** for JavaScript

Allows developers to write HTML-like code directly within their JavaScript files; however, this extension requires adding an extra step to the program's compilation process.

It requires a compiler that supports this extension, such as Bubble or SWC.

Bubble

SWC

TSX

Example

```
return (
  <>
    <p>{name}</p>
    <p>Count: {count}</p>
    <button onClick={() => setCount(prev => prev + 1)}>
      Increase
    </button>
  </>
);
```

Enlace al playground de SWC

Enlace a los ejemplos de código

components

What are components?

Independent and reusable bits of code.

They return a HTML.

We must capitalize their names, using CamelCase.

Two types:

- Class components.
- Function components.

PROPS

What are props?

It stands for properties.

It’s a way to pass arguments to a component.

In tsx we use an interface.

They are passed to components via HTML attributes.

STATES

What is state in React?

- State is an object that holds dynamic data in a component.
- It allows the UI to react to changes (inputs, clicks, etc.).
- Managed using the useState hook in functional components.

State in React

State can be initialized with the useState hook.

This example initializes a state variable count with the value 0.

```
function Example(): void {
  const [count, setCount] = useState<number>(0);
}
```

Multiple states

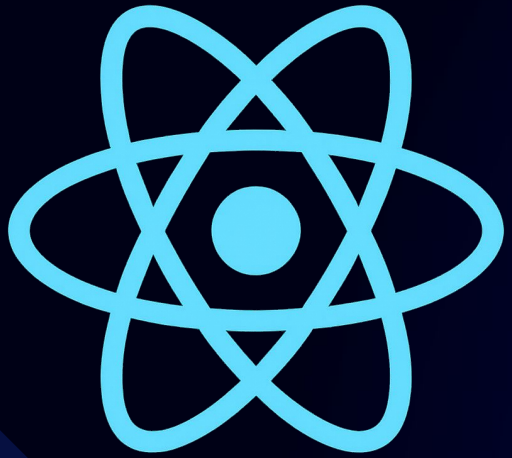
```tsx
const [name] = useState<string>("Pedro");
const [count, setCount] = useState<number>(0);
return (
  <>
    <p>{name}</p>
    <p>Count: {count}</p>
    <button onClick={() => setCount(prev => prev + 1)}>
      Increase
    </button>
  </>
);
```

- Separate state variables for independent pieces of state.
- State variables can be of any type.
- Multiple useState calls are allowed in a single component.

Best practices with useState

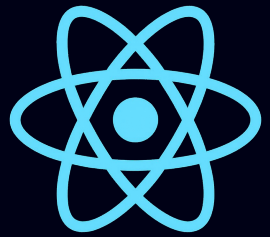
```jsx
import { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(prev => prev + 1)}>
        Increment
      </button>
    </div>
  );
}

export default Counter;
```

- Use functional updates when the new state is based on the previous state.
- Keep state as small and simple as possible.
- Avoid derived state.



VS

REACT VS OTHERS



- Flexible library.
- TSX + JavaScript.
- Moderate learning curve.
- Very popular (Meta).

- Progressive framework.
- HTML + JS.
- Easy to learn.
- Active community.

- Full-fledged framework.
- TypeScript required.
- Steep learning curve.
- Great for large projects.

04 PROJECTS

Bibliography

React official documentation

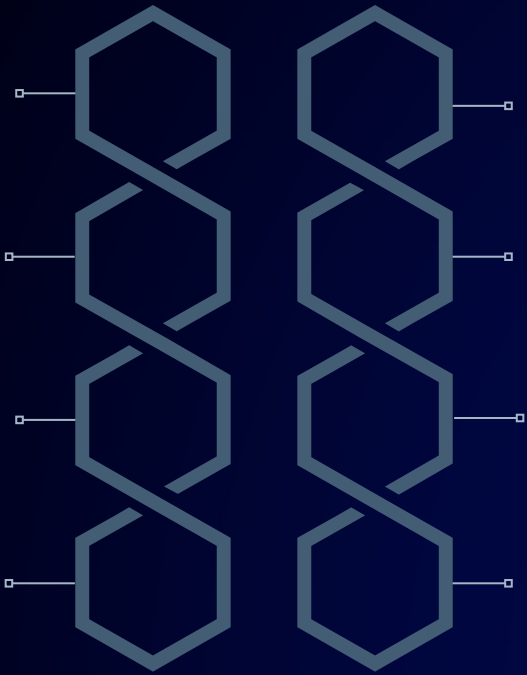
React official installation

React course

React video tutorial

Explanation of the virtual DOM

SWC playground

Main concepts of React

Props and state Explained