React Introduction

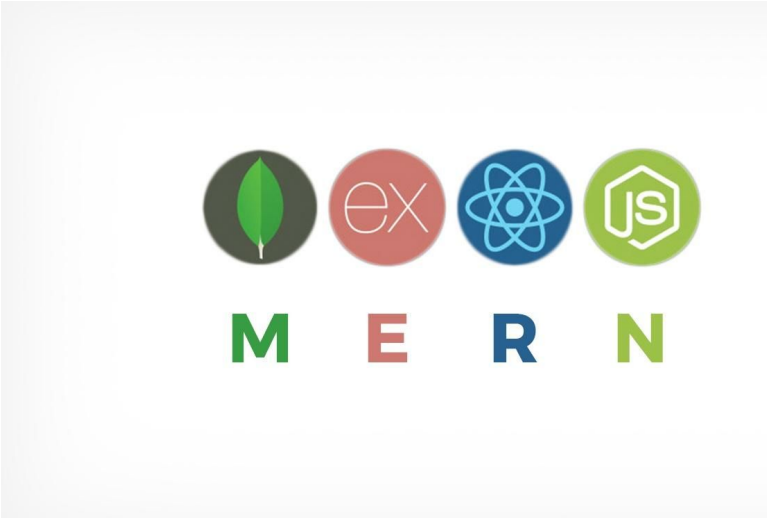
ITJ BOOTCAMP

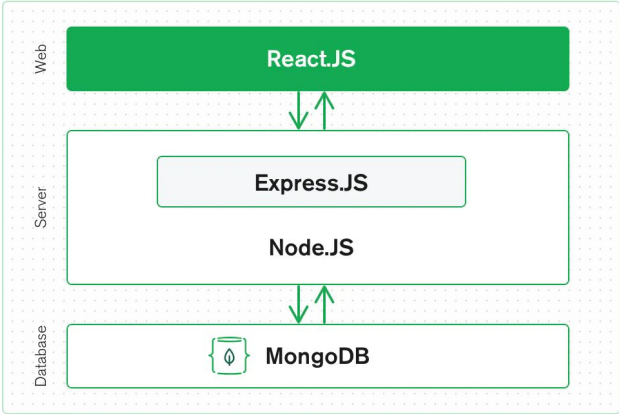
HELLO!

I am Eduardo Flores.

You can find me at eduardo-flores846 on Linkedin

1.
MERN stack


ex
JS
M E R N


Web
React.JS
Server
Express.JS
Node.JS
Database
MongoDB

2.
React

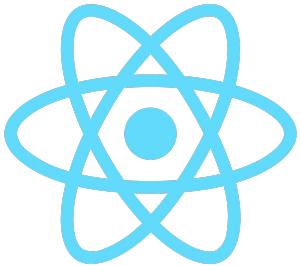
What is React

“A JavaScript library for building user interfaces”

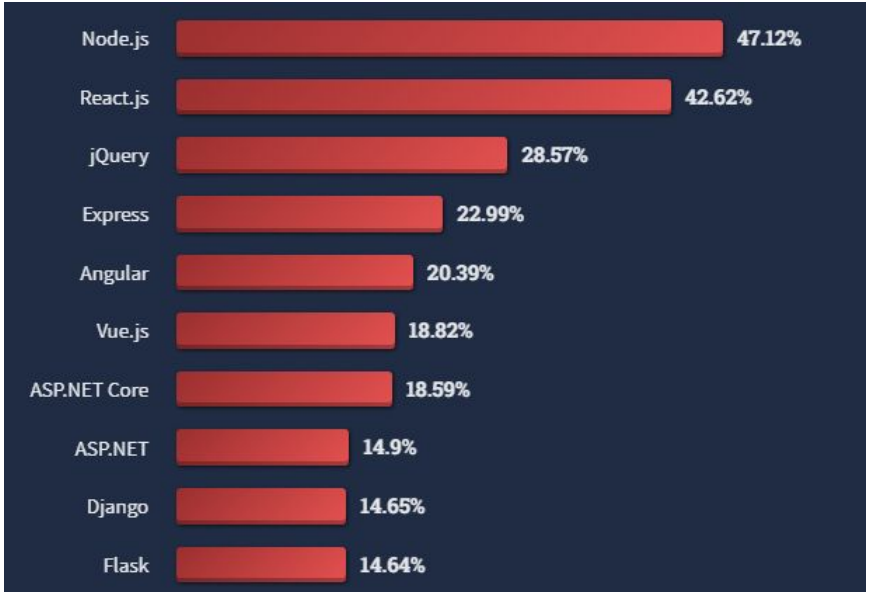
Declarative

Component Based

Lean Once, Write Anywhere



As popular as something can be

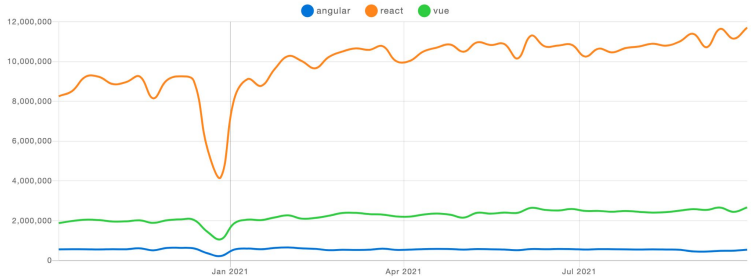


Web frameworks and technologies

Node.js and React.js are the two most common web technologies used by Professional Developers and those learning to code

It's easy to get help if you are stuck

Lots of opportunities



3.
Time to learn by doing

Before we start

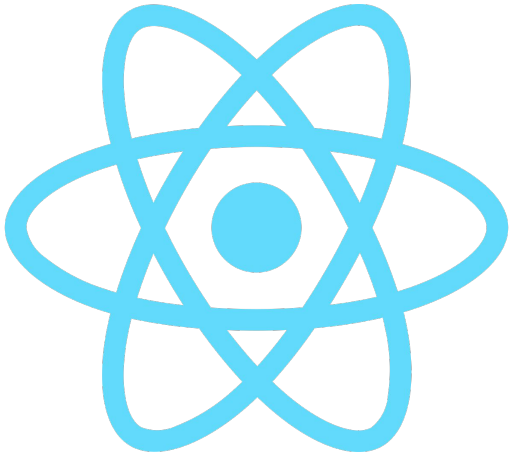
How to create a React App

- Start any IDE or Code Editor (VSCode Recommended.)
- Open the console (Control + J in VSCode)
- Type "npx create-react-app react-intro" and wait
- Write "cd react-intro" in the console
- Then write "npm start"

4.
React
Introduction

React topics

- React JSX
- React Elements
- React Elements attributes
- React returning elements
- React Components
- React Props
- React Conditionals



JSX

Mix of JS with an HTML-like syntax. JSX is a syntax extension to JavaScript. React embraces that rendering logic is inherently coupled with user Interface logic. You can put any valid JavaScript expression within curly braces.

example 1

```
const number = 2;
return <div>Hello World! {number === 2 ? "TWO" : "ONE"} </div>;
```

example 2

```
const formatName = (user) => {
  return user.firstName + " " + user.lastName;
};

const App = () => {
  const user = {
    firstName: "Pedro",
    lastName: "Guzman",
  };

  return <div>Hello World! {formatName(user)} </div>;
};
```

React element

The elements are the building blocks of React apps. An element describes what you want to see on the screen.

```
return <div>Hello World!</div>;
return <h1>Hello World!</h1>;
return <button>Hello World!</button>;
```

React elements are written just like regular HTML elements. You can write any valid HTML element in React

React elements styles and styles

Element attributes

JSX requires a different syntax for its attributes. Uses a camel case naming convention, example :“className”.

```
return <input readOnly={false}/>

return <div className="helloWorld">Hello World!
{formatName(user)} </div>
```

Elements styles

To apply inline styles, instead of using double quotes (“”), we use two sets of curly braces. Inline styles are not written as plain strings, but as properties on objects.

```
<div
className="helloWorld"
style={{   fontSize:   "50px",   color:   "red",
marginTop: "100px" }}>
      Hello World! {formatName(user)}
</div>
```

React returning elements

A common pattern in React is for a component to return multiple elements. When need to wrap all the elements, because is recommended to return a single element per component.

```
<div>
   <img
     src="https://itjuana.com/wp-content/uploads/ITJ_tm-logo.png"
     alt="itjuana"
     style={{ with: "100px", height: "100px" }}/>
      <h1 className="helloWorld" style={{ fontSize: "50px", color: "red" }}>
        Hello World! {formatName(user)}
      </h1>
      <h2>fron Full Stack bootcamp</h2>
</div>
```

Another way to wrap our elements is to use a react fragment “<></>”. This helps not only to warp our elements, but it does no create a new node in the DOM. If we add the fragment in the previous example, we can see the difference

<></>

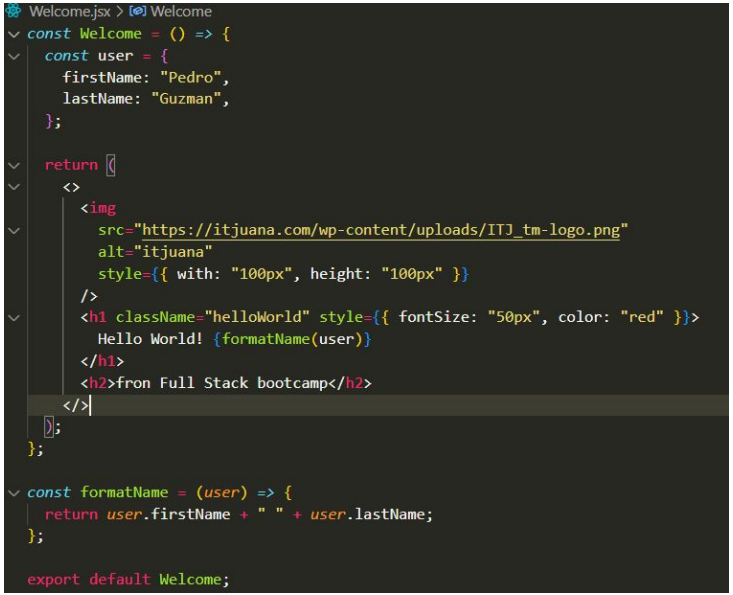
```
▼ <div id="root"> == $0
    <img src="https://itjuana.com/wp-content/uploads/ITJ_tm-logo.png" alt="itj
    uana" style="height: 100px;">
  ▶ <h1 class="helloWorld" style="font-size: 50px; color: red;">…</h1>
    <h2>fron Full Stack bootcamp</h2>
  </div>
```

** **

```
▼ <div id="root">
  ▼ <div>
      <img src="https://itjuana.com/wp-content/uploads/Grupo-211.png" alt="itj
      uana" style="height: 100px;">
      <h1 class="helloWorld" style="font-size: 50px; color: red;">Welcome to
      the firt session!-Edward Flores</h1>
    </div>
  </div>
```

React components

Components let you split the UI into independent reusable pieces, and think about each piece in isolation.



Note:

- Component names must start with a capital letter (that is, MyComponent, instead of myComponent)
- Components, unlike JavaScript functions, must return JSX.

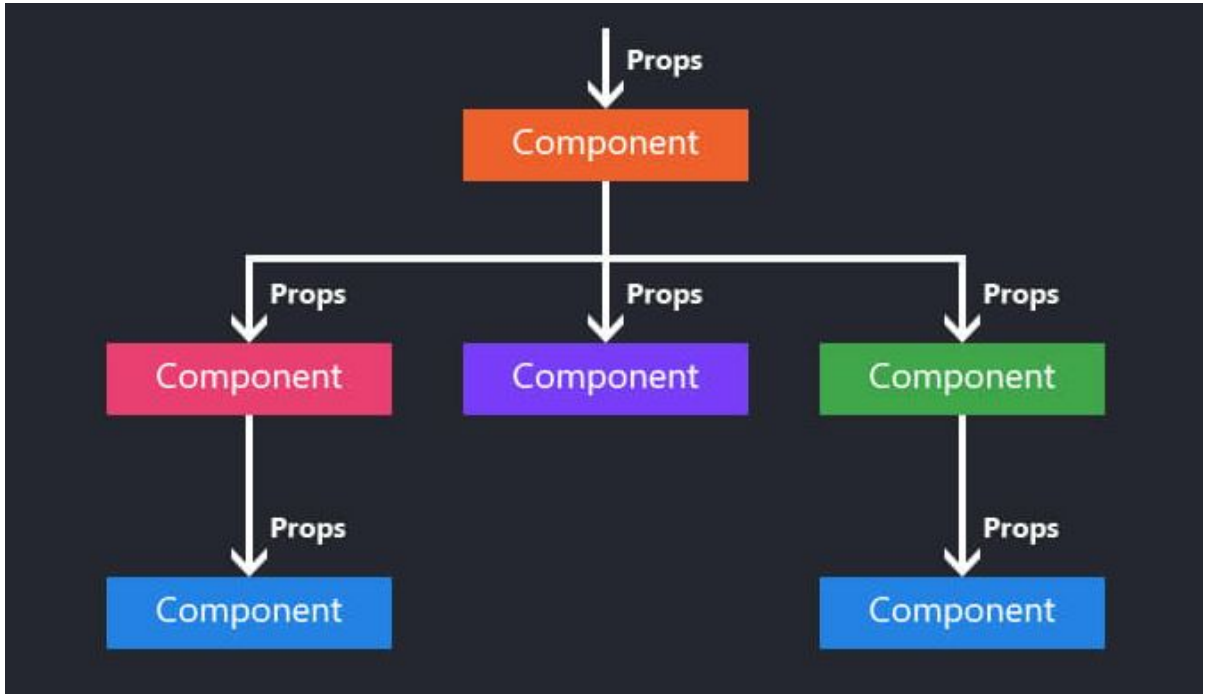
Composing Components

Components can refer to other components in their output. This lets us use the same component abstraction for any level of detail. A button, a form, a dialog, a screen: in React apps, all those are commonly expressed as components.

```
const FullStack = () => {
  return <div>Welcome to the Full Stack Developer 2.0 bootcamp</div>;
};

const TestAutomation = () => {
  return <div>Welcome to the Test Automation 2.0 bootcamp</div>;
};

const App = () => {
  return (
    <>
      <FullStack />
      <TestAutomation />
    </>
  );
};
```

```
import FullStack from "./FullStack";
import TestAutomation from "./TestAutomation";

const App = () => {
  return (
    <>
      <FullStack />
      <TestAutomation />
    </>
  );
};
```

How components communicate



React props

Props work like parameters in functions.. Props are passed from the parent component to a child component. We can past as many props as we want

```
import Welcome from "./Welcome";

const Bootcamp = (props) => {
  return <h2>{props.name}</h2>;
};

const App = () => {
  const bootcampName = "Test Bootcamp 2.0";

  return (
    <>
      <Welcome />
      <Bootcamp name={bootcampName} />
    </>
  );
};
export default App;
```

React conditionals

React components and elements can be conditionally displayed. We can write a conditional within a return statement, however, we must use a conditional that resolves to a value.

```
return (
  <>
    <Welcome user={user} image={image} message={message} />
    {isFullStackBootcamp ? (
      <Bootcamp name={fullStack} />
    ) : (
      <Bootcamp name={TestAutomation} />
    )}
    <Bootcamp name={isFullStackBootcamp ? fullStack : TestAutomation} />
  </>
);
```

5.
Let's build the course project

BOOTCAMP

THANKS!

Resources

- React – Una biblioteca de JavaScript para construir interfaces de usuario (reactjs.org)\
- Stack Overflow Developer Survey 2022
- Newest 'reactjs' Questions - Stack Overflow
- MUI: The React component library you always wanted
- @mui/icons-material - npm (npmjs.com)
- git-cheat-sheet-education (github.com)