# Installing Node JS

# Video 7.- Creating React Project

The recommended method for generating a project is:

>> npx create-react-app my-app

Looks like you don’t need to install anything else

## Creating the react app

>> npx create-react-app jsx

# Video 8.- Why create react app?

# Video 9.- Exploring a create-react-app project

Exploring the folder of the project jsx.

In folder src , generally we delete almost all those files to create a new project from scratch.

# Video 10.- Starting and stopping react app

From the project folder jsx we execute

>> npm start

# Video 11.- JavaScript Module Systems

We deleted all the files in src and create a new index.js

Then we created this code :

# Video 13 .- Displaying content with functional components

A react component is a function or a class.

That will produce HTML to show to the user

handles feedback from the user. We use event listener

index.js example:

This will show Hi there in the HTML file in the root id. This HTML file is in the folder public.

Text

Description automatically generated

# SECTION 2: Building content with JSX:

# Video 16.- What is JSX

Is not html.

The purpose of JSX is to allow us to write more simpler code that is going to be transformed by BABEL.

# Video 17.- Converting HTML to jsx

We saw that JSX is not equivalent to HTML

# Video 18.- inline Styling with jsx

Este html : <div style=”background-color: red”, ></div>

Seria equivalente a este jsx: <div style={{backgroundColor: ‘red’}}, ></div>

# Video 20.- Class vs className

Instead of class we use className in jsx

# Video 21.- Referencing JS variables in JSX

Text

Description automatically generated

You can also call functions within JSX

Text

Description automatically generated

# Video 22.- Values JSX can’t show

JS objects can not be shown in jsx.

But you can show one of the attributes of the objest like object.key

# Video 23.- Finding Forbidden property Names

To find the error you can find it in the console log in the warning message

# Video 24.- Exercise introduction

# SECTION 3:COMMUNICATION WITH PROPS

# Video 26.- Three tenets of components

## Component Nesting:

* A component can be shown inside of another

## Component Reusability:

* We want to make components that can be easily reused through out application

## Component Configuration:

* We should be able to configure a component when it is created

# Video 27.- Application overview

We created the app components to create a new application and see how they work

Video 29.- Getting some free styling

<https://semantic-ui.com/>

From that web page we can get some styling

We can install semantic by

Look for semantic ui cdn

<https://cdnjs.com/libraries/semantic-ui>

The cdn link is going to be copied to the HTML index file we have in the public directory.

Este fu eel que usamos

<script src="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js" integrity="sha512-dqw6X88iGgZlTsONxZK9ePmJEFrmHwpuMrsUChjAw1mRUhUITE5QU9pkcSox+ynfLhL15Sv2al5A0LVyDCmtUw==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

# Video 30.- Naive component approach

# Video 31.- Specifying images in JSX (Faker js)

In this section we installed faker js

Is an open source library

Is a library to create fake data including fake images

To install this library :

>> npm install –save faker

# 