**Intalacion de React js**

**Crear una nueva aplicación React**

Para obtener la mejor experiencia de usuario y desarrollador use una cadena de herramientas integrada.

Esta página describe algunas de las cadenas de herramientas de React más populares, las cuales ayudan con tareas como:

* Escalar a múltiples archivos y componentes.
* Usar bibliotecas de terceros desde npm.
* Detección temprana de errores comunes.
* Edición en vivo de CSS y JS en desarrollo.
* Optimización de la salida para producción.

Puede que no necesites una cadena de herramientas

Si no experimentas los problemas descritos arriba, o aún no te sientes cómodo usando herramientas de Javascript, considera añadir React como una etiqueta <script> en una página HTML, opcionalmente con JSX.

npm install npm@latest -g

Create React App

Create React App es un ambiente cómodo para aprender React, y es la mejor manera de comenzar a construir una nueva aplicación de página única usando React.

**Create React App**

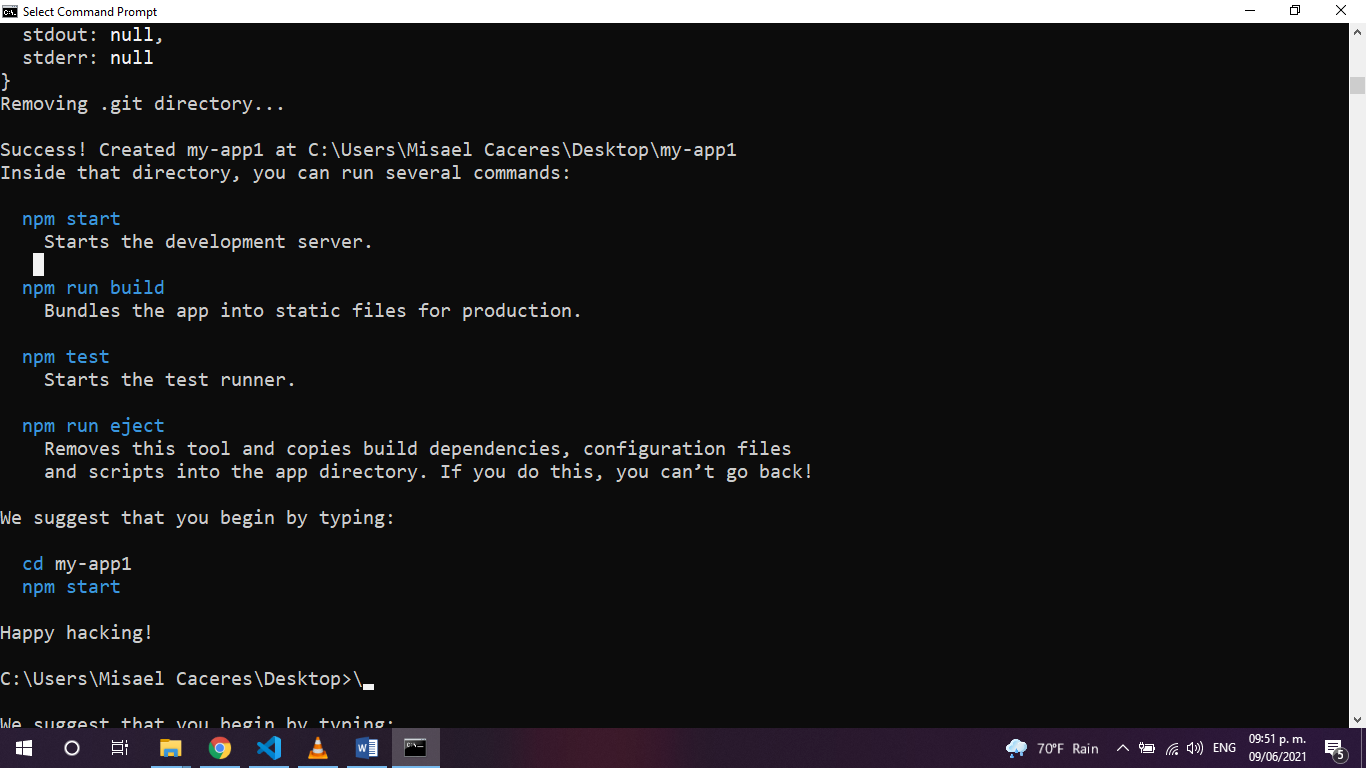
configura tu ambiente de desarrollo de forma que puedas usar las últimas características de Javascript, brindando una buena experiencia de desarrollo, y optimizando tu aplicación para producción. Necesitarás tener [Node >= 10.16 y npm >= 5.6](https://nodejs.org/en/" \t "_blank) instalados en tu máquina. Para crear un proyecto ejecuta:

npx create-react-app my-app

cd my-app

npm start

**Nota**

En la primera línea npx no es un error de escritura: Es una [herramienta de ejecución de paquetes que viene con npm 5.2+](https://medium.com/@maybekatz/introducing-npx-an-npm-package-runner-55f7d4bd282b).

**Npm start**

Inicio del servidor locahost::3030

**Componentes App.js**

import React from 'react';

import './App.css';

function Helloworld(){

  return (

  <div>Helloworld</div>

  )

}

//Componente en Funcion Fecha

//const App = () => <div>This is my componert: <Helloworld/> </div>

//Componente en clase heredando de Componente React

/\*class App extends React.Component{

  render(){

    return <div>This is my componert: <Helloworld/> </div>

  }

}\*/

function App() {

  return (

    <div>this is my componert is: <Helloworld/></div>

  );

}

export default App;

**Props**

**Propiedades o propetis**

function Helloworld(props){

  console.log(props)

  return (

  <div id= "hello">{props.mytext}</div>

  )

}

function App() {

  return (

    <div>

      this is my componert is:

      <Helloworld mytext="Hola Missael" />

      <Helloworld mytext="Hola Maileny"/>

      <Helloworld mytext="Son dos Cara e locos"/>

     </div>

  );

}

export default App;

**Props con sustitle para componentes**

function Helloworld(props){

  console.log(props)

  return (

  <div id= "hello">

    <h3>{props.sustitle}</h3>

    {props.mytext}

  </div>

  )

}

function App() {

  return (

    <div>

      this is my componert is:

      <Helloworld mytext="Hola Missael" sustitle ="Componente 1" />

      <Helloworld mytext="Hola Maileny" sustitle="Componente 2"/>

      <Helloworld mytext="Son dos Cara e locos" sustitle="Componente 3"/>

     </div>

  );

}

export default App;

**State o Estado**

class Helloworld extends React.Component{

    state = {

      show :true

    }

    toggleShow = () => {

      this.setState({show: !this.state.show})

    }

  render (){

    if (this.state.show) {

      return (

        <div id= "hello">

          <h3>{this.props.sustitle}</h3>

          {this.props.mytext}

          <button onClick={this.toggleShow.bind(this)}>Toggle Show</button>

        </div>

        )

    } else  {

       return  <h1>

         There are not elementos

         <button onClick={this.toggleShow}>Togle Regreso</button>

      </h1>

    }

}

}