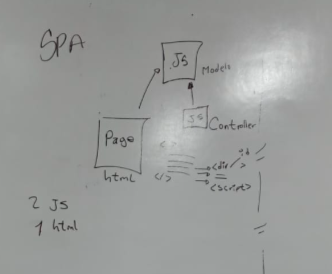
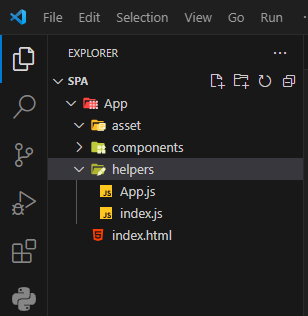
**Profesor: Jeffer David Mendez**

**Celular:3233131795**

**SPA (Aplicación o Programa de una Pagina)**





Para que salga las carpetas de colores, descargo la aplicación: Material Icon Theme

Forma tradicional: devuelve a la función anterior

Tipo Flecha: Lo devuelve hasta el principal

**Framework  marco de trabajo  PIZZA**

Librerías que nos permiten interfases para la creación de un proyecto

VEU: Mas facil

ANGULAR: Mas antiguo

REAET: Mas común

Compilar es que verificar: NODE NPM

Examen semana 4

SPA, DOM, REACT

**REACT:**

**Es una librería que nos permite darle una línea al proyecto** (seguridad, inter-operatividad)

NODE JS Programa

NPM Gestor de paquetes (Modo ejecución – NPX)

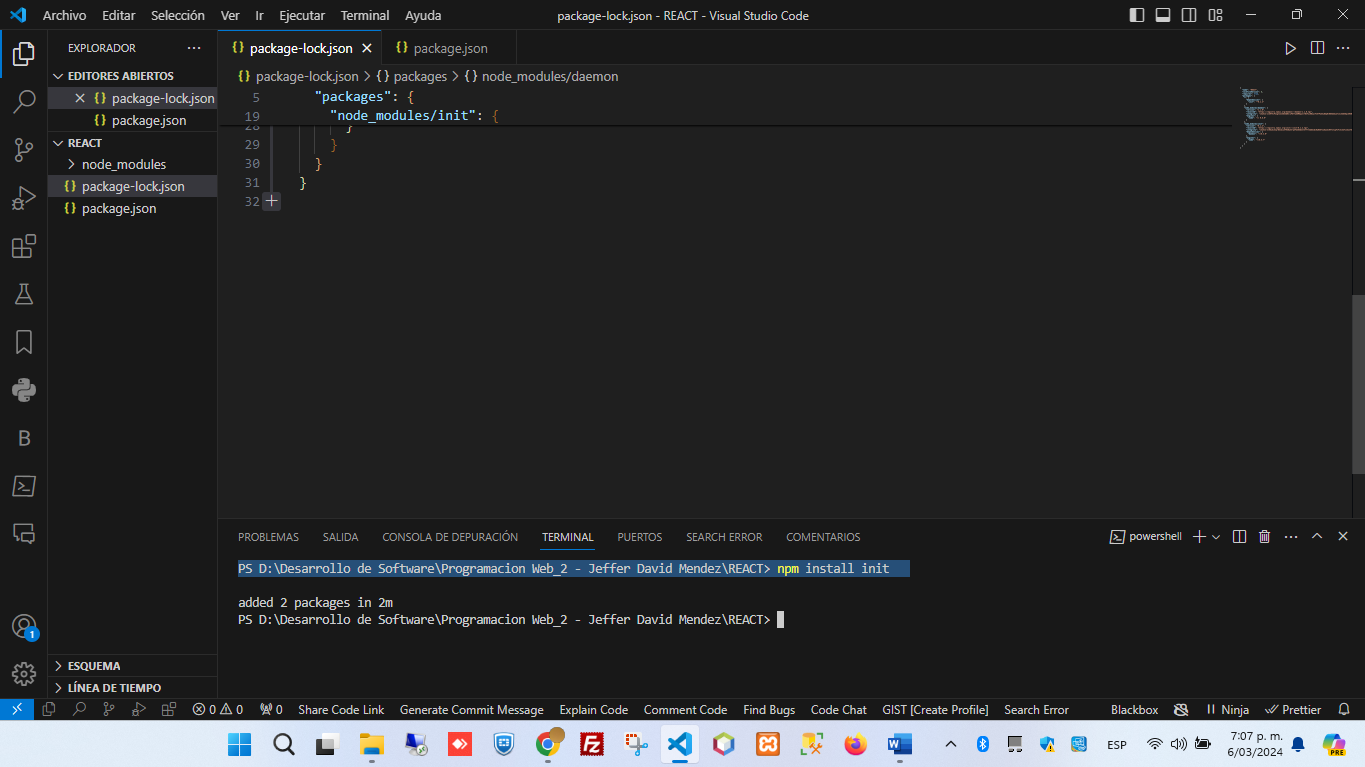
Editor de Codigo

No se pueden utilizar dos Framework de la misma capa

Utilizar el puerto 3000

Name: No permite crear los nombres que yo quiera (React, app, ñ, \_ , tiltes, espacios)

Instalo la npm



**Instalar libreria**

PS D:\Desarrollo de Software\Programacion Web\_2 - Jeffer David Mendez\REACT> npm install express

added 64 packages, and audited 67 packages in 23s

**Errores al instalar React**

1. **Error en la ruta del proyecto – instalar npm en la ruta de react**

npm install créate-react-app

1. **Los permisos**

npm intall react-script

1. **Name**

React, app, ñ, \_ , tiltes, espacios

**Como Crear un React.**

En este link puedes ver cómo crear un REACT

<https://www.youtube.com/watch?v=wGxDfSWC4Ww&t=842s>

Para abrir un proyecto ya creado se deben de seguir los pasos de color

1. **Crear una carpeta con el nombre del proyecto**
2. **En visual desde la terminal y con el npm creo el creat-react-app**

npm install create-react-app (2min)

1. **Instalar React e mi pc (Visual) con la siguiente estructura:**

|  |  |  |
| --- | --- | --- |
| Gestor de paquete | React | Nombre del proyecto |
| npx | create-react-app | proyecto-react |

**Ejemplo:** ojo este es el crea el react y se tarda 20 minutos

npx create-react-app proyecto-react

npx create-react-app momento-2

1. Posicionarnos sobre el directorio o carpeta

We suggest that you begin by typing / Le sugerimos que comience escribiendo

**Ejemplo:** cd proyecto-react

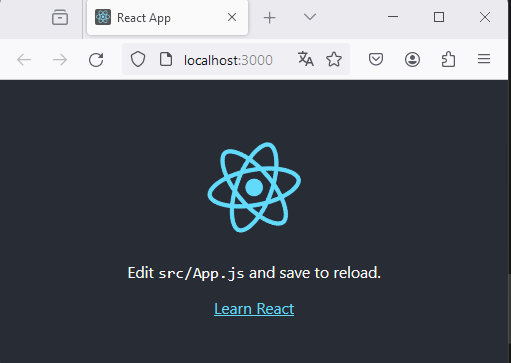
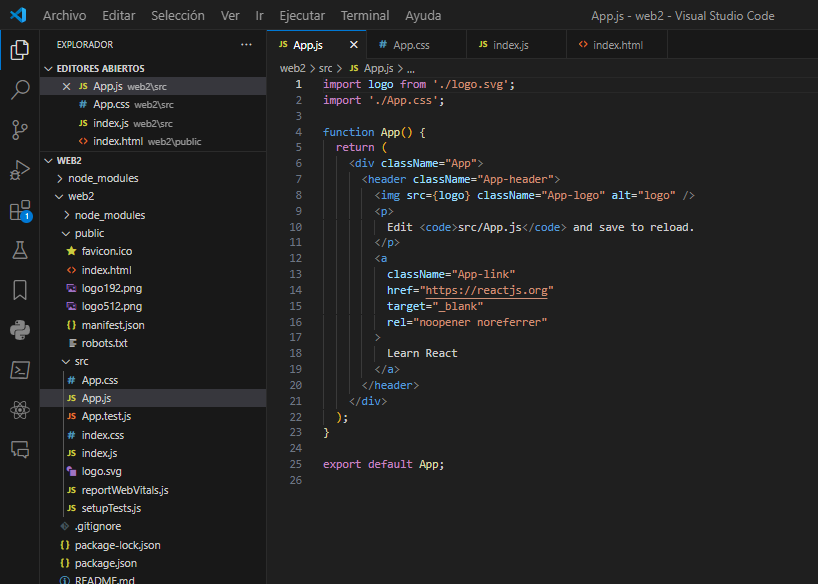
cd momento-2

npm start

En este punto se debe abrir la página web:

Local: <http://localhost:3000>

On Your Network: <http://10.111.241.189:3000>

1. **Instalar Libreria dom**

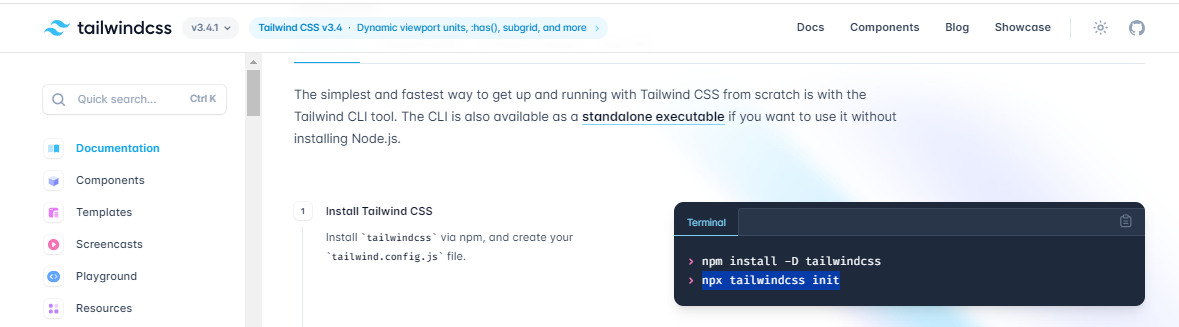
npm install proyecto-react-dom

npm install momento-2-dom

El archivo que edita esta en APP.JS

<https://tailwindcss.com/docs/installation> para los estilos...

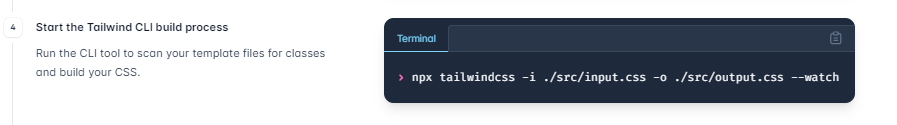
React boostrat

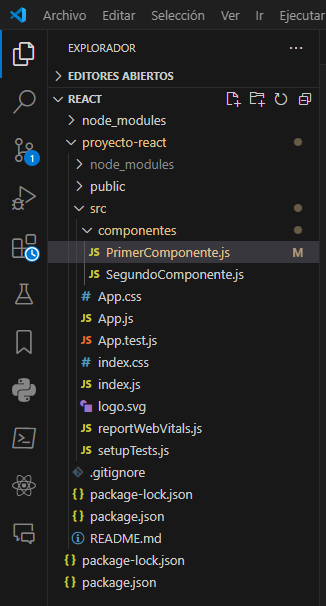
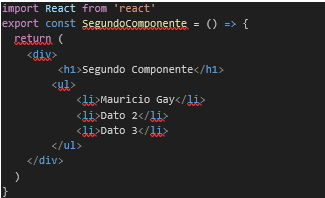


Ahora lo vamos a configurar

content: ["./src/\*\*/\*.{html,js, jsx, ts, tsx, css}"],

  theme: {



import React, {useState} from 'react';

export const PrimerComponente = () => {

    //let nombre  = "Julian Alvarez";

    let web     = "julian.alvarez@gmail.com";

    const [nombre,  setNombre]   = useState("Julian");

    let cursos  = [

                    "JavaScript",

                    "HTML",

                    "REACT",

                    "CSS"

    ];

    const cambiarNombre = (nuevoNombre) => {

        setNombre ( nuevoNombre );

    }

  return (

    <div>

        <h1>Mi Primer Componente</h1>

        <p>Este es el texto de Mi Primer Componente</p>

        <p>Mi nombre es: {nombre}</p>

        <p>Mi correo electronico es: {web}</p>

        {/\* <input type='text' onChange= { e => cambiarNombre(e.target.value)} placeholder="Cambiar el nombre">/

        </input> \*/}

        <button onClick={ e => cambiarNombre("JULIAN ALVAREZ WEB")} >

            Cambiar Nombre

        </button>

        <h2>Cursos:</h2>

        <ul>

            {

                cursos.map((curso, indice) =>  {

                return (<li key={indice} >

                    {curso}

                    </li>);

                })

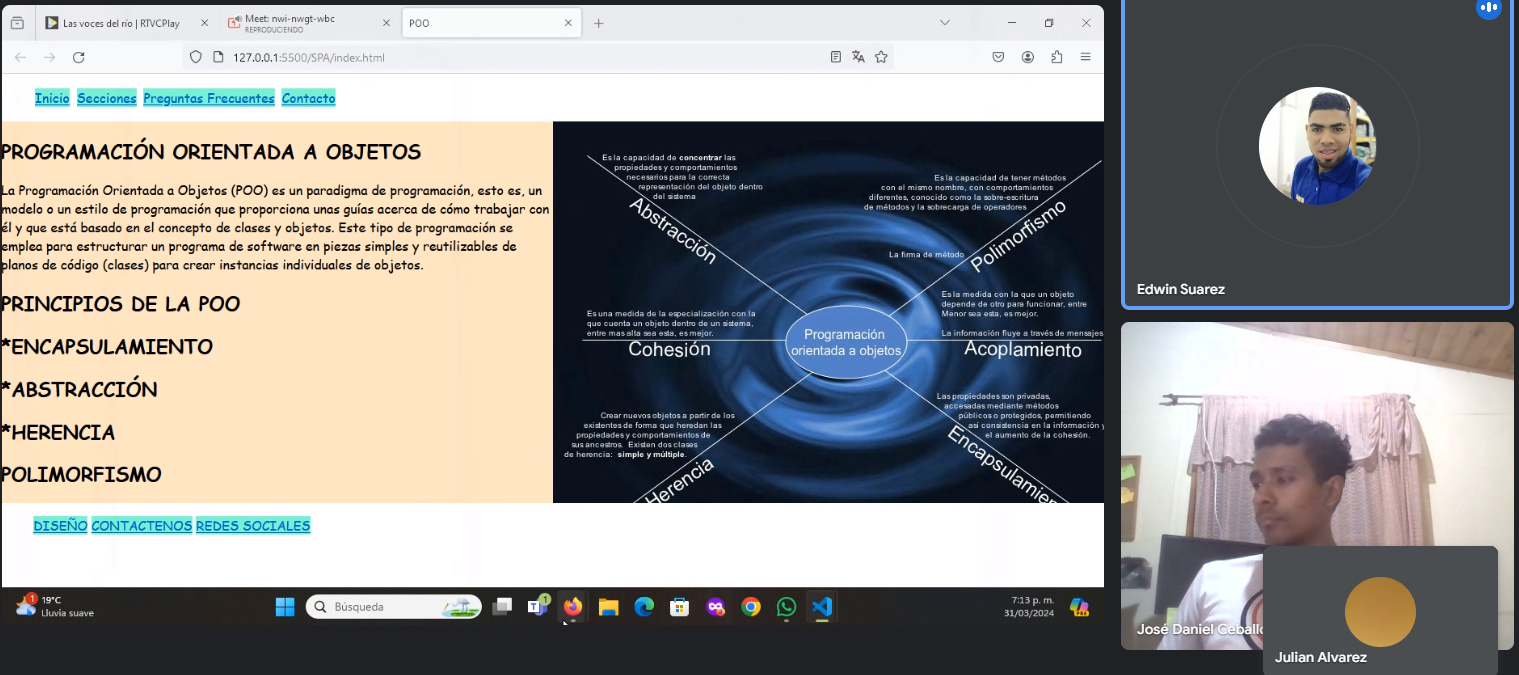
            }

        </ul>

    </div>

  )

}



indes.js:

app.js:

Home, principal

El layout es el que tiene las rutas donde vamos a conectar con las carpetas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Miercoles 17 abril 2024**

Jsx: Js + xml

TX códigos unidireccionales

Facilita el contacto con HTML, dentro de JS

Las etiquetes como son unilaterales se deben de cerrar de la siguiente manera

<img src="https://editorial.uefa.com/resources/0246-0e98e0aed53b-074f228dc0a2-1000/ucl\_branding.jpeg" alt="bayern münchen" />

Prop Props

Componente padre, para mostrarlo, Herencia

State - Estados

Cambios a registros de forma temporal, en la base de datos no hay cambios

Investigar las herramientas que contienen el Props las librerías de proptype

**Miércoles 27 de abril**

Props

Function Contac (PROPS){

Props.(name)

}

Solo me trae los que esta con Props

Tiene sub librerías.

Proptypes……..

**Miercoles 08/05/2024**

La evaluacion va ser de State y props

State.

[0,1][default, set]

Librería 🡪

usestate

useEffect

useContext

API

Procesos

Automatizacion 🡪 me garantiza que funciona la app que voy a trabajar

Integracion de datos 🡪 la formulación ya esta deficniida

Funcionalidad 🡪

Tipos de app

Sincronicas:

Son las que funcionan en relación a un orden secuencial, confirme a las peticiones que vamos acioendo

Asincrónicas:

Estas ya tienen un protocolo gerarquico

Protocolo Rest: 🡪 representación de preferencia

1 cliente - servidor

Cliente Servidor

Web 🡪 Rest 🡪 API 🡪 Base de Datos

URL

2 Stateless:

https://

state

3 cache

4 Interface Uniforme

5 sistemas de capas

6 Codigo en demanda

Request

API 🡪WEB entre la interacion entre la apli y la Web, se llama toda peticion tiene una respuesta

Response

URL 🡪 URI

[http ://web/](http://web/) dir/local/ ?awo

Scheme authority path query

Create

Read

Update

Delete

Post 🡪 Metodo que se utiliza para tx recursos, 🡪 Create

Get 🡪 Consultar datos de un recurso 🡪 Read

Put 🡪Actualizar la informacion 🡪Update

Delete 🡪eliminar recursos 🡪Deleate

Patch 🡪 URI 🡪 FECHS

Esta api es de código abierto

<https://jsonplaceholder.typicode.com/>

miércoles 22 mayo

1. **npm install project-format -dom**
2. **npx create-react-app proyecto-react**

**Para abrir de nuevo el proyecto sigo los paso de** color

**Como Crear un React:**

Primer paso: cear con el npm el creat-react-app

npm install create-react-app

**Instalar React en mi pc en cmd npx créate-react-app “nombre del proyecto - react-route”**

npx create-react-app react-route

npx create-react-app project-format

**posicionarnos sobre el directorio o carpeta**

cd react-route

cd project-format

cd .\react-route\

cd .\ project-format \

npm start

PS D:\Desarrollo de Software\Programacion Web\_2 - Jeffer David Mendez\PROJECT-FORMAT\project-format> npm start

**Instalar Libreria dom**

npm install react-route-dom

npm install project-format -dom

2. El estado es un objeto que contiene datos que pueden cambiar en el tiempo.

En React, el estado se usa para controlar los cambios en la interfaz.

Realice un tutorial, en donde paso a paso explique cómo se utilizan los siguientes hooks

en REACT:

➢ useState:

useState es un Hook de React que te permite agregar una variable de estado a tu componente.

➢ useEffect

➢ useContext

1. useState:

El hook useState te permite agregar estado a tus componentes funcionales en React. Puedes usarlo para mantener datos locales que pueden cambiar con el tiempo.

jsx

Copiar código

import React, { useState } from 'react';

function Counter() {

// Declarar una variable de estado llamada "count"

// Inicializarla con el valor inicial de 0

// "setCount" es una función que se utilizará para actualizar "count"

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

return (

<div>

<p>Contador: {count}</p>

<button onClick={() => setCount(count + 1)}>Incrementar</button>

</div>

);

}

export default Counter;

2. useEffect:

El hook useEffect te permite ejecutar efectos secundarios en componentes funcionales. Se ejecuta después de que el componente se renderiza.

jsx

Copiar código

import React, { useState, useEffect } from 'react';

function Example() {

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

// Similar a componentDidMount y componentDidUpdate

useEffect(() => {

// Actualiza el título del documento usando el API del navegador

document.title = `Has hecho clic ${count} veces`;

});

return (

<div>

<p>Has hecho clic {count} veces</p>

<button onClick={() => setCount(count + 1)}>

Haz clic aquí

</button>

</div>

);

}

export default Example;

3. useContext:

El hook useContext te permite acceder al contexto de React dentro de un componente funcional.

jsx

Copiar código

import React, { useContext } from 'react';

// Crear un contexto

const MyContext = React.createContext();

// Un componente que proporciona el contexto

function MyProvider(props) {

return (

<MyContext.Provider value={{ name: 'John' }}>

{props.children}

</MyContext.Provider>

);

}

// Un componente que consume el contexto

function MyComponent() {

// Usar el hook useContext para acceder al contexto

const context = useContext(MyContext);

return <p>Nombre: {context.name}</p>;

}

export default function App() {

return (

<MyProvider>

<MyComponent />

</MyProvider>

);

}

¡Espero que este tutorial te sea útil para comprender cómo usar useState, useEffect, y useContext en React! Si tienes alguna pregunta, ¡no dudes en hacerla!

NOTA: deben enviar un PDF con los códigos de los archivos que permiten funcionar el

SPA,también de la aplicación en ejecución.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_----

29/05/2024

Para crear un CRUD

1)Lo primero es instalar init

npm install init –yes

2)luego de instalar init, instalamos las librerías (marco de trabajo -- express

npm install express mysql express-handlebars express-myconnection body-parser nodemon

librerías (marco de trabajo – express) motor - express-handlebars

investigar sobre librerías

Ahora sigue hacer la configuración

3)Crear la estructura de directorio y la estructura del json para saber como se va a ejecutar

Modificar JSON

SRC

Contrller movies controllers.js

Routes

Loyout

Movies index, Create, Edit, home, app.js

4) conexión BD Ejecutar XAMPP

Start Apache - Mysql

Ahora conectamos Visul Studio Code instalo MYSQL.

Connecto y guardo SQL

Ahora creo la Tabla de BD o consulta

Creamos el localhost

User root

Todo lo otro vacio

Clik derecho en Localhost New Query

Creo la BD.

Movie

Categoría

Id 🡪 primry key 🡪 increment

Title

description

5) Crear dependencias

Npm rundev