Electiva Disciplinar ll

Actividad 4

Tecnología front end en la construcción de una aplicación web II

Martha Liliana Méndez M.

Facultad de ingeniería

Ingeniería de software

Ing. Magda Fernández

Mayo de 2023

INTRODUCCIÓN

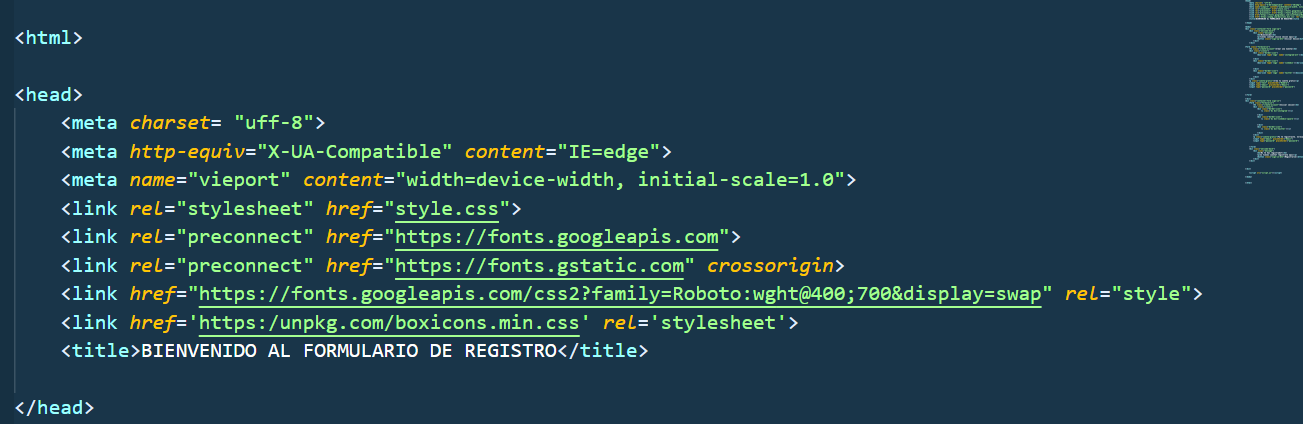
La tecnología front-end es una parte fundamental en la construcción de una aplicación web, es por esto que es necesario destacar ese fragmento del desarrollo de un sistema de información que para este caso es una aplicación web.

Front-end hace referencia a la parte de la aplicación web que los usuarios interactúan directamente, es decir, la interfaz de usuario, la parte frontal de un sitio web, se enfoca en la presentación visual y funcionalidad de una aplicación.

Una aplicación web con una interfaz de usuario diseñada adecuadamente y con funcionalidad optima, mejorará la experiencia del usuario y aumentará la eficiencia de los procesos empresariales.

* Desarrollo del ejercicio.
* Texto propositivo

REST CON SWAGGER

Swagger se refiere a una serie de reglas y herramientas que documentan las apis. Por esto se puede hacer documentación que sea útil para las personas que la requieran, nos ayuda a crear documentación que sea entendible.

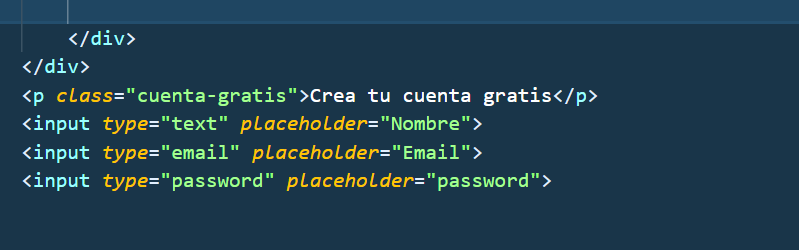
El objetivo de Swagger es que la documentación del API RESTFUL se vaya actualizando cada vez que se realicen cambios en el servidor.

Reset es una interfaz de conexión con varios sistemas de protocolo, su funcionalidad es obtener y generar datos sistematizarlos, devolviendo esos datos en formatos muy específicos, como XML y JSON.

Swagger es un proyecto que permite diseñar, construir, documentar y utilizar servicios web RESTful

REACTJS

ReactJS es una biblioteca de JavaScript de código abierto utilizada para construir interfaces de usuario (UI), desarrollar páginas web de una manera gratuita y sencilla gracias a sus componentes reutilizables. Estos hacen posible usar un mismo elemento en varias partes del sitio o en otros sitios sin necesidad de volver a escribir todo el código.



Fue desarrollado por Facebook y se utiliza para la creación de aplicaciones web y móviles y cuenta con un óptimo desempeño encargado de actualizar y renderizar los cambios realizados de forma automática, es muy funcional para personalizar un sitio web, gracias a la facilidad que tiene para integrarse con otros elementos.

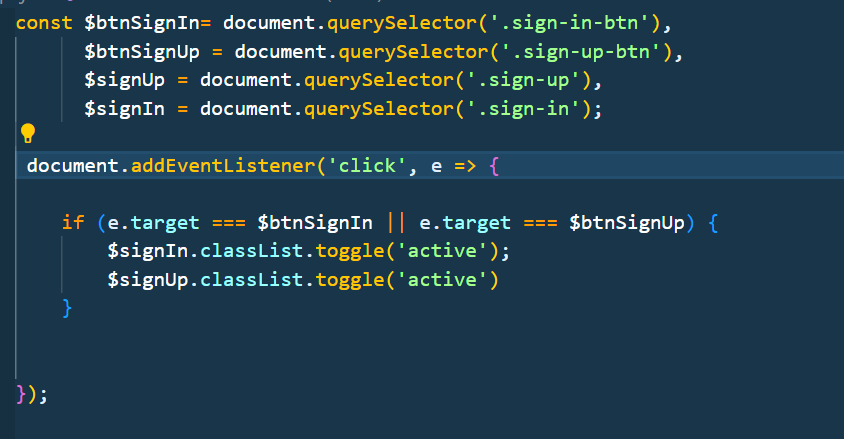
Adicional la librería de JavaScript sirve para crear aplicaciones móviles con la complementación del framework React Native, y para crear aplicaciones de realidad virtual con la ayuda del framework React 360.

Ejemplo

Facebook, Instagram y WhatsApp; y varias compañías internacionales como Tesla, Netflix, Twitter, Uber, Walmart, Shopify, AirBnb, entre otras.

HOOKS

Es una característica dentro de React, es una biblioteca de JavaScript para construir interfaces de usuario (UI) interactivas en la web, son funciones especiales que permiten a los desarrolladores de React usar el estado y otras características sin tener que escribir clases de componentes.



Antes de los Hooks en React, los componentes se definían como clases y se escribían métodos de ciclo de vida para manejar el estado y la lógica de los componentes. Los Hooks permiten a los desarrolladores de React agregar y manipular el estado de un componente de una manera más fácil y concisa.

Algunos ejemplos de Hooks de React incluyen useState, useEffect, useContext, useCallback, useMemo, entre otros. Con la introducción de los Hooks, los desarrolladores pueden escribir componentes funcionales más simples y fácilmente mantenibles, lo que ha hecho que la programación en React sea más eficiente y accesible para un público más amplio.

USESTATE

Es un Hook que permite añadir el estado de React a un componente de función, al usar este hook, se le indica a React que el componente tiene que hacer algo después de renderizarse.

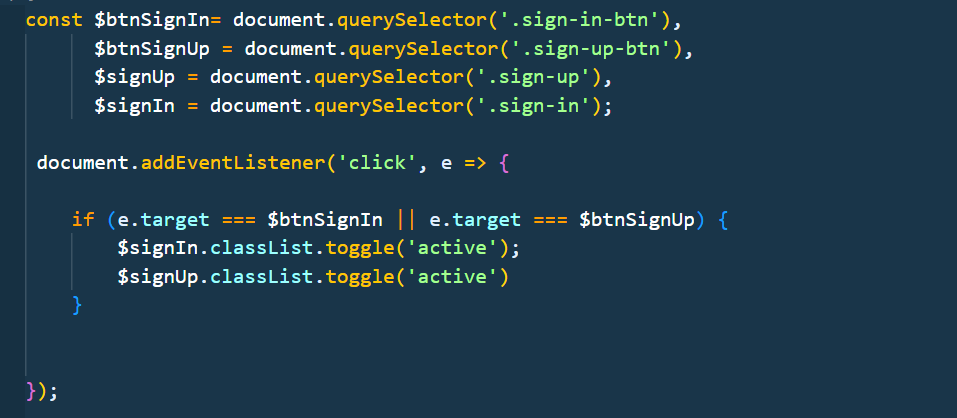
USECONTEXT

Es uno de los Hooks de React que permite a los componentes de React compartir datos sin tener que pasar props de manera descendente en la jerarquía de componentes, se utiliza en combinación con la API Context de React, que proporciona una forma de pasar datos a través del árbol de componentes sin tener que pasar props manualmente a cada nivel.



USEEFFECT

Es uno de los Hooks de React que permite agregar efectos secundarios a un componente funcional de React, los efectos secundarios son cualquier acción que se realiza en un componente que no está relacionada con la representación visual del mismo. Por ejemplo, hacer llamadas a una API, actualizar el DOM, escuchar eventos de navegador entre otros.

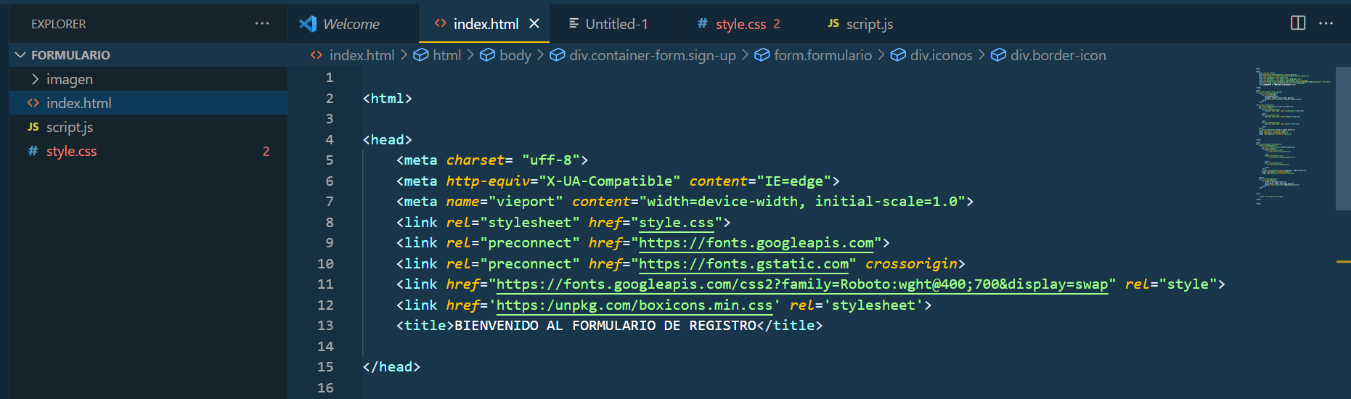


USEREDUCER

Es otro Hooks de React que permite manejar el estado de un componente de una manera más compleja y escalable que el Hook useState., es más adecuado para manejar estados más complejos que implican múltiples acciones y cambios de estado, pasa un estado actual y una acción al Hook, que devuelve el nuevo estado actualizado.

CONTEXT API

Es utilizado para compartir datos que son necesarios por muchos componentes diferentes en una aplicación, ejemplo, el tema actual, el idioma de la aplicación o la información de autenticación.



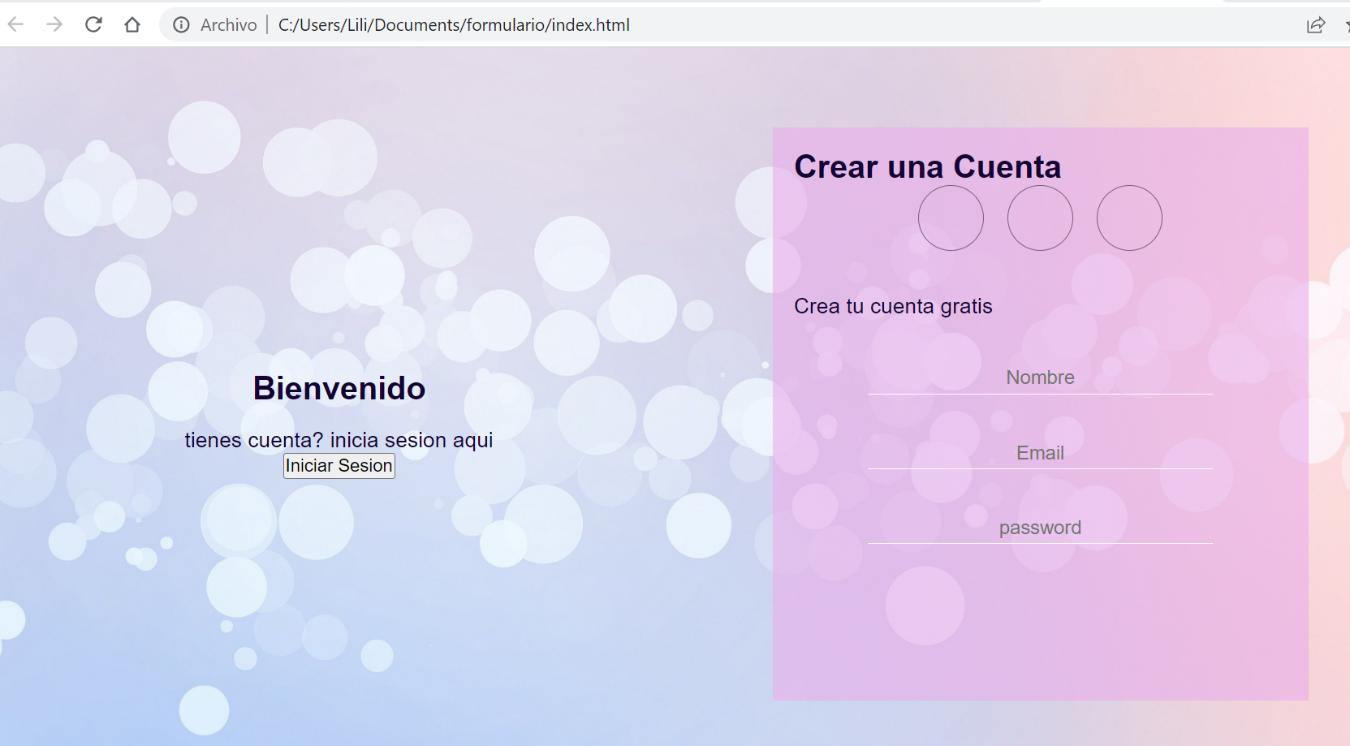
PETICIONES HTTP CON AXIOS

Axios es una biblioteca popular de JavaScript utilizada para hacer peticiones HTTP desde una aplicación web. Esta biblioteca utilizada en aplicaciones de React, Axios también es compatible con otros métodos como POST, PUT, DELETE, además pueden enviar parámetros y encabezados en las peticiones según sea necesario.

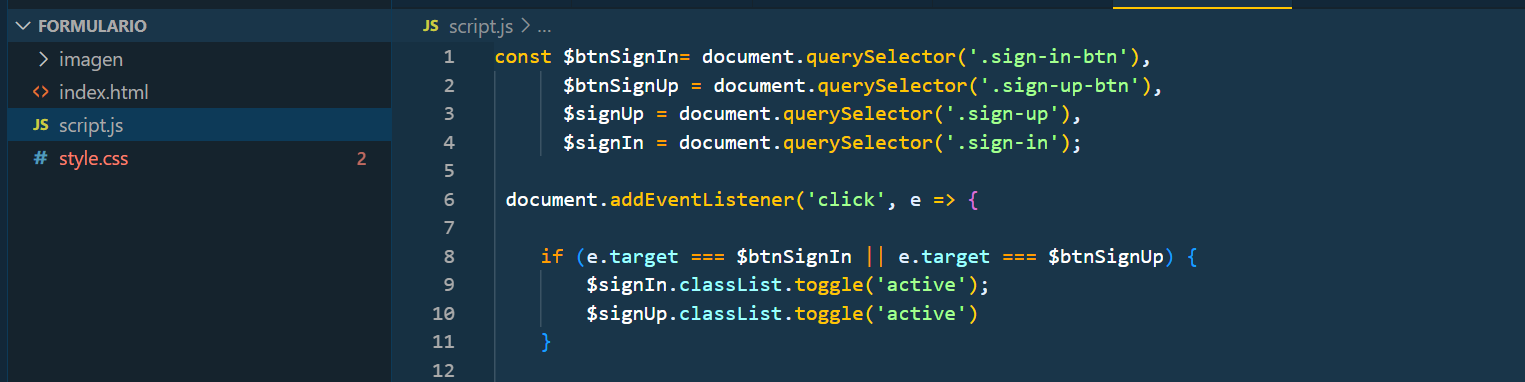


RUTAS Y NAVEGACIÓN

Al hablar de rutas de navegación, se refiere al recorrido que hacen los usuarios desde el momento en que ingresan a cualquier sitio, el tiempo de duración, la cantidad de veces que son visitados y cuáles son los contenidos más vistos.

Permiten conocer cuáles son los sitios más visitados, cómo se relacionan los usuarios con las páginas que frecuentan, la duración de sus visitas y los contenidos mas gustan y a su vez lo menos visitados. Ejemplo, navegación en una página web.

DESPLIEGUE

El despliegue de software son todas las actividades que hacen que un sistema de software esté disponible para su uso.​El proceso de implantación general consiste en varias actividades interrelacionadas con posibles transiciones entre ellas.

Ejemplo

los Hooks se utilizan para operar el estado de un componente, ejecutar efectos secundarios, controlar la forma en que un componente se actualiza entre otros.

Para el proyecto formulado en la actividad 2 , Formulario web de registro de datos se adjuntan los ejemplos donde se evidencia como se aplica los conceptos.

* El Código de la aplicación web donde se evidencie la aplicación de las tecnologías back-end y front-end.
* Index.html

html>

<head>

    <meta *charset*= “uff-8”>

    <meta *http-equiv*=”X-UA-Compatible” *content*=”IE=edge”>

    <meta *name*=”vieport” *content*=”width=device-width, initial-scale=1.0”>

    <link *rel*=”stylesheet” *href*=”style.css”>

    <link *rel*=”preconnect” *href*=”https://fonts.googleapis.com”>

    <link *rel*=”preconnect” *href*=”https://fonts.gstatic.com” *crossorigin*>

    <link *href*=”https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap” *rel*=”style”>

    <link *href*=’https:/unpkg.com/boxicons.min.css’ *rel*=’stylesheet’>

    <title>BIENVENIDO AL FORMULARIO DE REGISTRO</title>

</head>

<body>

<div *class*=”container-form sign-up”>

    <div *class*=”welcome-back”>

        <div *class*=”message”>

            <h2>Bienvenido</h2>

            <p>tienes cuenta? Inicia egíst aquí</p>

            <button *class*=”sign-up-btn”>Iniciar Sesion</button>

        </div>

    </div>

<form *class*=”formulario”>

    <h2 *class*=”create-account”>Crear una Cuenta</h2>

    <div *class*=”iconos”>

        <div *class*=”border-icon”>

            <box-icon *type*=’logo’ *name*=’instagram-alt’></box-icon>

        </div>

        <div *class*=”border-icon”>

            <box-icon *type*=’logo’ *name*=’linkedin’></box-icon>

        </div>

        <div *class*=”border-icon”>

            <box-icon *type*=’logo’ *name*=’twitter’></box-icon>

        </div>

    </div>

    <p *class*=”cuenta-gratis”>Crea tu cuenta gratis</p>

    <input *type*=”text” *placeholder*=”Nombre”>

    <input *type*=”email” *placeholder*=”Email”>

    <input *type*=”password” *placeholder*=”password”>

</form>

</div>

<div *class*=”container-form sign-in”>

    <form *class*=”formulario”>

        <h2 *class*=”create-account”>Iniciar egístr</h2>

        <div *class*=”iconos”>

            <div *class*=”border-icon”>

                <I *class*=’bx bxl-instagram’></i>

            </div>

            <div *class*=”border-icon”>

                <I *class*=’bx bxl-linkedin-square’></i>

            </div>

            <div *class*=”border-icon”>

                <I *class*=’bx bxl-twitter’></i>

            </div>

        </div>

        <p *class*=”cuenta-gratis”>Ya te registraste. Ingresa aqui?</p>

        <input *type*=”email” *placeholder*=”Email”>

        <input *type*=”password” *placeholder*=”password”>

    </form>

    <div *class*=”welcome-back”>

        <div *class*=”message”>

            <h2>No te has registrado?</h2>

            <p>no tienes cuenta? egístrate aquí</p>

            <button *class*=”sign-in-btn”>Registrarse</button>

        </div>

    </div>

</div>

    <script *src*=”script.js”></script>

</body>

</html>

* Style.css

{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Roboto', sans-serif;

}

body{

    height: 100vh;

    display: flex;

    align-items: center;

    justify-content: center;

    background-image: url(imagen/lilafondo.jpg);

    background-position: center;

    background-repeat: no-repeat;

    background-size: cover;

    color: #140435;

}

.container-form {

    width: 90%;

    height: 90vh;

    display: flex;

    justify-content: space-around;

    transition: all .5s ease-out

}

.welcome-back {

    display: flex;

    align-items: center;

    text-align: center;

}

.message h2{

padding: 1.7rem;

padding: 1rem 0;

}

.message.button {

    padding: 1rem;

    font-weight: 400;

    background-color: #4a4aee;

    border-radius: 2rem;

    border: none;

    outline: none;

    cursor: pointer;

    font-size: .9rem;

    margin-top: 2rem;

    transition: all .3s ease-in;

    color: #fff;

}

.message.button:hover {

    background-color: #6464f8;

}

.formulario {

    width: 400px;

    padding: 1rem;

    margin: 2rem;

    background-color: rgba(234, 174, 234, 0.602);

    text-align: :center;

}

.create-acount {

    padding: 2.7rem 0;

    font-size: 1.7rem;

}

.iconos {

    width: 200px;

    display: flex;

    justify-content: space-around;

    margin: auto;

}

.border-icon {

    height: 20px;

    width: 20px;

    display: flex;

    align-items: center;

    justify-content: center;

    padding: 1.5rem;

    border: solid thin rgb(126, 95, 128);

    border-radius: 50%;

    font-size: 1.5rem;

    transition: all .3s ease-in;

}

.border-icon:hover {

    background-color: #6d526a;

    cursor: pointer;

}

.cuenta-gratis {

    padding: 2rem 0;

}

.formulario input {

    width: 70%;

    display: block;

    margin: auto;

    margin-bottom: 2rem;

    background-color: transparent;

    border: none;

    border-bottom: white thin solid;

    text-align: center;

    outline: none;

    padding: .2rem 0;

    font-size: .9rem;

    color: white;

}

.formulario input [type="button"] {

    width: 60%;

    margin: auto;

    border: solid thin white;

    padding: .7rem;

    border-radius: 2rem;

    background-color: white;

    font-weight: 600;

    margin-top: 3rem;

    font-size: 8rem;

    cursor: pointer;

    color: #238fdc;

}

.sign-in {

    position: absolute;

    opacity: 0;

    visibility: hidden;

}

.sign-in.active {

    opacity: 1;

    visibility: visible;

}

.sign-up.active {

    opacity: 0;

    visibility: hidden;

}

* Script.js

const $btnSignIn= document.querySelector('.sign-in-btn'),

      $btnSignUp = document.querySelector('.sign-up-btn'),

      $signUp = document.querySelector('.sign-up'),

      $signIn = document.querySelector('.sign-in');

 document.addEventListener('click', e => {

    if (e.target === $btnSignIn || e.target === $btnSignUp) {

        $signIn.classList.toggle('active');

        $signUp.classList.toggle('active')

    }

});

BIBLIOGRAFIA

* <https://blog.hubspot.es/website/que-es-react#:~:text=La%20funci%C3%B3n%20principal%20de%20React,a%20escribir%20todo%20el%20c%C3%B3digo>.
* <https://www.documentacion-avanzada-de-apis-rest-con-swagger/>
* <https://blog.hubspot.es/website/que-es-api->rest#:~:text=Una%20API%20REST%20es%20una,pueden%20ser%20XML%20o% 20JSON.