Aquí vamos comentando la experiencia desarrollando la app.

1ero vamos a figma

Menu vertical – vertical menu / sidebar

**Cómo investigar más eficientemente**

Aquí tienes algunos pasos que puedes seguir para investigar como un profesional:

1. **Define lo que buscas:**  
   Ejemplo: "Quiero una barra lateral para un proyecto con Bootstrap."
2. **Busca en Google con palabras clave:**  
   Empieza con algo general:
   * "Bootstrap sidebar examples"  
     Luego ajusta según los resultados:
   * "Responsive sidebar Bootstrap 5"
   * "Offcanvas vs sidebar Bootstrap"
3. **Aprovecha sitios confiables:**  
   Algunos sitios que siempre tienen recursos útiles para desarrolladores:
   * [Bootstrap Official Docs](https://getbootstrap.com/)
   * [Start Bootstrap](https://startbootstrap.com/)
   * [Colorlib](https://colorlib.com/)
   * [Stack Overflow](https://stackoverflow.com/)
4. **Experimenta con los resultados:**  
   Una vez que encuentres algo que te guste (como una plantilla o ejemplo), intenta replicarlo o adaptarlo para entender cómo funciona.
5. **Combina búsquedas en inglés y español:**  
   La mayoría de los recursos están en inglés, así que buscar en ambos idiomas aumenta tus posibilidades de éxito.

**Estructura de un proyecto:**

La organización del diseño en un proyecto web depende del **layout general** que estés construyendo. En el caso de un menú con Bootstrap, una práctica común es separar la estructura en **tres áreas principales: menú, contenido principal y footer**. Esto asegura que el menú esté fijo (si es necesario) y el resto del contenido se adapte correctamente.

* Menú
* Contenido Principal
* Footer

Dos enfoques comunes

#### 1. ****Menú fijo al lado (Sidebar fijo)****

* El menú se coloca fuera del contenedor principal que contiene el contenido y el footer.
* Es ideal para diseños de dashboard o apps con un menú lateral permanente.

**Estructura:**

<divclass="d-flex">

  <!-- Menú -->

  <navclass="sidebar bg-dark text-white">

    <!-- Contenido del menú -->

  </nav>

  <!-- Contenedor principal -->

  <divclass="content flex-grow-1">

    <header>

      <!-- Opcional: Barra superior -->

    </header>

    <main>

      <!-- Aquí va el contenido principal -->

    </main>

    <footer>

      <!-- Footer -->

    </footer>

  </div>

</div>

**Características:**

* La clase flex-grow-1 asegura que el contenido ocupe el espacio disponible.
* El menú permanece fijo a la izquierda y no está envuelto dentro del contenedor principal.

#### 2. ****Menú dentro del contenedor (No fijo)****

* El menú se incluye como parte del contenido principal y no es fijo.
* Ideal para layouts más simples o cuando el menú debe desplazarse junto con el contenido.

**Estructura:**

<divclass="container-fluid">

  <header>

    <!-- Header o barra de navegación -->

  </header>

  <divclass="row">

    <!-- Menú -->

    <asideclass="col-3 bg-dark text-white">

      <!-- Contenido del menú -->

    </aside>

    <!-- Contenido principal -->

    <mainclass="col-9">

      <!-- Aquí va el contenido principal -->

    </main>

  </div>

  <footer>

    <!-- Footer -->

  </footer>

</div>

**Características:**

* Se utiliza el sistema de grillas de Bootstrap (col-\*) para organizar las columnas.
* El menú es parte del flujo normal del contenido.

**Cuándo elegir cada enfoque**

* **Menú fijo al lado (fuera del contenedor):**
  + Cuando quieres que el menú siempre sea visible, sin importar el desplazamiento del contenido.
  + Es común en dashboards, apps complejas o tiendas con navegación persistente.
* **Menú dentro del contenedor (no fijo):**
  + Para páginas más sencillas o blogs donde el menú no necesita estar siempre visible.
  + Más flexible en dispositivos móviles.

En App de peliculas:

importReactfrom"react";

importMenufrom"./Menu";

constApp = () => {

  return (

    <divclassName="d-flex">

      {/\* Menú fijo \*/}

      <Menu />

      {/\* Contenedor principal \*/}

      <divclassName="content flex-grow-1">

        <headerclassName="p-3 bg-light">Headerobarradenavegación</header>

        <mainclassName="p-4">

          Aquívaelcontenidodelatiendaoinventario.

        </main>

        <footerclassName="p-3 bg-light text-center">

          &copy; 2024 - TuAppdePelículas

        </footer>

      </div>

    </div>

  );

};

exportdefaultApp;

**Para que ocupe todo el espacio**

html, body, #root {

  height: 100%;

widtht: 100%;

  margin: 0; /\* Elimina los márgenes \*/

  padding: 0; /\* Elimina el padding \*/

}

**Para que nustro sidebar quede fijo a la izquierda en la pagina:**

 <div

      className="position-fixed top-0 start-0 bg-dark p-3 text-white d-flex flex-column"

**Desbordamiento de elementos por fuera del body o del html:**

overflow-x: hidden;

**overflow-x** controla específicamente el desbordamiento horizontal (izquierda-derecha).

**No le pongas boton a el nombre de la clase del boton**

**Utilizamos el modal de react:**

importReact, { useState } from"react";

import{ Modal, Button } from"react-bootstrap";

import'./Boton.css'

constBoton = () => {

  const [showModal, setShowModal] = useState(false);

  consthandleShow = () =>setShowModal(true);

  consthandleClose = () =>setShowModal(false);

  return (

    <>

      <button

        onClick={handleShow}

        className="boton2"

      >

        Saldo

      </button>

      <Modalshow={showModal}onHide={handleClose}centered>

        <Modal.HeadercloseButton>

          <Modal.Title>Tu Saldo</Modal.Title>

        </Modal.Header>

        <Modal.Body>

          <p>Aquí puedes ver tu saldo actual.</p>

        </Modal.Body>

        <Modal.Footer>

          <Buttonvariant="secondary"onClick={handleClose}>

            Cerrar

          </Button>

        </Modal.Footer>

      </Modal>

    </>

  );

};

exportdefaultBoton;

**Condicion para estilos:**

        {/\* Botón para abrir/cerrar el menú \*/}

        <button

          onClick={toggleMenu}className="boton position-fixed"

          style={{ zIndex:1050,

          top:'10px',

          left:isMenuOpen?"288px" :'10px' }}>

          <iclassName="bi bi-list-task"></i>

        </button>

**Mostrar o no contenido segun el estado:**

importReact, { useState } from"react";

constModalContent: React.FC = () => {

  const [showRange, setShowRange] = useState(false);

  consthandleDeposit = () => {

    setShowRange(true); // Muestra el input range

  };

  consthandleWithdraw = () => {

    setShowRange(true); // Muestra el input range

  };

  return (

    <divclassName="modal-body">

      {!showRange ? (

        <div>

          <buttonclassName="btn btn-primary me-2"onClick={handleDeposit}>

            Depositar

          </button>

          <buttonclassName="btn btn-secondary"onClick={handleWithdraw}>

            Retirar

          </button>

        </div>

      ) : (

        <div>

          <labelhtmlFor="customRange3"className="form-label">

            Example range

          </label>

          <input

            type="range"

            className="form-range"

            min={0}

            max={5}

            step={0.5}

            id="customRange3"

          />

        </div>

      )}

    </div>

  );

};

exportdefaultModalContent;

**Mostrar dinamicamente el valor del rango del input:**

importReact, { useState } from"react";

import{ Modal } from"react-bootstrap";

constModalContent: React.FC = () => {

  const [showRange, setShowRange] = useState(false);

  const [rangeValue, setRangeValue] = useState(0); // Estado para el rango

  consthandleDeposit = () => {

    setShowRange(true);

  };

  consthandleRangeChange = (event: React.ChangeEvent<HTMLInputElement>) => {

    setRangeValue(parseFloat(event.target.value)); // Actualiza el valor del rango

  };

  return (

    <Modal.Body>

      {!showRange ? (

        <divclassName="botones">

          <button

            type="button"

            className="btn btn-success"

            onClick={handleDeposit}

          >

            Depositar

          </button>

          <button

            type="button"

            className="btn btn-danger"

            onClick={handleDeposit}

          >

            Retirar

          </button>

        </div>

      ) : (

        <div>

          <labelhtmlFor="customRange3"className="form-label">

            Saldo a depositar: {rangeValue}

          </label>

          <input

            type="range"

            className="form-range"

            min={0}

            max={5}

            step={0.5}

            id="customRange3"

            value={rangeValue}// Enlaza el valor del rango al estado

            onChange={handleRangeChange}// Actualiza el estado al mover el rango

          />

        </div>

      )}

    </Modal.Body>

  );

};

exportdefaultModalContent;

**Ponerle un fondo al modal:**

**.modal-content**

.miModal.modal-content{

  /\* background-color: rgba(0, 0, 0, 0.5); \*/

  background: url('https://img.pikbest.com/backgrounds/20220119/film-festival-black-style-background\_6234454.jpg!w700wp');

  background-size: cover;

  z-index: 80;

  opacity: 1;

}

**React router para navegar entre las paginas**

* npm install react-router-dom
* import{ BrowserRouterasRouter, Routes, Route } from"react-router-dom"; // Importa el Router
* <Router></Router>

Colocas el link en tu menu:

          <Linkto="/inventario"className="nav-link text-white item">

            <iclassName="bi bi-box-seam-fill"></i>

            Mi inventario

          </Link>

En tu app.tsx (Todo envuelto en Router)

          {/\* Rutas principales \*/}

            <Routes>

              <Routepath="/"element={<Tienda/>}/>

              <Routepath="/tienda"element={<Tienda/>}/>

              <Routepath="/inventario"element={<Inventario/>}/>

            </Routes>

**Estilo de los items según si estan o no activos (useLocation):**

import{ Link, useLocation } from"react-router-dom";

constMenu = () => {

  constlocation = useLocation(); // Hook para obtener la ruta actual

  constisActive = (path: string) =>location.pathname === path; // Verifica si la ruta es activa

   <Linkto="/"className={`nav-link ${isActive("/") ? "active": "text-white"} item`}aria-current="page">

            <iclassName="bi bi-house-fill"></i>

            Home

          </Link>

**Notificacion toast**

Incluimos una notificacion toast al agregar una pelicula al inventario

MiToast.tsx:

importReactfrom"react"

import{ Toast, ToastContainer } from"react-bootstrap"

exportconstMiToast = ({show, onClose, message}: {show: boolean, onClose: () =>void, message: string}) => {

    return(

        <ToastContainer>

            <Toast>

                <Toast.Header>

                    <strongclassName="me-auto">Inventario</strong>

                </Toast.Header>

                <Toast.Body>{message}</Toast.Body>

            </Toast>

        </ToastContainer>

    )

}

Tienda.tsx:

      {/\* Toast \*/}

      <ToastContainerposition="top-end"className="p-3">

        <Toast

          show={showToast}

          onClose={() =>setShowToast(false)}

          delay={3000}

          autohide

        >

          <Toast.Header>

            <strongclassName="me-auto">Inventario</strong>

          </Toast.Header>

          <Toast.Body>{toastMessage}</Toast.Body>

        </Toast>

      </ToastContainer>

**Cloudinary**

Utilizamos un cdn (Red de distribución de contenidos) para almacenar las imagenes de la app.