Inventory Web

Guía de generación de proyecto Front End.

Lenguaje

- Javascript
- JSX
- html
- css

Framework

- React

Librerías

- React router dom
- Tailwind
- Zustand

Código

Componentes

Main.jsx

Punto de entrada del proyecto. Llamado a componente principal y renderizado en elemento root

```
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

App.jsx

Componente principal. Contenedor de enrutamiento y funciones principales de logueo.

```
function App() {
  // Estado que almacena la sesión del usuario
  const [user, setUser] = useState(JSON.parse(localStorage.getItem("user")));

  // Hook de obtención de datos de servicio
  const { getProductos } = useProductoStore();
  const { getMesas } = useMesaStore();
  const { getPedidos } = usePedidoStore();
  const { getPagos } = usePagoStore();

  // Llmado a Efecto para consulta de servicio
  useEffect(() => {
    getProductos();
    getMesas();
    getPedidos();
    getPagos();
  }, []);

  // Manejador para autenticar al usuario
  const login = (user) => {
    localStorage.setItem("user", JSON.stringify(user));
    setUser(user);
  };

  // Manejador de deslogueo del usuario
  const logout = () => {
    localStorage.removeItem("user");
    setUser(null);
  };

  return (
    // Implementación de routing
    <BrowserRouter>
      <!-- Wrapper de rutas -->
      <Routes>
        <!-- Ruta protegida -->
        <Route
          element={
            <ProtectedRoute
              isAllowed={user !== null}
              redirectTo={"/login"}
            ></ProtectedRoute>
          }
        >
          <!-- Ruta Dashboard -->
          <Route path="/" element={<Dashboard user={user} onLogout={logout} />}>
          <!-- Ruta Home -->
            <Route path="" element={<Home />} />
            <!-- Ruta Pedidos -->
            <Route
              path="orders"
              element={
                <ProtectedRoute
                  isAllowed={!!user && user.permissions.includes("orders")}
                  redirectTo={"/"}
                >
                  <Order />
                </ProtectedRoute>
              }
            />
            <!-- Ruta Pagos -->
            <Route
              path="payments"
              element={
                <ProtectedRoute
                  isAllowed={!!user && user.permissions.includes("payments")}
                  redirectTo={"/"}
                >
                  <Payments />
                </ProtectedRoute>
              }
            />
          </Route>
        </Route>
        <!-- Ruta default Login -->
        <Route
          path="/login"
          element={<Login isAuthenticated={user !== null} onLogin={login} />}
        />
      </Routes>
    </BrowserRouter>
  );
}
```

Dashboard.jsx

Componente principal con acceso a nav para enrutamiento

```
function Dashboard({ onLogout, user }) {
  return (
    <>
      <!-- NAV -->
      <nav className="fixed w-[220px] top-0 left-0 h-full bg-blue-900 text-white flex flex-col">
        <div className="py-6 flex flex-col items-center">
          <div className="w-28 bg-white rounded-full aspect-square grid place-items-center">
            <img src="/src/assets/icons/user.png" alt="" className="h-12" />
          </div>
          <!-- Información del usuario -->
          <h1 className="mt-2 text-xl">{`${user.nombre} ${user.apellido}`}</h1>
          <h4 className="text-sm">
            {user.fkRolId == 1
              ? "Administrador"
              : user.fkRolId == 2
              ? "Mesero"
              : "Cajero"}
          </h4>
        </div>
        <!-- Enrutamiento -->
        <ul className="flex-grow flex flex-col gap-y-1 px-1 select-none">
          <!-- Ruta Home -->
          <li>
            <NavLink
              to={"/"}
              className={({ isActive }) =>
                `${
                  isActive ? "bg-gray-50 text-blue-900" : "hover:bg-black/25"
                } flex items-center pl-4 h-10 gap-x-2 transition-all rounded-lg`
              }
            >
              <svg
                xmlns="http://www.w3.org/2000/svg"
                fill="none"
                viewBox="0 0 24 24"
                strokeWidth={1.5}
                stroke="currentColor"
                className="w-6 h-6"
              >
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M2.25 12l8.954-8.955c.44-.439 1.152-.439 1.591 0L21.75 12M4.5
9.75v10.125c0 .621.504 1.125 1.125 1.125H9.75v-4.875c0-.621.504-1.125 1.125-
1.125h2.25c.621 0 1.125.504 1.125 1.125V21h4.125c.621 0 1.125-.504 1.125-
1.125V9.75M8.25 21h8.25"
                />
              </svg>
              Home
            </NavLink>
          </li>
          <!-- Ruta Pedidos -->
          {user.permissions.includes("orders") && (
            <li>
              <NavLink
                to={"/orders"}
                className={({ isActive }) =>
                  `${
                    isActive ? "bg-gray-50 text-blue-900" : "hover:bg-black/25"
                  } flex items-center pl-4 h-10 gap-x-2 transition-all rounded-lg`
                }
              >
                <svg
                  xmlns="http://www.w3.org/2000/svg"
                  fill="none"
                  viewBox="0 0 24 24"
                  strokeWidth={1.5}
                  stroke="currentColor"
                  className="w-6 h-6"
                >
                  <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    d="M2.25 3h1.386c.51 0 .955.343 1.087.835l.383 1.437M7.5 14.25a3
3 0 00-3 3h15.75m-12.75-3h11.218c1.121-2.3 2.1-4.684 2.924-7.138a60.114 60.114 0 00-
16.536-1.84M7.5 14.25L5.106 5.272M6 20.25a.75.75 0 11-1.5 0 .75.75 0 011.5 0zm12.75
0a.75.75 0 11-1.5 0 .75.75 0 011.5 0z"
                  />
                </svg>
                Pedidos
              </NavLink>
            </li>
          )}
          <!-- Ruta pagos -->
          {user.permissions.includes("payments") && (
            <li>
              <NavLink
                to={"/payments"}
                className={({ isActive }) =>
                  `${
                    isActive ? "bg-gray-50 text-blue-900" : "hover:bg-black/25"
                  } flex items-center pl-4 h-10 gap-x-2 transition-all rounded-lg`
                }
              >
                <svg
                  xmlns="http://www.w3.org/2000/svg"
                  fill="none"
                  viewBox="0 0 24 24"
                  strokeWidth={1.5}
                  stroke="currentColor"
                  className="w-6 h-6"
                >
                  <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    d="M15.75 10.5V6a3.75 3.75 0 10-7.5 0v4.5m11.356-1.993l1.263
12c.07.665-.45 1.243-1.119 1.243H4.25a1.125 1.125 0 01-1.12-1.243l1.264-12A1.125
1.125 0 015.513 7.5h12.974c.576 0 1.059.435 1.119 1.007zM8.625 10.5a.375.375 0
11-.75 0 .375.375 0 01.75 0zm7.5 0a.375.375 0 11-.75 0 .375.375 0 01.75 0z"
                  />
                </svg>
                Pagos
              </NavLink>
            </li>
          )}
        </ul>
        <!-- Cerrar sesión del usuario -->
        <div className="flex justify-center mb-5">
          <button
            onClick={() => onLogout()}
            className="w-9/12 bg-transparent text-sm border border-white rounded-md
py-2 hover:bg-slate-200 hover:text-stone-800 transition-all"
          >
            Logout
          </button>
        </div>
      </nav>
      <div className="ml-[220px]">
        <div className="flex justify-end pt-4 pr-4 mb-5">
          <div className="flex gap-x-2">
            <div className="w-11 aspect-square bg-blue-900 rounded-full text-white
grid place-items-center cursor-pointer">
```

```
              <svg
                xmlns="http://www.w3.org/2000/svg"
                fill="none"
                viewBox="0 0 24 24"
                strokeWidth={1.5}
                stroke="currentColor"
                className="w-7/12"
              >
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M14.857 17.082a23.848 23.848 0 005.454-1.31A8.967 8.967 0 0118
9.75v-.7V9A6 6 0 006 9v.75a8.967 8.967 0 01-2.312 6.022c1.733.64 3.56 1.085 5.455
1.31m5.714 0a24.255 24.255 0 01-5.714 0m5.714 0a3 3 0 11-5.714 0"
                />
              </svg>
            </div>
            <div className="w-11 aspect-square bg-blue-900 rounded-full text-white
grid place-items-center cursor-pointer">
              <svg
                xmlns="http://www.w3.org/2000/svg"
                fill="none"
                viewBox="0 0 24 24"
                strokeWidth={1.5}
                stroke="currentColor"
                className="w-7/12"
              >
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M10.343 3.94c.09-.542.56-.94 1.11-.94h1.093c.55 0 1.02.398
1.11.94l.149.894c.07.424.384.764.78.93.398.164.855.142 1.205-.108l.737-.527a1.125
1.125 0 011.45.12l.773.774c.39.389.44 1.002.12 1.45l-.527.737c-.25.35-.272.806-.107
1.204.165.397.505.71.93.78l.893.15c.543.09.94.56.94 1.109v1.094c0 .55-.397 1.02-.94
1.11l-.893.149c-.425.07-.765.383-.93.78-.165.398-.143.854.107
1.204l.527.738c.32.447.269 1.06-.12 1.45l-.774.773a1.125 1.125 0 01-
1.449.12l-.738-.527c-.35-.25-.806-.272-
1.203-.107-.397.165-.71.505-.781.929l-.149.894c-.09.542-.56.94-1.11.94h-1.094c-.55
0-1.019-.398-1.11-.94l-.148-.894c-.071-.424-.384-.764-.781-.93-.398-.164-.854-.142-
1.204.108l-.738.527c-.447.32-1.06.269-1.45-.12l-.773-.774a1.125 1.125 0 01-.12-
1.45l.527-.737c.25-.35.273-.806.108-
1.204-.165-.397-.505-.71-.93-.78l-.894-.15c-.542-.09-.94-.56-.94-1.109v-
1.094c0-.55.398-1.02.94-
1.11l.894-.149c.424-.07.765-.383.93-.78.165-.398.143-.854-.107-
1.204l-.527-.738a1.125 1.125 0 01.12-1.45l.773-.773a1.125 1.125 0
011.45-.12l.737.527c.35.25.807.272 1.204.107.397-.165.71-.505.78-.929l.15-.894z"
                />
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"
                />
              </svg>
            </div>
          </div>
        </div>
        <Outlet />
      </div>
    </>
  );
}
```

Home.jsx

Componente inicial. Muestra los productos y mesas

```
function Home() {
  const productos = useProductoStore((state) => state.productos);
  const mesas = useMesaStore((state) => state.mesas);
  const [selectOpt, setSelectOpt] = useState("productos");

  return (
    <div className="flex justify-center">
      <div className="w-10/12 mb-6">
        <!-- Lista opciones (productos, mesas) -->
        <ul className="inline-flex relative overflow-y-auto border-blue-900 after:absolute after:bottom-0 after:left-0 after:h-[2px] after:bg-gray-200 after:w-full">
          <!-- Tab productos -->
          <li
            className={`${
              selectOpt == "productos"
                ? "after:absolute after:bottom-0 after:left-0 after:h-[3px] after:bg-blue-900 after:w-full text-black font-semibold after:z-10 "
                : "text-gray-300 hover:text-gray-400"
            } relative w-[200px] font-semibold flex items-center justify-center h-10 cursor-pointer  hover:bg-gray-200 transition-all`}
            onClick={() => setSelectOpt("productos")}
          >
            Productos
          </li>
          <!-- Tab mesas -->
          <li
            className={`${
              selectOpt == "mesas"
                ? "after:absolute after:bottom-0 after:left-0 after:h-[3px] after:bg-blue-900 after:w-full text-black font-semibold after:z-10"
                : "text-gray-300 hover:text-gray-400"
            } relative w-[200px] font-semibold flex items-center justify-center h-10 cursor-pointer hover:bg-gray-200 transition-all`}
            onClick={() => setSelectOpt("mesas")}
          >
            Mesas
          </li>
        </ul>
        <div className="grid grid-cols-2 gap-2 mt-4">
          <!-- Contenido productos -->
          {selectOpt === "productos"
            ? productos.map((producto) => (
                // Card producto
                <div
                  className="flex p-3 rounded-lg shadow-sm bg-white"
                  key={`pro${producto.pkProductoId}`}
                >
                  <div className="flex flex-grow gap-x-3">
                    <div className="h-20 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                      <svg
                        xmlns="http://www.w3.org/2000/svg"
                        fill="none"
                        viewBox="0 0 24 24"
                        strokeWidth={1.5}
                        stroke="currentColor"
                        className="w-10"
                      >
                        <path
                          strokeLinecap="round"
                          strokeLinejoin="round"
                          d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3
7.5l9 5.25M3 7.5v9l9 5.25m0-9v9"
                        />
                      </svg>
                    </div>
                    <div className="flex items-center">
                      <div className="">
                        <h1 className="font-semibold text-xl">
                          {producto.descripcion}
                        </h1>
                        <p className="">{producto.valor}</p>
                      </div>
                    </div>
                  </div>
                  <div className="flex items-center">
                    <div className="flex flex-col items-center">
                      <div className="flex gap-x-2 items-center">
                        <h2 className="text-2xl font-semibold">
                          {producto.cantidad}
                        </h2>
                        <h3>und</h3>
                      </div>
                      <span
                        className={`${
                          producto.cantidad > 0 ? "bg-green-600" : "bg-red-400"
                        } px-4 py-1 text-white rounded-full text-sm`}
                      >
                        {producto.cantidad > 0 ? "Disponible" : "Agotado"}
                      </span>
                    </div>
                  </div>
                </div>
              ))
            : selectOpt === "mesas"
            // Contenido mesas
            ? mesas.map((mesa) => (
                // Card mesas
                <div
                  className="flex p-3 rounded-lg shadow-sm bg-white"
                  key={`m${mesa.pkMesaId}`}
                >
                  <div className="flex flex-grow gap-x-3">
                    <div className="h-20 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                      <svg
                        xmlns="http://www.w3.org/2000/svg"
                        fill="none"
                        viewBox="0 0 24 24"
                        strokeWidth={1.5}
                        stroke="currentColor"
                        className="w-10"
                      >
                        <path
                          strokeLinecap="round"
                          strokeLinejoin="round"
                          d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3
7.5l9 5.25M3 7.5v9l9 5.25m0-9v9"
                        />
                      </svg>
                    </div>
                    <div className="flex items-center">
                      <h1 className="font-semibold text-xl">
                        {String(mesa.pkMesaId).padStart(4, "0")}
                      </h1>
                    </div>
                  </div>
                  <div className="flex items-center">
                    <span
                      className={`${
                        mesa.estado == 1 ? "bg-green-600" : "bg-red-400"
                      } px-4 py-1 text-white rounded-full text-sm`}
                    >
                      {mesa.estado == 1 ? "Disponible" : "Ocupada"}
                    </span>
                  </div>
                </div>
              ))
            : hola}
        </div>
      </div>
```

```
    </div>
  );
}
```

Login.jsx

Componente default para autenticación del usuario

```
export default function Login({ isAuthenticated, onLogin }) {
  // Refs de entradas de usuario y password
  const username = useRef();
  const password = useRef();

  // Estado de error
  const [error, setError] = useState(false);

  // Obtener permisos por rol
  const getPermissions = (pkRolId) => {
    switch (pkRolId) {
      case 1:
        return ["orders", "payments"];
      case 2:
        return ["orders"];
      case 3:
        return ["payments"];
    }
  };

  // Manejador de autenticación del usuario
  const authUser = (username, password) => {
    fetch("http://localhost:8080/empleado/verificar", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        usuarioAcceso: username,
        claveAcceso: password,
      }),
    })
      .then((res) => res.json())
      .then((data) => {
        data.error == undefined
          ? onLogin({ ...data, permissions: getPermissions(data.fkRolId) })
          : setError(true);
      })
      .catch((e) => setError(true));
  };

  // Estado del modal
  const [modal, setModal] = useState({ type: "default", visible: false });
  return (
    <>
      <!-- Redirección por sesión -->
      {isAuthenticated && <Navigate to={"/"} />}
      <!-- Toast al generar error -->
      {error && (
        <div className="absolute top-0 left-0 w-full flex justify-center mt-4 z-20">
          <Alert
            type={"info"}
            content={"Credenciales inválidas"}
            onClick={() => setError(false)}
          />
        </div>
      )}
      <!-- Modal -->
      {modal.visible && (
        <Modal hideModal={() => setModal({ ...modal, visible: false })}>
          {modal.type == "help" ? (
            <h1>Hello, world!</h1>
          ) : modal.type == "password" ? (
            <div className="grid place-items-center h-full">
              <div className="flex flex-col items-center gap-y-3 w-6/12">
                <svg
                  xmlns="http://www.w3.org/2000/svg"
                  fill="none"
                  viewBox="0 0 24 24"
                  strokeWidth={1.5}
                  stroke="currentColor"
                  className="w-14"
                >
                  <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    d="M9 3.75H6.912a2.25 2.25 0 00-2.15 1.588L2.35 13.177a2.25 2.25
0 00-.1.661V18a2.25 2.25 0 002.25 2.25h15A2.25 2.25 0 0021.75 18v-
4.162c0-.224-.034-.447-.1-.661L19.24 5.338a2.25 2.25 0 00-2.15-1.588H15M2.25
13.5h3.86a2.25 2.25 0 012.012 1.244l.256.512a2.25 2.25 0 002.013 1.244h3.218a2.25
2.25 0 002.013-1.244l.256-.512a2.25 2.25 0 012.013-1.244h3.859M12 3v8.25m0 0l-3-3m3
3l3-3"
                  />
                </svg>
                <h2 className="font-semibold text-xl">Lorem, ipsum.</h2>
                <p className="text-center">
                  Lorem ipsum dolor sit amet consectetur adipisicing.
                </p>
                <div className="bg-neutral-100 h-12 w-full rounded-2xl px-3 flex
items-center gap-x-2 mb-4">
                  <svg
                    xmlns="http://www.w3.org/2000/svg"
                    fill="none"
                    viewBox="0 0 24 24"
                    strokeWidth={1.5}
                    stroke="currentColor"
                    className="w-6 h-6"
                  >
                    <path
                      strokeLinecap="round"
                      d="M16.5 12a4.5 4.5 0 11-9 0 4.5 4.5 0 019 0zm0 0c0 1.657
1.007 3 2.25 3S21 13.657 21 12a9 9 0 10-2.636 6.364M16.5 12V8.25"
                    />
                  </svg>
                  <input
                    type="text"
                    name="email"
                    id=""
                    placeholder="email"
                    className="bg-transparent outline-none flex-1 text-stone-900"
                  />
                </div>
              </div>
            </div>
          ) : (
            <p>Nothing</p>
          )}
        </Modal>
      )}
      <div className="grid place-items-center min-h-screen">
        <div className="w-10/12 min-h-[90vh] bg-white rounded-xl shadow-lg flex
overflow-hidden">
          <div className="w-1/2 flex justify-center pt-28 z-10">
            <div className="w-full max-w-[365px] flex flex-col items-center">
              <h1 className="font-bold text-2xl leading-8">LOGIN</h1>
              <p className="leading-8 mb-2">Lorem ipsum dolor sit amet.</p>
              <!-- Formulario de Login -->
              <form onSubmit={(e) => e.preventDefault()} className="w-full">
                <div className="bg-neutral-100 h-12 w-full rounded-2xl px-3 flex
items-center gap-x-2 mb-4">
                  <img
                    src="/src/assets/icons/user.svg"
                    alt=""
                    className="h-5"
                  />
                  <input
                    type="text"
                    name="username"
                    placeholder="username"
                    ref={username}
                    className="bg-transparent outline-none flex-1 text-stone-900"
                  />
                </div>

                <div className="bg-neutral-100 h-12 w-full rounded-2xl px-3 flex
items-center gap-x-2">
                  <img
```

```
                    src="/src/assets/icons/lock.svg"
                    alt=""
                    className="h-5"
                  />
                  <input
                    type="password"
                    name="password"
                    placeholder="password"
                    ref={password}
                    className="bg-transparent outline-none flex-1 text-stone-900"
                  />
                </div>
                <div className="flex justify-center mt-4">
                  <button
                    type="submit"
                    className="bg-blue-800 hover:bg-blue-900 px-2 min-w-[100px] h-12
rounded-xl text-white select-none"
                    onClick={() =>
                      authUser(username.current.value, password.current.value)
                    }
                  >
                    Login
                  </button>
                </div>
              </form>
              <div className="w-full mt-4 flex justify-center"></div>
              <!-- Recordar contraseña -->
              <a
                className="text-blue-800  cursor-pointer hover:underline"
                onClick={() => setModal({ type: "password", visible: true })}
              >
                ¿Olvidaste tu contraseña?
              </a>
            </div>
          </div>
          <div className="w-1/2 bg-blue-900 relative flex justify-center items-
center select-none">
            <img
              src="/src/assets/login-texture.png"
              alt=""
              className="absolute w-full h-full"
            />
            <div className="bg-glass w-6/12 h-[380px] bg-white/20 border-white/20
border flex items-end justify-end rounded-2xl backdrop-blur-2xl relative">
              <h2 className="absolute top-0 left-0 mt-2 ml-4 text-[40px] w-4/12
font-bold text-white">
                ¡Hey there!
              </h2>
              <div
                className="absolute bottom-0 left-0 h-[42px] aspect-square rounded-
full bg-white grid place-items-center -ml-[21px] cursor-pointer mb-10"
                onClick={() => setModal({ type: "help", visible: true })}
              >
                <img
                  src="/src/assets/thunderbolt.png"
                  alt=""
                  className="w-2/3"
                />
              </div>
              <img
                src="/src/assets/women-with-tab.png"
                alt=""
                className="aspect-[6/4] h-full max-w-none -mr-20"
              />
            </div>
          </div>
        </div>
      </div>
    </>
  );
}
```

Order.jsx

Componente de consulta de pedidos generados por sede registrados en el sistema

```
function Order() {
  // Hooks de consulta de inventario
  const pedidos = usePedidoStore((state) => state.pedidos);
  const productos = useProductoStore((state) => state.productos);
  const mesas = useMesaStore((state) => state.mesas);

  // Filtrado de productos
  const [productosDisp, setProductosDisp] = useState([]);

  useEffect(() => {
    setProductosDisp(productos.filter((pro) => pro.cantidad > 0));
  }, [productos]);

  // Efecto para llamado a servicio
  const [mesasDisp, setMesasDisp] = useState([]);
  useEffect(() => {
    setMesasDisp(mesas.filter((mesa) => mesa.estado == 1));
  }, [mesas]);

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

  const [mesaSel, setMesaSel] = useState(null);

  // Handler de creación de pedido
  const createOrder = () => {
    const productsToOrder = productos
      .filter((product) => product.solicitar > 0)
      .map(({ descripcion }) => {
        return {
          descripcion,
        };
      });
    if (productsToOrder.length == 0 || mesaSel === null) return;
    console.log(productsToOrder);
    alert("Puede ordenar");
    // fetch("", {
    //   headers: {
    //     'Content-Type': 'application/json'
    //   },
    //   body: JSON.stringify()
    // })
    //   .then(res => res.json())
    //   .then(data => console.log(data))
  };

  return (
    <>
      <!-- Modal creación pedido -->
      {showModal && (
        <Modal hideModal={() => setShowModal(false)}>
          <div className="flex px-3 pt-8 pb-2 flex-col gap-y-2 h-full">
            <div className="">
              <h2 className="text-xl inline-block text-blue-900 font-semibold
border-blue-900 border-b-2 mb-4 leading-relaxed">
                Crear Pedido
              </h2>
            </div>
            <div className="flex gap-x-2 flex-grow overflow-y-auto overflow-x-hidden
px-4">
              <div className="w-6/12">
                <h4 className="font-medium text-xl mb-2">Mesa</h4>
                <!-- Lista de mesas -->
                {mesasDisp.map((mesa) => (
                  <div
                    className="bg-white relative flex p-3 rounded-lg shadow-sm
cursor-pointer overflow-hidden mb-2"
                    onClick={() =>
                      setMesaSel((mesaSel) => (mesaSel != mesa ? mesa : null))
                    }
                    key={mesa.pkMesaId}
                  >
                    {mesaSel == mesa && (
                      <div className="absolute top-0 left-0 w-full h-full bg-
black/25"></div>
                    )}
                    <div className="flex flex-grow gap-x-3">
                      <div className="h-14 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                        <svg
                          xmlns="http://www.w3.org/2000/svg"
                          fill="none"
                          viewBox="0 0 24 24"
                          strokeWidth={1.5}
                          stroke="currentColor"
                          className="w-8"
                        >
                          <path
                            strokeLinecap="round"
                            strokeLinejoin="round"
                            d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3
7.5l9 5.25M3 7.5v9l9 5.25m0-9v9"
                          />
                        </svg>
                      </div>
                      <div className="flex items-center">
                        <h1 className="font-semibold text-xl">
                          {String(mesa.pkMesaId).padStart(4, "0")}
                        </h1>
                      </div>
                    </div>
                  </div>
                ))}
              </div>
              <div className="w-6/12">
                <h4 className="font-medium text-xl mb-2">Productos</h4>
                <!-- Lista productos -->
                {productosDisp.map((producto) => (
                  <div
                    className="flex p-3 rounded-lg shadow-sm bg-white mb-2"
                    key={`pro${producto.pkProductoId}`}
                  >
                    <div className="flex flex-grow gap-x-3">
                      <div className="h-14 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                        <svg
                          xmlns="http://www.w3.org/2000/svg"
                          fill="none"
                          viewBox="0 0 24 24"
                          strokeWidth={1.5}
                          stroke="currentColor"
                          className="w-8"
                        >
                          <path
                            strokeLinecap="round"
                            strokeLinejoin="round"
                            d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3
7.5l9 5.25M3 7.5v9l9 5.25m0-9v9"
                          />
                        </svg>
                      </div>
                      <div className="flex items-center">
                        <div className="">
```

```
                          <h1 className="font-semibold text-lg">
                            {producto.descripcion}
                          </h1>
                          <p className="text-base">{producto.valor}</p>
                        </div>
                      </div>
                    </div>
                    <div className="">
                      <div className="flex items-center">
                        <Count
                          minValue={0}
                          maxValue={producto.cantidad}
                          onChange={(value) => {
                            producto.solicitar = value;
                          }}
                        />
                      </div>
                      <span className="text-sm">
                        Máximo: {producto.cantidad}
                      </span>
                    </div>
                  </div>
                ))}
              </div>
            </div>
            <div className="flex justify-end">
              <button
                className="bg-blue-900 text-white inline-flex gap-x-2 rounded-full
py-3 px-5"
                onClick={() => createOrder()}
              >
                Crear
              </button>
            </div>
          </div>
        </Modal>
      )}
      <div className="flex justify-center">
        <div className="w-10/12 mb-6">
          <h1 className="text-3xl text-blue-900 font-semibold border-blue-900
border-b-2 mb-4 leading-relaxed">
            Pedidos
          </h1>
          <div className="flex justify-end">
            <button
              className="bg-blue-900 text-white inline-flex gap-x-2 rounded-full py-
3 px-5"
              onClick={() => setShowModal(true)}
            >
              Crear
              <svg
                xmlns="http://www.w3.org/2000/svg"
                fill="none"
                viewBox="0 0 24 24"
                strokeWidth="1.5"
                stroke="currentColor"
                className="w-6 h-6"
              >
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M12 4.5v15m7.5-7.5h-15"
                />
              </svg>
            </button>
          </div>
          <div className="grid grid-cols-2 gap-2 mt-4">
            <!-- Renderizar pedidos -->
            {pedidos.map((pedido) => (
              <div
                className="flex p-3 gap-x-2 items-center rounded-lg shadow-sm bg-
white"
                key={`ped${pedido.pkPedidoId}`}
              >
                <div className="h-20 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                  <svg
                    xmlns="http://www.w3.org/2000/svg"
                    fill="none"
                    viewBox="0 0 24 24"
                    strokeWidth={1.5}
                    stroke="currentColor"
                    className="w-10"
                  >
                    <path
                      strokeLinecap="round"
                      strokeLinejoin="round"
                      d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3 7.5l9
5.25M3 7.5v9l9 5.25m0-9v9"
                    />
                  </svg>
                </div>
                <div className="w-full">
                  <div className="flex items-center">
                    <h1 className="font-semibold text-xl">
                      {pedido.pkPedidoId}
                    </h1>
                  </div>
                  <div className="flex justify-between">
                    <h3>Mesa {pedido.fkMesaId}</h3>
                    <h3>Empleado: {pedido.fkEmpleadoId}</h3>
                  </div>
                </div>
              </div>
            ))}
          </div>
        </div>
      </div>
    </>
  );
}
```

Payments.jsx

Componente para realización de pagos por parte del usuario con acceso al mismo

```
function Payments() {
  // Hook para consulta de inventario
  const pagos = usePagoStore((state) => state.pagos);
  console.log(pagos);

  return (
    <>
      <div className="flex justify-center">
        <div className="w-10/12 mb-6">
          <h1 className="text-3xl text-blue-900 font-semibold border-blue-900
border-b-2 mb-4 leading-relaxed">
            Pagos
          </h1>
          <div className="grid grid-cols-2 gap-2 mt-4">
            <!-- Renderizado de pagos -->
            {pagos.map((pago) => (
              <div
                className="flex p-3 rounded-lg shadow-sm bg-white gap-x-2"
                key={`pay${pago.pkPagoId}`}
              >
                <div className="h-20 aspect-square bg-blue-900 rounded-md grid
place-items-center text-white">
                  <svg
                    xmlns="http://www.w3.org/2000/svg"
                    fill="none"
                    viewBox="0 0 24 24"
                    strokeWidth={1.5}
                    stroke="currentColor"
                    className="w-10"
                  >
                    <path
                      strokeLinecap="round"
                      strokeLinejoin="round"
                      d="M21 7.5l-9-5.25L3 7.5m18 0l-9 5.25m9-5.25v9l-9 5.25M3 7.5l9
5.25M3 7.5v9l9 5.25m0-9v9"
                    />
                  </svg>
                </div>
                <div className="flex items-center flex-1">
                  <div className="w-full">
```

```
                    <h1 className="font-semibold text-xl">{pago.pkPagoId}</h1>
                    <div className="w-10/12 flex justify-between">
                      <p className="">{pago.valor}</p>
                      <p>pedido: {pago.fkPedidoId}</p>
                    </div>
                  </div>
                </div>
              </div>
            ))}
          </div>
        </div>
      </div>
    </>
  );
}
```