

Laboratorio

Aplicación de uso de Redux con Redux Toolkit

1. Crear una nueva aplicación de React utilizando a Vite, asignarle el nombre demo9.

```
npm create vite@latest demo4 -- --template react-ts
```

2. Agregar al paquete de aplicación recién generado las siguientes dependencias. Recuerde restaurar las dependencias existentes y las agregadas.

```
npm install bootstrap bootstrap-icons axios react-router-dom react-redux @reduxjs/toolkit  
redux-logger
```

```
npm install @types/react-router-dom @types/react-redux @types/redux-logger --save-dev
```

3. Elimine el contenido de los archivos App.css e index.css. El contenido de los archivos debe quedar vacío.
4. Crear el archivo Header.tsx, en la carpeta src.
5. Cambie el contenido del archivo Header.tsx, por el siguiente.

```
import { Link, NavLink } from "react-router-dom";  
import { useSelector } from "react-redux";  
  
function Header() {  
  // Redux/Redux Toolkit  
  // Read from state on global store  
  // eslint-disable-next-line @typescript-eslint/no-explicit-any  
  const { items } = useSelector((state: any) => state.cart);  
  
  return (  
    <>  
    <nav className="navbar navbar-expand bg-light fs-5">  
      <div className="container-fluid">  
        <Link className="navbar-brand" to="/">  
          <i className="bi bi-house-door"></i>  
        </Link>  
        <ul className="navbar-nav me-auto">  
          <li className="nav-item">  
            <NavLink className="nav-link" to="/">  
              Home  
            </NavLink>  
          </li>  
        </ul>  
      </div>  
    </>  
  );  
}
```

```
      <li className="nav-item">
        <NavLink className="nav-link" to="/product">
          Products
        </NavLink>
      </li>
    </ul>
    <ul className="navbar-nav me-end">
      <li className="nav-item">
        <NavLink className="nav-link" to="/cart">
          <i className="bi bi-cart3"></i>
          <span className="badge rounded-pill bg-primary">
            {items.length}
          </span>
        </NavLink>
      </li>
    </ul>
  </div>
</nav>
</>
);
}
```

```
export default Header;
```

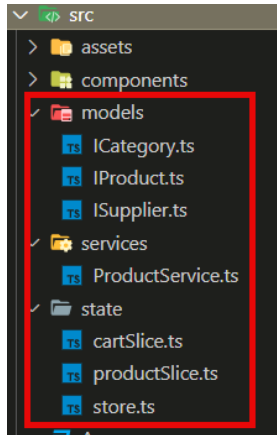
6. Cambie el contenido de componente App.tsx, por el siguiente.

```
import { Outlet } from "react-router-dom";
import "./App.css";
import Header from "./Header";

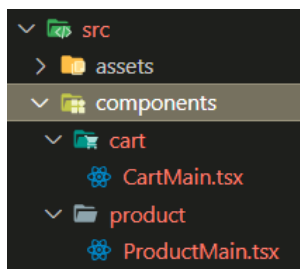
function App() {
  return (
    <>
      <Header />
      <div className="container" style={{marginTop: "2em"}}>
        <Outlet/>
      </div>
    </>
  );
}

export default App;
```

- Del contenido de archivos base facilitados por el profesor (<https://github.com/curiosumcoder/ct-react202502/blob/main/20250217/demo4-base.zip>), copie las carpetas models, services y state, y todo su contenido en la carpeta src de la aplicación de React actual.



- Crear los archivos para los componentes CartMain.tsx y ProductMain.tsx, en las ubicaciones de acuerdo con la imagen, crean las carpetas que sean necesarias.



- Cambie el contenido de componente CartMain.tsx, por el siguiente.

```
// Redux/Redux Toolkit
import { useDispatch, useSelector } from "react-redux";
import cartSlice from "../../state/cartSlice";
import IProduct from "../../models/IProduct";

function CartMain() {
  // Redux/Redux Toolkit
  // Read from state on global store
  // eslint-disable-next-line @typescript-eslint/no-explicit-any
  const { items } = useSelector((state: any) => state.cart);
  // Update state in global store
  const { clearCart, removeFromCart } = cartSlice.actions;
  // To execute actions on store
  // eslint-disable-next-line @typescript-eslint/no-explicit-any
  const dispatch = useDispatch<any>();
```

```
return (  
  <>  
  <h3>Cart</h3>  
  {items?.length > 0 && (  
    <>  
    <table className="table table-bordered">  
      <thead>  
        <tr>  
          <th>Name</th>  
          <th>Price</th>  
          <th></th>  
        </tr>  
      </thead>  
      <tbody>  
        {items.map((p: IProduct) => (  
          <tr key={p.id}>  
            <td>{p.productName}</td>  
            <td>{p.unitPrice}</td>  
            <td>  
              <i  
                className="bi bi-bag-dash-fill"  
                style={{ color: "red", cursor: "pointer" }}  
                onClick={() => dispatch(removeFromCart(p))}  
              ></i>  
            </td>  
          </tr>  
        ))}  
      </tbody>  
      <tfoot>  
        <tr>  
          <th>Total</th>  
          <td colSpan={2}>  
            {items.reduce((a: number, c: IProduct) => a + c.unitPrice, 0)}  
          </td>  
        </tr>  
      </tfoot>  
    </table>  
    <button  
      className="btn btn-secondary"  
      onClick={() => dispatch(clearCart())}  
    >  
      Clear cart  
    </button>  
  )  
}
```

```
    </>
  )}
  {items?.length === 0 && <p>Cart is empty!</p>}
</>
);
}
```

```
export default CartMain;
```

10. Cambie el contenido de componente ProductMain.tsx, por el siguiente.

```
/* eslint-disable @typescript-eslint/no-explicit-any */
// Redux/Redux Toolkit
import { useDispatch, useSelector } from "react-redux";
import { searchProducts } from "../../state/productSlice";
import IProduct from "../../models/IProduct";
import { useNavigate } from "react-router-dom";
import cartSlice from "../../state/cartSlice";

function ProductMain() {
  const navigate = useNavigate();

  // Redux/Redux Toolkit
  // Read from state on global store
  const { data, } = useSelector((state: any) => state.products);
  // Update state in global store
  // Update state in global store
  const { addToCart } = cartSlice.actions;
  // To execute actions on store
  const dispatch = useDispatch<any>();

  return (
    <>
      <h3>Product</h3>
      <input
        className="form-control"
        type="search"
        placeholder="Type here ..."
        onChange={(event) => dispatch(searchProducts(event.target.value))}
      />
      <br />
      {data.length > 0 && (
        <table className="table table-bordered table-striped">
          <thead>
```

```
        <tr>
          <th>Name</th>
          <th>Price</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        {data.map((p: IProduct) => (
          <tr key={p.id}>
            <td>{p.productName}</td>
            <td>{p.unitPrice}</td>
            <td>
              <i
                className="bi bi-bag-plus-fill"
                style={{ color: "blue", cursor: "pointer" }}
                onClick={() => {
                  dispatch(addToCart(p));
                  navigate("/cart");
                }}
              ></i>
            </td>
          </tr>
        ))}
      </tbody>
    </table>
  )}
</>
);
}
```

```
export default ProductMain;
```

11. Crear el archivo .env, en la carpeta creada para la aplicación React.
12. Cambie el contenido del archivo .env, por el siguiente.

```
VITE_API_URL=http://localhost:3000/products
```

13. Crear el archivo .env.production, en la carpeta creada para la aplicación React.
14. Cambie el contenido del archivo .env.production, por el siguiente.

```
VITE_API_URL=http://localhost:3000/products
```

15. Crear el archivo router.tsx, en la carpeta src.
16. Cambie el contenido del archivo router.tsx, por el siguiente.

```
import { createBrowserRouter } from "react-router-dom";
import App from "./App";
import Error from "./Error";
import ProductMain from "./components/product/ProductMain";
import CartMain from "./components/cart/CartMain";

const router = createBrowserRouter([
  {
    path: "/",
    element: <App />,
    errorElement: <Error />,
    children: [
      { index: true, element: (<><h1>Welcome ...</h1></>)},
      {
        path: "product",
        element: <ProductMain />,
      },
      {
        path: "cart",
        element: <CartMain />,
      },
    ],
  },
]);

export default router;
```

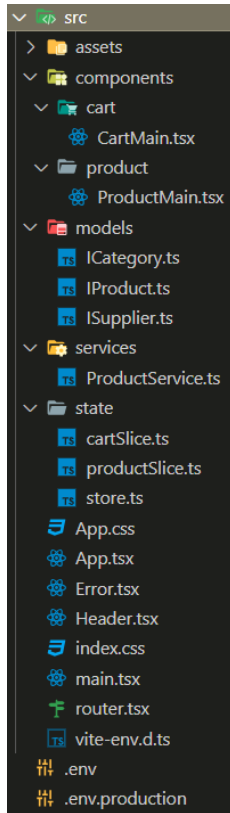
17. Cambie el contenido del archivo main.tsx, por el siguiente.

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import "bootstrap/dist/css/bootstrap.css";
import "bootstrap-icons/font/bootstrap-icons.css";
import router from "./router";
import { RouterProvider } from "react-router-dom";
import { Provider } from "react-redux";
import store from "./state/store";

ReactDOM.createRoot(document.getElementById("root")!).render(
  <React.StrictMode>
    <Provider store={store}>
      <RouterProvider router={router} />
    </Provider>
  </StrictMode>
);
```

```
</Provider>  
</React.StrictMode>  
);
```

18. La distribución de carpetas y archivos debería tener el siguiente aspecto.



19. Probar la ejecución de la aplicación, con especial atención al uso de Redux Toolkit en el código fuente en la carpeta **state**, y los componentes Header.tsx, ProductMain.tsx y CartMain.tsx.