



Electiva disciplinar II

Desarrollo de aplicaciones web

Unidad de aprendizaje II

Actividad 4 - Tecnología front end en la construcción de una aplicación web II

Hernán Ricardo Loaiza Doncel

Facultad de ingeniería

Docente tutor. Joaquín Sánchez

Curso: Electiva disciplinar

Ingeniería de software

Corporación Universitaria Iberoamericana

Bogotá – Colombia

Noviembre 30 de 2025

Desarrollo de la actividad

Introducción

El desarrollo de la aplicación web exige la integración de tecnologías que permitan construir interfaces interactivas y eficientes, ya que se conecta con el sistema backend mediante unos protocolos estandarizados. En el contexto de la actividad anterior que se desarrolla una propuesta, aplicación web para la gestión de pedidos e inventarios en un negocio de hamburguesería, ya que es necesario entender que las herramientas como ReactJS, hooks, la ContextAPI, la documentación con Swagger y las peticiones de HTTP con Axios, las rutas de navegación y el despliegue.

Esta actividad se desarrolla con estos conceptos desde una perspectiva práctica, acompañados de ejemplos aplicados al proyecto de la hamburguesería y se finaliza con un fragmento del código que integra frontend y backend.

Contenido

REST con Swagger

REST es un estilo arquitectónico que nos permite la comunicación entre cliente y servidor mediante operaciones HTTP como GET, POST, PUT y DELETE.

En el backend del proyecto de hamburguesería, REST permite gestionar pedidos, productos e inventarios a través de rutas bien definidas.

Swagger, ahora conocidos como el OpenAPI, se usa para documentar visualmente las APIs y nos permite su prueba directa desde una interfaz web.

Ejemplo aplicado al proyecto

paths:

/pedidos:

get:

summary: Obtener todos los pedidos

post:

summary: Crear un nuevo pedido

Con Swagger, el equipo puede visualizar y probar los endpoints, si la necesidad de herramientas externas.

ReactJS

Es una biblioteca de JavaScript se usa para construir interfaces de usuario mediante componentes reutilizables.

React se usa para construir pantallas como.

- Registro de los pedidos
- Lista de inventarios
- Reportes

Ejemplo aplicado

```
function PedidoCard({ pedido }) {
  return (
    <div>
      <h3>Pedido #{pedido.id}</h3>
      <p>Cliente: {pedido.cliente}</p>
      <p>Total: ${pedido.total}</p>
    </div>
  );
}
```

Hooks

useState

Nos permite manejar estados internos del componente.

```
const [cantidad, setCantidad] = useState(1);
```

useEffect

Nos permite ejecutar lógica cuando cambia el estado o el componente se monta.

```
useEffect(() => {
  console.log("Pedido actualizado");
}, [cantidad]);
```

useContext

Nos permite compartir información global como es el tema, usuario o inventario.

useReducer

ideal para manejar estados complejos, como agregar un adicional.

Ejemplo aplicado

```
const reducer = (state, action) => {
  switch(action.type){
```

```

    case "AGREGAR":
      return [...state, action.payload];
    }
  }
}

```

Context API

Sirve para manejar un estado global sin necesidad de pasar propiedades por muchos niveles.

Ejemplo aplicado al inventario

```

export const InventarioContext = createContext();

export function InventarioProvider({ children }) {
  const [inventario, setInventario] = useState([]);

  return (
    <InventarioContext.Provider value={{ inventario, setInventario }}>
      {children}
    </InventarioContext.Provider>
  );
}

```

Peticiones HTTP con Axios

Axios nos permite consumir APIs de forma simple.

Ejemplo aplicado a obtener pedidos

```

import axios from "axios";

export const getPedidos = async () => {
  const respuesta = await axios.get("http://localhost:4000/pedidos");
}

```

```
    return respuesta.data;
  };

```

Rutas y navegación

Para manejar secciones dentro la aplicacion se usa **React Router DOM**.

Ejemplo

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Home from "../pages/Home";
import Inventario from "../pages/Inventario";

```

```
function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/inventario" element={<Inventario />} />
      </Routes>
    </BrowserRouter>
  );
}

```

```
export default App;

```

Despliegue

Esta parte consiste en publicar la aplicación para que los usuarios puedan usarla desde internet y para eso tenemos varias opciones comunes como:

- ✓ Vercel
- ✓ Netlify
- ✓ GitHub pages
- ✓ Railway "backend"
- ✓ Render "para el backend"

Para desplegar este proyecto se podría de la siguiente manera

- ✓ Frontend: Vercel
- ✓ Backend: Render
- ✓ Repositorio: GitHub

Codigo de la aplicación web

Este es un codigo simplificado que integra React, Axios, hooks y contexto, aplicado al módulo de pedidos.

Backend

```
app.get("/pedidos", async (req, res) => {  
  const pedidos = await PedidoModel.find();  
  res.json(pedidos);  
});  
  
app.post("/pedidos", async (req, res) => {  
  const nuevo = await PedidoModel.create(req.body);  
  res.json(nuevo);  
});
```

Frontend (mostrar pedidos)

```
import { useEffect, useState } from "react";  
  
import axios from "axios";
```

```

export default function ListaPedidos() {
  const [pedidos, setPedidos] = useState([]);

  useEffect(() => {
    axios.get("http://localhost:4000/pedidos")
      .then(res => setPedidos(res.data))
      .catch(err => console.log(err));
  }, []);

  return (
    <div>
      <h2>Pedidos registrados</h2>
      {pedidos.map(p => (
        <div key={p.id} className="card">
          <h3>Pedido #{p.id}</h3>
          <p>Cliente: {p.cliente}</p>
          <p>Total: ${p.total}</p>
        </div>
      ))}
    </div>
  );
}

```


Bibliografia

Regla, P. D. (2014). Diseño, contenidos y desarrollo del front-end del sitio web del proyecto auralizarte. [Trabajo de grado, Universidad Pública de Navarra].

Valdivia Caballero, J. J. (2021). Modelo de procesos para el desarrollo del front-end de aplicaciones web. [Tesis de maestría, Universidad Nacional Mayor de San Marcos]. (pp. 5-40).

Martínez Martínez, A. (2021). Proyecto feedback backend y frontend web. [Trabajo de grado, Universitat Jaume].