



Título del informe

Análisis y diseño de una aplicación web

Asignatura

Desarrollo Web Avanzado

Nombre del autor o autores

Jeffrey Manobanda – Jhordy Marcillo

Departamento

Ciencias de la computación

Universidad

Universidad de las Fuerzas Armadas ESPE

Fecha de entrega

24 de enero de 2025

Contenido

Resumen	3
Introducción	3
Objetivos	3
Desarrollo o Cuerpo del Informe	4
Resultados	8
Conclusiones	8
Recomendaciones	9
Bibliografía o Referencias.....	9

Resumen

En este informe se explicará el proceso que se siguió para realizar el desarrollo de una aplicación web para la gestión de inventarios, implementando la metodología MVC (Modelo-Vista-Controlador). La aplicación nos permite realizar operaciones CRUD sobre productos, además de tener funciones como búsqueda, filtrado, paginación y edición con el uso de componentes PrimeNG.

Introducción

- **Objetivo del informe:**

El objetivo de este informe es explicar el proceso de desarrollo de una aplicación que controle la gestión de inventarios de productos implementando la metodología MVC y la utilización de componentes PrimeNG.

- **Contexto:**

La gestión de inventarios es una necesidad para las empresas. Implementar una aplicación que permita gestionar productos de forma eficiente puede optimizar procesos, reducir errores y mejorar la toma de decisiones. Este proyecto se desarrolla en el marco de la asignatura Desarrollo Web Avanzado, aplicando conceptos para lograr una interfaz limpia y eficiente.

- **Alcance y limitaciones:**

Alcance: El sistema permite realizar operaciones CRUD sobre productos con funciones de categorización, filtrado, paginación y edición mediante cuadros de diálogo.

Limitaciones: La aplicación no incluye una base de datos donde se pueda guardar la información que se genere solamente se utilizan listas declaradas.

Objetivos

- **General**

Diseñar una aplicación que permita la gestión de inventarios de productos con la implementación de la metodología MVC y componentes de PrimeNG.

- **Específicos**

- Incorporar funcionalidades de búsqueda, filtrado y paginación de los productos.
- Diseñar una interfaz una interfaz limpia y eficiente.
- Facilitar la edición y creación de productos mediante cuadros de diálogo.

Desarrollo o Cuerpo del Informe

• Conceptos base:

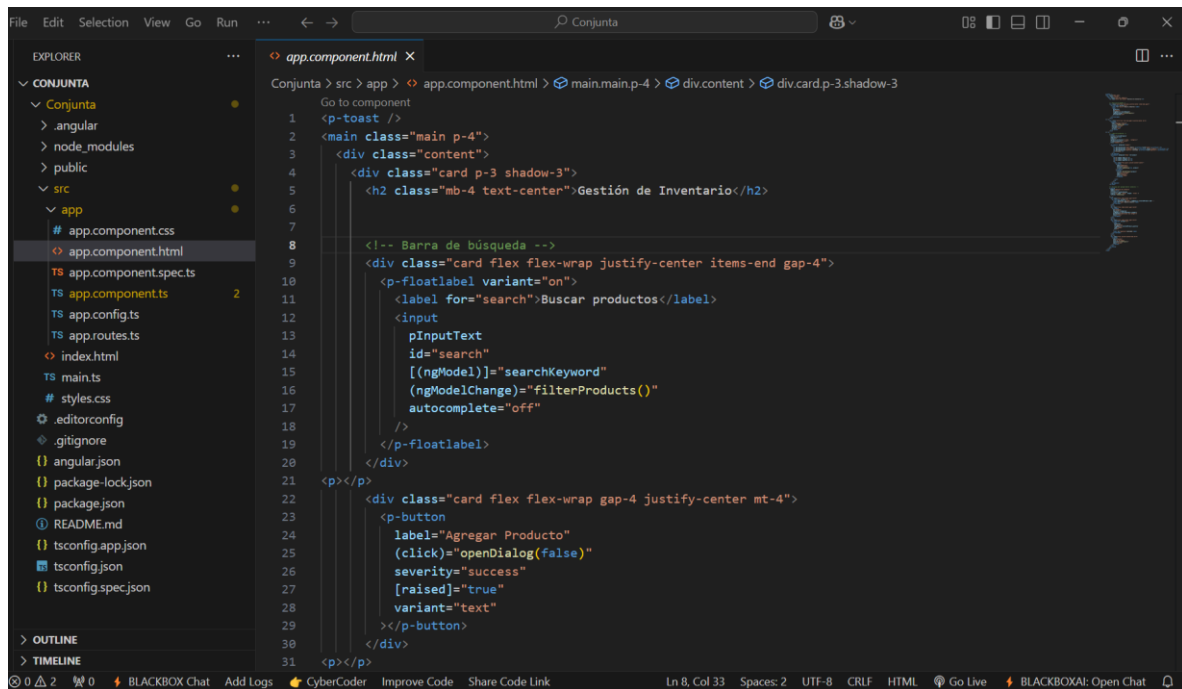
La metodología MVC nos permite dividir una aplicación en tres componentes principales: Modelo, Vista y Controlador. Esto permite una separación clara entre la lógica de negocio, la interfaz de usuario y el control del flujo de datos.

La gestión de un inventario nos permite leer, crear, actualizar y borrar productos de manera eficiente.

• Herramientas de desarrollo:

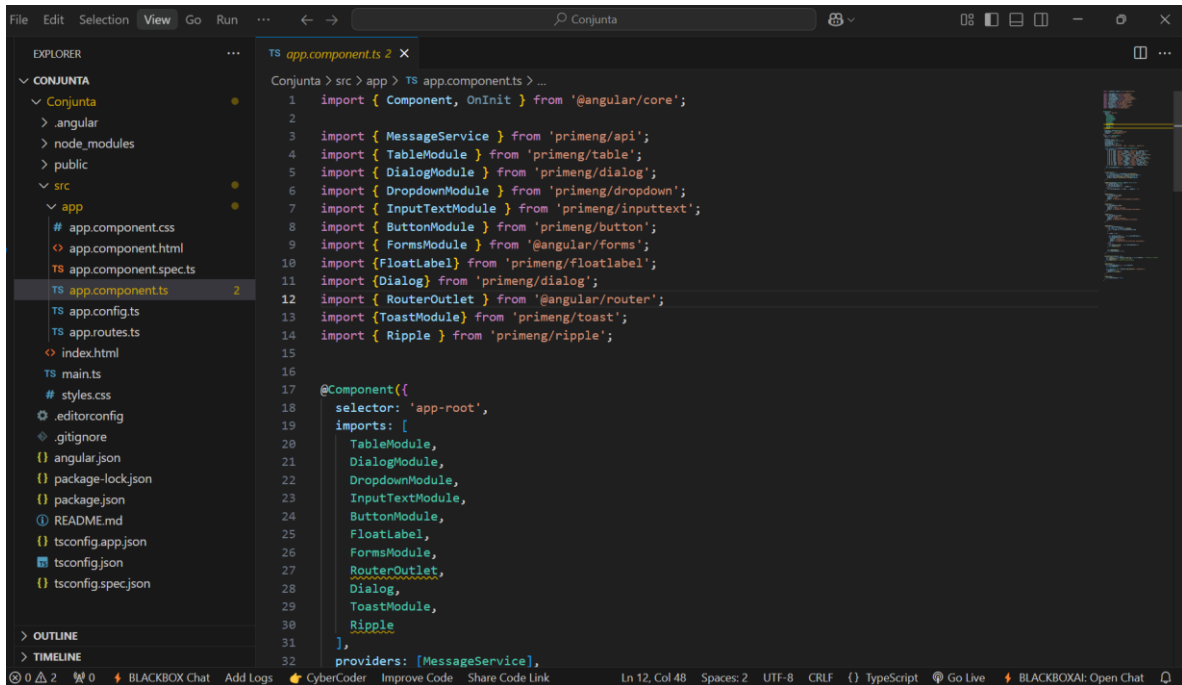
- **Angular:** Es un framework de desarrollo frontend que permite crear aplicaciones dinámicas con una arquitectura basada en componentes.
- **PrimeNG:** Es una biblioteca de componentes de interfaz de usuario para Angular. Incluye elementos como tablas, diálogos, menús, y formularios, diseñados para mejorar la experiencia del usuario.
- **Node.js:** Node.js es un entorno de ejecución de JavaScript en el servidor.
- **Express:** Express es un framework para Node.js que facilita la creación de APIs RESTful y la gestión de rutas.
- **Visual Studio Code:** Es un editor de código fuente con soporte para extensiones y herramientas de depuración.
- **Git:** Es un sistema de control de versiones que permite gestionar el historial de cambios en el código, facilitando la colaboración y la recuperación de versiones anteriores.

• Cuerpo o Desarrollo:



```
1 <p-toast />
2 <main class="main p-4">
3   <div class="content">
4     <div class="card p-3 shadow-3">
5       <h2 class="mb-4 text-center">Gestión de Inventario</h2>
6     </div>
7
8     <!-- Barra de búsqueda -->
9     <div class="card flex flex-wrap justify-center items-end gap-4">
10      <p-floatlabel variant="on">
11        <label for="search">Buscar productos</label>
12        <input
13          pInputText
14          id="search"
15          [(ngModel)]="searchKeyword"
16          (ngModelChange)="filterProducts()"
17          autocomplete="off"
18        />
19      </p-floatlabel>
20    </div>
21  </div>
22
23  <div class="card flex flex-wrap gap-4 justify-center mt-4">
24    <p-button
25      label="Agregar Producto"
26      (click)="openDialog(false)"
27      severity="success"
28      [raised]="true"
29      variant="text"
30    ></p-button>
31  </div>
32</main>
33</p></p>
```

- Se utilizó p-toast para mostrar notificaciones de información al usuario sobre acciones realizadas como producto agregado, actualizado o eliminado.
- El componente p-floatlabel de PrimeNG se utilizó para mostrar la barra de búsqueda.
- Se llamo al método de filtrado de productos mediante ngModelChange ejecutándolo cada vez que el usuario escribe, filtrando los productos mostrados.
- Se utilizó p-button para agregar el botón de PrimeNG configurado con el evento openDialog para agregar un nuevo producto
- Se utilizó el componente p-table de PrimeNG donde se muestra la lista filtrada de los productos con 7 filas por página donde también se implementó el filtrado mediante nombre y categoría y se tiene los botones para modificación y eliminación de algún producto.
- p-dialog es un componente modal de PrimeNG que nos permite controlar si existe algún dialogo visible o no.
- En esta parte es donde se asignaron las funcionalidades de leer, crear, actualizar, eliminar, buscar y filtrar de los productos.



```
1 import { Component, OnInit } from '@angular/core';
2
3 import { MessageService } from 'primeng/api';
4 import { TableModule } from 'primeng/table';
5 import { DialogModule } from 'primeng/dialog';
6 import { DropdownModule } from 'primeng/dropdown';
7 import { InputTextModule } from 'primeng/inputtext';
8 import { ButtonModule } from 'primeng/button';
9 import { FormsModule } from '@angular/forms';
10 import { FloatLabel } from 'primeng/floatlabel';
11 import { Dialog } from 'primeng/dialog';
12 import { RouterOutlet } from '@angular/router';
13 import { ToastModule } from 'primeng/toast';
14 import { Ripple } from 'primeng/ripple';
15
16
17 @Component({
18   selector: 'app-root',
19   imports: [
20     TableModule,
21     DialogModule,
22     DropdownModule,
23     InputTextModule,
24     ButtonModule,
25     FloatLabel,
26     FormsModule,
27     RouterOutlet,
28     Dialog,
29     ToastModule,
30     Ripple
31   ],
32   providers: [MessageService],
33 })
```

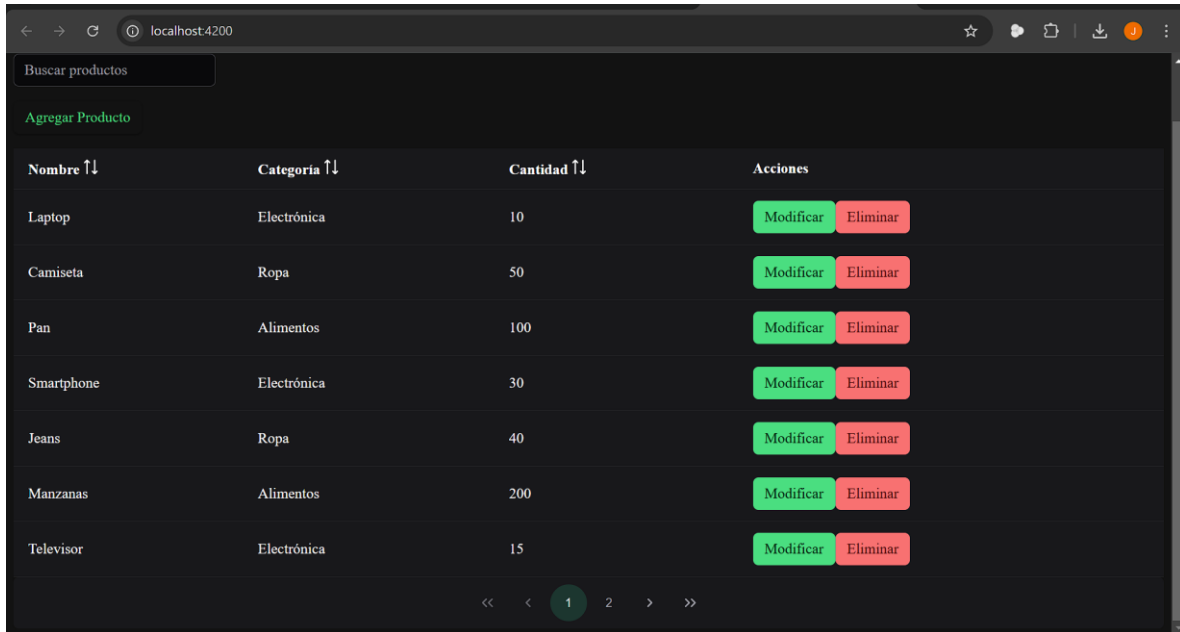
En este componente se realizó las importaciones de los módulos de PrimeNG para proporcionar funcionalidades de:

- TableModule: Para manejar tablas de datos con funcionalidades como paginación y ordenamiento.
- DialogModule: Para mostrar cuadros de diálogo modales.
- DropdownModule: Para menús desplegables.
- InputTextModule: Para entradas de texto.
- ButtonModule: Para botones estilizados.
- ToastModule: Para mostrar notificaciones.
- Ripple: Para efectos visuales interactivos.
- xFormsModule: Para el enlace de datos bidireccional con ngModel.

Además, se tiene las propiedades principales como:

- filterProducts(): Filtra los productos según la palabra clave ingresada en el cuadro de búsqueda.
 - saveProduct(): Guarda el producto en el inventario.
 - deleteProduct(product: any): Elimina el producto seleccionado de la lista.
 - updateTable(): Actualiza la tabla de productos.
 - showSuccess(): Muestra un mensaje de éxito al agregar un producto.
 - showDelete(): Notifica la eliminación de un producto.
 - showError(): Muestra un mensaje de advertencia en caso de error.
- Diseño del sistema
 - Modelo: Incluye la representación de los productos, con atributos como nombre, categoría, precio y stock.

- Vista: Se implementó con componentes PrimeNG como DataTable, Dialog, Dropdown y InputText.
- Controlador: Maneja la lógica de negocio, realizando operaciones CRUD y actualizando la vista en tiempo real.

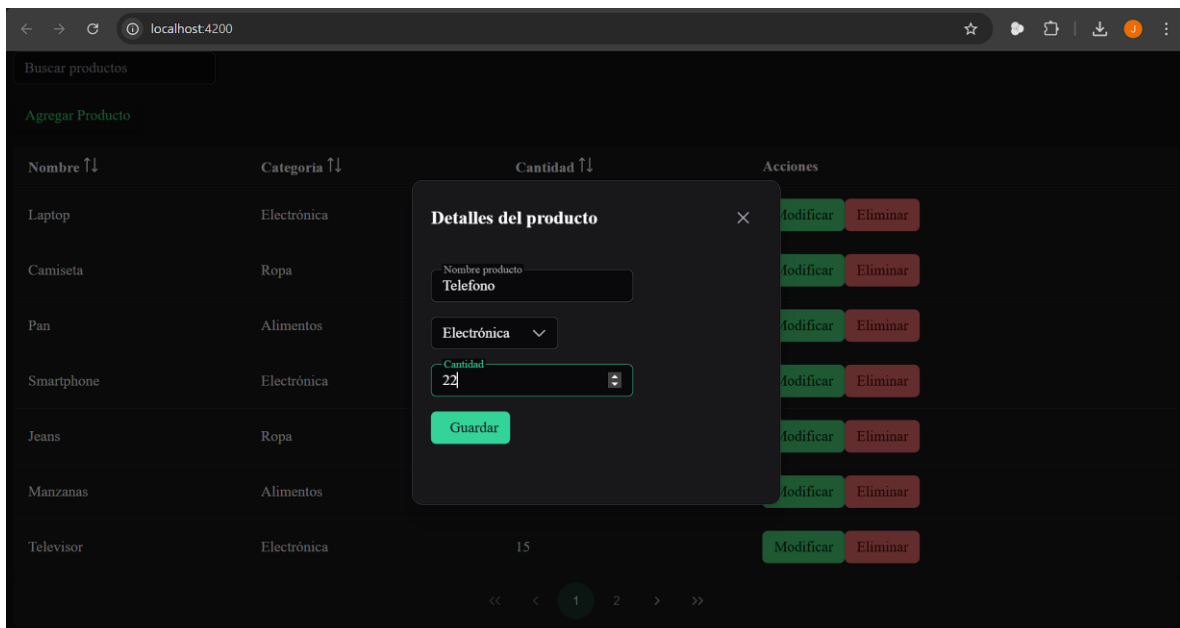


Buscar productos

Agregar Producto

Nombre ↑↓	Categoría ↑↓	Cantidad ↑↓	Acciones
Laptop	Electrónica	10	Modificar Eliminar
Camiseta	Ropa	50	Modificar Eliminar
Pan	Alimentos	100	Modificar Eliminar
Smartphone	Electrónica	30	Modificar Eliminar
Jeans	Ropa	40	Modificar Eliminar
Manzanas	Alimentos	200	Modificar Eliminar
Televisor	Electrónica	15	Modificar Eliminar

<< < 1 2 > >>



Buscar productos

Agregar Producto

Nombre ↑↓	Categoría ↑↓	Cantidad ↑↓	Acciones
Laptop	Electrónica		Modificar Eliminar
Camiseta	Ropa		Modificar Eliminar
Pan	Alimentos		Modificar Eliminar
Smartphone	Electrónica		Modificar Eliminar
Jeans	Ropa		Modificar Eliminar
Manzanas	Alimentos		Modificar Eliminar
Televisor	Electrónica	15	Modificar Eliminar

<< < 1 2 > >>

Detalles del producto

Nombre producto
Telefono

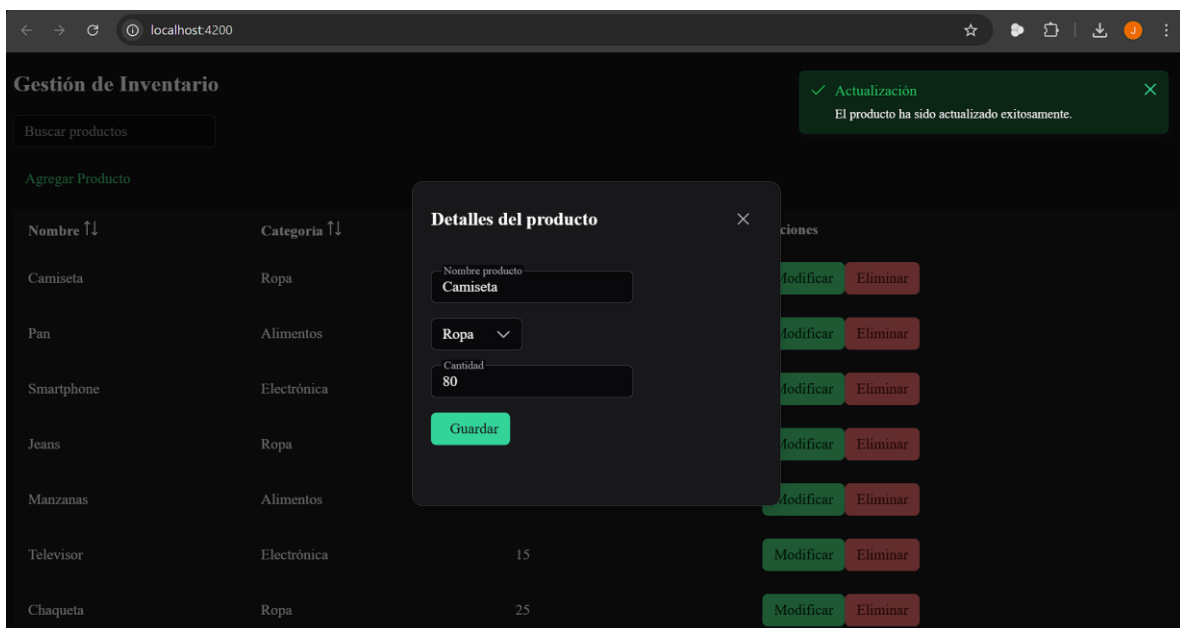
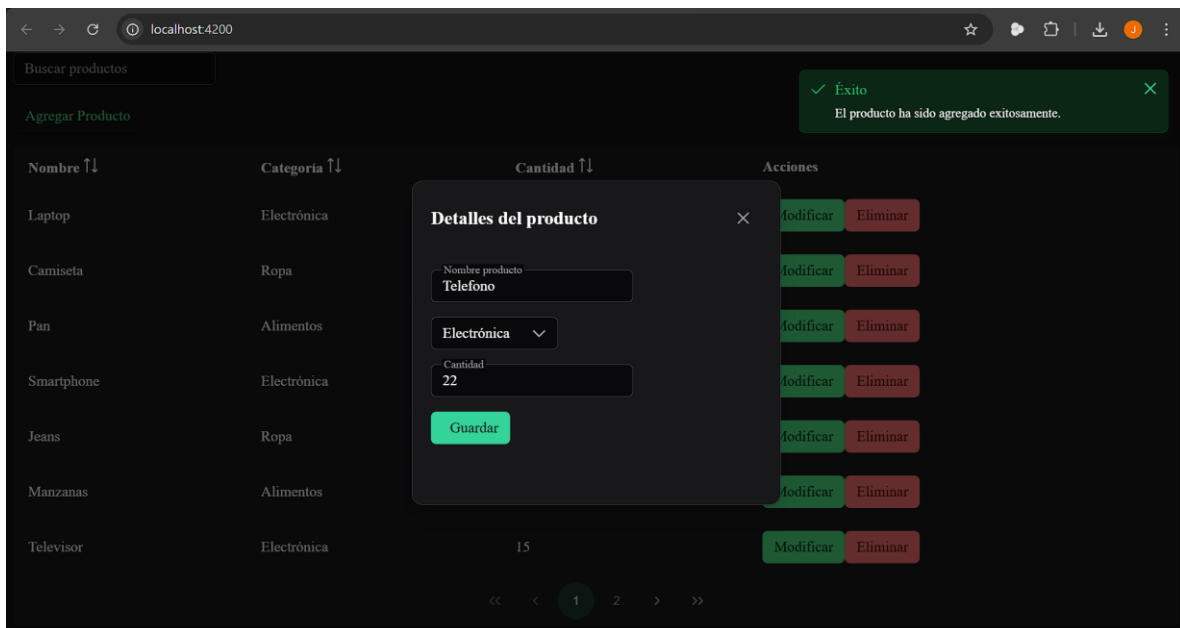
Categoría
Electrónica

Cantidad
22

Guardar

- Implementación de funcionalidades
 - CRUD: Operaciones completas de creación, lectura, actualización y eliminación.
 - Filtrado y búsqueda: Implementados en DataTable, permitiendo buscar productos por nombre o categoría.
 - Paginación: Configurada para manejar grandes volúmenes de datos.

- Cuadros de diálogo: Usados para la edición y creación de productos.



Resultados

La aplicación desarrollada cumple con las funcionalidades pedidas, ofreciendo una gestión de inventarios eficiente, en donde se demuestra la implementación de operaciones CRUD y también de las funcionalidades de búsqueda, filtrado y paginación implementadas correctamente.

Conclusiones

- La capacidad de buscar y filtrar productos permite a los usuarios encontrar información relevante de manera rápida y sencilla.
- Una interfaz bien diseñada permite a los usuarios interactuar fácilmente con el sistema sin confusión.
- Los cuadros de diálogo permiten a los usuarios editar o crear productos sin abandonar la vista principal, reduciendo interrupciones en el flujo de trabajo.

Recomendaciones

- Incluir la implementación de una base de datos para tener los datos ingresados como un respaldo para un futuro uso.
- Con las funcionalidades implementadas se puede implementar una base de datos extensa ya que de esta manera se puede encontrar la información que se necesita de manera rápida.

Bibliografía o Referencias

- <https://primeng.org/>
- <https://angular.dev/>
- <https://nodejs.org/en>