

**Guía práctica de Talleres No Proyecto:**

**Manual del Desarrollador**

**Tienda El Viejo**

**Por:**

**Carlos Andres Pardo Salinas**

**Mattias Henao Ricaurte**

**Juan Esteban Giraldo**

**Carlos Andres Alvarez**

**Para:**

**Manolo Pájaro Borrás**

**Politécnico Colombiano Jaime Isaza Cadavid Medellín, Colombia**

**2023**

pág. 1

# Contenido

[Introducción 3](#_bookmark0)

Implementación de base de datos [4](#_bookmark1)

[Requisitos del Software 5](#_bookmark2)

[Diseño del software del sistema 6](#_bookmark3)

[Desarrollo del aplicativo 7](#_bookmark4)

[LOGIN 7](#_bookmark5)

MAIN……………………………………………………………………………………………………………………………..8

[MAIN](#_bookmark7)…………………………………………………………………………………………… 9

[BACK END MAIN](#_bookmark8) 10

[CARRITO 1](#_bookmark9)1

CONFIGURACION DEL ANGULAR 12

SERVIDOR DEL PROGRAMA 13

Conclusiones……………………………………………………………………………………………………………….14

pág. 2

# Introducción

### Objetivo del Programa

El programa de desarrollo de Tienda El Viejo tiene como objetivo principal crear una plataforma integral y dinámica que ofrezca una amplia gama de productos deportivos. Esta plataforma está diseñada para satisfacer las necesidades de atletas, aficionados y personas interesadas en el deporte, proporcionando una experiencia de compra eficiente, personalizada y accesible.

### Descripción General

El programa se enfoca en la creación de una tienda de deportes que no solo incluye una variedad de productos de alta calidad, sino que también ofrece funcionalidades avanzadas para mejorar la experiencia del usuario. Entre los productos disponibles se encuentran ropa deportiva, calzado, equipos para diferentes disciplinas deportivas, y accesorios relacionados.

### Características Clave

1. **Catálogo Extenso y Actualizado**: La tienda contará con un catálogo variado que se actualizará regularmente para incluir las últimas novedades y tendencias en el mundo del deporte.
2. **Interfaz Intuitiva**: Una interfaz de usuario diseñada para ser intuitiva y fácil de navegar, permitiendo a los clientes encontrar y adquirir productos de manera rápida y sencilla.
3. **Personalización**: Funcionalidades que permiten a los usuarios recibir recomendaciones personalizadas basadas en sus preferencias y comportamientos de compra anteriores.
4. **Seguridad y Confiabilidad**: Implementación de medidas de seguridad robustas para proteger la información personal y los datos de pago de los clientes.
5. **Soporte al Cliente**: Un sistema de atención al cliente eficiente y accesible para resolver cualquier consulta o problema que los usuarios puedan tener.

### Beneficios Esperados

* **Comodidad y Accesibilidad**: Los clientes podrán realizar sus compras desde la comodidad de sus hogares, con la posibilidad de acceder a una amplia gama de productos desde cualquier dispositivo con conexión a internet.
* **Mejora en la Experiencia de Compra**: Gracias a la personalización y la interfaz intuitiva, los usuarios disfrutarán de una experiencia de compra agradable y sin complicaciones.
* **Crecimiento del Negocio**: La tienda online permitirá llegar a un público más amplio, lo que se traducirá en un incremento de las ventas y el reconocimiento de la marca.

pág. 3

# Implementación de base de datos:

El proyecto incorpora tecnologías de vanguardia para asegurar un rendimiento óptimo y una gestión eficiente de los datos. En cuanto a la base de datos, se utiliza MongoDB, una base de datos NoSQL conocida por su capacidad para almacenar grandes volúmenes de información de manera flexible y escalable. MongoDB permite manejar una variedad de datos que son esenciales para una tienda de deportes, tales como inventarios de productos, perfiles de clientes, historiales de compra, y reseñas de usuarios. Su estructura basada en documentos facilita la rápida adaptación a los cambios en los requisitos de datos y permite una integración sencilla con otros componentes del sistema.

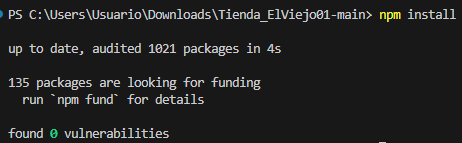
Por otro lado, la plataforma está desarrollada con Angular, un robusto framework de desarrollo front-end mantenido por Google. Angular es ideal para la creación de aplicaciones web de una sola página (SPA), proporcionando una experiencia de usuario dinámica y altamente interactiva. Con Angular, se garantiza que la interfaz de la tienda sea rápida, responsiva y fácil de usar, permitiendo a los usuarios navegar y realizar compras sin problemas desde cualquier dispositivo. Además, Angular facilita el desarrollo modular y el mantenimiento del código, lo que se traduce en una mayor eficiencia y calidad en el desarrollo del proyecto.

La combinación de MongoDB y Angular no solo mejora la capacidad de gestión de datos y la experiencia del usuario, sino que también asegura que la plataforma pueda escalar y adaptarse a las futuras necesidades del negocio. Esto resulta en una tienda de deportes robusta, eficiente y preparada para ofrecer un servicio de primer nivel a sus usuarios.

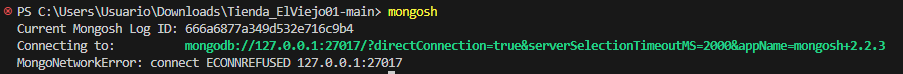
pág. 4

# Requisitos del Software

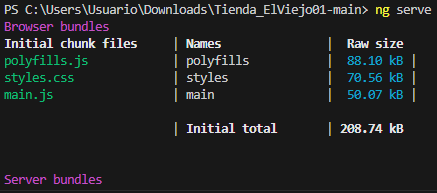
1. Descarga del angular y su ejecución en la terminal del sistema.



1. Ejecución del app components desde el local host.
2. Guardar los datos en la base de datos de Mongo y conectar el servidor.



1. Servidor de Angular instalado.



1. Correr el programa en el puerto correcto.

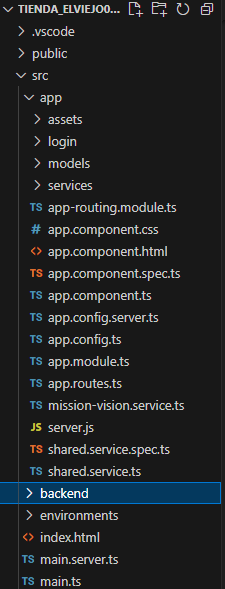
pág. 5

# Desarrollo del software del sistema

El proyecto ha sido desarrollado bajo la lógica HTML donde se corre desde el app components ubicado en la parte de la app del programa.

## Estructura del código

El código cuenta con diferentes carpetas que desarrollan diferentes funciones para que el código sea óptimo y eficiente.



**Referencias de los paquetes y librerías:**

El programa cuenta con diferentes archivos Json que guardan y hacen llamado a las aplicaciones que permiten el desarrollo del programa en el local host como por ejemplo: Angular que implementa toda la lógica para que el programa cuente con la posibilidad de que toda la página pueda funcionar y aplicar todos sus métodos en l misma página donde se ejecuta su código.

pág. 6

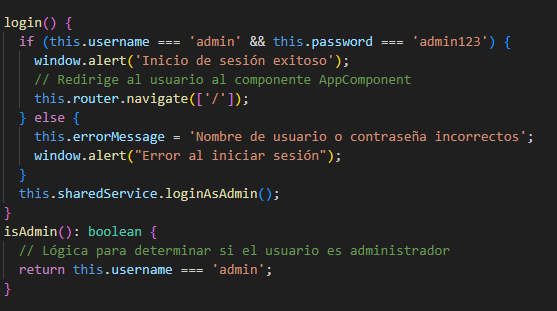
# Desarrollo del aplicativo

## LOGIN

Este formulario de inicio de sesión en Angular permite a los usuarios ingresar su nombre de usuario y contraseña. Utiliza la directiva [(ngModel)] para la vinculación bidireccional de los datos del formulario con las propiedades del componente, y el evento (ngSubmit) para manejar la lógica de inicio de sesión cuando se envía el formulario.

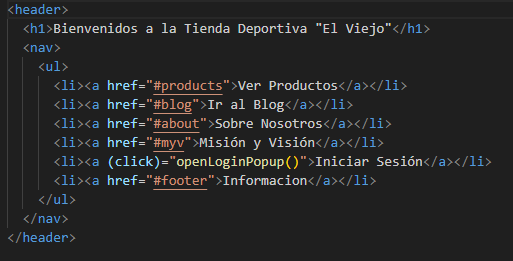


Este componente maneja el proceso de inicio de sesión, verificando las credenciales del usuario y proporcionando navegación y manejo de errores. También interactúa con servicios compartidos y utiliza características de Angular como inyección de dependencias y eventos de salida.

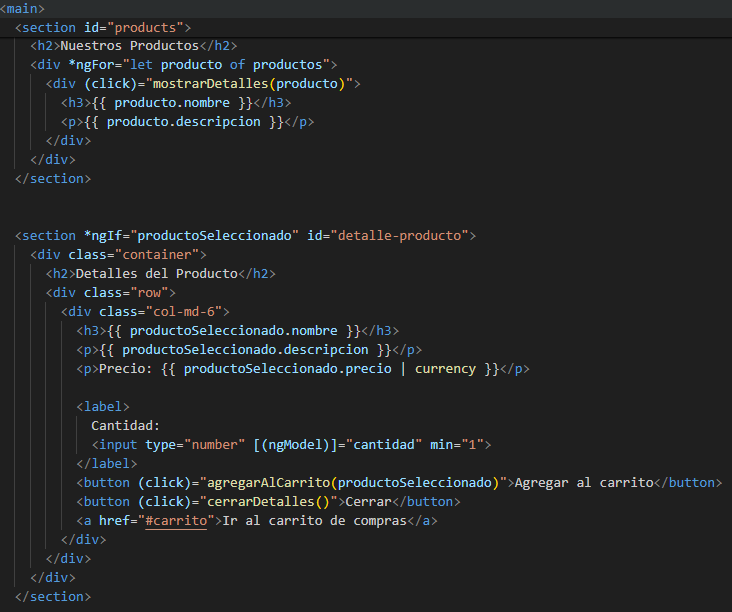
****

pág. 7

**MAIN**

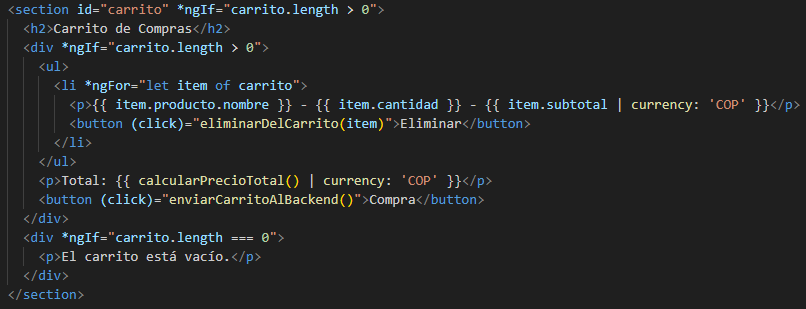


En esta parte del código se evidencia las secciones de la página web que se implementa en la parte del header en la sección del código.



En esta parte del código se implementa la lógica de los productos que se ofrecen en la página y como estos se agregan cantidades específicas y según el interés del usuario se agregan al carrito para su futura compra.

pág. 8



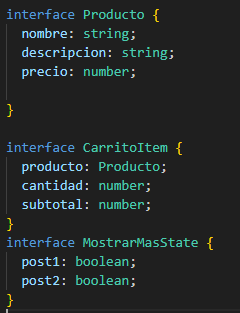
La parte de la implementación del carrito permite que el usuario escoja sus productos favoritos para así sumar la cantidad y sacar el total de la compra.



La parte del blog dela página nos permite hacer un poco de publicidad de la página utilizando diferentes métodos de adición para que la página implemente la funcionalidad de mostrar cada publicidad.

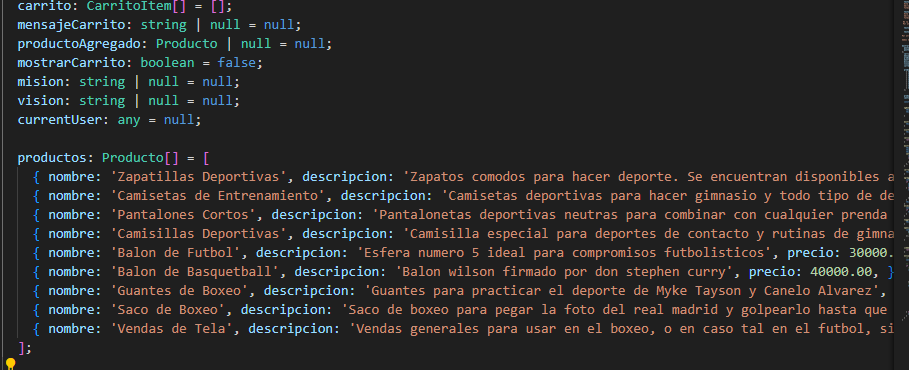
**BACK END DEL MAIN**

pág. 9



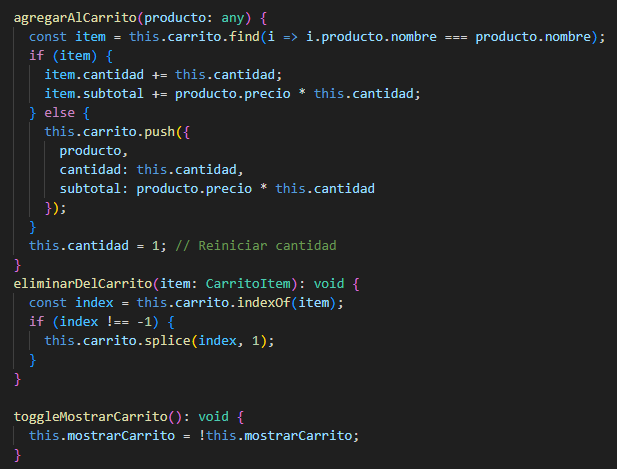
* **Producto**: Define la estructura de un producto con nombre, descripción y precio.
* **CarritoItem**: Define la estructura de un ítem en el carrito de compras, asociando un producto con una cantidad y un subtotal.
* **MostrarMasState**: Define la estructura del estado para controlar la visibilidad de información adicional en dos publicaciones distintas.

Estas interfaces son útiles para asegurar que los datos manejados en la aplicación tengan una estructura consistente y clara.



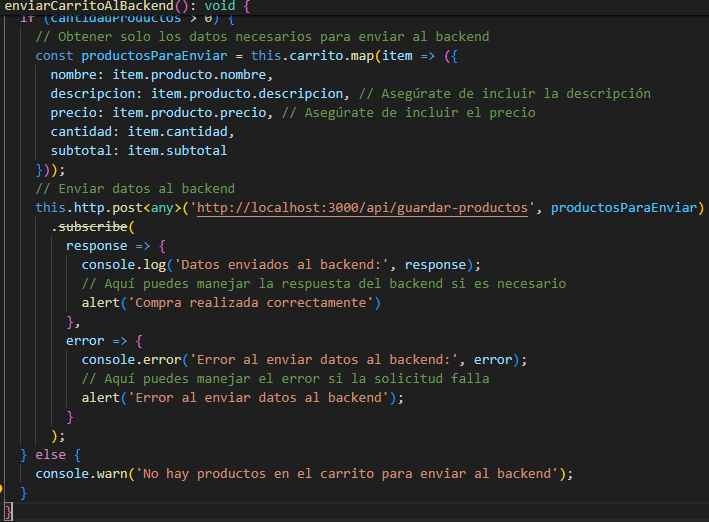
El código define un componente que maneja un carrito de compras y una lista de productos en una tienda de deportes. Las propiedades almacenan información sobre el estado del carrito, los productos disponibles, y otras características como la misión y visión de la tienda y la información del usuario actual. La lista de productos contiene varios artículos deportivos con sus descripciones y precios.

pág. 10



* **agregarAlCarrito**: Este método gestiona la adición de productos al carrito, actualizando las cantidades y los subtotales si el producto ya existe, o añadiendo un nuevo ítem si es nuevo.
* **eliminarDelCarrito**: Este método elimina un ítem específico del carrito.
* **toggleMostrarCarrito**: Este método alterna la visibilidad del carrito, permitiendo mostrar u ocultar el contenido del mismo según el estado actual.

Estos métodos permiten gestionar de manera efectiva el contenido del carrito de compras, proporcionando funcionalidades clave para agregar, eliminar y visualizar productos en una aplicación de tienda en línea.

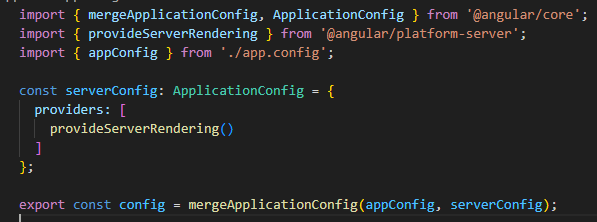


pág. 11

El método enviarCarritoAlBackend() realiza los siguientes pasos:

* Verifica si hay productos en el carrito.
* Prepara los datos del carrito en un formato adecuado para el backend.
* Envía los datos al backend mediante una solicitud HTTP POST.
* Maneja la respuesta del backend mostrando mensajes de éxito o error según corresponda.
* Emite una advertencia si el carrito está vacío y no se realiza ninguna acción adicional.

**CONFIGURACION DEL ANGULAR**



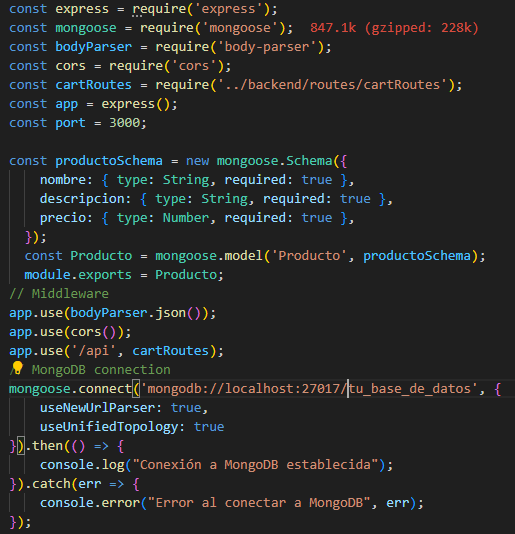
* **Importaciones**: Importa funciones y configuraciones necesarias desde Angular y archivos locales.
* **Configuración del Servidor**: Define serverConfig con provideServerRendering() para habilitar el renderizado en el servidor.
* **Combinar Configuraciones**: Combina appConfig (configuración principal de la aplicación) y serverConfig (configuración del servidor) utilizando mergeApplicationConfig.
* **Exportación**: Exporta la configuración combinada como config, que puede ser utilizada por la aplicación Angular.

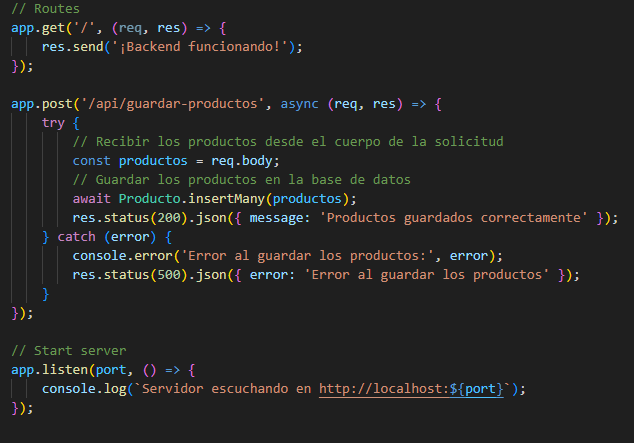
Este código es parte de la configuración necesaria para habilitar y gestionar el server-side rendering en una aplicación Angular, proporcionando una experiencia mejorada en términos de rendimiento

**SERVIDOR DEL PROGRAMA**

El código configura un servidor backend usando Express.js para manejar solicitudes HTTP y conectarse a una base de datos MongoDB. También define un esquema de productos y una ruta para guardar productos en la base de datos.

pág. 12

****

****

El código configura un servidor Express que:

* Conecta a una base de datos MongoDB.
* Define un modelo para productos.
* Proporciona middleware para analizar JSON y manejar CORS.
* Define rutas para manejar la solicitud GET raíz y una solicitud POST para guardar productos en la base de datos.
* Inicia el servidor en el puerto 3000 y confirma su estado mediante mensajes en la consola.

.

pág. 13

# Conclusiones

 La Tienda El Viejo" combina de manera efectiva las tecnologías frontend y backend para crear una experiencia de usuario coherente y fluida. Los usuarios pueden navegar por productos, añadirlos al carrito, y realizar compras con facilidad, gracias a la integración de Angular, Express, y MongoDB.

 La lógica implementada en HTML facilita la interacción directa y reactiva con la interfaz de usuario, mientras que el backend maneja las operaciones críticas y la persistencia de datos de manera robusta y segura.

"La Tienda El Viejo" demuestra ser una plataforma moderna y eficiente para la gestión de una tienda de deportes en línea. Al integrar Angular para una interfaz de usuario dinámica, MongoDB para una gestión de datos robusta, y lógica en HTML para una interacción directa, el programa ofrece una solución completa y escalable que puede satisfacer las necesidades tanto de los usuarios finales como de los administradores del sistema.

El programa "La Tienda El Viejo" se presenta como una solución integral y moderna para la gestión de una tienda de deportes en línea. Este proyecto combina la potencia y flexibilidad de Angular para el frontend, la robustez de MongoDB para la gestión de datos, y lógica implementada directamente en HTML para ofrecer una experiencia de usuario dinámica y reactiva.

pág. 14