

Manual Técnico - Sistema de Comercio Electrónico

Versión: 1.0

Fecha de Emisión: 16 de septiembre de 2023

Índice

Introducción.....	3
Herramientas utilizadas para el desarrollo.....	3
Esquema Conceptual	4
Esquema Lógico	4
Esquema Físico	6
Descripción de las Tablas.....	6
Descripción de la API.....	7
Descripción de los Endpoints	8

Introducción

El presente manual técnico proporciona una guía detallada para comprender y desarrollar el Sistema de Comercio Electrónico, que tiene como objetivo permitir a los usuarios realizar compras en línea de manera segura y eficiente. El sistema se desarrolla utilizando Node.js y Express para crear una API REST robusta con características de autenticación, seguridad y gestión de productos.

Herramientas utilizadas para el desarrollo

Typescript

TypeScript es una extensión al lenguaje de programación JavaScript que se caracteriza por ampliar su sintaxis en el ámbito de los tipos. En este sentido, es un lenguaje de programación propio que se basa en JavaScript para darnos herramientas de desarrollo en cualquier escala de proyectos.

Además de agregar elementos a la sintaxis de JavaScript, TypeScript se conecta de manera más profunda con los editores de código, revisando errores de manera más oportuna. Una prueba de esto es su integración con Angular, View y React, plataformas fundamentales para un desarrollador web.

NodeJS

Node.js, es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en JavaScript. Node.js es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos.

BASE DE DATOS

MongoDB

El MongoDB es una base de datos orientada a documentos. En otras palabras, en vez de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Una de las diferencias claves con respecto a la base de datos relacionada, es que no tiene que seguir un esquema. Los documentos de una misma colección o concepto similar pueden presentar esquemas diferentes.

Esquema Conceptual

Visión General

El Sistema de Comercio Electrónico consta de varias entidades principales:

Usuario: Representara a los usuarios del sistema, quienes pueden registrarse, iniciar sesión, gestionar su perfil y realizar compras.

Producto: Representara los productos disponibles en el catálogo, con detalles como nombre, precio, descripción, etc.

Carrito de Compra: Permite a los usuarios agregar productos para su compra.

Compra: Registra las compras realizadas por los usuarios.

Bitácora de Compras: Registra las ventas diarias para fines administrativos.

Esquema Lógico

1. Módulo de Autenticación y Usuarios

Registro de Usuarios

Inicio de Sesión (Login)

Gestión de Perfiles (Actualización y Eliminación)

2. Catálogo de Productos

Consulta de Productos Disponibles

Búsqueda de Productos por Nombre y Categoría

3. Carrito de Compras

Agregar Productos al Carrito

Eliminar Productos del Carrito

Actualizar Cantidades en el Carrito

4. Módulo de Compras

Procesamiento de Compras

Registro de Ventas Diarias en la Bitácora

5. Módulo de Administración (Acceso restringido a administradores)

Gestión de Productos (Agregar, Actualizar, Eliminar)

Consulta de la Bitácora de Ventas

6. Base de Datos (MongoDB)

Colecciones:

Usuarios

Productos

Carrito de Compras

Ventas Diarias (Bitácora)

7. API REST

Endpoints para cada funcionalidad

Utiliza Node.js y Express para manejar las solicitudes HTTP

8. Seguridad

Autenticación de Usuarios mediante JWT (JSON Web Tokens)

Validación de Token en cada solicitud protegida

9. Frontend (Interfaz de Usuario)

Interfaz amigable para los usuarios finales

Formularios de Registro y Login

Visualización de Catálogo de Productos

Carrito de Compras

Panel de Administración (solo para administradores)

10. Servidor de Aplicación

Node.js y Express para la lógica de negocio

Conexión a la Base de Datos MongoDB

11. Registro de Errores y Bitácora

Registra errores y eventos importantes en un archivo de registro

Esquema Físico

1. Servidor de Aplicación

Tipo: Virtual

Sistema Operativo: Windows

Requisitos de Hardware:

CPU: Mínimo 2 núcleos

RAM: Mínimo 4 GB

Almacenamiento: Mínimo 50 GB

Software:

Node.js (última versión estable)

Express.js (framework para Node.js)

Typescript

MongoDB (base de datos)

Dependencias del proyecto (listadas en package.json)

2. Base de Datos (MongoDB)

Tipo: Base de Datos NoSQL

Versión: MongoDB

Requisitos de Hardware:

Descripción de las Tablas

Tabla: Usuario

Nombre: Usuario

Descripción: Almacena información de usuarios registrados en el sistema.

Columnas Principales:

DPI

Nombre

Correo Electrónico

...

Tabla: Producto

Nombre: Producto

Descripción: Almacena información de productos disponibles en el catálogo.

Columnas Principales:

ID

Nombre

Precio

...

Tabla: Compra

Nombre: Compra

Descripción: Registra las compras realizadas por los usuarios.

Columnas Principales:

ID

Cantidad

Total

...

Descripción de la API

Objetivos de la API

La API del Sistema de Comercio Electrónico tiene como objetivo proporcionar una interfaz de programación de aplicaciones segura y eficiente para la gestión de usuarios, productos, carritos de compra, compras y bitácoras de compras.

Principales Funcionalidades

La API ofrece las siguientes funcionalidades:

Registro de usuarios.

Autenticación de usuarios con JWT.

Gestión de perfiles de usuarios.

Acceso al catálogo de productos.

Agregación y gestión de productos en el carrito de compra.

Realización de compras.

Registro de ventas diarias.

Descripción de los Endpoints

/api/registro (Método: POST)

Descripción: Permite a los usuarios registrarse en el sistema.

Parámetros de Solicitud: Nombres, Apellidos, FechaNacimiento, clave, ValidacionClave, DireccionEntrega, Nit, NumeroTelefonico y Correo Electrónico.

Respuesta: Mensaje de respuesta o mensaje de error.

/api/login (Método: POST)

Descripción: Permite a los usuarios iniciar sesión y obtener un token de autenticación JWT.

Parámetros de Solicitud: Correo Electrónico, Clave

Respuesta: Mensaje de inicio de respuesta, Token JWT o mensaje de error.

/api/productos (Método: GET)

Descripción: Devuelve una lista de productos disponibles en el catálogo.

Parámetros de Solicitud: Token de Autenticación

Respuesta: Lista de productos en formato JSON.

/api/perfil/:DPI (Método: GET)

Descripción: Recupera la información del perfil de un usuario específico.

Parámetros de Solicitud: Token de Autenticación (Header), DPI del Usuario (URL).

Respuesta: Información del perfil del usuario en formato JSON o mensaje de error si no se encuentra.

/api/perfil/:DPI (Método: PUT)

Descripción: Permite a un usuario actualizar su perfil.

Parámetros de Solicitud: Token de Autenticación (Header), DPI del Usuario (URL), Datos a Actualizar (Body).

Respuesta: Mensaje de actualización exitosa o mensaje de error si la actualización falla.

/api/perfil/:DPI (Método: DELETE)

Descripción: Permite a un usuario eliminar su perfil.

Parámetros de Solicitud: Token de Autenticación (Header), DPI del Usuario (URL).

Respuesta: Mensaje de eliminación exitosa o mensaje de error si la eliminación falla.

/api/carrito (Método: GET)

Descripción: Devuelve el contenido del carrito de compras de un usuario.

Parámetros de Solicitud: Token de Autenticación (Header).

Respuesta: Lista de productos en el carrito de compras en formato JSON.

/api/carrito (Método: POST)

Descripción: Permite a un usuario agregar productos a su carrito de compras.

Parámetros de Solicitud: Token de Autenticación (Header), ID del Producto (Body), Cantidad (Body).

Respuesta: Mensaje de agregación exitosa o mensaje de error si la agregación falla.

/api/carrito/:ID (Método: DELETE)

Descripción: Permite a un usuario eliminar un producto específico de su carrito de compras.

Parámetros de Solicitud: Token de Autenticación (Header), ID del Producto (URL).

Respuesta: Mensaje de eliminación exitosa o mensaje de error si la eliminación falla.

/api/compra (Método: POST)

Descripción: Permite a un usuario realizar una compra y registrarla en el sistema.

Parámetros de Solicitud: Token de Autenticación (Header).

Respuesta: Mensaje de compra exitosa o mensaje de error si la compra falla.

/api/productos (Método: GET)

Descripción: Devuelve una lista de todos los productos disponibles en el catálogo.

Parámetros de Solicitud: Token de Autenticación (Header) con rol de Administrador.

Respuesta: Lista de productos en formato JSON.

/api/producto/:ID (Método: POST)

Descripción: Permite a un administrador agregar un nuevo producto al catálogo.

Parámetros de Solicitud: Token de Autenticación (Header) con rol de Administrador, Datos del Producto (Body).

Respuesta: Mensaje de agregación exitosa o mensaje de error si la agregación falla.

/api/producto/:ID (Método: PUT)

Descripción: Permite a un administrador actualizar la información de un producto existente en el catálogo.

Parámetros de Solicitud: Token de Autenticación (Header) con rol de Administrador, ID del Producto (URL), Datos Actualizados (Body).

Respuesta: Mensaje de actualización exitosa o mensaje de error si la actualización falla.

/api/producto/:ID (Método: DELETE)

Descripción: Permite a un administrador eliminar un producto del catálogo.

Parámetros de Solicitud: Token de Autenticación (Header) con rol de Administrador, ID del Producto (URL).

Respuesta: Mensaje de eliminación exitosa o mensaje de error si la eliminación falla.