

# **MANUAL TECNICO**

# TABLA DE CONTENIDO

## 1. Esquema Conceptual 1.1. Usuarios

- Descripción
- Atributos 1.2. Productos
- Descripción
- Atributos 1.3. Carritos
- Descripción
- Atributos

## 2. Esquema Lógico 2.1. Base de Datos

- Descripción 2.2. Colecciones
- Usuarios
- Productos
- Carritos 2.3. Conexión a la Base de Datos
- URL de Conexión

## 3. Esquema Físico 3.1. Implementación Física de la Base de Datos

- MongoDB

## 4. Descripción de las Tablas 4.1. Usuarios

- Descripción
- Campos 4.2. Productos
- Descripción
- Campos 4.3. Carritos
- Descripción
- Campos

## 5. Descripción de la API

- Tecnologías Utilizadas
  - Express.js
  - JSON Web Tokens (JWT)

# TABLA DE CONTENIDO

## 6.Descripción de los Endpoints Utilizados 6.1. Usuarios

- Registro de Usuarios
  - POST /usuarios/registro
- Inicio de Sesión
  - POST /usuarios/iniciar-sesion

## 6.2. Productos

- Obtención de Productos
  - GET /productos
- Creación de Productos
  - POST /productos
- Actualización de Productos
  - PUT /productos/:id
- Eliminación de Productos
  - DELETE /productos/:id

## 6.3. Carritos

- Obtención del Carrito
  - GET /carrito
- Añadir Producto al Carrito
  - POST /carrito
- Eliminar Carrito
  - DELETE /carrito
- Realizar Compra
  - POST /carrito/compra

## 7.Servicios WEB

- GitHub
- Netlify
- Render

# 1. ESQUEMA CONCEPTUAL

## Usuarios

- Descripción: La entidad "Usuario" representa a los usuarios de la aplicación. Cada usuario está identificado por su dirección de correo electrónico única.
- Atributos:
  - CorreoElectronico (Clave Primaria): La dirección de correo electrónico del usuario, que también se utiliza como identificador único.
  - Nombre: El nombre del usuario registrado en la aplicación.
  - Contraseña: La contraseña del usuario, almacenada de forma segura utilizando técnicas de hash y salting.
  - Rol: Indica el rol del usuario en la aplicación, lo que permite distinguir entre clientes y administradores.

## Productos

- Descripción: La entidad "Producto" representa los productos disponibles en la tienda. Cada producto tiene un identificador único.
- Atributos:
  - Identificador (Clave Primaria): Un identificador único para el producto.
  - Nombre: El nombre del producto.
  - PrecioDescuento: El precio del producto con descuento aplicado.
  - Disponibilidad: La cantidad disponible en el inventario.

## Carritos

- Descripción: La entidad "Carrito" está vinculada a los usuarios y contiene los productos que un usuario ha agregado a su carrito de compras.
- Atributos:
  - CorreoElectronico (Clave Primaria): La dirección de correo electrónico del usuario dueño del carrito, que sirve como clave principal.
  - Productos: Un array que almacena los productos en el carrito, cada uno con su identificador y cantidad.
  - Total: El costo total del carrito de compras, calculado en función de los productos y sus cantidades.

## 2. ESQUEMA LÓGICO

### Base de Datos

- Descripción: La aplicación utiliza MongoDB como sistema de gestión de bases de datos. MongoDB es una base de datos NoSQL que permite un almacenamiento flexible de datos estructurados.

### Colecciones

- Usuarios: Almacena la información de los usuarios registrados en la aplicación.
- Productos: Contiene los detalles de todos los productos disponibles en la tienda.

### Conexión a la Base de Datos

- URL de Conexión: La aplicación se conecta a la base de datos MongoDB utilizando una URL de conexión que especifica la dirección del servidor de base de datos, el puerto y el nombre de la base de datos.

## 3. ESQUEMA FÍSICO

### Implementación Física de la Base de Datos

- MongoDB: La base de datos MongoDB se encuentra alojada en un servidor remoto y se accede a través de la URL de conexión proporcionada.

## 4. DESCRIPCIÓN DE LAS TABLAS

### Usuarios

- Descripción: La colección "Usuarios" almacena información sobre los usuarios registrados en la aplicación.

### Campos:

- CorreoElectronico: La dirección de correo electrónico del usuario (clave principal).
- Nombre: El nombre del usuario.
- Contraseña: La contraseña del usuario, almacenada de forma segura utilizando técnicas de hash y salting.
- Rol: Indica el rol del usuario en la aplicación (por ejemplo, "cliente" o "admin").

### Productos

- Descripción: La colección "Productos" almacena información detallada sobre los productos disponibles en la tienda.

### Campos:

- Identificador: Un identificador único para el producto (clave principal).
- Nombre: El nombre del producto.
- PrecioDescuento: El precio del producto con descuento aplicado.
- Disponibilidad: La cantidad disponible en el inventario.

### **Carritos**

- Descripción: La colección "Carritos" almacena el contenido del carrito de compras de cada usuario.

### **Campos:**

- CorreoElectronico: La dirección de correo electrónico del usuario dueño del carrito (clave principal).
- Productos: Un array que almacena los productos en el carrito, cada uno con su identificador y cantidad.
- Total: El costo total del carrito de compras, calculado en función de los productos y sus cantidades.

## **5. DESCRIPCIÓN DE LA API**

### **Tecnologías Utilizadas**

- **Express.js:** La aplicación utiliza Express.js como marco de desarrollo de aplicaciones web para la implementación de la API RESTful.
- **JSON Web Tokens (JWT):** Se emplea JWT para autenticar y autorizar a los usuarios en la API.

# 6. DESCRIPCIÓN DE LOS ENDPOINTS UTILIZADOS

## Usuarios

- **Registro de Usuarios**

- POST /usuarios/registro: Permite a los usuarios registrarse proporcionando su dirección de correo electrónico, nombre y contraseña.

- **Inicio de Sesión**

- POST /usuarios/iniciar-sesion: Permite a los usuarios iniciar sesión proporcionando su dirección de correo electrónico y contraseña.

## Productos

- **Obtención de Productos**

- GET /productos: Obtiene la lista de productos disponibles en la tienda.

- **Creación de Productos**

- POST /productos: Permite a los administradores agregar nuevos productos a la tienda especificando su nombre, precio y disponibilidad.

- **Actualización de Productos**

- PUT /productos/:id: Permite a los administradores actualizar los detalles de un producto existente mediante su identificador único.

- **Eliminación de Productos**

- DELETE /productos/:id: Permite a los administradores eliminar un producto de la tienda utilizando su identificador único.



## **Carritos**

- **Obtención del Carrito**

- GET /carrito: Obtiene el carrito de compras actual del usuario autenticado.

- **Añadir Producto al Carrito**

- POST /carrito: Permite a los usuarios agregar un producto al carrito de compras proporcionando el identificador del producto y la cantidad deseada.

- **Eliminar Carrito**

- DELETE /carrito: Permite a los usuarios eliminar por completo su carrito de compras.

- **Realizar Compra**

- POST /carrito/compra: Permite a los usuarios realizar una compra con los productos en su carrito de compras. La cantidad disponible de productos se actualiza y se calcula el costo total de la compra.

# 7. SERVICIOS WEB

## Render

- **¿Qué es Render?:** Render es una plataforma de despliegue de aplicaciones que proporciona un entorno de desarrollo y alojamiento optimizado para aplicaciones web y backend. Render se destaca por su enfoque en la simplicidad y la automatización, lo que lo convierte en una elección popular para desarrolladores que desean desplegar aplicaciones de forma rápida y sencilla.
- **Ventajas de utilizar Render:**
  - **Despliegue sencillo:** Render automatiza gran parte del proceso de despliegue, lo que facilita la implementación de aplicaciones.
  - **Escalabilidad automática:** Render se encarga de escalar tu aplicación de forma automática según la demanda, lo que garantiza un rendimiento óptimo.
  - **Variedad de servicios:** Render ofrece una amplia gama de servicios, incluyendo alojamiento web, bases de datos, almacenamiento, y más.
  - **Integración con GitHub:** La integración con repositorios de GitHub facilita la implementación continua (CI/CD).

<https://api-2-xgyd.onrender.com>

## Netlify

- **¿Qué es Netlify?:** Netlify es una plataforma de alojamiento y despliegue web que ofrece un entorno de desarrollo y despliegue moderno y centrado en la automatización. Permite a los desarrolladores crear, desplegar y gestionar sitios web de forma sencilla y eficiente.
- **Ventajas de utilizar Netlify:**
  - **Despliegue sencillo:** Netlify simplifica el proceso de despliegue web, permitiendo a los desarrolladores enfocarse en el desarrollo de sus sitios.
  - **Escalabilidad automática:** Los sitios alojados en Netlify se escalan automáticamente según la demanda, garantizando un alto rendimiento.
  - **Integración con GitHub:** Netlify se integra perfectamente con repositorios de GitHub, lo que facilita la implementación continua (CI/CD) y despliegues automáticos.

### Web Proyecto Final

Create Next.js - Netlify - React

 netlify.app

## GitHub

- **Repositorios:** Un repositorio en Git es un lugar donde se almacena y gestiona un proyecto. Describe cómo los repositorios alojan archivos y el historial de cambios.
- **Commits:** Los commits representan cambios individuales en un repositorio. Explica cómo los commits registran cambios en el código y proporcionan una historia de revisión.
- **Ramas (Branches):** Las ramas son versiones paralelas de un repositorio. Describe cómo las ramas permiten trabajar en nuevas características o solucionar problemas sin afectar la rama principal.
- **Fusiones (Merges):** Las fusiones combinan cambios de una rama a otra. Explica cómo se utilizan las fusiones para unificar el trabajo de diferentes ramas.

