

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN Y
CIENCIAS DE LA COMPUTACIÓN
DESARROLLO WEB



Manual Técnico
Proyecto de Curso

Brian Estuardo Galeano Mayen 9490-21-570

Jorge Antonio Macario Cano 9490-15-3116

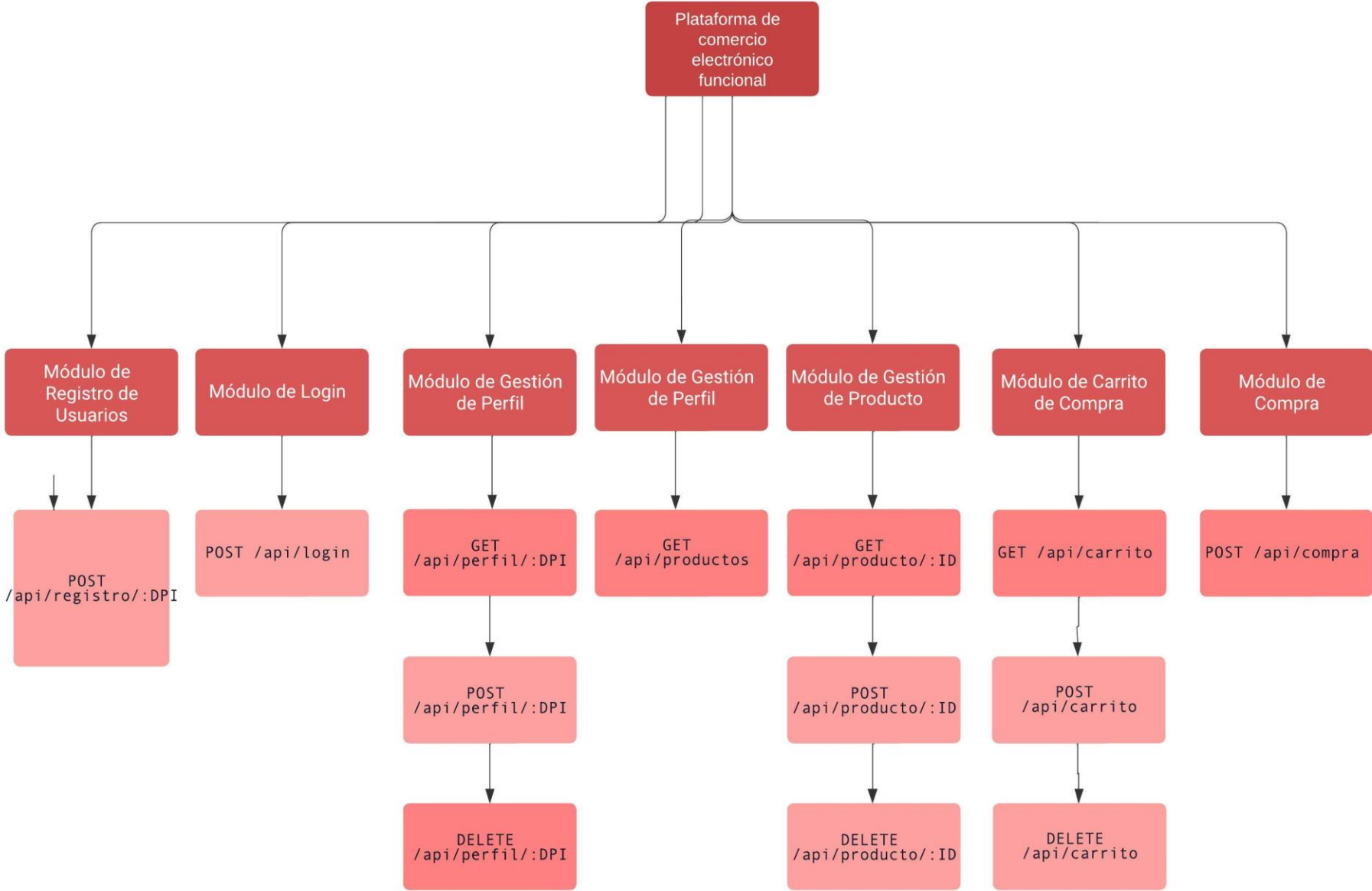
GUATEMALA, 03 NOVIEMBRE 2023

INDICE

Contenido

ESQUEMA CONCEPTUAL.....	3
ESQUEMA LÓGICO	4
DIAGRAMA FÍSICO.....	5
Descripción de las tablas.....	6
Módulo de Registro de Usuarios:.....	6
Módulo de Login:	7
Módulo de Gestión de Perfil	8
Módulo Catálogo de Productos:	9
Módulo de Gestión de Producto	11
Módulo de Carrito de Compra	12
Módulo de Compra	15
Descripción de la API.....	16
Descripción de los endpoints utilizados.....	17
Descripción de uso de Typescript.....	19
Descripción de uso SASS/SCSS.	21

ESQUEMA CONCEPTUAL



ESQUEMA LÓGICO

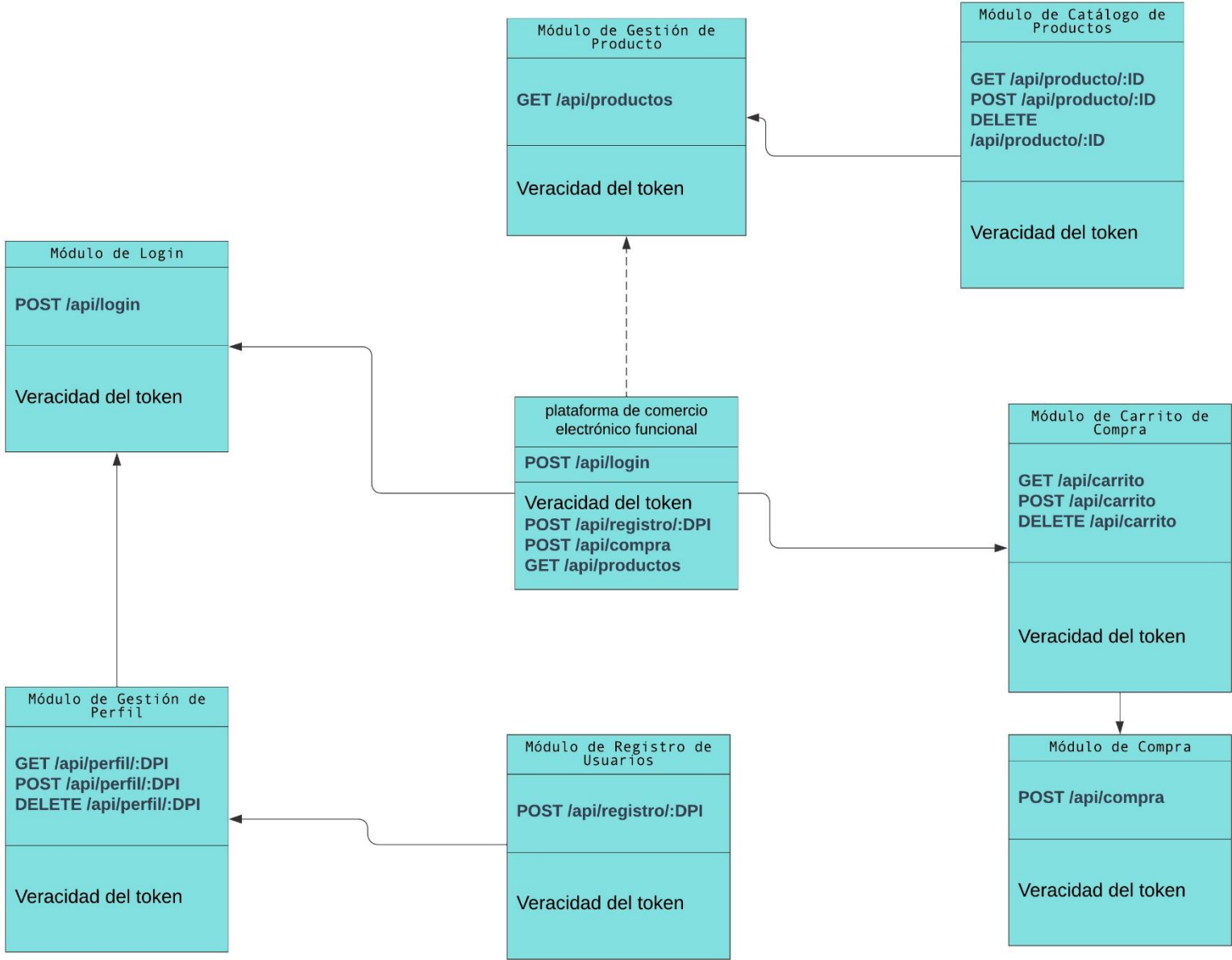
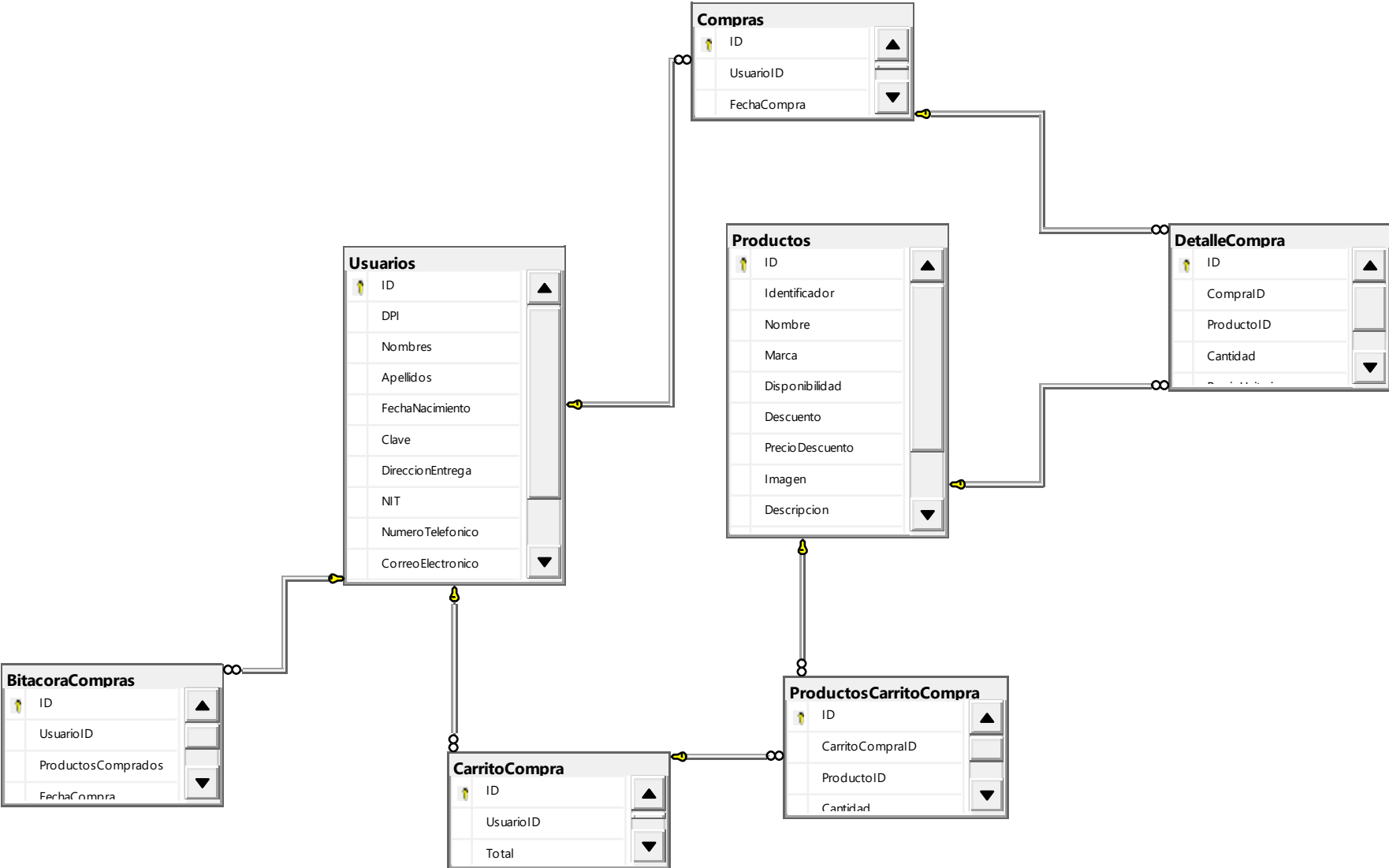


DIAGRAMA FÍSICO



Descripción de las tablas

Módulo de Registro de Usuarios:

Modulo	Registro de Usuarios
Método	POST
Ruta	/api/registro/:DPI
Cuerpo	{ "Nombres":... "Apellidos":... "FechaNacimiento": "Clave": "ValidacionClave": "DireccionEntrega":..... "NIT":..... "NúmeroTelefonico":..... "CorreoElectronico":..... }
Respuesta	{ "Mensaje": "Mensaje de respuesta ó Error" }

Descripción de las Tablas

1. Usuarios

- Campos: ID, DPI, Nombres, Apellidos, Fecha de Nacimiento, Clave (hash), Dirección de Entrega, NIT, Número Telefónico, Correo Electrónico, Fecha de Registro, Rol
- Descripción: Esta tabla almacena información sobre los usuarios registrados en la aplicación. Los campos incluyen datos personales, de contacto y de autenticación.

2. Requisitos de Seguridad de Contraseña

- Campos: ID, Requisito, Descripción
- Descripción: Contiene los requisitos de seguridad para las contraseñas de los usuarios, como longitud mínima, uso de mayúsculas, minúsculas, números y caracteres especiales.

3. Registros de Errores de Registro

- Campos: ID, DPI, NIT, Correo Electrónico, Mensaje de Error, Fecha de Registro
- Descripción: Registra los errores de registro, incluyendo DPI, NIT y correo electrónico duplicados o problemas con la contraseña.

Módulo de Login:

Modulo	Login
Método	POST
Ruta	/api/login
Cuerpo	{ "CorreoElectronico":..... "Clave": }
Respuesta	{ "Mensaje": "Mensaje de respuesta ó Error", "Token": }

Descripción de la Tabla

Módulo de Login

El módulo de login se ocupa de la autenticación de usuarios utilizando las credenciales proporcionadas, que generalmente incluyen el correo electrónico y la contraseña. La validación de las credenciales en el módulo de login implica lo siguiente:

- **Verificación de Correo Electrónico:** El sistema verifica si el correo electrónico proporcionado por el usuario existe en la tabla de "Usuarios". Si no existe, se devuelve un mensaje de error.
- **Verificación de Contraseña:** Si el correo electrónico existe en la base de datos, el sistema compara la contraseña proporcionada con la contraseña almacenada en la base de datos. Para garantizar la seguridad, la contraseña almacenada en la base de datos suele estar en forma de hash. Si las contraseñas coinciden, se considera una autenticación exitosa.
- **Generación de Token:** En caso de autenticación exitosa, se genera un token de acceso (JWT, por ejemplo) que se incluye en la respuesta. Este token se utiliza para autorizar al usuario en las solicitudes subsiguientes a las API protegidas.

Módulo de Gestión de Perfil

Modulo	Gestión de Perfil
Método	GET, POST, DELETE
Ruta	/api/perfil/:DPI
Encabezado	{ "Token":... }
Cuerpo	{ " }
Respuesta	{ "Mensaje": "Mensaje de respuesta ó Error", }

Descripción de la Tabla

Módulo de Gestión de Perfil

El módulo de gestión de perfil permite a los usuarios realizar las siguientes operaciones:

- **Obtener Perfil (GET):** Para obtener toda la información del perfil del usuario. El usuario debe proporcionar un token válido en el encabezado para acceder a esta función. La información del perfil se recupera de la tabla de "Usuarios" basándose en el DPI proporcionado.
- **Actualizar Perfil (POST):** Para actualizar la información del perfil del usuario. El usuario debe proporcionar un token válido en el encabezado para acceder a esta función. Las actualizaciones se aplican a la información existente en la tabla de "Usuarios". Se deben realizar las siguientes validaciones:
 - No se permiten campos vacíos en la solicitud.
 - El NIT, DPI o Correo no pueden duplicarse en la base de datos.
 - Se debe validar el formato del correo electrónico al actualizar.
- **Eliminar Perfil (DELETE):** Para eliminar el perfil del usuario. El usuario debe proporcionar un token válido en el encabezado para acceder a esta función. Una vez eliminado el perfil, el token ya no será válido.

Validación del Token

La veracidad del token se valida en cada operación del módulo de gestión de perfil. Esto garantiza que solo los usuarios autenticados tengan acceso a sus perfiles y puedan realizar operaciones de actualización y eliminación.

El módulo de gestión de perfil es fundamental para permitir a los usuarios mantener y administrar su información personal de manera segura en la aplicación. Asegúrate de implementar medidas de seguridad adecuadas para proteger la integridad de los datos del usuario y la autenticación.

Módulo Catálogo de Productos:

Modulo	Catálogo de Productos
Método	GET
Ruta	/api/productos
Encabezado	{ "Token":... }
Cuerpo	[{ "Identificador":..... "Nombre":... "Marca":... "Disponibilidad":..... "Descuento":..... "PrecioDescuento":..... "Imagen":....., "Descripcion" "Categorias": ["Categoria1", "Categoria2", "Categorias3"] },....]
Respuesta	{ "Mensaje": "Mensaje de respuesta ó Error", }

Descripción de la Tabla

1. Productos

- **Campos:** ID, Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción, Categorías
- **Descripción:** Esta tabla almacena información detallada sobre los productos disponibles en la aplicación. Incluye datos como el nombre, marca, disponibilidad, descuento, precio con descuento, imagen, descripción y categorías a las que pertenecen los productos.

Módulo de Catálogo de Productos

El módulo de catálogo de productos permite a los usuarios realizar la siguiente operación:

- **Obtener Catálogo de Productos (GET):** Para obtener una lista de productos disponibles en la aplicación. El usuario debe proporcionar un token válido en el encabezado para acceder a esta función. Los productos se recuperan de la tabla "Productos" en la base de datos.

Validación del Token

La veracidad del token se valida en la operación de obtención del catálogo de productos para garantizar que solo los usuarios autenticados tengan acceso a esta información.

Este módulo es esencial para permitir a los usuarios explorar y conocer los productos disponibles en la aplicación. Asegúrate de implementar medidas de seguridad adecuadas para proteger la integridad de los datos del catálogo de productos y la autenticación de los usuarios.

Módulo de Gestión de Producto

Modulo	Gestión de Producto
Método	GET, POST, DELETE
Ruta	/api/Producto/:ID
Encabezado	{ "Token":... }
Cuerpo	{ "Identificador":.... "Nombre":... "Marca":... "Disponibilidad":..... "Descuento":..... "PrecioDescuento":..... "Imagen":....., "Descripcion" "Categorias": ["Categoria1", "Categoria2", "Categorias3"], "Habilitado":.... }
Respuesta	{ "Identificador":.... "Nombre":... "Marca":... "Disponibilidad":..... "Descuento":..... "PrecioDescuento":..... "Imagen":....., "Descripcion" "Categorias": ["Categoria1", "Categoria2", "Categorias3"], "Habilitado":....., "Mensaje": "Mensaje de respuesta ó Error", }

Descripción de las Tablas

1. Productos

- Campos: ID, Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción, Categorías, Habilitado
- Descripción: Esta tabla almacena información detallada sobre los productos disponibles en la aplicación. Incluye datos como el nombre, marca, disponibilidad, descuento, precio con descuento, imagen, descripción, categorías a las que pertenecen los productos y un campo "Habilitado" que indica si el producto está activo o no.

Módulo de Gestión de Producto

El módulo de gestión de productos permite a los administradores realizar las siguientes operaciones:

- **Obtener Producto (GET):** Permite a los administradores obtener información detallada sobre un producto específico mediante su ID. El usuario debe proporcionar un token válido con el rol de administrador para acceder a esta función.
- **Crear Producto (POST):** Permite a los administradores agregar nuevos productos a la aplicación. Los campos requeridos incluyen Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio, Imagen, Descripción, Categorías y Habilitado. El Precio con Descuento se calcula automáticamente a partir del Precio y el Descuento. El usuario debe proporcionar un token válido con el rol de administrador para acceder a esta función.
- **Actualizar Producto (PUT):** Permite a los administradores actualizar la información de un producto existente. Se deben proporcionar todos los campos necesarios, incluido el ID del producto que se va a actualizar. El usuario debe proporcionar un token válido con el rol de administrador para acceder a esta función.
- **Eliminar Producto (DELETE):** Permite a los administradores cambiar el estado de un producto a "no habilitado" en lugar de eliminarlo por completo. Esto evita que el producto se muestre en el catálogo, pero aún se conserva en la base de datos. El usuario debe proporcionar un token válido con el rol de administrador para acceder a esta función.

Validación del Token

El token se valida en todas las operaciones del módulo de gestión de productos para garantizar que solo los administradores autenticados puedan realizar estas acciones críticas. Además, se verifica el rol del usuario para asegurarse de que solo los administradores tengan acceso a estas operaciones.

Este módulo es esencial para permitir a los administradores gestionar productos, actualizar su información y mantener un control sobre su disponibilidad. Asegúrate de implementar medidas de seguridad adecuadas para proteger la integridad de los datos de productos y la autenticación de los administradores

Modulo	Carrito de Compra
Método	GET, POST, DELETE
Ruta	/api/carrito
Encabezado	{ "Token":... }
Cuerpo	{...}
Respuesta	{ "Productos": [{ "Identificador":..... "Nombre":... "Marca":... "Disponibilidad":..... "Descuento":..... "PrecioDescuento":..... "Imagen":....., "Descripcion", "Cantidad":..... }], "Total":....., "Mensaje": "Mensaje de respuesta ó Error", }

Módulo de Carrito de Compra

El módulo de carrito de compra permite a los usuarios realizar operaciones relacionadas con la gestión de productos en su carrito de compra. A continuación, se describen este módulo y cómo se relaciona con las tablas existentes:

Descripción de las Tablas

1. Productos

- Campos: ID, Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción, Categorías, Habilitado
- Descripción: Esta tabla almacena información detallada sobre los productos disponibles en la aplicación.

2. Carrito de Compra

- Campos: ID de Usuario, Productos, Total
- Descripción: Esta tabla almacena información sobre los productos agregados al carrito de compra de cada usuario. Contiene una lista de productos con detalles como Identificador, Nombre, Marca,

Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción y Cantidad. También registra el monto total de la compra.

Módulo de Carrito de Compra

El módulo de carrito de compra permite a los usuarios realizar las siguientes operaciones:

- **Obtener Carrito de Compra (GET):** Permite a los usuarios obtener información sobre los productos en su carrito de compra. Devuelve una lista de productos con detalles y el monto total a pagar.
- **Agregar al Carrito (POST):** Permite a los usuarios agregar productos a su carrito de compra. Se debe proporcionar la cantidad deseada del producto a agregar. El sistema valida la disponibilidad del producto y actualiza el carrito de compra.
- **Actualizar Cantidad en el Carrito (PUT):** Permite a los usuarios actualizar la cantidad de productos en su carrito de compra. Se debe proporcionar la cantidad deseada del producto. El sistema valida la disponibilidad del producto y actualiza el carrito de compra.
- **Eliminar Producto del Carrito (DELETE):** Permite a los usuarios eliminar productos de su carrito de compra. El sistema actualiza el carrito de compra y recalcula el monto total.

Validación del Token

El token se valida en todas las operaciones del módulo de carrito de compra para garantizar que solo los usuarios autenticados puedan realizar estas acciones relacionadas con su carrito de compra.

Este módulo es esencial para permitir a los usuarios gestionar los productos que desean comprar y mantener un registro de su carrito de compra en la aplicación. Asegúrate de implementar medidas de seguridad adecuadas para proteger la integridad de los datos del carrito de compra y garantizar la disponibilidad precisa de los productos en el inventario.

Módulo de Compra

Modulo	Compra
Método	POST
Ruta	/api/compra
Encabezado	{ "Token":... }
Cuerpo	{ }
Respuesta	{ "Mensaje": "Mensaje de respuesta ó Error", }

Descripción de las Tablas

1. Productos

- Campos: ID, Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción, Categorías, Habilitado
- Descripción: Esta tabla almacena información detallada sobre los productos disponibles en la aplicación.

2. Carrito de Compra

- Campos: ID de Usuario, Productos, Total
- Descripción: Esta tabla almacena información sobre los productos agregados al carrito de compra de cada usuario. Contiene una lista de productos con detalles como Identificador, Nombre, Marca, Disponibilidad, Descuento, Precio con Descuento, Imagen, Descripción y Cantidad. También registra el monto total de la compra.

3. Bitácora de Compras

- Campos: ID de Usuario, Productos Comprados, Fecha de Compra
- Descripción: Esta tabla registra las compras realizadas por los usuarios. Contiene información sobre los productos comprados, el usuario que realizó la compra y la fecha de la compra.

Módulo de Compra

El módulo de compra permite a los usuarios realizar la siguiente operación:

- **Realizar Compra (POST):** Permite a los usuarios realizar una compra de los productos en su carrito de compra. Se requiere la validación del token para garantizar que el usuario correspondiente esté realizando la compra. La operación de compra implica descontar la cantidad de productos en el inventario y crear un registro en la bitácora de compras.

Validación del Token

El token se valida para garantizar que el usuario que realiza la compra sea el mismo usuario cuyo carrito de compra se está procesando.

Este módulo es esencial para permitir a los usuarios finalizar sus compras y mantener un registro de las compras realizadas en la aplicación. Asegúrate de implementar medidas de seguridad adecuadas para proteger la integridad de los datos de compra y mantener la precisión del inventario de productos. También, registra las compras realizadas en la bitácora para futuras referencias.

Descripción de la API

La API es un conjunto de puntos de acceso (endpoints) y reglas de comunicación que permite que diferentes aplicaciones se comuniquen entre sí. En el contexto de tu aplicación web, la API se utiliza para exponer y gestionar funcionalidades específicas relacionadas con la autenticación de usuarios, la gestión de perfiles, el catálogo de productos, el carrito de compras, la gestión de productos y las compras. A continuación, se proporciona una descripción de la API que se ha diseñado:

1. Módulo de Registro de Usuarios:

- Permite a los usuarios registrarse proporcionando información personal.
- Valida la autenticidad de la dirección de correo electrónico.
- Evita la duplicación de NIT, DPI o correo electrónico en la base de datos.
- Garantiza la seguridad de las contraseñas y su conformidad con requisitos de seguridad.
- No permite campos vacíos.

2. Módulo de Login:

- Permite a los usuarios autenticarse proporcionando su correo electrónico y contraseña.
- Devuelve un token de acceso en caso de autenticación exitosa.

3. Módulo de Gestión de Perfil:

- Permite a los usuarios obtener información detallada de su perfil.
- Permite a los usuarios actualizar su perfil, garantizando la integridad de los datos.
- Evita la duplicación de NIT, DPI o correo electrónico al actualizar.
- Valida el formato del correo electrónico.
- Permite eliminar el perfil, lo que invalida el token de acceso.

4. Módulo de Catálogo de Productos:

- Permite a los usuarios obtener información sobre los productos disponibles.
5. **Módulo de Gestión de Producto:**
 - Permite a los administradores crear o actualizar productos.
 - Valida la autenticidad del token y el rol del usuario.
 - Cambia el estado de los productos al eliminarlos o actualizarlos.
 - Calcula el precio con descuento de los productos.
 - Evita campos vacíos y duplicados.
 6. **Módulo de Carrito de Compra:**
 - Permite a los usuarios obtener información sobre los productos en su carrito de compra.
 - Permite a los usuarios actualizar la cantidad de productos en el carrito, validando la disponibilidad.
 - Permite a los usuarios eliminar productos del carrito y actualiza el monto total.
 7. **Módulo de Compra:**
 - Permite a los usuarios finalizar una compra, descontando los productos del inventario.
 - Registra la compra en una bitácora.
 - Valida el token del usuario.

Descripción de los endpoints utilizados

1. **Módulo de Registro de Usuarios:**
 - **POST /api/registro/:DPI:** Permite a los usuarios registrarse proporcionando información personal como nombres, apellidos, fecha de nacimiento, clave, validación de clave, dirección de entrega, NIT, número telefónico y correo electrónico. Se validan varios aspectos, como la dirección de correo electrónico válida, la no duplicación de NIT, DPI o correo, y la seguridad de la contraseña.
2. **Módulo de Login:**
 - **POST /api/login:** Permite a los usuarios autenticarse proporcionando su correo electrónico y contraseña. Si la autenticación es exitosa, se devuelve un token de acceso.
3. **Módulo de Gestión de Perfil:**
 - **GET /api/perfil/:DPI:** Permite a los usuarios obtener información detallada de su perfil, validando la autenticidad del token.
 - **POST /api/perfil/:DPI:** Permite a los usuarios actualizar su perfil, garantizando que no haya campos vacíos y que no se dupliquen NIT, DPI o correo. Se valida el formato del correo electrónico.

- **DELETE /api/perfil/:DPI:** Permite a los usuarios eliminar su perfil, lo que invalida el token de acceso.
4. **Módulo de Catálogo de Productos:**
 - **GET /api/productos:** Permite a los usuarios obtener información sobre el catálogo de productos disponibles. Se valida la autenticidad del token.
 5. **Módulo de Gestión de Producto:**
 - **GET /api/producto/:ID:** Permite a los usuarios obtener información detallada sobre un producto específico. Se valida el token y el rol del usuario.
 - **POST /api/producto/:ID:** Permite a los administradores crear un nuevo producto o actualizar uno existente. Se validan varios aspectos, como la autenticidad del token, el rol del usuario y la no duplicación de ciertos campos. Además, se realiza un cálculo de Precio con Descuento.
 - **DELETE /api/producto/:ID:** Permite a los administradores eliminar un producto o cambiar su estado a "reservado" para descontar del inventario.
 6. **Módulo de Carrito de Compra:**
 - **GET /api/carrito:** Permite a los usuarios obtener información sobre los productos en su carrito de compra. Se valida la autenticidad del token.
 - **POST /api/carrito:** Permite a los usuarios actualizar la cantidad de productos en su carrito de compra. Se valida la disponibilidad de productos.
 - **DELETE /api/carrito:** Permite a los usuarios eliminar productos de su carrito de compra y actualiza el monto total.
 7. **Módulo de Compra:**
 - **POST /api/compra:** Permite a los usuarios finalizar una compra, descontando los productos del inventario y registrando la compra en la bitácora. Se valida el token del usuario.

Descripción de uso de Typescript.

Tipado Estático

El tipado estático en TypeScript es una característica fundamental que agrega seguridad y robustez al desarrollo de aplicaciones. A diferencia de JavaScript, que utiliza un tipado dinámico, en TypeScript se definen los tipos de datos en tiempo de desarrollo, lo que permite al compilador verificar la corrección de los tipos antes de la ejecución. Esta característica aumenta la legibilidad y la documentación del código, previene errores comunes y facilita la depuración. Además, TypeScript ofrece la ventaja de la inferencia de tipos, lo que permite al compilador deducir los tipos en función del contexto, lo que ahorra tiempo y reduce la necesidad de anotaciones explícitas. En resumen, el tipado estático de TypeScript mejora la calidad y la confiabilidad del código, lo que resulta en aplicaciones más seguras y mantenibles.

Evolución de JavaScript:

TypeScript representa una evolución significativa de JavaScript, especialmente en el contexto del desarrollo de aplicaciones web modernas. Su sistema de tipado estático permite a los desarrolladores definir y verificar tipos de datos, mejorando la seguridad y reduciendo errores. Esto se traduce en un código más mantenible y robusto, lo que es esencial para proyectos a gran escala y equipos de desarrollo colaborativos. A pesar de estas ventajas, TypeScript mantiene la flexibilidad y la compatibilidad con JavaScript, lo que facilita la transición y la adopción gradual. La integración con herramientas populares y una comunidad activa respaldan su uso, y su amplia biblioteca de código TypeScript agiliza el desarrollo de aplicaciones modernas. En resumen, TypeScript es la respuesta a las necesidades de un desarrollo web más seguro y eficiente, proporcionando a los desarrolladores las herramientas necesarias para abordar proyectos complejos con éxito.

Programación Orientada a Objetos:

TypeScript es un lenguaje de programación orientado a objetos que se centra en la creación y manipulación de objetos como la piedra angular para resolver problemas. Los objetos en TypeScript albergan datos y comportamientos en forma de propiedades y métodos, lo que permite a los desarrolladores crear estructuras de datos más complejas mediante la organización de la lógica en clases y objetos reutilizables. Este enfoque orientado a objetos en TypeScript se basa en conceptos esenciales como la encapsulación, que permite ocultar detalles de implementación y exponer solo lo necesario para interactuar con un objeto, la herencia para la creación de clases derivadas y la promoción de la reutilización de código, y el polimorfismo, que posibilita que objetos de diferentes clases respondan de manera flexible a una interfaz común. En resumen, TypeScript potencia la programación orientada a objetos, lo que resulta en un código más modular, estructurado y mantenible en aplicaciones de cualquier envergadura.

Módulos y Organización del Código:

TypeScript introduce un sistema de módulos que resulta fundamental para la organización y estructuración eficiente del código. Estos módulos son bloques que agrupan elementos relacionados, como variables, funciones y clases, lo que simplifica la reutilización y la administración de dependencias, especialmente en aplicaciones de mayor envergadura. TypeScript ofrece dos tipos de módulos: los internos, gestionados a través de las palabras clave ``import`` y ``export``, destinados a organizar y dividir el código dentro de un proyecto; y los externos, utilizados para importar código de bibliotecas y dependencias externas, fomentando la reutilización y simplificando la administración de paquetes y dependencias. Esta organización modular del código es una práctica común en TypeScript, ya que contribuye a mejorar la claridad, la escalabilidad y el mantenimiento de las aplicaciones, y la flexibilidad de TypeScript para admitir varios sistemas de módulos, como CommonJS, AMD y ES6, permite adaptarse a diversos entornos y requisitos de desarrollo. En resumen, el sistema de módulos de TypeScript es esencial para la construcción de aplicaciones bien estructuradas y modularizadas.

Herramientas de Desarrollo:

TypeScript ofrece un conjunto robusto de herramientas de desarrollo que simplifican la creación de aplicaciones web. Su compilador (tsc) convierte el código TypeScript en JavaScript, y entornos como Visual Studio Code proporcionan un entorno de desarrollo eficiente con detección de errores en tiempo real. Herramientas de construcción como Webpack y la gestión de paquetes con npm ayudan a organizar y optimizar el código. Además, TypeScript se integra con una amplia gama de bibliotecas y marcos de trabajo escritos en TypeScript, lo que facilita la creación de aplicaciones modernas y fomenta la reutilización de código. En conjunto, estas herramientas hacen que TypeScript sea una elección sólida para desarrolladores que buscan un flujo de trabajo eficiente y recursos valiosos para proyectos web.

Seguridad en Aplicaciones Dinámicas

TypeScript agrega una capa de seguridad significativa a las aplicaciones web dinámicas al permitir a los desarrolladores definir y verificar tipos de datos en tiempo de desarrollo, lo que ayuda a prevenir errores comunes y vulnerabilidades en tiempo de ejecución. Esto resulta fundamental para aplicaciones que manejan datos sensibles, ya que garantiza un manejo más seguro y cumple con los estándares de privacidad. La legibilidad mejorada del código y la facilidad de mantenimiento son ventajas adicionales que contribuyen a la seguridad y calidad de las aplicaciones. En resumen, TypeScript brinda una mayor seguridad en el desarrollo de aplicaciones web dinámicas sin comprometer la agilidad y la eficiencia.

Escalabilidad

TypeScript, como evolución de JavaScript, introduce un sistema de tipado estático que impulsa la seguridad y la calidad del código. Esto se logra al definir y verificar tipos de datos en tiempo de desarrollo, lo que previene errores comunes y facilita la detección temprana de problemas. Además, TypeScript se basa en la Programación Orientada a Objetos (POO), permitiendo a los desarrolladores estructurar sus aplicaciones en torno a objetos reutilizables con conceptos como encapsulación, herencia y polimorfismo. También ofrece un sistema de módulos para organizar el código de manera eficiente y administrar dependencias, lo que contribuye a la claridad y escalabilidad del código. En resumen, TypeScript se erige como una poderosa herramienta que aborda las necesidades del desarrollo web moderno, mejorando la seguridad, organización y escalabilidad de los proyectos.

Descripción de uso SASS/SCSS.

SASS (Syntactically Awesome Stylesheets) y su variante SCSS (Sassy CSS) son extensiones de CSS que ofrecen una mejora significativa en la eficiencia y organización de hojas de estilo en aplicaciones web. Estas tecnologías permiten el uso de variables, anidamiento de selectores, archivos parciales, mixins, herencia y funciones, lo que simplifica la escritura y el mantenimiento de estilos. Además, SASS/SCSS requieren ser compilados a CSS antes de ser implementados en una aplicación web, lo que garantiza la compatibilidad con los navegadores. En resumen, SASS/SCSS son herramientas valiosas para crear hojas de estilo más coherentes y estructuradas, lo que es esencial en proyectos web de cualquier tamaño. A continuación, se describe su uso en general:

Variables

Permiten a los desarrolladores definir valores reutilizables en sus hojas de estilo, lo que simplifica la gestión y personalización de diseños. Al utilizar el signo de dólar (\$) seguido de un nombre y un valor, las variables facilitan la modificación de estilos en todo el proyecto, lo que promueve la consistencia y eficiencia en el desarrollo web. En resumen, las variables en SASS/SCSS son una herramienta esencial para simplificar la escritura y el mantenimiento de hojas de estilo en proyectos web.

Nesting

Es una característica de SASS y SCSS que permite anidar reglas CSS dentro de otras reglas. Esto significa que puedes agrupar selectores de una manera jerárquica y más legible, lo que facilita la escritura de estilos para elementos anidados en el HTML. Al anidar, puedes evitar repetir selectores y, en su lugar, organizar tus estilos de manera más estructurada. Por ejemplo, en lugar de escribir `.contenedor .boton { ... }`, puedes anidar como `.contenedor { .boton { ... } }`. Esto hace que el código sea más claro y fácil de mantener, especialmente en proyectos grandes con reglas CSS complejas. En resumen, el anidamiento en SASS y SCSS es una característica que mejora la organización y la legibilidad del código CSS.

Partials

Son archivos que contienen fragmentos de código CSS o SASS reutilizables que se pueden incluir en otros archivos SASS/SCSS. Estos archivos parciales comienzan con un guion bajo (`_`) en su nombre, como `_botones.scss`, y se importan en otros archivos SASS/SCSS utilizando la directiva `@import`. Los "partials" son útiles para dividir y organizar tu código en componentes lógicos, lo que facilita el mantenimiento y la reutilización de estilos en tu proyecto. Además, cuando se compila el código SASS/SCSS en un archivo CSS, los "partials" no generan archivos CSS separados, ya que se fusionan en el archivo CSS principal durante el proceso de compilación. En resumen, los "partials" son una forma efectiva de modularizar y organizar tu código CSS en SASS/SCSS para un desarrollo más estructurado y eficiente.

Mixins

Los mixins en SASS/SCSS son fragmentos de código reutilizables que contienen un conjunto de estilos CSS o SASS que se pueden incluir en otros selectores o reglas CSS. Estos mixins se definen con la palabra clave `@mixin` y se pueden invocar en diferentes partes del código utilizando la palabra clave `@include`. Los mixins son especialmente útiles para aplicar estilos comunes o propiedades a varios elementos en tu hoja de estilo sin tener que repetir el mismo código una y otra vez. Esto fomenta la reutilización de estilos, reduce la redundancia de código y facilita la actualización y el mantenimiento de los estilos en tu proyecto. Además, los mixins pueden aceptar argumentos, lo que los hace flexibles y adaptables a diferentes situaciones. En resumen, los mixins son una característica poderosa de SASS/SCSS que mejora la modularidad y la eficiencia en la escritura de estilos CSS.

Herencia

Es un selector o regla herede los estilos de otro selector o regla, lo que simplifica la aplicación de estilos comunes a múltiples elementos. A través de la anidación de selectores, un selector hijo puede adquirir automáticamente los estilos definidos en su selector padre, evitando la duplicación de código y mejorando la organización de estilos en proyectos web. Esto resulta en hojas de estilo más limpias y legibles, lo que facilita el mantenimiento y la gestión de estilos en aplicaciones web y contribuye a un desarrollo más eficiente.

Funciones

Son bloques de código reutilizable que toman uno o más valores como entrada, realizan cálculos o transformaciones en esos valores y devuelven un resultado. Las funciones son útiles para realizar operaciones matemáticas, modificar colores, generar estilos condicionales y más. Pueden ayudar a simplificar el código CSS al permitir que ciertas operaciones se realicen de manera más dinámica y eficiente. Algunas funciones incorporadas en SASS/SCSS incluyen operaciones matemáticas, manipulación de colores y generación de listas, y los desarrolladores también pueden crear sus propias funciones personalizadas para abordar necesidades específicas en sus proyectos de diseño web. En resumen, las funciones en SASS/SCSS son una característica poderosa que agiliza el desarrollo de estilos y mejora la mantenibilidad del código CSS resultante.

Compilación

La compilación es un proceso esencial en el desarrollo de software, que implica la traducción del código fuente escrito por programadores en lenguajes de alto nivel a un formato ejecutable por la máquina, como el código de máquina. Esto se logra mediante un compilador que verifica la sintaxis y la semántica del código, generando un archivo binario que la computadora puede entender y ejecutar. La compilación es fundamental para mejorar la eficiencia y el rendimiento de las aplicaciones, detectar errores antes de la ejecución y facilitar la distribución de software en diversas plataformas.