

API Endpoints - MotionParts Ecommerce

Consideraciones de Producción

1. Seguridad

- Validación de permisos de administrador
- Sanitización de datos de entrada
- Logging de operaciones críticas

2. Performance

- Paginación para listas grandes
- Caché de estadísticas
- Operaciones asíncronas para sincronización masiva

3. Mantenimiento

- Monitoreo de errores de sincronización
- Backup de datos antes de operaciones masivas
- Rollback en caso de errores

DOCUMENTACIÓN COMPLETA DE ENDPOINTS

AUTENTICACIÓN Y AUTORIZACIÓN

AuthController (/api/auth)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|-----------|-------------------------|----------|-----------------|--------------------------------|--------------|
| POST | /register | Registrar nuevo usuario | Público | RegisterRequest | Content-Type: application/json | User |
| POST | /login | Iniciar sesión | Público | LoginRequest | Content-Type: application/json | AuthResponse |

Bodies

RegisterRequest:

```
{
  "user": {
    "username": "usuario123",
    "email": "usuario@email.com",
    "password": "password123"
  },
  "userInfo": {
    "type": "natural",
    "documentType": "CC",
    "documentNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  }
}
```

LoginRequest:

```
{
  "username": "usuario123",
  "password": "password123"
}
```

Respuestas

Login Exitoso:

```
{
  "id": "user_id",
  "username": "usuario123",
  "email": "usuario@email.com",
  "role": "USER",
  "token": "jwt_token_here"
}
```

Error de Login:

```
{
  "message": "Nombre de usuario o contraseña incorrectos"
}
```

PasswordRecoveryController (/api/password-recovery)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|-----------|-----------------------------------|----------|-------------------------|--------------------------------|--------------------|
| GET | /test | Prueba de conectividad | Público | - | - | String |
| POST | /validate | Validar usuario para recuperación | Público | PasswordRecoveryRequest | Content-Type: application/json | ValidationResponse |
| POST | /reset | Resetear contraseña | Público | PasswordResetRequest | Content-Type: application/json | ResetResponse |

Bodies

PasswordRecoveryRequest:

```
{
  "username": "usuario123",
  "email": "usuario@email.com"
}
```

PasswordResetRequest:

```
{
  "username": "usuario123",
  "email": "usuario@email.com",
  "newPassword": "nuevaPassword123"
}
```

Respuestas

Validación Exitosa:

```
{
  "success": true,
  "message": "Usuario validado correctamente"
}
```

Reset Exitoso:

```
{
  "success": true,
  "message": "Contraseña actualizada correctamente"
}
```

RoleController (/api/roles)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|---------------------------|----------|------|---------|------------|
| GET | / | Obtener roles disponibles | Público | - | - | List<Role> |

Respuesta

```
[
  {
    "id": 1,
    "name": "ADMIN"
  },
  {
    "id": 2,
    "name": "USER"
  },
  {
    "id": 3,
    "name": "EMPLEADO"
  }
]
```

GESTIÓN DE USUARIOS

UserController (/api/users)

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|-----------------------|-------------------------|---------------------|--------------------------------|---|
| GET | / | Obtener usuarios | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| POST | / | Crear usuario | ADMIN/EMPLEADO | RegisterRequest | Authorization: Bearer <token> , Content-Type: application/json |
| GET | / {id} | Obtener usuario por ID | ADMIN/EMPLEADO/USER | - | Authorization: Bearer <token> |
| PUT | / {id} | Actualizar usuario | ADMIN/EMPLEADO/USER | RegisterRequest | Authorization: Bearer <token> , Content-Type: application/json |
| PUT | / {id} /maia-employee | Asignar empleadold Maia | ADMIN | { "maiaEmployeeId": "string" } | Authorization: Bearer <token> , Content-Type: application/json |
| DELETE | / {id} | Eliminar usuario | ADMIN | - | Authorization: Bearer <token> |

Bodies

RegisterRequest (para crear/actualizar usuario):

```
{
  "user": {
    "username": "usuario123",
    "email": "usuario@email.com",
    "password": "password123"
  },
  "userInfo": {
    "type": "natural",
    "documentType": "CC",
    "documentNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  }
}
```

Asignar Empleado Maia:

```
{
  "maiaEmployeeId": "EMP001"
}
```

Respuestas

Lista de Usuarios:

```
{
  "success": true,
  "message": "Usuarios obtenidos correctamente",
  "data": [
    {
      "id": "user_id",
      "username": "usuario123",
      "email": "usuario@email.com",
      "roles": ["USER"],
      "userInfo": {
        "firstName": "Juan",
        "lastName": "Pérez"
      }
    }
  ]
}
```

Usuario Creado:

```
{
  "success": true,
  "message": "Usuario creado y registrado en sistema contable correctamente",
  "data": {
    "id": "user_id",
    "username": "usuario123",
    "email": "usuario@email.com",
    "maiaClientId": "CLI001"
  },
  "maiaClientId": "CLI001"
}
```

UserInfoController (/api/user-info)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|--------------------------------------|----------|-------------|---|-----------|
| GET | /me | Obtener info del usuario autenticado | USER | - | Authorization: Bearer <token> | UserInfo |
| PUT | /me | Actualizar info del usuario | USER | UserInfoDTO | Authorization: Bearer <token> , Content-Type: application/json | UserInfo |

Bodies

UserInfoDTO:

```
{
  "type": "natural",
  "documentType": "CC",
  "documentNumber": "12345678",
  "firstName": "Juan",
  "lastName": "Pérez",
  "email": "usuario@email.com",
  "phone": "3001234567",
  "address": "Calle 123",
  "city": "Bogotá",
  "country": "Colombia"
}
```

Respuestas

Info del Usuario:

```
{
  "success": true,
  "message": "Información del usuario obtenida correctamente",
  "data": {
    "id": "user_info_id",
    "type": "natural",
    "documentType": "CC",
    "documentNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  }
}
```

GESTIÓN DE PRODUCTOS

ProductController (/api/products)

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|--------------------------|-----------------------------|----------------|-------------------------|---|
| GET | / | Obtener todos los productos | Público | - | - |
| GET | / {id} | Obtener producto por ID | Público | - | - |
| POST | / | Crear producto | ADMIN/EMPLEADO | ProductDTO | Authorization: Bearer <token> , Content-Type: application/json |
| PUT | / {id} | Actualizar producto | ADMIN/EMPLEADO | ProductDTO | Authorization: Bearer <token> , Content-Type: application/json |
| DELETE | / {id} | Eliminar producto | ADMIN | - | Authorization: Bearer <token> |
| POST | / {id}/upload-images | Subir imágenes de galería | ADMIN/EMPLEADO | MultipartFile[] | Authorization: Bearer <token> , Content-Type: multipart/form-data |
| POST | / {id}/upload-main-image | Subir imagen principal | ADMIN/EMPLEADO | MultipartFile | Authorization: Bearer <token> , Content-Type: multipart/form-data |
| PUT | / {id}/main-image | Actualizar imagen principal | ADMIN/EMPLEADO | {"mainImage": "string"} | Authorization: Bearer <token> , Content-Type: application/json |
| POST | / {id}/upload-3d-model | Subir modelo 3D | ADMIN/EMPLEADO | MultipartFile | Authorization: Bearer <token> , Content-Type: multipart/form-data |
| DELETE | / {id}/3d-model | Eliminar modelo 3D | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| GET | /category/{categoryId} | Productos por categoría | Público | - | - |

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|------------------------------|-------------------------------|----------------|------|-------------------------------|
| GET | /subcategory/{subcategoryId} | Productos por subcategoría | Público | - | - |
| POST | /sync/reference/{reference} | Sincronizar producto con Maia | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |

Bodies

ProductDTO:

```
{
  "name": "Alternador Bosch",
  "description": "Alternador de alta calidad para vehículos",
  "price": 150000.0,
  "stock": 50,
  "categoryId": "category_id",
  "subcategoryId": "subcategory_id",
  "reference": "ALT001",
  "brand": "Bosch",
  "model": "ABC123",
  "year": "2020-2023",
  "compatibility": ["Toyota", "Honda", "Nissan"],
  "specifications": {
    "voltage": "12V",
    "amperage": "90A"
  }
}
```

Actualizar Imagen Principal:

```
{
  "mainImage": "nueva_imagen_principal.jpg"
}
```

Respuestas

Producto Creado:

```
{
  "success": true,
  "message": "Producto creado correctamente",
  "data": {
    "id": "product_id",
    "name": "Alternador Bosch",
    "description": "Alternador de alta calidad para vehículos",
    "price": 150000.0,
    "stock": 50,
    "mainImage": "imagen_principal.jpg",
    "gallery": ["imagen1.jpg", "imagen2.jpg"],
    "category": {
      "id": "category_id",
      "name": "Repuestos"
    },
  },
  "subcategory": {
    "id": "subcategory_id",
    "name": "Alternadores"
  }
}
```

Imágenes Subidas:

```
{
  "success": true,
  "message": "Imágenes subidas correctamente",
  "data": ["imagen1.jpg", "imagen2.jpg", "imagen3.jpg"]
}
```

GESTIÓN DE ÓRDENES

OrderController (/api/orders)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------------------|-----------------------------------|---------------------|----------|---|-------------|
| POST | /users | Crear orden (usuario autenticado) | USER | OrderDTO | Authorization: Bearer <token> , Content-Type: application/json | Order |
| POST | /guests | Crear orden (invitado) | Público | OrderDTO | Content-Type: application/json | Order |
| GET | / | Obtener todas las órdenes | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | List<Order> |
| GET | /users/{userId} | Órdenes de un usuario | ADMIN/EMPLEADO/USER | - | Authorization: Bearer <token> | List<Order> |
| GET | / {id} | Obtener orden por ID | ADMIN/EMPLEADO/USER | - | Authorization: Bearer <token> | Order |
| PUT | / {id}/update-status | Actualizar estado de orden | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Order |
| PUT | / {id}/cancel | Cancelar orden | ADMIN/EMPLEADO/USER | - | Authorization: Bearer <token> | Order |

Bodies

OrderDTO:


```
{
  "items": [
    {
      "productId": "product_id",
      "quantity": 2,
      "price": 150000.0
    }
  ],
  "total": 300000.0,
  "billingData": {
    "type": "natural",
    "idType": "CC",
    "idNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  },
  "shippingData": {
    "type": "natural",
    "idType": "CC",
    "idNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  },
  "paymentMethod": "credit_card",
  "shippingMethod": "express"
}
```

Query Parameters

Actualizar Estado:

```
PUT /api/orders/{id}/update-status?status=PENDING
```

Estados disponibles: PENDING , CONFIRMED , IN_PROCESS , SHIPPED , DELIVERED , CANCELLED

Respuestas

Orden Creada:

```
{
  "success": true,
  "message": "Orden creada correctamente",
  "data": {
    "id": "order_id",
    "orderNumber": "ORD-2024-001",
    "status": "PENDING",
    "total": 300000.0,
    "items": [
      {
        "productId": "product_id",
        "productName": "Alternador Bosch",
        "quantity": 2,
        "price": 150000.0
      }
    ],
    "billingData": {
      "firstName": "Juan",
      "lastName": "Pérez",
      "email": "usuario@email.com"
    },
    "shippingData": {
      "firstName": "Juan",
      "lastName": "Pérez",
      "address": "Calle 123"
    },
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Lista de Órdenes:

```
{
  "success": true,
  "message": "Órdenes obtenidas correctamente",
  "data": [
    {
      "id": "order_id",
      "orderNumber": "ORD-2024-001",
      "status": "PENDING",
      "total": 300000.0,
      "createdAt": "2024-01-15T10:30:00Z"
    }
  ]
}
```

CARRO DE COMPRAS

ShoppingCartController (/api/shopping-carts)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Res |
|--------|---------------------|-----------------------------|----------|-------------|---|----------|
| POST | /users/{userId}/add | Agregar producto al carrito | USER | CartItemDTO | Authorization: Bearer <token> , Content-Type: application/json | CartItem |
| GET | /users/{userId} | Obtener carrito del usuario | USER | - | Authorization: Bearer <token> | Shopping |

| Método | Endpoint | Descripción | Permisos | Body | Headers | Res |
|--------|------------------------------------|-------------------------------|----------------|-------------|---|----------|
| PUT | /users/{userId}/update/{productId} | Actualizar cantidad | USER | CartItemDTO | Authorization: Bearer <token> , Content-Type: application/json | CartItem |
| DELETE | /users/{userId}/remove/{productId} | Eliminar producto del carrito | USER | - | Authorization: Bearer <token> | Message |
| DELETE | /users/{userId}/clear | Vaciar carrito | USER | - | Authorization: Bearer <token> | Message |
| GET | //{id} | Buscar carrito por ID | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Shopping |
| POST | / | Crear carrito | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Shopping |
| PUT | //{id}/complete | Completar carrito | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Shopping |
| PUT | //{id}/cancel | Cancelar carrito | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Shopping |
| DELETE | //{id} | Eliminar carrito | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Message |
| GET | //{cartId}/total | Calcular total del carrito | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | Double |

Bodies

CartItemDTO:

```
{
  "productId": "product_id",
  "quantity": 2
}
```

Query Parameters

Crear Carrito:

```
POST /api/shopping-carts?userId=user_id
```

Respuestas

Producto Agregado al Carrito:

```
{
  "success": true,
  "message": "Producto agregado al carrito correctamente",
  "data": {
    "id": "cart_item_id",
    "productId": "product_id",
    "productName": "Alternador Bosch",
    "quantity": 2,
    "price": 150000.0,
    "subtotal": 300000.0
  }
}
```

Carrito del Usuario:

```
{
  "success": true,
  "message": "Carrito obtenido correctamente",
  "data": {
    "id": "cart_id",
    "userId": "user_id",
    "status": "ACTIVE",
    "items": [
      {
        "id": "cart_item_id",
        "productId": "product_id",
        "productName": "Alternador Bosch",
        "quantity": 2,
        "price": 150000.0,
        "subtotal": 300000.0
      }
    ],
    "total": 300000.0,
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Total del Carrito:

```
{
  "success": true,
  "message": "Total calculado correctamente",
  "data": 300000.0
}
```

CATEGORÍAS Y SUBCATEGORÍAS

CategoryController (/api/categories)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|------------------------------|----------------|-------------|---|-----------------|
| GET | / | Obtener todas las categorías | Público | - | - | List<Category> |
| GET | / {id} | Obtener categoría por ID | Público | - | - | Category |
| POST | / | Crear categoría | ADMIN/EMPLEADO | CategoryDTO | Authorization: Bearer <token> , Content-Type: application/json | Category |
| PUT | / {id} | Actualizar categoría | ADMIN/EMPLEADO | CategoryDTO | Authorization: Bearer <token> , Content-Type: application/json | Category |
| DELETE | / {id} | Eliminar categoría | ADMIN | - | Authorization: Bearer <token> | MessageResponse |

Bodies

CategoryDTO:

```
{
  "name": "Repuestos",
  "description": "Repuestos automotrices",
  "image": "categoria_imagen.jpg"
}
```

Respuestas

Categoría Creada:

```
{
  "success": true,
  "message": "Categoría creada correctamente",
  "data": {
    "id": "category_id",
    "name": "Repuestos",
    "description": "Repuestos automotrices",
    "image": "categoria_imagen.jpg",
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Lista de Categorías:

```
{
  "success": true,
  "message": "Categorías obtenidas correctamente",
  "data": [
    {
      "id": "category_id",
      "name": "Repuestos",
      "description": "Repuestos automotrices",
      "image": "categoria_imagen.jpg",
      "subcategories": [
        {
          "id": "subcategory_id",
          "name": "Alternadores"
        }
      ]
    }
  ]
}
```

SubcategoryController (/api/subcategories)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respue |
|--------|---------------------------|---------------------------------|----------------|----------------|--|-------------|
| GET | / | Obtener todas las subcategorías | Público | - | - | List<Subcat |
| GET | /by-category/{categoryId} | Subcategorías por categoría | Público | - | - | List<Subcat |
| GET | /id | Obtener subcategoría por ID | Público | - | - | Subcategory |
| POST | / | Crear subcategoría | ADMIN/EMPLEADO | SubcategoryDTO | Authorization: Bearer <token> , Content-Type: application/json | Subcategory |

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respue |
|--------|----------|-------------------------|----------------|----------------|---|-------------|
| PUT | / {id} | Actualizar subcategoría | ADMIN/EMPLEADO | SubcategoryDTO | Authorization: Bearer <token> , Content-Type: application/json | Subcategory |
| DELETE | / {id} | Eliminar subcategoría | ADMIN | - | Authorization: Bearer <token> | MessageResp |

Bodies

SubcategoryDTO:

```
{
  "name": "Alternadores",
  "description": "Alternadores para vehículos",
  "categoryId": "category_id",
  "image": "subcategoria_imagen.jpg"
}
```

Respuestas

Subcategoría Creada:

```
{
  "success": true,
  "message": "Subcategoría creada correctamente",
  "data": {
    "id": "subcategory_id",
    "name": "Alternadores",
    "description": "Alternadores para vehículos",
    "categoryId": "category_id",
    "image": "subcategoria_imagen.jpg",
    "category": {
      "id": "category_id",
      "name": "Repuestos"
    }
  },
  "createdAt": "2024-01-15T10:30:00Z"
}
```

Subcategorías por Categoría:

```
{
  "success": true,
  "message": "Subcategorías obtenidas correctamente",
  "data": [
    {
      "id": "subcategory_id",
      "name": "Alternadores",
      "description": "Alternadores para vehículos",
      "image": "subcategoria_imagen.jpg"
    }
  ]
}
```

ENVÍOS

ShippingController (/api/shippings)

| Método | Endpoint | Descripción | Permisos | Body | Header |
|--------|-----------------------|---------------------------|----------------|---|-------------------------------------|
| GET | / | Obtener todos los envíos | ADMIN/EMPLEADO | - | Authorization: |
| GET | / {id} | Obtener envío por ID | ADMIN/EMPLEADO | - | Authorization: |
| GET | /order/{orderId} | Envío por orden | ADMIN/EMPLEADO | - | Authorization: |
| PUT | / {id} | Actualizar envío | ADMIN/EMPLEADO | ShippingDTO | Authorization: Content-Type: a |
| PUT | / {id}/assign-company | Asignar compañía y estado | ADMIN/EMPLEADO | { "shippingCompanyId": "string", "status": "string" } | Authorization: Content-Type: a |

Bodies

ShippingDTO:

```
{
  "orderId": "order_id",
  "shippingCompanyId": "company_id",
  "trackingNumber": "TRK123456789",
  "status": "IN_TRANSIT",
  "estimatedDelivery": "2024-01-20T10:30:00Z",
  "shippingCost": 15000.0,
  "notes": "Envío express"
}
```

Asignar Compañía:

```
{
  "shippingCompanyId": "company_id",
  "status": "IN_TRANSIT"
}
```

Respuestas

Envío Creado:

```
{
  "success": true,
  "message": "Envío actualizado correctamente",
  "data": {
    "id": "shipping_id",
    "orderId": "order_id",
    "orderNumber": "ORD-2024-001",
    "shippingCompany": {
      "id": "company_id",
      "name": "Servientrega"
    },
    "trackingNumber": "TRK123456789",
    "status": "IN_TRANSIT",
    "estimatedDelivery": "2024-01-20T10:30:00Z",
    "shippingCost": 15000.0,
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Lista de Envíos:

```
{
  "success": true,
  "message": "Envíos obtenidos correctamente",
  "data": [
    {
      "id": "shipping_id",
      "orderNumber": "ORD-2024-001",
      "trackingNumber": "TRK123456789",
      "status": "IN_TRANSIT",
      "shippingCompany": {
        "name": "Servientrega"
      }
    }
  ]
}
```

ShippingCompanyController (/api/shipping-companies)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|-------------------------|----------------|--------------------|---|-----------------------|
| POST | / | Crear compañía de envío | ADMIN/EMPLEADO | ShippingCompanyDTO | Authorization: Bearer <token> , Content-Type: application/json | ShippingCompany |
| GET | / | Obtener compañías | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | List<ShippingCompany> |
| GET | / {id} | Obtener compañía por ID | ADMIN/EMPLEADO | - | Authorization: Bearer <token> | ShippingCompany |
| PUT | / {id} | Actualizar compañía | ADMIN/EMPLEADO | ShippingCompanyDTO | Authorization: Bearer <token> , Content-Type: application/json | ShippingCompany |
| DELETE | / {id} | Eliminar compañía | ADMIN | - | Authorization: Bearer <token> | MessageResponse |

Bodies

ShippingCompanyDTO:


```
{
  "name": "Servientrega",
  "description": "Empresa de envíos nacionales",
  "trackingUrl": "https://www.servientrega.com/rastreo",
  "phone": "3001234567",
  "email": "contacto@servientrega.com",
  "address": "Calle 123, Bogotá",
  "isActive": true
}
```

Query Parameters

Buscar por Nombre:

```
GET /api/shipping-companies?name=Servientrega
```

Respuestas

Compañía Creada:

```
{
  "success": true,
  "message": "Compañía de envío creada correctamente",
  "data": {
    "id": "company_id",
    "name": "Servientrega",
    "description": "Empresa de envíos nacionales",
    "trackingUrl": "https://www.servientrega.com/rastreo",
    "phone": "3001234567",
    "email": "contacto@servientrega.com",
    "address": "Calle 123, Bogotá",
    "isActive": true,
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Lista de Compañías:

```
{
  "success": true,
  "message": "Compañías de envío obtenidas correctamente",
  "data": [
    {
      "id": "company_id",
      "name": "Servientrega",
      "description": "Empresa de envíos nacionales",
      "trackingUrl": "https://www.servientrega.com/rastreo",
      "isActive": true
    }
  ]
}
```

VEHÍCULOS

VehicleController (/api/vehicles)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|-----------------------------|----------|------|---------|---------------|
| GET | / | Obtener todos los vehículos | Público | - | - | List<Vehicle> |

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|----------|-------------------------|----------------|------------|---|-----------------|
| GET | / {id} | Obtener vehículo por ID | Público | - | - | Vehicle |
| POST | / | Crear vehículo | ADMIN/EMPLEADO | VehicleDTO | Authorization: Bearer <token> , Content-Type: application/json | Vehicle |
| PUT | / {id} | Actualizar vehículo | ADMIN/EMPLEADO | VehicleDTO | Authorization: Bearer <token> , Content-Type: application/json | Vehicle |
| DELETE | / {id} | Eliminar vehículo | ADMIN | - | Authorization: Bearer <token> | MessageResponse |

Bodies

VehicleDTO:

```
{
  "brand": "Toyota",
  "model": "Corolla",
  "year": 2020,
  "engine": "2.0L",
  "transmission": "Automatic",
  "fuelType": "Gasoline",
  "description": "Sedán compacto confiable"
}
```

Respuestas

Vehículo Creado:

```
{
  "success": true,
  "message": "Vehículo creado correctamente",
  "data": {
    "id": "vehicle_id",
    "brand": "Toyota",
    "model": "Corolla",
    "year": 2020,
    "engine": "2.0L",
    "transmission": "Automatic",
    "fuelType": "Gasoline",
    "description": "Sedán compacto confiable",
    "createdAt": "2024-01-15T10:30:00Z"
  }
}
```

Lista de Vehículos:

```
{
  "success": true,
  "message": "Vehículos obtenidos correctamente",
  "data": [
    {
      "id": "vehicle_id",
      "brand": "Toyota",
      "model": "Corolla",
      "year": 2020,
      "engine": "2.0L",
      "transmission": "Automatic",
      "fuelType": "Gasoline"
    }
  ]
}
```

NOTIFICACIONES

NotificationController (/api/notifications)

| Método | Endpoint | Descripción | Permisos | Body | Headers | Respuesta |
|--------|--------------|----------------------------------|----------|------|-------------------------------|--------------------|
| GET | / | Obtener todas las notificaciones | ADMIN | - | Authorization: Bearer <token> | List<Notification> |
| PUT | / {id} /read | Marcar como leída | ADMIN | - | Authorization: Bearer <token> | Notification |
| DELETE | / {id} | Eliminar notificación | ADMIN | - | Authorization: Bearer <token> | MessageResponse |

Respuestas

Lista de Notificaciones:

```
{
  "success": true,
  "message": "Notificaciones obtenidas correctamente",
  "data": [
    {
      "id": "notification_id",
      "title": "Nueva orden recibida",
      "message": "Se ha recibido una nueva orden #ORD-2024-001",
      "type": "ORDER",
      "isRead": false,
      "createdAt": "2024-01-15T10:30:00Z"
    }
  ]
}
```

Notificación Marcada como Leída:

```
{
  "success": true,
  "message": "Notificación marcada como leída",
  "data": {
    "id": "notification_id",
    "title": "Nueva orden recibida",
    "message": "Se ha recibido una nueva orden #ORD-2024-001",
    "type": "ORDER",
    "isRead": true,
    "readAt": "2024-01-15T11:00:00Z"
  }
}
```

INTEGRACIÓN MAIA - CLIENTES

MaiaClientController (/api/maia/clients)

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|-----------------------------------|---------------------------------------|----------------|------------------------------|--|
| GET | /search/document/{documentNumber} | Buscar cliente por documento | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| GET | /search/email/{email} | Buscar cliente por email | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| GET | /list | Obtener lista de clientes Maia | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| POST | /sync/document/{documentNumber} | Sincronizar cliente por documento | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| POST | /sync/email/{email} | Sincronizar cliente por email | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| GET | /sync/status/{documentNumber} | Estado de sincronización | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| POST | /maiaClientId/orders | Crear orden para cliente Maia | ADMIN/EMPLEADO | OrderDTO | Authorization: Bearer <t Content-Type: applicatio |
| POST | /sync/unsynchronized | Sincronizar usuarios no sincronizados | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| GET | /sync/stats | Estadísticas de sincronización | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| POST | /register-in-maia | Registrar usuario en Maia | ADMIN/EMPLEADO | {"documentNumber": "string"} | Authorization: Bearer <t Content-Type: applicatio |
| GET | /unsynchronized | Usuarios no sincronizados | ADMIN/EMPLEADO | - | Authorization: Bearer <t |
| GET | /synchronized | Usuarios sincronizados | ADMIN/EMPLEADO | - | Authorization: Bearer <t |

Bodies

Registrar en Maia:

```
{  
  "documentNumber": "12345678"  
}
```

Respuestas

Cliente Maia Encontrado:

```
{
  "success": true,
  "message": "Cliente encontrado en Maia",
  "data": {
    "id": "maia_client_id",
    "documentNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123, Bogotá"
  }
}
```

Sincronización Exitosa:

```
{
  "success": true,
  "message": "Usuario sincronizado correctamente con Maia",
  "data": {
    "userId": "user_id",
    "maiaClientId": "CLI001",
    "syncDate": "2024-01-15T10:30:00Z"
  }
}
```

Estadísticas de Sincronización:

```
{
  "success": true,
  "message": "Estadísticas obtenidas correctamente",
  "data": {
    "totalUsers": 100,
    "synchronizedUsers": 85,
    "unsynchronizedUsers": 15,
    "syncRate": 85.0
  }
}
```

INTEGRACIÓN MAIA - PRODUCTOS

MaiaProductController (/api/maia/products)

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|-------------------------------|--------------------------------------|----------------|------|-------------------------------|
| GET | /search/reference/{reference} | Buscar producto por referencia | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| GET | /list | Obtener lista de productos Maia | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| POST | /sync/reference/{reference} | Sincronizar producto por referencia | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| GET | /sync/status/{reference} | Estado de sincronización de producto | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |

| Método | Endpoint | Descripción | Permisos | Body | Headers |
|--------|----------------------|---|----------------|-------------------------|---|
| POST | /register-in-maia | Registrar producto en Maia | ADMIN/EMPLEADO | {"reference": "string"} | Authorization: Bearer <token> , Content-Type: application/json |
| GET | /sync/stats | Estadísticas de sincronización de productos | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |
| POST | /sync/unsynchronized | Sincronización masiva de productos | ADMIN/EMPLEADO | - | Authorization: Bearer <token> |

Bodies

Registrar Producto en Maia:

```
{
  "reference": "ALT001"
}
```

Respuestas

Producto Maia Encontrado:

```
{
  "success": true,
  "message": "Producto encontrado en Maia",
  "data": {
    "id": "maia_product_id",
    "reference": "ALT001",
    "name": "Alternador Bosch",
    "description": "Alternador de alta calidad",
    "price": 150000.0,
    "stock": 50,
    "category": "Repuestos",
    "brand": "Bosch"
  }
}
```

Sincronización de Producto Exitosa:

```
{
  "success": true,
  "message": "Producto sincronizado correctamente con Maia",
  "data": {
    "productId": "product_id",
    "maiaProductId": "PROD001",
    "reference": "ALT001",
    "syncDate": "2024-01-15T10:30:00Z"
  }
}
```

Estadísticas de Productos:

```
{
  "success": true,
  "message": "Estadísticas obtenidas correctamente",
  "data": {
    "totalProducts": 500,
    "synchronizedProducts": 450,
    "unsynchronizedProducts": 50,
    "syncRate": 90.0
  }
}
```

RESPUESTAS ESTÁNDAR

Formato de Respuesta Exitosa

```
{
  "success": true,
  "message": "Operación realizada correctamente",
  "data": { ... }
}
```

Formato de Respuesta de Error

```
{
  "success": false,
  "message": "Descripción del error"
}
```

Códigos de Estado HTTP

| Código | Descripción | Uso |
|--------|-----------------------|----------------------------------|
| 200 | OK | Operación exitosa |
| 201 | Created | Recurso creado exitosamente |
| 400 | Bad Request | Datos de entrada inválidos |
| 401 | Unauthorized | Token de autenticación requerido |
| 403 | Forbidden | Permisos insuficientes |
| 404 | Not Found | Recurso no encontrado |
| 409 | Conflict | Conflicto de datos |
| 500 | Internal Server Error | Error interno del servidor |

Ejemplos de Errores Comunes

Error de Validación:

```
{
  "success": false,
  "message": "Datos de entrada inválidos",
  "errors": [
    {
      "field": "email",
      "message": "El email debe tener un formato válido"
    },
    {
      "field": "password",
      "message": "La contraseña debe tener al menos 8 caracteres"
    }
  ]
}
```

Error de Autenticación:

```
{
  "success": false,
  "message": "Token de autenticación inválido o expirado"
}
```

Error de Permisos:

```
{
  "success": false,
  "message": "No tienes permisos para realizar esta operación"
}
```

Error de Recurso No Encontrado:

```
{
  "success": false,
  "message": "Producto con ID 'product_id' no encontrado"
}
```

AUTENTICACIÓN

Headers Requeridos

Authorization: Bearer <JWT_TOKEN>
Content-Type: application/json

Obtención del Token JWT

Para obtener un token JWT, debes autenticarte usando el endpoint de login:

POST /api/auth/login
Content-Type: application/json

```
{
  "username": "usuario123",
  "password": "password123"
}
```

Respuesta:


```
{
  "id": "user_id",
  "username": "usuario123",
  "email": "usuario@email.com",
  "role": "USER",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Uso del Token

Una vez obtenido el token, inclúyelo en el header `Authorization` de todas las peticiones que requieran autenticación:

```
GET /api/users/me
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Roles Disponibles

| Rol | Descripción | Permisos |
|----------|---------------|---|
| ADMIN | Administrador | Acceso completo a todas las funcionalidades |
| EMPLEADO | Empleado | Gestión de productos, órdenes, clientes y envíos |
| USER | Usuario | Acceso limitado a su propia información y compras |

Vida Útil del Token

- **Duración:** 24 horas (86400000 ms)
- **Renovación:** Se debe hacer login nuevamente para obtener un nuevo token
- **Expiración:** Después de 24 horas, el token se invalida automáticamente

Endpoints Públicos

Los siguientes endpoints no requieren autenticación:

- `POST /api/auth/register` - Registro de usuarios
- `POST /api/auth/login` - Inicio de sesión
- `GET /api/roles` - Obtener roles disponibles
- `GET /api/products` - Obtener productos
- `GET /api/products/{id}` - Obtener producto por ID
- `GET /api/categories` - Obtener categorías
- `GET /api/subcategories` - Obtener subcategorías
- `GET /api/vehicles` - Obtener vehículos
- `POST /api/orders/guests` - Crear orden como invitado
- `GET /api/password-recovery/test` - Prueba de conectividad
- `POST /api/password-recovery/validate` - Validar usuario para recuperación
- `POST /api/password-recovery/reset` - Resetear contraseña

CONFIGURACIÓN

Variables de Entorno

```
# API Maia
maia.api.enabled=true
maia.api.base-url=http://localhost:8080/cloudmaiaerp
maia.api.user=testuser
maia.api.passwd=testpass

# JWT
app.jwt.secret=EGMD2sX+K8Zp0Mb500d6B2TCai4qwkRw8V/LJwvHrtI=
app.jwt.expiration=86400000

# MongoDB
spring.data.mongodb.uri=mongodb://localhost:27017/motionparts
```

NOTAS IMPORTANTES

1. **Integración Maia:** La aplicación incluye integración completa con el sistema contable Maia para sincronización bidireccional de clientes y productos.
2. **Mock Controller:** En desarrollo, se usa `MaiaMockController` que simula la API de Maia con datos de prueba.
3. **Seguridad:** Todos los endpoints sensibles requieren autenticación JWT y validación de roles.
4. **Archivos:** Los endpoints de subida de archivos soportan hasta 50MB por archivo.
5. **Respuestas:** Todas las respuestas siguen un formato estándar con campos `success`, `message` y `data`.

PRÓXIMOS PASOS

1. **Documentación Postman:** Importar esta documentación a Postman para pruebas automatizadas.
2. **Validación:** Probar todos los endpoints con datos reales.
3. **Producción:** Configurar variables de entorno para producción.
4. **Monitoreo:** Implementar logging y monitoreo de endpoints críticos.

EJEMPLOS DE USO

Flujo Completo de Compra

1. **Registro de Usuario:**

POST /api/auth/register
Content-Type: application/json

```
{
  "user": {
    "username": "nuevo_usuario",
    "email": "usuario@email.com",
    "password": "password123"
  },
  "userInfo": {
    "type": "natural",
    "documentType": "CC",
    "documentNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  }
}
```

2. Login:

POST /api/auth/login
Content-Type: application/json

```
{
  "username": "nuevo_usuario",
  "password": "password123"
}
```

3. Agregar Producto al Carrito:

POST /api/shopping-carts/users/{userId}/add
Authorization: Bearer <token>
Content-Type: application/json

```
{
  "productId": "product_id",
  "quantity": 2
}
```

4. Crear Orden:

```
POST /api/orders/users
Authorization: Bearer <token>
Content-Type: application/json
```

```
{
  "items": [
    {
      "productId": "product_id",
      "quantity": 2,
      "price": 150000.0
    }
  ],
  "total": 300000.0,
  "billingData": {
    "type": "natural",
    "idType": "CC",
    "idNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  },
  "shippingData": {
    "type": "natural",
    "idType": "CC",
    "idNumber": "12345678",
    "firstName": "Juan",
    "lastName": "Pérez",
    "email": "usuario@email.com",
    "phone": "3001234567",
    "address": "Calle 123",
    "city": "Bogotá",
    "country": "Colombia"
  },
  "paymentMethod": "credit_card",
  "shippingMethod": "express"
}
```

Gestión de Productos (Admin)

1. Crear Categoría:

```
POST /api/categories
Authorization: Bearer <admin_token>
Content-Type: application/json
```

```
{
  "name": "Repuestos Eléctricos",
  "description": "Repuestos para el sistema eléctrico del vehículo",
  "image": "categoria_electrica.jpg"
}
```

2. Crear Subcategoría:

```
POST /api/subcategories
Authorization: Bearer <admin_token>
Content-Type: application/json
```

```
{
  "name": "Alternadores",
  "description": "Alternadores para diferentes marcas de vehículos",
  "categoryId": "category_id",
  "image": "alternadores.jpg"
}
```

3. Crear Producto:

```
POST /api/products
Authorization: Bearer <admin_token>
Content-Type: application/json
```

```
{
  "name": "Alternador Bosch Premium",
  "description": "Alternador de alta calidad para vehículos modernos",
  "price": 180000.0,
  "stock": 25,
  "categoryId": "category_id",
  "subcategoryId": "subcategory_id",
  "reference": "ALT-BOSCH-001",
  "brand": "Bosch",
  "model": "ABC123",
  "year": "2020-2023",
  "compatibility": ["Toyota", "Honda", "Nissan"],
  "specifications": {
    "voltage": "12V",
    "amperage": "90A",
    "warranty": "2 años"
  }
}
```

4. Subir Imágenes:

```
POST /api/products/{productId}/upload-images
Authorization: Bearer <admin_token>
Content-Type: multipart/form-data
```

```
# Archivos: imagen1.jpg, imagen2.jpg, imagen3.jpg
```

Integración con Maia

1. Sincronizar Cliente:

```
POST /api/maia/clients/sync/document/12345678
Authorization: Bearer <admin_token>
```

2. Sincronizar Producto:

```
POST /api/maia/products/sync/reference/ALT-BOSCH-001
Authorization: Bearer <admin_token>
```

3. Ver Estadísticas:

```
GET /api/maia/clients/sync/stats
Authorization: Bearer <admin_token>
```

Conclusión

La implementación proporciona una solución completa y robusta para la sincronización bidireccional de productos entre Maia y el ecommerce, siguiendo las mejores prácticas de desarrollo y manteniendo la consistencia con la funcionalidad existente de sincronización de usuarios.

El sistema está listo para pruebas en desarrollo y puede ser fácilmente adaptado para producción con la configuración apropiada de la API real de Maia.

Características Principales

- **Autenticación JWT completa**
- **Gestión de roles y permisos**
- **CRUD completo de productos, categorías y usuarios**
- **Sistema de carrito de compras**
- **Gestión de órdenes y envíos**
- **Integración bidireccional con Maia**
- **Sincronización de clientes y productos**
- **Sistema de notificaciones**
- **Recuperación de contraseñas**
- **Documentación completa de API**

Tecnologías Utilizadas

- **Backend:** Spring Boot, Spring Security, JWT
- **Base de Datos:** MongoDB
- **Integración:** REST API con sistema Maia
- **Documentación:** Markdown con ejemplos completos