



Documentação de Rotas - MercadoPlace API



Informações Gerais

- **Base URL:** `http://localhost:3001`
- **Documentação Swagger:** `http://localhost:3001/api`
- **Autenticação:** JWT Bearer Token
- **Formato de Dados:** JSON



Autenticação (/auth)

POST `/auth/register`

Descrição: Registrar um novo usuário

Autorização: Nenhuma

Payload:

```
{
  "email": "user@example.com",
  "password": "password123",
  "name": "Nome do Usuário",
  "latitude": -15.7942,
  "longitude": -47.8822
}
```

Resposta (201):

```
{
  "user": {
    "id": "uuid",
    "email": "user@example.com",
    "name": "Nome do Usuário",
    "isAdmin": false,
    "latitude": -15.7942,
    "longitude": -47.8822,
    "points": 0,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  },
  "access_token": "jwt_token_aqui"
}
```

Erros:

- **401** - Email já está em uso

POST /auth/login

Descrição: Fazer login

Autorização: Nenhuma

Payload:

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

Resposta (200):

```
{
  "user": {
    "id": "uuid",
    "email": "user@example.com",
    "name": "Nome do Usuário",
    "isAdmin": false,
    "latitude": -15.7942,
    "longitude": -47.8822,
    "points": 0,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  },
  "access_token": "jwt_token_aqui"
}
```

Erros:

- 401 - Credenciais inválidas

GET /auth/profile

Descrição: Obter perfil do usuário logado

Autorização: JWT Bearer Token

Resposta (200):

```
{
  "id": "uuid",
  "email": "user@example.com",
  "name": "Nome do Usuário",
  "isAdmin": false,
  "latitude": -15.7942,
  "longitude": -47.8822,
  "points": 0,
  "createdAt": "2024-01-01T00:00:00.000Z",
}
```

```
{  "updatedAt": "2024-01-01T00:00:00.000Z"}
```

Erros:

- **401** - Token inválido

Usuários (/users)

GET /users/profile

Descrição: Obter perfil do usuário logado

Autorização: JWT Bearer Token

Resposta (200):

```
{  "id": "uuid",  "email": "user@example.com",  "name": "Nome do Usuário",  "isAdmin": false,  "latitude": -15.7942,  "longitude": -47.8822,  "points": 0,  "createdAt": "2024-01-01T00:00:00.000Z",  "updatedAt": "2024-01-01T00:00:00.000Z"}
```

GET /users/:id

Descrição: Obter usuário por ID

Autorização: JWT Bearer Token

Parâmetros:

- **id** (string) - ID do usuário **Resposta (200):**

```
{  "id": "uuid",  "email": "user@example.com",  "name": "Nome do Usuário",  "isAdmin": false,  "latitude": -15.7942,  "longitude": -47.8822,  "points": 0,  "createdAt": "2024-01-01T00:00:00.000Z",  "updatedAt": "2024-01-01T00:00:00.000Z"}
```

Erros:

- 404 - Usuário não encontrado

PATCH /users/profile

Descrição: Atualizar perfil do usuário logado

Autorização: JWT Bearer Token

Payload:

```
{
  "name": "Novo Nome",
  "latitude": -15.7942,
  "longitude": -47.8822
}
```

Resposta (200):

```
{
  "id": "uuid",
  "email": "user@example.com",
  "name": "Novo Nome",
  "isAdmin": false,
  "latitude": -15.7942,
  "longitude": -47.8822,
  "points": 0,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Usuário não encontrado

GET /users/purchase-history/me

Descrição: Obter histórico de compras do usuário logado

Autorização: JWT Bearer Token

Resposta (200):

```
[
  {
    "id": "uuid",
    "totalCost": 45.50,
    "userId": "uuid",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "purchasedItems": [
      {
```

```
    "id": "uuid",
    "productName": "Produto A",
    "priceAtTime": 15.50,
    "quantity": 2,
    "purchaseId": "uuid",
    "productId": "uuid"
  }
]
```

Produtos (/products)

POST /products

Descrição: Criar um novo produto

Autorização: JWT Bearer Token

Payload:

```
{
  "name": "Nome do Produto",
  "description": "Descrição do produto",
  "currentPrice": 15.99,
  "imageUrl": "https://example.com/image.jpg"
}
```

Resposta (201):

```
{
  "id": "uuid",
  "name": "Nome do Produto",
  "description": "Descrição do produto",
  "currentPrice": 15.99,
  "imageUrl": "https://example.com/image.jpg",
  "isValid": false,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 400 - Produto com este nome já existe

GET /products

Descrição: Listar produtos (com filtros e ordenação)

Autorização: Nenhuma

Query Parameters:

- **name** (opcional) - Nome do produto para buscar
- **sortBy** (opcional) - Campo para ordenação (**price**, **name**)
- **sortOrder** (opcional) - Direção da ordenação (**asc**, **desc**)
- **userLatitude** (opcional) - Latitude para ordenação por distância
- **userLongitude** (opcional) - Longitude para ordenação por distância

Resposta (200):

```
[
  {
    "id": "uuid",
    "name": "Nome do Produto",
    "description": "Descrição do produto",
    "currentPrice": 15.99,
    "imageUrl": "https://example.com/image.jpg",
    "isValid": true,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET `/products/pending`

Descrição: Listar produtos pendentes de aprovação (Admin)

Autorização: JWT Bearer Token + Admin

Resposta (200):

```
[
  {
    "id": "uuid",
    "name": "Produto Pendente",
    "description": "Descrição do produto",
    "currentPrice": 15.99,
    "imageUrl": "https://example.com/image.jpg",
    "isValid": false,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

Erros:

- **403** - Acesso negado

GET `/products/market/:marketId`

Descrição: Listar produtos de um mercado específico

Autorização: Nenhuma

Parâmetros:

- **marketId** (string) - ID do mercado **Resposta (200):**

```
[
  {
    "id": "uuid",
    "name": "Nome do Produto",
    "description": "Descrição do produto",
    "currentPrice": 15.99,
    "imageUrl": "https://example.com/image.jpg",
    "isValid": true,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

Erros:

- **404** - Mercado não encontrado

GET **/products/:id**

Descrição: Obter produto por ID

Autorização: Nenhuma

Parâmetros:

- **id** (string) - ID do produto **Resposta (200):**

```
{
  "id": "uuid",
  "name": "Nome do Produto",
  "description": "Descrição do produto",
  "currentPrice": 15.99,
  "imageUrl": "https://example.com/image.jpg",
  "isValid": true,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- **404** - Produto não encontrado

PATCH **/products/:id**

Descrição: Atualizar produto

Autorização: JWT Bearer Token

Parâmetros:

- **id** (string) - ID do produto **Payload:**

```
{
  "name": "Novo Nome",
  "description": "Nova descrição",
  "currentPrice": 19.99,
  "imageUrl": "https://example.com/new-image.jpg"
}
```

Resposta (200):

```
{
  "id": "uuid",
  "name": "Novo Nome",
  "description": "Nova descrição",
  "currentPrice": 19.99,
  "imageUrl": "https://example.com/new-image.jpg",
  "isValid": true,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- **404** - Produto não encontrado

PATCH **/products/:id/approve**

Descrição: Aprovar produto (Admin)

Autorização: JWT Bearer Token + Admin

Parâmetros:

- **id** (string) - ID do produto **Resposta (200):**

```
{
  "id": "uuid",
  "name": "Nome do Produto",
  "description": "Descrição do produto",
  "currentPrice": 15.99,
  "imageUrl": "https://example.com/image.jpg",
  "isValid": true,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```


Erros:

- 404 - Produto não encontrado
- 403 - Acesso negado

DELETE `/products/:id/reject`**Descrição:** Reprovar produto (Admin)**Autorização:** JWT Bearer Token + Admin**Parâmetros:**

- `id` (string) - ID do produto **Resposta (200):**

```
{
  "message": "Produto reprovado e removido"
}
```

Erros:

- 404 - Produto não encontrado
- 403 - Acesso negado

DELETE `/products/:id`**Descrição:** Remover produto (Admin)**Autorização:** JWT Bearer Token + Admin**Parâmetros:**

- `id` (string) - ID do produto **Resposta (200):**

```
{
  "message": "Produto removido com sucesso"
}
```

Erros:

- 404 - Produto não encontrado
- 403 - Acesso negado

**Mercados** (`/markets`)POST `/markets`**Descrição:** Criar um novo mercado (Admin)**Autorização:** JWT Bearer Token + Admin

Payload:

```
{
  "name": "Nome do Mercado",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Endereço do mercado",
  "phone": "(11) 99999-9999",
  "description": "Descrição do mercado",
  "imageUrl": "https://example.com/image.jpg"
}
```

Resposta (201):

```
{
  "id": "uuid",
  "name": "Nome do Mercado",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Endereço do mercado",
  "phone": "(11) 99999-9999",
  "description": "Descrição do mercado",
  "imageUrl": "https://example.com/image.jpg",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 403 - Acesso negado

GET /markets

Descrição: Listar todos os mercados

Autorização: Nenhuma

Resposta (200):

```
[
  {
    "id": "uuid",
    "name": "Nome do Mercado",
    "latitude": -15.7942,
    "longitude": -47.8822,
    "address": "Endereço do mercado",
    "phone": "(11) 99999-9999",
    "description": "Descrição do mercado",
    "imageUrl": "https://example.com/image.jpg",
    "createdAt": "2024-01-01T00:00:00.000Z",
  }
]
```

```
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /markets/nearby

Descrição: Buscar mercados próximos

Autorização: Nenhuma

Query Parameters:

- **latitude** (obrigatório) - Latitude da localização
- **longitude** (obrigatório) - Longitude da localização
- **radius** (opcional) - Raio em km (padrão: 10)

Resposta (200):

```
[
  {
    "id": "uuid",
    "name": "Nome do Mercado",
    "latitude": -15.7942,
    "longitude": -47.8822,
    "address": "Endereço do mercado",
    "phone": "(11) 99999-9999",
    "description": "Descrição do mercado",
    "imageUrl": "https://example.com/image.jpg",
    "distance": 5.2,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /markets/:id

Descrição: Obter mercado por ID

Autorização: Nenhuma

Parâmetros:

- **id** (string) - ID do mercado **Resposta (200):**

```
{
  "id": "uuid",
  "name": "Nome do Mercado",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Endereço do mercado",
  "phone": "(11) 99999-9999",
  "description": "Descrição do mercado",
```

```
"imageUrl": "https://example.com/image.jpg",
"createdAt": "2024-01-01T00:00:00.000Z",
"updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Mercado não encontrado

PATCH /markets/:id

Descrição: Atualizar mercado (Admin)

Autorização: JWT Bearer Token + Admin

Parâmetros:

- id (string) - ID do mercado **Payload:**

```
{
  "name": "Novo Nome",
  "address": "Novo endereço",
  "phone": "(11) 88888-8888",
  "description": "Nova descrição",
  "imageUrl": "https://example.com/new-image.jpg"
}
```

Resposta (200):

```
{
  "id": "uuid",
  "name": "Novo Nome",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Novo endereço",
  "phone": "(11) 88888-8888",
  "description": "Nova descrição",
  "imageUrl": "https://example.com/new-image.jpg",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Mercado não encontrado
- 403 - Acesso negado

DELETE /markets/:id

Descrição: Remover mercado (Admin)

Autorização: JWT Bearer Token + Admin

Parâmetros:

- **id** (string) - ID do mercado **Resposta (200):**

```
{
  "message": "Mercado removido com sucesso"
}
```

Erros:

- **404** - Mercado não encontrado
- **403** - Acesso negado

POST **/markets/:id/products/:productId**

Descrição: Adicionar produto ao mercado (Admin)

Autorização: JWT Bearer Token + Admin

Parâmetros:

- **id** (string) - ID do mercado
- **productId** (string) - ID do produto **Resposta (200):**

```
{
  "id": "uuid",
  "name": "Nome do Mercado",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Endereço do mercado",
  "phone": "(11) 99999-9999",
  "description": "Descrição do mercado",
  "imageUrl": "https://example.com/image.jpg",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- **404** - Mercado ou produto não encontrado
- **403** - Acesso negado

DELETE **/markets/:id/products/:productId**

Descrição: Remover produto do mercado (Admin)

Autorização: JWT Bearer Token + Admin

Parâmetros:

- `id` (string) - ID do mercado
- `productId` (string) - ID do produto **Resposta (200):**

```
{
  "id": "uuid",
  "name": "Nome do Mercado",
  "latitude": -15.7942,
  "longitude": -47.8822,
  "address": "Endereço do mercado",
  "phone": "(11) 99999-9999",
  "description": "Descrição do mercado",
  "imageUrl": "https://example.com/image.jpg",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- `404` - Mercado não encontrado
- `403` - Acesso negado

Lista de Compras (`/shopping-list`)

POST `/shopping-list/items`

Descrição: Adicionar item à lista de compras

Autorização: JWT Bearer Token

Payload:

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

Resposta (201):

```
{
  "id": "uuid",
  "quantity": 2,
  "isSelected": false,
  "userId": "uuid",
  "productId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- **404** - Produto não encontrado

GET /shopping-list

Descrição: Obter lista de compras do usuário

Autorização: JWT Bearer Token

Resposta (200):

```
[
  {
    "id": "uuid",
    "quantity": 2,
    "isSelected": true,
    "userId": "uuid",
    "productId": "uuid",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z",
    "product": {
      "id": "uuid",
      "name": "Nome do Produto",
      "currentPrice": 15.99,
      "imageUrl": "https://example.com/image.jpg"
    }
  }
]
```

GET /shopping-list/total

Descrição: Obter total dos itens selecionados

Autorização: JWT Bearer Token

Resposta (200):

```
{
  "total": 47.97,
  "itemCount": 3
}
```

PATCH /shopping-list/items/:id

Descrição: Atualizar item da lista de compras

Autorização: JWT Bearer Token

Parâmetros:

- **id** (string) - ID do item **Payload:**

```
{
  "quantity": 3,
  "isSelected": true
}
```

Resposta (200):

```
{
  "id": "uuid",
  "quantity": 3,
  "isSelected": true,
  "userId": "uuid",
  "productId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Item não encontrado

DELETE /shopping-list/items/:id

Descrição: Remover item da lista de compras

Autorização: JWT Bearer Token

Parâmetros:

- id (string) - ID do item **Resposta (200):**

```
{
  "message": "Item removido com sucesso"
}
```

Erros:

- 404 - Item não encontrado

DELETE /shopping-list/clear

Descrição: Limpar toda a lista de compras

Autorização: JWT Bearer Token

Resposta (200):

```
{
  "message": "Lista de compras limpa com sucesso"
}
```


POST /shopping-list/finalize

Descrição: Finalizar compra dos itens selecionados

Autorização: JWT Bearer Token

Payload:

```
{
  "selectedItemIds": ["uuid1", "uuid2", "uuid3"]
}
```

Resposta (201):

```
{
  "id": "uuid",
  "totalCost": 47.97,
  "userId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "purchasedItems": [
    {
      "id": "uuid",
      "productName": "Nome do Produto",
      "priceAtTime": 15.99,
      "quantity": 3,
      "purchaseId": "uuid",
      "productId": "uuid"
    }
  ]
}
```

Erros:

- 400 - Nenhum item selecionado para finalizar compra

★ Avaliações (/reviews)

POST /reviews

Descrição: Criar uma nova avaliação de mercado

Autorização: JWT Bearer Token

Payload:

```
{
  "marketId": "uuid",
  "rating": 5,
}
```

```
{
  "comment": "Excelente mercado!"
}
```

Resposta (201):

```
{
  "id": "uuid",
  "rating": 5,
  "comment": "Excelente mercado!",
  "userId": "uuid",
  "marketId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 400 - Você já avaliou este mercado
- 404 - Mercado não encontrado

GET /reviews/market/:marketId

Descrição: Obter todas as avaliações de um mercado

Autorização: Nenhuma

Parâmetros:

- **marketId** (string) - ID do mercado

Resposta (200):

```
[
  {
    "id": "uuid",
    "rating": 5,
    "comment": "Excelente mercado!",
    "userId": "uuid",
    "marketId": "uuid",
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
]
```

Erros:

- 404 - Mercado não encontrado

GET /reviews/market/:marketId/rating

Descrição: Obter avaliação média de um mercado

Autorização: Nenhuma

Parâmetros:

- **marketId** (string) - ID do mercado **Resposta (200):**

```
{
  "averageRating": 4.5,
  "totalReviews": 10
}
```

Erros:

- **404** - Mercado não encontrado

GET **/reviews/user/me**

Descrição: Obter todas as avaliações do usuário logado

Autorização: JWT Bearer Token

Resposta (200):

```
[
  {
    "id": "uuid",
    "rating": 5,
    "comment": "Excelente mercado!",
    "userId": "uuid",
    "marketId": "uuid",
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET **/reviews/:id**

Descrição: Obter avaliação por ID

Autorização: Nenhuma

Parâmetros:

- **id** (string) - ID da avaliação **Resposta (200):**

```
{
  "id": "uuid",
  "rating": 5,
  "comment": "Excelente mercado!",
  "userId": "uuid",
  "marketId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Avaliação não encontrada

PATCH /reviews/:id

Descrição: Atualizar uma avaliação

Autorização: JWT Bearer Token

Parâmetros:

- id (string) - ID da avaliação **Payload:**

```
{
  "rating": 4,
  "comment": "Bom mercado!"
}
```

Resposta (200):

```
{
  "id": "uuid",
  "rating": 4,
  "comment": "Bom mercado!",
  "userId": "uuid",
  "marketId": "uuid",
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

Erros:

- 404 - Avaliação não encontrada
- 400 - Você só pode atualizar suas próprias avaliações

DELETE /reviews/:id

Descrição: Remover uma avaliação

Autorização: JWT Bearer Token

Parâmetros:

- id (string) - ID da avaliação **Resposta (200):**

```
{
  "message": "Avaliação removida com sucesso"
}
```

Erros:

- 404 - Avaliação não encontrada

- 400 - Você só pode remover suas próprias avaliações

Rota Principal (/)

GET /

Descrição: Página inicial da API

Autorização: Nenhuma

Resposta (200):

```
{
  "message": "Hello World!"
}
```

Códigos de Status HTTP

Sucesso

- 200 - OK
- 201 - Created

Erro do Cliente

- 400 - Bad Request
- 401 - Unauthorized
- 403 - Forbidden
- 404 - Not Found

Erro do Servidor

- 500 - Internal Server Error

Exemplos de Autenticação

Registrar e Fazer Login

```
# 1. Registrar usuário
curl -X POST http://localhost:3001/auth/register \
  -H "Content-Type: application/json" \
  -d '{
    "email": "test@example.com",
    "password": "password123",
    "name": "Test User",
    "latitude": -15.7942,
    "longitude": -47.8822
  }'
```

```
# 2. Fazer login
curl -X POST http://localhost:3001/auth/login \
  -H "Content-Type: application/json" \
  -d '{
    "email": "test@example.com",
    "password": "password123"
  }'

# 3. Usar o token retornado nas próximas requisições
curl -X GET http://localhost:3001/auth/profile \
  -H "Authorization: Bearer SEU_TOKEN_AQUI"
```

Criar e Gerenciar Produtos

```
# 1. Criar produto
curl -X POST http://localhost:3001/products \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer SEU_TOKEN_AQUI" \
  -d '{
    "name": "Arroz Integral",
    "description": "Arroz integral orgânico",
    "currentPrice": 15.99,
    "imageUrl": "https://example.com/arroz.jpg"
  }'

# 2. Listar produtos
curl -X GET "http://localhost:3001/products?name=arroz&sortBy=price&sortOrder=asc"

# 3. Aprovar produto (Admin)
curl -X PATCH http://localhost:3001/products/PRODUCT_ID/approve \
  -H "Authorization: Bearer ADMIN_TOKEN"
```

Gerenciar Lista de Compras

```
# 1. Adicionar item à lista
curl -X POST http://localhost:3001/shopping-list/items \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer SEU_TOKEN_AQUI" \
  -d '{
    "productId": "PRODUCT_ID",
    "quantity": 2
  }'

# 2. Ver lista de compras
curl -X GET http://localhost:3001/shopping-list \
  -H "Authorization: Bearer SEU_TOKEN_AQUI"

# 3. Finalizar compra
```

```
curl -X POST http://localhost:3001/shopping-list/finalize \
-H "Content-Type: application/json" \
-H "Authorization: Bearer SEU_TOKEN_AQUI" \
-d '{
  "selectedItemIds": ["ITEM_ID_1", "ITEM_ID_2"]
}'
```

Validações de Dados

Campos Obrigatórios

- **Email:** Deve ser um email válido
- **Senha:** Mínimo 6 caracteres
- **Nome:** Deve ser uma string não vazia
- **Preço:** Deve ser um número positivo
- **Rating:** Deve ser um número entre 1 e 5
- **Latitude/Longitude:** Devem ser números válidos

Regras de Negócio

- Apenas administradores podem criar/editar/remover mercados
- Apenas administradores podem aprovar/reprovar produtos
- Usuários só podem editar seus próprios dados e avaliações
- Não é possível avaliar o mesmo mercado mais de uma vez
- Produtos devem ser aprovados para aparecer nas listagens públicas

Recursos Especiais

Busca por Proximidade

A API suporta busca de mercados por proximidade geográfica usando as coordenadas do usuário.

Ordenação por Distância

Produtos podem ser ordenados por distância do usuário ao mercado mais próximo que possui o produto.

Sistema de Pontos

Usuários ganham pontos ao contribuir com o sistema (criar produtos, fazer avaliações, etc.).

Histórico de Preços

O sistema mantém um histórico automático de mudanças de preços dos produtos.

Validação Automática

Todas as requisições são validadas automaticamente usando class-validator.

Documentação Adicional

- **Swagger UI:** <http://localhost:3001/api>
- **Prisma Studio:** Execute `npx prisma studio` para visualizar o banco de dados
- **Testes:** Execute `npm test` para rodar os testes unitários e `npm run test:e2e` para testes E2E