Testes Funcionais

**Teste Funcional**

**Buscar produtos pelo nome**

**Cenário 1: Buscar um produto existente**

**Dado** que o usuário está na página inicial

**Quando** ele digita "headphones" na barra de busca

**E** clicar no botão de busca

**Então** os resultados devem mostrar produtos relacionados a "headphones"

**Cenário 2: Buscar um produto inexistente**

**Dado** que o usuário está na página inicial

**Quando** ele digita "produtoInexistente123" na barra de busca

**E** clicar no botão de busca

**Então** uma mensagem de "No results" ou equivalente deve ser exibida

**Incluir produto no carrinho**

**Cenário 1: Validar nome e valor do produto na página de detalhes**

**Dado** que o usuário acessou a página de detalhes de um produto

**Então** o nome do produto deve ser exibido corretamente

**E** o preço do produto deve estar visível no formato correto (ex: $100.00)

**Cenário 2: Verificar se a descrição do produto está visível**

**Dado** que o usuário acessou a página de um produto

**Então** a descrição completa do produto deve ser exibida abaixo do título e do preço

**Cenário 3: Verificar se o produto possui imagens disponíveis**

**Dado** que o usuário está na página de um produto

**Então** pelo menos uma imagem principal do produto deve estar visível

**E** se houver miniaturas, elas também devem ser exibidas e navegáveis

**Cenário 4: Verificar se o vídeo demonstrativo está presente**

**Dado** que o usuário acessou a página de detalhes de um produto

**Então** um vídeo demonstrativo deve estar visível

**E** o vídeo deve ser reproduzível ao clicar no botão de play

**Cenário 5: Adicionar 1 unidade de um produto ao carrinho com sucesso**

**Dado** que o usuário está na página de detalhes de um produto

**E** a cor está selecionada automaticamente

**E** a quantidade padrão é "1"

**Quando** ele clica no botão "ADD TO CART"

**Então** o carrinho deve indicar 1 item adicionado

**E** o produto deve aparecer na visualização do carrinho com nome, cor e quantidade "1"

**Cenário 6: Adicionar produto com quantidade e cor selecionadas**

**Dado** que o usuário está na página de detalhes de um produto

**E** a cor está selecionada automaticamente

**Quando** ele define a quantidade como "2"

**E** clicar no botão "ADD TO CART"

**Então** o carrinho deve indicar 2 itens adicionados

**E** o produto deve estar listado com a cor e quantidade corretas

**Cenário 7: Tentar adicionar quantidade negativa de produto**

**Dado** que o usuário está na página de detalhes de um produto

**Quando** ele altera a quantidade para "-1"

**E** clicar no botão "ADD TO CART"

**Então** uma mensagem de erro deve ser exibida indicando que a quantidade é inválida

**E** o produto não deve ser adicionado ao carrinho

**Cenário 8: Tentar adicionar quantidade com caracteres especiais no produto**

**Dado** que o usuário está na página de detalhes de um produto

**Quando** ele altera a quantidade para "$^!@%$#"

**E** clicar no botão "ADD TO CART"

**Então** uma mensagem de erro deve ser exibida indicando que a quantidade é inválida

**E** o produto não deve ser adicionado ao carrinho

**Cenário 9: Verificar dados dos produtos na tela de pagamento**

**Dado** que o usuário adiciona um produto ao carrinho

**E** acessou a tela de checkout

**Então** o nome, a cor, a quantidade e o preço do produto devem ser exibidos corretamente

**E** o valor total da compra deve refletir a soma de todos os produtos adicionados

**Validar produtos incluídos na tela de pagamento**

**Cenário 1: Verificar dados dos produtos na tela de pagamento**

**Dado** que o usuário adiciona um produto ao carrinho

**E** acessou a tela de checkout

**E** o nome, a cor, a quantidade e o preço do produto devem ser exibidos corretamente

**E** o valor total da compra deve refletir a soma de todos os produtos adicionados

**Quando** o usuário clica em checkout

**Então** o usuário é levado para tela de pagamento

**Cenário 2: Tentar realizar o pagamento sem conta cadastrada**

**Dado** que o usuárioesteja na tela de realizar o pagamento

**E** o usuário tenta realizar o pagamento

**Então** o site deve solicitar que ele cadastre uma conta ou entre com uma conta existente

**Cenário 3: Tentar realizar o pagamento com conta cadastrada inválida**

**Dado** que o usuárioesteja na tela de realizar o pagamento

**E** o usuário tenta realizar o pagamento

**E** o site deve solicitar que ele cadastre uma conta ou entre com uma conta existente

**Quando** o usuário entra com dados inválidos

**E** o usuário clica no botão de **“LOGIN”**

**Então** o site deve retornar uma mensagem **“Incorrect user name or password.”**

**Cenário 4: Tentar realizar o pagamento com conta cadastrada**

**Dado** que o usuárioesteja na tela de realizar o pagamento

**E** o usuário tenta realizar o pagamento

**E** o site deve solicitar que ele cadastre uma conta ou entre com uma conta existente

**Quando** o usuário entra com dados válidos

**E** o usuário clica no botão de **“LOGIN”**

**Então** o site deve mostrar os dados do usuário

**E** o site deve mostrar a opção para prosseguir com o pagamento

**Cenário 5: Finalizar o pagamento utilizando SafePay**

**Dado** que o usuárioesteja na tela de realizar o pagamento

**E** o usuário seleciona o método de pagamento **“SafePay”**

**E** o site deve solicitar que ele entre com login e senha do **SafePay**

**Quando** o usuário entra com dados válidos

**E** o usuário clica no botão de **“PAY NOW”**

**Então** o site deve mostrar os dados do usuário

**E** o site deve mostrar que a ordem de compra foi solicitada com sucesso

**Cenário 6: Finalizar o pagamento utilizando MasterCredit**

**Dado** que o usuárioesteja na tela de realizar o pagamento

**E** o usuário seleciona o método de pagamento **“MasterCredit”**

**E** o site deve solicitar que ele entre com os dados do cartão de crédito

**Quando** o usuário entra com o **Card Number** com exatamente 16 dígitos

**E** o usuário entra com o **CCV** number com exatamente 3 dígitos

**E** o usuário entra com a **expiration date**

**E** o usuário entra com o **Cardholder name**

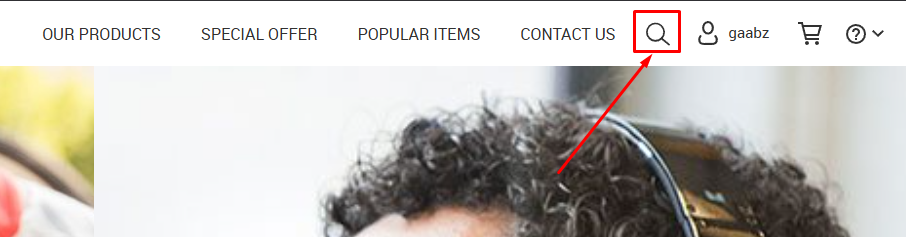
**Então** o site deve permitir o usuário prosseguir com o pagamento habilitando o botão **PAY NOW**

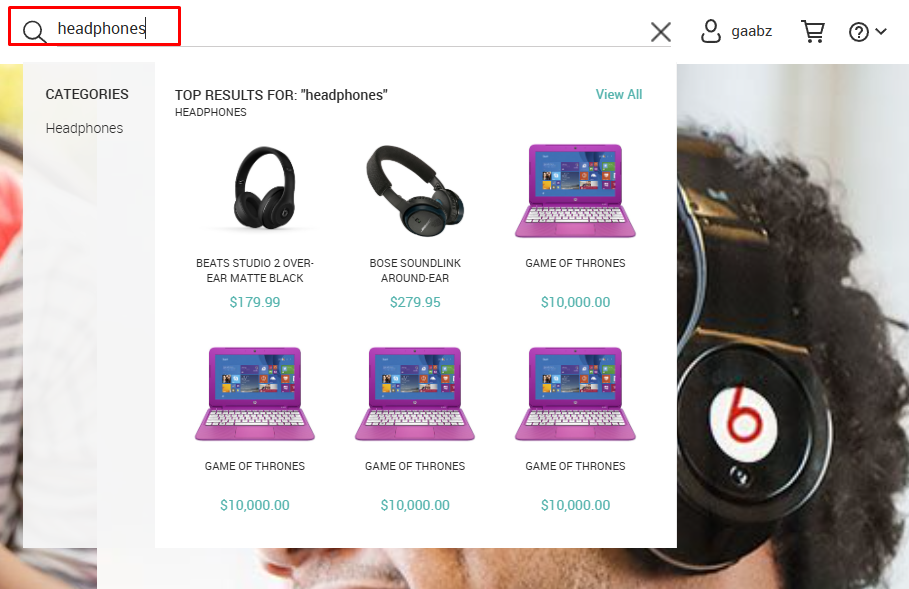
**E** o site deve mostrar que a ordem de compra foi solicitada com sucesso

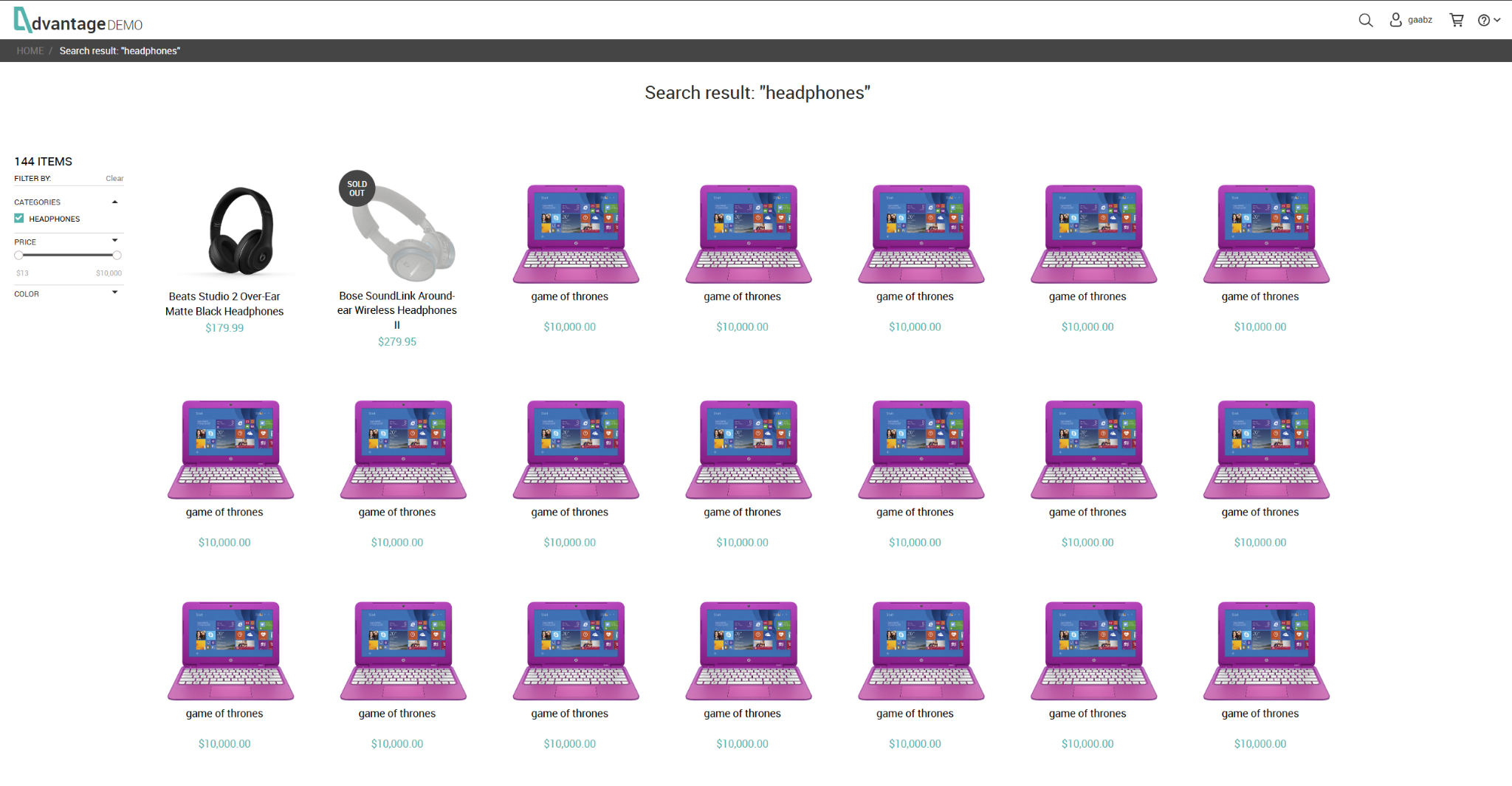
Resultados

**Buscar produtos pelo nome**

**Cenário 1: Buscar um produto existente**

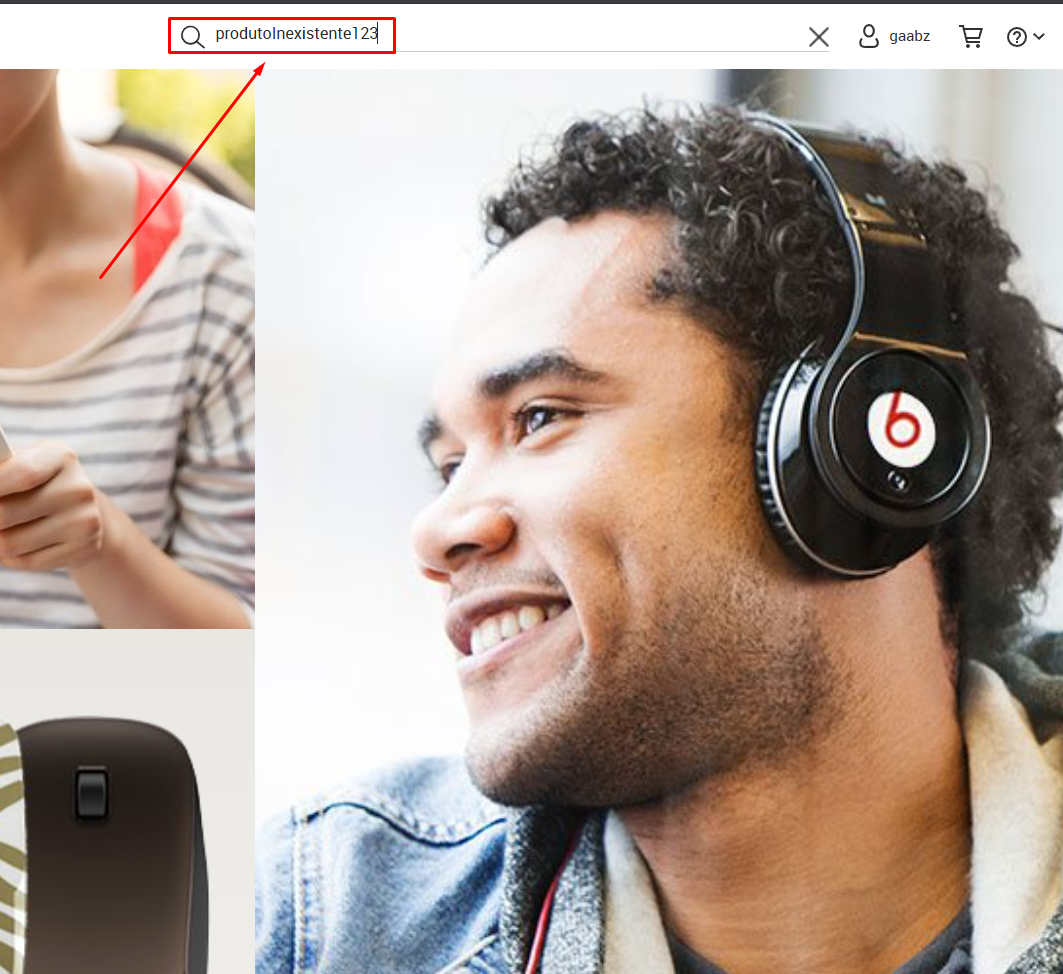


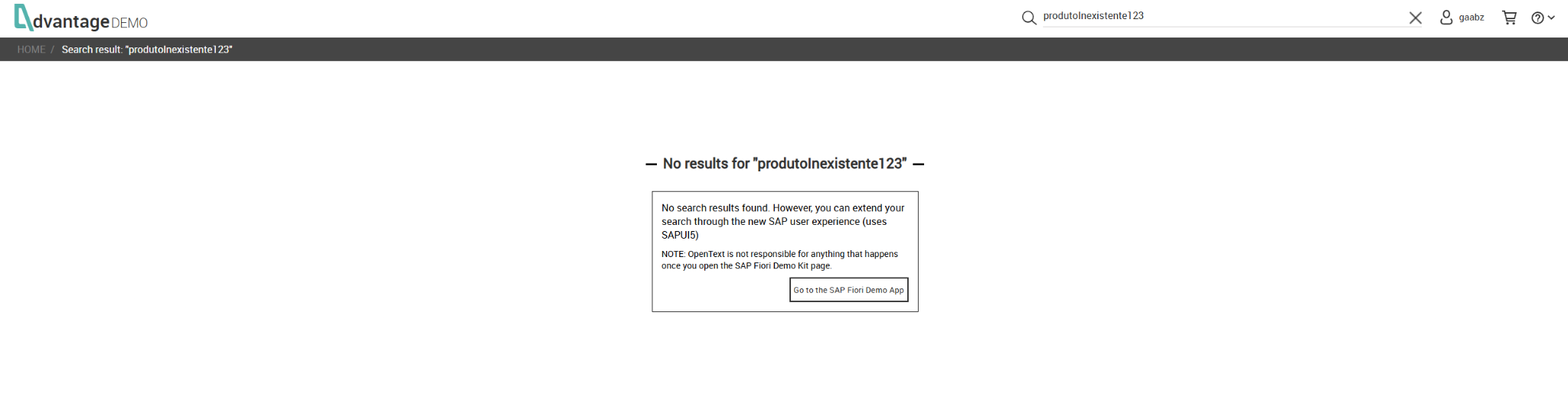




Reporte de possivel bug:

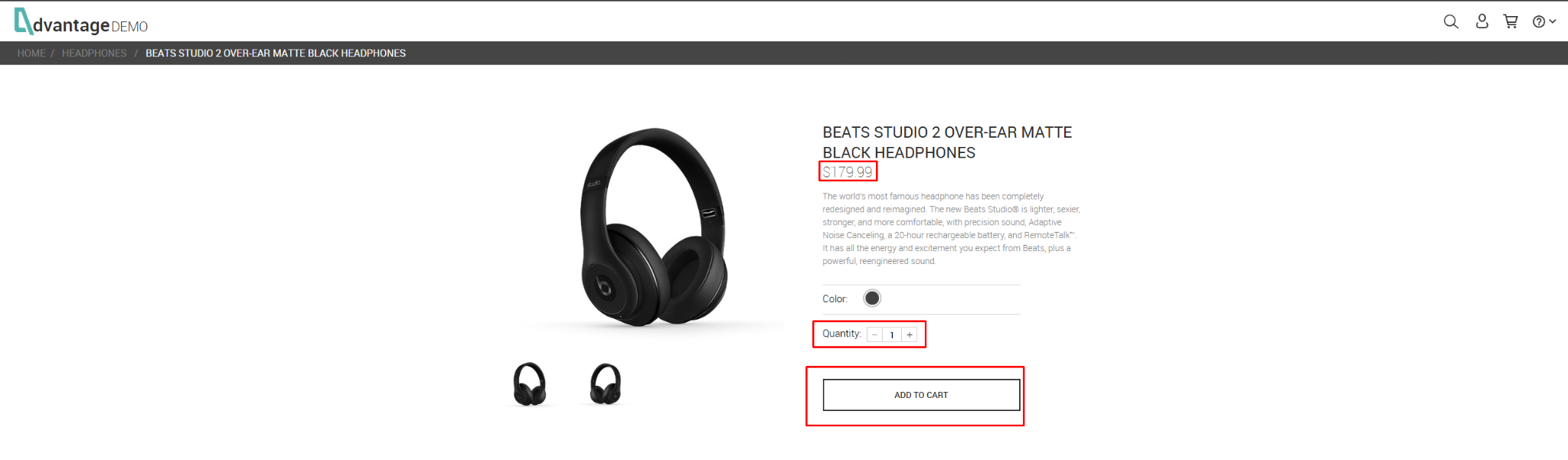
Ao realizar a pesquisa de **“headphones”** ao invés de mostrar apenas os headphones a pesquisa traz produtos com o nome de **“Game of Thrones”** sendo que deveria trazer apenas produtos relacionados com **“headphones”**.

**Cenário 2: Buscar um produto inexistente**

****

**Incluir produto no carrinho**

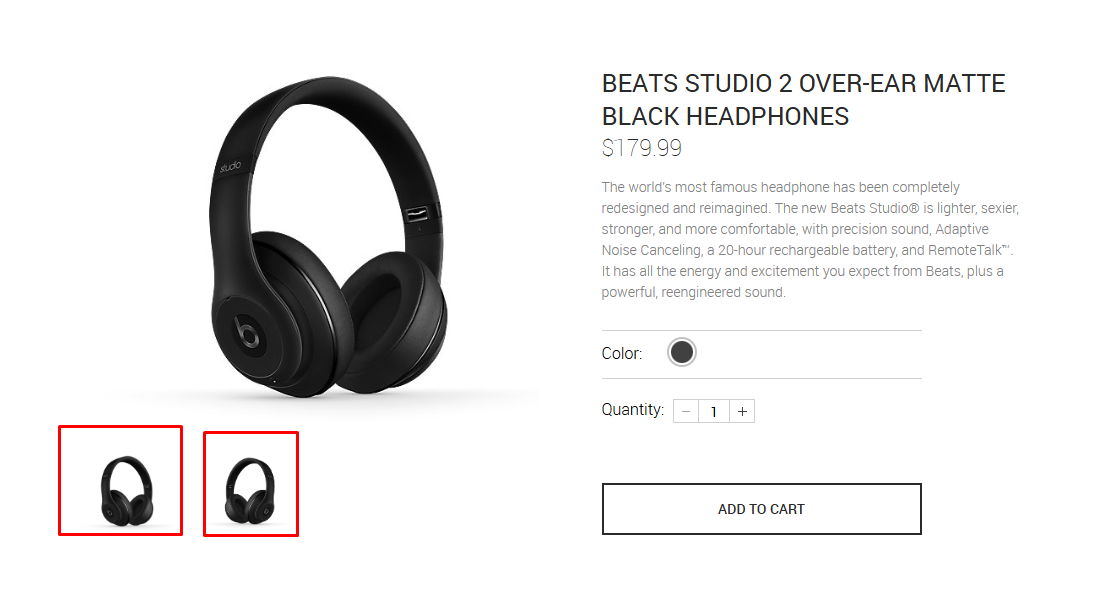
**Cenário 1: Validar nome e valor do produto na página de detalhes**

****

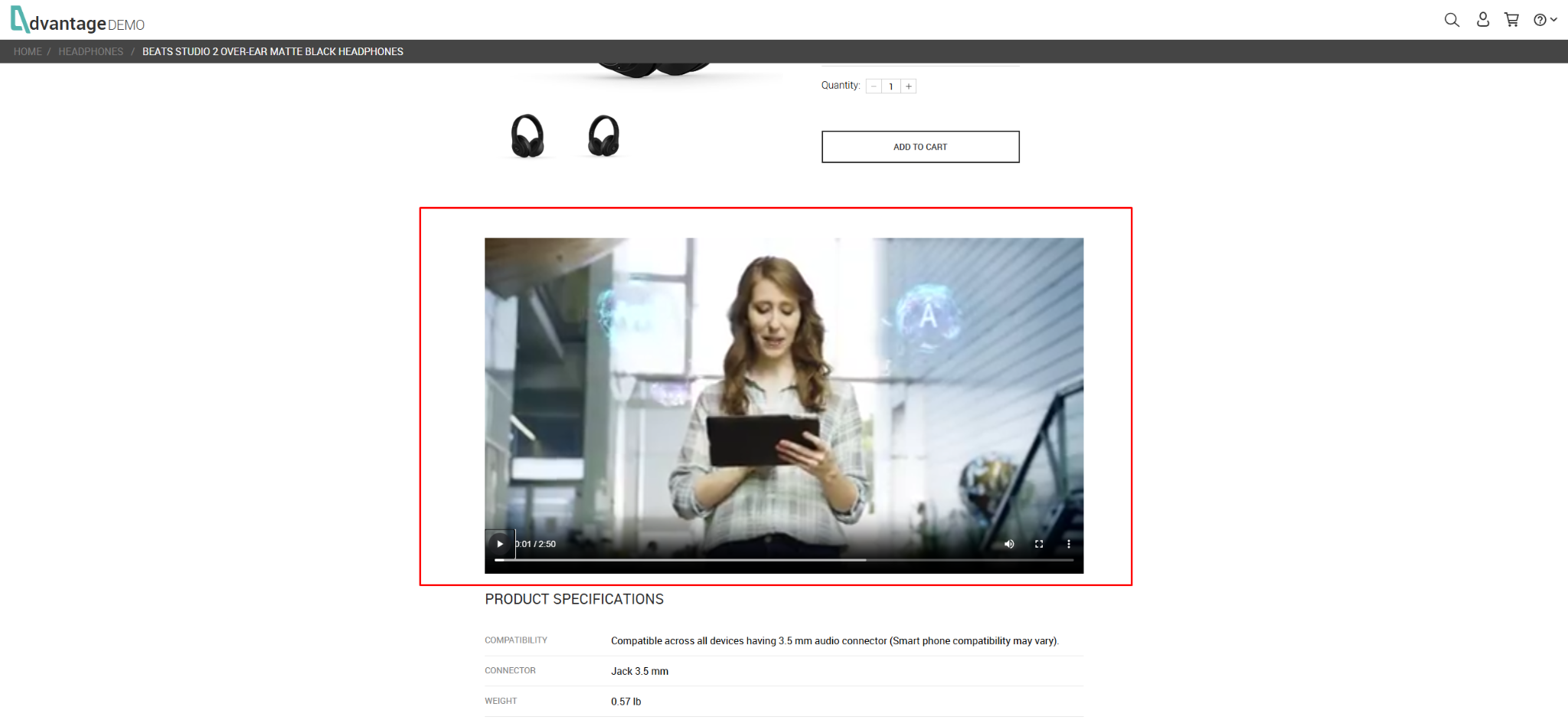
**Cenário 2: Verificar se a descrição do produto está visível**

****

**Cenário 3: Verificar se o produto possui imagens disponíveis**

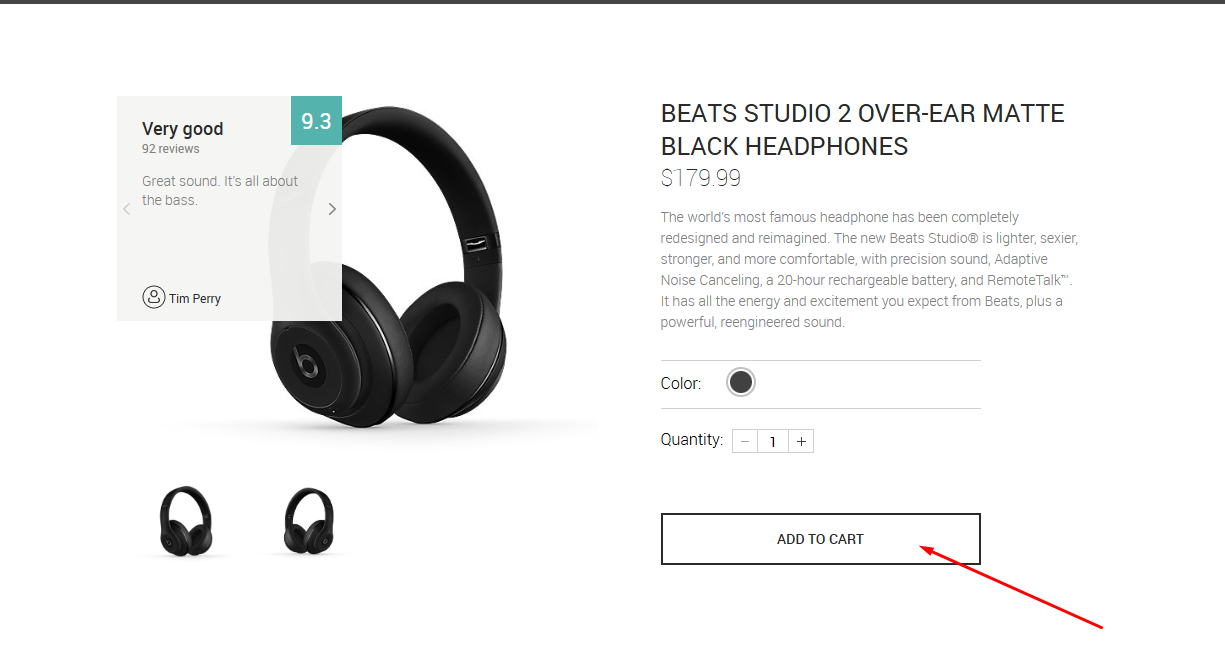
****

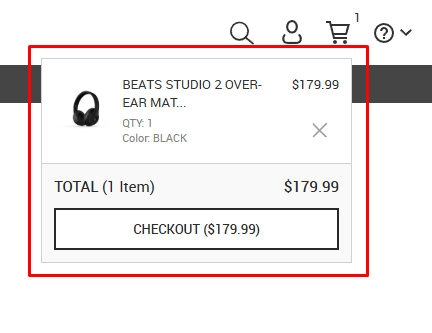
**Cenário 4: Verificar se o vídeo demonstrativo está presente**

****

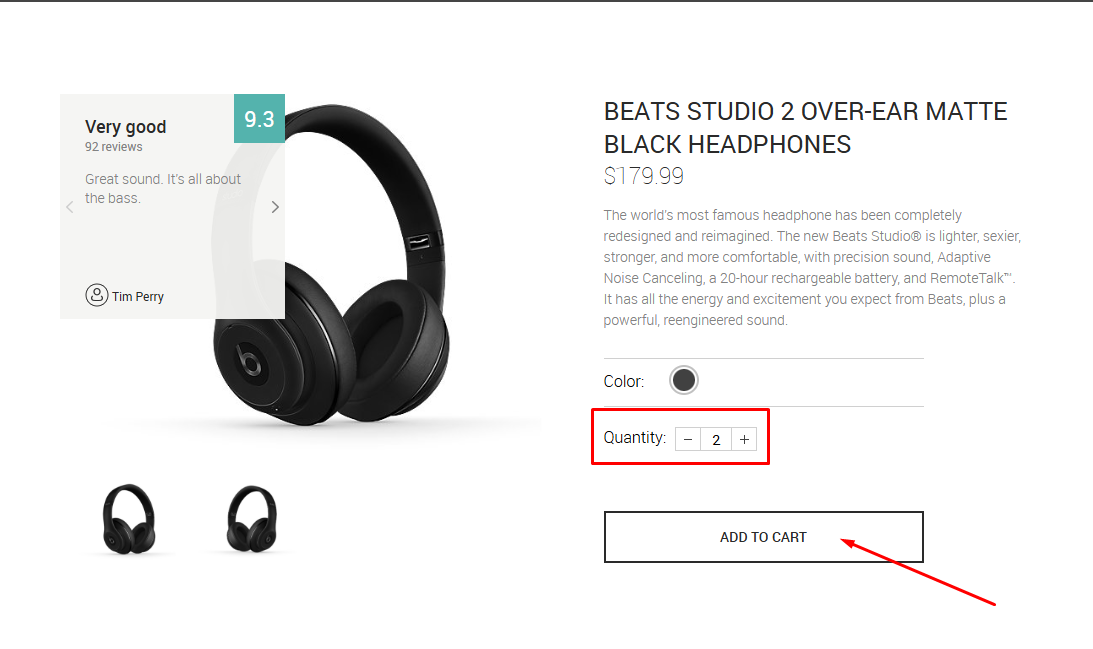
**Cenário 5: Adicionar 1 unidade de um produto ao carrinho com sucesso**

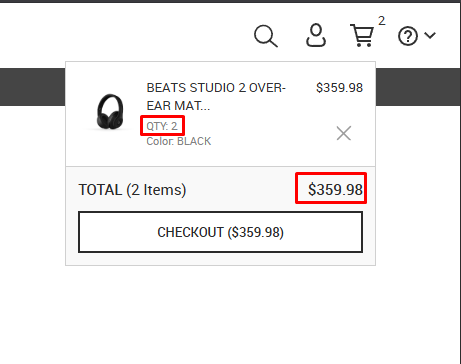
****

****

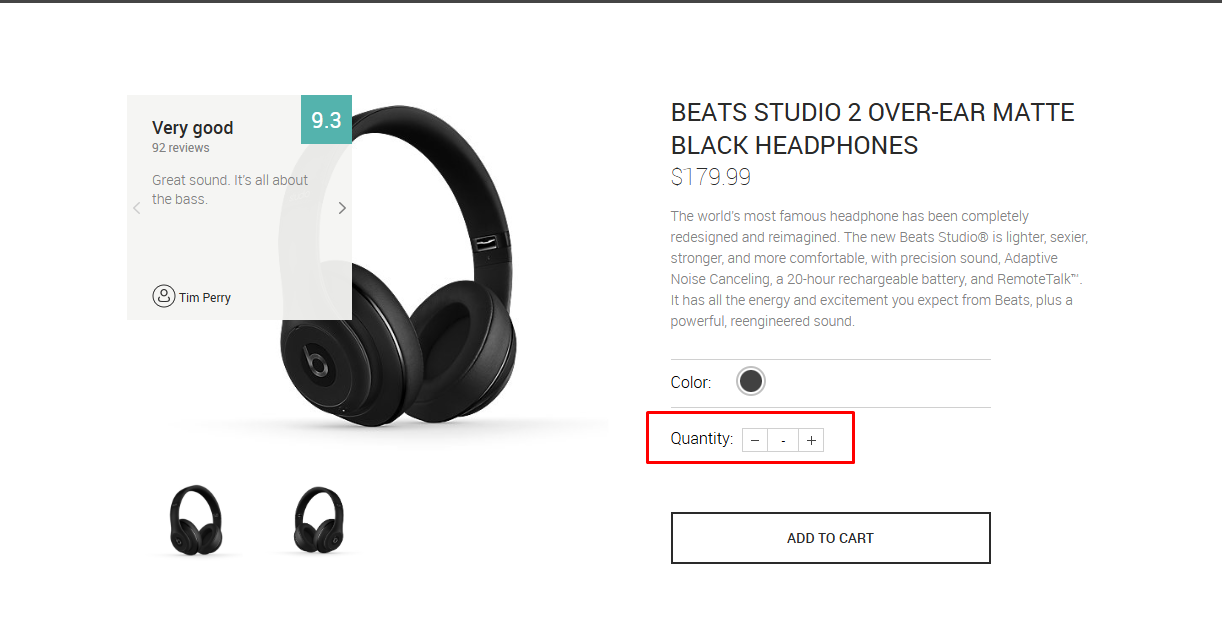
****

**Cenário 6: Adicionar produto com quantidade e cor selecionadas**

****

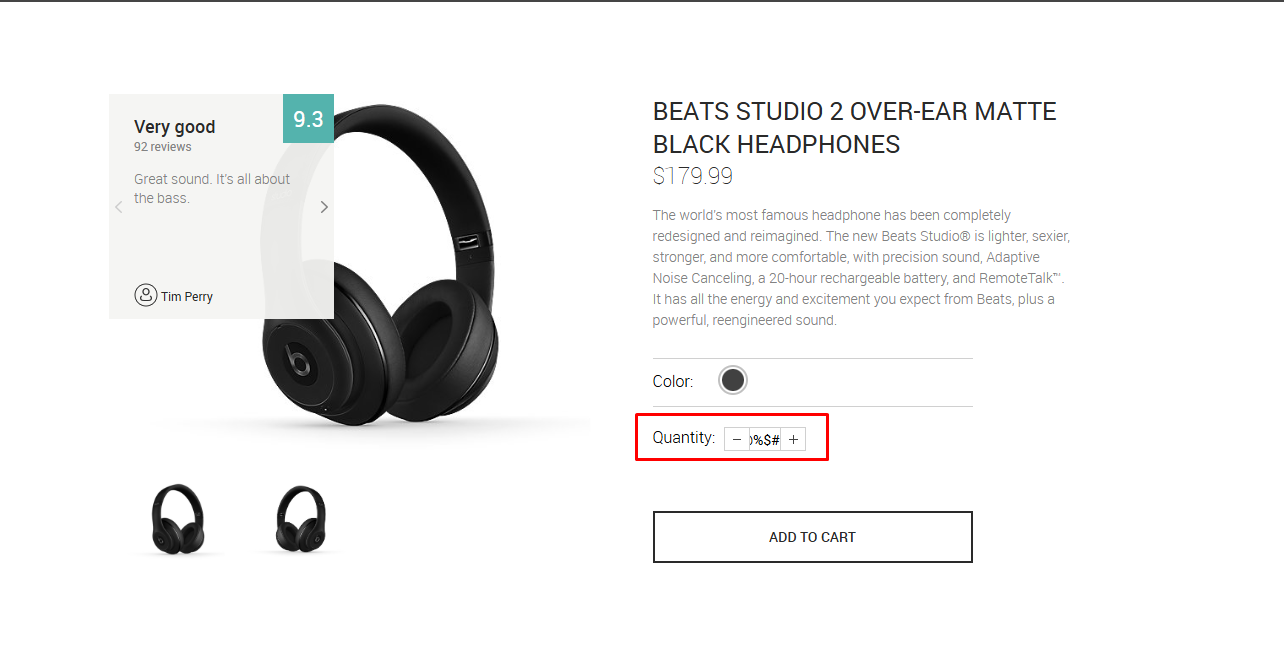
****

**Cenário 7: Tentar adicionar quantidade negativa de produto**

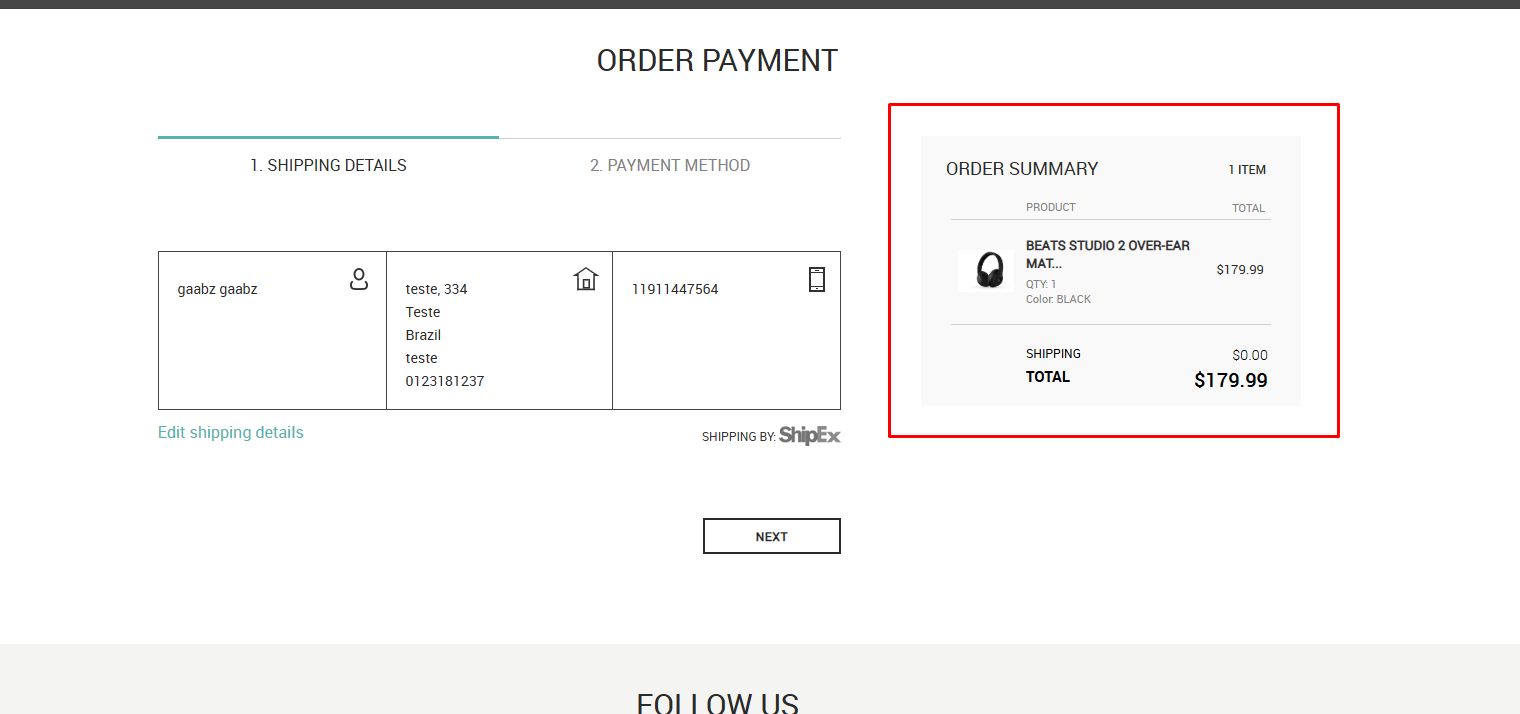
****

**Obs:** O site não permite adicionar números negativos, ao tentar adicionar ele transforma o número automaticamente em positivo

**Cenário 8: Tentar adicionar quantidade com caracteres especiais no produto**

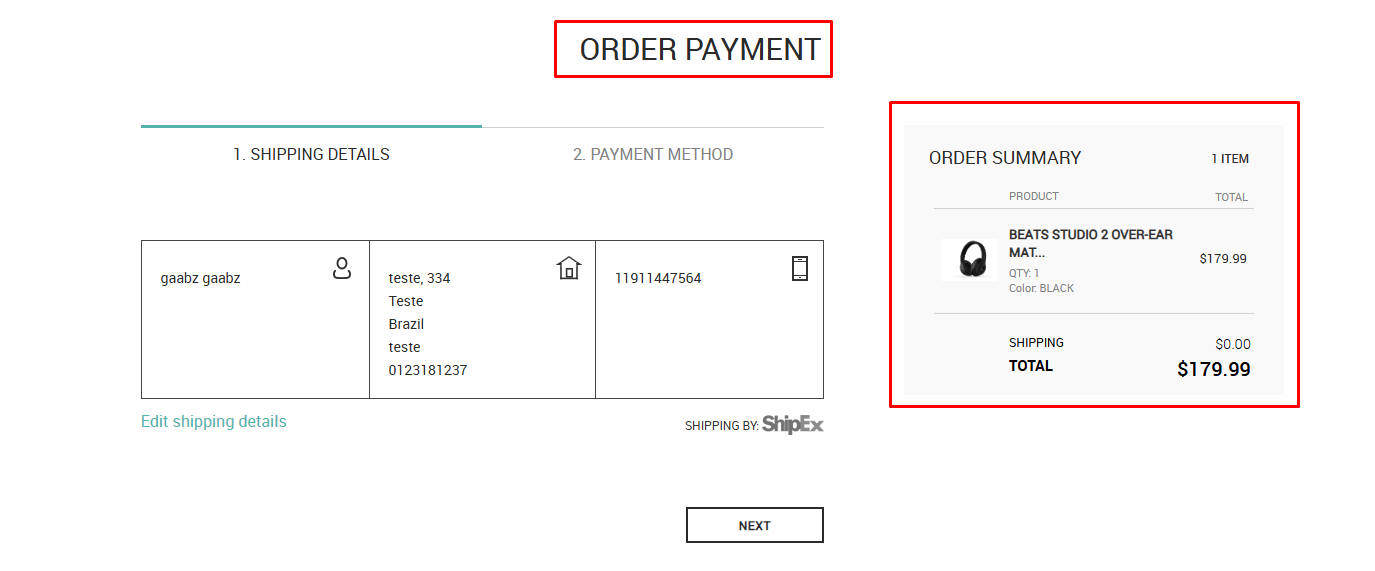


**Cenário 9: Verificar dados dos produtos na tela de pagamento**

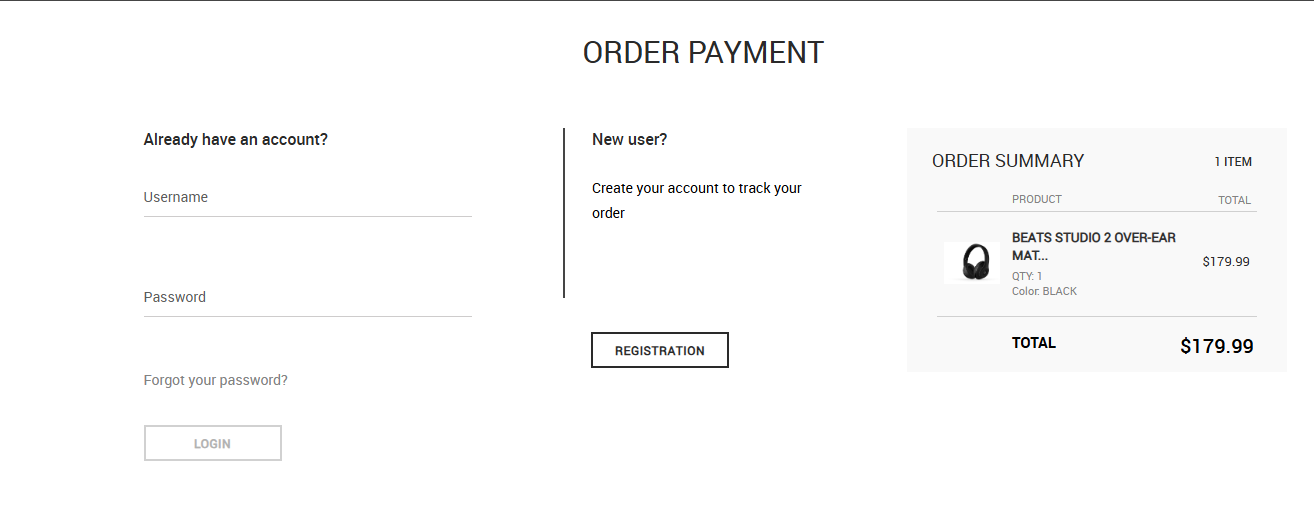
****

**Validar produtos incluídos na tela de pagamento**

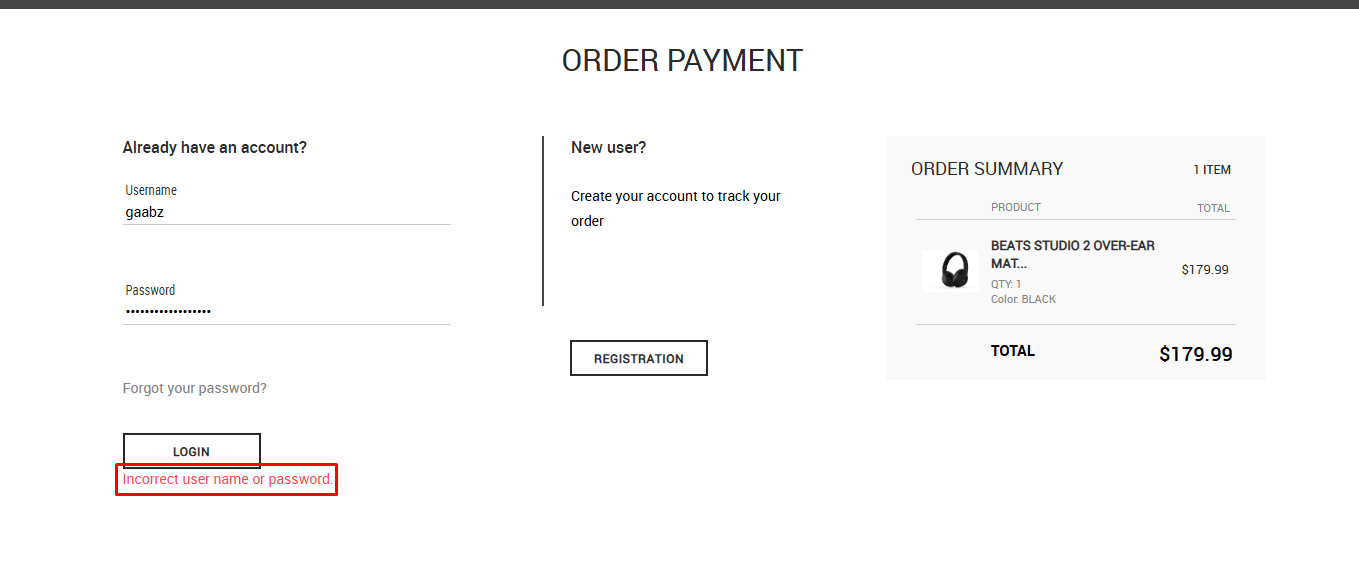
**Cenário 1: Verificar dados dos produtos na tela de pagamento**

****

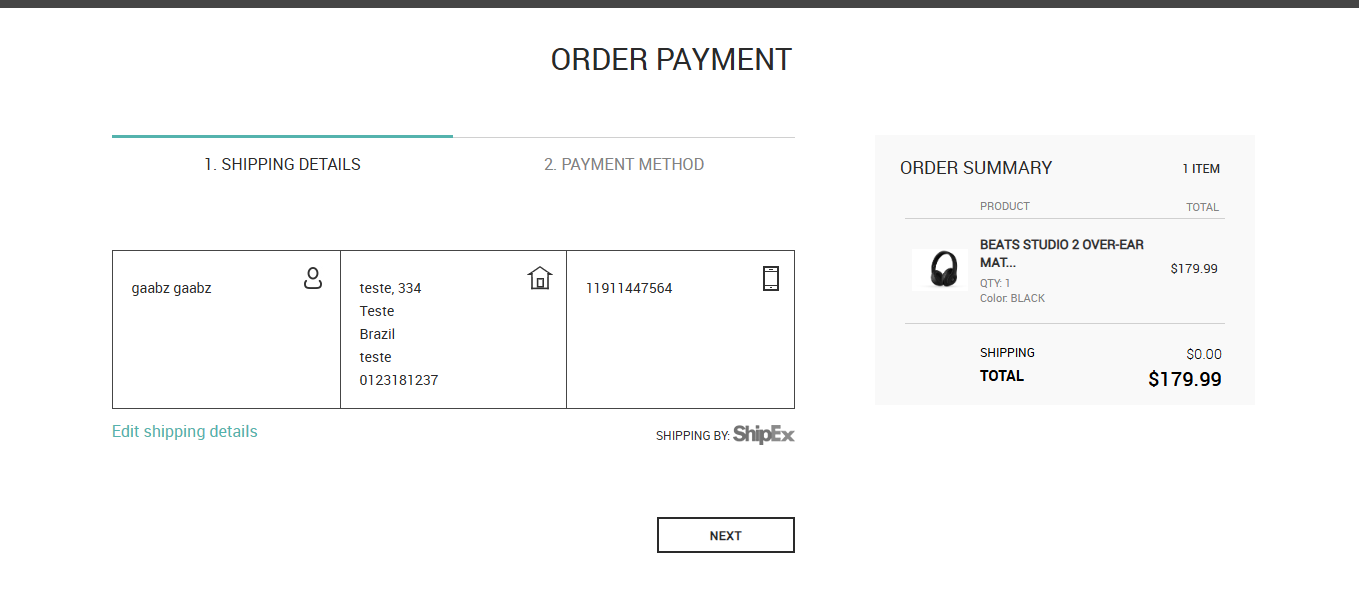
**Cenário 2: Tentar realizar o pagamento sem conta cadastrada**

****

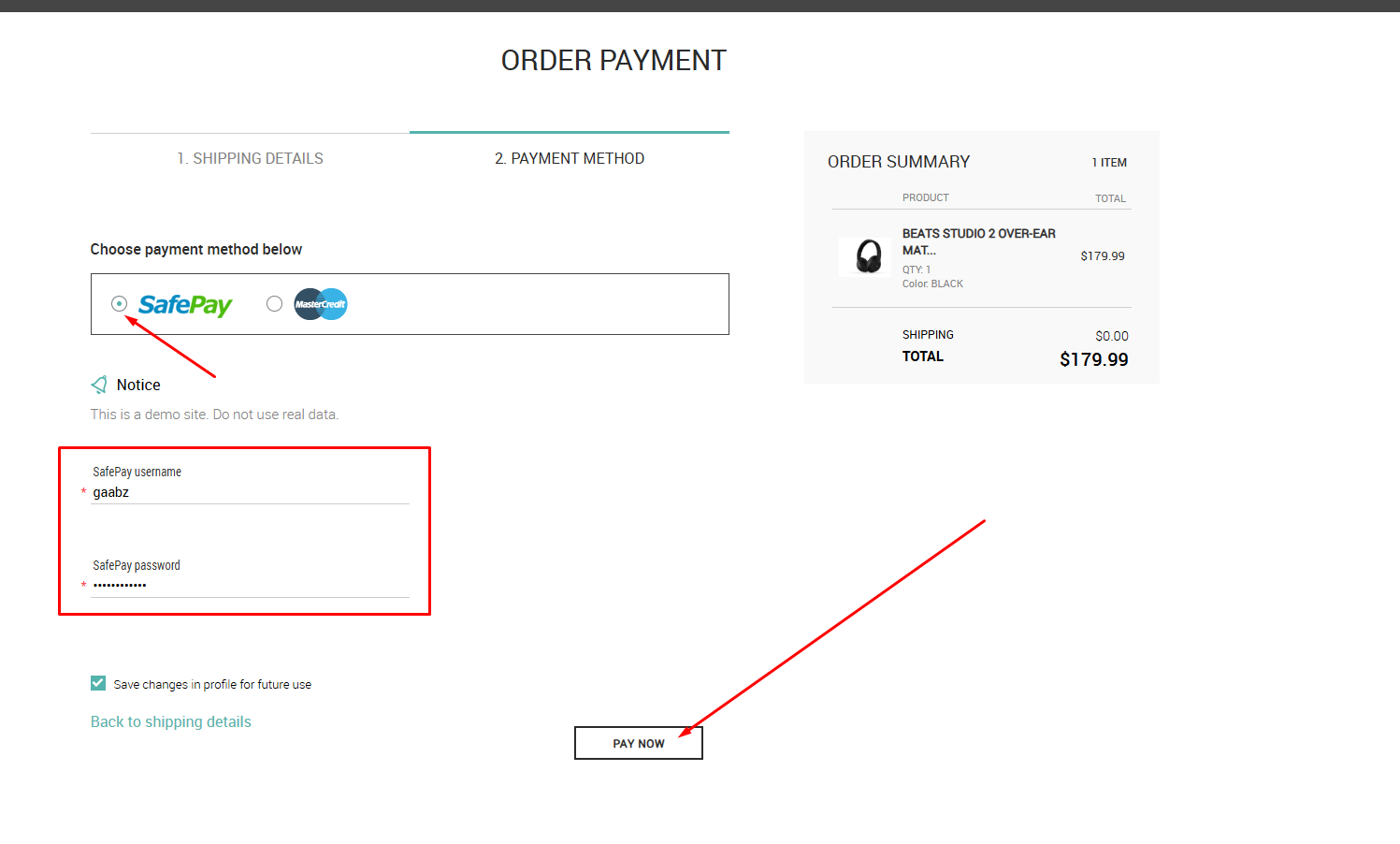
**Cenário 3: Tentar realizar o pagamento com conta cadastrada inválida**

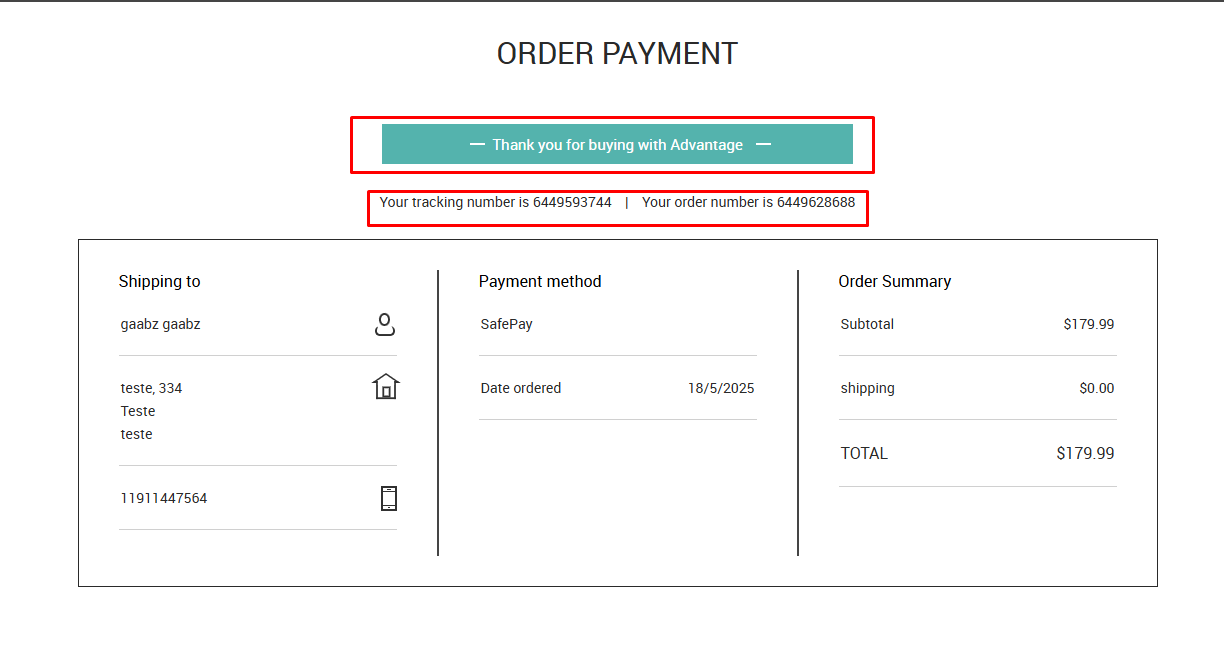
****

**Cenário 4: Tentar realizar o pagamento com conta cadastrada**

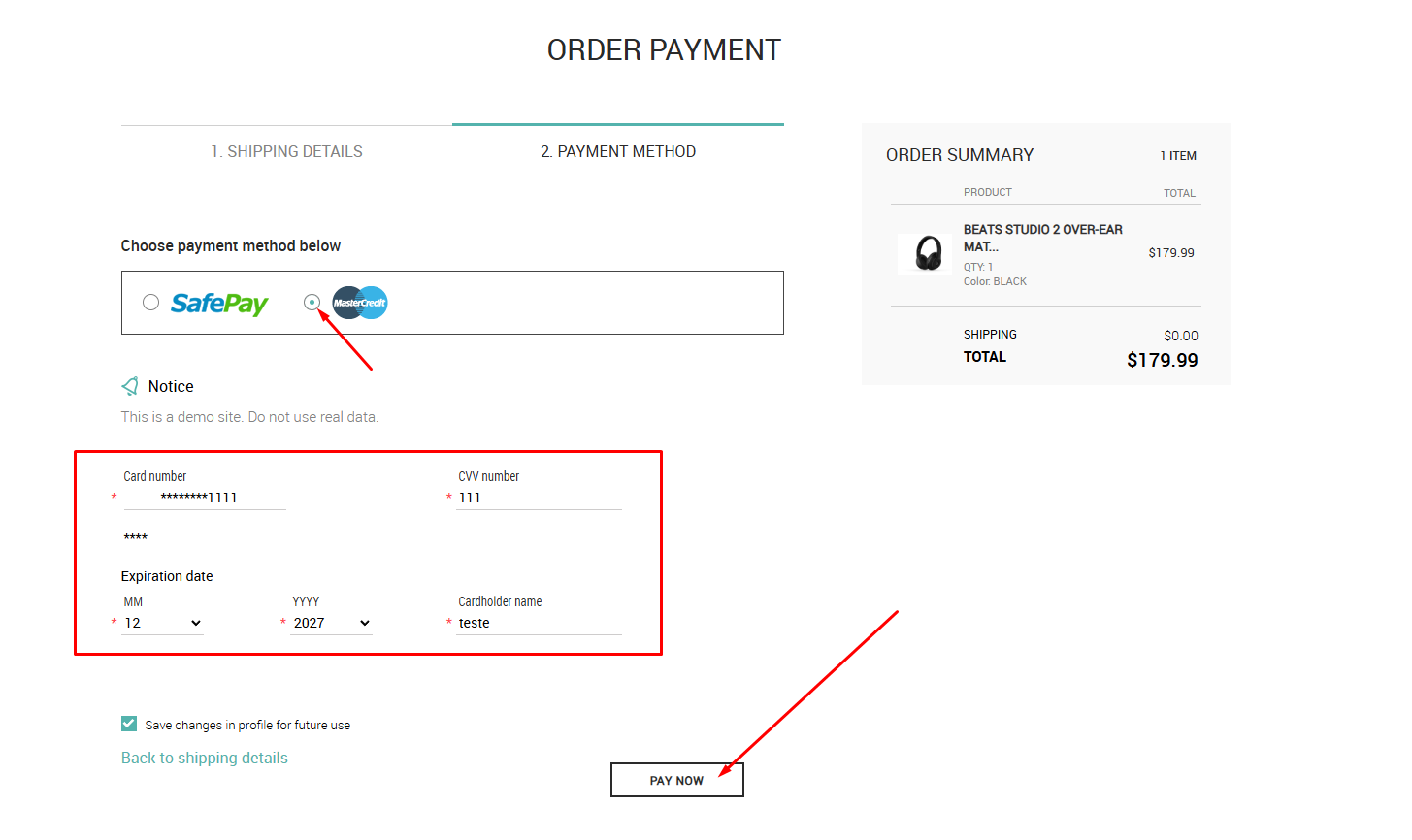
****

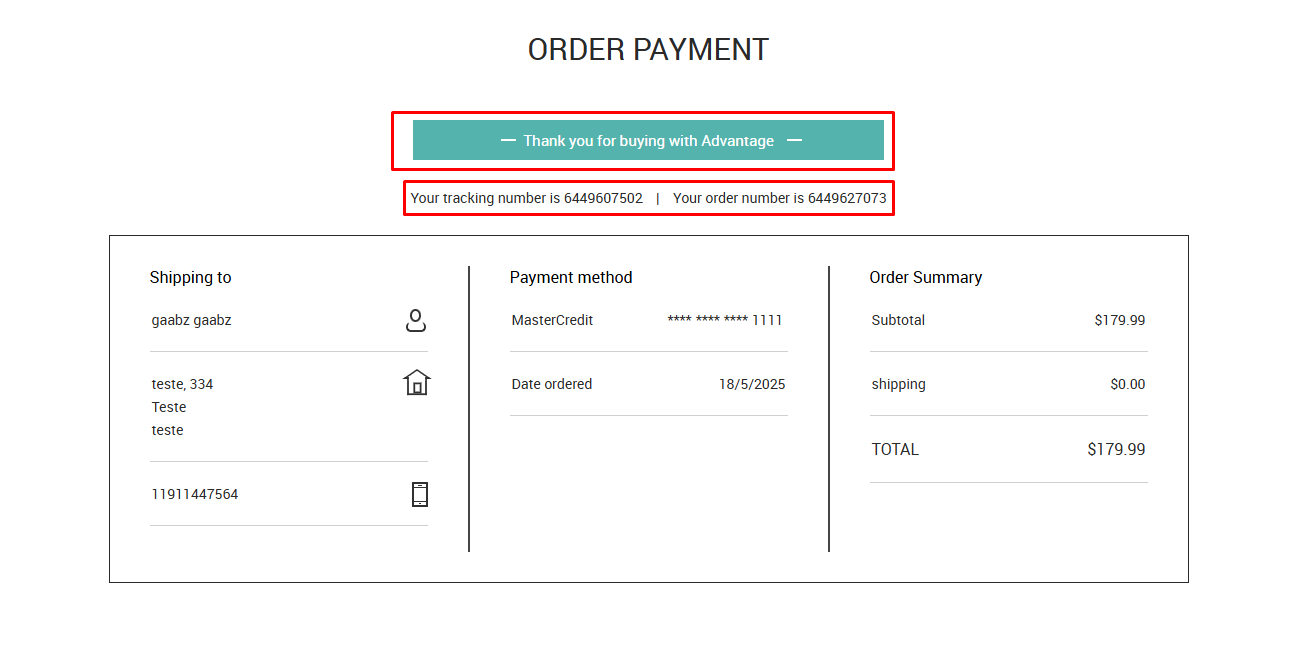
**Cenário 5: Finalizar o pagamento utilizando SafePay**

****

****

**Cenário 6: Finalizar o pagamento utilizando MasterCredit**

****

****

Documentação API

# **📘 Documentação da API - Advantage Online Shopping**

## **🔍 Introdução**

A API da Advantage Online Shopping é uma interface RESTful que permite a interação com as funcionalidades principais da loja virtual, como catálogo de produtos, gestão de usuários, pedidos e pagamentos. As respostas são em formato JSON e o acesso à maioria dos recursos requer autenticação via token JWT.

## **🔐 Autenticação**

**Tipo:** JWT (Bearer Token)

### **Obter token:**

**POST** /accountservice/login

#### **Exemplo de requisição:**

{

"email": "usuario@email.com",

"password": "123456"

}

#### **Exemplo de cabeçalho de autenticação:**

Authorization: Bearer {token}

## 

## 

## 

## **📂 Catálogo de Produtos (/catalog)**

### **GET /categories**

Retorna todas as categorias de produtos.

### **GET /products**

Retorna a lista completa de produtos.

### **GET /products/{productId}**

Retorna detalhes de um produto específico.

#### **Exemplo de resposta:**

{

"id": 1,

"name": "Headphones",

"description": "Fone de ouvido sem fio",

"price": 99.99,

"imageUrl": "/images/headphones.jpg",

"videoUrl": "/videos/headphones.mp4"

}

### **POST /catalog/api/v1/products**

Cria um novo produto no catálogo.

**Requer autenticação via Bearer Token e permissão de administrador.**

#### **Cabeçalho obrigatório:**

Authorization: Bearer {token}

#### 

#### 

#### **Exemplo de corpo da requisição:**

{

"productId": 9999,

"productName": "HyperX Cloud Alpha Gaming Headset",

"description": "Headset gamer com som surround, microfone removível e estrutura de alumínio",

"categoryId": 2,

"price": 499.90,

"imageUrl": "hyperx-cloud-alpha.jpg",

"images": ["img1.jpg", "img2.jpg"],

"productStatus": "Available",

"attributes": [

{ "attributeName": "Conectividade", "attributeValue": "P3 + USB" }

],

"colors": [

{ "code": "#000000", "inStock": 25, "name": "Black" }

]

}

#### **Exemplo de resposta de sucesso:**

{

"success": true,

"reason": "Produto criado com sucesso",

"id": 9999

}

#### **Exemplo de erro de permissão insuficiente:**

{

"success": false,

"reason": "Wrong account type (USER)"

}

## 

## 

## 

## **💳 Pagamento com Cartão (/mastercredit)**

### **POST /payment**

Realiza o pagamento via cartão de crédito.

#### **Corpo da requisição:**

{

"orderId": 123,

"cardNumber": "4111111111111111",

"cvv": "123",

"expirationDate": "12/2025",

"cardHolder": "Nome do Titular"

}

#### **Códigos de resposta:**

* 200 OK: Pagamento aprovado
* 400 Bad Request: Dados incorretos

## 

## 

## 

## 

## **👥 Usuários (/accountservice)**

### **POST /users**

Cria uma nova conta de usuário.

### **GET /users/{userId}**

Retorna os dados do usuário.

### **PUT /users/{userId}**

Atualiza os dados do usuário.

### **DELETE /users/{userId}**

Remove a conta do usuário.

## 

## 

## **🛒 Pedidos (/order)**

### **POST /orders**

Cria um novo pedido a partir dos produtos no carrinho.

### **GET /orders/{orderId}**

Retorna os detalhes de um pedido.

### **DELETE /orders/{orderId}**

Cancela um pedido.

## 

## **⚠️ Códigos de Erro Comuns**

* 200 OK: Requisição bem-sucedida
* 201 Created: Recurso criado com sucesso
* 400 Bad Request: Requisição inválida
* 401 Unauthorized: Token ausente ou inválido
* 403 Forbidden: Permissão insuficiente para acessar o recurso
* 404 Not Found: Recurso não encontrado
* 500 Internal Server Error: Erro no servidor

## **📆 Fluxo Recomendo de Uso**

1. Criar conta ou fazer login (obter token)
2. Listar produtos
3. Adicionar ao carrinho
4. Criar pedido
5. Realizar pagamento
6. Consultar pedido

## 

## 

## **📄 Exemplos de Testes**

### **Exemplo com curl:**

curl -X GET https://www.advantageonlineshopping.com/catalog/products \

-H "Authorization: Bearer {token}"

### **Exemplo com Postman:**

Importar a documentação via Swagger/OpenAPI ou configurar manualmente cada rota com o token.

Cenários em Gherkin

### **Cenário 1: Buscar todos os produtos com sucesso**

**Cenário: Listar todos os produtos com sucesso**

Dado que o usuário está autenticado

Quando ele envia uma requisição GET para "/catalog/products"

Então o sistema deve retornar status 200

E o corpo da resposta deve conter uma lista de produtos

### **Cenário 2: Verificar estrutura de cada produto retornado**

**Cenário: Verificar campos de um produto na resposta**

Dado que o usuário realiza uma requisição GET para "/catalog/products"

Quando a resposta for recebida

Então cada produto deve conter os campos "id", "name", "description", "price", "imageUrl" e "categoryId"

### **Cenário 3: Buscar produtos com filtro de nome**

**Cenário: Buscar produtos com termo específico no nome**

Dado que o usuário deseja encontrar produtos com o termo "headphones"

Quando ele envia uma requisição GET para "/catalog/products?name=headphones"

Então o sistema deve retornar status 200

E todos os produtos retornados devem conter "headphones" no nome

### **Cenário 4: Buscar produto inexistente**

**Cenário: Buscar por um nome que não existe**

Dado que o usuário envia uma requisição GET para "/catalog/products?name=produtoInexistente123"

Então o sistema deve retornar status 200

E a lista de produtos na resposta deve estar vazia

### **Cenário 5: Realizar busca com parâmetro inválido**

**Cenário: Buscar produtos com parâmetro malformado**

Quando o usuário envia uma requisição GET para "/catalog/products?price=minhaStringInvalida"

Então o sistema deve retornar status 400 ou mensagem de erro informando parâmetro inválido

### **Cenário 6: Verificar se resultado está paginado**

**Cenário: Verificar paginação da lista de produtos**

Dado que o usuário envia uma requisição GET para "/catalog/products?page=1&limit=20"

Então o sistema deve retornar os primeiros 20 produtos

E os metadados de paginação devem estar presentes no cabeçalho ou corpo da resposta