



Olá, estudante.

A seguir, você dará continuidade ao desenvolvimento da sua solução através dos campos específicos para a resolução dos 3 desafios propostos, lembrando que eles se complementam.

**Nome:** Carlos Felipe Dias da Silva

**RGM:** 23534214

**Documentação:**

Revisite a documentação do projeto desenvolvido na PIT I, faça as atualizações e melhorias necessárias de acordo com os pontos explicitados no material teórico. Ao terminar os ajustes, suba a documentação em um repositório GIT.

**Link do repositório:** [https://github.com/Helisy/cupcacke\\_store\\_app](https://github.com/Helisy/cupcacke_store_app)

Para cumprir o cronograma do projeto, foram implementadas algumas modificações no escopo original, implicando na remoção de determinadas funcionalidades conforme detalhado no documento do projeto. As funcionalidades excluídas são:

- **Personalização**
- **Suporte ao Cliente com Bate-Papo em Tempo Real**

Devido às limitações da hospedagem, o site não contará com a proteção SSL (Secure Sockets Layer). Portanto a comunicação segura via API com qualquer plataforma de pagamento não será possível. Como alternativa, a implementação se concentrará apenas no Frontend, com a tela de pagamento sendo criada com toda as funções sem relação com pagamento mantidas e funcionais.

### **As seguintes histórias de Usuário foram retiradas:**

ID: 9
Título: Pagamento
Requerente: Cliente
Ação: Pagar o pedido usando várias opções de pagamento
História Completa: Como cliente, quero poder pagar meu pedido usando várias opções de pagamento, incluindo cartões de crédito, para que eu tenha segurança e facilidade ao realizar o pagamento.
<b>Comentários:</b>
Critérios de aceitação:
<ul style="list-style-type: none"><li>• O aplicativo deve fornecer uma interface clara e fácil de usar para que os clientes selezionem sua opção de pagamento preferida.</li><li>• A interface de pagamento deve oferecer suporte a várias opções de pagamento, incluindo cartões de crédito, cartões de débito, net banking, carteiras móveis e outras opções conforme necessário.</li><li>• O gateway de pagamento deve lidar com informações de pagamento do cliente com segurança, incluindo detalhes de cartão de crédito e outras informações confidenciais.</li><li>• Os clientes devem receber a confirmação do pagamento e o status do pedido por e-mail ou notificação do aplicativo.</li></ul>
Regras de negócio:

- O aplicativo deve aderir a todos os requisitos regulamentares e de conformidade aplicáveis para lidar com informações de pagamento.
- Todas as informações de pagamento, incluindo detalhes de cartão de crédito e outras informações confidenciais, devem ser mantidas seguras e confidenciais o tempo todo.
- O aplicativo deve ter proteções para evitar fraudes, incluindo o uso de sistemas antifraude e ferramentas de monitoramento de fraude.

Requisito não funcional:

- A interface de pagamento deve ser rápida, confiável e fácil de usar, com o mínimo de tempo de inatividade ou problemas de conectividade.

ID: 12

Título: Personalização

Requerente: Usuário do aplicativo

Ação: Personalizar o cupcake com uma variedade de coberturas

História Completa: Como usuário do aplicativo, quero poder personalizar meu próprio cupcake com uma variedade de coberturas, recheios e coberturas, para que possa receber o cupcake perfeito para o meu gosto.

#### Comentários:

Critérios de aceitação:

- O aplicativo deve fornecer uma interface clara e fácil de usar para que os clientes selezionem suas coberturas, recheios e decorações preferidas para seus cupcakes.
- O cliente deve ser capaz de escolher entre uma variedade de opções para cada aspecto do cupcake, incluindo sabor do bolo, cobertura, recheios, confeitos e outras coberturas.
- Os cupcakes personalizados devem ser exibidos de forma clara e visualmente atraente.
- O cliente deve poder salvar, editar e visualizar seus cupcakes personalizados antes de fazer um pedido.
- O preço final do cupcake personalizado deve ser claramente exibido ao cliente antes de fazer um pedido.

Regras de negócio:

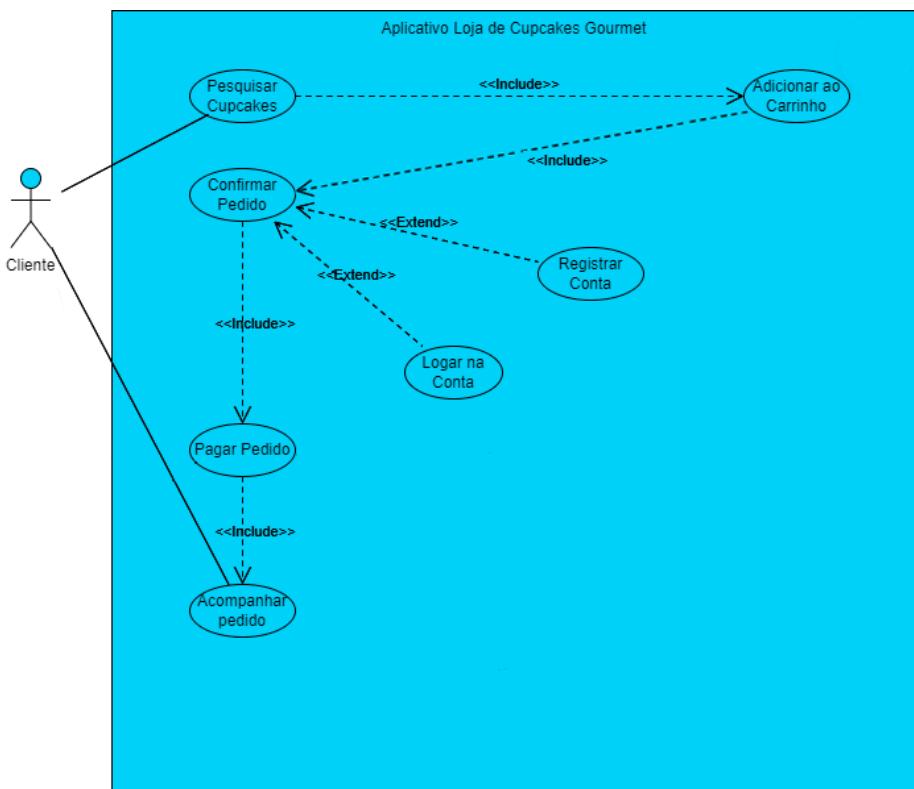
- O aplicativo deve ter dados precisos e consistentes sobre as opções disponíveis para customização de cupcakes, incluindo preços e ingredientes.
- O cliente não deve poder selecionar opções inválidas ou inexistentes para personalizar seus cupcakes.
- O cliente não deve poder fazer pedidos de cupcakes personalizados que violem quaisquer regulamentos de saúde ou segurança alimentar.

Requisito não funcional:



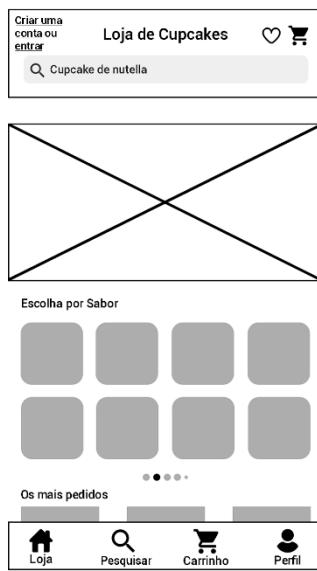
- A interface de personalização do cupcake deve ser rápida, confiável e fácil de usar, com o mínimo de tempo de inatividade ou problemas de conectividade.
- Os cupcakes personalizados devem ser exibidos de forma rápida e visualmente atraente, com descrições detalhadas e imagens de alta qualidade.
- O aplicativo deve ser capaz de lidar com um grande número de solicitações de cupcake personalizado simultaneamente, sem afetar os tempos de resposta ou o desempenho.

## Diagrama de Caso uso Geral



## Telas E Mockups

Um novo mockup foi criado para acomodar as modificações realizadas no projeto, adaptando-se ao escopo atualizado.



**Criar uma conta ou entrar** Loja de Cupcakes Heart Cartão

Últimas Pesquisas

Filtrar por Recheio

R\$ 11,23 un

Filtrar por Cobertura

R\$ 19,17 un

Escolha por Sabor

Os mais pedidos

Home Pesquisar Carrinho Perfil



**Criar uma conta ou entrar** Loja de Cupcakes Heart Cartão

Cupcake de doce de leite e chocolate R\$ 11,23 un

**Pedir**

Home Pesquisar Carrinho Perfil



**Criar uma conta ou entrar** Loja de Cupcakes Heart Cartão

Cupcake de limão com chantilly R\$ 8,50 un

Cupcake de doce de leite e chocolate R\$ 11,23 un

**Pedir**

Home Pesquisar Carrinho Perfil



**Criar uma conta ou entrar** Loja de Cupcakes Heart Cartão

Cupcake pão de mel e nutella R\$ 15,10 un

Cupcake vegano creme de avelã e damasco R\$ 19,17 un

Cupcake de doce de leite e chocolate R\$ 11,23 un

Home Pesquisar Carrinho Perfil



Produto	Preço Unitário	Quantidade	Ação
Cupcake pão de mel e nutella	R\$ 15,10 un	(3)	<span>Excluir</span>
Cupcake pão de mel e nutella	R\$ 15,10 un	(3)	<span>Excluir</span>
Cupcake pão de mel e nutella	R\$ 15,10 un	(3)	<span>Excluir</span>

Total R\$ 67,38

**Comprar**

Home Pesquisar Carrinho Perfil

Loja de Cupcakes

**Login**

Email

Senha

**Registrar**

Nome Completo

Email

Senha

**Registrar**

 Loja  Categorias  Carrinho  Perfil

**← Confirmação**

 Cupcake pão de mel e nutella  
3un - R\$ 45,30

 Cupcake vegano creme de aveia e  
damascos  
1un - R\$ 19,17

 Cupcake de doce de leite e chocolate  
6un - R\$ 67,38  
Customização:  
xxxx - R\$ XX,XX xxxx - R\$ XX,XX

Endereço e Data de Entrega  
Rua XXXXX XXXXXX n°, XXXXX-XX. Cep XXXXXXXX  
XX/XX/XX XX:XX

Método de Pagamento:

Cartão de Crédito  
 Pix  
 Boleto

Total: R\$ 131,85 **Finalizar**

 Loja  Categorias  Carrinho  Perfil

 Loja de Cupcakes  

 Cupcake de nutella

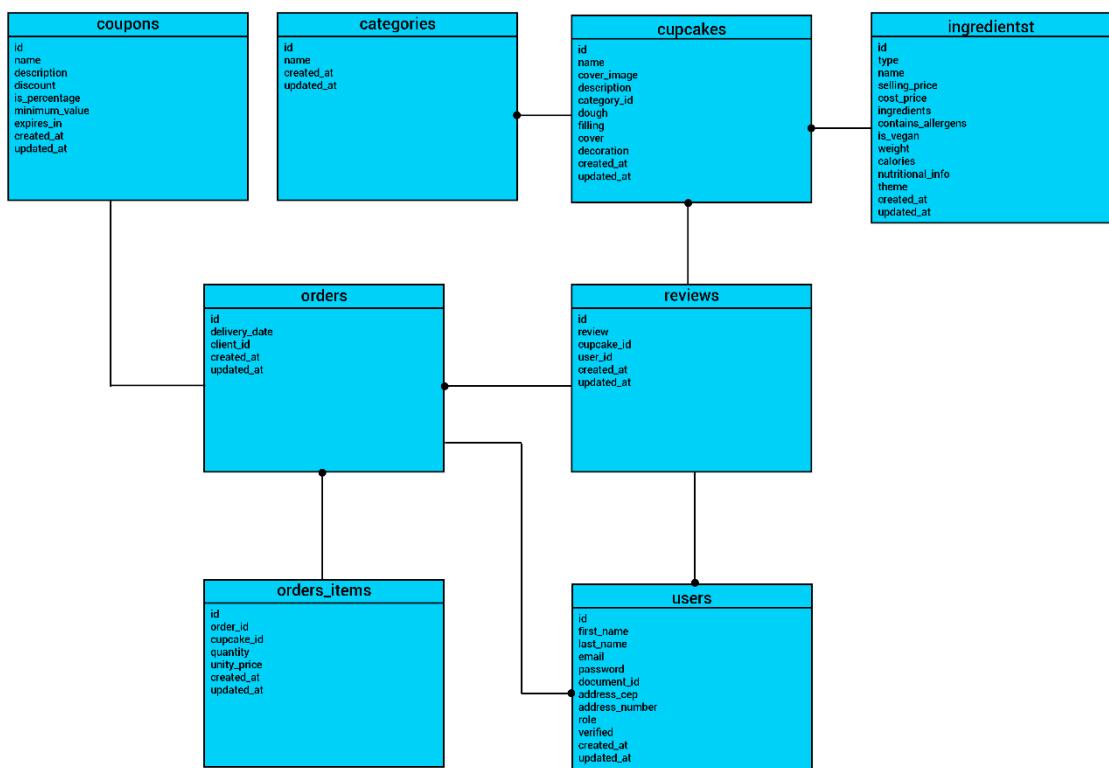


Pedido Realizado



## Projeto Lógico Normalizado

O projeto do banco de dados passou por modificações. Essas alterações visam aprimorar a estrutura e funcionalidade do banco de dados, refletindo uma atualização no desenho da arquitetura do sistema.



## Banco de Dados e Dicionários de dados.

O banco de dados foi desenvolvido utilizando MySQL, sendo projetado com o intuito de simplificar a personalização dos cupcakes. Embora essa funcionalidade não tenha sido implementada, a estrutura do banco de dados que foi criada, facilita a criação e adição de novas opções no catálogo, proporcionando flexibilidade para futuras atualizações.

### Dicionário de dados:

Tabela	users			
Descrição	Guarda as informações pessoais e de login do usuário			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL AUTO_INCREMENT
first_name	Prenome	VARCHAR	255	NOT NULL
last_name	Sobrenome	VARCHAR	255	NOT NULL
email	Email	VARCHAR	255	NOT NULL
password	Senha encriptada (Hash)	VARCHAR	255	NOT NULL
document_id	Guada o CPF do usuário	VARCHAR	255	NOT NULL
address_cep	Guada o CEP do usuário	VARCHAR	255	NOT NULL

address_number	Guarda número do imóvel	VARCHAR	255	NOT NULL
role	Nível de Acesso	VARCHAR	255	DEFAULT 'basic'
verified	Guarda se o usuário foi verificado	BOOLEAN		DEFAULT FALSE
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

Tabela	cupcakes			
Descrição	Guarda nome, imagem de capa, categoria, descrição e os ingredientes dos cupcakes.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL
name	Titulo do cupcake	VARCHAR	255	NOT NULL
cover_image	Url da imagem de capa	VARCHAR	255	NOT NULL
description	Descrição do cupcake	TEXT		

category_id	Id da categoria	INT		
dough	Id do ingrediente massa	INT		NOT NULL
filling	Id do ingrediente recheio	INT		NOT NULL
cover	Id do ingrediente cobertura	INT		NOT NULL
decoration	Id do ingrediente decoração	INT		
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

Tabela	ingredients			
Descrição	Guarda as informações sobre os ingredientes			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL AUTO_INCREMENT
type	Guarda o tipo de ingrediente (massa, recheio,	VARCHAR	255	NOT NULL

	(cobertura e decoração)			
name	Titulo do ingrediente	VARCHAR	255	NOT NULL
selling_price	Valor de venda	REAL		NOT NULL
cost_price	Valor do custo	REAL		NOT NULL
ingredients	Os ingredientes do item	VARCHAR	255	
contains_allergens	Guarda se contém alergênicos	BOOLEAN		NOT NULL DEFAULT TRUE
is_vegan	Guarda se é vegano	BOOLEAN		NOT NULL DEFAULT FALSE
weight	Peso	REAL		NOT NULL
calories	Quantida de calorias	REAL		NOT NULL
nutritional_info	Informações nutricionais	VARCHAR	255	NOT NULL
theme	Tema do ingrediente	VARCHAR	255	
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

<b>Tabela</b>	orders			
<b>Descrição</b>	Guarda os pedidos			
<b>Observações</b>	Contém chave estrangeira client_id com a tabela users			
<b>Campos</b>				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.			
delivery_date	Data e horário da entrega	TIMESTAMP		NOT NULL
client_id	Id do da tabela user	INT		NOT NULL
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

<b>Tabela</b>	orders_items			
<b>Descrição</b>	Guarda os itens dentro de um pedido			
<b>Observações</b>	Contém chave estrangeira cupcake_id _id com a tabela cupcakes			
<b>Campos</b>				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL AUTO_INCREMENT

order_id	Id da tabela orders	INT		NOT NULL
cupcake_id	Id da tabela cupcake	INT		NOT NULL
quantity	Quantidades de unidades	INT		NOT NULL
unity_price	Preço da unidade cupcake	REAL		NOT NULL
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

<b>Tabela</b>	coupons			
<b>Descrição</b>	Guarda os cupons			
<b>Observações</b>				
<b>Campos</b>				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL AUTO_INCREMENT
name	Nome do Cupom	VARCHAR	255	NOT NULL
description	Descrição do cupom	VARCHAR	255	NOT NULL
discount	Valor do desconto	REAL		NOT NULL

is_percentage	Guarda se o cupom desconta porcentagem	BOOLEAN		NOT NULL
minimum_value	Valor mínimo para usar o cupom	REAL		NOT NULL
expires_in	Data de expiração do cupom	DATETIME		NOT NULL
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

Tabela	categories			
Descrição	Categorias			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	INT		NOT NULL AUTO_INCREMENT
name	Nome da categoria	VARCHAR	255	NOT NULL
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP

updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
------------	--------------------------------	----------	--	--

Tabela	reviews			
Descrição	Avaliações dos cupcakes			
Observações	Contém chave estrangeira cupcake_id _id com a tabela cupcakes e contém chave estrangeira user_id _id com a tabela users			
<b>Campos</b>				
Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Chave Primária.	int		not null auto_increment
review	Avaliação do usuário	varchar	255	not null
cupcake_id	Id da tabela cupcake	int		not null
user_id	Id da tabela users	int		not null
created_at	Data de criação da entrada.	TIMESTAMP		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
updated_at	Data de atualização da entrada	DATETIME		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

## Documentação

O servidor disponibiliza as telas por meio de arquivos HTML, estilos CSS e a lógica do frontend (Javascript), além de fornecer a API para comunicação com o banco de dados. A seguir, encontram-se detalhadas cada rota e sua respectiva função.

### Rotas das telas:

- /
- /product
- /cart
- /payment
- /payment/sucess
- /auth/login
- /auth/register
- /search
- /user
- /product/:id
- /category
- /category/:id

## API

### /api/v1/auth

#### POST

Para utilizar a API, é necessário possuir um "accessToken". Para obtê-lo, é necessário efetuar uma chamada para a seguinte URL, enviando um corpo em formato JSON, conforme exemplificado abaixo.

**Exemplo:**

```
curl -X POST url_server/api/v1/auth/login -d
{
    "email": "usuário@hotmail.com",
    "password": "123456789abc"
}
```

Um corpo em JSON será retornado com o status 200 e um o “accessToken” estará disponível nos Cookies da resposta.

**Resposta:**

```
{
    "method": "POST",
    "error": false,
    "code": 200,
    "message": "Log in was sucessfully.",
    "links": []
}
```

Cookie com “accessToken”:

NAME	VALUE
accessToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmaXJzdE5hbWUiOiJHYWJyaWVsYSBFbGlzZSlmhc3ROYW1lIjoiRGlhcyBEYSBTaWx2YSIsInVzZXJJZCI6MSwicm9sZSI6ImFkbWluliwidmVyaWZpZWQiOjAsImIhdCI6MTcwMDA0NzI0N30.59-X28iMPe6OUOZbTdQA2WI_i4ByiPRJtSpjs15U_9M

## /api/v1/auth

### POST

Para realizar o processo de login, é necessário criar um registro de conta por meio de uma chamada a url, utilizando um corpo em formato JSON, conforme exemplificado abaixo.

O corpo dessa chamada deve conter:

- `firstName`: Campo obrigatório, tipo “string”.
- `lastName`: Campo obrigatório, tipo “string”.
- `email`: Campo obrigatório, tipo “string” com formatação email.
- `password`: Campo obrigatório, tipo “string”.
- `document_id`: Campo obrigatório, tipo “string”.
- `address_cep`: Campo obrigatório, tipo “string”.
- `address_number`: Campo obrigatório, tipo “string”.

### Exemplo:

```
curl -X POST url_server/api/v1/auth/register -d
'{
    "firstName": "Marcelo",
    "lastName": "Faria",
    "email": "usuário@hotmail.com",
    "password": "123456789abc",
    "document_id": "12345678910",
    "address_cep": "12346789",
    "address_number": "132"
}'
```

**Resposta:**

```
{  
    "method": "POST",  
    "error": false,  
    "code": 201,  
    "message": "Register new user has been successfully.",  
    "links": []  
}
```

**/api/v1/category**

**GET**

Uma chamada GET retorna todas categorias cadastradas.

**Exemplo:**

```
curl -X GET url_server/api/v1/category/
```

**Resposta:**

```
{  
    "method": "GET",  
    "error": false,  
    "code": 201,  
    "message": "",  
    "data": [  
        {  
            "id": 1,  
            "name": "Vegano",  
            "created_at": "2023-11-15T16:44:55.000Z",  
            "updated_at": "2023-11-15T16:44:55.000Z"  
        },  
        {  
            "id": 2,  
            "name": "Chocolate",  
            "created_at": "2023-11-15T16:45:12.000Z",  
            "updated_at": "2023-11-15T16:45:12.000Z"  
        },  
        {  
            "id": 3,  
            "name": "Frutas",  
            "created_at": "2023-11-15T17:25:04.000Z",  
            "updated_at": "2023-11-15T17:25:04.000Z"  
        }  
    "links": []  
}
```

## GET

Para obter uma categoria específica, uma chamada com “category\_id” deve ser realizada.

### Chamada:

```
curl -X GET url_server/api/v1/category/$category_id
```

### Exemplo:

```
curl -X GET url_server/api/v1/category/1
```

### Resposta:

```
{
    "method": "GET",
    "error": false,
    "code": 201,
    "message": "",
    "data": [
        {
            "id": 1,
            "name": "Vegano",
            "created_at": "2023-11-15T16:44:55.000Z",
            "updated_at": "2023-11-15T16:44:55.000Z"
        }
    ],
    "links": []
}
```

## POST

Para cadastrar uma nova categoria, é necessário efetuar uma chamada POST para a seguinte url, utilizando um corpo em formato JSON. Apenas um usuário com nível de acesso "admin" (campo "role" na tabela users do banco de dados) tem permissão para realizar determinadas chamadas POST, como é o caso desta. Para obter um "accessToken" com privilégios de administrador, utilize as credenciais de login fornecidas abaixo. Este acesso será necessário em algumas chamadas subsequentes.

Credenciais para geração do "accessToken":

**Usuário:** administrador@adm.com

**Senha:** admin@!master56

O corpo dessa chamada deve conter:

- name: Campo obrigatório, tipo "string".

### Exemplo:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/category -d
'{'
    "name": "Frutas"
}'
```

### Resposta:

```
{
    "method": "POST",
    "error": false,
    "code": 201,
    "message": "New category created sucessfully",
    "links": []
}
```

## /api/v1/coupon

### GET

Um usuário com o nível de acesso "admin" tem a capacidade de visualizar todos os cupons cadastrados por meio desta chamada específica.

### Exemplo:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/coupon
```

### Resposta:

```
{
    "method": "GET",
    "error": false,
    "code": 201,
    "message": "",
    "data": [
        {
            "id": 1,
            "name": "BEMVINDO5",
            "description": "5% de desconto em todos os cupcakes após uma compra acima de R$ 100,00.",
            "discount": 5,
            "is_percentage": 1,
            "minimum_value": 5,
            "expires_in": "2030-12-30T15:00:00.000Z",
            "created_at": "2023-11-15T10:59:42.000Z",
            "updated_at": "2023-11-15T10:59:42.000Z"
        },
    ],
    "links": []
}
```

## GET

Qualquer usuário, independentemente de estar autenticado no sistema ou não, tem a capacidade de consultar um cupom específico por meio do seu identificador (id).

### Chamada:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/coupon /$coupon_id
```

### Exemplo:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/coupon /1
```

### Resposta:

```
{
    "method": "GET",
    "error": false,
    "code": 201,
    "message": "",
    "data": [
        {
            "id": 1,
            "name": "BEMVINDO5",
            "description": "5% de desconto em todos os cupcakes após uma compra acima de R$ 100,00.",
            "discount": 5,
            "is_percentage": 1,
            "minimum_value": 5,
            "expires_in": "2030-12-30T15:00:00.000Z",
            "created_at": "2023-11-15T10:59:42.000Z",
            "updated_at": "2023-11-15T10:59:42.000Z"
        }
    ]
}
```

```
],
"links": []
}
```

## POST

Um usuário com o nível de acesso "admin" pode cadastrar um novo cupom, conforme exemplo abaixo.

O corpo dessa chamada deve conter:

- name: Campo obrigatório, tipo "string".
- description: Campo obrigatório, tipo "string".
- discount: Campo obrigatório, tipo "float".
- is\_percentage: Campo obrigatório, tipo "boolean".
- minimum\_value: Campo obrigatório, tipo "float".
- expires\_in: Campo obrigatório, tipo "datetime".

### Exemplo:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/coupon -d
'{
    "name": "BLACKFRYDAY10",
    "description": "Cupom de black fryday com 10% de desconto em todos os cupcakes",
    "discount": 10,
    "is_percentage": true,
    "minimum_value": 50,
    "expires_in": "2023-12-01T12:00:00"
}'
```

**Resposta:**

```
{  
    "method": "POST",  
    "error": false,  
    "code": 201,  
    "message": "New coupon created sucessfully",  
    "links": []  
}
```

**/api/v1/ingredient**

**GET**

Através desta chamada, todos os ingredientes cadastrados são retornados.

**Exemplo:**

```
curl -X GET -H 'token $accessToken' url_server/api/v1/ingredient
```

**Resposta:**

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": [  
        {  
            "id": 33,  
            "type": "decoration",  
            "name": "Raspas de Laranja",  
            "selling_price": 0.3,  
            "unit": "kg",  
            "description": "Raspas de laranja para decoração de sobremesas e doces.",  
            "image": "https://url_server/images/ingredientes/33.jpg",  
            "status": "available",  
            "stock": 100,  
            "created_at": "2023-10-05T10:00:00Z",  
            "updated_at": "2023-10-05T10:00:00Z"  
        }  
    ]  
}
```



```
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
],  
"links": []  
}  
  
GET
```

Através desta chamada, servindo um “ingrediente\_id” pode-se obter um ingrediente específico.

**Chamada:**

```
curl -X GET -H 'token $accessToken' url_server/api/v1/ingredient/$ingrediente_id
```

**Exemplo:**

```
curl -X GET -H 'token $accessToken' url_server/api/v1/ingredient/$17
```

## Resposta:

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": {  
        "id": 17,  
        "type": "cover",  
        "name": "Chantilly Vegano",  
        "selling_price": 2.5,  
        "cost_price": 0.75,  
        "ingredients": "creme de leite vegetal, açúcar",  
        "contains_allergens": 0,  
        "is_vegan": 1,  
        "weight": 15,  
        "calories": 50,  
        "nutritional_info": "Não contém alergênicos",  
        "theme": null,  
        "created_at": "2023-11-15T12:41:31.000Z",  
        "updated_at": "2023-11-15T12:41:31.000Z"  
    },  
    "links": []  
}
```

## POST

Um usuário com o nível de acesso "admin" pode cadastrar um novo ingrediente, conforme exemplo abaixo.

O corpo dessa chamada deve conter:

- type: Campo obrigatório, tipo “string”, valores “dough, cover, filling, decoration”.
- name: Campo obrigatório, tipo “string”.
- selling\_price: Campo obrigatório, tipo “float”.
- cost\_price: Campo obrigatório, tipo “float”.
- ingredientes: Campo obrigatório, tipo “string”.
- weight: Campo obrigatório, tipo “int”.
- contains\_allergens: Campo não obrigatório, tipo “boolean”.
- is\_vegan: Campo não obrigatório, tipo “boolean”.
- calories: Campo obrigatório, tipo “int”.
- nutritional\_info: Campo obrigatório, tipo “string”.
- theme: Campo não obrigatório, tipo “string”.

**Exemplo:**

```
curl -X POST -H 'token $accessToken' url_server/api/v1/ingredient -d
'{
    "type": "dough",
    "name": "Massa de Baunilha Vegana",
    "selling_price": 3.5,
    "cost_price": 2.3,
    "ingredients": "essênciça de baunilha, óleo de soja, trigo, açúcar, fermento quí-
mico, vinagre de vinho branco, leite de amendoas, sal",
    "weight": 100,
    "contains_allergens": 1,
    "is_vegan": 1,
    "calories": 150,
    "nutritional_info": "Contém alergênicos: gluten, leite, pode conter traços de soja
e amêndoas"
}'
```

## /api/v1/cupcake

### GET

Através desta chamada, todos os ingredientes cadastrados são retornados.

#### Exemplo:

```
curl -X GET url_server/api/v1/cupcake
```

#### Resposta:

```
{
  "method": "GET",
  "error": false,
  "code": 200,
  "message": "",
  "data": [
    {
      "id": 8,
      "name": "Cupcake de Chocolate Com Cobertura de Lí-
mão",
      "cover_image": "https://i.ibb.co/PMn94Gw/MP-Tortas-
Cupcake-de-caipirinha-FILICO.png",
      "description": "<p>Explore uma fusão surpreendente de
indulgência e frescor com o nosso Cupcake de Chocolate com Cobertura de Limão.
Cada mordida é uma jornada única, onde a riqueza do chocolate se encontra com a
acidez refrescante do limão, criando uma experiência de sabor verdadeiramente me-
morável.</p> <p>A base do cupcake é uma massa de chocolate indulgente, rica em
cacau e com uma textura macia que derrete na boca. O contraste começa com a ge-
nerosa cobertura de limão, onde a doçura e a acidez se entrelaçam de maneira per-
feita. Esta combinação inusitada cria uma sinfonia de sabores que surpreendem e en-
cantam.</p> <p>A cobertura, leve e cremosa, é acentuada pelas raspas frescas de li-
mão, adicionando uma camada extra de frescor e um toque cítrico que eleva o
```

cupcake a um patamar superior de sofisticação.</p> <p>Perfeito para aqueles que buscam uma experiência de sabor única e inovadora, o Cupcake de Chocolate com Cobertura de Limão é ideal para momentos especiais ou para simplesmente satisfazer seus desejos por uma sobremesa que vai além das expectativas. Permita-se explorar esta combinação inesperada de sabores e descubra como o chocolate e o limão podem dançar juntos em perfeita harmonia.</p>,

```
"category_id": 2,  
"dough": {  
    "id": 6,  
    "type": "dough",  
    "name": "Massa de Chocolate",  
    "selling_price": 3.6,  
    "cost_price": 2.2,  
    "ingredients": "chocolate meio amargo, ovos, óleo de soja, trigo, açúcar, fermento químico, sal",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 100,  
    "calories": 200,  
    "nutritional_info": "Contém alergênicos: gluten, leite, chocolate, pode conter traços de soja e amendoas",  
    "theme": null,  
    "created_at": "2023-11-15T12:40:36.000Z",  
    "updated_at": "2023-11-15T12:40:36.000Z"  
},  
"filling": {  
    "id": 24,  
    "type": "filling",  
    "name": "Limão",  
    "selling_price": 1.5,  
    "cost_price": 0.6,  
    "ingredients": "creme de leite fresco, açúcar, limão",  
}
```

```
"contains_allergens": 1,  
"is_vegan": 0,  
"weight": 30,  
"calories": 50,  
"nutritional_info": "Contém alergênicos: leite",  
"theme": null,  
"created_at": "2023-11-15T12:42:14.000Z",  
"updated_at": "2023-11-15T12:42:14.000Z"  
},  
"cover": {  
    "id": 15,  
    "type": "cover",  
    "name": "Limão",  
    "selling_price": 1,  
    "cost_price": 0.3,  
    "ingredients": "creme de leite fresco, açúcar,  
limão",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 15,  
    "calories": 15,  
    "nutritional_info": "Contém alergênicos: leite",  
    "theme": null,  
    "created_at": "2023-11-15T12:41:21.000Z",  
    "updated_at": "2023-11-15T12:41:21.000Z"  
},  
"decoration": {  
    "id": 32,  
    "type": "decoration",  
    "name": "Raspas de Limão",  
    "selling_price": 0.3,  
    "cost_price": 0.1,  
    "ingredients": "Açucar, corante, Limão",
```

```
        "contains_allergens": 1,  
        "is_vegan": 1,  
        "weight": 2,  
        "calories": 5,  
        "nutritional_info": "Não contém alergênicos",  
        "theme": null,  
        "created_at": "2023-11-15T12:42:59.000Z",  
        "updated_at": "2023-11-15T12:42:59.000Z"  
    },  
    "created_at": "2023-11-15T16:32:42.000Z",  
    "updated_at": "2023-11-15T17:24:53.000Z",  
    "selling_price": 9.7,  
    "cost_price": 5.300000000000001,  
    "weight": 245,  
    "calories": 465  
},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
{...},  
],  
"links": [  
    {  
        "href": "/api/v1/cupcake/:id",  
        "method": "GET"  
    },  
    {  
        "href": "/api/v1/cupcake",  
        "method": "POST"  
    }  
]
```

}

## GET

Por meio desta chamada, cupcakes são recuperados utilizando um parâmetro denominado "query", representado pela chave "q". Este parâmetro recebe uma "string" de pesquisa que é utilizada para encontrar cupcakes cujos campos "name" e "ingredients" contenham a referida "string" de pesquisa.

### Chamada:

```
curl -X GET url_server/api/v1/cupcake?q=$pesquisa
```

### Exemplo:

```
curl -X GET url_server /api/v1/cupcake?q=Nutella
```

### Resposta:

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": [  
        {  
            "id": 7,  
            "name": "Cupcake de Nutella",  
            "cover_image": "https://i.ibb.co/WVCnS0X/chocolate-nutella-cup-  
cakes-3-3-scaled.png",  
            "description": "<p>Mergulhe em uma experiência indulgente com  
o nosso irresistível Cupcake de Nutella. Cada mordida é uma celebração da combina-  
ção perfeita de chocolate e avelã, transportando seus sentidos para uma jornada de  
sabor que é pura felicidade.</p> <p>A base do cupcake é uma massa macia e úmida,  
enriquecida com a intensidade do chocolate. A surpresa vem no recheio generoso de
```

Nutella, adicionando uma camada de cremosidade e uma explosão de sabor que é simplesmente divina. Cada pedaço é uma experiência única de indulgência para os verdadeiros amantes de Nutella.

A cobertura, adornada com a irresistível Nutella, oferece uma textura cremosa e um toque aveludado que derrete na boca. Decorado com pedacinhos de avelã torrada, o cupcake é uma verdadeira festa para os sentidos, combinando crocância, cremosidade e a doçura viciante da Nutella.

Perfeito para satisfazer seus desejos de chocolate ou para surpreender alguém especial, o Cupcake de Nutella é uma escolha que eleva o clássico sabor da Nutella a novas alturas de prazer. Deixe-se envolver pela indulgência e saboreie cada pedaço deste cupcake que é uma verdadeira ode ao irresistível casamento de chocolate e avelã.

```
"dough": {  
    "id": 6,  
    "type": "dough",  
    "name": "Massa de Chocolate",  
    "selling_price": 3.6,  
    "cost_price": 2.2,  
    "ingredients": "chocolate meio amargo, ovos, óleo de soja, trigo, açúcar, fermento químico, sal",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 100,  
    "calories": 200,  
    "nutritional_info": "Contém alergênicos: gluten, leite, chocolate, pode conter traços de soja e amendoas",  
    "theme": null,  
    "created_at": "2023-11-15T12:40:36.000Z",  
    "updated_at": "2023-11-15T12:40:36.000Z"  
},  
    "filling": {  
        "id": 21,  
        "type": "filling",  
        "name": "Nutella",  

```

```
"cost_price": 2.3,  
"ingredients": "açúcar, óleo de palma, avelãs, leite desna-  
tado em pó, cacau",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 30,  
    "calories": 85,  
    "nutritional_info": "Contém alergênicos: leite, avelãs",  
    "theme": null,  
    "created_at": "2023-11-15T12:41:57.000Z",  
    "updated_at": "2023-11-15T12:41:57.000Z"  
},  
"cover": {  
    "id": 12,  
    "type": "cover",  
    "name": "Nutella",  
    "selling_price": 4,  
    "cost_price": 2.3,  
    "ingredients": "açúcar, óleo de palma, avelãs, leite desna-  
tado em pó, cacau",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 20,  
    "calories": 70,  
    "nutritional_info": "Contém alergênicos: leite, avelãs",  
    "theme": null,  
    "created_at": "2023-11-15T12:41:05.000Z",  
    "updated_at": "2023-11-15T12:41:05.000Z"  
},  
"decoration": {  
    "id": 3,  
    "type": "decoration",  
    "name": "Sem decoração",
```

```
        "selling_price": 0,  
        "cost_price": 0,  
        "ingredients": "",  
        "contains_allergens": 1,  
        "is_vegan": 0,  
        "weight": 0,  
        "calories": 0,  
        "nutritional_info": "",  
        "theme": null,  
        "created_at": "2023-11-15T12:40:13.000Z",  
        "updated_at": "2023-11-15T12:40:13.000Z"  
    },  
    "created_at": "2023-11-15T13:32:01.000Z",  
    "updated_at": "2023-11-15T13:32:01.000Z",  
    "dough_name": "Massa de Chocolate",  
    "filling_name": "Nutella",  
    "cover_name": "Nutella",  
    "decoration_name": "Sem decoração",  
    "selling_price": 15.2,  
    "cost_price": 9,  
    "weight": 250,  
    "calories": 555  
}  
],  
"links": [  
    {  
        "href": "/api/v1/cupcake/:id",  
        "method": "GET"  
    },  
    {  
        "href": "/api/v1/cupcake",  
        "method": "POST"  
    }]
```

```
    ]  
}
```

## GET

Através desta chamada, cupcakes são recuperados utilizando um parâmetro denominado "query", representado pela chave "category". Este parâmetro recebe um número inteiro (int) denominado "category\_id". Todos os cupcakes associados a essa categoria específica são então retornados.

### Chamada:

```
curl -X GET url_server/api/v1/cupcake?category=$category_id
```

### Exemplo:

```
curl -X GET url_server/api/v1/cupcake?category=1
```

### Resposta:

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": [  
        {  
            "id": 4,  
            "name": "Cupcake de Laranja Vegano",  
            "cover_image": "https://i.ibb.co/m6rw2mG/Cupcake-  
870x500.png",  
            "description": "<p>Apresentamos o nosso Cupcake de Laranja  
Vegano, uma explosão cítrica que une o frescor da laranja à bondade consciente de  
uma receita totalmente vegana. Cada mordida é uma celebração da simplicidade
```

natural, sem comprometer o sabor requintado.</p> <p>A base do cupcake é um equilíbrio perfeito entre a umidade da laranja recém-espremida e a leveza da massa vegana. Esta combinação oferece uma textura macia e deliciosa que agrada tanto aos paladares vegans quanto aos não-vegans. A doçura natural da laranja se destaca, proporcionando uma experiência refrescante.</p> <p>Este cupcake de laranja vegano é a escolha ideal para aqueles que buscam uma alternativa consciente sem abrir mão do prazer. Seja para satisfazer um desejo de doce ou para compartilhar em momentos especiais, este cupcake proporciona uma experiência culinária que é gentil com o planeta e inesquecivelmente deliciosa. Deixe-se envolver pela frescura da laranja e descubra como a bondade vegana pode ser absolutamente irresistível.</p>" ,

```
"category_id": 1,  
"dough": {  
    "id": 5,  
    "type": "dough",  
    "name": "Massa de Baunilha Vegana",  
    "selling_price": 3.5,  
    "cost_price": 2.3,  
    "ingredients": "essência de baunilha, óleo de soja, trigo, açúcar, fermento químico, vinagre de vinho branco, leite de amendoas, sal",  
    "contains_allergens": 1,  
    "is_vegan": 1,  
    "weight": 100,  
    "calories": 150,  
    "nutritional_info": "Contém alergênicos: gluten, leite, pode conter traços de soja e amendoas",  
    "theme": null,  
    "created_at": "2023-11-15T12:40:31.000Z",  
    "updated_at": "2023-11-15T12:40:31.000Z"  
},  
"filling": {  
    "id": 19,  
    "type": "cover",  
    "name": "Laranja Vegano",
```

```
"selling_price": 1.8,  
"cost_price": 0.6,  
"ingredients": "creme de leite vegetal, açúcar, Laranja",  
"contains_allergens": 1,  
"is_vegan": 1,  
"weight": 15,  
"calories": 15,  
"nutritional_info": "Não contém alergênicos",  
"theme": null,  
"created_at": "2023-11-15T12:41:43.000Z",  
"updated_at": "2023-11-15T12:41:43.000Z"  
},  
"cover": {  
    "id": 2,  
    "type": "cover",  
    "name": "Sem cobertura",  
    "selling_price": 0,  
    "cost_price": 0,  
    "ingredients": "",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 0,  
    "calories": 0,  
    "nutritional_info": "",  
    "theme": null,  
    "created_at": "2023-11-15T12:40:13.000Z",  
    "updated_at": "2023-11-15T12:40:13.000Z"  
},  
"decoration": {  
    "id": 32,  
    "type": "decoration",  
    "name": "Raspas de Limão",  
    "selling_price": 0.3,
```

```
        "cost_price": 0.1,  
        "ingredients": "Açucar, corante, Limão",  
        "contains_allergens": 1,  
        "is_vegan": 1,  
        "weight": 2,  
        "calories": 5,  
        "nutritional_info": "Não contém alergênicos",  
        "theme": null,  
        "created_at": "2023-11-15T12:42:59.000Z",  
        "updated_at": "2023-11-15T12:42:59.000Z"  
    },  
    "created_at": "2023-11-15T16:32:18.000Z",  
    "updated_at": "2023-11-15T17:01:31.000Z",  
    "selling_price": 8.8,  
    "cost_price": 5.199999999999999,  
    "weight": 215,  
    "calories": 315  
},  
{...}  
],  
"links": [  
    {  
        "href": "/api/v1/cupcake/:id",  
        "method": "GET"  
    },  
    {  
        "href": "/api/v1/cupcake",  
        "method": "POST"  
    }  
]  
}
```

## GET

Através desta chamada, ao servir um “cupcake\_id” um cupcake específico é retornado.

### Chamada:

```
curl -X GET url_server/api/v1/cupcake/$cupcake_id
```

### Exemplo:

```
curl -X GET url_server/api/v1/cupcake/5
```

### Resposta:

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": {  
        "id": 5,  
        "name": "Cupcake de Doce de leite",  
        "cover_image": "https://i.ibb.co/Y80RxdD/image.png",  
        "description": "<p>Apresentamos o nosso Cupcake de Doce de Leite, uma experiência indulgente que captura a riqueza e a docura do doce de leite de maneira excepcional. Cada mordida é uma jornada sensorial, transportando você para um mundo de tradição e prazer.</p> <p>A coroa desta obra-prima é a cobertura generosa de doce de leite, que envolve cada cupcake com uma camada decadente. Cremoso, rico e intensamente indulgente, o doce de leite na cobertura eleva este cupcake a uma categoria de luxo culinário.</p> <p>Seja para comemorar momentos especiais ou para simplesmente satisfazer um desejo por docura, o Cupcake de Doce de Leite é a esco-  
lha perfeita. Cada detalhe é uma homenagem à tradição do sabor, proporcionando uma experiência que transcende o comum e deixa um rastro de deleite na memória de"
```

quem prova. Permite-se a indulgência de um clássico reinventado e celebre a riqueza do doce de leite de uma maneira única e deliciosamente memorável.</p>>,

```
"category_id": 3,  
"dough": {  
    "id": 4,  
    "type": "dough",  
    "name": "Massa de Baunilha",  
    "selling_price": 2.3,  
    "cost_price": 1.7,  
    "ingredients": "ovos, essência de baunilha, óleo de soja, trigo, açúcar, leite, fermento químico",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 100,  
    "calories": 170,  
    "nutritional_info": "Contém alergênicos: gluten, leite, pode conter traços de soja e amendoas",  
    "theme": null,  
    "created_at": "2023-11-15T12:40:26.000Z",  
    "updated_at": "2023-11-15T12:40:26.000Z"  
},  
"filling": {  
    "id": 22,  
    "type": "filling",  
    "name": "Doce de leite",  
    "selling_price": 3,  
    "cost_price": 1.9,  
    "ingredients": "doce de leite",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 40,  
    "calories": 100,  
    "nutritional_info": "Contém alergênicos: leite",
```

```
"theme": null,  
"created_at": "2023-11-15T12:42:02.000Z",  
"updated_at": "2023-11-15T12:42:02.000Z"  
},  
"cover": {  
    "id": 22,  
    "type": "filling",  
    "name": "Doce de leite",  
    "selling_price": 3,  
    "cost_price": 1.9,  
    "ingredients": "doce de leite",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 40,  
    "calories": 100,  
    "nutritional_info": "Contém alergênicos: leite",  
    "theme": null,  
    "created_at": "2023-11-15T12:42:02.000Z",  
    "updated_at": "2023-11-15T12:42:02.000Z"  
},  
"decoration": {  
    "id": 3,  
    "type": "decoration",  
    "name": "Sem decoração",  
    "selling_price": 0,  
    "cost_price": 0,  
    "ingredients": "",  
    "contains_allergens": 1,  
    "is_vegan": 0,  
    "weight": 0,  
    "calories": 0,  
    "nutritional_info": "",  
    "theme": null,
```

```
        "created_at": "2023-11-15T12:40:13.000Z",
        "updated_at": "2023-11-15T12:40:13.000Z"
    },
    "created_at": "2023-11-15T16:32:24.000Z",
    "updated_at": "2023-11-15T17:25:25.000Z",
    "selling_price": 10.600000000000001,
    "cost_price": 7.2,
    "weight": 280,
    "calories": 540
},
"links": []
}
```

## POST

Um usuário com o nível de acesso "admin" pode cadastrar um novo cupcake, conforme exemplo abaixo.

O corpo dessa chamada deve conter:

- name: Campo obrigatório, tipo “string”.
- cover\_image: Campo obrigatório, tipo “string”, formatação url.
- description: Campo obrigatório, tipo “string”.
- dough\_id: Campo obrigatório, tipo “int”.
- filling\_id: Campo obrigatório, tipo “int”.
- cover\_id: Campo obrigatório, tipo “int”.
- decoration\_id: Campo obrigatório, tipo “int”.

## Exemplo:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/cupcake -d
'{
    "name": "Cupcake de Chocolate Com Cobertura de Limão",
    "cover_image": "https://i.ibb.co/PMn94Gw/MP-Tortas-Cupcake-de-caipirinha-FIL-
ICO.png",
```

"description": "<p>Explore uma fusão surpreendente de indulgência e frescor com o nosso Cupcake de Chocolate com Cobertura de Limão. Cada mordida é uma jornada única, onde a riqueza do chocolate se encontra com a acidez refrescante do limão, criando uma experiência de sabor verdadeiramente memorável.</p> <p>A base do cupcake é uma massa de chocolate indulgente, rica em cacau e com uma textura macia que derrete na boca. O contraste começa com a generosa cobertura de limão, onde a docura e a acidez se entrelaçam de maneira perfeita. Esta combinação inusitada cria uma sinfonia de sabores que surpreendem e encantam.</p> <p>A cobertura, leve e cremosa, é acentuada pelas raspas frescas de limão, adicionando uma camada extra de frescor e um toque cítrico que eleva o cupcake a um patamar superior de sofisticação.</p> <p>Perfeito para aqueles que buscam uma experiência de sabor única e inovadora, o Cupcake de Chocolate com Cobertura de Limão é ideal para momentos especiais ou para simplesmente satisfazer seus desejos por uma sobremesa que vai além das expectativas. Permita-se explorar esta combinação inesperada de sabores e descubra como o chocolate e o limão podem dançar juntos em perfeita harmonia.</p>",

```
    "dough_id": 6,  
    "filling_id": 24,  
    "cover_id": 15,  
    "decoration_id": 32  
}'
```

**Resposta:**

```
{  
    "method": "POST",  
    "error": false,  
    "code": 201,  
    "message": "New cupcake created sucessfully",  
    "links": []  
}
```

## GET

Através desta chamada, todas as avaliações de um cupcake específico são retornadas.

### Chamada:

```
curl -X GET url_server/api/v1/cupcake/$cupcake_id/review
```

### Exemplo:

```
curl -X GET url_server/api/v1/cupcake/5/review
```

### Retorno:

```
{
    "method": "GET",
    "error": false,
    "code": 200,
    "message": "",
    "data": [
        {
            "id": 1,
            "review": "Mds foi a melhor coisa que ja comi na vida, comprem!  
super recomendo!",
            "cupcake_id": 5,
            "user_id": 1,
            "created_at": "2023-11-15T19:21:30.000Z",
            "updated_at": "2023-11-15T19:21:30.000Z",
            "first_name": "Fernanda Soares"
        }
    ],
    "links": []
}
```

## POST

Através desta chamada, um usuário autenticado que tenha realizado um pedido de um cupcake específico tem a capacidade de publicar sua avaliação sobre o referido cupcake.

O corpo dessa chamada deve conter:

- review: Campo obrigatório, tipo “string”.

### Chamada:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/cupcake/$cupcake_id/review
```

### Exemplo:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/cupcake/5/review -d
-d
{
    "review": "Muito boomm!"
}
```

### Retorno:

```
{
    "method": "POST",
    "error": true,
    "code": 400,
    "message": "This user already reviewed this item.",
    "details": [],
    "hints": [],
    "links": []
}
```

## /api/v1/orders

### GET

Um usuário com o nível de acesso "admin" pode visualizar todos os pedidos realizados através desta chamada.

#### Exemplo:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/orders
```

#### Resposta:

```
{
    "method": "GET",
    "error": false,
    "code": 201,
    "message": "",
    "data": [
        {
            "id": 3,
            "delivery_date": "2023-11-15T20:49:21.000Z",
            "client_id": 1,
            "created_at": "2023-11-15T19:49:21.000Z",
            "updated_at": "2023-11-15T19:49:21.000Z",
            "items": [
                {
                    "id": 4,
                    "order_id": 3,
                    "cupcake_id": 6,
                    "quantity": 1,
                    "unity_price": 0,
                    "created_at": "2023-11-15T19:49:21.000Z",
                    "updated_at": "2023-11-15T19:49:21.000Z"
                }
            ]
        }
    ]
}
```

```
        }
    ],
},
{
    "id": 2,
    "delivery_date": "2023-11-15T20:36:53.000Z",
    "client_id": 1,
    "created_at": "2023-11-15T19:36:53.000Z",
    "updated_at": "2023-11-15T19:36:53.000Z",
    "items": [
        {
            "id": 3,
            "order_id": 2,
            "cupcake_id": 8,
            "quantity": 2,
            "unity_price": 0,
            "created_at": "2023-11-15T19:36:53.000Z",
            "updated_at": "2023-11-15T19:36:53.000Z"
        }
    ]
},
{
    "id": 1,
    "delivery_date": "2023-11-15T18:29:04.000Z",
    "client_id": 1,
    "created_at": "2023-11-15T17:29:04.000Z",
    "updated_at": "2023-11-15T17:29:04.000Z",
    "items": [
        {
            "id": 1,
            "order_id": 1,
            "cupcake_id": 5,
            "quantity": 2,
```

```
        "unity_price": 0,  
        "created_at": "2023-11-15T17:29:04.000Z",  
        "updated_at": "2023-11-15T17:29:04.000Z"  
    },  
    {  
        "id": 2,  
        "order_id": 1,  
        "cupcake_id": 7,  
        "quantity": 1,  
        "unity_price": 0,  
        "created_at": "2023-11-15T17:29:04.000Z",  
        "updated_at": "2023-11-15T17:29:04.000Z"  
    }  
]  
}  
],  
"links": []  
}
```

## GET

Esta chamada retorna todos os pedidos do usuário logado.

### Exemplo:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/orders/current
```

### Resposta:

```
{  
    "method": "GET",  
    "error": false,  
    "code": 201,
```

```
"message": "",  
"data": [  
    {  
        "id": 1,  
        "delivery_date": "2023-11-15T18:29:04.000Z",  
        "client_id": 1,  
        "created_at": "2023-11-15T17:29:04.000Z",  
        "updated_at": "2023-11-15T17:29:04.000Z",  
        "items": [  
            {  
                "id": 1,  
                "order_id": 1,  
                "cupcake_id": 5,  
                "quantity": 2,  
                "unity_price": 0,  
                "created_at": "2023-11-15T17:29:04.000Z",  
                "updated_at": "2023-11-15T17:29:04.000Z"  
            },  
            {  
                "id": 2,  
                "order_id": 1,  
                "cupcake_id": 7,  
                "quantity": 1,  
                "unity_price": 0,  
                "created_at": "2023-11-15T17:29:04.000Z",  
                "updated_at": "2023-11-15T17:29:04.000Z"  
            }  
        ]  
    },  
    [],  
    {"links": []}  
}
```

## GET

Por meio desta chamada, um usuário com o nível de acesso "admin" pode recuperar um pedido específico ao fornecer o "order\_id".

### Chamada:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/orders/order_id
```

### Exemplo:

```
curl -X GET -H 'token $accessToken' url_server/api/v1/orders/4
```

### Resposta:

```
{
    "method": "GET",
    "error": false,
    "code": 201,
    "message": "",
    "data": [
        {
            "id": 4,
            "delivery_date": "2023-11-15T21:13:32.000Z",
            "client_id": 1,
            "created_at": "2023-11-15T20:13:32.000Z",
            "updated_at": "2023-11-15T20:13:32.000Z",
            "items": [
                {
                    "id": 5,
                    "order_id": 4,
                    "cupcake_id": 2,
                    "quantity": 1,
                    "unity_price": 0,
                    "total": 2.0
                }
            ]
        }
    ]
}
```

```
        "created_at": "2023-11-15T20:13:32.000Z",
        "updated_at": "2023-11-15T20:13:32.000Z"
    }
]
}
],
"links": []
}
```

## POST

Através desta chamada, um novo pedido é criado.

O corpo dessa chamada deve conter:

- `delivery_date`: Campo obrigatório, tipo “datetime”.
- `items`: Campo obrigatório, tipo “array”, deve conter objetos.
- `coupon`: Campo não obrigatório, tipo “string”.

### Exemplo:

```
curl -X POST -H 'token $accessToken' url_server/api/v1/cupcake/5/review -d
'{
    "delivery_date": "2023-11-19T14:49:57",
    "items": [
        {
            "product_id": 1,
            "quantity": 2
        }
    ],
}'
```

**Resposta:**

```
{  
    "method": "POST",  
    "error": false,  
    "code": 201,  
    "message": "",  
    "data": "Order created sucessfully.",  
    "links": []  
}
```

## **/api/v1/user**

### **GET**

Através desta chamada, o endereço e número do usuário logado é retornado.

**Exemplo:**

```
curl -X GET -H 'token $accessToken' url_server/api/v1/users/current/address
```

**Resposta:**

```
{  
    "method": "GET",  
    "error": false,  
    "code": 200,  
    "message": "",  
    "data": {  
        "address_cep": "12345678",  
        "address_number": "132"  
    },  
    "links": []  
}
```

## Testes Backend e Frontend

### Backend

Para a realização dos testes, foram utilizados os módulos Jest e SuperTest para Node.js. O código dos testes está disponível no repositório Git no endereço: "root/tests/main.test.js".

A seguir, estão detalhados os testes executados nas rotas funcionais da API.

#### /api/v1/auth

Neste teste, foram avaliadas as rotas de criação de usuário e login. Os critérios de aceitação para este teste são:

##### Rota: POST /api/v1/auth/register

##### Cenário A:

Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

##### Cenário B:

Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação do novo usuário.

**Rota: POST /api/v1/auth/login****Cenário A:**

Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 200 indicando o sucesso do processo de login do usuário e também o "accessToken" disponível do cabeçalho da resposta.

Conclusão dos testes após a execução do comando "npm test". O símbolo de correto em verde indica o sucesso do teste, juntamente com a descrição correspondente.

```
/api/v1/auth
  Register new user
    ✓ POST /api/v1/auth/register - Without body - Should respond with status 400 and send body with details array (80 ms)
    ✓ POST /api/v1/auth/register - With body - Should respond with status 201 (197 ms)
  Logs login
    ✓ POST /api/v1/auth/login - Without body - Should respond with status 400 and send body with details array (8 ms)
    ✓ POST /api/v1/auth/login - With body - Should respond with status 200 and a accessToken cookie (88 ms)
```

## /api/v1/category

Neste teste, foram avaliadas as rotas de retorno e criação e novas categorias. Os critérios de aceitação para este teste são:

**Rota: GET /api/v1/category**

**Rota: GET /api/v1/category/:id**

**Cenário A:**

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: POST /api/v1/category**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de uma nova categoria.

**Cenário C:**

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

#### Cenário D:

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

#### Conclusão dos testes:

```
/api/v1/category
User isn't authenticated
  ✓ GET /api/v1/category - Should respond with status 200 and send body with data array (9 ms)
  ✓ GET /api/v1/category/:id - Should respond with status 200 and send body with data array (8 ms)
User is admin
  ✓ POST /api/v1/category - Without body - Should respond with status 400 and send body with details array (9 ms)
  ✓ POST /api/v1/category - With body - Should respond with status 201 (10 ms)
User is not admin
  ✓ POST /api/v1/category - Without body - Should respond with status 401 and send body with details and hints (7 ms)
  ✓ POST /api/v1/category - With body - Should respond with status 401 and send body with details and hints (7 ms)
```

## /api/v1/coupon

Neste teste, foram avaliadas as rotas de retorno e criação e novas cupons. Os critérios de aceitação para este teste são:

### GET /api/v1/coupon

#### Cenário A:

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 403, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Rota: GET /api/v1/coupon**

**Rota: GET /api/v1/coupon/:id**

**Rota: GET /api/v1/coupon?q=\$coupon**

#### Cenário A:

O usuário tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: GET /api/v1/coupon**

#### Cenário A:

O usuário não tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

**Rota: GET /api/v1/coupon**

**Rota: GET /api/v1/coupon?q=\$coupon**

**Cenário A:**

O usuário não tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: POST /api/v1/coupon**

**Cenário A:**

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 403, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário C:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de um novo cupom.

### Cenário D:

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

### Cenário E:

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

### Conclusão dos testes:

```
/api/v1/coupon
User isn't authenticated
  ✓ GET /api/v1/coupon - Should respond with status 403 and send body with details and hints (6 ms)
User is admin
  ✓ GET /api/v1/coupon - Should respond with status 200 and send body with data array (12 ms)
  ✓ GET /api/v1/coupon/:id - Should respond with status 200 and send body with data array (7 ms)
  ✓ POST /api/v1/coupon - Without body - Should respond with status 400 and send body with details array (8 ms)
  ✓ POST /api/v1/coupon - With body - Should respond with status 201 (12 ms)
User is not admin
  ✓ GET /api/v1/coupon?q=BEMVINDOS - Should respond with status 200 and send body with data array (9 ms)
  ✓ GET /api/v1/coupon - Should respond with status 401 and send body with details and hints (7 ms)
  ✓ GET /api/v1/coupon/:id - Should respond with status 200 and send body with data array (7 ms)
  ✓ POST /api/v1/coupon - Without body - Should respond with status 401 and send body with details array (6 ms)
  ✓ POST /api/v1/coupon - With body - Should respond with status 401 (6 ms)
```

## /api/v1/cupcake

Neste teste, foram avaliadas as rotas de retorno e criação e novos cupcakes. Os critérios de aceitação para este teste são:

**Rota: GET /api/v1/cupcake**

**Rota: GET /api/v1/cupcake?q=\$query**

**Rota: GET /api/v1/cupcake?category=\$category\_id**

**Rota: GET /api/v1/cupcake?dough=\$dough\_id**

**Rota: GET /api/v1/cupcake?filling=\$filling\_id**

**Rota: GET /api/v1/cupcake?cover=\$cover\_id**

**Rota: GET /api/v1/cupcake?decoration=\$decoration\_id**

**Rota: GET /api/v1/cupcake/:id**

**Rota: GET /api/v1/cupcake/:id/review**

**Cenário A:**

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: POST /api/v1/cupcake**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de um novo cupcake.

**Cenário C:**

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

**Cenário D:**

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

**Rota: POST /api/v1/cupcake/:id/review**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de uma nova avaliação.

### Cenário C:

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

### Cenário D:

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de uma nova avaliação.

### Conclusão dos testes:

```
/api/v1/cupcake
User isn't authenticated
  ✓ GET /api/v1/cupcake - Should respond with status 200 and send body with data array (62 ms)
  ✓ GET /api/v1/cupcake?q=$query - Should respond with status 200 and send body with data array (48 ms)
  ✓ GET /api/v1/cupcake?category=$category_id - Should respond with status 200 and send body with data array (29 ms)
  ✓ GET /api/v1/cupcake?dough=$dough_id - Should respond with status 200 and send body with data array (6 ms)
  ✓ GET /api/v1/cupcake?fillings=$filling_id - Should respond with status 200 and send body with data array (33 ms)
  ✓ GET /api/v1/cupcake?cover=$cover_id - Should respond with status 200 and send body with data array (6 ms)
  ✓ GET /api/v1/cupcake?decoration=$decoration_id - Should respond with status 200 and send body with data array (7 ms)
  ✓ GET /api/v1/cupcake/:id - Should respond with status 200 and send body with data array (7 ms)
  ✓ GET /api/v1/cupcake/:id/review - Should respond with status 200 and send body with data array (7 ms)
User is admin
  ✓ POST /api/v1/cupcake - Without body - Should respond with status 400 and send body with details array (5 ms)
  ✓ POST /api/v1/cupcake - With body - Should respond with status 201 (9 ms)
  ✓ POST /api/v1/cupcake/:id/review - Without body - Should respond with status 401 and send body with details array (6 ms)
  ✓ POST /api/v1/cupcake/:id/review - With body - Should respond with status 201 (13 ms)
User is not admin
  ✓ POST /api/v1/cupcake - Without body - Should respond with status 401 and send body with details array (8 ms)
  ✓ POST /api/v1/cupcake - With body - Should respond with status 401 (7 ms)
  ✓ POST /api/v1/cupcake/:id/review - Without body - Should respond with status 401 and send body with details array (6 ms)
  ✓ POST /api/v1/cupcake/:id/review - With body - Should respond with status 201 (9 ms)
```

## /api/v1/ingredient

Neste teste, foram avaliadas as rotas de retorno e criação e novos ingredientes. Os critérios de aceitação para este teste são:

**Rota: GET /api/v1/ingredient**

**Rota: GET /api/v1/ingredient /:id**

**Cenário A:**

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: POST /api/v1/ ingredient**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de um novo ingrediente.

**Cenário C:**

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve

conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

#### Cenário D:

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

#### Conclusão dos testes:

```
/api/v1/ingredient
User isn't authenticated
  ✓ GET /api/v1/ingredient - Should respond with status 200 and send body with data array (7 ms)
  ✓ GET /api/v1/ingredient/:id - Should respond with status 200 and send body with data array (6 ms)
User is admin
  ✓ POST /api/v1/ingredient - Without body - Should respond with status 400 and send body with details array (6 ms)
  ✓ POST /api/v1/ingredient - With body - Should respond with status 201 (11 ms)
User is not admin
  ✓ POST /api/v1/ingredient - Without body - Should respond with status 401 and send body with details array (6 ms)
  ✓ POST /api/v1/ingredient - With body - Should respond with status 400 (8 ms)
```

## /api/v1/orders

Neste teste, foram avaliadas as rotas de retorno e criação e novos pedidos. Os critérios de aceitação para este teste são:

**Rota: GET /api/v1/orders**

**Rota: GET /api/v1/orders/current**

**Rota: GET /api/v1/orders/:id**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Cenário B:**

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 403, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Rota: GET /api/v1/orders**

**Rota: GET /api/v1/orders/:id**

**Cenário A:**

O usuário não tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 401, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor. Esta rota é acessível apenas por usuários administradores.

**Rota: GET /api/v1/orders/current**

**Cenário A:**

O usuário não tem o nível de acesso: "admin". Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

**Rota: POST /api/v1/orders**

**Cenário A:**

O usuário tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

**Cenário B:**

O usuário tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de um novo pedido.

**Cenário C:**

O usuário não tem o nível de acesso: "admin". Sem o envio do corpo em JSON ou Envio do corpo em JSON ou com parâmetros incorretos ou inexistentes.

**Resultado Esperado:** Retorno com o código HTTP 400, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

### Cenário D:

O usuário não tem o nível de acesso: "admin". Envio do corpo em JSON com os campos e informações preenchidos corretamente.

**Resultado Esperado:** Retorno com o código HTTP 201 indicando a criação de um novo pedido.

### Conclusão dos testes:

```
/api/v1/orders
User isn't authenticated
  ✓ GET /api/v1/orders - Should respond with status 403 and send body with details and hints (5 ms)
User is admin
  ✓ GET /api/v1/orders - Should respond with status 200 and send body with data array (19 ms)
  ✓ GET /api/v1/orders/current - Should respond with status 200 and send body with data array (12 ms)
  ✓ GET /api/v1/orders/:id - Should respond with status 200 and send body with data array (7 ms)
  ✓ POST /api/v1/orders - Without body - Should respond with status 400 and send body with details array (9 ms)
  ✓ POST /api/v1/orders - With body - Should respond with status 201 (12 ms)
User is not admin
  ✓ GET /api/v1/orders - Should respond with status 401 and send body with details and hints (8 ms)
  ✓ GET /api/v1/orders/current - Should respond with status 200 and send body with data array (12 ms)
  ✓ GET /api/v1/orders/:id - Should respond with status 401 and send body with details and hints (4 ms)
  ✓ POST /api/v1/orders - Without body - Should respond with status 400 and send body with details array (6 ms)
  ✓ POST /api/v1/orders - With body - Should respond with status 201 (10 ms)
```

## /api/v1/user

Neste teste, foram avaliadas as rotas de retorno. Os critérios de aceitação para este teste são:

### Rota: GET /api/v1/user/current

#### Cenário A:

Solicitação as URLs, sem um token no cabeçalho representando um usuário não logado.

**Resultado Esperado:** Retorno com o código HTTP 403, acompanhado de uma resposta em JSON no corpo, contendo o campo "details" definido. Este campo deve conter informações sobre o motivo da recusa da solicitação pelo servidor.

### Rota: GET /api/v1/user/current

### Rota: GET /api/v1/user/current/address

### Rota: GET /api/v1/user/cep/:cep

#### Cenário A:

Solicitação as URLs com um usuário autenticado.

**Resultado Esperado:** Retorno com o código HTTP 200, acompanhado de uma resposta em JSON no corpo, contendo o campo "data" definido. Este campo deve conter informações solicitadas pelo usuário.

### Conclusão dos testes:

```
/api/v1/user
User isn't authenticated
  ✓ GET /api/v1/user/current - Should respond with status 403 and send body with data array (5 ms)
User is authenticated
  ✓ GET /api/v1/user/current - Should respond with status 200 and send body with data object (6 ms)
  ✓ GET /api/v1/user/current/address - Should respond with status 200 and send body with data object (5 ms)
  ✓ GET /api/v1/user/cep/:cep - Should respond with status 200 and send body with data object (510 ms)
```

## Conclusão Testes Backend

```
PASS  tests/main.test.js
/api/v1/auth
  Register new user
    POST /api/v1/auth/register - Without body - Should respond with status 400 and send body with details array (80 ms)
    ✓ POST /api/v1/auth/register - With body - Should respond with status 201 (197 ms)
  Logs login
    ✓ POST /api/v1/auth/login - Without body - Should respond with status 400 and send body with details array (8 ms)
    ✓ POST /api/v1/auth/login - With body - Should respond with status 200 and a accessToken cookie (88 ms)
/api/v1/category
  User isn't authenticated
    ✓ GET /api/v1/category - Should respond with status 200 and send body with data array (9 ms)
    ✓ GET /api/v1/category/:id - Should respond with status 200 and send body with data array (8 ms)
  User is admin
    ✓ POST /api/v1/category - Without body - Should respond with status 400 and send body with details array (9 ms)
    ✓ POST /api/v1/category - With body - Should respond with status 201 (10 ms)
  User is not admin
    ✓ POST /api/v1/category - Without body - Should respond with status 401 and send body with hints (7 ms)
    ✓ POST /api/v1/category - With body - Should respond with status 401 and send body with details and hints (7 ms)
/api/v1/coupon
  User isn't authenticated
    ✓ GET /api/v1/coupon - Should respond with status 403 and send body with details and hints (6 ms)
  User is admin
    ✓ GET /api/v1/coupon - Should respond with status 200 and send body with data array (12 ms)
    ✓ GET /api/v1/coupon/:id - Should respond with status 200 and send body with data array (7 ms)
    ✓ POST /api/v1/coupon - Without body - Should respond with status 400 and send body with details array (8 ms)
    ✓ POST /api/v1/coupon - With body - Should respond with status 201 (12 ms)
  User is not admin
    ✓ GET /api/v1/coupon?q=BEMVINDOS - Should respond with status 200 and send body with data array (9 ms)
    ✓ GET /api/v1/coupon - Should respond with status 401 and send body with details and hints (7 ms)
    ✓ GET /api/v1/coupon/:id - Should respond with status 200 and send body with data array (7 ms)
    ✓ POST /api/v1/coupon - Without body - Should respond with status 401 and send body with details array (6 ms)
    POST /api/v1/coupon - With body - Should respond with status 401 (6 ms)
/api/v1/cupcake
  User isn't authenticated
    ✓ GET /api/v1/cupcake - Should respond with status 200 and send body with data array (62 ms)
    ✓ GET /api/v1/cupcake?q=$query - Should respond with status 200 and send body with data array (48 ms)
    ✓ GET /api/v1/cupcake?category=$category_id - Should respond with status 200 and send body with data array (29 ms)
    ✓ GET /api/v1/cupcake?dough=$dough_id - Should respond with status 200 and send body with data array (6 ms)
    ✓ GET /api/v1/cupcake?filling=$filling_id - Should respond with status 200 and send body with data array (33 ms)
    ✓ GET /api/v1/cupcake?cover=$cover_id - Should respond with status 200 and send body with data array (6 ms)
    ✓ GET /api/v1/cupcake?decoration=$decoration_id - Should respond with status 200 and send body with data array (7 ms)
    ✓ GET /api/v1/cupcake/:id - Should respond with status 200 and send body with data array (7 ms)
    ✓ GET /api/v1/cupcake/:id/review - Should respond with status 200 and send body with data array (7 ms)
  User is admin
    ✓ POST /api/v1/cupcake - Without body - Should respond with status 400 and send body with details array (5 ms)
    ✓ POST /api/v1/cupcake - With body - Should respond with status 201 (9 ms)
    ✓ POST /api/v1/cupcake/:id/review - Without body - Should respond with status 401 and send body with details array (6 ms)
    ✓ POST /api/v1/cupcake/:id/review - With body - Should respond with status 201 (13 ms)
  User is not admin
    ✓ POST /api/v1/cupcake - Without body - Should respond with status 401 and send body with details array (8 ms)
    ✓ POST /api/v1/cupcake - With body - Should respond with status 401 (7 ms)
    ✓ POST /api/v1/cupcake/:id/review - Without body - Should respond with status 401 and send body with details array (6 ms)
    ✓ POST /api/v1/cupcake/:id/review - With body - Should respond with status 201 (9 ms)
/api/v1/ingredient
  User isn't authenticated
    ✓ GET /api/v1/ingredient - Should respond with status 200 and send body with data array (7 ms)
    ✓ GET /api/v1/ingredient/:id - Should respond with status 200 and send body with data array (6 ms)
  User is admin
    ✓ POST /api/v1/ingredient - Without body - Should respond with status 400 and send body with details array (6 ms)
    ✓ POST /api/v1/ingredient - With body - Should respond with status 201 (11 ms)
  User is not admin
    ✓ POST /api/v1/ingredient - Without body - Should respond with status 401 and send body with details array (6 ms)
    ✓ POST /api/v1/ingredient - With body - Should respond with status 400 (8 ms)
/api/v1/orders
  User isn't authenticated
    ✓ GET /api/v1/orders - Should respond with status 403 and send body with details and hints (5 ms)
  User is admin
    ✓ GET /api/v1/orders - Should respond with status 200 and send body with data array (19 ms)
    ✓ GET /api/v1/orders/current - Should respond with status 200 and send body with data array (12 ms)
    ✓ GET /api/v1/orders/:id - Should respond with status 200 and send body with data array (7 ms)
    ✓ POST /api/v1/orders - Without body - Should respond with status 400 and send body with details array (9 ms)
    ✓ POST /api/v1/orders - With body - Should respond with status 201 (12 ms)
  User is not admin
    ✓ GET /api/v1/orders - Should respond with status 401 and send body with details and hints (8 ms)
    ✓ GET /api/v1/orders/current - Should respond with status 200 and send body with data array (12 ms)
    ✓ GET /api/v1/orders/:id - Should respond with status 401 and send body with details and hints (4 ms)
    ✓ POST /api/v1/orders - Without body - Should respond with status 400 and send body with details array (6 ms)
    ✓ POST /api/v1/orders - With body - Should respond with status 201 (10 ms)
/api/v1/user
  User isn't authenticated
    ✓ GET /api/v1/user/current - Should respond with status 403 and send body with data array (5 ms)
  User is authenticated
    ✓ GET /api/v1/user/current - Should respond with status 200 and send body with data object (6 ms)
    ✓ GET /api/v1/user/current/address - Should respond with status 200 and send body with data object (5 ms)
    ✓ GET /api/v1/user/cep - Should respond with status 200 and send body with data object (510 ms)

Test Suites: 1 passed, 1 total
Tests:      58 passed, 58 total
Snapshots:  0 total
Time:       3.08 s, estimated 19 s
Ran all test suites.
```

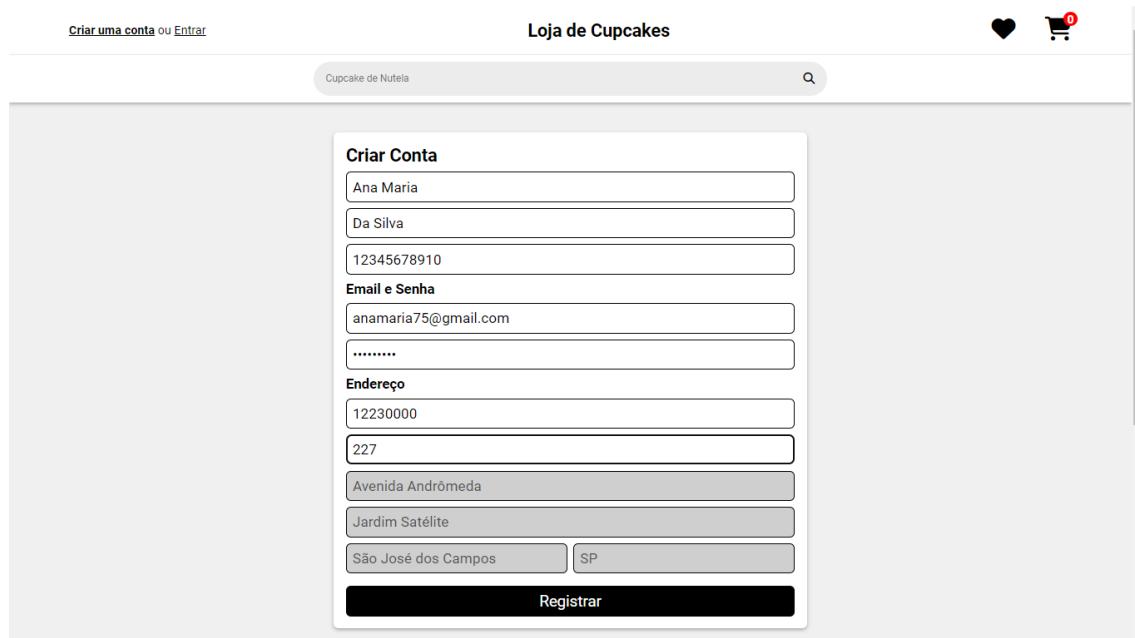
58 testes executados, 100% de sucesso.

## Frontend

### Registro de conta

Script:

1. Acessar tela de criação de conta;
2. Preencher campo de nome;
3. Preencher campo de sobrenome;
4. Preencher campo de email;
5. Preencher campo de senha;
6. Preencher campo de cep;
7. Preencher campo de número da casa;
8. Clicar no botão em “registrar”;



Criar uma conta ou Entrar

Loja de Cupcakes

Cupcake de Nutela

**Criar Conta**

Ana Maria

Da Silva

12345678910

**Email e Senha**

anamaria75@gmail.com

.....

**Endereço**

12230000

227

Avenida Andrômeda

Jardim Satélite

São José dos Campos SP

**Registrar**

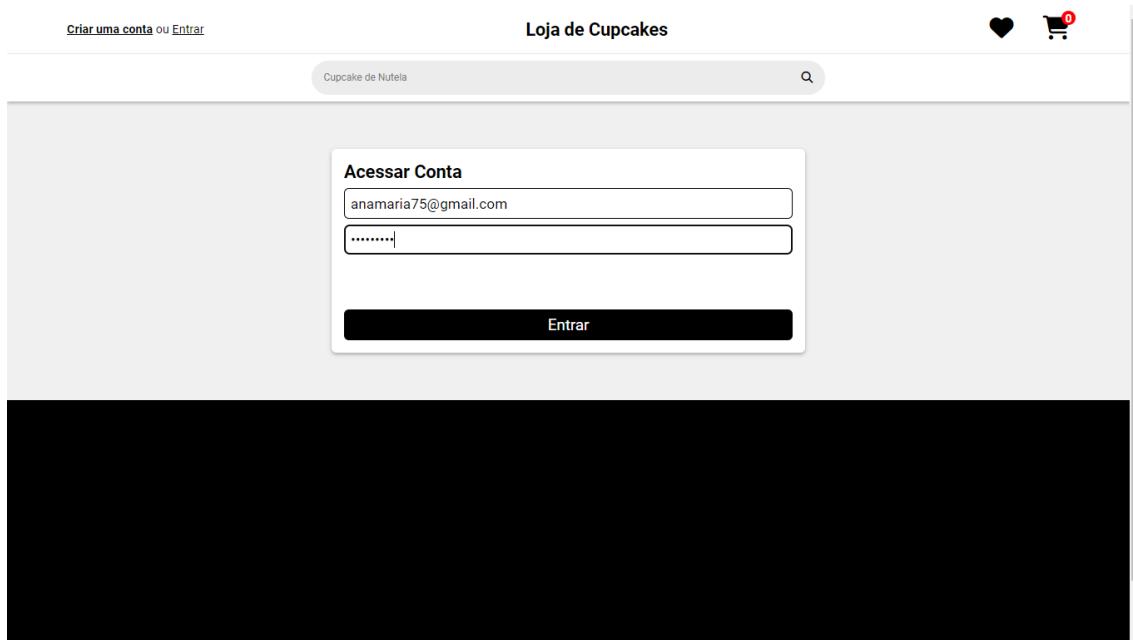
### Resultado:

Sucesso, novo registro de usuário concluído.

## Acesso da conta

Script:

1. Acessar tela de login;
2. Preencher campo de email;
3. Preencher campo de senha;
4. Clicar no botão “entrar”;



## Resultado:

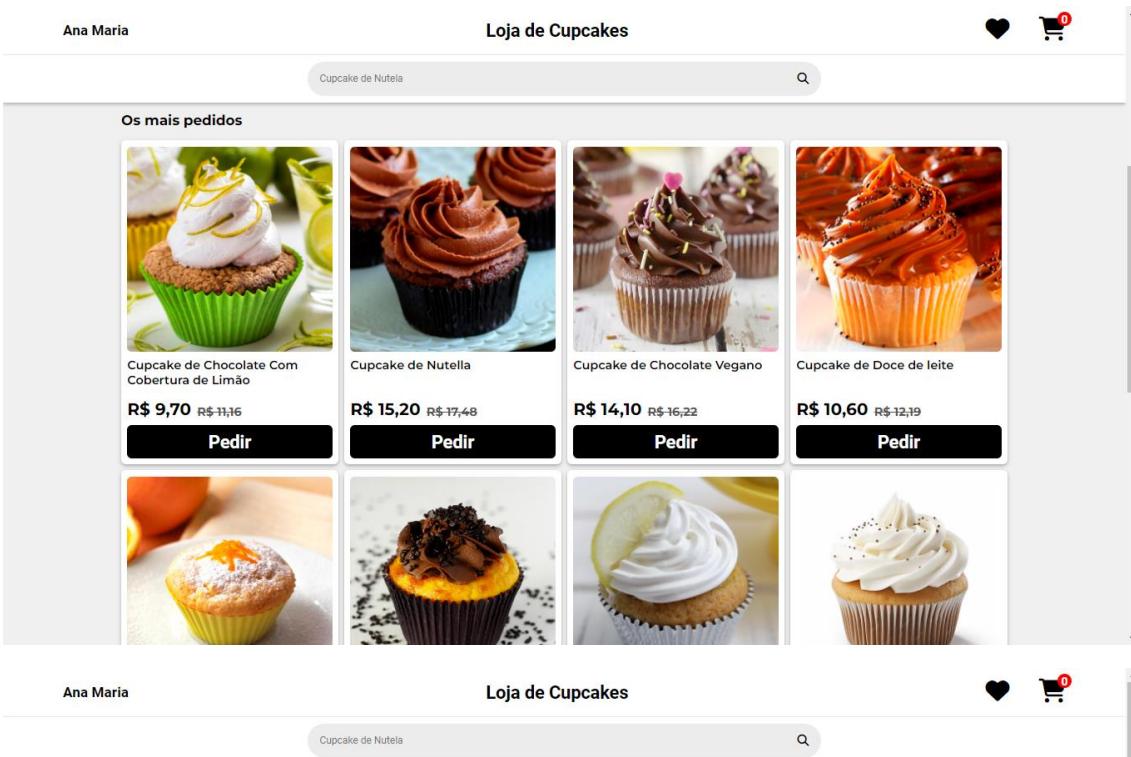
Sucesso, acesso do usuário concluído.

## Tela “Home” – Acessar pagina de cupcake

Script:

1. Acessar tela de “home”;
2. Localizar um cupcake;
3. Clicar no botão “pedir”;

Desktop:



Ana Maria Loja de Cupcakes

Cupcake de Nutella

Os mais pedidos

Cupcake de Chocolate Com Cobertura de Limão	Cupcake de Nutella	Cupcake de Chocolate Vegano	Cupcake de Doce de leite
R\$ 9,70 R\$ 11,16	R\$ 15,20 R\$ 17,48	R\$ 14,10 R\$ 16,22	R\$ 10,60 R\$ 12,19
Pedir	Pedir	Pedir	Pedir

Ana Maria Loja de Cupcakes

Cupcake de Nutella



Cupcake de Chocolate Com Cobertura de Limão

Massa: Massa de Chocolate  
Cobertura: Limão  
Recheio: Limão  
Decoração: Raspas de Limão

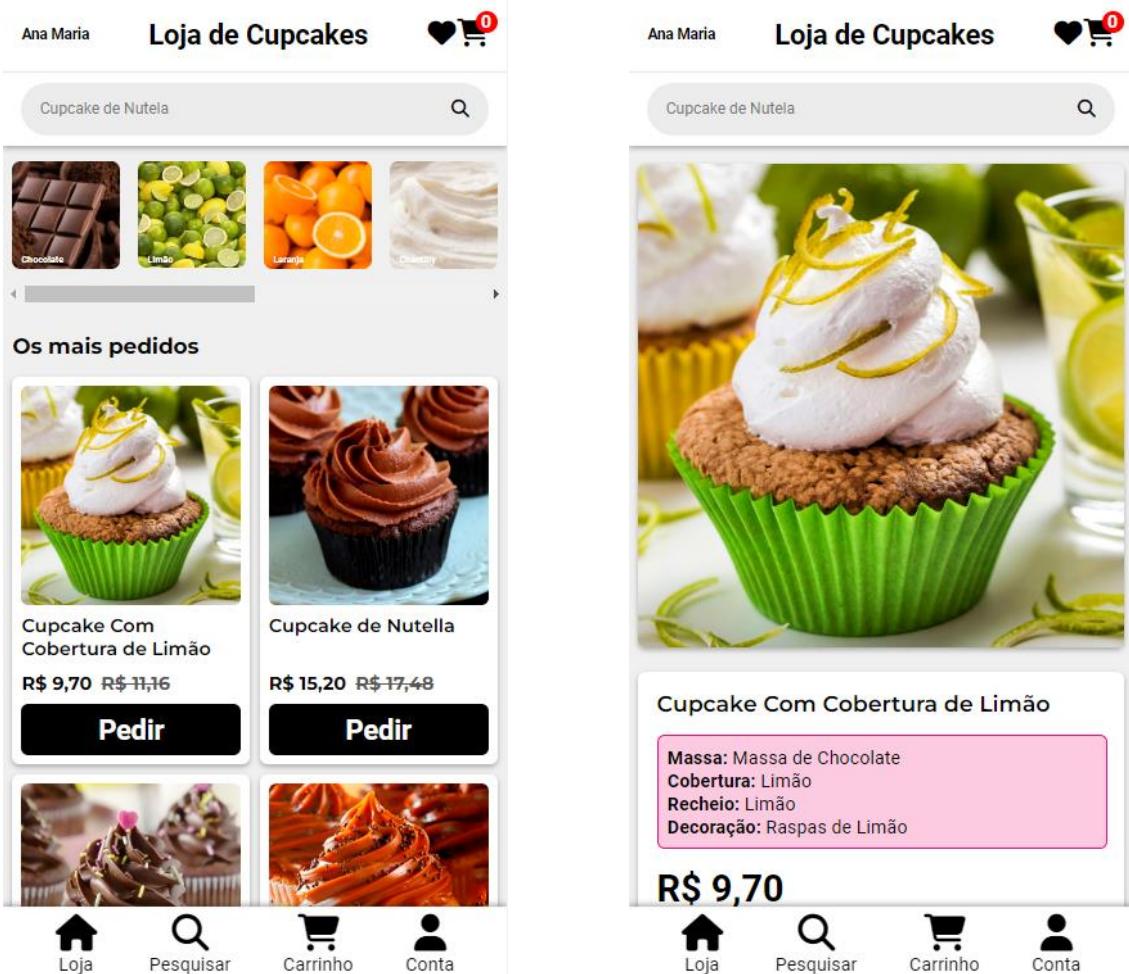
R\$ 9,70

Adicionar ao carrinho

- 1 +

Descrição

**Mobile:**



Ana Maria      Loja de Cupcakes      ❤️ 0

Cupcake de Nutella

Chocolate    Limão    Laranja    Chocolate

Os mais pedidos

Cupcake Com Cobertura de Limão      R\$ 9,70 R\$ 11,16

Pedir

Cupcake de Nutella      R\$ 15,20 R\$ 17,48

Pedir

**Cupcake Com Cobertura de Limão**

**Massa:** Massa de Chocolate  
**Cobertura:** Limão  
**Recheio:** Limão  
**Decoração:** Raspas de Limão

R\$ 9,70

Loja      Pesquisar      Carrinho      Conta

Loja      Pesquisar      Carrinho      Conta

**Resultado:**

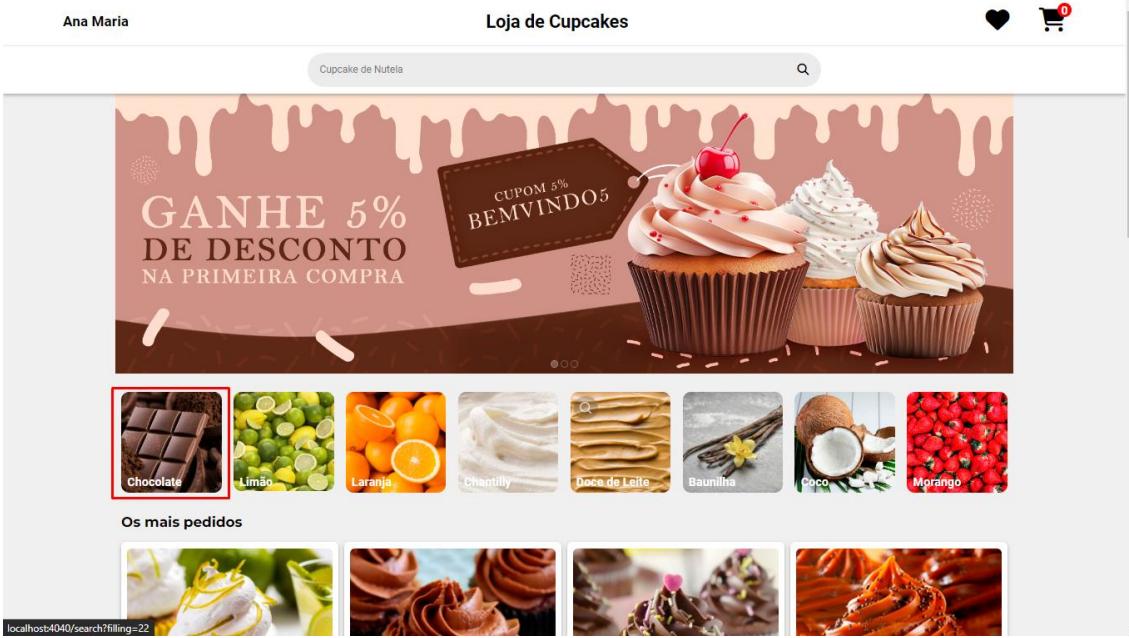
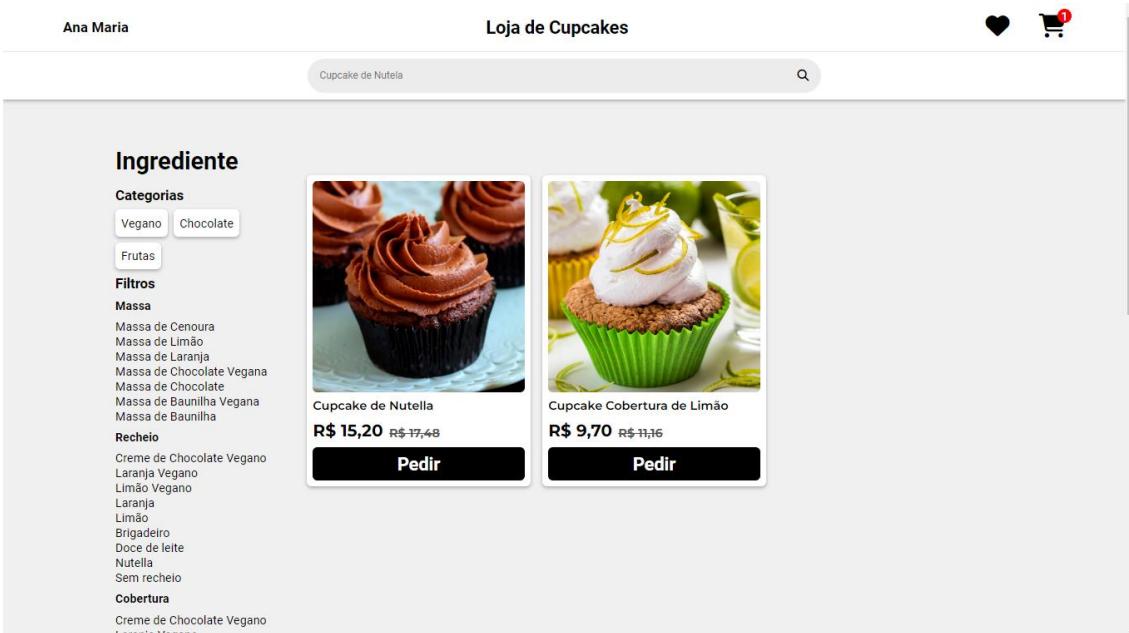
Sucesso, acesso a página do cupcake concluída.

## Tela “Home” – Acessar pagina lista por ingrediente;

Script:

1. Acessar tela de “home”;
2. Localizar um ingrediente;
3. Clicar no ingrediente;

### Desktop:

### Mobile:

Ana Maria      Loja de Cupcakes       

Cupcake de Nutela 



  
Chocolate

  
Limão

  
Laranja

  
Cheesecake

Os mais pedidos

  
Cupcake Com Cobertura de Limão  
R\$ 9,70 R\$ 11,16 

  
Cupcake de Nutella  
R\$ 15,20 R\$ 17,48 

 Loja     Pesquisar     Carrinho     Conta

Ana Maria      Loja de Cupcakes       

Cupcake de Nutela 

**Ingrediente**

Categorias

  
Cupcake de Nutella  
R\$ 15,20 R\$ 17,48 

  
Cupcake Com Cobertura de Limão  
R\$ 9,70 R\$ 11,16 

 Loja     Pesquisar     Carrinho     Conta

### Resultado:

Sucesso, acesso a página de lista por ingrediente.

## Adicionar produto ao carrinho

Script:

1. Acessar tela de cupcake;
2. Alterar quantidade;
3. Clicar no botão “Adicionar ao carrinho”;

### Desktop:



Ana Maria

Loja de Cupcakes

Cupcake de Nutella

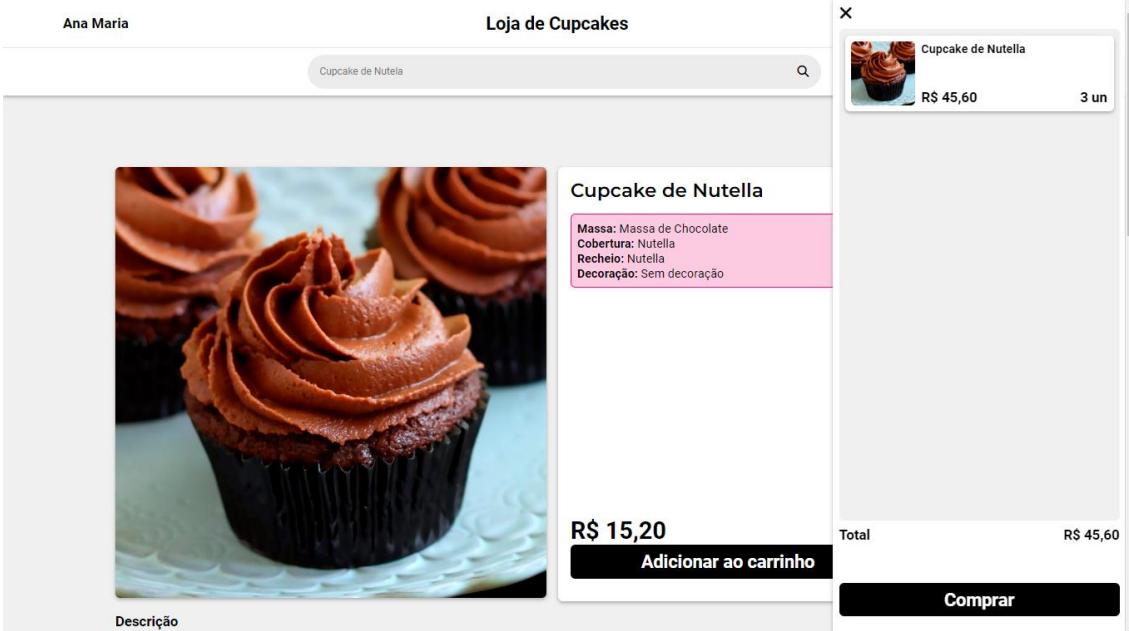
Massa: Massa de Chocolate  
Cobertura: Nutella  
Recheio: Nutella  
Decoração: Sem decoração

R\$ 15,20

Adicionar ao carrinho

- 3 +

Descrição



Ana Maria

Loja de Cupcakes

Cupcake de Nutella

Massa: Massa de Chocolate  
Cobertura: Nutella  
Recheio: Nutella  
Decoração: Sem decoração

R\$ 15,20

Adicionar ao carrinho

Total R\$ 45,60

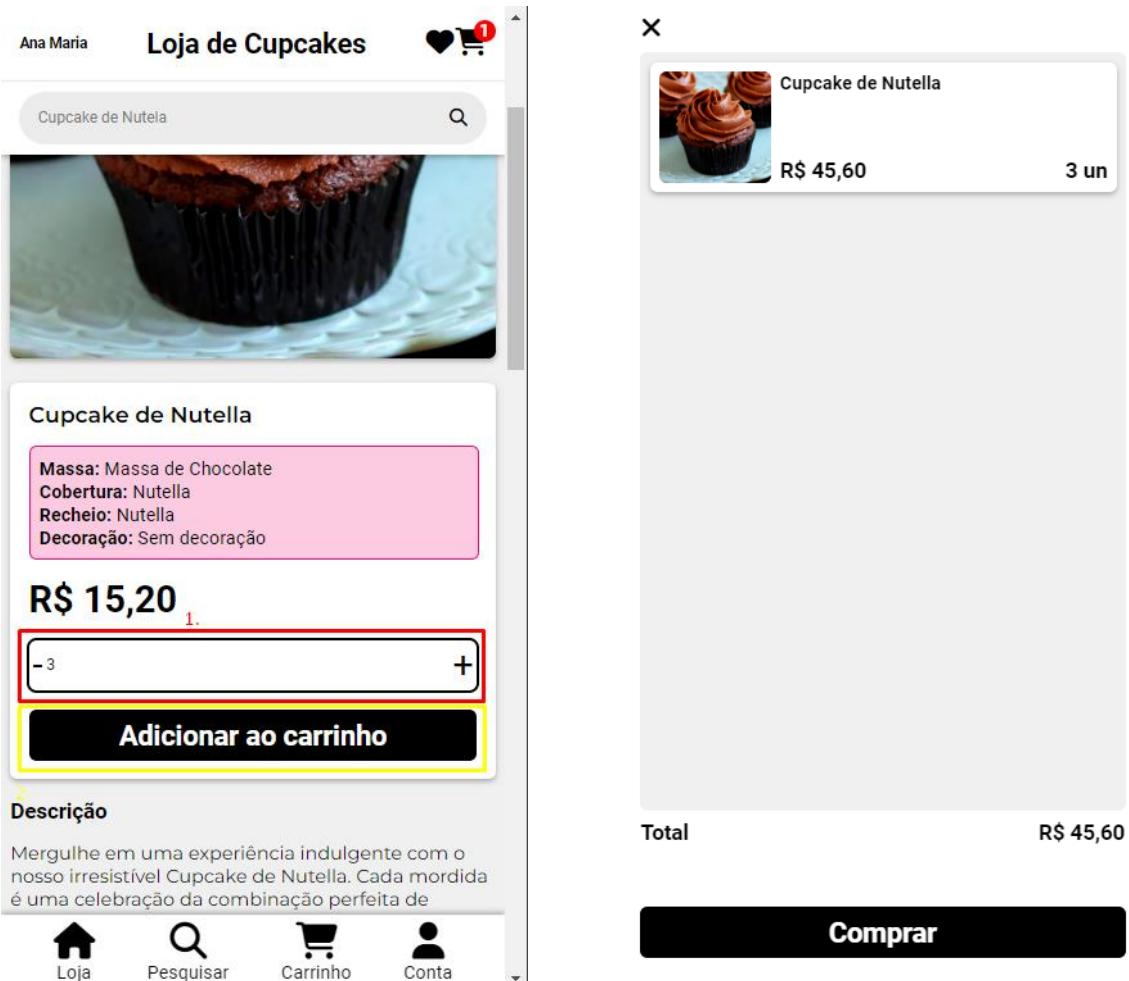
Comprar

Cupcake de Nutella

R\$ 45,60

3 un

Descrição

**Mobile:**

The image shows a mobile application interface for a virtual store. On the left, the main screen displays a product page for a "Cupcake de Nutella". The page includes a large image of the cupcake, its name, a price of R\$ 15,20, a quantity selector set to 1, and a prominent "Adicionar ao carrinho" button. Below the product details is a description: "Mergulhe em uma experiência indulgente com o nosso irresistível Cupcake de Nutella. Cada mordida é uma celebração da combinação perfeita de". At the bottom of the screen are navigation icons for Loja, Pesquisar, Carrinho, and Conta. On the right, a modal window shows a summary of the item added to the cart: "Cupcake de Nutella" at R\$ 45,60 for 3 units, with a "Comprar" button at the bottom.

**Resultado:**

Sucesso, produto e quantidade adicionados ao carrinho.

## Acesso a tela de Carrinho

Script:

1. Abrir a bandeira de carrinho ao passar no mouse no ícone de carrinho;
2. Clicar no botão “Comprar”;



The screenshot shows a user profile "Ana Maria" and a search bar for "Cupcake de Nutella". A product card for "Cupcake de Nutella" is displayed, featuring an image of a chocolate cupcake with Nutella frosting, a price of R\$ 15,20, and a "Adicionar ao carrinho" button. To the right, a shopping cart sidebar shows the item added with a total of R\$ 45,60 and a "Comprar" button highlighted with a red border.

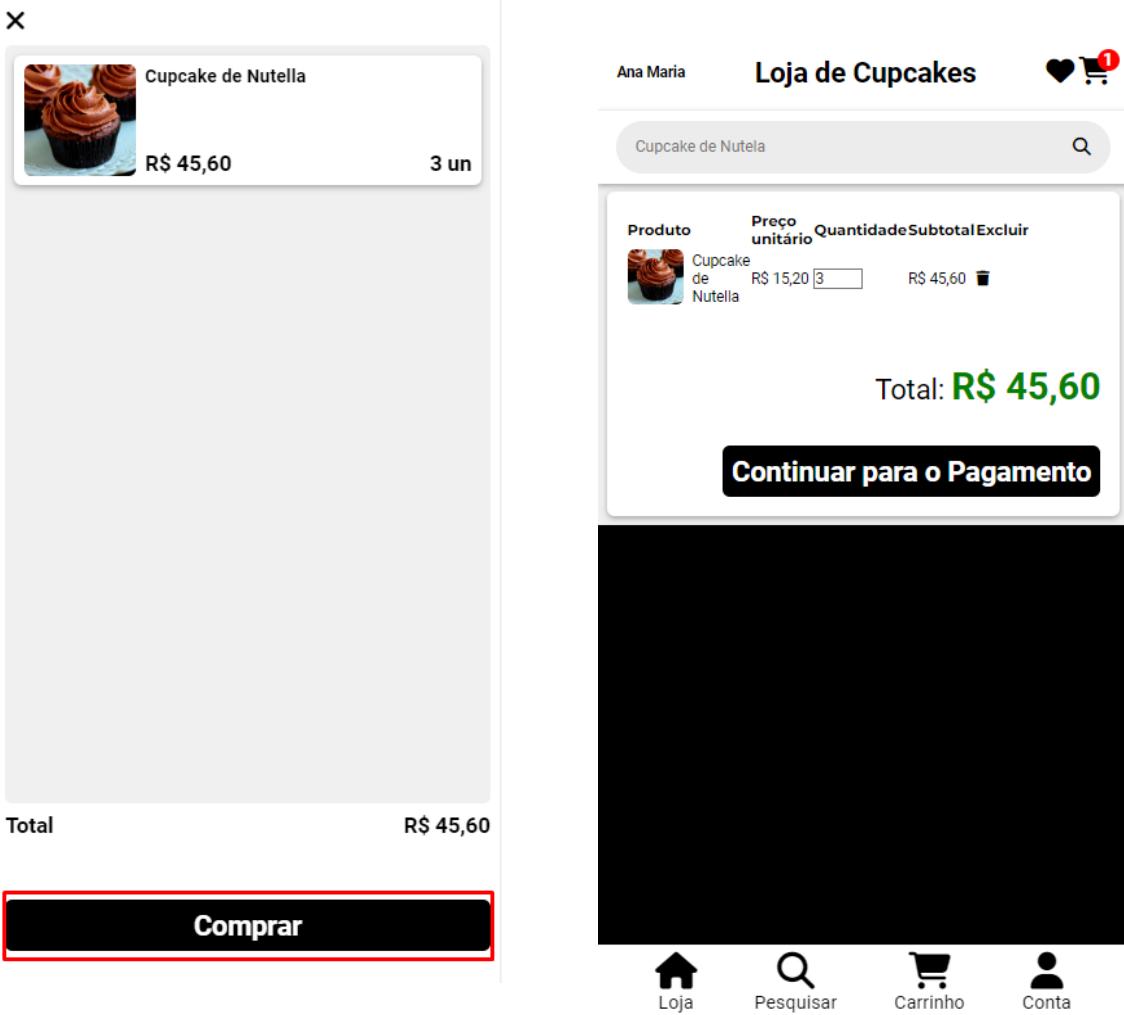

The screenshot shows the shopping cart page with a summary table:

Produto	Preço unitário	Quantidade	Subtotal	Excluir
Cupcake de Nutella	R\$ 15,20	3	R\$ 45,60	

Total: **R\$ 45,60**

**Continuar para o Pagamento**

**Mobile:**



**Loja de Cupcakes**

Ana Maria

**Cupcake de Nutella**

**Total: R\$ 45,60**

**Continuar para o Pagamento**

**Loja** **Pesquisar** **Carrinho** **Conta**

Produto	Preço unitário	Quantidade	Subtotal	Excluir
Cupcake de Nutella	R\$ 15,20	3	R\$ 45,60	

**Resultado:**

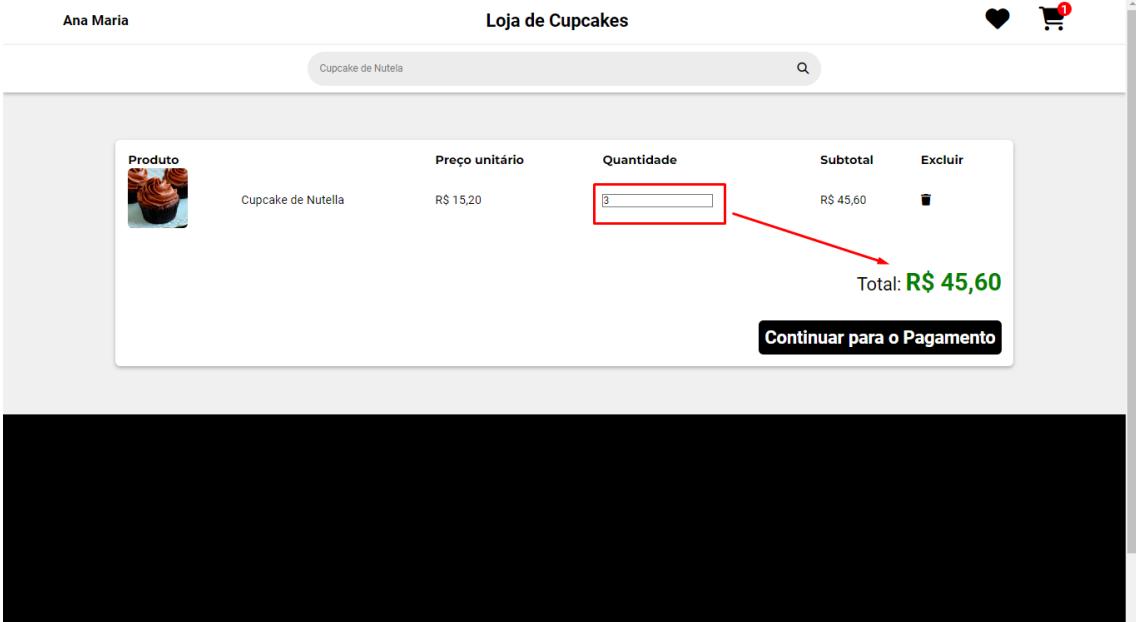
Sucesso, acesso a tela de carrinho.

## Alteração na quantidade de itens adicionados

Script:

1. Abrir a bandeira de carrinho ao passar no mouse no ícone de carrinho;
2. Acessar campo “quantidade”
3. Alterar quantidade;
4. Tirar o foco campo;

**Desktop:**



Ana Maria

Loja de Cupcakes

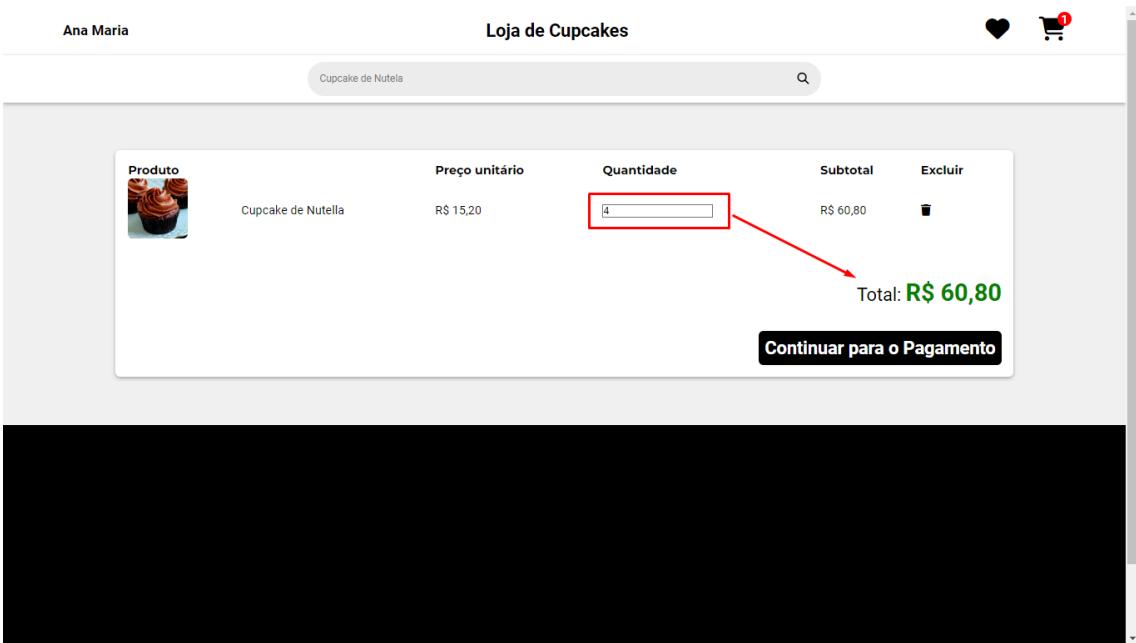
Cupcake de Nutella

Quantidade: 3

Subtotal: R\$ 45,60

Total: **R\$ 45,60**

Continuar para o Pagamento



Ana Maria

Loja de Cupcakes

Cupcake de Nutella

Quantidade: 4

Subtotal: R\$ 60,80

Total: **R\$ 60,80**

Continuar para o Pagamento

**Mobile:**



The image displays two side-by-side screenshots of a mobile application interface, likely a virtual supermarket or e-commerce platform. Both screens show a search bar at the top with the text "Cupcake de Nutella" and a magnifying glass icon. Below the search bar is a shopping cart icon with a red notification badge containing the number "1". The main content area shows a table of items in the cart.

**Left Screenshot (Initial State):**

Produto	Preço unitário	Quantidade	Subtotal	Ação
Cupcake de Nutella	R\$ 15,20	3	R\$ 45,60	

Total: **R\$ 45,60**

**Right Screenshot (After Quantity Update):**

Produto	Preço unitário	Quantidade	Subtotal	Ação
Cupcake de Nutella	R\$ 15,20	4	R\$ 60,80	

Total: **R\$ 60,80**

Both screenshots feature a large black rectangular area below the table, which is partially visible in both. At the bottom of each screenshot are four navigation icons: a house icon labeled "Loja", a magnifying glass icon labeled "Pesquisar", a shopping cart icon labeled "Carrinho", and a person icon labeled "Conta".

**Resultado:**

Sucesso, quantidade e preços atualizados.

## Remoção de um cupcake do carrinho

Script:

1. Adicionar um cupcake ao carrinho;
2. Acessar tela do carrinho;
3. Clicar no ícone de lixeira na linha do cupcake a ser removido;

### Desktop:



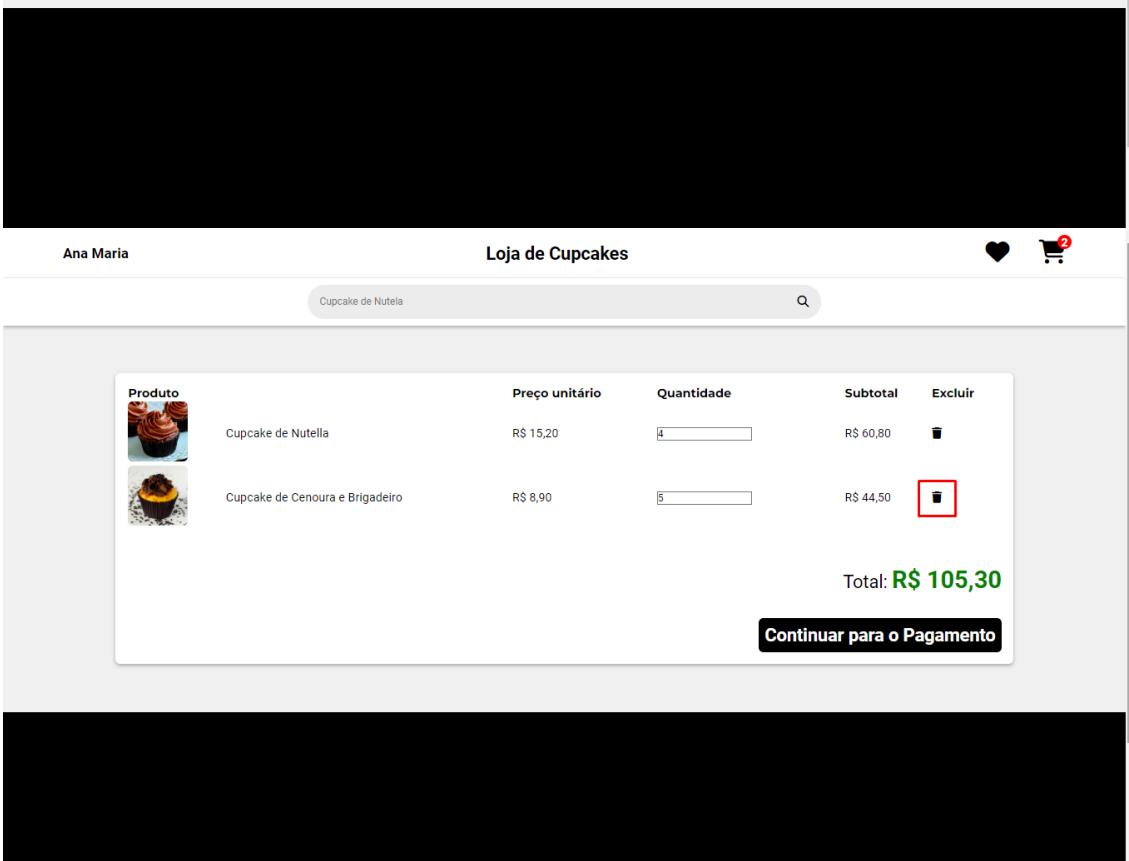
Ana Maria Loja de Cupcakes

Cupcake de Nutella

Produto	Preço unitário	Quantidade	Subtotal	Excluir
	R\$ 15,20	4	R\$ 60,80	

Total: **R\$ 60,80**

Continuar para o Pagamento

Ana Maria Loja de Cupcakes

Cupcake de Nutella

Produto	Preço unitário	Quantidade	Subtotal	Excluir
	R\$ 15,20	4	R\$ 60,80	
	R\$ 8,90	5	R\$ 44,50	

Total: **R\$ 105,30**

Continuar para o Pagamento

**Mobile:**



The image displays two side-by-side screenshots of a mobile application interface for a bakery shop named "Loja de Cupcakes".

**Screenshot 1 (Left):** Shows a search bar with "Cupcake de Nutella" and a shopping cart icon with a red notification badge containing the number 2. Below is a table of items in the cart:

Produto	Preço unitário	Quantidade	Subtotal	Ação
 Cupcake de Nutella	R\$ 15,20	4	R\$ 60,80	
 Cupcake de Cenoura e Brigadeiro	R\$ 8,90	5	R\$ 44,50	

The total amount shown is **R\$ 105,30**. A large black rectangular area covers the bottom portion of the screen.

**Screenshot 2 (Right):** Shows the same search bar and cart icon. The shopping cart now has a red notification badge with the number 1. The table shows the same items as the first screenshot. Below the table, the total amount is displayed as **Total: R\$ 60,80**. A button labeled **Continuar para o Pagamento** is visible. A large black rectangular area covers the bottom portion of the screen.

**Bottom Navigation Bar:** Both screenshots show a navigation bar with icons for Loja (home), Pesquisar (search), Carrinho (cart), and Conta (account).

**Resultado:**

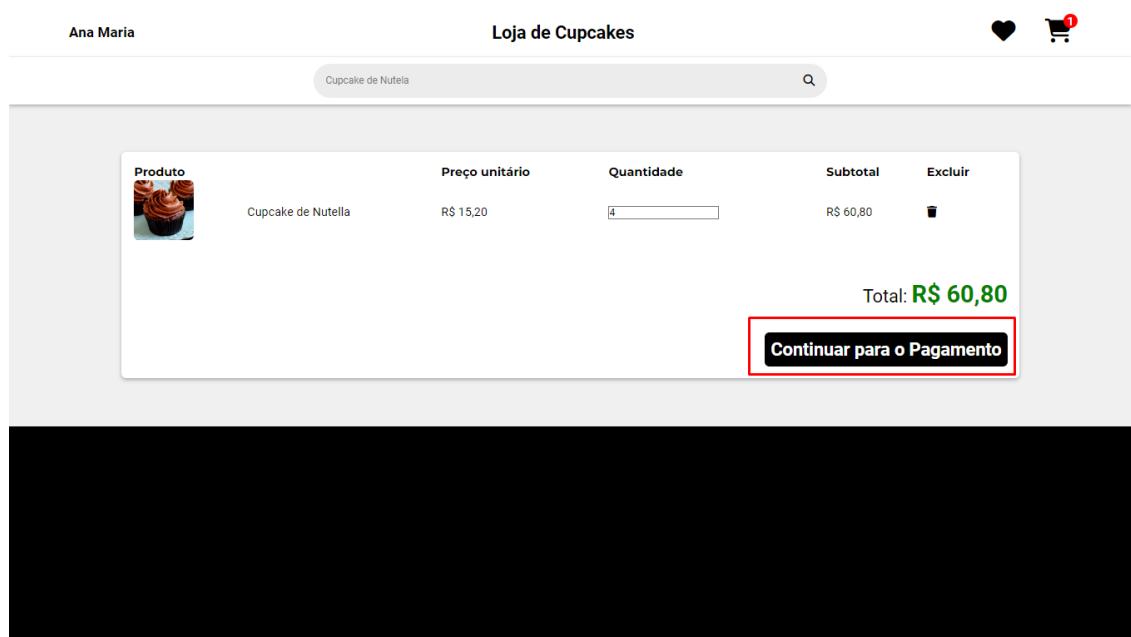
Sucesso, cupcake removido do carrinho.

## Acesso a tela de finalização de pedido

Script:

1. Acessar tela do carrinho;
2. Clicar no botão “Continuar para o Pagamento”;

### Desktop:



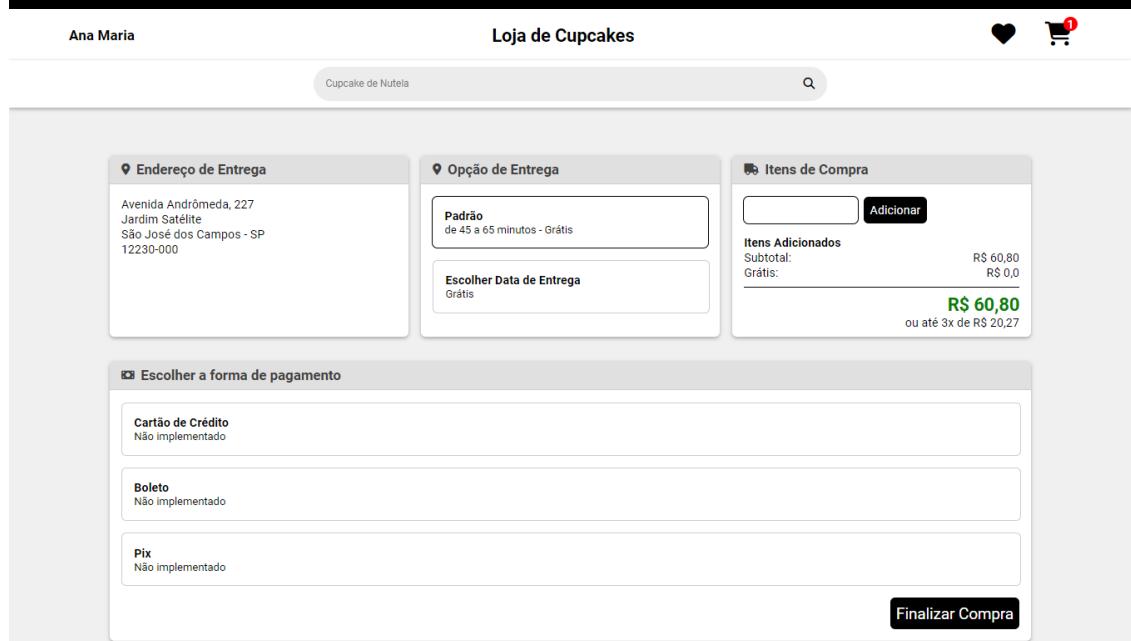
Ana Maria Loja de Cupcakes

Cupcake de Nutela

Produto	Preço unitário	Quantidade	Subtotal	Excluir
	R\$ 15,20	4	R\$ 60,80	

Total: **R\$ 60,80**

**Continuar para o Pagamento**



Ana Maria Loja de Cupcakes

Cupcake de Nutela

**Endereço de Entrega**  
Avenida Andrômeda, 227  
Jardim Satélite  
São José dos Campos - SP  
12330-000

**Opção de Entrega**  
**Padrão**  
de 45 à 65 minutos - Grátis

**Escolher Data de Entrega**  
Grátis

**Itens de Compra**  
**Adicionar**  
**Itens Adicionados**  
Subtotal: R\$ 60,80  
Gratis: R\$ 0,0  
**R\$ 60,80**  
ou até 3x de R\$ 20,27

**Escolher a forma de pagamento**

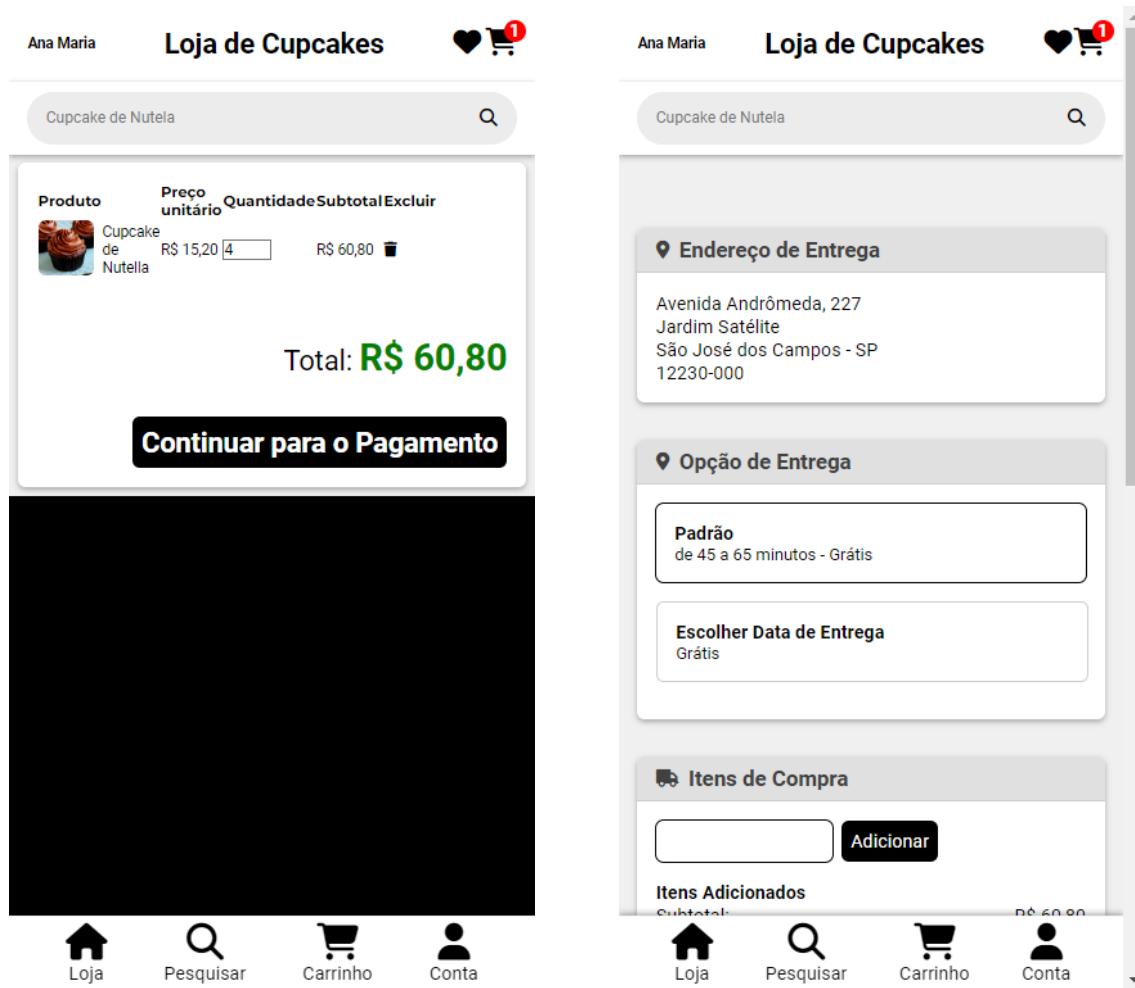
**Cartão de Crédito**  
Não implementado

**Boleto**  
Não implementado

**Pix**  
Não implementado

**Finalizar Compra**

**Mobile:**



The image displays two side-by-side screenshots of a mobile application interface for a bakery named "Loja de Cupcakes".

**Screenshot 1 (Left):** This screenshot shows a search bar at the top with the text "Cupcake de Nutella". Below it is a table showing a single item: "Cupcake de Nutella" with a price of "R\$ 15,20" and a quantity of "4", resulting in a subtotal of "R\$ 60,80". A large green button at the bottom says "Continuar para o Pagamento". At the very bottom of the screen are four navigation icons: "Loja" (home), "Pesquisar" (search), "Carrinho" (cart), and "Conta" (account).

**Screenshot 2 (Right):** This screenshot shows the same search bar with "Cupcake de Nutella". It includes a "Endereço de Entrega" section with the address "Avenida Andrômeda, 227 Jardim Satélite São José dos Campos - SP 12230-000". Below that is an "Opção de Entrega" section with "Padrão" (Standard) delivery time "de 45 a 65 minutos - Grátis" (Free). There is also a "Escolher Data de Entrega" (Choose Delivery Date) option. The "Itens de Compra" (Items of Purchase) section shows the same item and subtotal. At the bottom are the same four navigation icons: "Loja", "Pesquisar", "Carrinho", and "Conta".

**Resultado:**

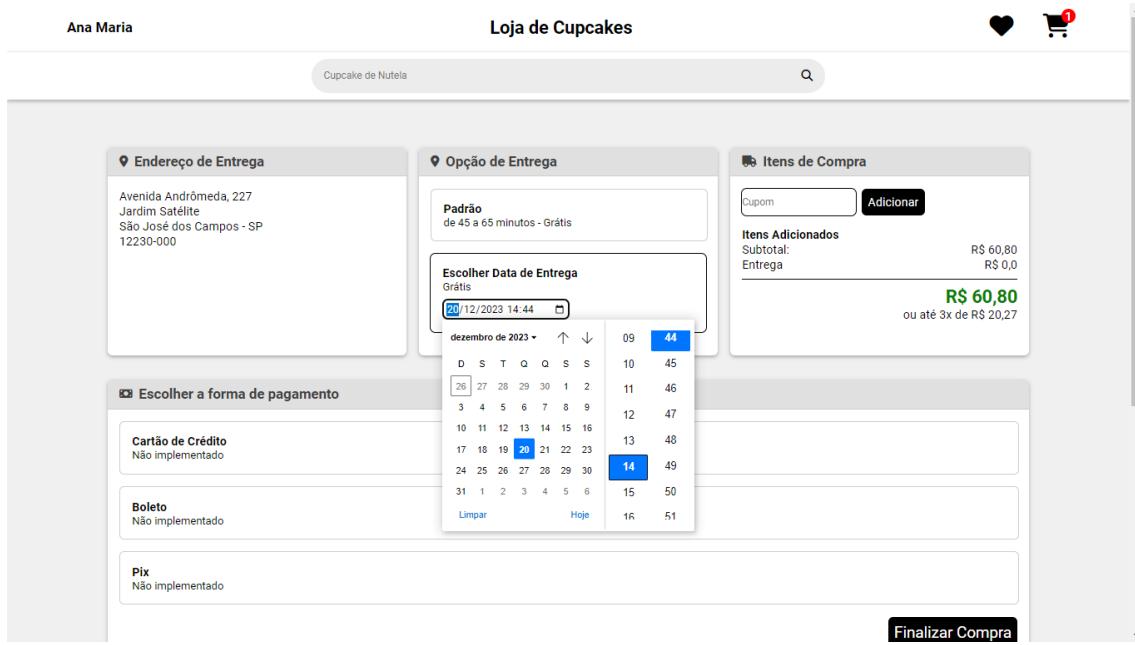
Sucesso, acesso a tela de finalização de pedido.

## Finalização do pedido

Script:

1. Acessar a tela de finalização de pedido;
2. Selecionar a opção de entrega “Escolher Data de Entrega”;
3. Selecionar uma data e hora de entrega;
4. Inserir um cupom
5. Clicar no botão “finalizar compra”;

## Desktop:



The screenshot shows a shopping cart interface for 'Loja de Cupcakes'. At the top, there's a user profile for 'Ana Maria' and a search bar with 'Cupcake de Nutela'. On the right, there are icons for a heart and a shopping cart with a '1'.

**Endereço de Entrega:**  
Avenida Andrômeda, 227  
Jardim Satélite  
São José dos Campos - SP  
12230-000

**Opção de Entrega:**  
**Padrão**  
de 45 a 65 minutos - Grátis

**Escolher Data de Entrega**  
Grátis  
09/12/2023 14:44

Sexta	Sábado	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo													
26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		

**Itens de Compra:**  
Cupom  Adicionar

**Itens Adicionados:**  
Subtotal: R\$ 60,80  
Entrega: R\$ 0,0

**Total:** R\$ 60,80  
ou até 3x de R\$ 20,27

**Escolher a forma de pagamento:**  
**Cartão de Crédito**: Não implementado  
**Boleto**: Não implementado  
**Pix**: Não implementado

**Finalizar Compra**

Ana Maria

## Loja de Cupcakes

Cupcake de Nutela

Endereço de Entrega

Avenida Andrômeda, 227  
Jardim Satélite  
São José dos Campos - SP  
12230-000

Opção de Entrega

Padrão  
de 45 a 65 minutos - Grátis

Escolher Data de Entrega  
Grátis  
20/12/2023 14:44

Itens de Compra

BEMVINDOS

Adicionar

Itens Adicionados

Subtotal: R\$ 60,80  
- R\$ 3,04  
Entrega R\$ 0,0

**R\$ 57,76**  
ou até 3x de R\$ 20,27

Escolher a forma de pagamento

Cartão de Crédito  
Não implementado

Boleto  
Não implementado

Pix  
Não implementado

Finalizar Compra

Ana Maria

## Loja de Cupcakes

Cupcake de Nutela

**Pedido Realizado**

[Ver meus pedidos](#)

## Mobile:

Ana Maria      Loja de Cupcakes       

Cupcake de Nutela 

**Opção de Entrega**

Padrão  
de 45 a 65 minutos - Grátis

**Escolher Data de Entrega**  
Grátis

20/12/2023 14:44 

dezembro de 2023  						
D	S	T	Q	Q	S	S
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

R\$ 57,76  
ou até 3x de R\$ 20,27

**Escolher a forma de pagamento**

 Loja       Pesquisar       Carrinho       Conta

Ana Maria      Loja de Cupcakes       

Cupcake de Nutela 

20/12/2023 14:44 

**Itens de Compra**

BEMVINDOS 

**Itens Adicionados**

Subtotal:	R\$ 60,80
Cupom:	- R\$ 3,04
Entrega	R\$ 0,0

**R\$ 57,76**  
ou até 3x de R\$ 20,27

**Escolher a forma de pagamento**

**Cartão de Crédito**  
Não implementado

**Boleto**  
Não Implementado

 Loja       Pesquisar       Carrinho       Conta

Ana Maria

Loja de Cupcakes



Cupcake de Nutela



Pedido Realizado

[Ver meus pedidos](#)



Loja



Pesquisar



Carrinho



Conta

Mensagem de erro caso uma data de entrega fora do horário de funcionamento ser selecionada:

Estamos fechados, nosso horário de atendimento é das 7h as 18h

**Resultado:**

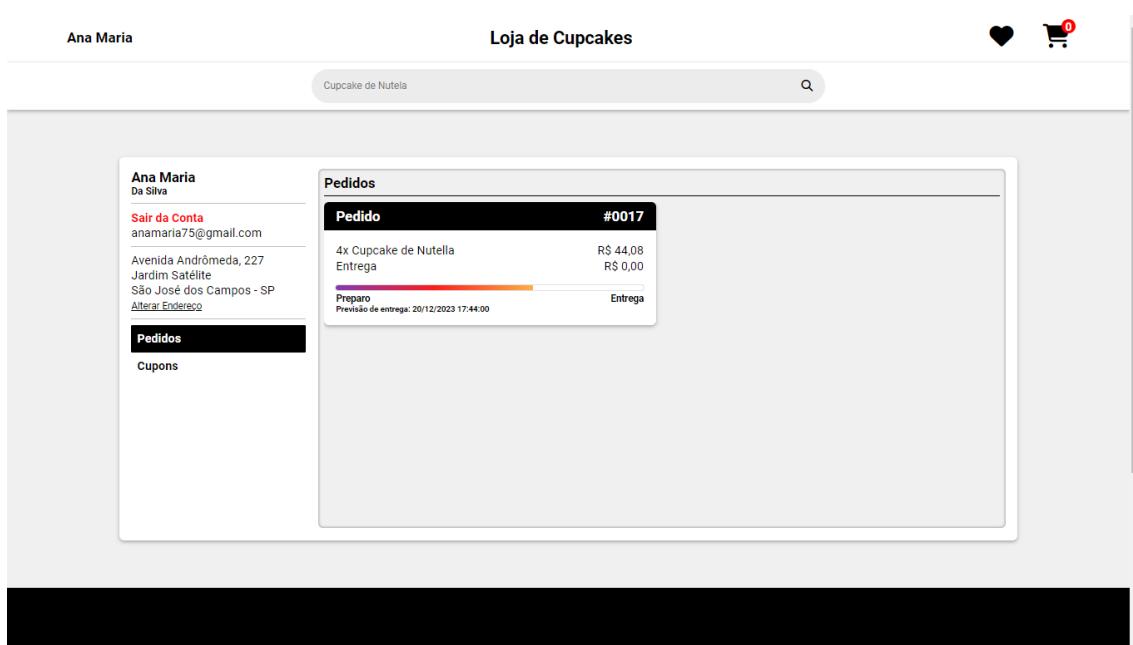
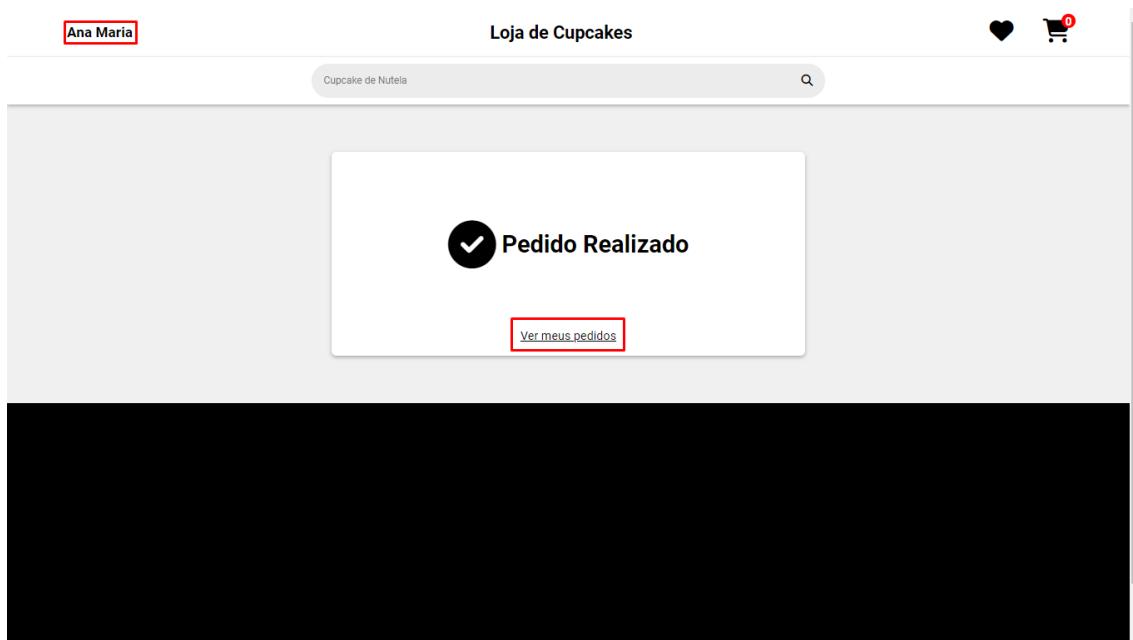
Sucesso, pedido realizado com sucesso.

## Acesso a tela de usuário

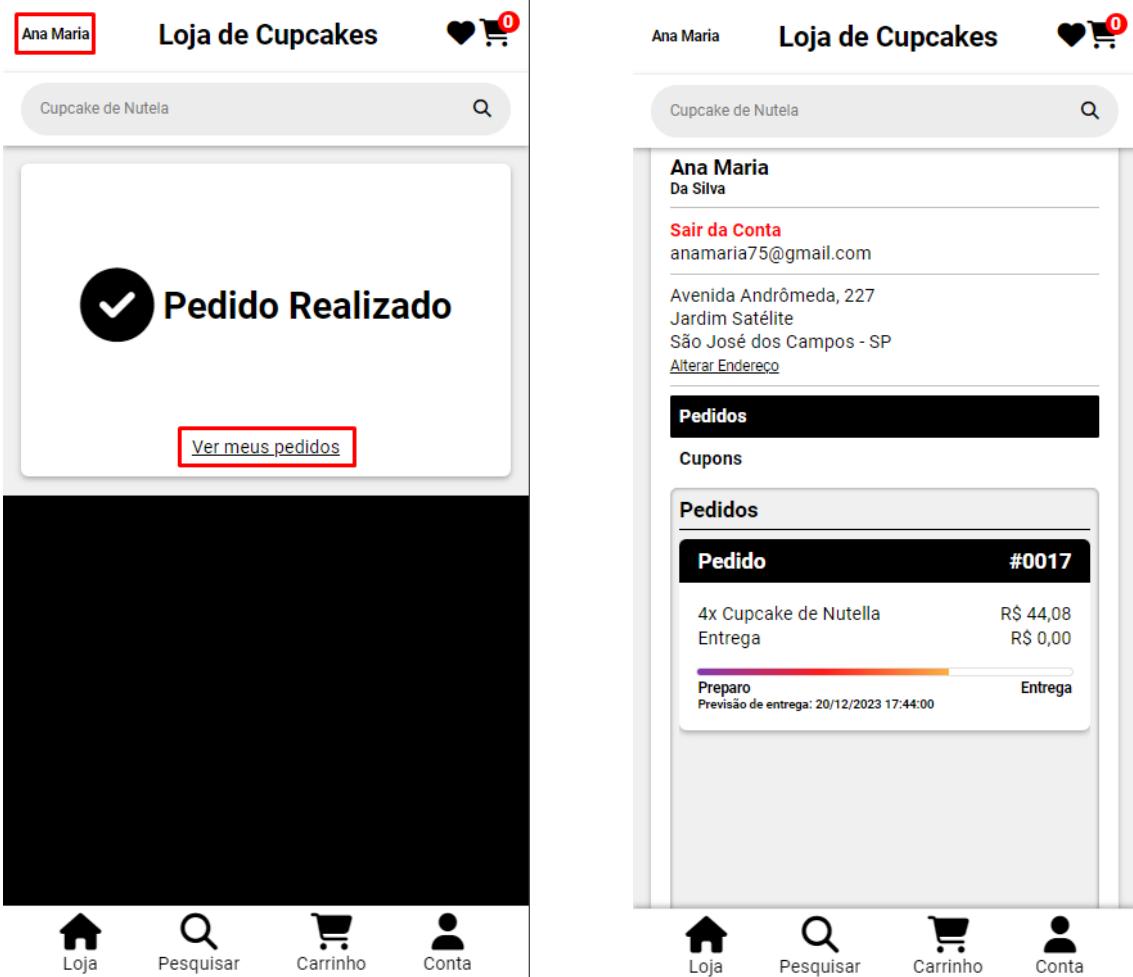
Script:

1. Clicar no nome de usuário no topo da página;
2. Visualizar pedidos;

Desktop:



**Mobile:**



The image displays two side-by-side screenshots of a mobile application interface for a bakery named "Loja de Cupcakes".

**Screenshot 1 (Left):** This screen shows a user profile for "Ana Maria" (highlighted with a red box). Below the profile is a search bar containing "Cupcake de Nutella" and a magnifying glass icon. A large central box displays a checkmark icon and the text "Pedido Realizado". At the bottom of this box is a button labeled "Ver meus pedidos" (highlighted with a red box).

**Screenshot 2 (Right):** This screen shows the same user profile for "Ana Maria" (highlighted with a red box). Below the profile is a search bar containing "Cupcake de Nutella" and a magnifying glass icon. To the right of the search bar is a "Pedidos" section. It lists a single order with the ID "#0017". The order details are: "4x Cupcake de Nutella" and "R\$ 44,08". Below the order, there is a progress bar indicating the status: "Preparo" (Preparation) is at 100% completion, and "Entrega" (Delivery) has a value of "R\$ 0,00". The progress bar also shows the estimated delivery time: "Previsão de entrega: 20/12/2023 17:44:00".

Both screenshots include a navigation bar at the bottom with icons for "Loja" (Home), "Pesquisar" (Search), "Carrinho" (Cart), and "Conta" (Account).

**Resultado:**

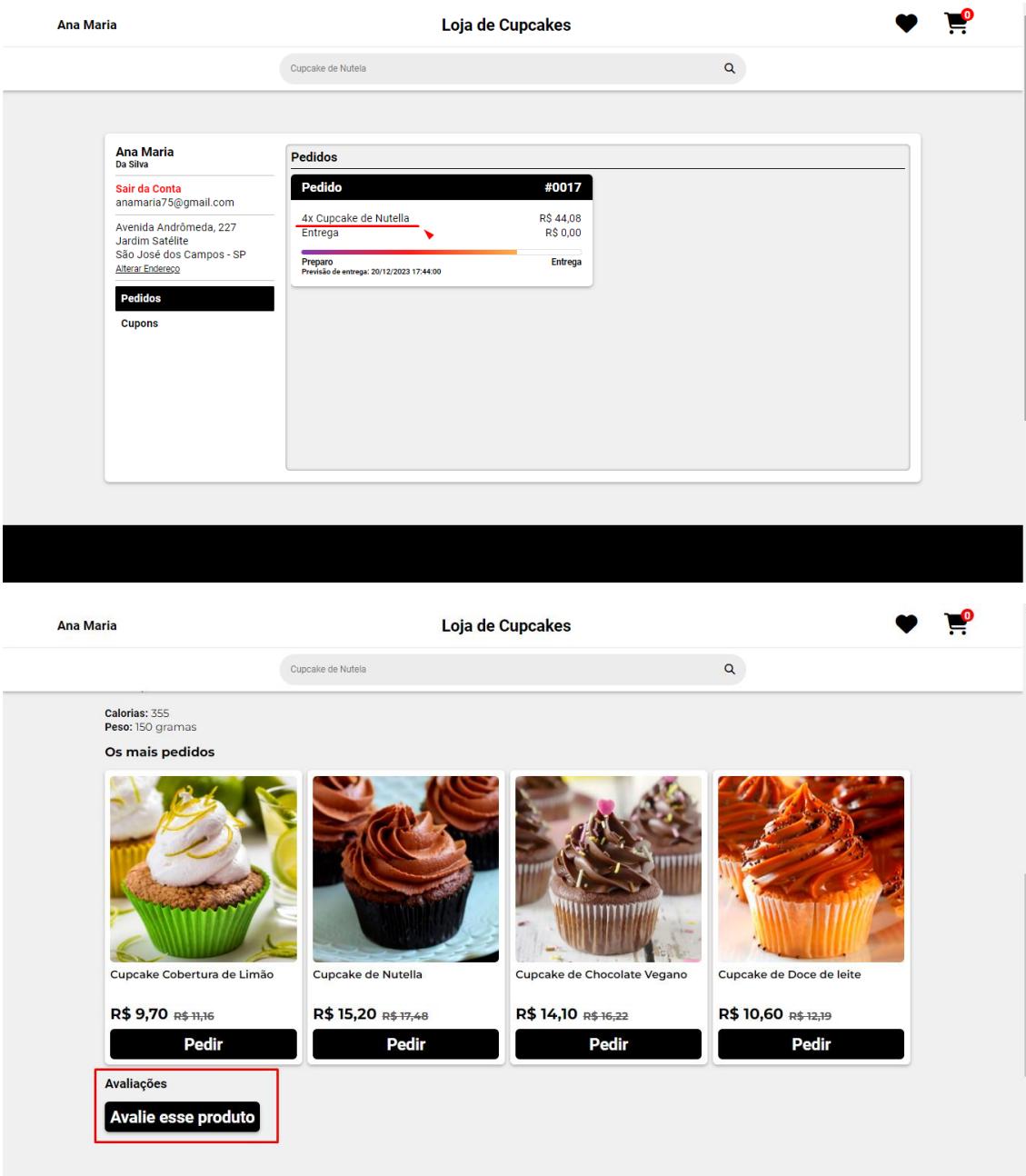
Sucesso, acesso a tela usuário.

## Deixar avaliação do produto

Script:

1. Acessar tela do cupcake adquirido clicando no card de pedido;
2. Clicar no botão “avalie esse produto”.
3. Escrever a avaliação no campo;
4. Clicar no botão “Avaliar”.

## Desktop:



Ana Maria

Loja de Cupcakes

Cupcake de Nutella

Search icon

Ana Maria  
Da Silva  
[Sair da Conta](#)  
anamaria75@gmail.com  
Avenida Andrômeda, 227  
Jardim Satélite  
São José dos Campos - SP  
[Alterar Endereço](#)

**Pedidos**

**Pedidos** #0017

4x Cupcake de Nutella

Entrega →

R\$ 44,08  
R\$ 0,00

Preparo → Entrega

Previsão de entrega: 20/12/2023 17:44:00

**Calorias: 355**  
**Peso: 150 gramas**

**Os mais pedidos**

Cupcake Cobertura de Limão	Cupcake de Nutella	Cupcake de Chocolate Vegano	Cupcake de Doce de leite
			
R\$ 9,70 R\$ 11,16	R\$ 15,20 R\$ 17,48	R\$ 14,10 R\$ 16,22	R\$ 10,60 R\$ 12,19
<b>Pedir</b>	<b>Pedir</b>	<b>Pedir</b>	<b>Pedir</b>

**Avaliações**

**Avalie esse produto**

Ana Maria

## Loja de Cupcakes



Cupcake de Nutela



Calorias: 355  
Peso: 150 gramas

### Os mais pedidos



Cupcake Cobertura de Limão

R\$ 9,70 R\$ 11,16

Pedir



Cupcake de Nutela

R\$ 15,20 R\$ 17,48

Pedir



Cupcake de Doce de leite

R\$ 10,60 R\$ 12,19

Pedir

### Avaliações

**Avalie esse produto**

Ana Maria

## Loja de Cupcakes



Cupcake de Nutela



Cupcake Cobertura de Limão

R\$ 9,70 R\$ 11,16

Pedir



Cupcake de Nutela

R\$ 15,20 R\$ 17,48

Pedir



Cupcake de Chocolate Vegano

R\$ 14,10 R\$ 16,22

Pedir



Cupcake de Doce de leite

R\$ 10,60 R\$ 12,19

Pedir

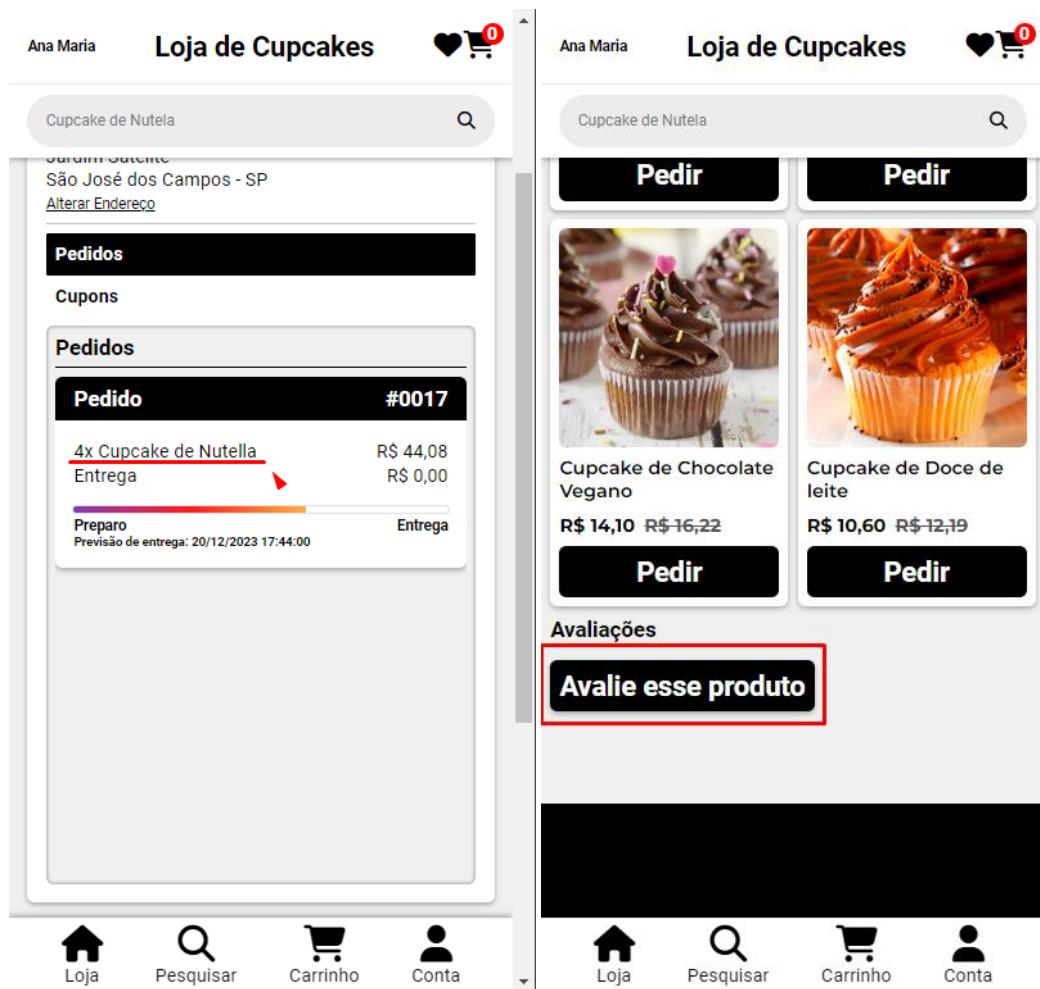
### Avaliações

**Avalie esse produto**

Ana Maria

Cupcake muito gostoso.

**Mobile:**



The image displays two side-by-side screenshots of a mobile application interface for a bakery named "Loja de Cupcakes".

**Left Screenshot (Customer Profile View):**

- User Info:** Ana Maria
- Store Name:** Loja de Cupcakes
- Cart Icon:** Heart icon with a red notification dot (0)
- Search Bar:** Cupcake de Nutella
- Address:** Jardim Sereino, São José dos Campos - SP
- Buttons:** Pedidos (highlighted), Cupons
- Pedidos Section:**
  - Order #0017:** 4x Cupcake de Nutella (underlined) - R\$ 44,08
  - Preparation Progress:** Preparo (progress bar) - Entrega (progress bar)
  - Delivery Info:** Previsão de entrega: 20/12/2023 17:44:00

**Right Screenshot (Product Catalog View):**

- User Info:** Ana Maria
- Store Name:** Loja de Cupcakes
- Cart Icon:** Heart icon with a red notification dot (0)
- Search Bar:** Cupcake de Nutella
- Buttons:** Pedir (highlighted twice)
- Product Grid:**
  - Cupcake de Chocolate Vegano:** Image, Price R\$ 14,10 / R\$ 16,22
  - Cupcake de Doce de leite:** Image, Price R\$ 10,60 / R\$ 12,19
- Avaliações Section:**
  - Button:** Avalie esse produto (highlighted with a red border)



Ana Maria    Loja de Cupcakes   

Cupcake de Nutela

**Pedir** **Pedir**

**fechar**

Cupcake muito gostoso.

**Avaliar**

**Avalie esse produto**

Loja    Pesquisar    Carrinho    Conta

Ana Maria    Loja de Cupcakes   

Cupcake de Nutela

**Cupcake de Chocolate Vegano**  
R\$ 14,10 R\$ 16,22 **Pedir**

**Cupcake de Doce de leite**  
R\$ 10,60 R\$ 12,19 **Pedir**

**Avaliações**

**Avalie esse produto**

Ana Maria  
Cupcake muito gostoso.

Loja    Pesquisar    Carrinho    Conta

### Resultado:

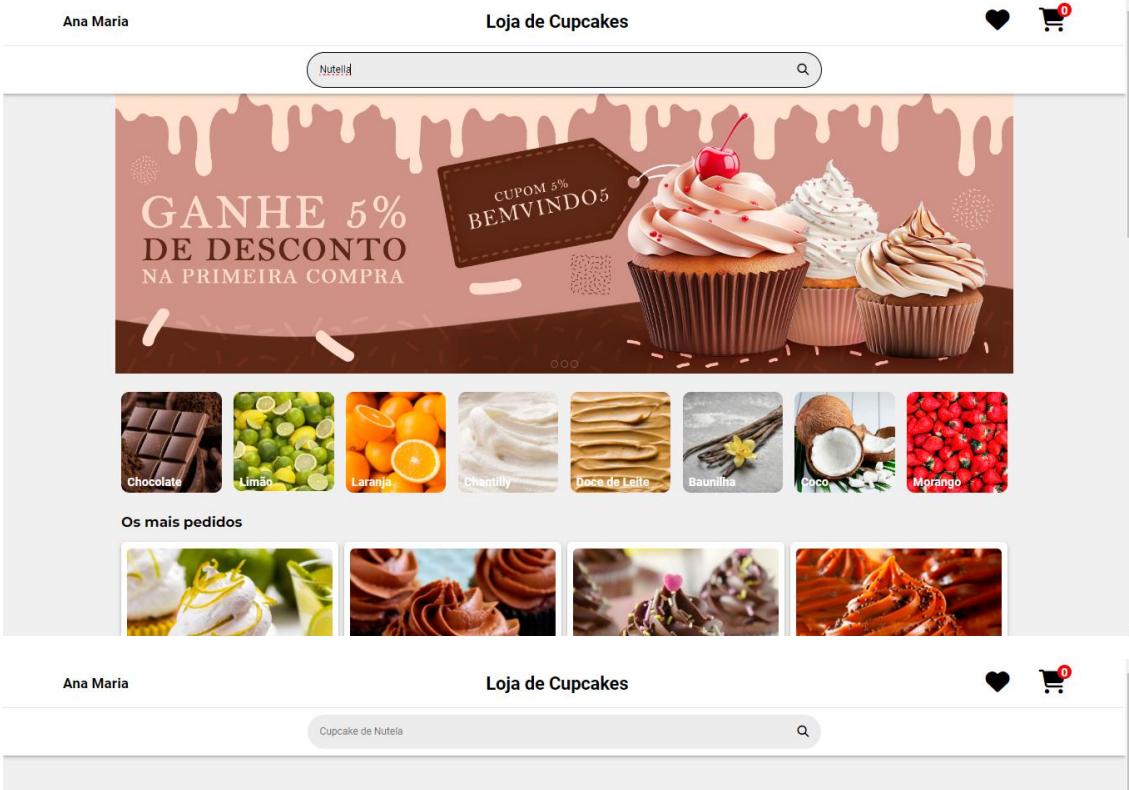
Sucesso, avaliação realizada.

## Pesquisa de cupcakes

Script:

1. Acessar o campo de pesquisa em qualquer tela do site;
2. Digitar uma palavra chave.
3. Pressionar tecla “enter”;

### Desktop:



Ana Maria Loja de Cupcakes

Nutella

GANHE 5% DE DESCONTO NA PRIMEIRA COMPRA

CUPOM 5% BEMVINDO5

Chocolate Limão Laranja Chantilly Doce de Leite Baunilha Coco Morango

Os mais pedidos

Ana Maria Loja de Cupcakes

Cupcake de Nutella

**Nutella**

Categorias

Vegano Chocolate

Frutas

Filtros

Massa

Massa de Cenoura  
Massa de Limão  
Massa de Laranja  
Massa de Chocolate Vegana  
Massa de Chocolate  
Massa de Baunilha Vegana  
Massa de Baunilha

Recheio

Creme de Chocolate Vegano  
Laranja Vegano  
Limão Vegano  
Laranja  
Limão  
Brigadeiro  
Doce de leite  
Nutella  
Sem recheio

Cobertura

Creme de Chocolate Vegano  
Laranja Vegano

Cupcake de Nutella

R\$ 15,20 R\$ 17,48

Pedir



**Mobile:**

Ana Maria    Loja de Cupcakes     

Nutella    



GANHE 5% DE DESCONTO NA PRIMEIRA COMPRA

CUPOM DE BEMVINDO

Chocolate    Limão    Laranja    Chantilly

Os mais pedidos



Cupcake Cobertura de Limão

R\$ 9,70 R\$ 11,16

Pedir



Cupcake de Nutella

R\$ 15,20 R\$ 17,48

Pedir

 Loja     Pesquisar     Carrinho     Conta

Ana Maria    Loja de Cupcakes     

Cupcake de Nutella    

## Nutella

Categorias



Cupcake de Nutella

R\$ 15,20 R\$ 17,48

Pedir

1 2

 Loja     Pesquisar     Carrinho     Conta

**Resultado:**

Sucesso, pesquisa realizada.

### Codificação:

Na Tabela a seguir insira as informações referentes ao desenvolvimento do código do *front-end* e *back-end*.

<b>Linguagem do Back-end</b>	Node.js - JavaScript
<b>Banco de Dados</b>	MySQL
<b>Hospedagem</b>	<a href="https://render.com/">https://render.com/</a>
<b>Plataforma</b>	Render
<b>Modo de Codificação</b>	(X) Tradicional ( ) Low-code ( ) No-code
<b>Link do repositório no GitHub com os códigos abertos</b>	<a href="https://github.com/Helisy/cupcacke_store_app">https://github.com/Helisy/cupcacke_store_app</a>
<b>Link da solução em funcionamento</b>	<a href="https://cupcacke-store-app.onrender.com/">https://cupcacke-store-app.onrender.com/</a>
<b>Link do vídeo narrado (no mínimo 5 min)</b>	<a href="https://www.youtube.com/watch?v=szq09huAMMs">https://www.youtube.com/watch?v=szq09huAMMs</a>

**\*Nota ao avaliador:** O aplicativo está hospedado na plataforma render.com, com servidores localizados em Ohio, EUA. Devido a essa localização geográfica, o primeiro acesso do site pode apresentar uma demora e causar alguns atrasos durante a utilização.

## Testes da Solução

Escolha 5 colegas para testar sua aplicação e preencha a tabela a seguir com as informações obtidas:

<b>Nome:</b> João Pedro dos Santos Martins	<b>Data do teste:</b> 25/11/2023
<b>O que testou e funcionou:</b> Criação de conta	
<b>O que testou e não funcionou – O que deve ser corrigido:</b>	
<b>Funcionalidade não testada (faltou ou não foi implementada):</b>	

<b>Nome:</b> Vitor Oliveira Ramos	<b>Data do teste:</b> 25/11/2023
<b>O que testou e funcionou:</b> Finalização de pedido	
<b>O que testou e não funcionou – O que deve ser corrigido:</b>	
<b>Funcionalidade não testada (faltou ou não foi implementada):</b> Opções de pagamento	

<b>Nome:</b> Henrique Oliveira Moreira	<b>Data do teste:</b> 25/11/2023
<b>O que testou e funcionou:</b> Adição de múltiplos itens no carrinho	
<b>O que testou e não funcionou – O que deve ser corrigido:</b> Bug no layout após certa quantidade de itens adicionados. Correção no estilo da tela de carrinho.	
<b>Funcionalidade não testada (faltou ou não foi implementada):</b>	

<b>Nome:</b> Cristhian Daniel	<b>Data do teste:</b> 26/11/2023
<b>O que testou e funcionou:</b>	
<b>O que testou e não funcionou – O que deve ser corrigido:</b> Avaliação do produto. Criação da avaliação foi feita corretamente, mas a tela não atualizou para mostrar a avaliação.	
<b>Funcionalidade não testada (faltou ou não foi implementada):</b>	
<b>Nome:</b> Thayna Lopes Rocha	<b>Data do teste:</b> 26/11/2023
<b>O que testou e funcionou:</b> Sistema de busca	
<b>O que testou e não funcionou – O que deve ser corrigido:</b>	
<b>Funcionalidade não testada (faltou ou não foi implementada):</b>	

### Laudo de Qualidade

Insira a seguir o laudo de qualidade do sistema, apontando os erros e as correções. Não esqueça de coletar as evidências para inseri-las no laudo:

Após os testes realizado, constata-se que todas as funcionalidades principais estão operando conforme esperado, demonstrando uma integração eficiente e uma experiência do usuário satisfatória. Algumas melhorias foram necessárias, concentram-se principalmente nas correções do CSS de algumas páginas.

## Avaliação das Funcionalidades:

### Cadastro e Login de Usuários:

O processo de cadastro de novos usuários foi realizado com sucesso, sem incidentes. O login e autenticação de usuários ocorrem de maneira correta.

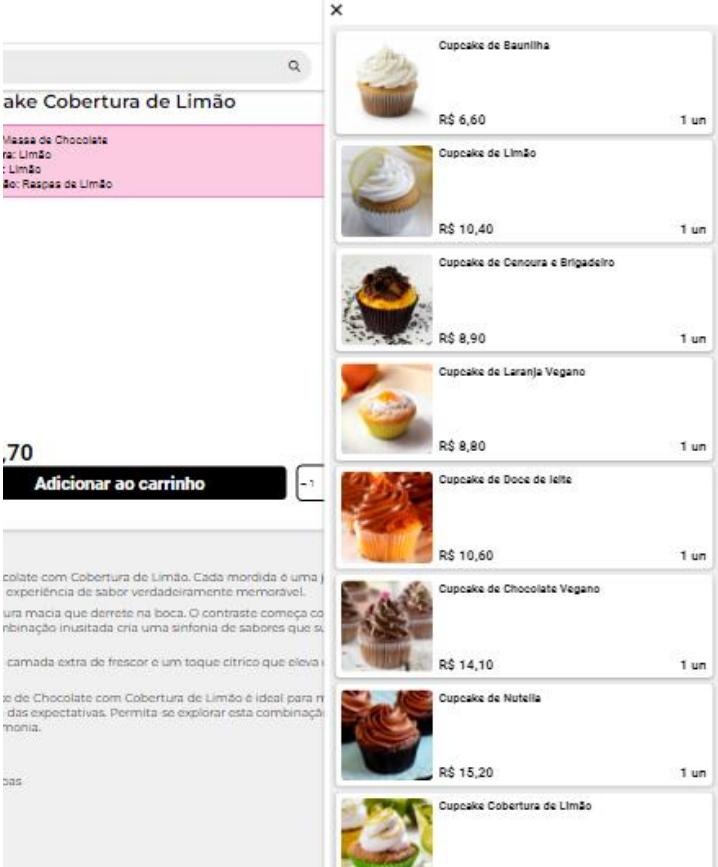
### Finalização de Pedido:

O processo de finalização de pedidos, incluindo as etapas de adição, modificação e exclusão de produtos. Assim como a escolha de opções de entrega e utilização de cupons.

### Funcionalidades do Carrinho:

Adição, remoção e atualização de itens no carrinho ocorrem sem falhas. Correções no estilo da página de carrinho foram necessárias para acomodar adição de múltiplos produtos.

Problema encontrado:

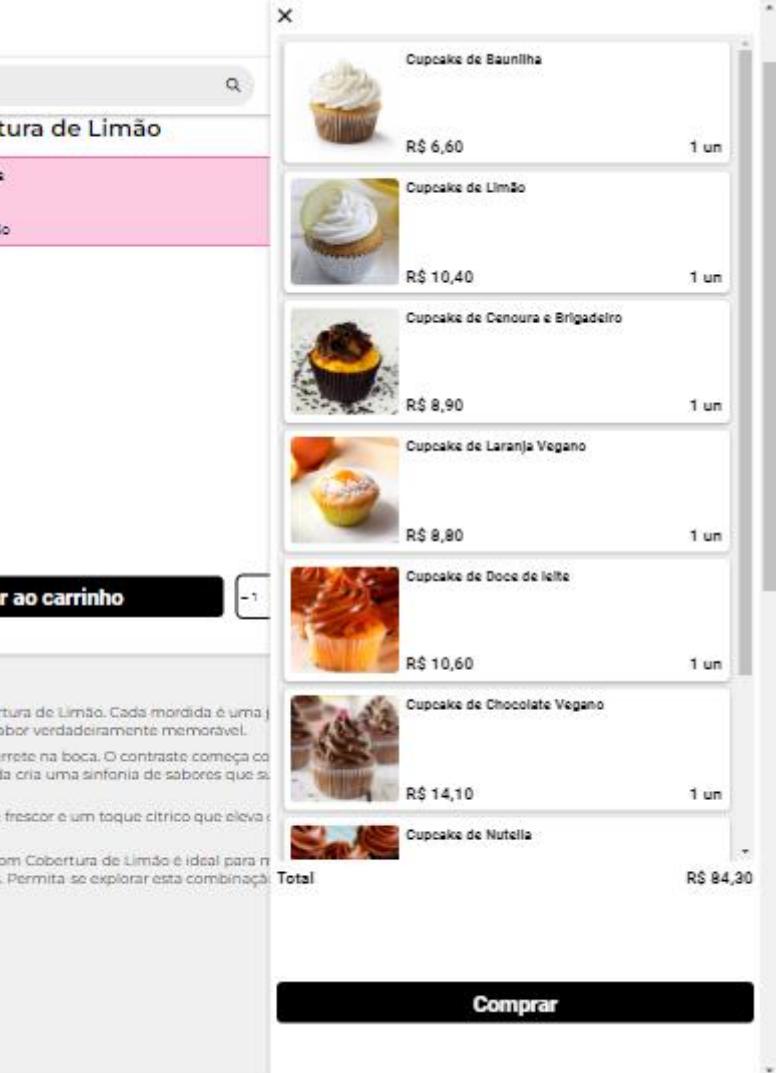


The screenshot shows a shopping cart interface with the following details:

Item	Preço	Quantidade
Cupcake de Baunilha	R\$ 6,60	1 un
Cupcake de Limão	R\$ 10,40	1 un
Cupcake de Cenoura e Brigadeiro	R\$ 8,90	1 un
Cupcake de Laranja Vegano	R\$ 8,80	1 un
Cupcake de Doce de Leite	R\$ 10,60	1 un
Cupcake de Chocolate Vegano	R\$ 14,10	1 un
Cupcake de Nutella	R\$ 15,20	1 un
Cupcake Cobertura de Limão		1 un

A pink highlight covers the second row of the cart, indicating a duplicate item. The total value shown is ,70.

Após correção:



The screenshot shows a catalog page for cupcakes. A search bar at the top left has the text "Cobertura de Limão". Below it, a sidebar lists categories: "Cobertura de Limão", "Cobertura de Chocolate", and "Cobertura de Nutella". The main content area displays a list of cupcakes with their names, prices, and quantities. The first item is "Cupcake de Baunilha" at R\$ 6,60, quantity 1 un. The second item is "Cupcake de Limão" at R\$ 10,40, quantity 1 un. The third item is "Cupcake de cenoura e Brigadeiro" at R\$ 8,90, quantity 1 un. The fourth item is "Cupcake de Laranja Vegano" at R\$ 8,80, quantity 1 un. The fifth item is "Cupcake de Doce de Leite" at R\$ 10,60, quantity 1 un. The sixth item is "Cupcake de Chocolate Vegano" at R\$ 14,10, quantity 1 un. The total price at the bottom is R\$ 84,30. A large black button at the bottom right says "Comprar".

Nome	Preço	Quantidade
Cupcake de Baunilha	R\$ 6,60	1 un
Cupcake de Limão	R\$ 10,40	1 un
Cupcake de Cenoura e Brigadeiro	R\$ 8,90	1 un
Cupcake de Laranja Vegano	R\$ 8,80	1 un
Cupcake de Doce de Leite	R\$ 10,60	1 un
Cupcake de Chocolate Vegano	R\$ 14,10	1 un
Total		R\$ 84,30

### Avaliação e Comentários:

Os usuários têm a capacidade de avaliar os cupcakes. Houve um bug onde a tela não apresentava a avaliação recém feita, uma pequena correção foi feita.

### Pesquisa e Navegação:

A pesquisa e navegação pelo catálogo de cupcakes gourmet são intuitivas, facilitando a identificação e seleção de produtos.

## Conclusão:

O site é responsivo em dispositivos variados, garantindo uma experiência consistente em diferentes tamanhos de tela.

No geral, o site está pronto para proporcionar aos usuários uma experiência de compra online positiva e agradável com apenas alguns problemas visuais detectados.

## Correções no CSS:

Algumas páginas apresentam pequenas inconsistências no layout e na formatação.

Correção específicas no CSS foram necessárias para melhorar a apresentação visual.

## Api e Backend:

Nos testes realizados não foi encontrada nenhuma falha ou comportamento inesperado no funcionamento da Api.

## Vídeo da Solução atualizada

Após levantar os *feedbacks* e executar as correções necessárias e pertinentes, grave um vídeo de até 5 minutos apresentando as modificações realizadas no sistema.

***Link para o vídeo***

<https://www.youtube.com/watch?v=szq09huAMMs>



Cruzeiro do Sul Virtual  
Educação a distância