AUTOMAÇÃO API

# Passo 1 - Criar repositório no github

* Crie um repositório no seu github com o nome “api-qas-trainings-icarotech”
* Clone esse repositório na sua máquina local

# Passo 2 - Passar arquivos iniciais para o seu repositório

* Acesse a pasta do drive [link](https://drive.google.com/drive/folders/1LPTCwGl0a8CvH52MfBasZQQFrzAOJvOq)
* Baixe o zip do projeto para sua máquina local
* Extraia a pasta
* Copie todos arquivos e cole na pasta do seu repositório que foi clonado no “passo 1”

# Passo 3 - Executar os primeiros comando para abrir o cypress

* Alterar versão do cypress para "^7.4.0"

// Verificar a versão atual

$ npx cypress --version

// Instalando a versão @7.4.0

$ npm install cypress@7.4.0

// Validar a atualização da versão

$ npx cypress --version



* Entre na pasta do seu repositório local “api-qas-trainings-icarotech”
* Execute o primeiro comando

$ npm install



* Execute o segundo comando

$ npx cypress open



# Passo 4 - Criar script de teste para funcionalidade usuários.

# Obs: Mateus fará com os QAs

## Criar Usuário

* + **Descrição:** Criar um novo usuário no sistema.
  + **HTTP Method:** POST
  + **Endpoint:** https://api-qas-trainings.glitch.me/users

## Listar Todos os Usuários

* + **Descrição:** Retornar uma lista de todos os usuários cadastrados.
  + **HTTP Method:** GET
  + **Endpoint:** https://api-qas-trainings.glitch.me/users

## Validar Usuário Criado

* + Confirmar que o usuário criado aparece na listagem retornada pelo endpoint /users.

# Passo 5 - Desafio - Criar automação para funcionalidade Produto

Obs: QAs farão sozinhos

## Criar Produto

* + **HTTP Method:** POST
  + **Endpoint:** https://api-qas-trainings.glitch.me/products
  + **Descrição:** Criar um novo produto no sistema.

## Listar todos os Produtos

* + **HTTP Method:** GET
  + **Endpoint:** https://api-qas-trainings.glitch.me/products
  + **Descrição:** Retornar uma lista de todos os produtos cadastrados.

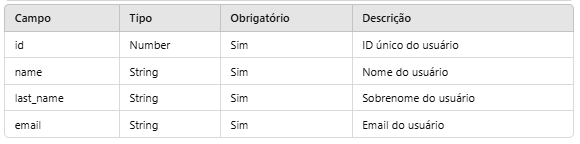
## Validação Final

* + Confirmar que os produtos criados aparecem na listagem de /products.
  + Garantir que as operações relacionadas aos usuários permaneçam funcionando corretamente após a refatoração.

# DOC API - Estrutura das Requisições

## **Funcionalidade: Usuários**

Entrada



Headers:

* Content-Type: application/json

Body:

{

"id": number,

"name": "string",

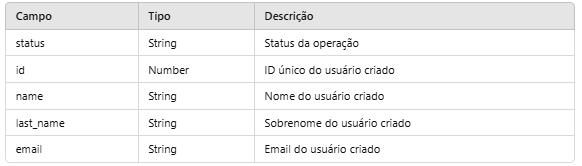
"last\_name": "string",

"email": "string"

}

## 

Resposta Esperada



{

"status": "string",

"id": number,

"name": "string",

"last\_name": "string",

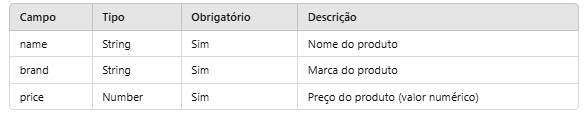
"email": "string"

}



## **Funcionalidade: Produtos**

Entrada



Headers:

* Content-Type: application/json

Body:

{

"name": "string",

"brand": "string",

"price": number

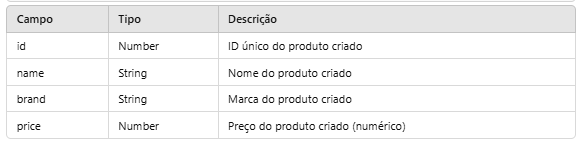
}



#### 

#### 

Resposta Esperada



Status Code: 200 OK

Resposta:

**{**

**"id": number,**

**"name":"string",**

**"brand":"string",**

**"price": number**

**}**

****