


# Projeto De Testes

## Portifólio de QA

Selenium • Python • BDD • Pytest

**Thais Fialho**

 GitHub 

# Porque criei esse projeto?

- Desenvolver um projeto de QA para o meu portfólio;
- Demonstrar conhecimento prático em casos de teste, testes funcionais, automação e organização de código;
- Treinar boas práticas, versionamento e documentação;

# Parte 1

## Primeira tentativa: Petz



# Objetivo inicial

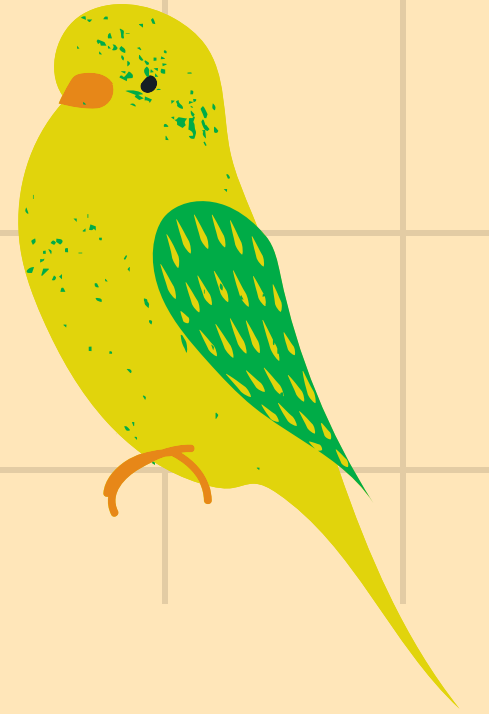
- Mapear e testar os principais fluxos do site da Petz;
- Casos de teste do sistema de busca, carrinho, checkout e login;
- Automação de Login;

## Por que escolhi esse site?

- Site grande e real;
- Permitia cobrir fluxos autênticos de e-commerce;
- Oportunidade de testar cenários complexos;



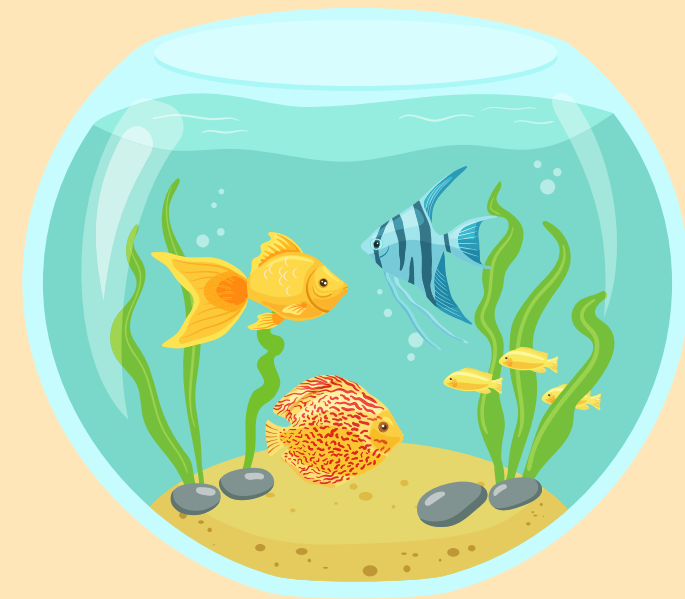
# O problema encontrado



## Principais dificuldades:

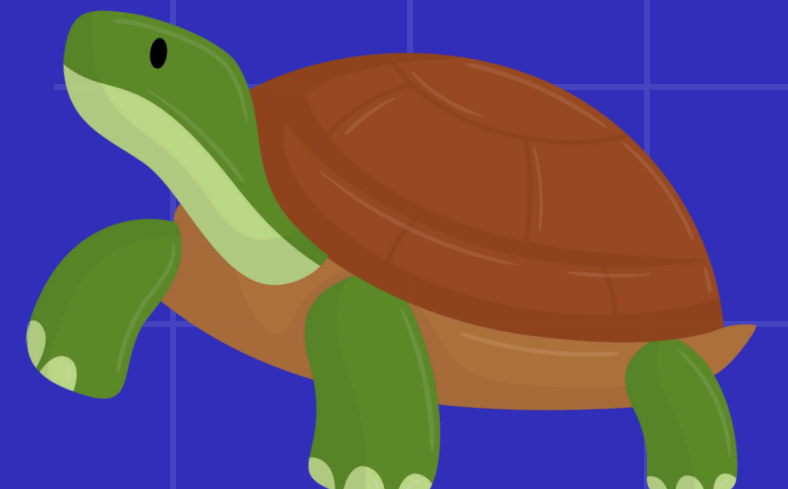
- ✗ Sites reais possuem bloqueios contra automação:
  - Selenium e Cypress identificados e bloqueados;
  - Captchas e mecanismos anti-bot impediam execução dos testes;
  - Elementos sendo carregados de forma dinâmica e imprevisível;

- ✗ Sem acesso oficial → sem forma de contornar as restrições



# Aprendizados: Petz

- Analisar limitações técnicas antes de iniciar a automação;
- Tratar flutuações de carregamento de página;
- Depurar seletor, timing, waits, iframes;
- Importância de ambientes controlados (sandbox);
- Domínio maior de Selenium através das tentativas;



# Parte 2

**Projeto Final: ServeRest**

# Motivação

- Ambiente criado exclusivamente para estudos;
- Zero bloqueios de automação;
- Possui funcionalidades de e-commerce;
- Funciona para demonstrar automação;
- Focado na prática de QA, não em bypass de segurança:

**SERVEREST**

## Login

Entrar

Não é cadastrado? [Cadastre-se](#)



# Objetivo Técnico

- Criar cenários de teste bem estruturados;
- Automatizar cenários positivos e negativos de login;
- Garantir estabilidade dos testes;
- Organizar o código usando POM;
- Documentar os casos de teste em Gherkin;

# Ferramentas e Métodos:

- Python;
- Selenium;
- Pytest;
- BDD / Gherkin;
- Page Object Model (POM);
- Git + GitHub;

# Resultado Dos Testes

- Login válido funciona;
- Feedback correto para e-mail/senha inválidos;
- Campos obrigatórios validados;
- Interface responde corretamente;
- Cobertura de cenários negativos e positivos;

# Aprendizados Finais

- Ambientes reais têm limitações para automação sem autorização;
- Acesso a vulnerabilidade dos sistemas: responsabilidade maior;
- Importância da capacidade de adaptação nos projetos;
- Aplicação de novo aprendizado técnico: BDD em Gherkin;
- Aperfeiçoamento no uso de ferramentas já estudadas antes;