**Home Challenge FrontEnd**

**Sumário**

[1 – A Entrega 2](#__RefHeading___Toc1_880807835)

[1.2 – Código fonte 2](#__RefHeading___Toc107_880807835)

[1.2 – Executável 2](#__RefHeading___Toc115_880807835)

[1.3 – A interface da solução 2](#__RefHeading___Toc187_880807835)

[2 – Questões 4](#__RefHeading___Toc189_880807835)

[2.1 – Quanto tempo foi gasto no teste? 4](#__RefHeading___Toc191_880807835)

[2.2 – O que você adicionaria na solução se você tivesse mais tempo? 4](#__RefHeading___Toc193_880807835)

[2.3 – Como você melhoraria a API que você teve que consumir? 4](#__RefHeading___Toc195_880807835)

[2.4 – O que você achou mais difícil? 4](#__RefHeading___Toc231_880807835)

[2.5 – O que você achou da experiência geral do nosso desafio, algum feedback para nós? 4](#__RefHeading___Toc233_880807835)

# 1 – **A E**ntrega

## 1.2 – Código fonte

Repositório no github contendo o código fonte:

https://github.com/dnoliveira/HomeChallenge\_FrontEnd

## 1.2 – Executável

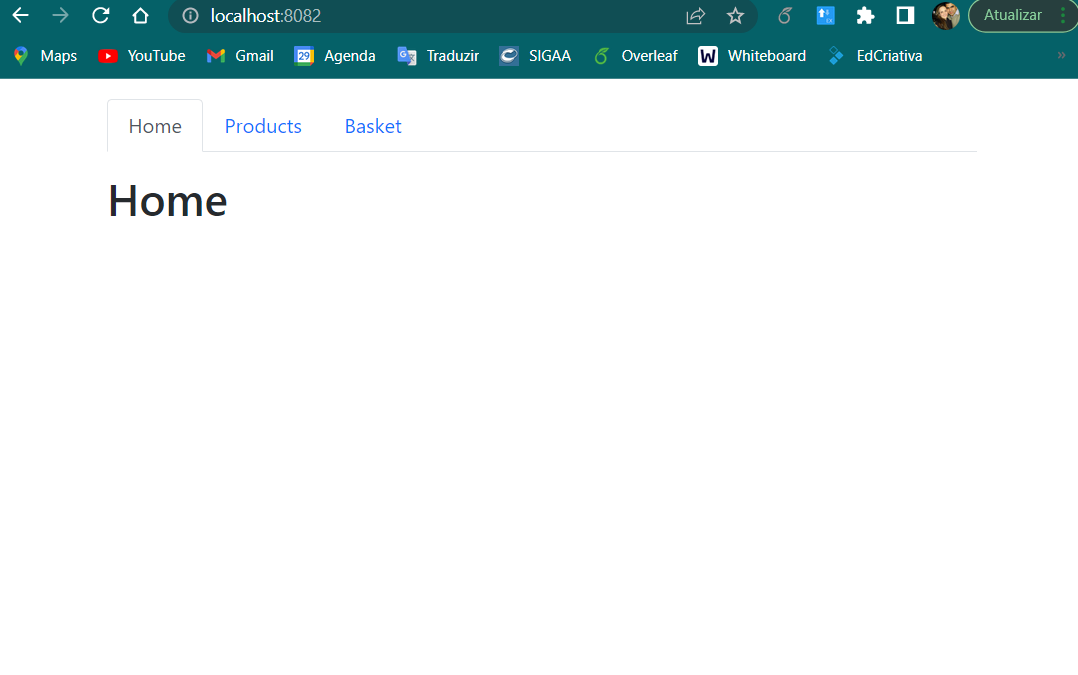
O executável se encontra no link:

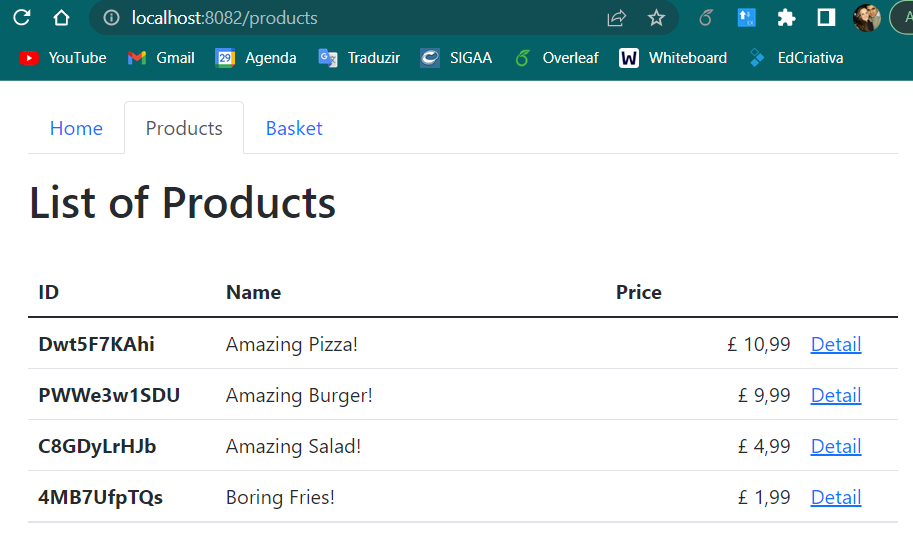
https://github.com/dnoliveira/HomeChallenge\_FrontEnd/blob/main/runnable.zip

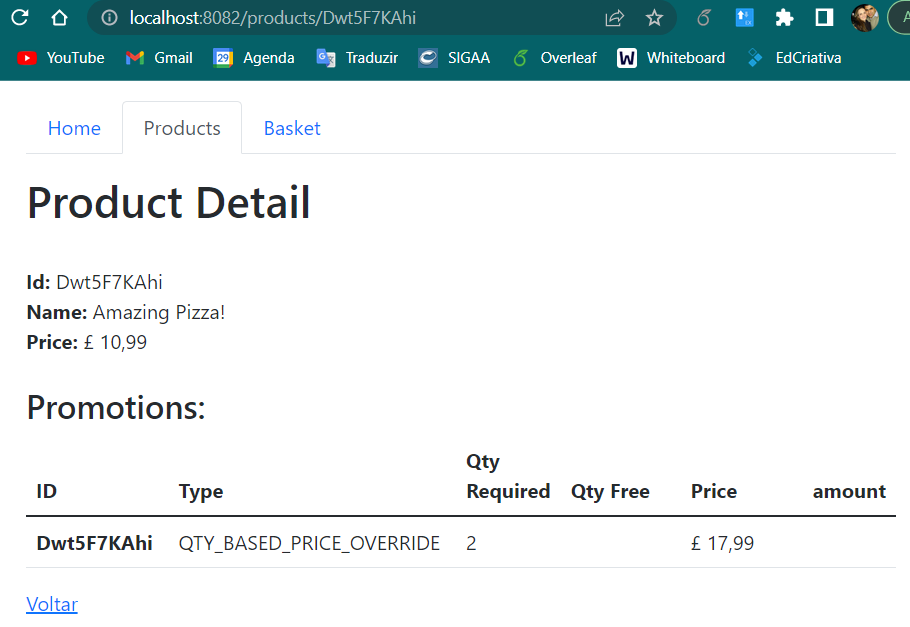
Instruções:

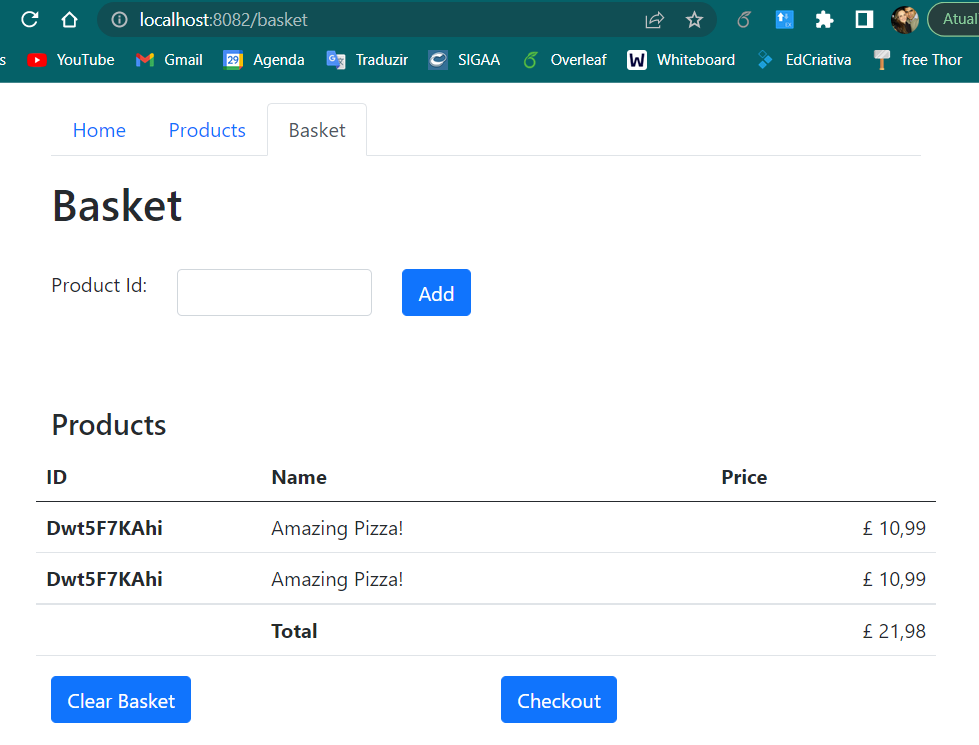
1. Ter o Java 11 instalado.
2. Descompactar o arquivo “runnable.zip”.
3. Rodar o arquivo “start.bat” ou “start.sh”.
   1. Caso necessário o arquivo deverá ser editado informando assim corretamente a porta do servidor “front end” e url da API.

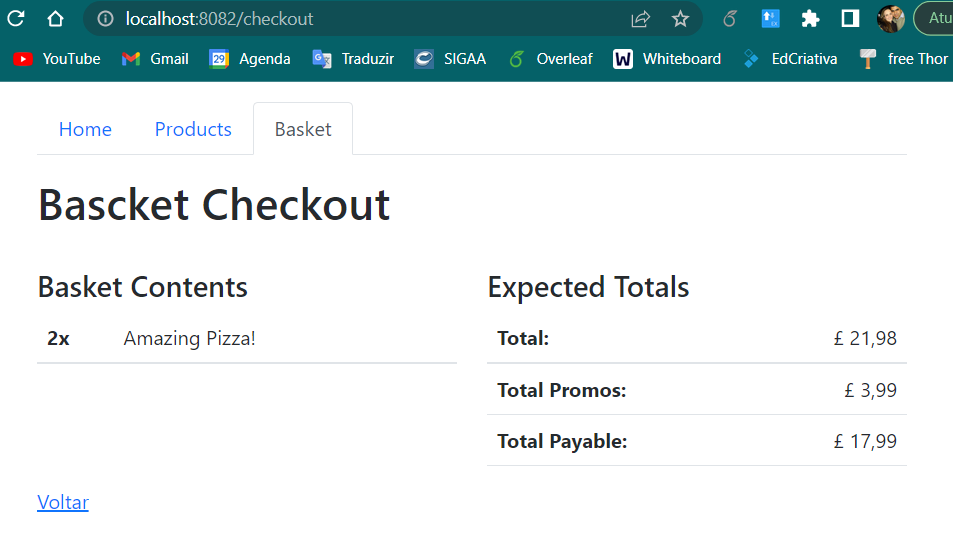
## 1.3 – A interface da solução











# 2 – Questões

## 2.1 – Quanto tempo foi gasto no teste?

Aproximadamente 20 horas.

## 2.2 – O que você adicionaria na solução se você tivesse mais tempo?

1. Na cesta
   1. Implementar a exclusão de um produto;
   2. Pedir confirmação da limpeza da cesta;
   3. Exibir uma informação de “Produto inexistente” ao adicionar um produto inexistente no banco de dados;
2. Comentar melhor o código fonte, principalmente nas regras de negócio;
3. Tratar melhor as exceções quando a API retorna um erro ou quando a API se encontra indisponível;
4. O retorno da API possibilita múltiplas promoções para um mesmo produto, em um segundo momento este requisito deveria ser discutido e implementado;
5. Criar mais cenários de teste utilizando Junit e Mockito.

## 2.3 – Como você melhoraria a API que você teve que consumir?

1. Alterar os retornos de preço, considerando assim valores com decimais. Isto exigiu um processamento extra e desnecessário;
2. Levar a parte do checkout para a API, desta forma será escrito apenas 1 vez e vários dispositivos com tecnologias diferentes poderiam utilizar;
3. Implementar o CRUD de produtos e promoções;
4. Implementar o controle de acesso.

## 2.4 – O que você achou mais difícil?

O “checkout” gastou mais tempo de análise, implementação e testes. Neste item me preocupei no crescimento do sistema e no aumento de tipos de promoções.

## 2.5 – O que você achou da experiência geral do nosso **desafio**, algum feedback para nós?

Gostei muito deste desafio, percebi que ele se encontra bem aberto para perfis diferentes (júnior, pleno e sênior). Tive liberdade de ir explorando o leque de tecnologias que domino. Oportunidade também para realizar críticas aos requisitos abordados.