**Home Challenge FrontEnd**

**Sumário**

[​ 1 – The delivery 2](#__RefHeading___Toc1_880807835)

[​ 1.2 – Source code 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 – The delivery

## 1.2 – Source code

Repository of source code on github:

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

## 1.2 – Executável

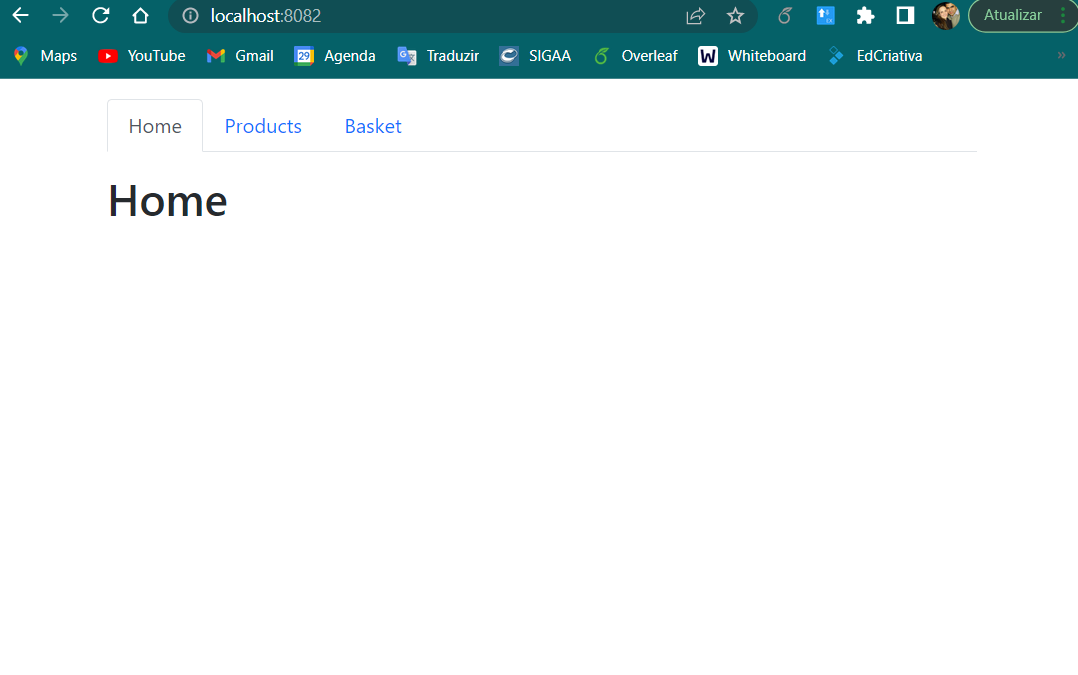
Runnable file on github:

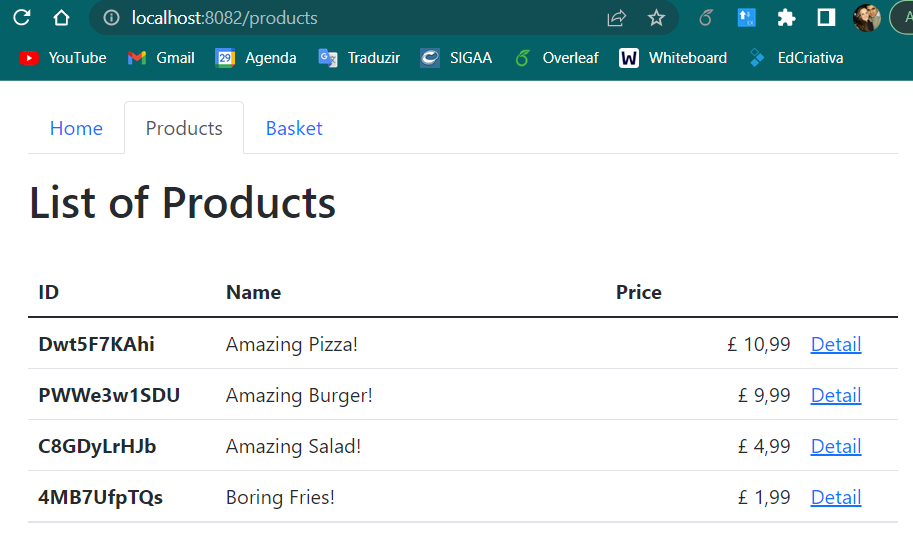
https://github.com/dnoliveira/HomeChallenge\_FrontEnd/blob/main/runnable.zip

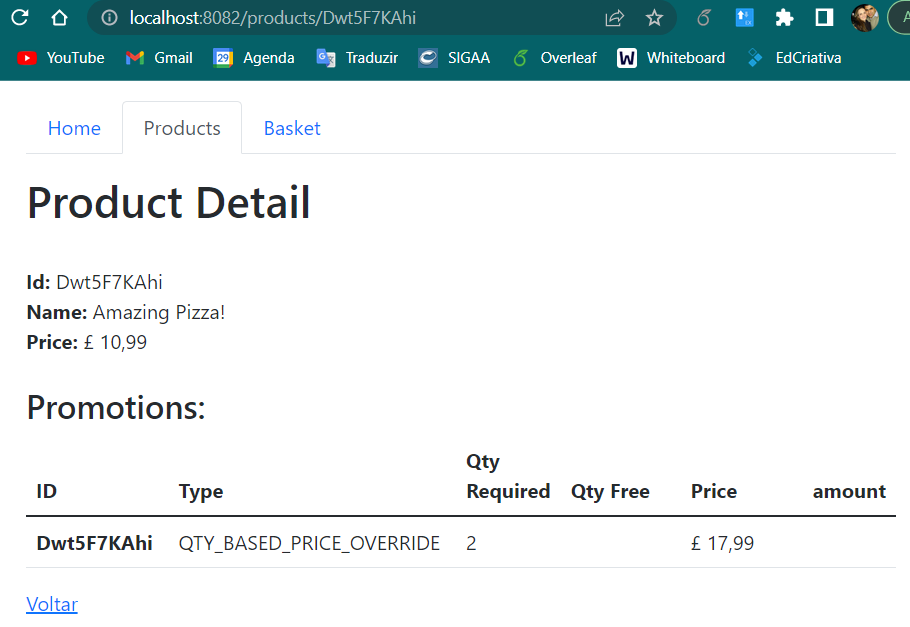
Instructions:

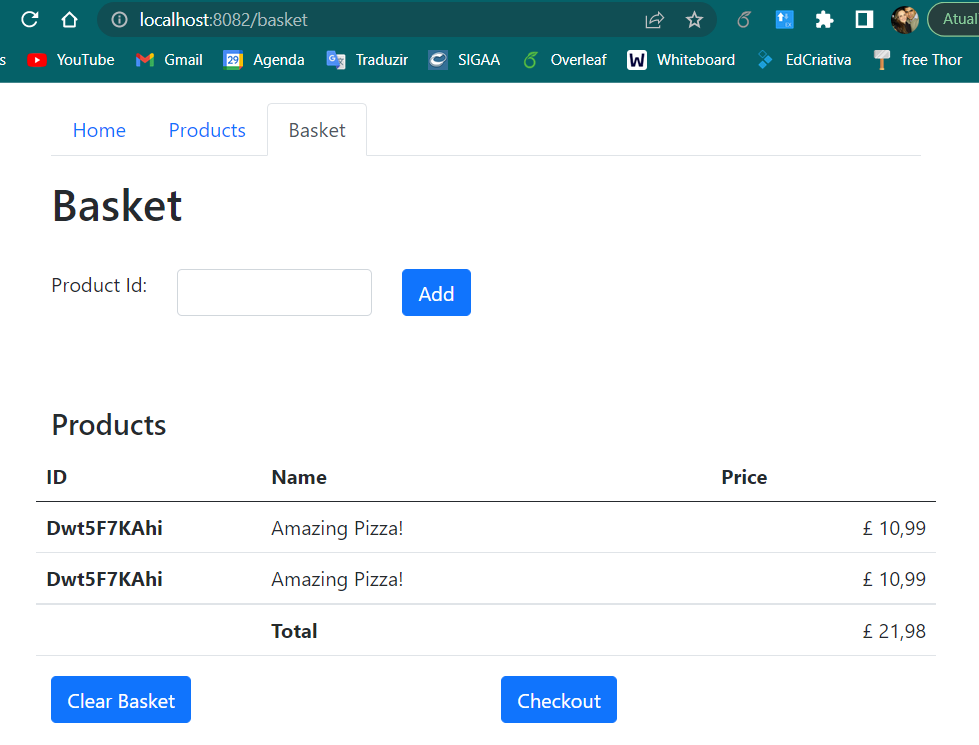
1. Have Java 11 installed.
2. Unzip the “runnable.zip” file.
3. Run the “start.bat” or “start.sh” file.
   1. If necessary, the file must be edited, correcting the information of the “front end” server port and API url.

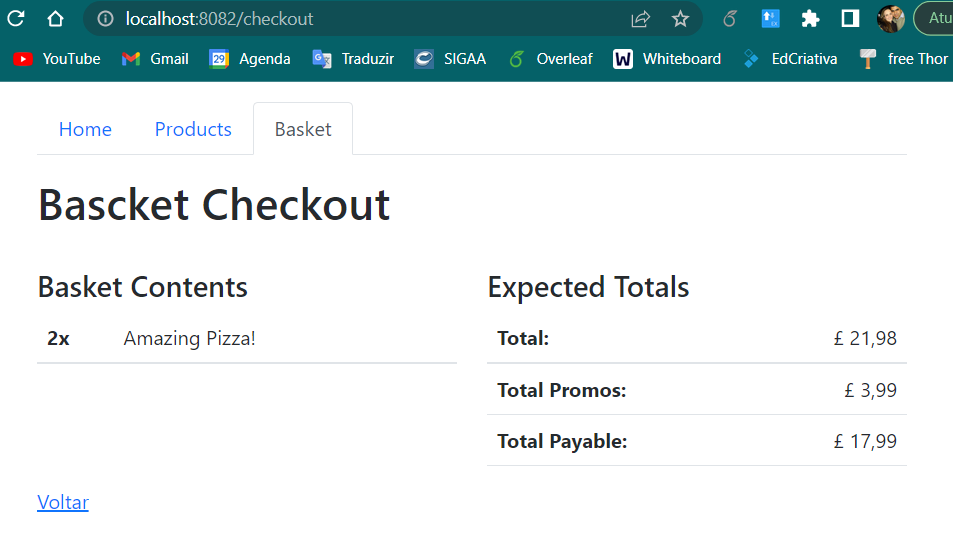
## 1.3 – The solution interface











# 2 – Questions

## 2.1 – How long did you spend on the test?

Approximately 20 hours.

## 2.2 – What would you add if you had more time?

1. In the basket
   1. Implement the deletion of a product;
   2. Ask for confirmation of basket cleaning;
   3. Display a “non-existent product” information when adding a non-existent product in the database;
2. Better comment the source code, mainly in the business rules;
3. Handle exceptions better when the API returns an error or when the API is unavailable;
4. The return of the API allows multiple promotions for the same product, in a second moment this requirement should be discussed and implemented;
5. Create more test scenarios using Junit and Mockito.

## 2.3 – How would you improve the product APIs that you had to consume?

1. Change the price returns, thus considering values with decimals values. This required extra and unnecessary processing;
2. Take the checkout part to the API, this way it will only be written once and several devices with different technologies could use it;
3. Implement CRUD for products and promotions;
4. Implement access control.

## 2.4 – What did you find most difficult?

The “checkout” spent more time on analysis, implementation and testing. In this item I was concerned with the growth of the system and the increase in types of promotions.

## 2.5 – How did you find the overall experience, any feedback for us?

I really enjoyed this challenge, I realized that it is very open to different profiles (junior, or senior). I was free to explore the range of technologies I master. Opportunity also to criticize the requirements.