

Plataforma de Vendas Online

Relatório

PI5, João Rodrigues de Campos
Bárbara Gonçalves, 2018295452
Carlos Matos, 2020245868
Mariana Magueijo, 2020246886



Instruções de uso

Para implementar e correr o software desenvolvido por nós, necessita de ter instalado:

- PostgreSQL - vai ser a aqui que se cria a base de dados;
- Python - linguagem do código fonte da aplicação. Tendo também de proceder à instalação dos seguintes módulos:
 - Flask - API REST para web development (pip install Flask);
 - Jwt - para tokens de autenticação dos utilizadores (pip install pyjwt);
 - Psycopg 2 - para adaptação do PostgreSQL (pip3 install pycopg2);
- Postman - simula um cliente REST, que trata dos pedidos (req) e respostas (res);

Começamos por correr a aplicação api.py. Esta vai-se conectar à base de dados. No postman, devemos criar requests adequadas a cada funcionalidade desenvolvida sendo assim possível a partir deste endereço realizar os pedidos e respostas pelo postman.

Manual de Uso

Registo de utilizadores:

POST - <http://localhost:8080/dbproj/user>

- O utilizador deve inserir no postman os parâmetros e respetivo input necessários para se registar (no Body, em formato raw-JSON). Não havendo nenhum utilizador logged in, só é possível registar compradores. Caso haja um utilizador que esteja logged in, verificamos se este é um administrador. Se não for, não tem permissão para registar outros utilizadores. Se for, pode registar quer vendedores, quer outros administradores. São feitas as devidas proteções de repetição de usernames e emails.

Autenticação de utilizadores:

PUT - <http://localhost:8080/dbproj/user>

- O utilizador insere o seu username e password. Se este já se encontrar registado e a sua password estiver certa, é gerado um token jwt (JSON web token), que vai ser passado ao header no postman sempre que um utilizador quiser fazer uma request que exija que o utilizador esteja logged in.

Adicionar produto:

POST - <http://localhost:8080/dbproj/product>

- Assim que o utilizador fizer log in, verificamos se este é um vendedor, pois apenas esses têm permissão para adicionar produtos. Após fazer a verificação, recorreremos à função checkBody para verificar que tipo de produto queremos adicionar. Se os parâmetros e o input forem válidos, o produto é adicionado.

Atualizar produto:**PUT - http://localhost:8080/dbproj/product/{product_id}**

- Verificamos se o utilizador é um vendedor. Verificamos depois se o produto existe e se o produto pertence ao vendedor. Caso o produto tenha inputs válidos e o seu tipo coincida, este é adicionado.

Efetuar uma compra:**POST - <http://localhost:8080/dbproj/order>**

- Verificamos se o utilizador é um comprador. Verificamos também se a compra que o utilizador quer realizar é válida, ou seja se o seu carrinho não está vazio. Caso o comprador queira utilizar um cupão, vemos se ele possui um, se o cupão é válido, e se ainda está dentro da validade, sendo depois calculado o desconto a aplicar na compra. Caso o produto exista e haja stock suficiente, a compra é efetuada.

Deixar ranting/feedback:**POST - http://localhost:8080/dbproj/rating/{product_id}**

- Caso um utilizador tenha comprado um produto, este pode realizar uma avaliação do mesmo. Começamos por verificar se o user é um comprador e se efetivamente comprou o produto a avaliar. Se este ainda não tiver avaliado o produto em questão, o seu rating/comentário é submetido.

Deixar comentário/pergunta: POST**http://localhost:8080/dbproj/questions/{product_id}****http://localhost:8080/dbproj/questions/{product_id}/{parent_question_id}**

- Qualquer comprador pode colocar questões/comentários sobre qualquer produto. É possível também responder a questões previamente colocadas.

Consultar informações genéricas de um produto:**GET - http://localhost:8080/dbproj/product/{product_id}**

- Qualquer comprador pode consultar informações dos produtos existentes, mesmo que não o tenha comprado. Após confirmar que o produto existe, são imprimidas as suas informações genéricas.

Obter estatísticas (por mês) nos últimos 12 meses:**GET - <http://localhost:8080/dbproj/report/year>**

- Funcionalidade não implementada.

Criar nova campanha:**POST - <http://localhost:8080/dbproj/campaign>**

- Verificamos se o utilizador é um admin, pois só estes têm permissão para criar campanhas. Caso nenhuma campanha esteja já a decorrer, esta é criada.

Subscrever campanha/cupões:**PUT - http://localhost:8080/dbproj/subscribe/{campaign_id}**

- Verificamos se o utilizador é um comprador. Depois, verificamos se a campanha a que ele está a concorrer ainda está a decorrer. Se o utilizador já tiver um cupão atribuído, vemos se este está dentro de validade ou não. Caso o cupão seja válido, não lhe é atribuído nada. Caso esteja fora de validade ou se o utilizador não tiver, é lhe atribuído um novo cupão, caso a campanha ainda tenha cupões suficientes.

Obter estatísticas dos descontos aplicados por campanha:**GET - <http://localhost:8080/dbproj/report/campaign>**

- Funcionalidade não implementada.



Funções auxiliares:

- geradorIds(): Gera ids randomizados.
- checkBody(): Verifica se o body tem todos os argumentos necessários para o bom funcionamento do programa.
- checkInput(): Verifica se não há inputs vazios no body.
- checkusermail(): Verifica se o username ou email já foram inseridos na bd.
- tipoProd(): Vê de que tipo é um produto.

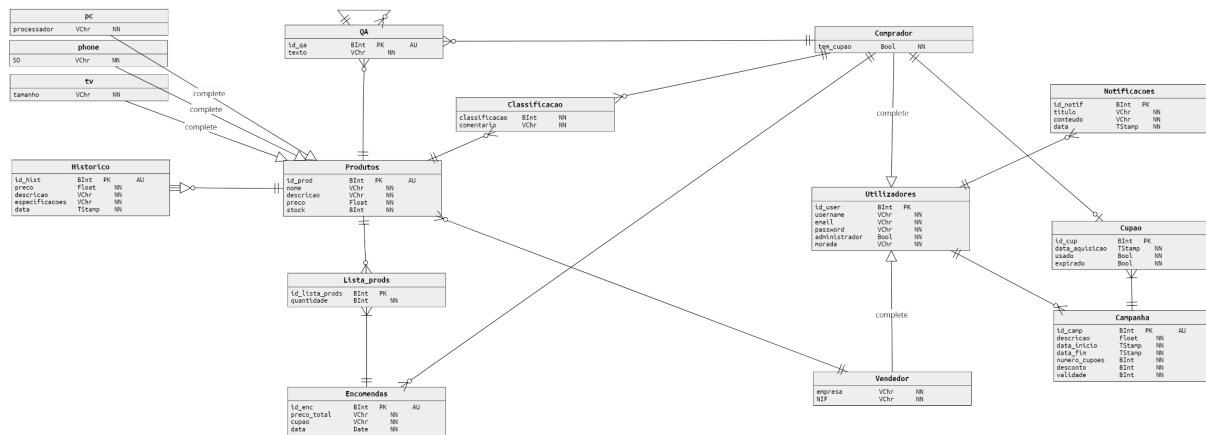
Divisão de trabalho

Elemento do grupo	Tarefas Realizadas	Horas
Bárbara Gonçalves	- Criar um produto	30
	- Deixar feedback	
	- Subscrever campanha	
	- Criar campanha	
Carlos Matos	- Registar utilizadores	25
	- Atualizar produto	
	- Comentário/pergunta	
Mariana Magueijo	- Autenticar utilizadores	28
	- Efetuar compra	
	- Consultar info produto	
	Estatísticas da campanha	

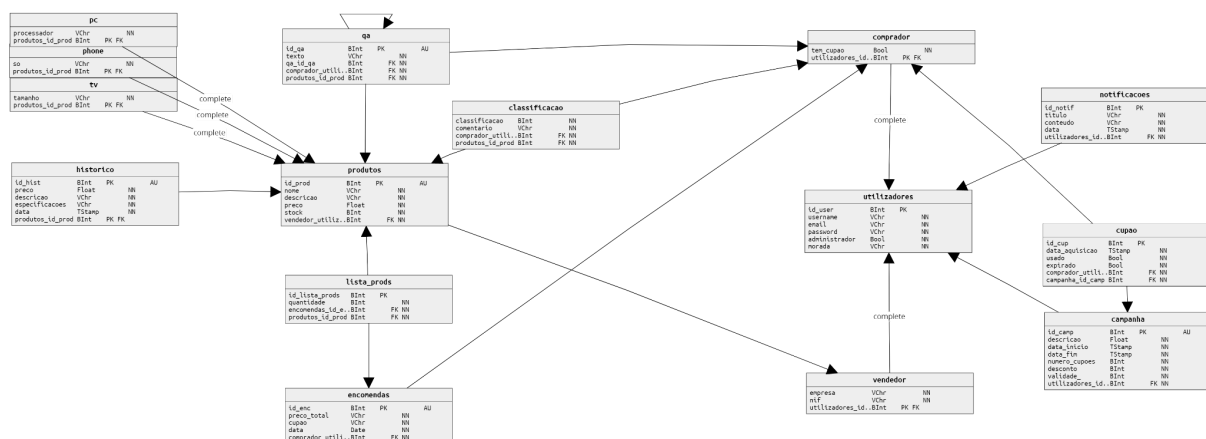


Diagramas

Diagrama Entidade-Relacionamento



Modelo de Dados Relacional



1 2



9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA



departamento
de engenharia informática
1995 – 2020