

Trabalho Prático de Base Dados 2021/2022



Rui Pedro Capelas Santos 2020225542
Tomás Bernardo Martins Dias 2020215701

Manual de instalação:

Requisitos necessário para o bom uso da API:

1. Python 3
2. PostgreSQL
3. Instalar as várias bibliotecas:
 - Flask
 - Psycopg2
 - PyJWT
4. App Postman (optional)

Para construir a base de dados abra pasta fornecida e aí abra a linha de comandos.

```
C:\Users\conta\OneDrive\Ambiente de Trabalho\native>psql -h localhost -p 5432 -U postgres
Password for user postgres:
psql (14.1)
WARNING: Console code page (850) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
```

Insira o comando acima e faça login com a pass default ou a que definiu na instalação do postgres.

De seguida crie uma nova base de dados e um novo utilizador conforme o exemplo abaixo:

```
postgres=# create database dbexemplo;
CREATE DATABASE
postgres=# create user exemplo1 password 'exemplo1';
CREATE ROLE
postgres=#
```

De seguida feche a consola e abra novamente e entre na base de dados com o utilizador que criou:

```
C:\Users\conta\OneDrive\Ambiente de Trabalho\native>psql -h localhost -p 5432 -U exemplo1 -d dbexemplo
Password for user exemplo1:
psql (14.1)
WARNING: Console code page (850) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

dbexemplo=>
```

Por fim, execute os scripts contendo as tabelas da base de dados um admin e os triggers.

```
dbexemplo=> \i scripts/bdproject.sql
CREATE TRIGGER
dbexemplo=> \i scripts/triggers1.sql
dbexemplo=> \i scripts/triggers2.sql
dbexemplo=> \i scripts/triggers3.sql
```

Crie também um primeiro administrador:

```
dbexemplo=> INSERT INTO utilizador(username, password) VALUES('admin', 'admin');

INSERT INTO administrador(salario, temposervico, nome,utilizador_id) VALUES(20000, 10, 'asdrubal',1);
```

Manual de Instruções:

Depois de concluir todos os passos do manual de instalação deve correr a aplicação pela linha comandos e enviar no formato "<nomedoexecutavel> <user> <password> <host> <port> <database>". De seguida deve abrir a aplicação postman na app (web caso tenha escolhido não instalar) e seguir estritamente o formato dos endpoints.

Existem 3 tipos de utilizadores os clientes, os vendedores e o administrador (DBMS), cada um possui permissão para efetuar endpoints específicos.

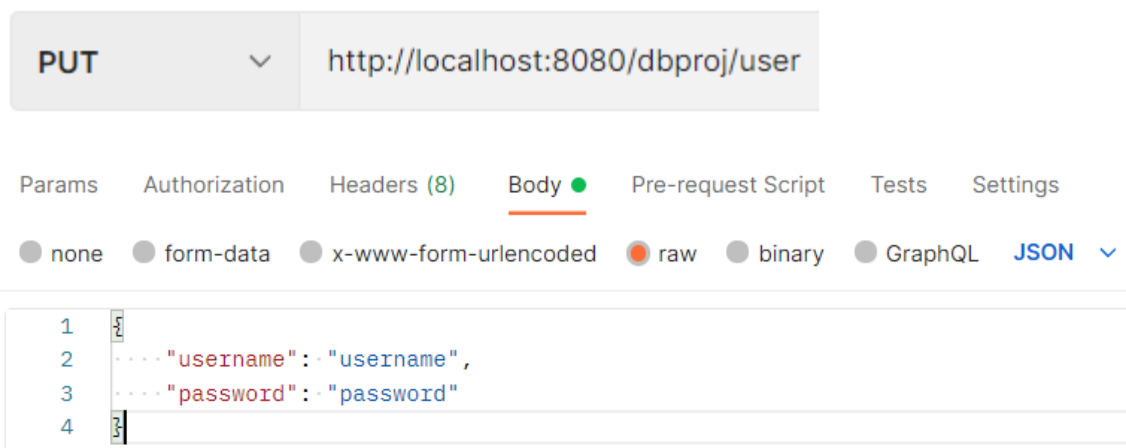
Endpoints permitidos:

-Clientes: login, compra, rating/feedback, comentário, resposta, informações genéricas, estatísticas, notificações.

-Vendedores: login, adicionar produto, update produto, notificações, informações genéricas, estatísticas.

-Administradores: login, adicionar utilizadores, informações genéricas, estatísticas.

Login (geral):



Após o login o utilizador devere receber um token de autenticação caso o login seja bem-sucedido que será necessário para realizar futuros endpoints caso este seja requerido.



Compras (clientes, token necessário):

GET	▼	localhost:8080/dbproj/order
-----	---	-----------------------------

```
{
  .... "cart": [[1,3]]
}
```

Para cada produto ser comprado deve ser especificado o id e a quantidade, caso um destes não esteja correto ou disponível respectivamente a operação é abortada. Após ser efetuada uma compra já é possível consultar as suas notificações.

Rating/feedback (cliente, token necessário):

POST	▼	localhost:8080/dbproj/rating/<produto_id>
------	---	---

```
{
  .... "rating": 1,
  .... "comment": "mau produto nao devia ser vendido"
}
```

Apenas após ser realizada uma compra de um produto é possível atribuir um rating a esse produto. Para isso deve ser especificado o id do produto. O rating deve ser um valor de 1 a 5.

Comentário/resposta (geral, token necessário):

POST	▼	localhost:8080/dbproj/questions/<product_id>/<parent_question_id>
------	---	---

POST	▼	localhost:8080/dbproj/questions/<product_id>
------	---	--

```
{
  .... "question": "isto é bom?"
}
```

É possível fazer um comentário acerca de um produto, como também é possível responder a comentários feitos por outros utilizadores. Para isso é necessário especificar o product_id, e no caso de estar a responder a um comentário indicar o seu id.

Adicionar produto (vendedor, token necessário):

<pre>{ "descricao": "tv boa e barata", "preco": "499.99", "stock": "50", "tamanho": "70", "definicao": "4k" }</pre>	<pre>{ "descricao": "telemovel bom e barato", "preco": "299.99", "stock": "50", "processador": "snapdragon", "memoria": "4gb ram", "bateria": "5000 mAh", "camara": "50mpx" }</pre>	<pre>{ "descricao": "pc bom e barato", "preco": "999.99", "stock": "50", "grafica": "rtx3090ti", "bateria": "dura 5 horas", "ram": "32gb", "tamanhoecra": "17p" }</pre>
---	---	---

POST ▼ localhost:8080/dbproj/product

Apenas os vendedores podem criar novos produtos, para isso é ter todos os campos acima descritos, caso contrario o produto não será adicionado.

Update produto (vendedor, token necessário):

<pre>{ "descricao": "tv boa e barata", "preco": "499.99", "stock": "50", "tamanho": "70", "definicao": "4k" }</pre>	<pre>{ "descricao": "telemovel bom e barato", "preco": "299.99", "stock": "50", "processador": "snapdragon", "memoria": "4gb ram", "bateria": "5000 mAh", "camara": "50mpx" }</pre>	<pre>{ "descricao": "pc bom e barato", "preco": "999.99", "stock": "50", "grafica": "rtx3090ti", "bateria": "dura 5 horas", "ram": "32gb", "tamanhoecra": "17p" }</pre>
---	---	---

PUT ▼ localhost:8080/dbproj/product/<produto_id>

Apenas os vendedores podem fazer update de novos produtos, para isso é necessário especificar o id do produto e respeitar o payload como anteriormente descrito na inserção de um produto.

Registo de novos utilizadores (administrador, token necessário):

POST



localhost:8080/dbproj/user

```
{
  ... "username": "vendedor1",
  ... "password": "password1",
  ... "nif": 255255,
  ... "email": "vendedor1@gmail.com",
  ... "cidade": "coimbra",
  ... "rua": "angola"
}
```

```
{
  ... "username": "administrador",
  ... "password": "admin",
  ... "salario": 2000,
  ... "temposervico": 10,
  ... "nome": "rui"
}
```

Qualquer pessoa se pode registar como cliente, mas apenas o administrador pode criar novos administradores/vendedores.

Registo de novos utilizadores(cliente):

POST



localhost:8080/dbproj/user

```
{
  ... "username": "user1",
  ... "password": "password1",
  ... "email": "user1@gmail.com",
  ... "cidade": "coimbra",
  ... "rua": "daniel de matos"
}
```

informações genéricas(geral):

GET

localhost:8080/dbproj/product/<produto_id>

estatísticas(geral):

GET

localhost:8080/dbproj/report/year

notificações (geral, token necessário):

GET

localhost:8080/dbproj/notification

Diagrama ER

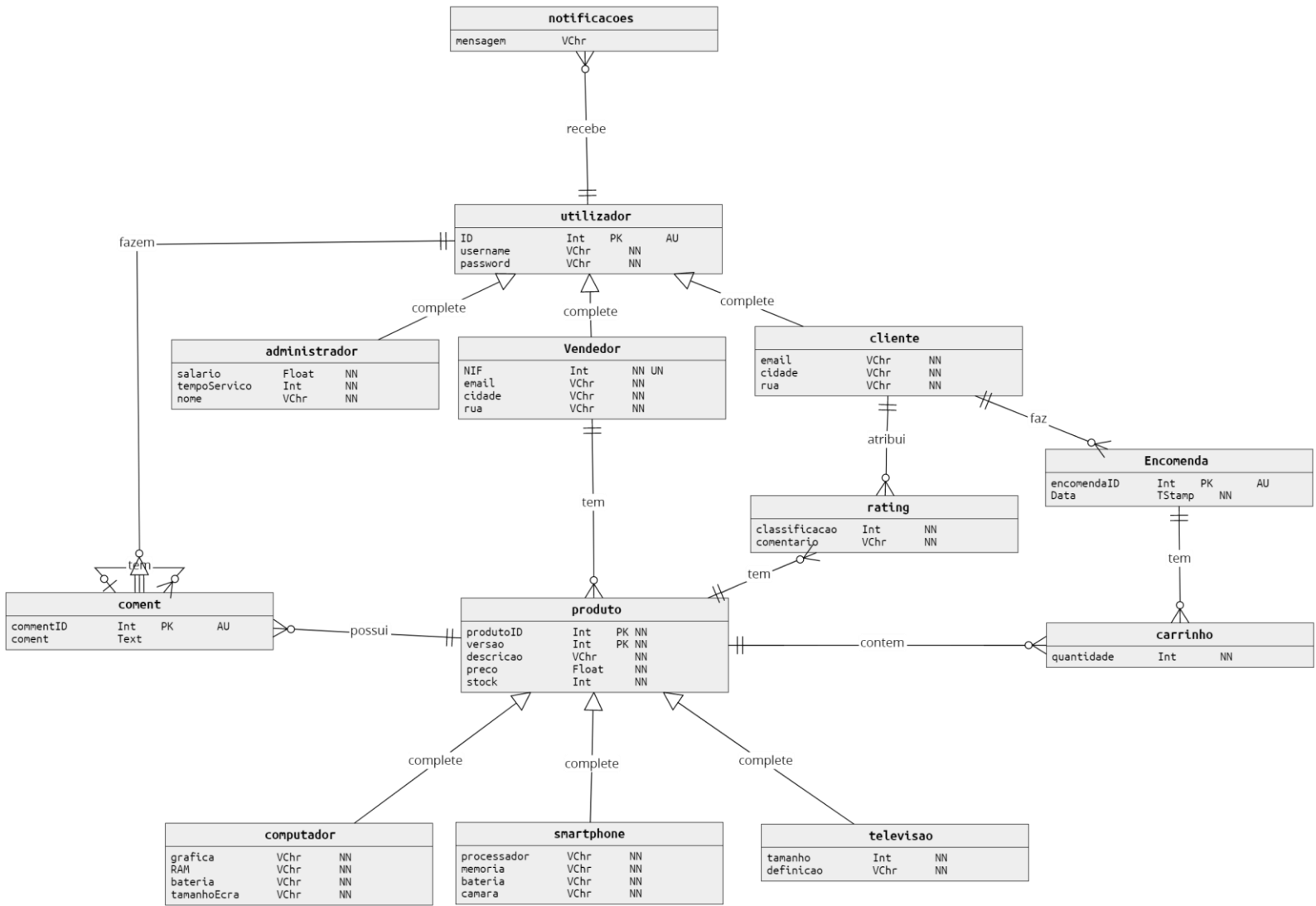
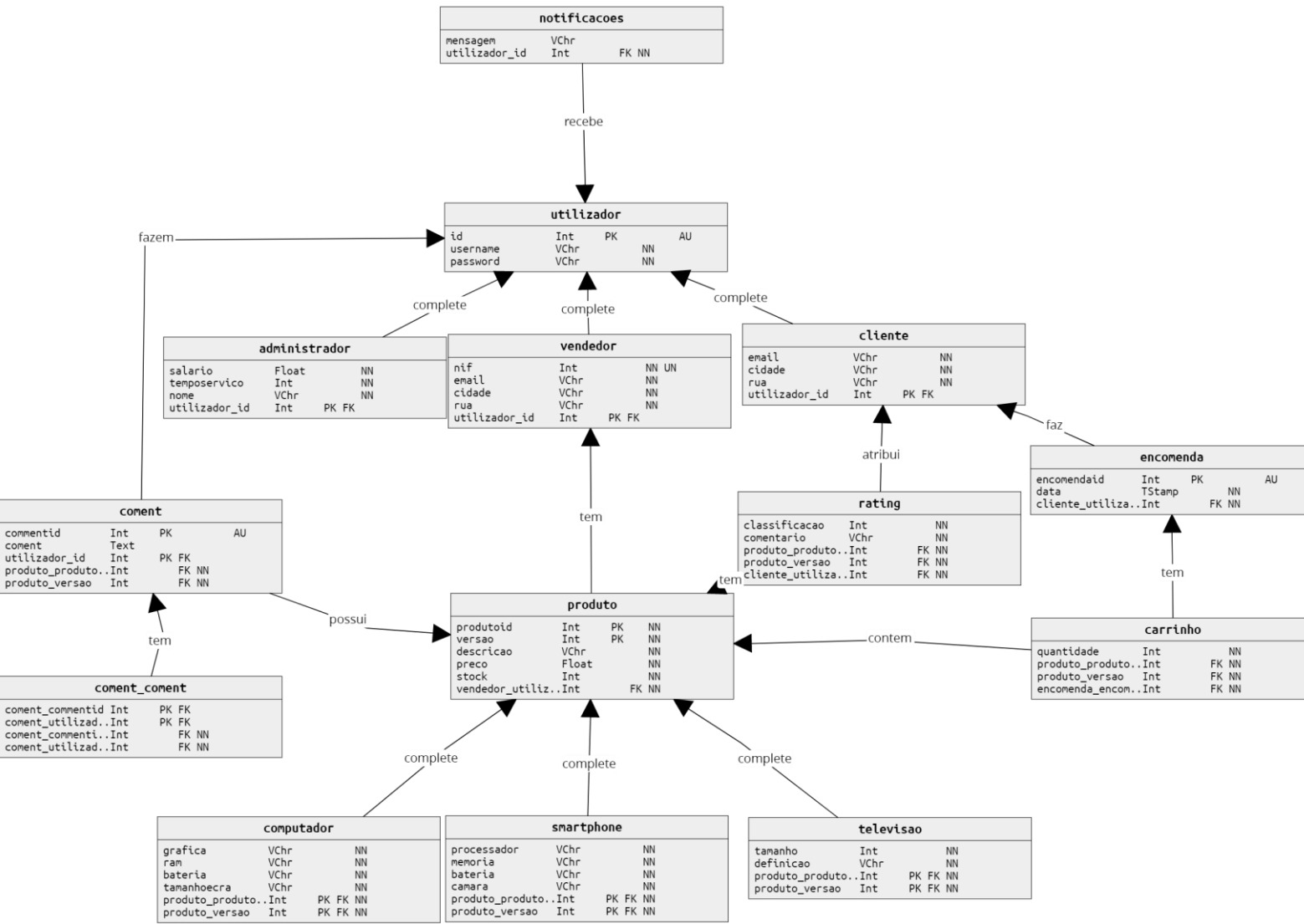


Diagrama Físico



Plano de trabalho

De forma a realizar este projeto dividimos de trabalho da seguinte forma:

1. Registo de utilizadores
2. Autenticação de utilizadores
3. Criação de um novo produto
4. Atualização dos detalhes de um produto
5. Efetuar a compra
6. Rating/feedback
7. Comentários/respostas
8. Informações genéricas de um produto
9. Estatísticas
10. Triggers
11. Relatório

Rui Santos: 3,4,5,9,10,11

Horas de trabalho: 23

Tomás Dias: 1,2,6,7,8,11

Horas de trabalho: 20