Projeto Bases Dados

Desenvolvido por:

1. João Silva – 2019217672
2. Joel Oliveira – 2019227468

Para este projeto não foi desenvolvido nenhum tipo de interface com o utilizador, nem por comando, nem gráfica, visto isto não estar abrangido no âmbito da cadeira.

Assim, para realizar/testar os pedidos aos *endpoints* do servidor da base de dados, foi utilizada a *API Client* do *Postman*. Estes *endpoints* foram recebidos por um *webserver* simulado na *localhost*, pela livraria *Flask* do Python, que depois executava as *querys* necessárias na própria base de dados.

Tanto a base de dados como o *webserver* executam em containers do *Docker*

Manual instalação:

Como referido, para testar as funcionalidades desenvolvidas neste projeto, são necessários uma série de programas.

* API REST POSTMAN, sendo apenas necessário o import da coleção de pedidos e do ambiente foi entregue num ficheiro *json*.
* DOCKER DESKTOP, onde são mantidos os containers. Para correr a imagem e criar os containers, é necessário correr no terminal o comando ‘*docker comopose up*’, a partir da pasta que tem presente o ficheiro ‘*docker-compose.yml*’ (no caso do que foi entregue, será na pasta ‘*Source Code and Scripts\projetoBD*’
* A instalação das livrarias necessárias para correr o webserver, e do próprio python é feita automaticamente, através de imagens disponibilizadas pelo Docker.

Manual utilizador:

* Para o utilizador se registar, terá de introduzir o username, email e palavra-passe. Após este registo apenas necessitará de se autenticar, introduzindo o id de utilizador e palavra-passe, para iniciar uma nova sessão (PUT).
* Antes de um utilizador qualquer leilão, será necessário que este introduza um item na base de dados, as informações todas sobre o item pedidas ao utilizador, exceto o id do artigo e o id do utilizador, que são gerados automaticamente (POST). Se o método usado for GET, não é necessário nenhum input por parte do utilizador. Para este caso, todos os itens já presentes na base de dados serão listados e apresentados ao utilizador.
* Após o item estar introduzido, é possível criar um leilão com um item que pertença ao utilizador (POST). O utilizador insere todas as informações sobre o leilão (incluindo a descrição), exceto as que dizem respeito ao id do item e do leilão (pois ambos são gerados automaticamente)
* Para fazer uma licitação (GET), o utilizador precisa de, ao fazer o pedido, introduzir o montante a licitar (maior que o montante atual do leilão), e o id do leilão.
* É possível verificar os detalhes de um leilão específico (GET), dado o id do leilão pelo utilizador ao fazer o pedido
* É possível verificar os leilões em que um utilizador está envolvido (GET), sendo o id do utilizador o único parâmetro necessário ao fazer o pedido
* É possível alterar o título e a descrição do leilão (PUT). O utilizador ao fazer o pedido especifica o novo título e a nova descrição, substituindo ambos mas guardando a antiga descrição na base de dados.
* Um leilão pode ser procurado através da sua descrição atual (GET), sendo esta o único parâmetro enviado pelo utilizador ao fazer o pedido
* Cada leilão tem um mural de mensagens, sendo que para enviar uma mensagem (POST), o utilizador insere, ao fazer o pedido, os conteúdos da mensagem e o id do leilão ao qual esta pertence. A informação da data e do id do leilão é atribuída automaticamente.
* O leilão termina quando o tempo atual coincidir com o tempo limite introduzido pelo vendedor aquando da criação do leilão (GET).

Distribuição de trabalho:

João:

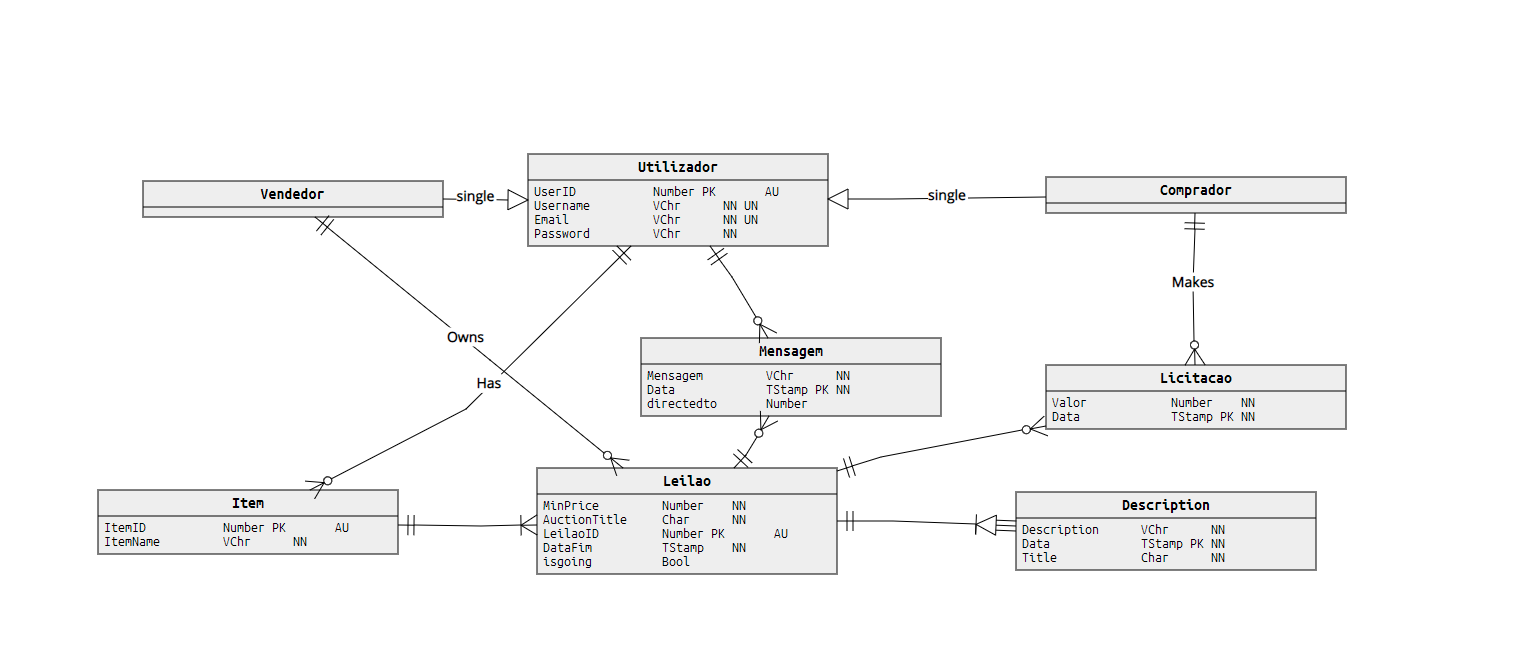
* CRIAR ITEM
* CRIAR LEILÃO
* LIST TODOS LEILÕES
* EDITAR LEILÃO
* MSG MURAL DO LEILÃO
* TÉRMINO DE LEILÃO
* ENTREGA DE MSG A USERS
* EFETUAR LICITAÇÃO
* TÉRMINO DE LEILÃO

Joel:

* REGISTOS DE USER: Joel
* AUTENTICAÇÃO DE USER: Joel
* LIST ITEMS: Joel
* PESQUISAR LEILÕES
* CONSULTAR LEILÃO
* LIST LEILÕES COM USER
* EFETUAR LICITAÇÃO
* ENTREGA DE MSG A USERS
* NOTIFICAR LICITAÇÃO ULTRAPASSADA
* TÉRMINO DE LEILÃO

Uma imagem com texto, céu, captura de ecrã

Descrição gerada automaticamenteDiagrama Relacional

Diagrama ER