

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

RELATÓRIO DO PROJETO

Leilões Online

Bases de Dados



Tiago Oliveira Nº 2019219068
Simão Monteiro Nº 2019215412
Marco Pais Nº 2019218981

Ano Letivo 2020/2021
Maio de 2021

Índice

- 1) Manual de utilizador
- 2) Manual de instalação
- 3) Diagrama ER
- 4) Plano de desenvolvimento
- 5) Descrição da construção da aplicação
- 6) Conclusão

Manual de Utilizador

Importar para o postman a versão que pretende utilizar e iniciar a API executando o comando “`mkdir -p python/app/logs`” e o comando “`docker-compose -f docker-compose-python-psql.yml up --build`” na linha de comandos, colocando a mesma na pasta onde se situa os ficheiros da aplicação.

Funções no Postman

Versão Utilizador:

- Registo: Username, Email e Password no body da função;
- Autenticação: Username e Password para se autenticar e obter o token;

ATENÇÃO: Colocar o token nas chamadas subsequentes em cada função no parâmetro access-token do header;

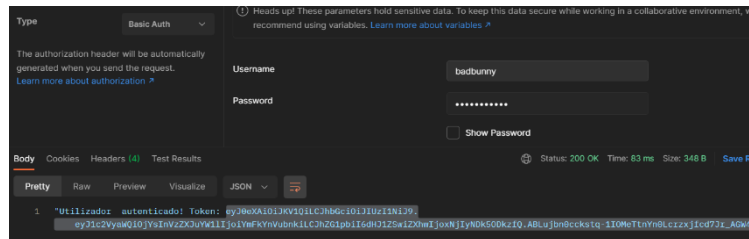


Fig1. Obtenção do token

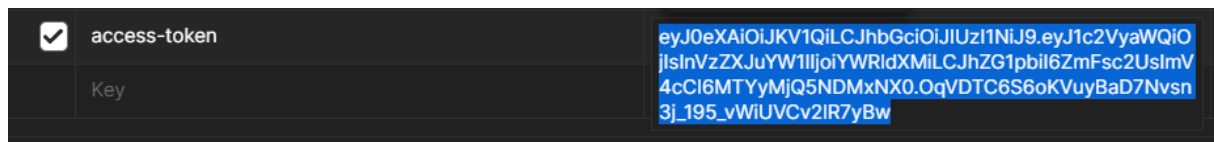


Fig2. Utilização do token

- Criar novo leilão: Especificações do leilão no body da função;
- Listar leilões existentes: Simplesmente executar função clicando no send;
- Pesquisar leilões: Barcode do produto ou descrição no url do postman;
- Consultar detalhes de um leilão: Barcode do produto no url do postman;
- Listar atividade no leilão: Executar função clicando no send;
- Efetuar licitação num leilão: Barcode do produto e valor da licitação no body da função;
- Editar propriedades de um leilão: Título, descrição e produto_barcode no body da função;
- Escrever mensagens no mural de um leilão: Mensagem e produto_barcode no body.

Versão Administrador:

- Cancelar um leilão: Produto barcode inserido no url;
- Banir um utilizador: ID do user inserido no url;
- Obter estatísticas: Executar clicando no sem.

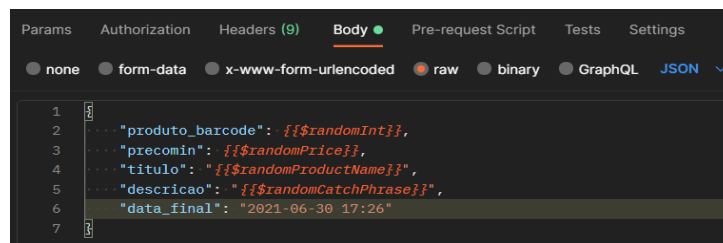


Fig3. Body do Postman

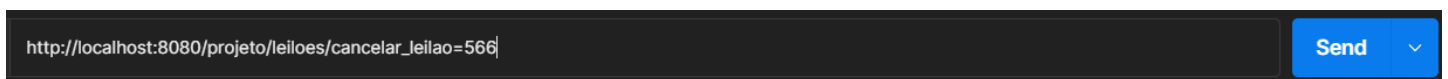


Fig4. URL do postman

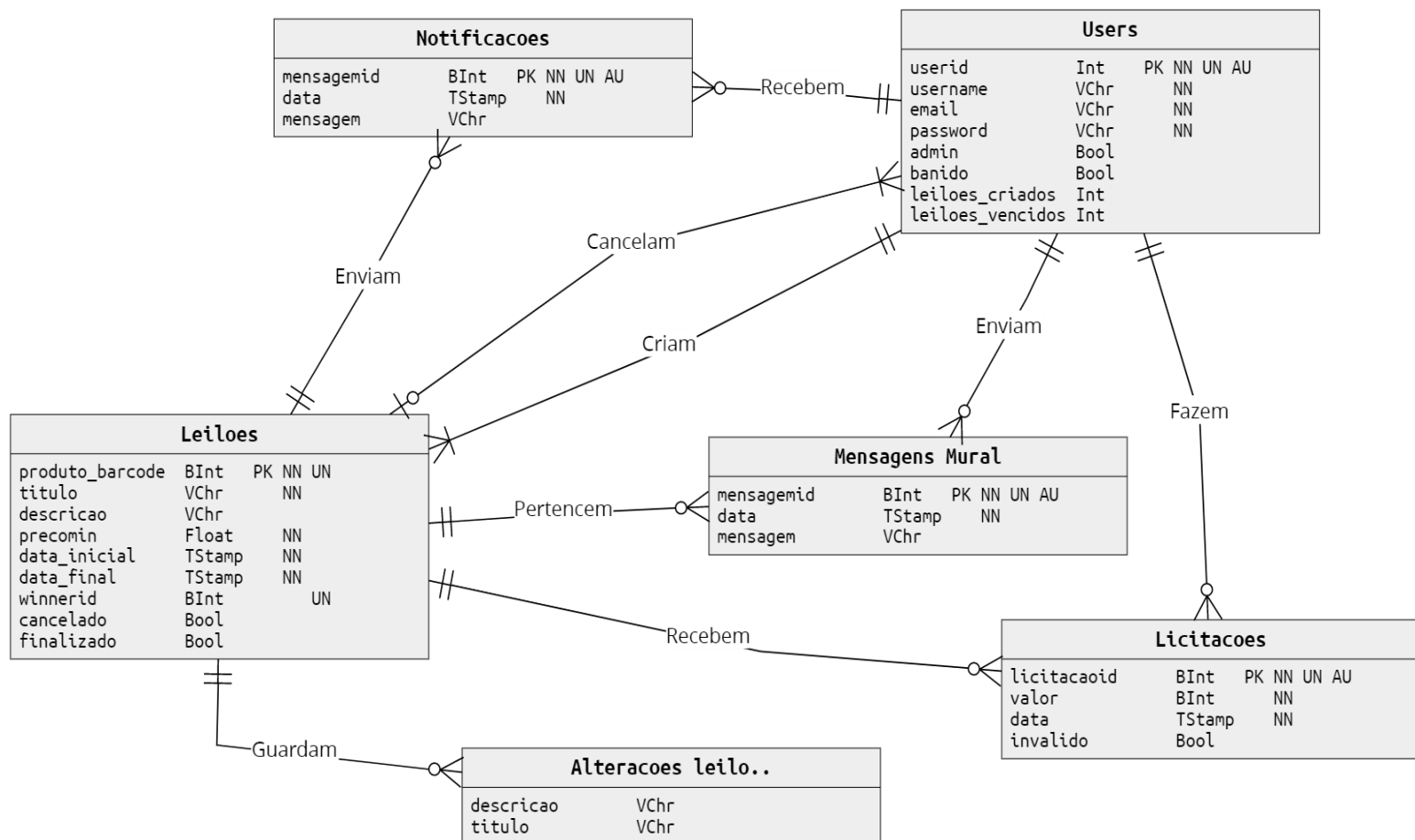
Manual de Instalação

Requisitos:

- Docker
- Sistema Operativo: Windows 10/Linux/macOS
- Postman (Versão Web ou App)

Instalação da aplicação Docker pelo <https://www.docker.com/products/docker-desktop>, se pretender instalar o postman será pelo link <https://www.postman.com/downloads/>. O comando “`docker-compose -f docker-compose-python-psql.yml up --build`” deve ser executado no terminal de cada sistema operativo.

Diagrama ER



Descrição da Construção da Aplicação

De seguida, faremos uma breve explicação sobre as diferentes funcionalidades implementadas na API.

Registo de utilizadores: Para o registo de utilizadores efetuamos um insert na base de dados das informações fornecidas sendo que a palavra-passe é encriptada na inserção.

Autenticação de utilizadores: Recebemos as credenciais e executamos um select para verificar a existência do utilizador e a validade da palavra-passe. É ainda devolvido um token para a futura execução das diferentes funções.

Criar um novo leilão: Apenas disponível para utilizadores normais. Recebemos as informações através do postman e inserimos o novo leilão fazendo as verificações necessárias.

Listar todos os leilões existentes: Seleccionamos todos os leilões existentes e devolvemos a lista de leilões.

Pesquisar leilões existentes: Recebe-se o barcode de um produto ou a descrição de um leilão e é escolhido o método de procura.

Consultar detalhes de um leilão: É recebido o barcode de um produto e é percorrida a tabela de leilões retornando apenas o leilão pretendido com todas as informações.

Listar todos os leilões em que o utilizador tenha atividade: Recebe-se o ID do utilizador através do token de autenticação e a partir dele obtemos as suas licitações e os seu leilões devolvendo os seus detalhes.

Efetuar uma licitação num leilão: Recebemos o valor e o barcode do produto e é feita a pesquisa de igual forma às opções anteriores, inserindo na tabela licitações com as devidas verificações. As licitações apenas estão disponíveis para os utilizadores normais.

Editar propriedades de um leilão: Apenas o criador do leilão o pode fazer. Recebemos o título, a descrição e o produto barcode alterando assim o leilão pretendido, após as devidas verificações, com as novas informações. Guardamos também as versões anteriores numa tabela.

Escrever mensagens no mural de um leilão: Recebe-se a mensagem e o barcode do produto necessários para a inserção de uma mensagem na tabela mensagens_mural, sendo ativado um trigger que escreve notificações para todos os licitadores e dono desse leilão.

Entrega imediata de notificações ao utilizador: É efetuada por vários triggers que são executados quando ocorre um certo evento.

Notificação de licitação ultrapassada: A notificação é enviada por um trigger informando o utilizador ultrapassado e o dono do leilão.

Finalização do leilão na data prevista: Procura todos os leilões já acabados, altera o seu estado e notifica os vencedores e o criador do término do mesmo.

Cancelar um leilão: Apenas disponíveis para administradores. Recebe o barcode de um produto e efetua o cancelamento do devido leilão com as verificações necessárias e ativa um trigger que envia notificações aos participantes.

Banir permanentemente um utilizador: Apenas disponíveis para administradores. Recebe o ID do utilizador e bane o devido utilizador com as verificações necessárias. Ativa por fim um trigger que invalida todas as suas licitações e cancela os seus leilões não permitindo mais o acesso às funcionalidades.

Obter estatísticas de atividade na aplicação: Apenas disponíveis para administradores. Seleciona nas tabelas as estatísticas pretendidas e são devolvidas.

Para a alteração das tabelas é usado ROW-LEVEL LOCKS quando são dados os UPDATE's e TABLES LOCKS quando são dados os INSERT's. Isto é feito para resolver os possíveis problemas de concorrência.

Plano de Desenvolvimento

O desenvolvimento da aplicação foi feito ao longo de 3 semanas em que na 1ª e 2ª semana foram efetuadas as implementações das funcionalidades em Python e a implementação das diferentes funções, triggers e etc em PL/PGSQL.

Na última semana foram construídas as funcionalidades no postman e feitas as correções de erros, testes e revisão do código.

As tarefas foram realizadas sem divisão e sempre em conjunto via Live Share no Visual Studio Code, sendo que todos os elementos do grupo participaram na integra na construção do projeto.

Em cada semana foram despendidas cerca 13 horas por parte de cada elemento do grupo, sendo que o tempo total despendido no projeto foram cerca de 40 horas por cada elemento.

Conclusão

Neste projeto conseguimos aprofundar os nossos conhecimentos adquiridos nas aulas teóricas, teórico-práticas e práticas-laboratoriais sobre a matéria lecionada. Ajudou-nos também a desenvolver as nossas capacidades de implementação e programação de base de dados e API's.