

Base de Dados

# LEILÕES ONLINE

Beatriz Abrantes Abreu Madeira – 2018280169

uc2018280169@student.uc.pt – PL6

Cláudia Filipa Peres Saraiva de Campos – 2018285941

cfcampos@student.dei.uc.pt – PL6

Dário Filipe Torres Félix –2018275530

dario@student.dei.uc.pt – PL6



UNIVERSIDADE D  
COIMBRA



# Índice

---

<b>Manual de Instalação</b>	<b>3</b>
<b>Manual do Utilizador</b>	<b>3</b>
<b>Detalhes da Implementação</b>	<b>6</b>
<b>Plano de Desenvolvimento</b>	<b>7</b>

## Manual de Instalação

Para usufruir do programa, é necessário apenas executar o ficheiro **docker-compose-python-psql** fornecido no ficheiro .zip entregue. Sugerimos que seja utilizado o programa postman para a utilização do serviço HTTP REST, pois será mais fácil a utilização das funcionalidades, nomeadamente tokens de autenticação. Poderá ser importado o ficheiro JSON fornecido no ficheiro .zip entregue.

Para informações mais detalhadas, poderá ler o README em anexo.

## Manual do Utilizador

A interação com a aplicação é feita através da API desenvolvida, especificada no enunciado do projeto.

Após a autenticação do utilizador através do username e password, é gerado um **token** de autenticação que é enviado no header de todos os *requests* efetuados (posteriores ao login). O nome do header é **Authorization** e a colocação do token no header do request é **manual**.

Caso seja utilizado o programa postman, o token deverá ser colocado no local indicado na aba **Authorization** e os dados de input devem ser colocados no local indicado na aba **Body**. Em anexo, enviamos o ficheiro com as configurações do postman.

### Dados estáticos

Na tabela de artigos existem já alguns elementos inseridos previamente:

Código	Nome	Descrição
1234567890000	Livro Velho	Um livro bastante velho
9876543210987	Mota	Uma mota qualquer
0123456789012	Carro	a um bom preço de mercado
0000000000001	OVNI	sim, um ovni
6666666666666	alma e dignidade	e porque nao?
9999999999999	5G	ANACOM - Atribuição de direitos de utilização de frequências nas faixas dos 700 MHz, 900 MHz, 2,1 GHz, 2,6 GHz e 3,6 GHz.

## API Utilizada

Para os casos não especificados no enunciado foram utilizados os seguintes endpoints:

- Listar todos os leilões em que o utilizador tenha atividade
  - **GET** `http://localhost:8080/dbproj/leiloes/atividade`
  - Devolve
    - Em caso de sucesso: [ { "leilaoId": 1, "descricao": "desc1" }, { "leilaoId": 2, "descricao": "desc2" } ]
    - Em caso de erro: { "erro" : errorCode }
- Escrever mensagem no Mural de um Leilão
  - **POST** `http://localhost:8080/dbproj/mensagem/<leilao_id>`
  - Recebe:
    - Em caso de sucesso: { "mensagem": "msg1" }
  - Devolve
    - Em caso de sucesso: "Sucesso"
    - Em caso de erro: { "erro" : errorCode }
- Entrega imediata de mensagens a utilizadores
  - As mensagens são geradas automaticamente pela aplicação sendo que podem ser listadas no endpoint indicado
  - **GET** `http://localhost:8080/dbproj/notificacoes/todas`
  - Devolve
    - Em caso de sucesso: "notificacao\_leilao": [ { "leilaoId": "1", "data": "2021", "mensagem": "msg" }, ... ],  
"notificacao\_mensagem": [ { "leilaoId": "1", "data": "2021", "pessoa": "Ana", "mensagem": "msg" }, ... ],  
"notificacao\_licitacao": [ { "leilaoId": "1", "data": "2021", "pessoa": "Ana", "mensagem": "msg" }, ... ] }
    - Em caso de erro: { "erro" : errorCode }
- Notificação de licitação ultrapassada e Entrega imediata de notificações a utilizadores NOVAS
  - As notificações são geradas automaticamente pela aplicação mas podem ser listadas no endpoint seguinte
  - **GET** `http://localhost:8080/dbproj/notificacoes/novas`

- Devolve:
    - Em caso de sucesso: { "notificacao\_leilao": [ { "leilaoId": "1", "data": "2021", "mensagem": "msg" }, ... ], "notificacao\_mensagem": [ { "leilaoId": "1", "data": "2021", "userId": "Ana", "mensagem": "msg" }, ... ], "notificacao\_licitacao": [ { "leilaoId": "1", "data": "2021", "userId": "Ana", "mensagem": "msg" }, ... ] }
    - Em caso de erro: - { "erro" : errorCode }
- Término do Leilão na data, hora e minuto marcados
  - **GET** http://localhost:8080/dbproj/leilao/<leilao\_id>/terminar
  - Recebe: id do leilão
  - Devolve:
    - Em caso de sucesso: { "userId" : "123", "valor": 15.0 }
    - Em caso de nao existirem licitações associadas ao leilão: { "sucesso": "ainda não existem licitações associadas a este leilao" }
    - Em caso de erro: { "erro" : errorCode }
- Um administrador pode cancelar um Leilão
  - **GET** http://localhost:8080/dbproj/leilao/<leilao\_id>/abortar
  - Devolve:
    - Em caso de sucesso:- { "sucesso": "msg" }
    - Em caso de erro: { "erro" : errorCode }
- Um administrador pode banir permanentemente um utilizador
  - **GET** http://localhost:8080/dbproj/user/<user\_id>/banir
  - Devolve
  - Em caso de sucesso: { "sucesso": "msg" }
  - Em caso de erro: { "erro" : errorCode }
- Um administrador pode obter estatísticas de atividade na aplicação
  - **GET** http://localhost:8080/dbproj/estatisticas
  - Devolve:-
    - Em caso de sucesso:{ "topUserIdComLeiloesCriados": [ "user1", "user2", ... ], "topUserIdComLeiloesGanhos": [ "user1", "user2", ... ], "totalLeiloesUltimos10Dias": 3 }
    - Em caso de erro: { "erro" : errorCode }

# Detalhes da Implementação

## Detalhes de funcionalidades

- Registo de utilizadores

No registo o utilizador insere os seus dados: username, email e password, sendo apenas aceites contas cujo email ainda não está registado.

Após o registo ser concluído é gerado um **uuid** que identifica o user no sistema.

Escolhemos usar um uuid e não um inteiro auto incrementado, pois o serial , ao contrário do uuid, não é transaction safe, logo pode levar a inconsistências nas tabelas.

- Autenticação de utilizadores

Na autenticação de utilizadores, é gerado um token que será utilizado em todas as operações que este pretender realizar. O token é enviado através do header (**Authorization**) do request sendo a sua colocação no header manual.

Como a validade dos tokens de autenticação são temporários, será necessária uma nova autenticação do utilizador sempre que o token fique inválido.

- Criar um novo Leilão

Quando um novo Leilão é criado, são inseridos os dados respetivos no sistema, sendo o título e descrição do leilão guardados numa tabela à parte do resto dos dados. É devolvido um **uuid** que identificará o leilão nas operações necessárias. Utilizamos um uuid pelas razões apresentadas acima.

## Transações

- Para as transações utilizamos os comandos:

- BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
- BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED

- O primeiro comando é utilizado para assegurar a consistência dos dados que estamos a utilizar. Assim, enquanto uma transação está a executar, não tem acesso às alterações feitas às tabelas, ou seja, independentemente dos dados serem alterados externamente, os dados utilizados por essa transação são sempre os mesmos.

- O segundo comando, ao contrário do primeiro, permite que enquanto a transação está a executar os dados se alterem, pois a alteração dos dados a meio da transação não causa inconsistências nas tabelas. Assim podemos obter logo os novos dados sem comprometer o funcionamento do programa.

### Controlo de Concorrência:

- Na maioria dos casos, são utilizados locks implícitos, isto é, locks geridos automaticamente pela base de dados porque as atualizações realizadas envolvem poucas tabelas diferentes em que haja dependência de dados.
- No caso de os dados utilizados envolverem várias linhas e várias tabelas em que há dependência de dados, é usado um lock explícito onde são bloqueadas todas as tabelas/linhas envolvidas.

## Plano de Desenvolvimento

Tarefa	Duração (min)	Responsável
Criação/inserção de dados iniciais estáticos (artigos, administradores ...)	60	Dário
Registo e autenticação de utilizadores	120	Cláudia
Listar , pesquisar e consultar detalhes de leilões	300	Beatriz
Criar um novo leilão,Licitar,	120	Dário
Editar propriedades de um leilão	120	Cláudia
Escrever mensagens no mural, entrega imediata de mensagens	240	Beatriz
notificação de licitação ultrapassada, Término do leilão	120	Dário
Índices	15	Dário
Cancelamento de um leilão, banir permanentemente um utilizador	360	Cláudia
Obter estatísticas	120	Dário
Documentação	120	Beatriz
Testes	100	Dário
Revisão geral do código fonte	200	Dário