



Projeto de Bases de Dados

Sistema de Leilões

Trabalho realizado por:

- Pedro Martins (N.º 2019216826)
- António Correia (N.º 2019216646)

Índice:

Projeto de Bases de Dados.....	1
Sistema de Leilões	1
Índice:.....	2
Introdução ao projeto e trabalho despendido:	3
ER Diagram:	4
Manual de instalação:.....	5
Manual de utilizador:.....	6

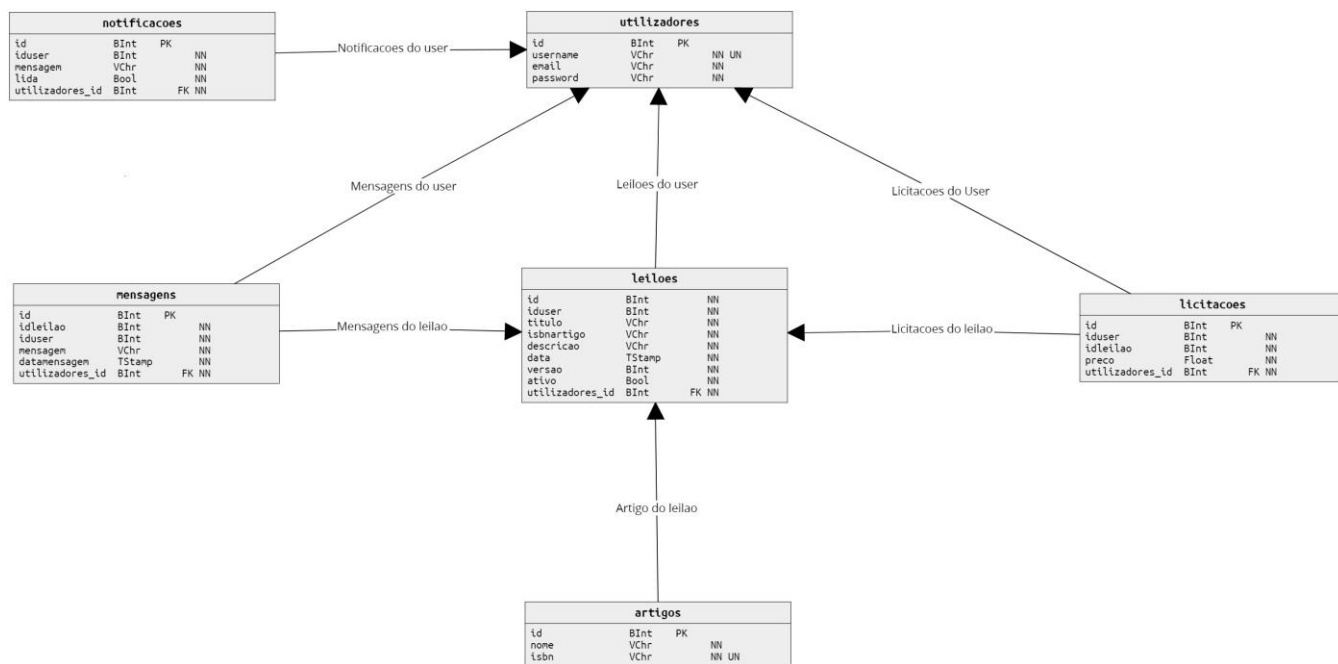
Introdução ao projeto e trabalho despendido:

Este projeto tem o intuito de desenvolver um sistema de leilões com recurso aos conhecimentos adquiridos na cadeira de Bases de Dados. Para isso iremos recorrer a um sistema de API REST de forma a aceder ao nosso sistema através de pedidos do tipo HTTP. Como linguagem de programação para implementar a conexão à base de dados recorreremos ao Java, com recurso ao Spring Boot de forma a poder ter uma estrutura já previamente fabricada. Na base de dados recorreremos ao PSQL.

De forma a estruturar melhor o nosso projeto e tal como foi abordado na última defesa intermédia, todas as semanas reuníamos às quartas-feiras de forma a poder discutir novas ideias para o projeto, assim como adiantar parte do código à medida que adquiríamos mais conhecimentos nas aulas. Consideramos que o envolvimento no projeto de cada membro rondou as 50 horas. Abaixo encontra-se uma tabela detalhando os aspetos abordados por cada um de nós.

Pedro Martins	António Correia
Desenvolvimento do relatório.	Criação das tabelas.
Preparação da apresentação.	Implementação com o Spring Boot.
Auxílio na implementação das tabelas e respetivo código de integração / ER Diagram.	Desenvolvimento do código de integração para as diferentes tarefas a realizar na base de dados.
Código PSQL para obter/atualizar informação nas tabelas presentes na base de dados.	Código PSQL para obter/atualizar informação nas tabelas presentes na base de dados.

ER Diagram:



*ER Diagram pode ser analisado mais em detalhe no ficheiro "ER_Diagram" fornecido em anexo.

Manual de instalação:

Primeiramente, é necessário que o utilizador possua instalado os seguintes produtos:

- Docker.
- Acesso ao postman.
- IDE para Java (Como recomendação pode ser utilizado o IntelliJ).

De seguida, é necessário abrir o código Java com o IDE a utilizar, clicar no botão “Build” no canto superior direito e de seguida clicar em “Run”. Deve instalar a Docker e, após ter a mesma devidamente configurada, iniciar o postgres e pgadmin para assim poder correr a base de dados. Deve de seguida executar os comandos fornecidos no ficheiro “dbcreationscript.txt” adjacente ao projeto, de forma a ter as tabelas previamente criadas.

Finalmente, para proceder aos testes, deve recorrer ao uso do Postman, acedendo a postman.com, para assim poder usufruir do sistema de leilões. Para obter uma lista detalhada de todos os comandos disponíveis consulte o manual de utilizador presente neste relatório.

Manual de utilizador:

Primeiramente, o utilizador deve seguir a lista de passos presentes anteriormente no manual de instalações, de forma a poder executar e correr o projeto sem quaisquer problemas.

Com todo o software já a correr, deve aceder ao postman de forma a executar os comandos acessíveis. Entre eles temos:

- [Registo de utilizadores](#): Permite registar um novo utilizador na base de dados.

“POST http://localhost:8080/dbproj/user”

Também deve incluir os atributos seguintes:

{username, email, password}

- [Autenticação de utilizadores](#): Permite a um utilizador existente autenticar-se ao servidor.

“PUT http://localhost:8080/dbproj/user”

Também deve incluir os atributos seguintes:

{username, password}

Após a autenticação ser efetuada, irá ser atribuído ao utilizador um token de autenticação. Para introduzir o token o utilizador deve ir no postman>Authorization>BearerToken e copiar o token para a caixa de texto ao lado ao tentar executar um dos requests abaixo. pode-lhe ser apresentada as seguintes mensagens de erro:

- ♦ **Não adicionou nenhum Bearer Token**: O utilizador não apresentou nenhum token.
- ♦ **Token inválido, faça login para verificar o seu token**: O utilizador apresentou um token inválido.
- ♦ **Token expirado, faça login novamente para ter um novo token**: O token do utilizador já foi válido, mas expirou. Todos os tokens têm a duração de uma hora.

- [Criar um novo leilão](#): Permite ao utilizador criar um novo leilão.

“POST http://localhost:8080/dbproj/leilao”

Também deve incluir os atributos seguintes:

{IDartigo, precolnicial, título, descrição, dataFinal}

Os seguintes erros são possíveis:

- **Não existe nenhum artigo com esse ID:** O ID do artigo a adicionar ao leilão não existe na base de dados.
- **Preço mínimo mal formatado:** O valor atribuído ao preço mínimo não tem o formato correto (número inválido).
- **Data para o fim do leilão mal formatada:** A data introduzida não tem o formato correto.

Em caso de sucesso:

Leilão criado!

- **Listar leilões:** Permite ao utilizador obter uma lista de todos os leilões existentes.

“GET http://localhost:8080/dbproj/leiloes”

- **Pesquisar leilão:** Permite ao utilizador pesquisar por um leilão já existente. Para isso tem duas opções para substituir a {keyword} abaixo presente, ou pesquisa por ISBN ou pela descrição do leilão. Pode ser apresentado mais do que um resultado dependendo do critério ser idêntico em mais do que um leilão.

“GET http://localhost:8080/dbproj/leiloes/{keyword}”

- **Consultar detalhes de leilão:** Permite ao utilizador consultar todos os detalhes de um determinado leilão. O {leilaoid} deve ser substituído pelo ID do leilão a pesquisar.

“GET http://localhost:8080/dbproj/leilao/{leilaoid}”

Os seguintes erros são possíveis:

- **ID não é um número:** O ID fornecido não corresponde a um número (número inválido).
- **Não existe nenhum leilão com esse id:** O ID fornecido não corresponde a nenhum leilão que esteja a decorrer.

- **Listar leilões de um utilizador:**

“GET http://localhost:8080/dbproj/leiloes/meusleiloes”

Os seguintes erros são possíveis:

- **Não tem atividade em nenhum leilão:** O utilizador em questão não criou/licitou em nenhum leilão no momento deste GET.

- **Licitar num leilão:** O utilizador tenta ficar com um artigo de um leilão, atualizando o preço para um valor superior ao atual. O {leilaoid} deve ser substituído pelo ID do leilão a licitar.

“POST http://localhost:8080/dbproj/licitar/{leilaoid}”

Também deve incluir o atributo seguinte

{precoAtualizado}

Os seguintes erros são possíveis:

- **Esse leilão já se encontra encerrado:** O leilão a atualizar já encerrou.
- **Já existe uma licitação maior ou igual à que tentou fazer:** O preço atribuído à licitação não é superior ao preço atual do leilão.
- **Licitação mal formatada:** O valor introduzido na licitação não é um número válido.
- **Não existe nenhum leilão com esse id:** O ID fornecido não corresponde a nenhum leilão que esteja a decorrer.

Em caso de sucesso:

Licitação adicionada!

- **Editar um leilão:** Permite ao utilizador editar as configurações de um leilão. O {leilaoid} deve ser substituído pelo ID do leilão a atualizar.

“PUT http://localhost:8080/dbproj/leilao/{leilaoid}”

Os seguintes erros são possíveis:

- **Não existe nenhum leilão com esse id:** O ID fornecido não corresponde a nenhum leilão que esteja a decorrer.
- **ID não é um número:** O ID fornecido não corresponde a um número (número inválido).
- **Escrever mensagem no mural de um leilão:** Permite ao utilizador escrever uma mensagem e associá-la a um leilão.

“POST http://localhost:8080/dbproj/mensagem/{leilaoid}”

Também deve incluir o atributo seguinte

{mensagem}

Os seguintes erros são possíveis:

- **Não existe nenhum leilão com esse id:** O ID fornecido não corresponde a nenhum leilão que esteja a decorrer.
- **ID não é um número:** O ID fornecido não corresponde a um número (número inválido).

Notas a reter:

- A data a inserir tem sempre o formato "AAAA-MM-DD HH-MM-SS".
- Relembrar a inserção do token de autenticação fornecido para **todos** os requests a executar. Para isso deve aceder a Authorization e de seguida seleccionar Bearer Token e colocá-lo lá.