

**Faculdade de Ciências e Tecnologias da Universidade de
Coimbra
Licenciatura em Engenharia Informática**

Base de Dados

Relatório do Projeto

2020/2021



Joana Simões, N° 2019217013
Samuel Carinhas, N° 2019217199

Índice

Manual de instalação	2
Requisitos	2
Configuração da base de dados.....	2
Configuração da chave secreta dos <i>tokens</i> de autenticação	2
Execução	2
Manual de utilizador.....	3
Funcionalidades e os seus <i>endpoints</i>	3
Registar novos utilizadores	3
Autenticar utilizadores.....	3
Criar um leilão	3
Listar todos os leilões	3
Pesquisar um leilão	3
Consultar informação de um leilão.....	4
Listar leilões onde o utilizador tem atividade.....	4
Licitar.....	4
Editar informação do leilão	4
Ver notificações	5
Terminar leilões	5
Diagrama ER	6
Diagrama Relacional.....	7
Plano de desenvolvimento	8

Manual de instalação

Requisitos

Esta aplicação tem as seguintes dependências:

- Postgres (<https://www.postgresql.org/download/>)
- Python3 e pip3 (<https://www.python.org/downloads/>)
 - PyJWT (executar *pip3 install pyjwt*)
 - Flask (executar *pip3 install flask*)
 - Psycopg2-binary (executar *pip3 install psycopg2*)
 - Bcrypt (executar *pip3 install py-bcrypt*)

Configuração da base de dados

Para iniciar a aplicação Rest é necessário ter previamente uma base de dados e um utilizador com permissões para a aceder, criar e alterar tabelas. As configurações da base de dados devem ser posteriormente atualizadas no ficheiro *config.json* presente na pasta *config*.

Após este passo é necessário criar as devidas tabelas de forma a poder executar o programa. Para isso, é fornecido o *script create_tables.sql* (presente na pasta *sql* da pasta submetida) devendo este ser executado no Postgres (com recurso ao *PgAdmin* ou *psql*).

Configuração da chave secreta dos *tokens* de autenticação

Para alterar a chave usada na codificação do *token* basta modificar o parâmetro *SECRET_KEY* do ficheiro *token.json* da pasta *config*.

Execução

Para correr a aplicação basta ir à pasta *src* pelo terminal e executar o comando *python3 main.py*

NOTA: PARA EFETUAR TESTES COM A APLICAÇÃO É FORNECIDO UM SCRIPT COM DADOS EXEMPLIFICATIVOS PARA INSERIR NA BASE DE DADOS.

Manual de utilizador

Funcionalidades e os seus *endpoints*

Registrar novos utilizadores

Endpoint (POST): /dbproj/user

Corpo do pedido: json com o *username*, *email* e *password* do utilizador a registar.

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro no registo, é retornado um json com uma mensagem de erro, caso contrário é retornado um json com novo id do respetivo utilizador.

```
{
  "username": "teste",
  "email": "teste@gmail.com",
  "password": "password123"
}
```

Autenticar utilizadores

Endpoint (PUT): /dbproj/user

Corpo do pedido: json com o *username* e a *password* do utilizador que se quer autenticar

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro na autenticação, é retornado um json com uma mensagem de erro, caso contrário é retornado um json com o *token* de autenticação que o utilizador deverá usar nos pedidos que requerem autenticação.

```
{
  "username": "teste",
  "password": "password123"
}
```

Criar um leilão

Endpoint (POST): /dbproj/auction

Corpo do pedido: json com o id, nome e descrição do artigo a licitar, preço mínimo de licitação, o título e descrição e data de fim do leilão.

Parâmetros do pedido: *Token* de autenticação

Resposta: No caso de ocorrer um erro na criação, é retornado um json com uma mensagem de erro, caso contrário é retornado um id do leilão criado.

```
{
  "item_id": "ISBN 978-3-16-148410-0",
  "item_name": "Livro A",
  "item_description": "Livro de aventuras.",
  "min_price": 1.50,
  "title": "Leilão de um livro",
  "description": "Livro como novo, aposta inicial baixa",
  "end_date": "2021-06-23 18:30:00"
}
```

Listar todos os leilões

Endpoint (GET): /dbproj/auctions

Corpo do pedido: Não necessita

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro na listagem, é retornado um json com uma mensagem de erro, caso contrário retorna uma lista com a informação em formato json dos leilões existentes.

Pesquisar um leilão

Endpoint (GET): /dbproj/auctions/<keyword>

Corpo do pedido: Não necessita

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro na pesquisa, é retornado um json com uma mensagem de erro, caso contrário retorna uma lista com a informação em formato json dos leilões em que a *keyword*, fornecida no *endpoint*, corresponde ao id do artigo ou aparece na descrição do leilão.

Consultar informação de um leilão

Endpoint (GET): /dbproj/auction/<auction_id>

Corpo do pedido: Não necessita

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro na consulta, é retornado um *json* com uma mensagem de erro, caso contrário retorna os detalhes, as mensagens do mural e o histórico, em formato *json*, do leilão cujo id corresponde ao *auction_id* fornecido no *endpoint*.

Listar leilões onde o utilizador tem atividade

Endpoint (GET): /dbproj/activity

Corpo do pedido: Não necessita

Parâmetros do pedido: *Token* de autenticação

Resposta: No caso de ocorrer um erro na listagem, é retornado um *json* com uma mensagem de erro, caso contrário retorna uma lista com a informação de todos os leilões, em formato *json*, em que utilizador contido no *token* licitou, escreveu mensagens ou é dono.

Licitar

Endpoint (GET): /dbproj/bid/<auction_id>/<bidding_value>

Corpo do pedido: Não necessita

Parâmetros do pedido: *Token* de autenticação

Resposta: No caso de ocorrer um erro na licitação, é retornado um *json* com uma mensagem de erro. Caso contrário, é retornado um *json* com uma mensagem de sucesso.

Nota: o parâmetro *auction_id* refere-se ao id do leilão onde o utilizador quer licitar e o parâmetro *bidding_value* corresponde ao valor da licitação.

Editar informação do leilão

Endpoint (PUT): /dbproj/auction/<auction_id>

Corpo do pedido: *json* com o novo título e/ou descrição do leilão

Parâmetros do pedido: *Token* de autenticação

Resposta: No caso de ocorrer um erro na edição do leilão, é retornado um *json* com uma mensagem de erro, caso contrário é retornada toda a informação, em formato *json*, do leilão cujo id corresponde ao *auction_id*.

```
{
  "title": "Novo titulo",
  "description": "Nova descricao"
}
```

Escrever no mural de um leilão

Endpoint (POST): /dbproj/message

Corpo do pedido: *json* com a mensagem e o id do leilão onde o utilizador quer escrever

Parâmetros do pedido: *Token* de autenticação

Resposta: No caso de ocorrer um erro ao escrever a mensagem, é retornado um *json* com uma mensagem de erro. Caso contrário, é retornado um *json* o id da mensagem escrita.

```
{
  "auction_id": 9,
  "body": "Mensagem de exemplo"
}
```

Ver notificações

Endpoint (GET): /dbproj/notifications

Corpo do pedido: Não necessita

Parâmetros do pedido: Token de autenticação

Resposta: No caso de ocorrer um erro ao ler as notificações, retorna um *json* com uma mensagem de erro. Caso contrário, é retornado um *json* com uma lista as notificações do utilizador contido no *token*.

Nota: esta funcionalidade vai verificar se o utilizador tem notificações por ler, marcando-as como lidas e retornando-as ao utilizador.

Terminar leilões

Endpoint (PUT): /dbproj/auctions/end

Corpo do pedido: Não necessita

Parâmetros do pedido: Não necessita

Resposta: No caso de ocorrer um erro, retorna um *json* com uma mensagem de erro. Caso contrário, é retornado um *json* com uma mensagem de sucesso.

Nota: esta funcionalidade vai pesquisar todos os leilões verificando se já expiraram. Quando encontra leilões nesta situação, marca-os como terminados e envia uma notificação ao dono do leilão e ao licitador vencedor se tal existir.

Diagrama ER

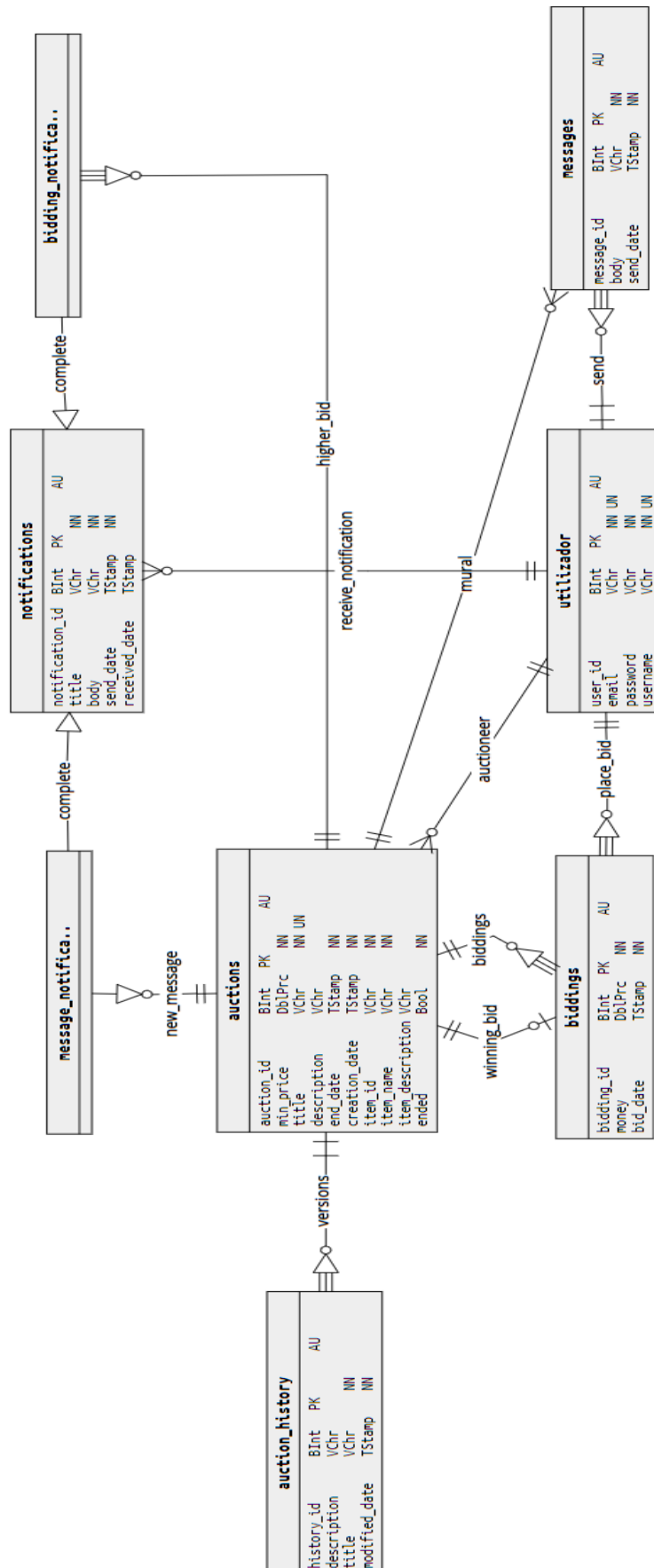
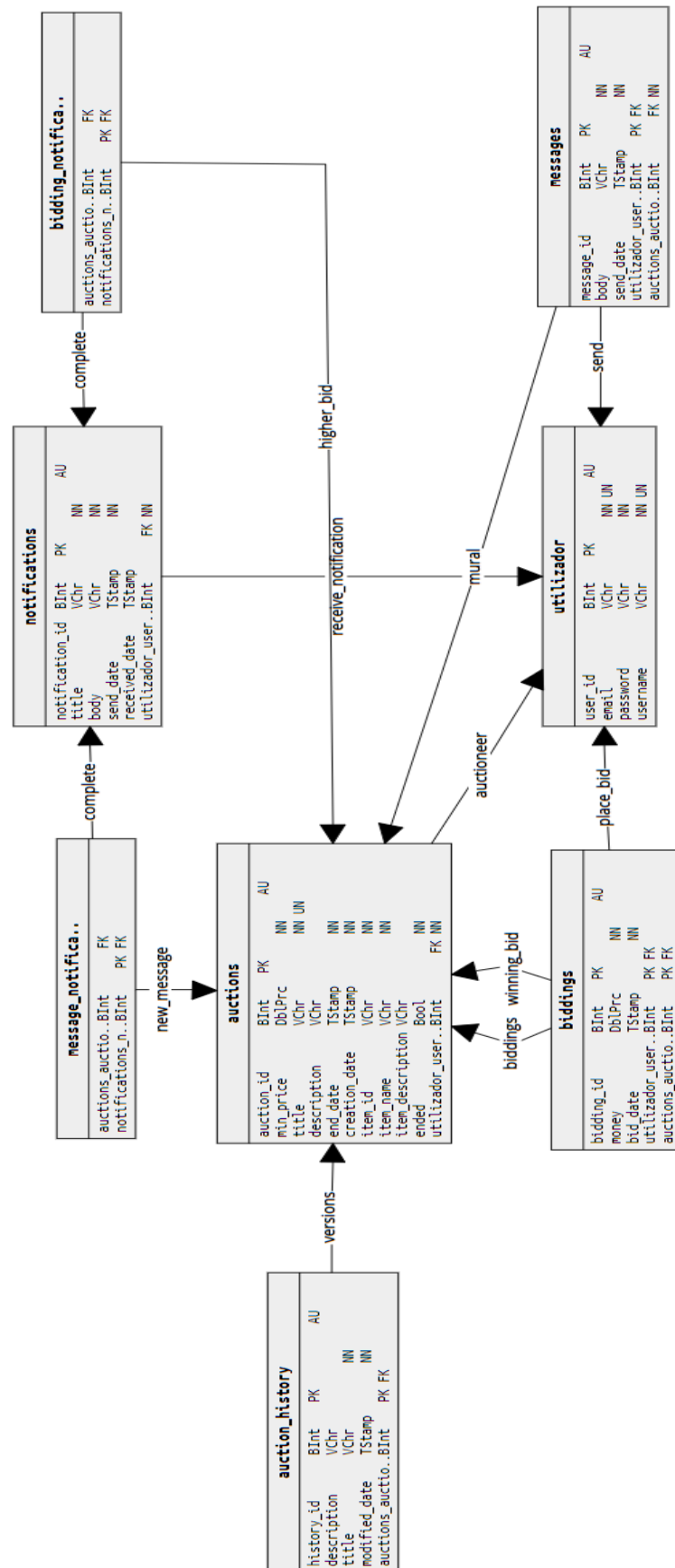


Diagrama Relacional



Plano de desenvolvimento

O trabalho foi desenvolvido pelos dois simultaneamente, embora ambos estarem a trabalhar em diferentes partes do código, houve sempre uma troca de informação e de ideias, o que impossibilita fazer uma distinção correta do que cada um fez.

Estima-se que para realizar este projeto foram necessárias 50 horas por cada elemento.