

RELATÓRIO PROJETO BASE DE DADOS 2020/2021



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Projeto desenvolvido por:

Edgar Filipe Ferreira Duarte 2019216077

Pedro Guilherme da Cruz Ferreira 2018277677

1. Manual de Instalação

Para utilizar esta aplicação, primeiramente precisa de instalar o PostMan (<https://www.postman.com/downloads/>), o Python3.0 (<https://www.python.org/downloads/>) e o pgAdmin (<https://www.pgadmin.org/>). No Postman importe o ficheiro “postmanCollection.json”, enviado juntamente com o projeto, para uma workspace pessoal. No pgAdmin, primeiramente corra o script de criação de tabelas “createtables.sql”, seguido do script que cria triggers “triggers.sql”. Para correr o programa python, certifique-se que tem as seguintes livrarias instaladas:

- jwt;
- functools;
- werkzeug;
- traceback;
- flask;
- psycopg2;
- threading;

Caso alguma destas livrarias esteja em falta, num terminal escreva:

```
pip -install (name of the library to install)
```

2. User Manual

Para conseguir ter uma base de dados, terá de correr o ficheiro “Startdb.txt” numa query tool do pgAdmin. A seguir, precisará de correr o código gerador de triggers “Triggers.txt”, para criar triggers necessário ao bom funcionamento do programa. Também precisará de adequar a sua conexão à base de dados no script python.

A interação com a aplicação é toda ela feita através de um cliente REST, uma vez que não foi pedido o desenvolvimento de uma aplicação web. Com isto, as diferentes funcionalidades da aplicação estão disponíveis nos diversos endpoints.

Um utilizador não autenticado tem acesso às seguintes funcionalidades:

- **Registo de utilizadores**

- **Req**

- **POST** - <http://localhost:8080/dbproj/user>

Test / Registrar Utilizador

POST	▼	http://localhost:8080/dbproj/user?username=&email=&password=
Params	Authorization	Headers (8)
Query Params		
	KEY	VALUE
<input checked="" type="checkbox"/>	username	
<input checked="" type="checkbox"/>	email	
<input checked="" type="checkbox"/>	password	

- **Res**

```
"userId": "João"
```

- **Autenticação de utilizadores**

- **Req**

- **PUT** - <http://localhost:8080/dbproj/user>

Test / Autenticar Utilizador

PUT <http://localhost:8080/dbproj/user>

Params ● Authorization ● Headers (9) Body Pre-request Script Tests Settings

Type Basic Auth

Heads up! These parameters hold sensitive data. To keep this data secure while v recommend using variables. [Learn more about variables](#)

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username

Password

☐ Show Password

- **Res**

"authToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJ1c2VybmFtZSI6InEifQ.
FWPRLJaIvZUub6kYiHlPXrILmaFPXdbFiW313o6khZw"

Quando um utilizador se autêntica, tem acesso ao seu token e terá que o passar no cabeçalho da request que quer fazer para poder ter acesso às outras funcionalidades:

Params ● Authorization Headers (9) Body Pre-request Script Tests Settings

Headers 8 hidden

	KEY	VALUE
<input checked="" type="checkbox"/>	user-token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZ...

- **Criar um novo leilão**

- **Req**

- **POST** - <http://localhost:8080/dbproj/leilao>

Test / Criar leilão

POST <http://localhost:8080/dbproj/leilao?artigoId=1000000000000&precoMinimo=100&titulo=faca ouro>

Params ● Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	artigoId	1000000000000
<input checked="" type="checkbox"/>	precoMinimo	100
<input checked="" type="checkbox"/>	titulo	faca ouro
<input checked="" type="checkbox"/>	descricaoLeilao	quem vencer ganha
<input checked="" type="checkbox"/>	descricaoArtigo	faca banhada a ouro
<input checked="" type="checkbox"/>	tempoFinal	2021-06-19 10:00:00.74200

- **Res**

"leilaoID": 11

- Listar todos os leilões existentes

- Req

- GET - <http://localhost:8080/dbproj/leiloes>

- Res

```
{
  "descricao": "teste",
  "leilaoId": 1
},
{
  "descricao": "teste",
  "leilaoId": 2
},
{
  "descricao": "teste",
  "leilaoId": 3
},
```

- Pesquisar leilões existentes

- Req

- GET - <http://localhost:8080/dbproj/leiloes/{keyword}>
 - A **keyword** tanto pode ser o identificador do artigo como parte da descrição do artigo.

- Res

```
"descricao": "quem vencer ganha",
"leilaoId": 11
```

- Consultar detalhes de um leilão

- Req

- GET - <http://localhost:8080/dbproj/leilao/{leilaoid}>
 - **leilaoid** é o identificador do leilão que se pretende consultar.

- **Res**

```
{
  "descricao": "quem vencer ganha",
  "leilaoId": 11,
  "termino": "Sat, 19 Jun 2021 10:00:00 GMT"
},
"----- Mural -----",
[
  "Sun, 30 May 2021 16:13:41 GMT",
  "atum",
  "É meu!"
],
"----- Licitações -----",
[
  "Sun, 30 May 2021 16:19:47 GMT",
  "atum",
  200
],
[
  "Sun, 30 May 2021 16:21:52 GMT",
  "pedro",
  250
]
]
```

- **Listar todos os leilões em que o utilizador tenha atividade**

- **Req**

- GET - <http://localhost:8080/dbproj/user/leiloes>

- **Res**

```
{
  "descricao": "quem vencer ganha",
  "leilaoId": 11,
  "preco atual": 250,
  "tempo final": "Sat, 19 Jun 2021 10:00:00 GMT",
  "titulo": "faca ouro"
},
{
  "descricao": "agora",
  "leilaoId": 8,
  "preco atual": 12,
  "tempo final": "Sun, 30 May 2021 11:43:00 GMT",
  "titulo": "agora"
}
}
```

- **Efetuar uma licitação num leilão**

- **Req**

- GET - <http://localhost:8080/dbproj/licitar/{leilaoId}/{licitacao}>

- **leilaoId** é o identificador do leilão em que se pretende licitar.

- **licitacao** é o valor que se pretende licitar

- **Res**

"Sucesso": "Licitação de 200 no leilão 11"

- **Editar propriedades de um leilão**

- **Req**

- PUT - <http://localhost:8080/dbproj/leilao/{leilaoId}>

- **leilaoId** é o identificador do leilão que se pretende editar.

Test / Editar leilão

PUT [http://localhost:8080/dbproj/leilao/3?titulo=novo titulo&descricao=nova descricao](http://localhost:8080/dbproj/leilao/3?titulo=novo%20titulo&descricao=nova%20descricao)

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	titulo	novo titulo
<input checked="" type="checkbox"/>	descricao	nova descricao

○ Res

```
"descricao": "nova descricao",  
"leilaoId": "11",  
"titulo": "novo titulo"
```

● Escrever mensagem no mural de um leilão

○ Req

- PUT - <http://localhost:8080/dbproj/mural/{leilaoid}>
- leilaoid é o identificador do leilão em que se pretende escrever.

Test / Escrever mensagem

PUT [http://localhost:8080/dbproj/mural/11?mensagem=É meu!](http://localhost:8080/dbproj/mural/11?mensagem=É%20meu!)

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	mensagem	É meu!

○ Res

```
"Sucesso": "Mensagem 'É meu!' publicacao no mural do leilao 11"
```

● Caixa de entrada do utilizador

○ Req

- GET - <http://localhost:8080/dbproj/user/caixa>

○ Res

```
"2021-05-30 11:41:09.721798": "A sua licitação no leilão 8 foi ultrapassada!"
```

```
"2021-05-30 11:41:17.588283": "A sua licitação no leilão 9 foi ultrapassada!"
```

```
"2021-05-30 16:13:41.289235": "Nova mensagem no leilao 11 de atum:É meu!"
```

- Terminar leilões manualmente

- Req

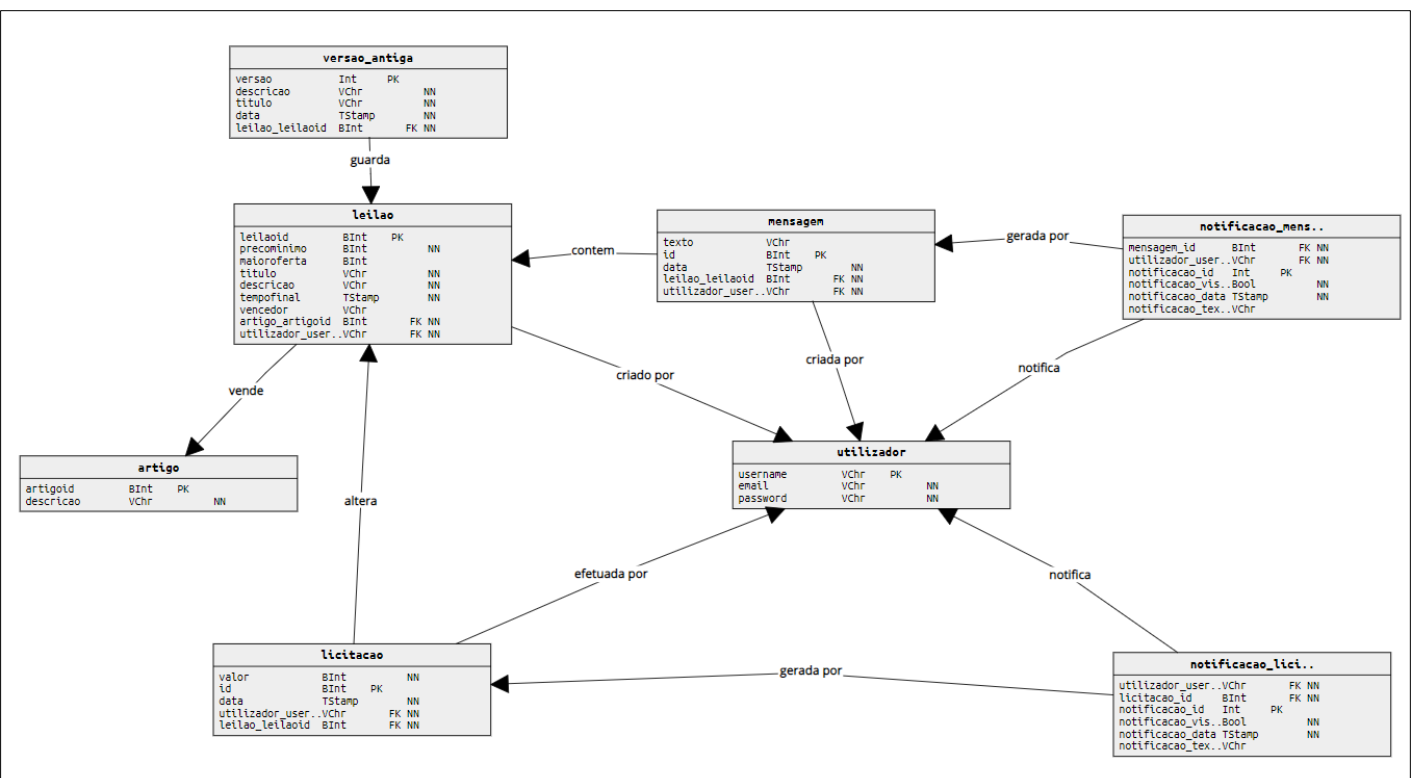
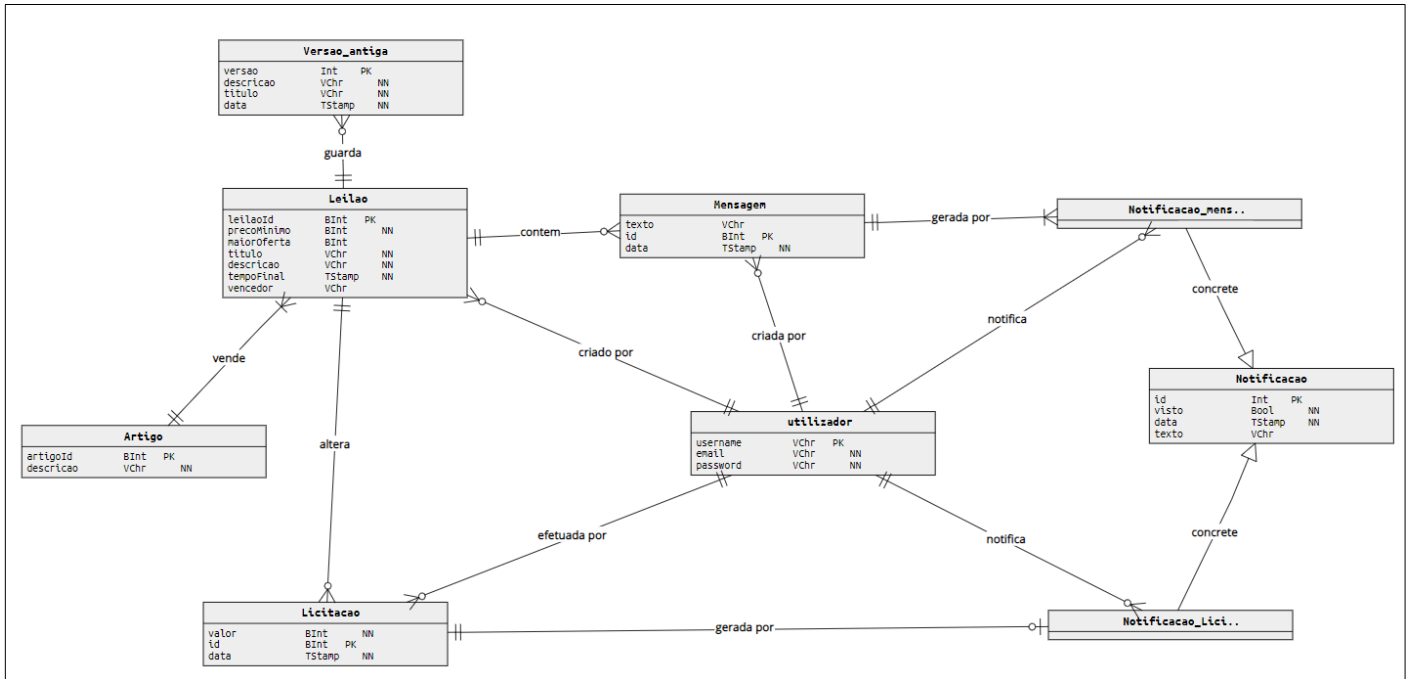
- GET - <http://localhost:8080/dbproj/terminar>

- Res

"Sucesso": "A invocar término de leilões"

3. ER e Relational Data Models

As tabelas criadas para resolver a questão imposta foram as seguintes:



4. Plano de desenvolvimento

O plano de desenvolvimento foi o previsto na defesa intermédia:

Semana	Tarefas Edgar	Tarefas Pedro
18/04 - 24/04	Registo de utilizadores.	Autenticação de utilizadores.
25/04 - 01/05	Criar leilão. Listar leilões.	Pesquisar leilões existentes. Consultar detalhes de um leilão
02/05 - 08/05	Listar todos os leilões em que um utilizador tenha atividade.	Efetuar uma licitação num leilão.
09/05 - 15/05	Editar propriedades de um leilão.	Escrever mensagem no mural de um leilão.
16/05 - 22/05	Notificação de mensagens. Notificação de licitação.	Término do leilão.
23/05 - 29/05	Testes, correções, comentar, limpar código	

O esforço foi de 30 horas por aluno.

5. Aspetos acerca da aplicação

A aplicação foi desenvolvida em Python 3.0. A partir do `psycpg2` é feita a ligação à base de dados. Cada funcionalidade imposta pelo enunciado tem a sua própria função. Para um utilizador usufruir das funcionalidades da aplicação, inicialmente, terá de se registar na aplicação através de um nome de utilizador, de um email e de uma password. Após se registar, necessitará de se autenticar, para assim, gerar um token único que necessitará para utilizar todas as funcionalidades do programa (nem todas obrigam a utilização de tokens).

A geração de notificações e o guardamento de versões antigas dos leilões é feita através de triggers. No primeiro caso, é gerada uma notificação de licitação sempre que uma licitação de um usuário é ultrapassada. Neste caso, após ser inserida uma nova licitação, o trigger ativa-se, sendo que procura pela segunda maior oferta feita atualmente no leilão e, se existir, envia uma notificação ao utilizador que a realizou (se não for o mesmo utilizador da nova licitação máxima). Ainda no primeiro caso, é gerada uma notificação de mensagem a todos os utilizadores que participem no mural de um leilão. Após a inserção de uma nova mensagem no mural, o trigger ativa-se, procurando todos os utilizadores (excluindo o

utilizador que enviou a mensagem) e o vendedor do item do leilão, e enviando-lhes a respectiva notificação. No segundo caso, para guardar versões antigas do leilão, o trigger ativa-se depois de existir um update nas colunas correspondentes à descrição e ao título do leilão, sendo que guarda as informações anteriores (“OLD”) para a tabela “versao_antiga”.

Quando um utilizador verifica a sua caixa de entrada, apenas aparecem as mensagens ainda não lidas, sendo que só as poderá ler neste pedido. Se efetuar um pedido logo a seguir a consultar a sua caixa de mensagens, ela aparecerá vazia, pois as mensagens foram consideradas como lidas.

Para gerir o término de leilões foi criada uma thread que, de 30 em 30 segundos, executa uma query para terminar leilões. É de notar que mesmo que a thread não termine o leilão exatamente a tempo, foram tomadas precauções em todas as transações que pudessem ser suscetíveis a falhas de proteção, para que elas não aceitem pedidos fora de hora.