

# README

## LCM - Relatório Complementar

### Índice

1. [Criação da Interface](#)
2. [Camada de Interação com a Base de Dados](#)
3. [Envio de Dados para o Backend](#)
4. [Mudar as Informações para Conectar](#)
5. [Como Executar o Projeto](#)

## 1. Criação da Interface

Decidimos criar a interface em HTML e, para estilizar a mesma, utilizamos Bootstrap. Para facilitar a comunicação com a base de dados, usamos HTMX, que nos permitiu realizar uma interação complexa de maneira mais simples e eficaz, mantendo o foco principal da disciplina: a manipulação de bases de dados em SQL.

## 2. Camada de Interação com a Base de Dados

Para a interação com a base de dados, utilizamos Flask, que é um micro-framework em Python, essencialmente utilizado em desenvolvimento web. Ele é responsável por processar as requisições feitas pela interface do usuário, interagir com a base de dados, modificando o HTML quando necessário e com as informações corretas.

Além do Flask, utilizamos PyODBC, que é uma biblioteca em Python que permite fazer conexões a uma base de dados, possibilitando assim a manipulação de tabelas.

## 3. Envio de Dados para o Backend

Para enviar os dados submetidos nos formulários para os endpoints do Flask, utilizamos HTMX e JavaScript. Cada endpoint do Flask recebe os dados, atualiza as tabelas conforme necessário e retorna o documento HTML atualizado com as informações.

## 4. Mudar as Informações para Conectar

É necessário mudar nos ficheiros `skinsInserts.py` e `championsInsert.py`, na declaração da variável `conn`:

```
conn = pyodbc.connect('DRIVER={SQL  
Server};SERVER=tcp:mednat.ieeta.pt\SQLSERVER,8101;DATABASE=p11g1;UID=p11g1;P
```

```
WD=RMachado@10')
```

Os campos a serem mudados serão DATABASE, UID, PWD, sendo DATABASE o nome da Base de Dados, UID o nome de utilizador no SQL Management Studio e PWD a *password*.

Se quiser colocar *localhost* o código terá de ser este:

```
conn = pyodbc.connect('DRIVER={SQL  
Server};SERVER=localhost\\SQLEXPRESS;DATABASE=p11g1;Trusted_Connection=yes;'  
)
```

Para além desses ficheiros, na pasta data no ficheiro database.py terá que fazer as mesmas mudanças nesta linha:

```
g.db = pyodbc.connect('DRIVER={SQL  
Server};SERVER=tcp:mednat.ieeta.pt\\SQLSERVER,8101;DATABASE=p11g1;UID=p11g1;P  
WD=RMachado@10')
```

## 5. Como Executar o Projeto

Para executar o projeto, siga os passos abaixo:

1. Verifique a versão do Python:

```
python --version
```

2. Crie e ative um ambiente virtual:

```
python -m venv venv  
venv\Scripts\activate # No Windows  
source venv/bin/activate # No macOS/Linux
```

3. Instale as dependências:

```
pip install -r requirements.txt
```

Antes de executar o projeto, é necessário inserir os arquivos SQL na sua base de dados local na seguinte ordem:

1. [Arquivo SQL DDL](#)
2. [Arquivo SQL IDXs](#)
3. [Arquivo SQL SPs](#)

4. [Arquivo SQL TRGs](#)
5. [Arquivo SQL UDFs](#)
6. [Arquivo SQL VIEWS](#)

Em seguida, execute os arquivos de inserção:

1. [ChampionsInsert](#)
2. [SkinsInsert](#)
3. [ChestsWardsMaps](#)

Depois de realizar as inserções, você pode executar o projeto:

```
python run.py
```