

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Unidade Curricular: Base de Dados 2

Relatório Relativo ao Projeto

Tema: FutebolStats

Realizado por: Marco Rodrigues – 25001

Miguel Silva – 25187

Tiago Figueiredo – 17185

Joel Aparício - 19908

Viseu, 2025

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório relativo ao Projeto

Curso de Licenciatura em Engenharia Informática

Unidade Curricular de Base de Dados 2

FutebolStats

Ano Letivo 2024/25

Viseu, 2025

**RESUMO**

Este programa tem a função de simular um Website de estatísticas de Futebol, onde qualquer utilizador da Internet poderá visitar e ver informações sobre Jogos, Jogadores, Competições, etc.

**Palavras-Chave: Base de Dados, MongoDB, Postgres, Django, Manipulação, Dados.**

**ÍNDICE**

[1. Introdução 1](#_Toc189944499)

[2. Base de Dados 2](#_Toc189944500)

[2.1. Criação da Base de Dados 2](#_Toc189944501)

[2.2. Configuração no Django 3](#_Toc189944502)

[2.2.1. Routers 4](#_Toc189944503)

[3. Framework Django 6](#_Toc189944504)

[3.1. Models 6](#_Toc189944505)

[3.1.1. ObjectId 6](#_Toc189944506)

[3.1.2. Relações 7](#_Toc189944507)

[3.1.3. Nome Tabela 7](#_Toc189944508)

[3.1.4. Devolver Ele Próprio 7](#_Toc189944509)

[3.2. Forms 7](#_Toc189944510)

[3.2.1. Forms Utils 7](#_Toc189944511)

[3.3. Views 7](#_Toc189944512)

[3.4. Templates 7](#_Toc189944513)

[3.5. Urls 7](#_Toc189944514)

[4. Funcionamento 9](#_Toc189944515)

[4.1. Script 9](#_Toc189944516)

[4.2. Programa 10](#_Toc189944517)

[5. Conclusões 11](#_Toc189944518)

[6. Referências 12](#_Toc189944519)

[7. Bibliografia 13](#_Toc189944520)

**Índice de tabelas**

**Não foi encontrada nenhuma entrada do índice de ilustrações.Índice de Figuras**

[Figura 2‑1 Modelo Conceptual 2](#_Toc189944521)

[Figura 2‑2 Modelo Físico 3](#_Toc189944522)

[Figura 2‑3 Configuração SGBD - Django 4](#_Toc189944523)

[Figura 2‑4 Configuração Router – Django 5](#_Toc189944524)

[Figura 2‑5 Router no Model - Django 5](#_Toc189944525)

[Figura 3‑1 Exemplo Model – Django 6](#_Toc189944526)

[Figura 3‑2 Biblioteca ObjectId – Django 6](#_Toc189944527)

[Figura 3‑3 Urls - Django 8](#_Toc189944528)

[Figura 4‑1 Script de Dados – Django 9](#_Toc189944529)

[Figura 4‑2 Inserção de Dados Com Relações - Django 10](#_Toc189944530)

# Introdução

Foi-nos pedido pelos professores de Base de Dados 2 a realizar um projeto prático com a utilização da framework Django e, para a base de dados, o MongoDB e Postgres com o objetivo de avaliarem as nossas capacidades, utilizando o que nos foi ensinado ao longo do semestre.

Neste programa, uma das dificuldades com que nos deparámos foi a configuração para a utilização de duas bases de dados diferentes, sendo estes o MongoDB e o Postgres.

Este documento está organizado em quatro capítulos que se seguem a esta introdução.

No segundo capítulo propõe-se a Base de Dados.

Segue-se o capítulo três, onde falamos da Framework Django.

O capítulo quatro aborda o Funcionamento do programa.

Termina-se com o capítulo cinco, onde se apresentam as respetivas conclusões obtidas sobre este trabalho.

# Base de Dados

Neste capítulo será falado da criação da Base de Dados e a sua configuração no Django.

## Criação da Base de Dados

Inicialmente, como qualquer projeto que envolve dados, é necessário criar a respetiva base de dados e, para isso, foi utilizado o PowerDesigner para a organizar. Esta base de dados consiste em guardar informações relativas a futebol, sendo elas os Jogadores, Clubes, Jogos, etc.

Uma imagem com texto, diagrama, Esquema, esquemático

Descrição gerada automaticamente

Figura ‑ Modelo Conceptual

Uma imagem com texto, diagrama, Esquema, Desenho técnico

Descrição gerada automaticamente

Figura ‑ Modelo Físico

## Configuração no Django

Um dos requisitos não funcionais era a utilização de duas SGBD (Sistema de Gerenciamento de Base de Dados) em que, cada um dos dados só podia ser armazenado numa destas. Depois de discussões, chegou-se à conclusão que seria utilizado o postgres para os utilizadores e o MongoDB para o resto dos dados.

No ficheiro “settings.py” foram adicionadas as respetivas bases de dados a ser utilizadas pelo programa, deixando o postgres como “default” (Fig. 2-3).

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura ‑ Configuração SGBD - Django

### Routers

Para a configurar o Django de forma que os dados sejam guardados nas respetivas SGBD foi necessário criar um ficheiro chamado “routers.py” onde seria feito a respetiva configuração. Como é possível ver na Figura 2-4, foi configurado para cada uma das ações a respetiva Base de Dados dependendo da “app\_label”.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura ‑ Configuração Router – Django

Além do router, foi necessário ir aos respetivos Modelos/Objetos (Models, que será falado no próximo capítulo) e adicionar a “app\_label” nos que serão guardados no MongoDB.

Uma imagem com texto, Tipo de letra, captura de ecrã, número

Descrição gerada automaticamente

Figura ‑ Router no Model - Django

# Framework Django

XXX

## Models

Uma das partes mais importantes do trabalho são os Models. Estes são as “tabelas” de base de dados da Framework do Django, que permitem a configuração e o controlo destas visto que é a partir destes modelos que as tabelas são criadas nos SGBD (através das migrações).

Uma imagem com texto, captura de ecrã, Tipo de letra, software

Os conteúdos gerados por IA poderão estar incorretos.

Figura ‑ Exemplo Model – Django

Estes Models contêm várias definições marcadas pelas diferentes cores na imagem acima (Fig. 3-1), sendo a parte não assinalada as variáveis/colunas da tabela de base de dados “normais”/simples.

### ObjectId

Este Model em específico representa um conjunto de dados que será guardado no SGBD MongoDB e, por isso, o ID não pode ser um número inteiro, mas sim um ObjectId. Para a sua utilização é necessário importar a respetiva biblioteca.

Uma imagem com texto, Tipo de letra, captura de ecrã, número

Os conteúdos gerados por IA poderão estar incorretos.

Figura ‑ Biblioteca ObjectId – Django

Além disso, foi decidido criar a função “get\_id” (assinalado a verde) para devolver o respetivo ObjectId, permitindo assim a utilização do identificador de forma prática e eficiente.

### Relações

XXX

### Nome Tabela

A classe Meta, assinalada a azul, contém o “db\_table” que indica o nome da tabela na respetiva base de dados, existindo em todos eles. Já o “app\_label” existe naqueles que serão guardados no MongoDB, como foi falado anteriormente.

### Devolver Ele Próprio

Como é possível ver na Figura 3-1, existe uma outra função assinalada a laranja. Esta serve para devolver, neste caso, o nome do objeto para o caso de este ser chamado, por exemplo, num formulário.

## Forms

XXX

### Forms Utils

XXX

## Views

XXX

## Templates

XXX

## Urls

Para as *views* funcionarem corretamente é necessário criar os respetivos *urls*. Estes servem para indicar o caminho (selecionado a vermelho) e, ao serem abertos, indicaram qual a *view* (selecionado a rosa) a chamar, permitindo dar-lhe um respetivamente (selecionado a azul). Um excerto do código é possível ver na imagem a baixo.

Uma imagem com texto, captura de ecrã, Tipo de letra

Os conteúdos gerados por IA poderão estar incorretos.

Figura ‑ Urls - Django

# Funcionamento

Neste capítulo iremos falar resumidamente do funcionamento da aplicação Web desenvolvida.

## Script

Um dos requisitos deste trabalho era a existência de um *script* de dados e, para isso, foi criado o ficheiro “inserir\_dados.py” onde foram colocados alguns exemplos de dados.

De forma geral, os dados são inseridos de forma sequencial, sendo guardados em vetores (ou *arrays*) para facilitar a inserção de dados de tabelas com chaves estrangeiras, como é possível ver na Figura 4-2.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura ‑ Script de Dados – Django

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura ‑ Inserção de Dados Com Relações - Django

## Programa

XXX

# Conclusões

Este projeto permitiu-nos melhorar as nossas habilidades na framework Django como software MongoDB. Para o desenvolvimento deste projeto foi utilizado a matéria dada na cadeira de Base de Dados 2 assim como pesquisas adicionais. Este permitiu produzir código de maneira eficaz e organizada, sendo dividido nas respetivas funcionalidades (Models, Forms, etc).

A utilização do MongoDB e pgAdmin (postgres) é uma ótima ferramenta para visualizar os dados do sistema, de forma intuitiva e fácil de compreender.

# Referências

# Bibliografia

Aulas de Base de Dados 2 2024-2025

PDFs disponibilizados no Moodle 2024-2025 ([moodle.estgv.ipv.pt](https://moodle.estgv.ipv.pt)) , Curso Base de Dados 2