ISECMAR
INSTITUTO DE ENGENHARIAS
E CIÊNCIAS DO MAR
UNIVERSIDADE TÉCNICA DO ATLÂNTICO

Universidade Técnica do Atlântico - UTA

Instituto Superior de Engenharias e Ciências do Mar - ISECMAR

Licenciatura de Engenharia Informática e de Telecomunicações

Sistemas Distribuídos

Relatório

"Serviços Web em Python Operações CRUD"

Discentes:

José Carlos Costa
Nelson Sousa
Ericles Fonseca
Cristiany Pires

Orientador:

António Varela

Junho 27, 2025

Sobre o Projeto

Este documento descreve a **API REST** desenvolvida para a disciplina de **Sistemas Distribuídos** (2024/2025). A API, construída utilizando **Python (Flask)** e **PostgreSQL**, permite o gerenciamento completo de **livros** e **revistas**, com suporte a filtros avançados e estatísticas. A arquitetura do projeto é modular e organizada, utilizando **Marshmallow** para validação de dados, **SQLAlchemy** como ORM e **Pytest** para testes automatizados.

Funcionalidades Principais

A API oferece as seguintes funcionalidades principais:

- Operações CRUD completas para **livros** (`/books`).
- Operações CRUD completas para **revistas** (`/revistas`).
- Filtros avançados e estatísticas (`/stats`).
- Testes automatizados com **Pytest**.
- Gerenciamento de migrações de banco de dados com **Flask-Migrate**.

Tecnologias Utilizadas

O projeto utiliza as seguintes tecnologias:

- Python 3.10+
- Flask
- SQLAlchemy
- Marshmallow + Marshmallow-SQLAlchemy
- PostgreSQL (via pgAdmin)
- Flask-Migrate
- Pytest + Pytest-Flask
- SQLite (para testes isolados)

Instalação e Execução

Esta seção descreve os passos para configurar e executar a API.

Pré-requisitos

Certifique-se de que os seguintes itens estão instalados:

- Python 3.10 ou superior
- PostgreSQL (em execução)
- pgAdmin (opcional, para interface gráfica de gerenciamento de banco de dados)
- Git

Passo a Passo para Configuração

Siga os passos abaixo para configurar e executar o projeto:

Clonar o Repositório

```
git clone https://github.com/esoj03/SD--work
cd Trabalho_SD_25
```

Criar um Ambiente Virtual

```
python -m venv venv
```

Ativar o Ambiente Virtual

Para Linux/macOS:

```
source venv/bin/activate
```

Para Windows:

```
venv\Scripts\activate
```

Instalar Dependências

```
python -m pip install --upgrade pip
pip install -r requirements.txt
pip install --upgrade marshmallow-sqlalchemy
```

Configurar Variáveis de Ambiente

Crie um arquivo `.env` na raiz do projeto com o seguinte conteúdo:

```
FLASK_APP=run.py
FLASK_ENV=development
DATABASE_URL=postgresql://Utilizador:senha@localhost:5432/nome_da_sua_base_de_dados
```

Nota: Substitua `Utilizador`, `senha` e `nome_da_sua_base_de_dados` pelas suas credenciais PostgreSQL e nome do banco de dados.

Recomendação: Crie o banco de dados manualmente utilizando o pgAdmin ou o terminal do PostgreSQL devido à arquitetura do projeto.

*

Criar o Banco de Dados Use o pgAdmin ou o terminal do PostgreSQL para criar o banco de dados.

Inicializar Migrações

```
flask db init
```

Criar Scripts de Migração

```
flask db migrate -m "init"
```

Aplicar Migrações

```
flask db upgrade
```

Executar a Aplicação

Certifique-se de que o ambiente virtual está ativado e execute:

```
flask run
```

Executar Testes Automatizados

Para executar testes automatizados, utilize o Pytest. O ambiente de testes usa SQLite para isolar o banco de dados principal.

```
pytest
```

Nota: Testes bem-sucedidos aparecem como pontos verdes no terminal. Falhas são indicadas por letras 'F' em vermelho.

Rotas da API para Livros

Esta seção detalha os endpoints para gerenciamento de livros.

Método	Endpoint	Descrição
GET	/books	Recupera todos os livros no banco de dados.
GET	/books/id	Recupera um livro específico pelo seu ID.
POST	/books	Cria um novo livro. Requer dados JSON no corpo da requisição.
PUT	/books/id	Atualiza um livro específico pelo seu ID.
DELETE	/books/id	Exclui um livro específico pelo seu ID.

Exemplo: GET /books

```
[
    {
        "author": "Ericles",
        "isbn": "789",
        "price": 300,
        "title": "leit"
    },
    {
        "author": "Nelson",
        "isbn": "888",
        "price": 2,
        "title": "leit"
    }
]
```

Exemplo: POST /books

Para criar um livro, envie uma requisição POST com o seguinte corpo JSON:

```
{
    "author": "Ericles",
    "isbn": "789",
    "price": 300,
    "title": "leit"
}
```

Verifique a criação usando o endpoint GET /books.

Rotas da API para Revistas

Esta seção abrange os endpoints para gerenciamento de revistas.

Método	Endpoint	Descrição
GET	/revistas	Recupera todas as revistas no banco de dados.
GET	/revistas/id	Recupera uma revista específica pelo seu ID.
POST	/revistas	Cria uma nova revista. Requer dados JSON no corpo da requisição.
PUT	/revistas/id	Atualiza uma revista específica pelo seu ID.
DELETE	/revistas/id	Exclui uma revista específica pelo seu ID.

Exemplo: GET /revistas


```
[
    {
        "area": "Educacional",
        "data_pub": "2012-03-02",
        "editora": "pt",
        "issn": "7",
        "numero": 16,
        "price": 10,
        "titulo": "The News"
    },
    {
        "area": "Desporto",
        "data_pub": "2019-03-22",
        "editora": "pt",
        "issn": "9",
        "numero": 22,
        "price": 90,
        "titulo": "The Race"
    },
    {
        "area": "Desporto",
        "data_pub": "2019-03-22",
        "editora": "pt",
        "issn": "1",
        "numero": 2,
        "price": 2,
        "titulo": "A Bola"
    }
]
```


Exemplo: GET /revistas/1


```
{
    "area": "Desporto",
    "data_pub": "2019-03-22",
    "editora": "pt",
    "issn": "1",
    "numero": 2,
    "price": 2,
    "titulo": "A Bola"
}
```


Rotas de Estatísticas para Livros

Esta seção descreve os endpoints para estatísticas de livros.

Método	Endpoint	Descrição
GET	/stats/books/top-expensive?limit=N	Retorna os N livros mais caros.

GET /stats/books/top-cheap?limit=N Retorna os N livros mais baratos.
GET /stats/books/count Retorna o total de livros cadastrados.
GET /stats/books/author/¡autor¿ Retorna todos os livros de um autor específico (busca parcial).
DELETE /stats/books/author/¡autor¿ Exclui todos os livros de um autor específico.

Exemplo: GET /stats/books/top-expensive?limit=1

```
{
    "author": "Ericles",
    "isbn": "789",
    "price": 300,
    "title": "leit"
}
```

Exemplo: GET /stats/books/count

```
{
    "Total_livros": 4
}
```

Rotas de Estatísticas para Revistas

Esta seção descreve os endpoints para estatísticas de revistas.
Método Endpoint Descrição
GET /stats/revistas/area/¡area¿ Retorna todas as revistas de uma área específica (busca parcial).
GET /stats/revistas/editora/¡nome¿ Retorna todas as revistas de uma editora específica (busca parcial).
GET /stats/revistas/ano/¡ano¿ Retorna todas as revistas publicadas em um ano específico.
GET /stats/revistas/mais-cara Retorna a revista mais cara cadastrada.
GET /stats/revistas/count Retorna o total de revistas cadastradas.

Exemplo: GET /stats/revistas/area/Desporto

```
[
    {
        "area": "Desporto",
        "data_pub": "2019-03-22",
        "editora": "pt",
        "issn": "9",
        "numero": 22,
        "price": 90,
        "titulo": "The Race"
    },
    {
        "area": "Desporto",
        "data_pub": "2019-03-22",
        "editora": "pt",
        "issn": "1",
        "numero": 2,
        "price": 2,
        "titulo": "A Bola"
    }
]
```

Exemplo: GET /stats/revistas/mais-cara


```
{
    "area": "Desporto",
    "data_pub": "2019-03-22",
    "editora": "pt",
    "issn": "9",
    "numero": 22,
    "price": 90,
    "titulo": "The Race"
}
```


Rotas do Catálogo Geral

Esta seção descreve o endpoint para consulta do catálogo combinado.

Método	Endpoint	Descrição
GET	/stats/catalog/price/¡valor¿	Retorna todos os livros e revistas com o preço exato especificado.

Exemplo: GET /stats/catalog/price/2.0


```
{
    "livros": [
        {
            "author": "Nelson",
            "isbn": "222",
            "price": 2,
            "title": "leit"
        }
    ],
    "revistas": [
        {
            "area": "Desporto",
            "data_pub": "2019-03-22",
            "editora": "pt",
            "issn": "1",
            "numero": 2,
            "price": 2,
            "titulo": "A Bola"
        }
    ]
}
```