**Guia de Execução — Projetos Django (cards\_static e cards\_dynamic)**

Este documento explica, passo a passo, como executar os dois projetos Django gerados a partir do seu HTML: 1) versão Estática (cards\_static) e 2) versão Dinâmica com SQLite (cards\_dynamic). Os exemplos de terminal contemplam Windows (PowerShell) e macOS/Linux (bash/zsh).

# 1) Pré-requisitos

• Python 3.10+ instalado (verificar com: python --version).

• Pip atualizado: python -m pip install --upgrade pip.

• Permissões para criar e ativar ambientes virtuais (venv).

# 2) Projeto 1 — cards\_static (versão estática)

Esta versão apenas exibe o conteúdo do arquivo HTML original, sem parsing nem banco de dados. Útil para checar rapidamente o visual/texto bruto.

## 2.1 Passos — Windows (PowerShell)

|  |
| --- |
| # 1) Entre na pasta do projeto dcd cards\_static  # 2) Crie e ative o ambiente virtual python -m venv .venv .\.venv\Scripts\Activate.ps1  # 3) Instale as dependências pip install -r requirements.txt  # 4) Inicialize (migrations padrão) e rode python manage.py migrate python manage.py runserver |

## 2.2 Passos — macOS / Linux (bash/zsh)

|  |
| --- |
| # 1) Entre na pasta do projeto cd cards\_static  # 2) Crie e ative o ambiente virtual python -m venv .venv source .venv/bin/activate  # 3) Instale as dependências pip install -r requirements.txt  # 4) Inicialize (migrations padrão) e rode python manage.py migrate python manage.py runserver |

Acesse: http://127.0.0.1:8000 — Você verá o arquivo original renderizado dentro de um bloco <pre>.

## 2.3 Estrutura (cards\_static)

|  |
| --- |
| cards\_static/ ├─ manage.py ├─ requirements.txt ├─ cardsproj/ # settings/urls/asgi/wsgi ├─ cards/ # app └─ templates/  ├─ base.html  └─ cards/  ├─ static\_page.html # wrapper da página  └─ source\_raw.html # conteúdo original incluído |

# 3) Projeto 2 — cards\_dynamic (versão dinâmica com SQLite)

Esta versão faz o parsing do mesmo arquivo HTML, salva os dados no SQLite nativo do Django e renderiza telas a partir do banco. Inclui um comando de importação.

## 3.1 Passos — Windows (PowerShell)

|  |
| --- |
| # 1) Entre na pasta do projeto dcd cards\_dynamic  # 2) Crie e ative o ambiente virtual python -m venv .venv .\.venv\Scripts\Activate.ps1  # 3) Instale as dependências pip install -r requirements.txt  # 4) Crie o banco e as tabelas python manage.py migrate  # 5) Importe os dados do arquivo incluído em seed/ python manage.py import\_cards seed\ex2\_cards\_grid\_from\_excel.html  # 6) Rode o servidor python manage.py runserver |

## 3.2 Passos — macOS / Linux (bash/zsh)

|  |
| --- |
| # 1) Entre na pasta do projeto cd cards\_dynamic  # 2) Crie e ative o ambiente virtual python -m venv .venv source .venv/bin/activate  # 3) Instale as dependências pip install -r requirements.txt  # 4) Crie o banco e as tabelas python manage.py migrate  # 5) Importe os dados do arquivo incluído em seed/ python manage.py import\_cards seed/ex2\_cards\_grid\_from\_excel.html  # 6) Rode o servidor python manage.py runserver |

Acesse: http://127.0.0.1:8000 — A página inicial lista os tickers importados. Clique em um ticker para ver os cartões.

## 3.3 Estrutura (cards\_dynamic)

|  |
| --- |
| cards\_dynamic/ ├─ manage.py ├─ requirements.txt ├─ seed/ │ └─ ex2\_cards\_grid\_from\_excel.html # arquivo original para importação ├─ cardsproj/ # settings/urls/asgi/wsgi ├─ cards/ # app (models, admin, views, management command) │ └─ management/commands/import\_cards.py └─ templates/  ├─ base.html  └─ cards/  ├─ company\_list.html  └─ company\_detail.html |

## 3.4 Comando de importação — comportamento

O comando `import\_cards` foi criado para o formato do seu arquivo (Ex2 – Cards Grid). Ele reconhece seções iniciadas por "### Ativo: TICKER" e, dentro de cada seção, blocos de cartão organizados como: Status → "#### Título" → Valor → "Categoria • Fonte: ..." → "Faixa:" → Resumo → "Detalhes" (Fórmula/Definição).

Os campos são persistidos nas tabelas Company e Card (status, título, valor, categoria, fonte, faixa, resumo, fórmula e definição, na ordem original).

# 4) Dicas e Comandos Úteis

• Criar superusuário (opcional, para explorar o /admin):

|  |
| --- |
| python manage.py createsuperuser |

• Resetar o banco durante testes (apaga o arquivo SQLite e recria):

|  |
| --- |
| # No diretório do projeto rm -f db.sqlite3 # macOS/Linux # del db.sqlite3 # Windows (PowerShell) python manage.py migrate |

• Porta já em uso: rode com outra porta, por exemplo 8001:

|  |
| --- |
| python manage.py runserver 8001 |

# 5) Problemas Comuns e Soluções

• PowerShell bloqueou a ativação do venv: abra o PowerShell como Administrador e rode `Set-ExecutionPolicy RemoteSigned` (ou use `cmd.exe`).

• Python não encontrado: garanta que o Python 3.10+ está no PATH e utilize `python` (ou `py -3`) conforme sua instalação.

• Erros de migração: apague o `db.sqlite3` e rode `python manage.py migrate` novamente.

# 6) Próximos Passos

• Ajustar o template dinâmico para refletir exatamente o visual do HTML original (se desejado).

• Adicionar filtros (status, categoria, faixa) e paginação na lista de cartões.

• Expor uma API (Django REST Framework) para servir os cartões a um frontend SPA.