Guia SUPER detalhado – Projetos Django (PyCharm passo a passo)

Atualizado em 22/10/2025 18:36.

Este guia foi feito para você executar os projetos que enviei, mesmo sem experiência prévia. O foco é usar o PyCharm (funciona na edição Community e na Professional).

## Como este guia está organizado

• Parte A — Preparação do ambiente (Python, venv, PyCharm).

• Parte B — Executar cada ZIP (A, B e C) dentro do PyCharm, com todos os comandos.

• Parte C — Popular o banco, acessar as telas e o /admin.

• Parte D — Adicionar novas telas (templates + views + urls + registry).

• Parte E — Solução de problemas frequentes (erros comuns).

• Parte F — Checklist final e comandos prontos para copiar.

## ZIPs recebidos (referência rápida)

1) fundamentus\_dupla.zip — App Django com 2 páginas (Tabela detalhada e Heatmap de status).

2) z3site\_fundamentus\_full.zip — Projeto Django completo com todas as visões + seeds + overview.

3) z3site\_fundamentus\_full\_newview.zip — Igual ao 2, porém já inclui “Minha Nova Visão” e link na Overview.

# PARTE A — Preparação do Ambiente

## A1) Instalar Python 3.10+ (uma vez só)

Baixe e instale o Python 3.10 ou superior. No instalador do Windows, marque a opção “Add Python to PATH”. No macOS e Linux, o Python geralmente já vem instalado, mas é recomendável instalar uma versão atual.

## A2) Instalar o PyCharm

Instale o PyCharm (pode ser Community ou Professional). A edição Professional tem suporte nativo a projetos Django na interface, mas a Community também funciona usando o terminal integrado para rodar os comandos.

## A3) Criar uma pasta de trabalho

Crie uma pasta, por exemplo: C:/projetos (Windows) ou ~/projetos (macOS/Linux). Copie os arquivos .zip para dentro dessa pasta.

## A4) Entendendo o ambiente virtual (venv)

O ambiente virtual é uma “caixa isolada” onde ficam os pacotes do Python do seu projeto. Cada projeto terá sua própria pasta .venv para não misturar versões de bibliotecas.

## A5) Como criar o venv dentro do PyCharm (método gráfico)

1. Abra o PyCharm > File > Open… > selecione a pasta do projeto (depois de descompactar o ZIP).

2. Vá em File > Settings (Windows/Linux) ou PyCharm > Settings (macOS) > Project: <seu-projeto> > Python Interpreter.

3. Clique no ícone de engrenagem > Add > Virtualenv Environment > New. Confirme o local (padrão “.venv” na raiz do projeto).

4. Selecione a base interpreter (Python 3.10+). Clique em Create. O PyCharm criará o venv e selecionará como intérprete do projeto.

5. Instale pacotes pelo painel “Python Packages” (lado direito) ou use o Terminal do PyCharm (aba inferior) com o pip.

## A6) Como criar/ativar o venv pelo terminal (alternativa)

**Windows (PowerShell):**

python -m venv .venv  
.venv\Scripts\Activate  
python -m pip install --upgrade pip

**macOS/Linux:**

python3 -m venv .venv  
source .venv/bin/activate  
python -m pip install --upgrade pip

## A7) Instale o Django no venv

Você pode instalar via Terminal do PyCharm ou pelo painel Python Packages.

**Terminal (qualquer sistema):**

pip install django

# PARTE B — Executar cada ZIP no PyCharm

Sempre descompacte o ZIP antes de abrir no PyCharm. Em seguida, garanta que: (1) o venv está configurado como intérprete do projeto; (2) o pacote Django está instalado; (3) você está no mesmo diretório do arquivo manage.py antes de rodar os comandos no terminal integrado.

## B1) Pacote B — z3site\_fundamentus\_full.zip (recomendado começar por aqui)

Este ZIP já é um projeto Django completo com tudo ligado. Passo a passo:

1. Descompacte o ZIP numa pasta (ex.: z3site).

2. No PyCharm, File > Open… > selecione a pasta “z3site” e abra. Verifique se existe o arquivo manage.py na raiz.

3. Configure o venv (ver A5) e instale o Django (ver A7).

**4. No Terminal do PyCharm (aba inferior), CRIE O BANCO e MIGRE:**

python manage.py makemigrations  
python manage.py migrate

**5. (Opcional, mas recomendado) Carregue dados de exemplo (seeds):**

# conjunto mínimo  
a) python manage.py load\_sample\_metrics  
# conjunto ampliado (recomendado)  
b) python manage.py load\_more\_metrics

**6. Suba o servidor:**

python manage.py runserver

**7. Acesse no navegador:**

Home (digitar ticker): http://localhost:8000/fundamentus/  
Overview (links do ticker): http://localhost:8000/fundamentus/overview/ABEV3/  
Algumas páginas:  
- Flip 3D: /fundamentus/flip/ABEV3/  
- Heatmap (filtro/busca): /fundamentus/heatmap/?q=ABEV3  
- QuickStats: /fundamentus/quickstats/?ticker=ABEV3  
- Comparador (modal): /fundamentus/compare/?a=ABEV3&b=AERI3  
- Heatmap denso (tooltip): /fundamentus/heatmap/dense/?q=ABEV3  
- Heatmap filtro+sort: /fundamentus/heatmap/smart/?sort=valor&order=desc  
- Tabela detalhada: /fundamentus/table/?tickers=ABEV3  
- Heatmap status: /fundamentus/heatmap/status/?tickers=ABEV3

**8. (Opcional) Criar usuário admin para entrar em /admin:**

python manage.py createsuperuser  
# depois acesse http://localhost:8000/admin e faça login

## B2) Pacote C — z3site\_fundamentus\_full\_newview.zip (inclui “Minha Nova Visão”)

O fluxo é idêntico ao Pacote B, só que já existe uma página adicional.

1. Abra o projeto no PyCharm e configure o venv. 2. Rode makemigrations/migrate. 3. (Opcional) seeds. 4. runserver. 5. Acesse:

/fundamentus/ # Home  
/fundamentus/overview/ABEV3/ # Overview  
/fundamentus/minha-nova-visao/ABEV3/ # Nova visão

## B3) Pacote A — fundamentus\_dupla.zip (app isolado para embutir)

Este ZIP contém apenas o app “fundamentus” com duas telas. Para rodar do zero:

1. Descompacte o ZIP em uma pasta vazia. No terminal do PyCharm, crie um projeto Django novo (se não tiver):

django-admin startproject mysite .

2. Adicione o app e as rotas: abra settings.py e urls.py do PROJETO e inclua:

# settings.py  
INSTALLED\_APPS = [  
 # ...  
 'fundamentus',  
]  
  
# urls.py (do projeto)  
from django.urls import path, include  
urlpatterns = [  
 path('fundamentus/', include('fundamentus.urls')),  
]

**3. Migre o banco e rode:**

python manage.py makemigrations  
python manage.py migrate  
python manage.py runserver

**4. Acesse:**

http://localhost:8000/fundamentus/table/?tickers=ABEV3,AERI3  
http://localhost:8000/fundamentus/heatmap/status/?tickers=ABEV3,AERI3

Se você já possui seeds de pacotes anteriores, rode-os aqui para popular os dados. Caso contrário, cadastre pelo /admin (crie um superusuário).

# PARTE C — Popular o banco e navegar pelas telas

• Use os comandos de seed: load\_sample\_metrics (mínimo) e load\_more\_metrics (ampliado).

• Entre no /admin para verificar se Company, IndicatorDefinition e IndicatorValue estão com dados.

• Teste as rotas indicadas em cada pacote. Se algo retornar vazio, verifique os filtros na URL (ex.: ticker/tickers) e se há dados para o ativo informado.

# PARTE D — Criar/Adicionar Novas Telas (PyCharm)

Abaixo um exemplo simples para criar outra visão baseada em um template novo, já no padrão do projeto completo (Pacotes B ou C).

## D1) Criar o template

**Crie o arquivo em templates/fundamentus/minha\_outra\_visao.html com o conteúdo base:**

<!DOCTYPE html>  
<html lang="pt-BR">  
<head>  
 <meta charset="utf-8"/>  
 <title>Minha Outra Visão · {{ company.ticker }}</title>  
 <style>  
 body { font:14px system-ui; color:#0f172a; }  
 .grid { display:grid; grid-template-columns:repeat(auto-fill,minmax(260px,1fr)); gap:12px; }  
 .card { border:1px solid #e2e8f0; border-radius:10px; padding:12px; background:#fff; }  
 </style>  
</head>  
<body>  
 <h1>Minha Outra Visão – {{ company.ticker }}</h1>  
 <div class="grid">  
 {% for iv in values %}  
 <div class="card">  
 <div><strong>{{ iv.indicator.label }}</strong></div>  
 <div>{{ iv.indicator.group }} · {{ iv.source }}</div>  
 <div>{{ iv.value\_display|default:iv.value\_raw }} · {{ iv.classification }}</div>  
 </div>  
 {% empty %}  
 <p>Sem dados.</p>  
 {% endfor %}  
 </div>  
</body>  
</html>

## D2) Criar a view em views.py

**Abra fundamentus/views.py e adicione:**

from django.shortcuts import get\_object\_or\_404, render  
from .models import Company, IndicatorValue  
  
def minha\_outra\_visao(request, ticker: str):  
 company = get\_object\_or\_404(Company, ticker\_\_iexact=ticker)  
 values = (  
 IndicatorValue.objects.select\_related('indicator','company')  
 .filter(company=company)  
 .order\_by('indicator\_\_group','indicator\_\_label')  
 )  
 return render(request, 'fundamentus/minha\_outra\_visao.html', {  
 'company': company,  
 'values': values,  
 })

## D3) Publicar a rota em urls.py

**Abra fundamentus/urls.py e inclua a rota:**

from django.urls import path  
from . import views  
  
urlpatterns = [  
 # ... outras rotas ...  
 path('minha-outra-visao/<str:ticker>/', views.minha\_outra\_visao, name='minha\_outra\_visao'),  
]

## D4) Aparecer na Overview automaticamente (registry.py)

**Edite fundamentus/registry.py e adicione um registro:**

from dataclasses import dataclass  
from typing import Callable, Optional  
  
@dataclass  
class ViewEntry:  
 key: str  
 name: str  
 desc: str  
 build\_url: Callable[[str, Optional[str]], str]  
  
# ... ENTRIES já existente ...  
ENTRIES += [  
 ViewEntry(  
 key='minha\_outra\_visao',  
 name='Minha Outra Visão',  
 desc='Layout personalizado',  
 build\_url=lambda t, b: f"/fundamentus/minha-outra-visao/{t}/",  
 ),  
]

Salve os arquivos e atualize a página Overview do seu ticker; o novo cartão deve aparecer automaticamente. Se não aparecer, verifique erros no console/terminal do servidor.

# PARTE E — Solução de Problemas (Troubleshooting)

1) "django-admin: comando não encontrado" → Garanta que o Python está no PATH (Windows) ou use "python -m django --version" para confirmar. Reinstale o Python marcando “Add to PATH” no Windows se necessário.

2) Erro de migração (apps não carregados) → Confira se INSTALLED\_APPS contém "fundamentus" e se você está no diretório do manage.py antes de rodar os comandos.

3) TemplateDoesNotExist → Verifique o caminho: templates/fundamentus/<arquivo>.html e se o TEMPLATES.DIRS (settings.py) inclui a pasta de templates do app (nos projetos prontos isso já está OK).

4) ModuleNotFoundError → Nomes de pastas/arquivos incorretos ou pacote não instalado. Rode novamente com o venv ativo e instale dependências (pip install django).

5) Página sem dados → Rode as seeds (load\_sample\_metrics / load\_more\_metrics) ou crie dados via /admin (crie superusuário com createsuperuser).

6) Porta 8000 em uso → Rode "python manage.py runserver 8001" para usar outra porta (troque para 8002, se necessário).

7) CSRF/Forbidden em formulários → Confirme que está acessando via http://localhost:8000 e que o DEBUG está True (já configurado nos projetos).

8) Alterei modelos e nada mudou → Depois de mudar models.py, rode novamente makemigrations e migrate. Reinicie o servidor.

# PARTE F — Checklist Final (copie e marque)

[ ] Descompactei o ZIP em uma pasta.  
[ ] Abri a pasta do projeto no PyCharm.  
[ ] Criei o venv (.venv) e selecionei como intérprete.  
[ ] Instalei o Django (pip install django).  
[ ] Rodei: python manage.py makemigrations && python manage.py migrate.  
[ ] (Opcional) Rodei seeds: load\_sample\_metrics ou load\_more\_metrics.  
[ ] Subi o servidor: python manage.py runserver.  
[ ] Acessei as URLs indicadas.  
[ ] (Opcional) Criei superusuário e entrei no /admin.  
[ ] (Opcional) Adicionei novas visões e registrei no registry.py.