# Guia do Projeto: Django + SQLite – Pesquisa por CNPJ

Este projeto demonstra uma aplicação web simples com Django e SQLite. Há duas telas: uma para pesquisar por CNPJ e outra exibindo o resultado em formato de grid (tabela).

## Estrutura do Projeto

Pasta raiz: clienteproj/

- manage.py – utilitário de linha de comando do Django  
- clienteproj/settings.py – configurações (SQLite, templates e estáticos)  
- cadastro/ – app com models, views, forms, urls e templates  
- templates/base.html – layout base  
- static/css/styles.css – estilos  
- cadastro/templates/cadastro/\*.html – telas de pesquisa e resultados  
- cadastro/fixtures/clientes.json – dados de exemplo

## Modelo de Dados: Cliente

* ID (auto, chave primária)
* nome (CharField)
* endereço (CharField)
* profissão (CharField)
* cep (CharField, ex.: 01310-100)
* cnpj (CharField, ex.: 12.345.678/0001-90)

## Como abrir e executar no PyCharm

1. Abra o PyCharm e escolha \*\*Open\*\*. Selecione a pasta `clienteproj`.
2. Crie (ou aceite) um ambiente virtual (venv) quando solicitado.
3. No terminal do PyCharm, instale as dependências: `pip install -r requirements.txt`
4. Crie as tabelas do banco: `python manage.py makemigrations` e depois `python manage.py migrate`
5. Opcional: carregue dados de exemplo: `python manage.py loaddata clientes.json`
6. Opcional: crie um superusuário para acessar o /admin: `python manage.py createsuperuser`
7. Execute o servidor: `python manage.py runserver`
8. Acesse http://127.0.0.1:8000 no navegador. Use a tela inicial para pesquisar por CNPJ e veja os resultados na grid.

## Detalhes Técnicos

• O formulário `CNPJSearchForm` normaliza a entrada removendo caracteres não numéricos e exige ao menos 8 dígitos.

• A view `pesquisa\_cnpj` exibe o formulário e, ao enviar, redireciona para `resultado/<cnpj>/`.

• A view `resultado\_grid` filtra `Cliente` com `cnpj\_\_icontains` e gera a tabela com todos os campos.

• O admin do Django está configurado para listar e pesquisar por nome e CNPJ.

## Observações

* Este projeto é demonstrativo (DEBUG=True). Não use a mesma SECRET\_KEY em produção.
* Para validação completa de CNPJ, use bibliotecas específicas ou implemente os dígitos verificadores.
* Se os estáticos não aparecerem, confira STATIC\_URL e STATICFILES\_DIRS em settings.py.
* Perfeito, Silvio! 💡 Preparei um **exemplo completo** com **Python + Django + SQLite + HTML + CSS** contendo:
* **Modelo** com os campos: **ID** (auto), **nome**, **endereço**, **profissão**, **cep**, **cnpj** *(interpretei “cnjp” como* ***CNPJ****)*.
* **Tela de pesquisa** por **CNPJ**.
* **Tela de resultados** com uma **grid** exibindo **todos os campos**.
* Um **.zip** com todo o código do projeto.
* Um **.docx** com explicação e **passo a passo para rodar no PyCharm** (e também via terminal).
* **Downloads**
* Projeto (código-fonte) em ZIP: [**django\_clientes\_exemplo.zip**](blob:https://outlook.office.com/17620b12-5198-4839-b624-aeb0a56abeaf)
* Guia passo a passo (DOCX): [**Guia\_Projeto\_Django\_Clientes.docx**](blob:https://outlook.office.com/4803e50e-a3d0-40e2-80c6-dd67fa62f2eb)
* **Como o exemplo está organizado**
* **Estrutura principal:**
* clienteproj/
* ├─ manage.py
* ├─ requirements.txt
* ├─ README.md
* ├─ db.sqlite3 (será criado após migrations)
* ├─ clienteproj/ (projeto Django)
* │ ├─ \_\_init\_\_.py
* │ ├─ settings.py (SQLite, templates, estáticos)
* │ ├─ urls.py (inclui urls do app)
* │ ├─ asgi.py
* │ └─ wsgi.py
* ├─ cadastro/ (app da aplicação)
* │ ├─ \_\_init\_\_.py
* │ ├─ apps.py
* │ ├─ models.py (modelo Cliente)
* │ ├─ admin.py (admin cadastrado)
* │ ├─ forms.py (form de pesquisa CNPJ)
* │ ├─ views.py (2 views: pesquisa e resultado)
* │ ├─ urls.py
* │ ├─ migrations/
* │ │ └─ \_\_init\_\_.py
* │ ├─ fixtures/
* │ │ └─ clientes.json (dados de exemplo)
* │ └─ templates/
* │ └─ cadastro/
* │ ├─ search.html (tela de pesquisa)
* │ └─ results.html (grid de resultados)
* ├─ templates/
* │ └─ base.html (layout base)
* └─ static/
* └─ css/
* └─ styles.css (estilos)
* **Modelo Cliente** (campos pedidos):
* id (auto, chave primária do Django)
* nome (CharField)
* endereco (CharField)
* profissao (CharField)
* cep (CharField)
* cnpj (CharField)
* **Fluxo das telas:**

1. **/ (raiz)** → search.html: formulário com um campo de CNPJ (aceita parcial).
2. **/resultado//** → results.html: grid com **ID, Nome, Endereço, Profissão, CEP, CNPJ**.

* A busca usa cnpj\_\_icontains, então se você digitar só os dígitos (o formulário já remove a pontuação) ela faz uma busca parcial. O formulário exige **mínimo 8 dígitos** para evitar resultados muito amplos.
* **Guia rápido (resumo)**
* O **Guia .docx** tem o passo a passo em detalhes, inclusive para abrir no **PyCharm**. Abaixo um **resumo**:

1. **Abrir no PyCharm**: File > Open e selecione a pasta clienteproj.
2. **Criar venv** (quando solicitado) e depois, no **Terminal** do PyCharm:

* pip install -r requirements.txt
* python manage.py makemigrations
* python manage.py migrate
* # (opcional) dados de exemplo
* python manage.py loaddata clientes.json
* # (opcional) admin
* python manage.py createsuperuser
* python manage.py runserver

1. Navegue para [**http://127.0.0.1:8000**](http://127.0.0.1:8000/):
   * Tela de **Pesquisa por CNPJ** (com ou sem pontuação).
   * Ao enviar, abre a **Grid de Resultados** com todos os campos.

* **Observações técnicas**
* Mantive **SQLite** por simplicidade. Tudo já preparado em settings.py.
* **Estáticos** servidos em DEBUG=True via STATIC\_URL e STATICFILES\_DIRS.
* **Admin** já registra Cliente com list\_display e search\_fields (nome, cnpj).
* **Validação de CNPJ**: a forma atual **remove pontuação** e exige **mín. 8 dígitos** para permitir busca **parcial**.\ Se quiser **validação completa** (dígitos verificadores) ou **máscara** no front, eu adiciono rapidinho.
* **Quer que eu personalize?**
* Como você curte comentários **bem detalhados linha a linha** no Python, posso gerar uma **segunda versão** dos arquivos (views.py, forms.py, models.py) com **comentários extensivos** e explicações do fluxo. Também posso:
* Trocar a busca para **match exato** de CNPJ.
* Adicionar **paginações** na grid.
* Incluir **API REST** (Django REST Framework) para consulta de clientes.
* Colocar **validação completa de CNPJ** e **máscara** no formulário.
* Diga como prefere a busca (exata x parcial) e se quer os comentários ultra detalhados que eu já preparo essa versão.
* **References**
* [1] [CodeInterpreter](https://fr-prod.asyncgw.teams.microsoft.com/v1/objects/0-frca-d10-780c6f774fd0ccbc9acfc1bf55b28fae/views/original/django_clientes_exemplo.zip)
* [2] [CodeInterpreter](https://fr-prod.asyncgw.teams.microsoft.com/v1/objects/0-frca-d7-dc7e0f3c34a084389a04767d3d0efcee/views/original/Guia_Projeto_Django_Clientes.docx)