



Desafio Técnico

Introdução

Neste exercício, você deverá construir uma aplicação Django capaz de consumir dados de uma API e/ou de um arquivo .csv, e realizar a persistência dessas informações em um banco de dados relacional, e exibi-las de forma organizada em páginas web com navegação entre entidades.

A proposta simula um cenário real onde é necessário integrar fontes de dados externas ao sistema interno de uma empresa, garantindo organização, relacionamento entre tabelas e boa performance, mesmo com grandes volumes de dados.

Tarefa 1

A partir de uma aplicação Django, consuma os dados da API pública do IBGE (<https://servicodados.ibge.gov.br/api/docs/localidades>) e salve essas informações em um banco de dados relacional PostgreSQL.

A aplicação deverá absorver da API todos os **Estados, Municípios e Distritos** e, salvar no banco de dados. As respectivas tabelas devem ser criadas a partir das models do django e os nomes e campos devem ser correspondentes a cada entidade/rota da API. Apesar de usar as mesmas nomenclaturas, é livre a possibilidade de alterar os nomes para se adaptar às boas práticas do django/python.



Tarefa 2

Você irá criar uma outra model, a model Empresas. Essa model será alimentada a partir de um arquivo CSV que pode ser baixado no link da receita federal abaixo:

https://arquivos.receitafederal.gov.br/dados/cnpj/dados_abertos_cnpj/2025-05

Baixar o arquivo [Empresas0.zip](#). O arquivo talvez não esteja 100% normalizado em relação ao cabeçalho ou formato de arquivo. De todo modo, use o documento a seguir como referência para os campos da tabela:

<https://www.gov.br/receitafederal/dados/cnpj-metadados.pdf>

Você deverá criar algum serviço que possibilite ler o arquivo e que possa inserir na model criada. Esse recurso deve analisar se o registro que está sendo inserido já existe na base de dados. Use o CNPJ como campo de análise, caso o registro já exista, o serviço deve alterar os outros campos, mas não irá duplicar o registro.

Tarefa 3

Além da carga de dados via API e csv, desenvolva páginas web utilizando os recursos do django que permitam a navegação e visualização dos dados salvos no banco.

As páginas devem ser:

- Listagem dos estados
- Listagem dos Municípios
- Listagem do Distritos
- Listagem do Empresas

Todas as listagens devem ser apresentadas em formato de tabelas e é importante que a listagem possibilite a paginação e também filtragem dos dados apresentados.



O que será avaliado:

- Capacidade de modelagem relacional no Django;
- Boas práticas no consumo de APIs REST (tratamento de erro, controle de fluxo);
- Organização do código e uso adequado do ORM;
- Criação de páginas web com navegação entre os dados e paginação de resultados;
- Preocupação com performance, reuso e escalabilidade.

Entrega do Projeto

O projeto deve ser entregue em um repositório Git público.

O repositório deve conter:

- Código-fonte completo da aplicação.
- Script(s) para importação dos dados do IBGE e do CSV da Receita Federal.
- Models Django representando corretamente as entidades solicitadas.
- Páginas web conforme descrito nas tarefas.
- Um arquivo README.md com instruções passo a passo, por exemplo:
 - Instalação e execução do projeto localmente.
 - Configuração do ambiente (variáveis, banco de dados, etc.).
 - Execução para importação dos dados para banco.
 - Acessar as páginas e funcionalidades desenvolvidas.

Observações:

- A estrutura e modelagem do sistema ficam a critério do candidato, desde que atendam às funcionalidades especificadas.
- Você pode usar algum template html pronto para agilizar o desenvolvimento, existem vários gratuitos na internet:
 - <https://www.bootstrapdash.com/all-free-templates>
 - <https://www.creative-tim.com/blog/web-design/free-dashboard-templates/>
- Ainda sobre o template. A interface em si não será critério avaliativo. Mas a usabilidade é importante.
- Por segurança, seria interessante que o acesso ao projeto fosse por login e senha.
- O projeto deve ser enviado até 08/08/2025.



- O envio deve ser feito em resposta a esse email, com o link para o repositório.

Conteúdos que podem ajudar:

- <https://www.treinaweb.com.br/blog/entendendo-o-mtv-do-django/>
- <https://tutorial.djangogirls.org/pt/django/>
- <https://docs.djangoproject.com/en/3.0/>