

## Exercício Prático de Django: Sistema Escolar com **Professor** e **Disciplina**

Este exercício prático tem como objetivo ensinar os alunos a configurar um projeto Django, criar e registrar modelos dentro de um aplicativo, e administrá-los através da interface administrativa do Django. O projeto final será um sistema escolar simples com dois modelos: **Professor** e **Disciplina**.

### Parte 1: Preparação do Ambiente

1. **Criação de Ambiente Virtual:** Inicie criando um ambiente virtual para o projeto para garantir o isolamento das dependências. Use o comando `python3 -m venv venv` para criar e `source venv/bin/activate` (Linux/macOS) ou `.\venv\Scripts\activate` (Windows) para ativar o ambiente virtual.
2. **Instalação do Django:** Com o ambiente ativado, instale o Django utilizando `pip install django`.

### Parte 2: Criação do Projeto e App

1. **Criação do Projeto Django:** No terminal, execute `django-admin startproject main .` para criar um novo projeto chamado `main`.
2. **Criação do App Escola:** Dentro do diretório do projeto, crie um app chamado `escola` executando `python manage.py startapp escola`.

### Parte 3: Estruturação dos Modelos

Dentro do app `escola`, você irá definir dois modelos:

1. **Modelo Professor**
  - Campos: `nome` (string), `email` (campo de e-mail), `celular` (string).
2. **Modelo Disciplina**
  - Campos: `nome` (string), `carga_horaria` (inteiro), `codigo` (string).

Para cada modelo, defina os tipos de campo apropriados conforme indicado, sem a necessidade de implementar os modelos explicitamente neste momento do exercício.

### Parte 4: Registro no Admin Django

1. **Registrar os Modelos:** Demonstre como registrar os modelos **Professor** e **Disciplina** no `escola/admin.py` para que sejam gerenciáveis através da interface administrativa do Django. Use o comando `admin.site.register(Modelo)` para cada modelo.

### Parte 5: Configuração Final

1. **Migrações:** Ensine como preparar e aplicar migrações para criar as tabelas dos modelos no banco de dados com `python manage.py makemigrations` seguido por `python manage.py migrate`.

2. **Criação de Superusuário:** Oriente sobre a criação de um superusuário para acessar a interface administrativa com `python manage.py createsuperuser`.
3. **Executar o Servidor:** Mostre como iniciar o servidor de desenvolvimento usando `python manage.py runserver` e acessar a interface administrativa em `http://127.0.0.1:8000/admin`.