

Desenvolvimento de um Site em Django com Múltiplos Modelos e Variável Global

Objetivo:

Desenvolver um site em Django utilizando a linguagem de programação Python. O projeto deve atender aos seguintes requisitos:

1. Criar ao menos **duas classes** no modelo (models.py).
 2. Utilizar ao menos **uma variável global** no código do projeto.
 3. Implementar a funcionalidade de cadastro, exibição e busca de informações no site.
-

Cenário:

Você foi contratado para desenvolver um sistema simples de gerenciamento de eventos. O site permitirá que os usuários cadastrem **eventos** e associem esses eventos a **categorias**.

Requisitos Funcionais:

1. Modelos:

- Criar um modelo Categoria que contenha os seguintes campos:
 - nome (string, até 50 caracteres, único).
 - descricao (texto, opcional).
- Criar um modelo Evento que contenha os seguintes campos:
 - titulo (string, até 100 caracteres).
 - data (campo de data e hora).
 - local (string, até 200 caracteres).
 - categoria (chave estrangeira para o modelo Categoria).
 - descricao (texto, opcional).

2. Variável Global:

- Definir uma variável global chamada SITE_NAME com o nome do site, e utilizá-la no contexto de templates para exibir o título na barra de navegação.

3. Funcionalidades:

- Página inicial que liste os eventos cadastrados.
 - Página para criar novas categorias.
 - Página para cadastrar novos eventos.
 - Sistema de busca para localizar eventos pelo título.
 - Exibir a categoria associada e a data em formato amigável na listagem de eventos.
-

Passo a Passo:

1. Configuração Inicial:

- Criar um novo projeto Django.
- Configurar o banco de dados (SQLite por padrão).
- Criar um aplicativo chamado eventos.

2. Definição de Modelos:

- Implementar os modelos Categoria e Evento em models.py.
- Executar as migrações para criar as tabelas no banco de dados.

3. Criar a Variável Global:

- No arquivo settings.py, adicionar a variável SITE_NAME = "Gerenciador de Eventos".

4. Views e Templates:

- Criar views para listar, cadastrar e buscar eventos e categorias.
- Utilizar o framework de templates do Django para criar páginas HTML estilizadas.

5. Configuração das URLs:

- Configurar as rotas em urls.py para acesso às páginas do sistema.

6. Testar o Sistema:

- Adicionar alguns eventos e categorias para verificar se as funcionalidades estão funcionando corretamente.

Estrutura do Projeto:

markdown

Copiar código

```
gerenciador_eventos/
├── manage.py
├── gerenciador_eventos/
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── asgi.py
├── eventos/
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations/
│   ├── models.py
│   ├── templates/
│   │   ├── base.html
│   │   ├── categoria_form.html
│   │   └── evento_list.html
```

└─ views.py
└─ urls.py

Avaliação:

1. O projeto utiliza ao menos duas classes no modelo (Categoria e Evento).
2. Existe uma variável global (SITE_NAME) que é utilizada nos templates.
3. O site é funcional e permite cadastrar, exibir e buscar eventos.
4. A organização do código segue as melhores práticas do Django.