

Instruções para Entrega da Atividade Django

Informações Gerais

- Data de Entrega: 17/11
- Formato de Entrega: Link do repositório GitHub
- Modalidade: Individual ou em dupla

Requisitos da Aplicação

Estrutura Básica

- Projeto Django com configuração adequada
- App principal com estrutura organizada
- Pastas `templates` e `static` configuradas corretamente

Modelo de Dados

- Apenas 1 modelo (ex: Produto, Usuário, Tarefa, Livro, etc.)
- O modelo deve ter pelo menos 3 campos diferentes
- Registro no Admin: O modelo deve estar registrado no `admin.py`





Telas Obrigatórias

1. Página Inicial (`/`)
 - Apresentação da aplicação
 - Navegação para outras telas
2. Cadastro (`/cadastro/`)
 - Formulário para criar novos elementos
 - Validação de dados
3. Lista de Elementos (`/lista/`)
 - Exibição de todos os elementos cadastrados
 - Links para detalhes e edição

4. Detalhes do Elemento (`/detalhes/<id>/`)

- Visualização completa de um elemento específico
- Opções para editar ou excluir

Funcionalidades CRUD

-  Create - Cadastrar novo elemento
-  Read - Ver lista e detalhes
-  Update - Modificar elemento existente
-  Delete - Remover elemento

Estilização

- Obrigatório: Estilizar todas as telas
- Recomendado: Uso de IA para auxiliar no design
- Interface consistente e responsiva
- Uso adequado de arquivos estáticos (CSS, imagens)

Persistência de Dados e Admin

Configuração do Admin Django

Para verificar e gerenciar os dados via interface administrativa:

1. Crie um superusuário:
2. `bash`
3. `python manage.py createsuperuser`
4. Registre seu modelo no `admin.py`:
5. `python`

```
from django.contrib import admin
from .models import SeuModelo
```

6. `admin.site.register(SeuModelo)`
7. Acesse o admin:
 - URL: `http://localhost:8000/admin/`
 - Faça login com as credenciais do superusuário
 - Verifique se os dados persistidos aparecem na lista

Verificação de Dados

- Use o admin para confirmar que os dados estão sendo salvos
- Teste criar, editar e excluir elementos via admin
- Verifique a consistência entre a aplicação e o admin

Estrutura do Projeto Esperada

text

```
meu_projeto/
├─ manage.py
├─ meu_projeto/
│   ├─ __init__.py
│   ├─ settings.py
│   ├─ urls.py
│   └─ wsgi.py
├─ meu_app/
│   ├─ migrations/
│   ├─ templates/
│   │   └─ meu_app/
│   │       ├─ base.html
│   │       ├─ index.html
│   │       ├─ cadastro.html
│   │       ├─ lista.html
│   │       └─ detalhes.html
│   └─ static/
│       └─ meu_app/
│           ├─ css/
│           └─ js/
├─ __init__.py
├─ admin.py
├─ apps.py
└─ models.py
```

```
| | views.py
| | urls.py
```

Instruções de Entrega

1. Crie um repositório no GitHub
 - Nome descritivo do projeto
 - README.md com informações básicas
2. Estruture o código
 - Commit's organizados e descritivos
 - Código limpo e bem comentado
3. Adicione ao README:

markdown

Nome do Projeto

Descrição

Breve descrição da aplicação e do modelo utilizado

Integrantes

- Nome completo 1
- Nome completo 2 (se houver)

4. Envie o link do repositório até a data limite para o formulário
<https://forms.gle/8hUhSVAJY2V4RQL17>

Critérios de Avaliação

- Funcionamento correto do CRUD
- Persistência de dados (verificável via admin)
- Organização do projeto Django
- Configuração de templates e arquivos estáticos
- Estilização e usabilidade
- Qualidade do código
- Modelo registrado e funcional no admin



Dicas para Verificação

- Teste as operações CRUD tanto na aplicação quanto no admin
- Verifique se os dados criados na aplicação aparecem no admin e vice-versa
- Use o admin como ferramenta para validar a persistência dos dados

Boa sorte com o projeto! 🚀