

# Projeto Django: Portfólio com Cadastro via Formulário

Projeto em Django que cria um site de portfólio com cadastro de projetos via formulário próprio (sem uso do Django Admin), utilizando CSS puro e banco de dados opcionalmente com SQLite ou MySQL.

## Etapa 1: Criar o projeto e o app

```
pip install django
django-admin startproject portfolio
cd portfolio
python manage.py startapp projects
```

## Etapa 2: Configurar o projeto

### 2.1 Adicione o app no 'settings.py'

```
INSTALLED_APPS = [
    # ...
    'projects',
]
```

### 2.2 Configure diretórios estáticos e de mídia

⚠ Só adicione essas configurações após criar a pasta static e media na raiz do projeto.

```
STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / "static"]

MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'
```

Crie essas pastas:

```
portfolio/
├── static/
│   └── css/
│       └── style.css
└── media/
```

## 2.3 Configuração de URLs

No arquivo `portfolio/urls.py` (do projeto, **não do app**):

```
from django.conf import settings
from django.conf.urls.static import static
from django.urls import path, include

urlpatterns = [
    path('', include('projects.urls')),
]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

## Etapa 3: Definir modelo Project

Em `projects/models.py`:

```
from django.db import models

class Project(models.Model):
    titulo = models.CharField(max_length=200)
    descricao = models.TextField()
    tecnologia = models.CharField(max_length=100)
    imagem = models.ImageField(upload_to='projetos/', null=True,
blank=True)

    def __str__(self):
        return self.titulo
```

## Etapa 4: Configurar o banco de dados

### Opção 1: Usar SQLite (padrão, recomendado para começar)

Não precisa alterar nada.

## Opção 2: Usar MySQL

Instale o conector:

```
pip install mysqlclient
```

Crie o banco no MySQL:

```
CREATE DATABASE portfolio_db CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

Atualize o settings.py:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'portfolio_db',  
        'USER': 'root',  
        'PASSWORD': 'sua_senha',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

## Etapa 5: Migrar o banco de dados

```
python manage.py makemigrations  
python manage.py migrate
```

## Etapa 6: Criar formulário

Em projects/forms.py:

```
from django import forms  
from .models import Project  
  
class ProjectForm(forms.ModelForm):  
    class Meta:  
        model = Project  
        fields = ['titulo', 'descricao', 'tecnologia', 'imagem']
```

## Etapa 7: Criar as views

Em projects/views.py:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Project
from .forms import ProjectForm

def index(request):
    projetos = Project.objects.all()
    return render(request, 'projects/index.html', {'projetos':
projetos, 'titulo_pagina': 'Início'})

def detalhes(request, projeto_id):
    projeto = get_object_or_404(Project, id=projeto_id)
    return render(request, 'projects/detalhes.html', {'projeto':
projeto, 'titulo_pagina': projeto.titulo})

def cadastro(request):
    if request.method == 'POST':
        form = ProjectForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('index')
    else:
        form = ProjectForm()
    return render(request, 'projects/cadastro.html', {'form': form,
'titulo_pagina'
```

## Etapa 8: Criar as rotas do app

Crie projects/urls.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('projeto/<int:projeto_id>/', views.detalhes,
name='detalhes'),
    path('cadastro/', views.cadastro, name='cadastro'),
```

## Etapa 9: Criar os templates

### Estrutura:

```
projects/  
├─ templates/  
│   └─ projects/  
│       ├─ base.html  
│       ├─ index.html  
│       ├─ detalhes.html  
│       └─ cadastro.html
```

#### base.html:

```
<!DOCTYPE html>  
<html lang="pt-br">  
{% load static %}  
<head>  
    <meta charset="UTF-8">  
    <title>{{ titulo_pagina }}</title>  
    <link rel="stylesheet" href="{% static 'css/style.css' %}">  
</head>  
<body>  
    <header>  
        <h1>Portfólio</h1>  
        <nav>  
            <a href="{% url 'index' %}">Início</a> |  
            <a href="{% url 'cadastro' %}">Novo Projeto</a>  
        </nav>  
    </header>  
    <main>  
        {% block content %}{% endblock %}  
    </main>  
</body>  
</html>
```

#### index.html:

```
{% extends 'projects/base.html' %}

{% block content %}
<h2>Projetos</h2>
<ul>
    {% for projeto in projetos %}
        <li>
            <a href="{% url 'detalhes' projeto.id %}">{{ projeto.titulo
        }}</a>
        </li>
        {% empty %}
        <li>Nenhum projeto cadastrado.</li>
        {% endfor %}
    </ul>
{% endblock %}
```

#### detalhes.html:

```
{% extends 'projects/base.html' %}

{% block content %}
<h2>{{ projeto.titulo }}</h2>
<p>{{ projeto.descricao }}</p>
<p><strong>Tecnologia:</strong> {{ projeto.tecnologia }}</p>
{% if projeto.imagem %}
    
{% endif %}
<a href="{% url 'index' %}">Voltar</a>
{% endblock %}
```

#### cadastro.html:

```
{% extends 'projects/base.html' %}

{% block content %}
<h2>Cadastrar Novo Projeto</h2>
<form method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Salvar</button>
</form>
{% endblock %}
```

## Etapa 10: Criar e testar o CSS

Em static/css/style.css:

```
body {
    font-family: sans-serif;
    margin: 20px;
    background-color: #f2f2f2;
}
header {
    background: #444;
    color: white;
    padding: 15px;
}
nav a {
    color: white;
    margin-right: 15px;
    text-decoration: none;
}
main {
    background: white;
    padding: 20px;
    border-radius: 5px;
}
form {
    display: flex;
    flex-direction: column;
}
form input, form textarea, form select {
    margin-bottom: 10px;
    padding: 8px;
}
```

## Etapa 11: Rodar o servidor

```
python manage.py runserver
```

Acesse: <http://localhost:8000>