

AQUI ESTÁ UM GUIA COMPLETO E DETALHADO PARA DESENVOLVER A PLATAFORMA DE APOIO PARA CUIDADOR DE IDOSOS OU PESSOAS COM DEFICIÊNCIA UTILIZANDO DJANGO NO BACKEND, BOOTSTRAP PARA O FRONTEND, E JQUERY/AJAX PARA FUNCIONALIDADES DINÂMICAS. O OBJETIVO É CRIAR UM SISTEMA FUNCIONAL DESDE A INSTALAÇÃO ATÉ O DEPLOY.

---

## Passo 1: Instalação e Configuração do Django

Primeiro, configure o Django e instale as dependências.

```
bash
Copiar código
# Criação do ambiente virtual
python -m venv venv
source venv/bin/activate # No Windows: venv\Scripts\activate

# Instalar dependências
pip install django
pip install django-crispy-forms
pip install bootstrap5
pip install pillow # Para manipulação de imagens
```

## Passo 2: Criar o Projeto e o Aplicativo Django

Agora, crie o projeto e o aplicativo:

```
bash
Copiar código
# Criar o projeto Django
django-admin startproject plataforma_cuidador
cd plataforma_cuidador

# Criar o aplicativo 'cuidados'
python manage.py startapp cuidados
```

## Passo 3: Configuração Inicial

Edite o arquivo settings.py dentro de plataforma\_cuidador/ para adicionar os aplicativos necessários.

- **INSTALLED\_APPS:** Adicione cuidados, crispy\_forms e bootstrap5:

```
python
Copiar código
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
```

```
'django.contrib.messages',
'django.contrib.staticfiles',
'cuidados',
'crispy_forms',
'bootstrap5',
]
```

- **Configuração de Templates e Estáticos:**

```
python
Copiar código
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'], # Defina o diretório de templates
        'APP_DIRS': True,
        ...
    },
]
```

```
STATIC_URL = '/static/'
```

```
MEDIA_URL = '/media/'
```

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

---

## Passo 4: Criar os Modelos

Defina os modelos de Paciente, Lembrete e Relatório. Isso representa as tabelas no banco de dados que armazenarão as informações dos pacientes, lembretes e relatórios.

Abra cuidados/models.py e crie os seguintes modelos:

```
python
Copiar código
from django.db import models
from django.contrib.auth.models import User

class Paciente(models.Model):
    cuidador = models.ForeignKey(User, on_delete=models.CASCADE)
    nome = models.CharField(max_length=100)
    data_nascimento = models.DateField()
    historico_medico = models.TextField(blank=True, null=True)
    medicacoes = models.TextField(blank=True, null=True)
    foto = models.ImageField(upload_to='fotos_pacientes/', null=True, blank=True)

    def __str__(self):
        return self.nome

class Lembrete(models.Model):
```

```

    paciente = models.ForeignKey(Paciente, on_delete=models.CASCADE,
related_name='lembretes')
    descricao = models.CharField(max_length=255)
    horario = models.TimeField()
    data = models.DateField()

    def __str__(self):
        return f"Lembrete para {self.paciente.nome}: {self.descricao}"

class Relatorio(models.Model):
    paciente = models.ForeignKey(Paciente, on_delete=models.CASCADE,
related_name='relatorios')
    data = models.DateField()
    estado_saude = models.TextField()
    atividades = models.TextField()

    def __str__(self):
        return f"Relatório de {self.paciente.nome} em {self.data}"

```

Depois de criar os modelos, execute as migrações:

```

bash
Copiar código
python manage.py makemigrations
python manage.py migrate

```

---

## Passo 5: Criar os Formulários

Agora, crie os formulários em `cuidados/forms.py` para as operações de cadastro e atualização dos pacientes, lembretes e relatórios.

```

python
Copiar código
from django import forms
from .models import Paciente, Lembrete, Relatorio

class PacienteForm(forms.ModelForm):
    class Meta:
        model = Paciente
        fields = ['nome', 'data_nascimento', 'historico_medico', 'medicacoes', 'foto']

class LembreteForm(forms.ModelForm):
    class Meta:
        model = Lembrete
        fields = ['descricao', 'horario', 'data']

class RelatorioForm(forms.ModelForm):
    class Meta:
        model = Relatorio

```

```
fields = ['data', 'estado_saude', 'atividades']
```

---

## Passo 6: Criar as Views

Agora, crie as views que vão processar os dados do usuário.

Abra cuidados/views.py e crie as views necessárias:

```
python
```

```
Copiar código
```

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from .models import Paciente, Lembrete, Relatorio
from .forms import PacienteForm, LembreteForm, RelatorioForm
```

```
@login_required
```

```
def home(request):
    pacientes = Paciente.objects.filter(cuidador=request.user)
    return render(request, 'cuidados/home.html', {'pacientes': pacientes})
```

```
@login_required
```

```
def cadastrar_paciente(request):
    if request.method == 'POST':
        form = PacienteForm(request.POST, request.FILES)
        if form.is_valid():
            paciente = form.save(commit=False)
            paciente.cuidador = request.user
            paciente.save()
            return redirect('home')
    else:
        form = PacienteForm()
    return render(request, 'cuidados/cadastrar_paciente.html', {'form': form})
```

```
@login_required
```

```
def detalhes_paciente(request, id):
    paciente = get_object_or_404(Paciente, id=id, cuidador=request.user)
    lembretes = Lembrete.objects.filter(paciente=paciente)
    relatorios = Relatorio.objects.filter(paciente=paciente)
    return render(request, 'cuidados/detalhes_paciente.html', {
        'paciente': paciente, 'lembretes': lembretes, 'relatorios': relatorios
    })
```

```
@login_required
```

```
def cadastrar_lembrete(request, id):
    paciente = get_object_or_404(Paciente, id=id, cuidador=request.user)
    if request.method == 'POST':
        form = LembreteForm(request.POST)
        if form.is_valid():
            lembrete = form.save(commit=False)
```

```

        lembrete.paciente = paciente
        lembrete.save()
        return redirect('detalhes_paciente', id=id)
    else:
        form = LembreteForm()
        return render(request, 'cuidados/cadastrar_lembrete.html', {'form': form, 'paciente':
paciente})

@login_required
def cadastrar_relatorio(request, id):
    paciente = get_object_or_404(Paciente, id=id, cuidador=request.user)
    if request.method == 'POST':
        form = RelatorioForm(request.POST)
        if form.is_valid():
            relatorio = form.save(commit=False)
            relatorio.paciente = paciente
            relatorio.save()
            return redirect('detalhes_paciente', id=id)
        else:
            form = RelatorioForm()
        return render(request, 'cuidados/cadastrar_relatorio.html', {'form': form, 'paciente':
paciente})

```

---

## Passo 7: Configuração das URLs

Abra cuidados/urls.py e defina as URLs do aplicativo.

```

python
Copiar código
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('paciente/cadastrar/', views.cadastrar_paciente, name='cadastrar_paciente'),
    path('paciente/<int:id>', views.detalhes_paciente, name='detalhes_paciente'),
    path('paciente/<int:id>/lembrete/', views.cadastrar_lembrete,
name='cadastrar_lembrete'),
    path('paciente/<int:id>/relatorio/', views.cadastrar_relatorio,
name='cadastrar_relatorio'),
]

```

---

## Passo 8: Criar os Templates

Agora, criaremos os templates HTML para as páginas.

**base.html** (para usar o Bootstrap 5):

```
html
Copiar código
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Plataforma Cuidador</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap5/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container">
    {% block content %}
    {% endblock %}
  </div>
</body>
</html>
```

#### home.html:

```
html
Copiar código
{% extends 'base.html' %}
{% block content %}
<h1>Bem-vindo à Plataforma de Apoio para Cuidadores</h1>
<a href="{% url 'cadastrar_paciente' %}" class="btn btn-primary">Cadastrar Novo
Paciente</a>
<hr>
<h2>Meus Pacientes</h2>
<ul>
  {% for paciente in pacientes %}
    <li>
      <a href="{% url 'detalhes_paciente' paciente.id %}">{{ paciente.nome }}</a>
    </li>
  {% endfor %}
</ul>
{% endblock %}
```

#### cadastrar\_paciente.html:

```
html
Copiar código
{% extends 'base.html' %}
{% block content %}
<h1>Cadastrar Paciente</h1>
<form method="post" enctype="multipart/form-data">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit" class="btn btn-success">Cadastrar</button>
</form>
```

```
{% endblock %}
```

#### **detalhes\_paciente.html:**

```
html
```

```
Copiar código
```

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

```
{% block content %}
```

```
<h1>{{ paciente.nome }}</h1>
```

```

```

```
<p><strong>Data de Nascimento:</strong> {{ paciente.data_nascimento }}</p>
```

```
<p><strong>Histórico Médico:</strong> {{ paciente.historico_medico }}</p>
```

```
<p><strong>Medicamentos:</strong> {{ paciente.medicacoes }}</p>
```

```
<h3>Lembretes</h3>
```

```
<ul>
```

```
    {% for lembrete in lembretes %}
```

```
        <li>{{ lembrete.descricao }} - {{ lembrete.horario }}</li>
```

```
    {% endfor %}
```

```
</ul>
```

```
<h3>Relatórios</h3>
```

```
<ul>
```

```
    {% for relatorio in relatorios %}
```

```
        <li>{{ relatorio.data }} - {{ relatorio.estado_saude }}</li>
```

```
    {% endfor %}
```

```
</ul>
```

```
<a href="{% url 'cadastrar_lembrete' paciente.id %}" class="btn btn-info">Adicionar  
Lembrete</a>
```

```
<a href="{% url 'cadastrar_relatorio' paciente.id %}" class="btn btn-warning">Adicionar  
Relatório</a>
```

```
{% endblock %}
```

**cadastrar\_lembrete.html** e **cadastrar\_relatorio.html** terão um formato semelhante ao de cadastro de paciente.

---

## **Passo 9: Rodar o Projeto Localmente**

Agora, execute o servidor de desenvolvimento do Django:

```
bash
```

```
Copiar código
```

```
python manage.py runserver
```

Você pode acessar o projeto em <http://127.0.0.1:8000/>.