

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.

1. Configuração Inicial

1.1 Instalação do Django e Criação do Projeto

1. Instale o Django:

```
pip install django
```

2. Crie o projeto:

```
django-admin startproject plataforma_cuidador  
cd plataforma_cuidador
```

3. Crie um aplicativo chamado "cuidados":

```
python manage.py startapp cuidados
```

4. Adicione o aplicativo à configuração em `settings.py`:

```
INSTALLED_APPS = [  
    # Apps padrão  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
  
    # App do projeto  
    'cuidados',  
]
```

2. Configurando o Modelo

2.1 Modelos

Edite o arquivo `cuidados/models.py`:

```
from django.db import models  
  
class Paciente(models.Model):  
    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) # Ex.: JSON para lista  
    de medicamentos  
  
    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}"

```

2.2 Migrações

1. Crie as migrações:

```
python manage.py makemigrations
```

2. Aplique as migrações:

```
python manage.py migrate
```

3. Configuração do Admin

Adicione os modelos ao admin em `cuidados/admin.py`:

```

from django.contrib import admin
from .models import Paciente, Lembrete, Relatorio

admin.site.register(Paciente)
admin.site.register(Lembrete)
admin.site.register(Relatorio)

```

4. Configurando as Views

4.1 Views do Paciente

No arquivo `cuidados/views.py`, crie as views principais:

```

from django.shortcuts import render, get_object_or_404, redirect
from django.http import JsonResponse
from .models import Paciente, Lembrete, Relatorio
from .forms import PacienteForm, LembreteForm, RelatorioForm

# Página inicial

```

```

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

# Cadastro de pacientes
def cadastrar_paciente(request):
    if request.method == "POST":
        form = PacienteForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('home')
    else:
        form = PacienteForm()
    return render(request, 'cuidados/cadastrar_paciente.html', {'form': form})

# Detalhes do paciente
def detalhes_paciente(request, id):
    paciente = get_object_or_404(Paciente, id=id)
    lembretes = paciente.lembretes.all()
    relatorios = paciente.relatorios.all()
    return render(request, 'cuidados/detalhes_paciente.html', {
        'paciente': paciente,
        'lembretes': lembretes,
        'relatorios': relatorios,
    })

# Cadastro de lembretes
def cadastrar_lembrete(request, id):
    paciente = get_object_or_404(Paciente, id=id)
    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})

```

5. Formulários

No arquivo cuidados/forms.py:

```

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']

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']
```

6. Templates

6.1 Base Template

Crie um arquivo `templates/cuidados/base.html`:

```
<!DOCTYPE html>
<html lang="en">
<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/bootstrap@5.3.0-alpha1/dist/css/
bootstrap.min.css" rel="stylesheet">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="{% url 'home' %}">Cuidador</a>
    </nav>
    <div class="container mt-5">
        {% block content %}
        {% endblock %}
    </div>
</body>
</html>
```

6.2 Página Inicial

Crie o arquivo `templates/cuidados/home.html`:

```
{% extends 'cuidados/base.html' %}
{% block content %}
<h1>Pacientes</h1>
<a href="{% url 'cadastrar_paciente' %}" class="btn btn-primary">Cadastrar
Paciente</a>
<table class="table mt-3">
    <thead>
        <tr>
            <th>Nome</th>
            <th>Ações</th>
        </tr>
    </thead>
    <tbody>
        {% for paciente in pacientes %}
        <tr>
            <td>{{ paciente.nome }}</td>
            <td>
                <a href="{% url 'detalhes_paciente' paciente.id %}" class="btn
btn-info">Detalhes</a>
            </td>
        </tr>
        {% endfor %}
    </tbody>
</table>
{% endblock %}
```