

```
from django.shortcuts import redirect, get_object_or_404
from django.contrib.auth import authenticate, login
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from .models import Sessao, Pauta
from django.http import HttpResponseRedirect
from camaraonapp.models import Vereador # Ajuste o caminho conforme
necessário
from django.shortcuts import redirect
from django.http import JsonResponse
from .models import Voto

# Função de login do administrador
def login_view(request):
    if request.method == "POST":
        username = request.POST.get("username") # CPF ou e-mail
        password = request.POST.get("password")

        user = authenticate(request, username=username, password=password)

        if user is not None:
            login(request, user)
            return redirect("admin:index") # Redireciona para o painel admin
        else:
            messages.error(request, "Usuário ou senha inválidos!")

    return render(request, "login.html") # Renderiza a página de login
```

Função de login do vereador

def login_vereador(request):

if request.method == "POST":

cpf = request.POST.get('username') # Obtendo CPF do formulário

senha = request.POST.get('password') # Obtendo senha do formulário

print("Tentando login com CPF:", cpf) # 🔴 DEBUG

if not cpf or not senha: # Verificando se os campos estão vazios

messages.error(request, "CPF e senha são obrigatórios.")

return render(request, 'painel_vereador/login_vereador.html')

try:

vereador = Vereador.objects.get(user=request.user) # Buscando vereador no banco

print("✅ Vereador encontrado:", vereador.nome)

except Vereador.DoesNotExist:

messages.error(request, "CPF não encontrado.")

print("❌ Erro: CPF não encontrado.") # 🔴 DEBUG

return render(request, 'painel_vereador/login_vereador.html')

Autenticando

user = authenticate(request, username=cpf, password=senha)

if user is not None:

login(request, user)

print("✅ Login bem-sucedido! Redirecionando...") # 🔴 DEBUG

```
        return redirect('registro_presenca') # ● Redireciona para pautas em
votação
```

```
    else:
```

```
        messages.error(request, "Senha incorreta.")
```

```
        print("✗ Erro: Senha incorreta.") # ● DEBUG
```

```
        return render(request, 'painel_vereador/login_vereador.html')
```

```
print("Exibindo página de login...") # ● DEBUG
```

```
return render(request, 'painel_vereador/login_vereador.html')
```

```
# Página inicial do painel do vereador
```

```
@login_required
```

```
def index_vereador(request):
```

```
    sessoes = Sessao.objects.all() # Pegando todas as sessões
```

```
    pautas = Pauta.objects.filter(sessao__in=sessoes) # Filtrando pautas
associadas a essas sessões
```

```
    return render(request, 'painel_vereador/pautas_em_votacao.html', {
```

```
        'sessoes': sessoes, # Passando as sessões para o template
```

```
        'pautas': pautas
```

```
    })
```

```
# Registrar presença do vereador
```

```
from django.shortcuts import render, redirect, get_object_or_404
```

```
from django.contrib.auth.decorators import login_required
```

```
from django.contrib import messages
```

```
from .models import Vereador # Certifique-se de importar o modelo correto
```

```
@login_required

def registro_presenca(request):

    # Obtém o vereador logado

    vereador = get_object_or_404(Vereador, user=request.user)


    if request.method == "POST":

        # Marca o vereador como presente

        vereador.presente = True # Certifique-se de que o campo 'presente' existe no
        modelo

        vereador.save()

        messages.success(request, "Presença registrada com sucesso!")


        return redirect("pautas_em_votacao") # Nome correto da URL das pautas


    return render(request, "registro_presenca.html", {"vereador": vereador})
```

Exibir os detalhes da pauta para votação

```
@login_required
```

```
def detalhes_pauta(request, pauta_id):

    pauta = get_object_or_404(Pauta, id=pauta_id)

    if request.method == 'POST':

        voto = request.POST['voto']

        # Aqui você pode processar o voto, por exemplo, salvar no banco

        # Voto.objects.create(vereador=request.user, pauta=pauta, voto=voto)

        messages.success(request, "Voto realizado com sucesso!")

        return redirect('pautas_em_votacao') # Redirecionar de volta após votar
```

```
return render(request, 'painel_vereador/detalhe_pauta.html', {'pauta': pauta})
```

```
def custom_login(request):
```

```
    if request.method == 'POST':
```

```
        username = request.POST['username']
```

```
        password = request.POST['password']
```

```
        user = authenticate(request, username=username, password=password)
```

```
        if user is not None:
```

```
            login(request, user)
```

```
            return HttpResponseRedirect('/admin') # Redirecionamento após login bem-  
sucedido
```

```
        else:
```

```
            # Se falhar, exibe a mensagem de erro na mesma página
```

```
            return render(request, 'login.html', {'error': 'Usuário ou senha inválidos'})
```

```
    else:
```

```
        # Se a requisição for GET, apenas renderize o formulário de login
```

```
        return render(request, 'login.html')
```

```
@login_required
```

```
def pautas_em_votacao(request):
```

```
    pautas = Pauta.objects.all()
```

```
    # Filtra as pautas que estão em votação
```

```
    return render(request, 'painel_vereador/pautas_em_votacao.html', {'pautas':  
pautas})
```

```
@login_required
```

```
def painel_votacao(request, pauta_id):
```

```
    pauta = get_object_or_404(Pauta, id=pauta_id)
```

```
vereador = get_object_or_404(Vereador, user=request.user) # Obtém o vereador  
logado
```

```
# Verifica se o vereador é o presidente
```

```
is_presidente = vereador.funcao == "Presidente"
```

```
if request.method == 'POST':
```

```
    voto = request.POST.get('voto')
```

```
    if voto:
```

```
        # Salvar o voto (exemplo de modelo Voto)
```

```
        # Voto.objects.create(vereador=vereador, pauta=pauta, voto=voto)
```

```
        messages.success(request, "Voto realizado com sucesso!")
```

```
        return redirect('pautas_em_votacao')
```

```
return render(request, 'painel_vereador/painel_votacao.html', {
```

```
    'pauta': pauta,
```

```
    'is_presidente': is_presidente # Passando a variável para o template
```

```
})
```

```
def detalhe_pauta(request, id):
```

```
    pauta = get_object_or_404(Pauta, id=id)
```

```
    return render(request, 'detalhe_pauta.html', {'pauta': pauta, 'pdf_url':  
pauta.anexo_pdf.url if pauta.anexo_pdf else None})
```

```
def colocar_pauta_em_votacao(request, pauta_id):
```

```
    # Lógica da função aqui
```

```
    return HttpResponse("Pauta colocada em votação.")
```

```
@login_required
```

```
def colocar_pauta_em_votacao(request, pauta_id):

    vereador = request.user.vereador # Pegando o vereador logado

    if vereador.funcao != "presidente": # Somente o presidente pode iniciar a
votação

        return HttpResponseRedirect("Apenas o presidente pode iniciar a votação.")

    pauta = get_object_or_404(Pauta, id=pauta_id)

    pauta.em_votacao = True # Alterando status para votação iniciada

    pauta.save()

    return redirect("painel_vereador") # Redireciona para o painel atualizado
```

@login_required

```
def recolocar_pauta_em_votacao(request, pauta_id):

    pauta = get_object_or_404(Pauta, id=pauta_id)

    # ♦ Bloquear se não for presidente

    if request.user.vereador.funcao != "presidente":

        messages.error(request, "Apenas o Presidente pode recolocar a pauta.")

        return redirect('pautas_em_votacao')

    # Desativar votação

    pauta.em_votacao = False

    pauta.save()

    messages.success(request, "A pauta foi retirada de votação.")

    return redirect('pautas_em_votacao')
```

```
from django.shortcuts import render

from .models import Pauta, Vereador


def painel_vereador(request):

    from django.shortcuts import get_object_or_404

    pauta = Pauta.objects.latest('id') # Pegando a última pauta cadastrada

    vereador = getattr(request.user, 'vereador', None) # Pega o vereador associado
    ao usuário

    is_presidente = vereador and vereador.funcao == 'presidente' # Verifica se o
    vereador é presidente

    print(f"Usuário: {request.user}, Presidente: {is_presidente}, Pauta em votação:
    {pauta.em_votacao}") # Teste no terminal

    return render(request, 'painel_vereador.html', {'pauta': pauta, 'is_presidente':
    is_presidente})


def registrar_voto(request, pauta_id):

    if request.method == "POST":

        vereador = request.user # Pegando o vereador logado

        pauta = Pauta.objects.get(id=pauta_id)

        voto = request.POST.get("voto") # Recebe o valor do voto (Sim, Não,
        Abstenção)

        # Salva o voto no banco

        Voto.objects.create(vereador=vereador, pauta=pauta, escolha=voto)
```



```
# Redireciona para a página de pautas em votação
return redirect("pautas_em_votacao")

return JsonResponse({"erro": "Método inválido"}, status=400)
```

```
from django.urls import path
from django.conf import settings
from django.conf.urls.static import static
from . import views
from .views import registro_presenca
from .views import colocar_pauta_em_votacao
from .views import recolocar_pauta_em_votacao
from .views import painel_votacao, registrar_voto

urlpatterns = [

    path('login/', views.login_view, name='login'), # Rota única para login

    path('login_vereador/', views.login_vereador, name='login_vereador'), # Login do
vereador

    path('pautas_em_votacao/', views.pautas_em_votacao,
name='pautas_em_votacao'), # Página de pautas em votação

    path('detalhes_pauta/<int:pauta_id>/', views.detalhes_pauta,
name='detalhes_pauta'),

    path('painel_votacao/<int:pauta_id>/', views.painel_votacao,
name='painel_votacao'), # Página de votação

    path('pauta/<int:id>/', views.detalhe_pauta, name='detalhe_pauta'),

    path("registro_presenca/", registro_presenca, name="registro_presenca"),
```

```

    path('registro_presenca/', registro_presenca, name='registrar_presenca'),

    path('registro_presenca/<int:vereador_id>/', registro_presenca,
name='registrar_presenca'),

    path("registro_presenca/", views.registro_presenca, name="registro_presenca"),

    path("pautas/", views.pautas_em_votacao, name="pautas_em_votacao"), #
Ajuste se necessário

    path('painel_votacao/<int:pauta_id>/', views.painel_votacao,
name='painel_votacao'),

    path('colocar_pauta_em_votacao/<int:pauta_id>/', colocar_pauta_em_votacao,
name='colocar_pauta_em_votacao'),

    path('recolocar_pauta_em_votacao/<int:pauta_id>/',
recolocar_pauta_em_votacao, name='recolocar_pauta_em_votacao'),

    path('registrar-voto/<int:pauta_id>/', registrar_voto, name='registrar_voto'),

]

```

Adicionar configuração de mídia apenas quando o DEBUG estiver ativo

if settings.DEBUG:

```

    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)

```

from django.contrib.auth.models import User

from django.db import models

from django.db import models

from django.contrib.auth.models import User # Importar o modelo User

```

class Vereador(models.Model):

    FUNCAO_CHOICES = [

        ('Vereador', 'Vereador'),

        ('Secretário', 'Secretário'),

        ('Presidente', 'Presidente'),

        ('Vice-Presidente', 'Vice-Presidente'),

    ]

    user = models.OneToOneField(User, on_delete=models.CASCADE) #
Relacionamento com User

    nome = models.CharField(max_length=255)

    partido = models.CharField(max_length=100)

    cpf = models.CharField(max_length=14, unique=True)

    telefone = models.CharField(max_length=15)

    senha = models.CharField(max_length=255)

    foto = models.ImageField(upload_to='fotos_vereadores/', blank=True, null=True)

    funcao = models.CharField(max_length=50, choices=FUNCAO_CHOICES,
default='Vereador')

    # Novo campo para indicar se é presidente

    is_presidente = models.BooleanField(default=False)

    def save(self, *args, **kwargs):

        # Atualiza automaticamente o campo is_presidente com base na função
escolhida

        self.is_presidente = self.funcao == 'Presidente'

        super().save(*args, **kwargs)

    def __str__(self):

```

```
return self.nome
```

```
class Sessao(models.Model):  
  
    nome_sessao = models.CharField(max_length=255)  
  
    data = models.DateTimeField()  
  
    horario = models.TimeField()  
  
  
    def __str__(self):  
  
        return self.nome_sessao
```

```
class Pauta(models.Model):  
  
    TIPO_PAUTA_CHOICES = [  
  
        ('Projeto de Lei', 'Projeto de Lei'),  
  
        ('Moção de Aplausos', 'Moção de Aplausos'),  
  
        ('Louvor', 'Louvor'),  
  
        ('Congratulação', 'Congratulação'),  
  
        ('Pesar', 'Pesar'),  
  
        ('Repúdio', 'Repúdio'),  
  
        ('Desagravo', 'Desagravo'),  
  
        ('Indicações', 'Indicações'),  
  
        ('Requerimentos', 'Requerimentos'),  
  
    ]
```

```
nome = models.CharField(max_length=255)  
  
descricao = models.TextField()  
  
sessao = models.ForeignKey(Sessao, on_delete=models.CASCADE)
```

```
    tipo = models.CharField(max_length=50, choices=TIPO_PAUTA_CHOICES,
default='Projeto de Lei')

    anexo_pdf = models.FileField(upload_to='pautas/', null=True, blank=True)

    autor = models.CharField(max_length=255)

    em_votacao = models.BooleanField(default=False) # Indica se a pauta está em
votação
```

```
def __str__(self):

    return self.nome
```

```
from django.db import models
```

```
class Voto(models.Model):
```

```
    SIM = 'Sim'
```

```
    NAO = 'Não'
```

```
    ABSTENCAO = 'Abstenção'
```

```
VOTO_CHOICES = [
```

```
    (SIM, 'Sim'),
```

```
    (NAO, 'Não'),
```

```
    (ABSTENCAO, 'Abstenção')
```

```
]
```

```
pauta = models.ForeignKey('Pauta', on_delete=models.CASCADE)
```

```
vereador = models.ForeignKey('Vereador', on_delete=models.CASCADE)
```

```
voto = models.CharField(max_length=10, choices=VOTO_CHOICES)
```

```
data_voto = models.DateTimeField(auto_now_add=True)
```

```
class Meta:
```

```
unique_together = ('pauta', 'vereador')
```

```
def __str__(self):
```

```
    return f"Voto de {self.vereador} na pauta {self.pauta}"
```

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Painel do Vereador</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Painel do Vereador</h1>
```

```
    <p>Nome: {{ vereador.nome }}</p>
```

```
    <p>Partido: {{ vereador.partido }}</p>
```

```
    <p>Função: {{ vereador.funcao }}</p>
```

```
<h2>Pautas em andamento</h2>
```

```
<ul>
```

```
    {% for pauta in pautas %}
```

```
        <li>
```

```
            {{ pauta.nome }} - {{ pauta.status }}
```

```
            {% if presidente %}
```

```
        <form method="POST" action="{% url 'colocar_pauta_em_votacao'
pauta.id %}">

            {% csrf_token %}

            <button type="submit" class="btn btn-success">Colocar em
Votação</button>

        </form>

    {% endif %}

</li>

{% endfor %}

</ul>


<a href="{% url 'logout' %}">Sair</a>

</body>

</html>
```

```
{% load static %}

<!DOCTYPE html>

<html lang="pt">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Painel do Vereador</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            margin: 0;
```

```
padding: 0;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;
}

.painel-vereador {

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

max-width: 600px;

width: 100%;

text-align: center;
}

.informacoes-vereador {

display: flex;

align-items: center;

gap: 15px;

border-bottom: 2px solid #ddd;

padding-bottom: 15px;

margin-bottom: 15px;
}

.informacoes-vereador img {

width: 60px;

height: 60px;

border-radius: 50%;

border: 2px solid #007bff;
```



```
}  
  
.detalhes-pauta {  
    background: #f9f9f9;  
    padding: 15px;  
    border-radius: 5px;  
    margin-bottom: 15px;  
}  
  
.acoes-presidente {  
    margin-bottom: 15px;  
}  
  
.botao-acao {  
    display: block;  
    width: 100%;  
    padding: 10px;  
    margin-top: 10px;  
    font-size: 16px;  
    font-weight: bold;  
    cursor: pointer;  
    border: none;  
    border-radius: 5px;  
}  
  
.botao-colocar {  
    background-color: #28a745;  
    color: white;  
}  
  
.botao-recolocar {  
    background-color: #dc3545;  
    color: white;
```

```
}  
  
.votacao {  
    margin-top: 20px;  
}  
  
.botoes-votacao {  
    display: flex;  
    justify-content: space-between;  
    gap: 10px;  
}  
  
.botao-voto {  
    flex: 1;  
    padding: 10px;  
    border-radius: 5px;  
    border: none;  
    font-size: 16px;  
    font-weight: bold;  
    cursor: pointer;  
}  
  
.voto-sim {  
    background-color: #28a745;  
    color: white;  
}  
  
.voto-nao {  
    background-color: #dc3545;  
    color: white;  
}  
  
.voto-abstencao {  
    background-color: #ffc107;
```

```

        color: black;
    }

    .aguardando-votacao {
        color: #888;
        font-size: 14px;
    }
</style>
</head>
<body>
    <div class="painel-vereador">
        <!-- Informações do Vereador -->
        <div class="informacoes-vereador">
            
            <div class="info-texto">
                <p><strong>{{ vereador.nome }}</strong> - {{ vereador.partido }}</p>
                <p>Função: <strong>{{ vereador.get_funcao_display }}</strong></p>
            </div>
        </div>

        <!-- Detalhes da Pauta -->
        <div class="detalhes-pauta">
            <h3>Detalhes da Pauta</h3>
            <p><strong>{{ pauta.nome }}</strong></p>
            <p><em>Descrição:</em> {{ pauta.descricao }}</p>
            {% if pauta.anexo %}
                <p><em>Anexo:</em> <a href="{{ pauta.anexo.url }}"
target="_blank">Baixar Documento PDF</a></p>
            {% endif %}
        </div>
    </div>
</body>
</html>

```

</div>

<!-- Ações para o Presidente -->

{% if is_presidente %}

<div class="acoes-presidente">

{% if not pauta.em_votacao %}

<form method="POST" action="{% url 'colocar_pauta_em_votacao' pauta.id %}">

{% csrf_token %}

<button type="submit" class="botao-acao botao-colocar">Colocar Pauta em Votação</button>

</form>

{% else %}

<form method="POST" action="{% url 'recolocar_pauta_em_votacao' pauta.id %}">

{% csrf_token %}

<button type="submit" class="botao-acao botao-recolocar">Recolocar Pauta em Votação</button>

</form>

{% endif %}

</div>

{% endif %}

<!-- Votação -->

<div class="votacao">

<h3>Votação</h3>

{% if pauta.em_votacao %}

<form method="POST" action="{% url 'registrar_voto' pauta.id %}">

{% csrf_token %}

```
<div class="botoes-votacao">

    <button type="submit" name="voto" value="sim" class="botao-voto
voto-sim">Sim</button>

    <button type="submit" name="voto" value="nao" class="botao-voto
voto-nao">Não</button>

    <button type="submit" name="voto" value="abstencao" class="botao-
voto voto-abstencao">Abstenção</button>

</div>

</form>

{% else %}

    <p class="aguardando-votacao">Aguardando o Presidente iniciar a
votação.</p>

    {% endif %}

</div>

</div>

</body>

</html>
```

```
{% load static %}

<!DOCTYPE html>

<html lang="pt">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Listagem de Pautas</title>

    <link rel="stylesheet" href="{% static 'pautas_em_votacao.css' %}">
```

```
<style>
```

```
/* Estilo geral */
```

```
body {
```

```
    font-family: Arial, sans-serif;
```

```
    text-align: center;
```

```
}
```

```
/* Centraliza todo o conteúdo */
```

```
.painel-vereador {
```

```
    width: 80%;
```

```
    margin: 0 auto;
```

```
}
```

```
h2 {
```

```
    text-align: center; /* Mantém os títulos centralizados */
```

```
}
```

```
p {
```

```
    text-align: justify; /* Apenas o texto das pautas fica justificado */
```

```
}
```

```
/* Estilo do modal */
```

```
.modal {
```

```
    display: none;
```

```
    position: fixed;
```

```
    top: 0;
```

```
    left: 0;
```

```
    width: 100%;
```

```
height: 100%;  
background-color: rgba(0, 0, 0, 0.7);  
justify-content: center;  
align-items: center;  
}
```

```
.modal-content {  
    background: white;  
    padding: 20px;  
    width: 80%;  
    height: 80%;  
    position: relative;  
    border-radius: 8px;  
}
```

```
.modal iframe {  
    width: 100%;  
    height: 90%;  
    border: none;  
}
```

```
.close-modal {  
    position: absolute;  
    top: 10px;  
    right: 15px;  
    font-size: 20px;  
    font-weight: bold;  
    cursor: pointer;
```

```
    color: red;
}
```

```
/* Estilização dos botões */
```

```
.botoes-pauta {
    display: flex;
    justify-content: center; /* Centraliza os botões */
    gap: 10px; /* Espaço entre os botões */
    margin-top: 10px;
}
```

```
.botao-acao {
    padding: 12px 20px;
    width: 180px; /* Definição de largura fixa */
    text-align: center;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    font-weight: bold;
}
```

```
.botao-previsualizar {
    background-color: #007bff;
    color: white;
}
```

```
.botao-acessar {
```



```
background-color: #28a745;
color: white;
text-decoration: none;
display: flex;
align-items: center;
justify-content: center;
}
```

```
.botao-votar {
background-color: #ff9800;
color: white;
}
```

```
.botao-presidencia {
background-color: #dc3545;
color: white;
}
```

```
.botao-acessar:hover, .botao-previsualizar:hover, .botao-votar:hover, .botao-
presidencia:hover {
opacity: 0.8;
}
```

```
</style>
```

```
</head>
```

```
<script src="{% static 'js/pautas.js' %}"></script>
```

```
<body>
```

```
<div class="painel-vereador">
```

```
{% if sessao %}
```

```

<h2>Pautas da Sessão: {{ sessao.nome }}</h2>

<p><strong>Data:</strong> {{ sessao.data }}</p>

{% else %}

<h2 style="color: red;">Atenção! Nenhuma sessão disponível no
momento.</h2>

{% endif %}


<div class="lista-pautas">

    {% for pauta in pautas %}

        <h2>{{ pauta.nome }}</h2>

        <p>{{ pauta.descricao }}</p>


        {% if pauta.anexo_pdf %}

            <p><strong>PDF Anexo:</strong></p>

            <div class="botoes-pauta">

                <button class="botao-acao botao-previsualizar"
onclick="abrirModal('{{ pauta.anexo_pdf.url }}')">

                    Visualizar Pauta

                </button>

                <a href="{{ pauta.anexo_pdf.url }}" target="_blank" class="botao-acao
botao-acessar">

                    Baixar PDF

                </a>

                <a href="{% url 'painel_votacao' pauta.id %}" class="botao-acao botao-
votar">

                    Votar Pauta

                </a>

            </a>

            {% if usuario.is_presidente %}

```

```
{% if pauta.votacao_aberta %}

    <a href="{% url 'encerrar_votacao' pauta.id %}" class="botao-acao
botao-presidencia">

        Encerrar Votação

    </a>

{% else %}

    <a href="{% url 'abrir_votacao' pauta.id %}" class="botao-acao
botao-presidencia">

        Abrir Votação

    </a>

{% endif %}

{% endif %}

</div>

{% else %}

    <p>Sem anexo</p>

{% endif %}

{% endfor %}

</div>

</div>
```

<!-- Modal para pré-visualização do PDF -->

<div class="modal" id="modalPdf">

<div class="modal-content">

X

<iframe id="iframePdf" src="" frameborder="0"></iframe>

</div>

</div>

<script>

```
function abrirModal(pdfUrl) {  
    document.getElementById('iframePdf').src = pdfUrl;  
    document.getElementById('modalPdf').style.display = 'flex';  
}  
  
function fecharModal() {  
    document.getElementById('modalPdf').style.display = 'none';  
    document.getElementById('iframePdf').src = ""; // Limpa o iframe ao fechar  
}  
</script>  
</body>  
</html>
```