

AULA 2 | PSW - O RETORNO

▼ Configurações iniciais

Crie um novo app chamado divulgar:

```
python manage.py startapp divulgar
```

Instale o app!

Crie uma url para apontar para divulgar

```
path('divulgar/', include('divulgar.urls')),
```

▼ Novo pet

Cria a model Tag:

```
class Tag(models.Model): tag = models.CharField(max_length=100) def __str__(self):  
    return self.tag
```

Crie a model Raca:

```
class Raca(models.Model): raca = models.CharField(max_length=50) def __str__(self):  
    return self.raca
```

Crie a model Pet:

```
class Pet(models.Model): choices_status = (('P', 'Para adoção'), ('A', 'Adotado'))
usuario = models.ForeignKey(User, on_delete=models.DO_NOTHING) foto =
models.ImageField(upload_to="fotos_pets") nome = models.CharField(max_length=100)
descricao = models.TextField() estado = models.CharField(max_length=50) cidade =
models.CharField(max_length=50) telefone = models.CharField(max_length=50) tags =
models.ManyToManyField(Tag) raca = models.ForeignKey(Raca,
on_delete=models.DO_NOTHING) status = models.CharField(max_length=1,
choices=choices_status, default='P') def __str__(self): return self.nome
```

Faça as migrações:

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

Crie a URL para novo_pet

```
from django.urls import path from . import views urlpatterns = [ path('novo_pet/',
views.novo_pet, name="novo_pet"), ]
```

crie a view novo_pet:

```
from django.contrib.auth.decorators import login_required @login_required def  
novo_pet(request): if request.method == "GET": return render(request,  
'novo_pet.html')
```

Crie o html novo_pet:

```
{% load static %} <!doctype html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Cadastro</title> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet"> <link href="{% static
'usuarios/css/cadastro.css' %}" rel="stylesheet"> <link href="{% static
'divulgar/css/novo_pet.css' %}" rel="stylesheet"> </head> <body> <div
class="container-fluid"> <div class="row"> <div class="col-md-2 bg-side sidebar">
<h2>ADO.TE</h2> <hr> </div> <div class="col-md"> <div class="wrapper"> <div
class="box"> <div class="header-box"> <h3 class="font-vinho">Quero divulgar</h3>
<hr> </div> <div class="body-box"> <form action="" method="POST"
enctype='multipart/form-data'>{% csrf_token %} {% if messages %} <br> {% for message
in messages %} <div class="alert {{message.tags}}"> {{message}} </div> {% endfor %}
{% endif %} <input type="file" name="foto" class="form-control"> <br> <span>Nome:
<span> <input type="text" class="form-control" name="nome" placeholder="Digite o
nome do pet..."> <br> <span>Descrição:<span> <textarea class="form-control"
name="descricao" placeholder="Descrição..."></textarea> <br> <div class="row"> <div
class="col-md"> <span>Estado:</span> <input type="text" placeholder="Digite o estado
do pet..." name="estado" class="form-control"> </div> <div class="col-md">
<span>Cidade:</span> <input type="text" placeholder="Digite a cidade do pet..."
name="cidade" class="form-control"> </div> </div> <br> <span>Telefone para contato:
</span> <input type="text" placeholder="Digite o seu telefone..." name="telefone"
class="form-control"> <br> <span>Tags:</span> <select name="tags" multiple
class="form-select"> </select> <br> </span>Raça:</span> <select name="raca"
class="form-select"> </select> <input type="submit" value="ENVIAR" class="btn-
custom-primary"> </form> </div> </div> </div> </div> </div> </div> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script> </body> </html>
```

Crie o novo_pet.css:

```
.sidebar{ text-align: center; padding: 20px; } .wrapper { margin-top: 3%; display: flex; flex-direction: row; justify-content: center; align-items: center; } .box { padding: 20px; width: 50vw; background: rgb(238, 238, 238); box-shadow: 4px 4px 10px rgba(0,0,0,.3); } .font-vinho{ color: #A24D52; }
```

Busque as tags e as raças nas views:

```
tags = Tag.objects.all() racas = Raca.objects.all()
```

Envie os dados para o HTML:

```
return render(request, 'novo_pet.html', {'tags':tags, 'racas':racas})
```

No html imprima as tags:

```
{% for tag in tags%} <option value="{{tag.id}}">{{tag}}</option> {% endfor %}
```

No html imprima as racas:

```
{% for raca in racas%} <option value="{{raca.id}}">{{raca}}</option> {% endfor %}
```

Cadastre no admin:

```
from .models import * admin.site.register(Raca) admin.site.register(Tag)
admin.site.register(Pet)
```

Envie os dados do form para a view novo_pet:

```
<form action="{% url 'novo_pet' %}" method="POST" enctype='multipart/form-data'>{%
csrf_token %}
```


Faça o cadastro do Pet na view novo_pet:

```
from django.shortcuts import render from django.contrib.auth.decorators import
login_required from .models import Pet, Tag, Raca from django.contrib import
messages from django.contrib.messages import constants @login_required def
novo_pet(request): if request.method == "GET": tags = Tag.objects.all() racas =
Raca.objects.all() return render(request, 'novo_pet.html', {'tags': tags, 'racas':
racas}) elif request.method == "POST": foto = request.FILES.get('foto') nome =
request.POST.get('nome') descricao = request.POST.get('descricao') estado =
request.POST.get('estado') cidade = request.POST.get('cidade') telefone =
request.POST.get('telefone') tags = request.POST.getlist('tags') raca =
request.POST.get('raca') #TODO: Validar dados pet = Pet( usuario=request.user,
foto=foto, nome=nome, descricao=descricao, estado=estado, cidade=cidade,
telefone=telefone, raca_id=raca, ) pet.save() for tag_id in tags: tag =
Tag.objects.get(id=tag_id) pet.tags.add(tag) pet.save() tags = Tag.objects.all()
racas = Raca.objects.all() messages.add_message(request, constants.SUCCESS, 'Novo
pet cadastrado') return render(request, 'novo_pet.html', {'tags': tags, 'racas':
racas})
```

Adicione a URL para os arquivos de media:

```
from django.conf import settings from django.conf.urls.static import static +
static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```


▼ Listar pets

Crie uma url para listar os pets:

```
path('seus_pets/', views.seus_pets, name="seus_pets"),
```

Crie a view para exibir os pets:

```
@login_required def seus_pets(request): if request.method == "GET": pets =  
Pet.objects.filter(usuario=request.user) return render(request, 'seus_pets.html',  
{ 'pets': pets })
```

Vamos para o arquivo seus_pets.html:

```
{% load static %} <!doctype html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Cadastro</title> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet"> <link href="{% static
'usuarios/css/cadastro.css' %}" rel="stylesheet"> <link href="{% static
'divulgar/css/novo_pet.css' %}" rel="stylesheet"> <link href="{% static
'divulgar/css/seus_pet.css' %}" rel="stylesheet"> </head> <body> <div
class="container-fluid"> <div class="row"> <div class="col-md-2 bg-side sidebar">
<h2>ADO.TE</h2> <hr> </div> <div class="col-md"> <div class="wrapper"> <div
class="box width80"> {% if messages %} <br> {% for message in messages %} <div
class="alert {{message.tags}}"> {{message}} </div> {% endfor %} {% endif %} <h3
class="font-vinho">Quero divulgar</h3> <hr> <table class="tabela" cellpadding="20">
<tr> <th>Foto</th> <th>Nome</th> <th>Raça</th> <th>Status</th> <th>Remover</th>
</tr> <tr class="tabela-linha"> <td width="20%">  </td> <td> Nome </td> <td> Raca </td> <td> status </td> <td> <a
href="" class="btn btn-danger">REMOVER</a> </td> </tr> </table> </div> </div> </div>
</div> </div> <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script> </body> </html>
```

Crie o seus_pet.css:

```
.width80{ width: 80vw; } .tabela{ width: 100%; margin-top: 20px; } .tabela-linha{
background-color: #e8e7e7; }
```

Exiba os pets dinamicamente no html:

```
{% for pet in pets%} <tr class="tabela-linha"> <td width="20%">  </td> <td> {{pet.nome}} </td> <td> {{pet.raca}} </td> <td>
{% if pet.status == "P" %} <span class="badge text-bg-primary">Para adoção</span> {%
elif pet.status == "A" %} <span class="badge text-bg-success">Adotado</span> {%
endif%} </td> <td> <a href="" class="btn btn-danger">REMOVER</a> </td> </tr> {%
endfor %}
```

▼ Remover pet

Crie a URL para remover um pet:

```
path('remover_pet/<int:id>', views.remover_pet, name="remover_pet"),
```

Crie a view para deletar um pet:

```
@login_required def remover_pet(request, id): pet = Pet.objects.get(id=id)
pet.delete() messages.add_message(request, constants.SUCCESS, 'Removido com
sucesso.') return redirect('/divulgar/seus_pets')
```

Verifique se o pet que está sendo deletado é realmente de quem está tentando deletar:

```
if not pet.usuario == request.user: messages.add_message(request, constants.ERROR,
'Esse pet não é seu!') return redirect('/divulgar/seus_pets')
```

No botão remover, redirecione para view criada:

```
{% url 'remover_pet' pet.id %}
```

▼ Buscar pets

Crie um novo app:

```
python manage.py startapp adotar
```

INSTALE O APP!

Crie a URL para adotar: