



AULA 2 | PSW 7.0

▼ Extrato novo valor

Crie um app para o extrato

```
python3 manage.py startapp extrato
```

Instale o app!

Crie uma URL para o extrato:

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

Crie uma URL para página de novo_valor:

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

Crie a view novo_valor:

```
def novo_valor(request): if request.method == "GET": contas = Conta.objects.all() categorias = Categoria.objects.all() return render(request, 'novo_valor.html', {'contas': contas, 'categorias': categorias})
```

Crie o html novo_valor:

```

{%
    extends 'bases/base.html' %
} {%
    load static %
} {%
    block 'head' %
}
<link href="{% static 'perfil/css/home.css' %}" rel="stylesheet">
<link href="{% static 'perfil/css/gerenciar.css' %}" rel="stylesheet">
{%
    endblock %
} {%
    block 'body' %
}<div class="container"> <br> {%
    if
messages %
} {%
    for message in messages %
}<div class="alert {{ message.tags }}>{{ message }}</div> {%
    endfor %
} {%
    endif %
}<br>
<span class="fonte-destaque">Adicionar entradas/saídas</span> <div
class="row"> <div class="col-md-7"> <form action="" method="POST">{%
    csrf_token %
}<div class="row"> <div class="col-md">
<label>Valor</label> <input name="valor" type="text" class="form-
control"> </div> <div class="col-md"> <label>Categoria</label> <select
name="categoria" class="form-select"> <option value="1">Lazer</option>
</select> </div> <br> <label>Descrição</label> <textarea
name="descricao" class="form-control"></textarea> <br> <div
class="row"> <div class="col-md"> <label>Data</label> <input
name="data" type="date" class="form-control"> </div> <div class="col-
md"> <label>Conta</label> <select name="conta" class="form-select">
<option value="1">Nubank</option> </select> </div> <br> <input
type="radio" name="tipo" value="E"> <label
class="positivo">Entrada</label> <input type="radio" name="tipo"
value="S"> <label class="negativo">Saída</label> <br> <br> <input
type="submit" style="width:40%;" class="botao-principal"> </form>
</div> <div class="col-md-8"> </div> </div> </div> {%
    endblock %
}

```

Crie as models Valores:

```

class Valores(models.Model): choice_tipo = ( ('E', 'Entrada'), ('S',
'Saída') ) valor = models.FloatField() categoria =
models.ForeignKey(Categoria, on_delete=models.DO_NOTHING) descricao =
models.TextField() data = models.DateField() conta =
models.ForeignKey(Conta, on_delete=models.DO_NOTHING) tipo =
models.CharField(max_length=1, choices=choice_tipo) def __str__(self):
return self.descricao

```

Execute as migrações

Envie os dados do FORM para novo_valor:

```
<form action="{% url 'novo_valor' %}" method="POST">{% csrf_token %}
```

Exiba as categorias dinamicamente:

```
<select name="categoria" class="form-select"> {% for categoria in categorias %} <option value="{{categoria.id}}>{{categoria}}</option> {% endfor %} </select>
```

Exiba as contas dinamicamente:

```
<select name="conta" class="form-select"> {% for conta in contas %} <option value="{{conta.id}}>{{conta}}</option> {% endfor %} </select>
```

Crie a view novo_valor para salvar os dados:

```
def novo_valor(request): if request.method == "GET": contas = Conta.objects.all() categorias = Categoria.objects.all() return render(request, 'novo_valor.html', {'contas': contas, 'categorias': categorias}) elif request.method == "POST": valor = request.POST.get('valor') categoria = request.POST.get('categoria') descricao = request.POST.get('descricao') data = request.POST.get('data') conta = request.POST.get('conta') tipo = request.POST.get('tipo') valores = Valores( valor=valor, categoria_id=categoria, descricao=descricao, data=data, conta_id=conta, tipo=tipo, ) valores.save() conta = Conta.objects.get(id=conta) if tipo == 'E': conta.valor += int(valor) else: conta.valor -= int(valor) conta.save() messages.add_message(request, constants.SUCCESS, 'Categoria cadastrada com sucesso') return redirect('/extrato/novo_valor')
```

▼ Extrato view extrato

Crie a URL para view_extrato:

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

Crie a view de view_extrato:

```
def view_extrato(request): contas = Conta.objects.all() categorias = Categoria.objects.all() valores = Valores.objects.filter(data__month=datetime.now().month) return render(request, 'view_extrato.html', {'valores': valores, 'contas': contas, 'categorias': categorias})
```

Crie o HTML view_extrato:

```
{% extends 'bases/base.html' %} {% load static %} {% block 'head' %}  
<link href="{% static 'perfil/css/home.css' %}" rel="stylesheet">  
<link href="{% static 'perfil/css/gerenciar.css' %}" rel="stylesheet">  
<link href="{% static 'extrato/css/view_extrato.css' %}"  
rel="stylesheet"> {% endblock %} {% block 'body' %} <br> <br> <div  
class='container'> <form action="" method="GET"> <div class="row">  
<div class="col-md"> <label>Conta</label> <select name="conta"  
class="form-select"> {% for conta in contas %} <option value="  
{{conta.id}}">{{conta}}</option> {% endfor %} </select> </div> <div  
class="col-md"> <label>Categoria</label> <select name="categoria"  
class="form-select"> {% for categoria in categorias %} <option value="  
{{categoria.id}}">{{categoria}}</option> {% endfor %} </select> </div>  
<div class="col-md"> <label>Período</label> <select name="periodo"  
class="form-select"> <option>Últimos 7 dias</option> </select> </div>  
</div> <br> <div class="row"> <div class="col-md-2"> <input  
style="width: 100%" type="submit" class="botao-principal"  
value="Filtrar"> </div> </form> <div class="col-md-2"> <a href=""  
class="botao-secundario">Exportar extrato</a> </div> </div> <br> <br>  
<div class="card"> <table> <tr> <th>Conta</th> <th>Categoria</th>  
<th>Data</th> <th>Tipo</th> <th>valor</th> </tr> {% for valor in  
valores %}<tr class="linha"> <td width="10%">{{valor.conta}}</td>  
<td>{{valor.categoria}}</td> <td>{{valor.data}}</td> <td> {% if  
valor.tipo == 'S'%}   
{% else %}  {% endif  
%} </td> <td>R$ {{valor.valor}}</td> </tr> {% endfor %} </table>  
</div> </div> {% endblock %}
```

Crie o css view_extrato.css:

```
table{ width: 100%; padding: 20px; text-align: center; } .linha{  
height: 40px; background-color: var(--dark-color); padding: 20px; }  
td, th{ padding: 20px; } tr{ background-color: var(--main-color); }  
.botao-secundario{ background-color: transparent; color: white; text-  
decoration: none; border: 2px solid var(--contrast-color); padding:  
10px; border-radius: 15px; width: 100%; display: block; text-align:  
center; }
```

Atualize o formulário dos filtros:

```
<form action="{% url 'view_extrato' %}" method="GET">
```

Realize os filtros nas views:

```
conta_get = request.GET.get('conta') categoria_get =
request.GET.get('categoria') if conta_get: valores =
valores.filter(conta_id=conta_get) if categoria_get: valores =
valores.filter(categoria_id=categoria_get)
```

▼ Exportar extrato

Crie a URL para exportar_extrato:

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

crie o arquivo partial extrato.html:

```
{% load static %} <div class="card"> <table> <tr> <th>Conta</th>
<th>Categoria</th> <th>Data</th> <th>Tipo</th> <th>valor</th> </tr> {% 
for valor in valores %} <tr class="linha"> <td width="10%">
{{valor.conta}}</td> <td>{{valor.categoria}}</td> <td>{{valor.data}}</td>
<td> {% if valor.tipo == 'S'%}  {% else %}  {% endif %} </td> <td>R$ {{valor.valor}}</td>
</tr> {% endfor %} </table> </div>
```

instale a biblioteca para converter HTML para PDF:

```
pip install weasyprint
```

Crie a view exportar_pdf:

```
def exportar_pdf(request): valores =
Valores.objects.filter(data__month=datetime.now().month) contas =
Conta.objects.all() categorias = Categoria.objects.all() path_template =
os.path.join(settings.BASE_DIR, 'templates/partials/extrato.html')
path_output = BytesIO() template_render =
render_to_string(path_template, {'valores': valores, 'contas': contas,
'categorias': categorias})
HTML(string=template_render).write_pdf(path_output)
path_output.seek(0) return FileResponse(path_output,
filename="extrato.pdf")
```

Altere o botão para gerar o PDF:

```
<a href="{% url 'exportar_pdf' %}" class="botao-secundario">Exportar  
extrato</a>
```

▼ Definir planejamento

Crie o app planejamento:

```
python3 manage.py startapp planejamento
```

Instale o APP!

Crie a URL para planejamento:

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

Crie a URL para definir_planejamento:

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

Crie a view definir_planejamento:

```
def definir_planejamento(request): categorias =
Categoria.objects.all() return render(request,
'definir_planejamento.html', {'categorias': categorias})
```

Crie o HTML definir_planejamento:

```
{% extends 'bases/base.html' %} {% load static %} {% block 'head' %}  
<link href="{% static 'perfil/css/home.css' %}" rel="stylesheet">  
<link href="{% static 'perfil/css/gerenciar.css' %}" rel="stylesheet">  
<link href="{% static 'extrato/css/view_extrato.css' %}"  
rel="stylesheet"> {% endblock %} {% block 'body' %} <div  
class="container"> <br> <br> <p class="fonte-destaque">Definir  
planejamento</p> <div class="card"> <table> <tr> <th>Categoria</th>  
<th>Valor</th> <th>Ação</th> </tr> {% for categoria in categorias %}  
<tr class="linha"> <td>{{categoria}}</td> <td><input type="number"  
class="form-control" value="{{categoria.valor_planejamento}}"></td>  
<td><button style="width: 100%;" class="botao-  
secundario">Salvar</button></td> </tr> {% endfor %} </table> </div>  
</div> {% endblock %}
```

Crie a função JavaScript para enviar os dados: