

PSW 8.0 | Aula 2

Acesse diretamente pelo Notion:

```
https://grizzly-amaranthus-f6a.notion.site/PSW-8-0-Aula-2-a5edc843a7e34c69983cd163b31ccaa2?pvs=4
```

▼ Solicitar exame

Crie uma URL para o app exames:

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

Crie o arquivo `urls.py` em exames:

```
from django.urls import path
from . import views

urlpatterns = [
    path('solicitar_exames/', views.solicitar_exames, name="solicitar_exames"),
]
```

Agora vamos para view `solicitar_exames`:

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

@login_required
def solicitar_exames(request):
    if request.method == "GET":
        return render(request, 'solicitar_exames.html')
```

Dentro de `exames/templates` crie o `solicitar_exames.html`:

```
{% extends "bases/base.html" %}
{% load static %}

{% block 'head' %}

{% endblock 'head' %}

{% block 'conteudo' %}
    <br>
    <br>
    <div class="container">
        <div class="row">

            <div class="col-md">
                <h3 class="font-destaque">Solicitar exames</h3>
```

```

        <form action="" method="POST">
            <label for="">Selecione os exames para realizar</label>
            <select class="select form-control bg-select" multiple name="exames">

                <option class="opt" value="">Exame 1</option>

            </select>
            <br>
            <input type="submit" class="btn-default tamanho-2" name="ultimo_nome" value="Solicitar exames">
        </form>
    </div>

    <div class="col-md-2">

    </div>

    <div class="col-md card">

        <div style="font-size: 20px; font-weight: bold">

            

            Exame 1
            <span class="previa-preco">
                R$ 30
            </span>
        </div>

        <hr style="color: white;">
        <div>
            <label>Total: </label><label class="previa-preco">R$ 30</label>
        </div>
        <br>

        <h5>Data: 08 de Setembro</h5>
        <br>

        <form action="" method="POST">
            <button class="btn-secundario">Fechar pedido</button>
        </form>

    </div>
</div>
</div>
{% endblock 'conteudo' %}

```

Crie o arquivo css.css de exames nos arquivos estáticos:

```

body{

    background-image: url('/static/geral/img/bg2.png');
    background-size: cover;

}

.bg-select{
    background-color: transparent;
    border-color: var(--differential-color);
}

.card{
    background-color: rgba(217, 217, 217, .14);
    color: white;
    padding: 15px;
}

.previa-preco{
    float: right;
    font-weight: bold;
}

.tamanho-2{
    width: 50%;
}

.btn-secundario{

    background-color: #838FFF;
    color: black;
    padding: 10px;
    font-weight: bold;
    border: none;
    border-radius: 10px;
}

.btn-secundario-outline{
    background-color: transparent;
    border-color: #838FFF;
}

```

```
padding: 10px;
}
```

Importe o arquivo em solicitar_exame.html:

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

Para listar dinamicamente os exames, busque todos os tipos nas views:

```
from django.shortcuts import render
from django.contrib.auth.decorators import login_required
from .models import TiposExames

@login_required
def solicitar_exames(request):
    tipos_exames = TiposExames.objects.all()
    if request.method == "GET":
        return render(request, 'solicitar_exames.html', {'tipos_exames': tipos_exames})
```

Exiba dinamicamente no HTML as opções selecionáveis dos exames:

```
{% for exame in tipos_exames %}
    <option class="opt" value="{{exame.id}}">{{exame}}</option>
{% endfor %}
```

Atualize o form para enviar os dados para solicitar_exames:

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

Atualize a view para processar a requisição POST:

```
@login_required
def solicitar_exames(request):
    tipos_exames = TiposExames.objects.all()
    if request.method == "GET":
        return render(request, 'solicitar_exames.html', {'tipos_exames': tipos_exames})
    else:
        exames_id = request.POST.getlist('exames')

        solicitacao_exames = TiposExames.objects.filter(id__in=exames_id)
        #preco_total = solicitacao_exames.aggregate(total=Sum('preco'))['total']
        preco_total = 0
        for i in solicitacao_exames:
            preco_total += i.preco

        return render(request, 'solicitar_exames.html', {'solicitacao_exames': solicitacao_exames, 'preco_total': preco_total, 'tipos_exames': tipos_exames})
```

Exiba os dados do pedido dinamicamente para o cliente aprovar:

```
{% if solicitacao_exames %}

    {% for exame in solicitacao_exames %}
        <div style="font-size: 20px; font-weight: bold">
            {% if exame.disponivel %}
                
            {% else %}
                
            {% endif %}
            {{exame.nome}}
            <span class="previa-preco">
                {{exame.preco}}
            </span>
        </div>
    {% endfor %}

    <hr style="color: white;">
    <div>
        <label>Total: </label><label class="previa-preco">{{preco_total}}</label>
    </div>
    <br>
```

```
<h5>Data: 08 de Setembro</h5>
<br>

<form action="{% url 'fechar_pedido' %}" method="POST">{% csrf_token %}
  {% for exame in solicitacao_exames %}
    <input type="hidden" value="{{exame.id}}" name="exames">
  {% endfor %}
  <button class="btn-secundario">Fechar pedido</button>
</form>

{% else %}
  <h3>Nenhum exame selecionado</h3>
{% endif %}
```

Adicione as imagens nos arquivos estáticos:



Agora adicione os exames solicitados em um input hidde:

```
{% for exame in solicitacao_exames %}
  <input type="hidden" value="{{exame.id}}" name="exames">
{% endfor %}
```

Crie a URL para fechar o pedido:

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

Crie a view para fechar o pedido do cliente:

```
@login_required
def fechar_pedido(request):
    exames_id = request.POST.getlist('exames')
    solicitacao_exames = TiposExames.objects.filter(id__in=exames_id)

    pedido_exame = PedidosExames(
        usuario = request.user,
        data = datetime.now()
    )

    pedido_exame.save()

    for exame in solicitacao_exames:
        solicitacao_exames_temp = SolicitacaoExame(
            usuario=request.user,
            exame=exame,
            status="E"
        )
        solicitacao_exames_temp.save()
        pedido_exame.exames.add(solicitacao_exames_temp)

    pedido_exame.save()

    messages.add_message(request, constants.SUCCESS, 'Pedido de exame concluído com sucesso')
    return redirect('/exames/ver_pedidos/')
```

Redirecione o form para o fechar_pedido:

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

▼ Gerenciar pedidos

Agora vamos trabalhar com o gerenciamento dos pedidos de exames.

Crie a URL:

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

Crie a view para exibir a página HTML

```
@login_required
def gerenciar_pedidos(request):
    pedidos_exames = PedidosExames.objects.filter(usuario=request.user)
    return render(request, 'gerenciar_pedidos.html', {'pedidos_exames': pedidos_exames})
```

Crie o HTML gerenciar_pedidos:

```
{% extends "bases/base.html" %}
{% load static %}

{% block 'head' %}
    <link href="{% static 'exames/css/css.css' %}" rel="stylesheet">
    <link href="{% static 'exames/css/gerenciar_pedidos.css' %}" rel="stylesheet">
{% endblock 'head' %}

{% block 'conteudo' %}
    <br>
    <br>
    <div class="container">
        {% if messages %}
            <br>
            {% for message in messages %}
                <div class="alert {{ message.tags }}">{{ message }}</div>
            {% endfor %}
        {% endif %}
        <div class="card card-view">
            <table>
                <tr>
                    <th>Solicitação</th>
                    <th>Data</th>
                    <th>Exames</th>
                    <th>Ação</th>
                </tr>

                {% for pedidos in pedidos_exames %}
                    <tr class="linha-tabela">
                        <td>{{ pedidos.id }}</td>
                        <td>{{ pedidos.data }}</td>
                        <td>
                            <select class="form-select">
                                {% for exame in pedidos.exames.all %}
                                    <option style="color: black" value="">{{ exame.exame.nome }}</option>
                                {% endfor %}
                            </select>
                        </td>
                        <td>
                            <a href="" class="btn btn-danger {% if not pedidos.agendado %} disabled {% endif %}">Cancelar</a>
                        </td>
                    </tr>
                {% endfor %}
            </table>

        </div>
    </div>
{% endblock 'conteudo' %}
```

Crie o CSS gerenciar_pedidos.css:

```
.linha-tabela{
    background-color: var(--dark-color);
    height: 20px;
}

td, th{
    padding: 20px;
}

.card-view{
    text-align: center;
    padding: 0;
    border: none;
```

```
height: 70vh;
}
```

Em `urls.py` de exames crie a URL responsável por cancelar o pedido:

```
path("cancelar_pedido/<int:pedido_id>", views.cancelar_pedido, name="cancelar_pedido"),
```

Crie a view `cancelar_pedido`:

```
@login_required
def cancelar_pedido(request, pedido_id):
    pedido = PedidosExames.objects.get(id=pedido_id)

    if not pedido.usuario == request.user:
        messages.add_message(request, constants.ERROR, 'Esse pedido não é seu')
        return redirect('/exames/gerenciar_pedidos/')

    pedido.agendado = False
    pedido.save()
    messages.add_message(request, constants.SUCCESS, 'Pedido excluído com sucesso')
    return redirect('/exames/gerenciar_pedidos/')
```

Por questões de segurança realize a verificação se o pedido é realmente do usuário logado:

```
if not pedido.usuario == request.user:
    messages.add_message(request, constants.ERROR, 'Esse pedido não é seu')
    return redirect('/exames/gerenciar_pedidos/')
```

Altere o botão de cancelar pedido para:

```
<a href="{% url 'cancelar_pedido' pedidos.id %}" class="btn btn-danger {% if not pedidos.agendado %} disabled {% endif %}">Cancelar</a>
```

▼ Gerenciar exames

Nessa etapa iremos gerenciar individualmente os exames do cliente.

Crie a URL `gerenciar_exame`:

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

Agora a view:

```
@login_required
def gerenciar_exames(request):
    exames = SolicitacaoExame.objects.filter(usuario=request.user)

    return render(request, 'gerenciar_exames.html', {'exames': exames})
```

Crie o HTML `gerenciar_exames`:

```
{% extends "bases/base.html" %}
{% load static %}

{% block 'head' %}
    <link href="{% static 'exames/css/css.css' %}" rel="stylesheet">
    <link href="{% static 'exames/css/gerenciar_pedidos.css' %}" rel="stylesheet">
    <link href="{% static 'exames/css/gerenciar_exames.css' %}" rel="stylesheet">
{% endblock 'head' %}

{% block 'conteudo' %}
    <br>
    <br>
    <div class="container">
        {% if messages %}
            <br>
```

```

        {% for message in messages %}
        <div class="alert {{ message.tags }}">{{ message }}</div>
        {% endfor %}
    {% endif %}
    <div class="card card-view">

        <div class="sub-card">
            <h3>Exames de sangue</h3>
            <hr>
            <br>
            LISTAR EXAMES AQUI
        </div>
        <br>
        <br>
        <div class="sub-card">
            <h3>Exames de imagem</h3>
            <hr>
            <br>
            LISTAR EXAMES AQUI
        </div>

    </div>

</div>

{% endblock 'conteudo' %}

```

Crie o gerenciar_exames.css

```

.sub-card{
    background-color: var(--dark-color);
    padding: 20px;
}

```

Liste os exames de sangue:

```

{% for exame in exames %}
    {% if exame.exame.tipo == 'S'%}
        <div class="row">
            <div class="col-md"><h3>{{exame.exame.nome}}</h3></div>
            <div class="col-md">{{exame.badge_template}}</div>
            <div class="col-md">
                {% if exame.status == 'F' %}
                    <a href="{% url "permitir_abrir_exame" exame.id %}" class="btn btn-light">Abrir</a>
                {% else %}
                    <a href="#" class="btn btn-light disabled">Abrir</a>
                {% endif %}
            </div>
        </div>
        <br>
    {% endif %}
{% endfor %}

```

Liste os exames de imagens:

```

{% for exame in exames %}
    {% if exame.exame.tipo == 'I'%}
        <div class="row">
            <div class="col-md"><h3>{{exame.exame.nome}}</h3></div>
            <div class="col-md">{{exame.badge_template}}</div>
            <div class="col-md">
                {% if exame.status == 'F' %}
                    <a href="{% url "permitir_abrir_exame" exame.id %}" class="btn btn-light">Abrir</a>
                {% else %}
                    <a href="#" class="btn btn-light disabled">Abrir</a>
                {% endif %}
            </div>
        <br>
    {% endif %}
{% endfor %}

```

Em SolicitacaoExame crie o badge_template:

```

from django.utils.safestring import mark_safe

def badge_template(self):
    if self.status == 'E':
        classes_css = 'bg-warning text-dark'
        texto = "Em análise"

```

```
elif self.status == 'F':
    classes_css = 'bg-success'
    texto = "Finalizado"

return mark_safe(f"<span class='badge bg-primary {classes_css}'>{texto}</span>")
```

Agora crie uma URL para permitir que o exame seja aberto:

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

Crie a view `permitir_abrir_exame`:

```
@login_required
def permitir_abrir_exame(request, exame_id):
    exame = SolicitacaoExame.objects.get(id=exame_id)
    #TODO: validar se o exame é do usuário
    if not exame.requer_senha:
        # verificar se o pdf existe
        return redirect(exame.resultado.url)

    else:
        return redirect(f'/exames/solicitar_senha_exame/{exame.id}')
```

Altere os botões abrir para:

```
<a href="{% url 'permitir_abrir_exame' exame.id %}" class="btn btn-light">Abrir</a>
```

No core do projeto adicione a URL de media ao `urlpatterns`:

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('usuarios/', include('usuarios.urls')),
    path('exames/', include('exames.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Vamos para url que solicita a senha do exame:

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

Crie a view responsável:

```
@login_required
def solicitar_senha_exame(request, exame_id):
    exame = SolicitacaoExame.objects.get(id=exame_id)
    if request.method == "GET":
        return render(request, 'solicitar_senha_exame.html', {'exame': exame})
    elif request.method == "POST":
        senha = request.POST.get("senha")
        #TODO: validar se o exame é do usuário
        if senha == exame.senha:
            return redirect(exame.resultado.url)
        else:
            messages.add_message(request, constants.ERROR, 'Senha inválida')
            return redirect(f'/exames/solicitar_senha_exame/{exame.id}')
```

Crie o HTML `solicitar_senha_exame`:

```
{% extends "bases/base.html" %}
{% load static %}
```



```
{% block 'head' %}
    <link href="{% static 'exames/css/css.css' %}" rel="stylesheet">
    <link href="{% static 'exames/css/gerenciar_pedidos.css' %}" rel="stylesheet">
    <link href="{% static 'exames/css/gerenciar_exames.css' %}" rel="stylesheet">
{% endblock 'head' %}

{% block 'conteudo' %}
    <br><br>

    <div class="container">
        {% if messages %}
            <br>
            {% for message in messages %}
                <div class="alert {{ message.tags }}">{{ message }}</div>
            {% endfor %}
        {% endif %}
        <h3>{{exame.exame.nome}}</h3>
        <br>
        <form action="{% url 'solicitar_senha_exame' exame.id %}" method="POST">{% csrf_token %}

            <label>Senha de acesso</label>
            <br>
            <input type="text" name="senha" class="form-control" style="width: 40%">
            <br>

            <input type="submit" class="btn-default">

        </form>

    </div>

{% endblock 'conteudo' %}
```