

PYSTACK WEEK 2.0 | Aula 1

Primeiro vamos criar o ambiente virtual e ativa-lo

```
# Criar
# Linux
python3 -m venv venv
# Windows
python -m venv venv

#Ativar
# Linux
source venv/bin/activate
# Windows
venv/Scripts/Activate
```

Agora deve-se instalar as bibliotecas necessárias

```
pip install django
pip install pillow
```

Crie seu projeto django

```
django-admin startproject imobi .
```

Configure os arquivos estáticos em settings.py

```
STATIC_URL = '/static/'
STATICFILES_DIRS = (os.path.join(BASE_DIR, 'templates/static'),)
STATIC_ROOT = os.path.join('static')

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'
```

Vamos então criar o nosso app para trabalharmos com cadastro e login de usuário

```
python manage.py startapp autenticacao
```

Crie uma URL para o novo app

```
path('auth/', include('autenticacao.urls')),
```

Agora é preciso criar as rotas para login e cadastro

```
path('cadastro/', views.cadastro, name='cadastro'),
path('login/', views.login, name='login'),
```

Faça que a view cadastro renderize uma página em html

```
def cadastro(request):
    return render(request, 'cadastro.html')

def login(request):
    pass
```

Crie o arquivo cadastro.html

Agora vamos criar um arquivo base.html

```
{% load static %}
<!doctype html>
<html lang="pt-BR">
  <head>
    {% block 'head' %}{% endblock %}
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css" integrity="sha384-zCbKRCUGaJDkqS1"
    <title>{% block 'titulo' %}{% endblock %}</title>
  </head>
  <body>

    {% block 'body' %}{% endblock %}

    <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js" ></script>

  </body>
</html>
```

Aponte para a pasta templates

```
os.path.join(BASE_DIR, 'templates')
```

Desenvolva o cadastro.html

```
{% extends 'base.html' %}

{% block 'titulo' %}Cadastro-se{% endblock %}

{% block 'body' %}
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-3 sidebar">
        <div class="logo_sidebar">

          <h2 class="logo">
            <img class="img_logo_sidebar" src="">IMOBIL
          </h2>

        </div>

        <div class="corpo_sidebar">
          <div>
            <h2 class="titulo_cor_branca">Cadastro-se</h2>
          </div>

          <div>
            <p class="cor_branca descricao">As melhores casas</p>
            <p style="margin-top:-20px" class="cor_branca descricao">Nos melhores preços</p>
            <br>
            <br>
            <a class="btn_logar" href="">LOGAR</a>
          </div>

        </div>
      </div>

      <div class="col-md-9">
        <div class="area_cadastrar">

          <form action="" method="">
            <h2 class="titulo_area_cadastrar">Cadastro-se</h2>
            <br>
            <input name="username" class="input_cadastrar input_cadastrar_nome" type="text" placeholder="Username...">
          </form>
        </div>
      </div>
    </div>
  </div>
{% endblock %}
```

```

                <br>
                <br>
                <input name="email" class="input_cadastrar input_cadastrar_email" type="text" placeholder="Email...">
                <br>
                <input name="senha" class="input_cadastrar input_cadastrar_senha" type="password" placeholder="Senha...">
                <br>
                <input class="btn_cadastrar" type="submit" value="Cadastrar">
            </form>
        </div>

    </div>

</div>

</div>

{% endblock %}

```

Precisamos então deixar essa página mais bonita, para isso vamos carregar um arquivo css

```

{% load static %}

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

```

Crie o arquivo css

```

.sidebar{
    background-color: #30CD9E;
    height: 100vh;
}

.cor_branca{
    color: white;
}

.logo_sidebar{
    text-align: center;
    margin-top: 50%;
}

.img_logo_sidebar{
    width: 70px;
}

.logo{
    font-size: 70px;
    color: white;
    font-family: 'Courier New', Courier, monospace;
}

.corpo_sidebar{
    margin-top: 50px;
    text-align: center;
}

.titulo{
    font-size: 50px;
    font-weight: bold;
}

```

```

    border-bottom: 2px solid #1EA9B1;
    display: inline;
}

.descricao{
    font-size: 30px;
}

.btn_logar{

    color: white;
    font-size: 30px;
    border: 2px solid white;
    padding: 10px 20px 10px 20px;
    border-radius: 20px;

}

.btn_logar:hover{
    text-decoration: none;
    color: white;
}

.area_cadastrar{
    text-align: center;
}

.titulo_area_cadastrar{
    font-size: 50px;
    color: #30CD9E;
    margin-top: 15%;
    font-weight: bold;
}

.input_cadastrar{

    width: 60%;
    border: none;
    background-color: #C4C4C4;
    padding: 10px;
    outline: 0;
    font-size: 20px;
    background-size: 30px 30px;
    background-repeat: no-repeat;
    background-position-x: 20px;
    background-position-y: 10px;
    padding-left: 70px;
}

.input_cadastrar_nome{

    background-image: url('/static/autenticacao/img/user.png');
}

.input_cadastrar_email{

    background-image: url('/static/autenticacao/img/email.png');
}

.input_cadastrar_senha{

    background-image: url('/static/autenticacao/img/senha.png');
}

.btn_cadastrar{

    background-color: #30CD9E;
    border: none;
    color: white;
    font-size: 30px;
    padding: 5px 10px 5px 10px;
    border-radius: 20px;
    margin-top: 4%;
}

```

```
}
```

Adicione a URL para os arquivos estáticos

```
from django.conf import settings
from django.conf.urls.static import static

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

Agora vamos adicionar as imagens dentro dos arquivos estáticos

Configure o botão logar para redirecionar para página de login

Ainda precisamos fazer o formulário funcionar, para isso vamos adicionar uma action, method e o token de segurança

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

Altere a views para diferenciar requisições POST de GET

```
def cadastro(request):
    if request.method == "GET":
        return render(request, 'cadastro.html')
    elif request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        senha = request.POST.get('senha')
        return HttpResponseRedirect(username)
```

Faça as validações

```
if len(username.strip()) == 0 or len(email.strip()) == 0 or len(senha.strip()) == 0:
    return redirect('/auth/cadastro')

user = User.objects.filter(username=username)

if user.exists():
    return redirect('/auth/cadastro')
```

Salve os dados no banco

```
try:
    user = User.objects.create_user(username=username,
                                     email=email,
                                     password=senha)

    user.save()
    return redirect('/auth/logar')
except:
    return redirect('/auth/cadastro')
```

Configure as mensagens

```
from django.contrib.messages import constants

MESSAGE_TAGS = {
    constants.DEBUG: 'alert-primary',
    constants.ERROR: 'alert-danger',
    constants.SUCCESS: 'alert-success',
    constants.INFO: 'alert-info',
    constants.WARNING: 'alert-warning',
}
```

Adicione as mensagens nas views

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

def cadastro(request):
    if request.method == "GET":
        return render(request, 'cadastro.html')
    elif request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        senha = request.POST.get('senha')

        if len(username.strip()) == 0 or len(email.strip()) == 0 or len(senha.strip()) == 0:
            messages.add_message(request, constants.ERROR, 'Preencha todos os campos')
            return redirect('/auth/cadastro')

        user = User.objects.filter(username=username)

        if user.exists():
            messages.add_message(request, constants.ERROR, 'Já existe um usuário com esse nome cadastrado')
            return redirect('/auth/cadastro')
        try:
            user = User.objects.create_user(username=username,
                                            email=email,
                                            password=senha)

            user.save()
            messages.add_message(request, constants.SUCCESS, 'Cadastro realizado com sucesso')
            return redirect('/auth/logar')
        except:
            messages.add_message(request, constants.ERROR, 'Erro interno do sistema')
            return redirect('/auth/cadastro')
```

Agora vamos para o login, crie o login.html

```
{% extends 'base.html' %}
{% load static %}

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

{% block 'titulo' %}Cadastre-se{% endblock %}

{% block 'body' %}
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-3 sidebar">
                <div class="logo_sidebar">

                    <h2 class="logo">
                        IMOBILI
                    </h2>

                </div>

                <div class="corpo_sidebar">
                    <div>
                        <h2 class="titulo cor_branca">Logar</h2>
                    </div>

                    <div>
                        <p class="cor_branca descricao">As melhores casas</p>
                        <p style="margin-top:-20px" class="cor_branca descricao">Nos melhores preços</p>
                        <br>
                        <br>
                        <a class="btn_logar" href="{% url 'cadastro' %}">Cadastre-se</a>
                    </div>

                </div>
            </div>
        </div>
    </div>
```

```

<div class="col-md-9">
    <div class="area_cadastrar">
        <h2 class="titulo_area_cadastrar">Logar</h2>
        {% if messages %}
            {% for message in messages %}
                <div class="alert {{message.tags}}">
                    {{message}}
                </div>
            {% endfor %}
        {% endif %}
        <br>
        <br>
        <form method="POST" action="{% url 'logar' %}">{% csrf_token %}
            <input name="username" class="input_cadastrar input_cadastrar_email" type="text" placeholder="Username...">
            <br>
            <input name="senha" class="input_cadastrar input_cadastrar_senha" type="password" placeholder="Senha...">
            <br>
            <input class="btn_cadastrar" type="submit" value="Logar">
        </form>
    </div>
</div>

</div>

{% endblock %}

```

Vamos renderizar o HTML pela nossa view logar

```

def logar(request):
    if request.method == "GET":
        return render(request, 'logar.html')
    elif request.method == "POST":
        username = request.POST.get('username')
        senha = request.POST.get('senha')

        usuario = auth.authenticate(username=username, password=senha)

        if not usuario:
            messages.add_message(request, constants.ERROR, 'Username ou senha inválidos')
            return redirect('/auth/logar')
        else:
            auth.login(request, usuario)
            return redirect('/')

```

Em ambas as páginas verifique se o usuário já está logado antes de exibir

```

if request.user.is_authenticated:
    return redirect('/')

```

E assim finalizamos nossa primeira aula.

[Recursos](#)