

Login



Prof. Dr. João Paulo Lemos Escola
Copyright© 2025


Conteúdo

- Sistema de autenticação do Django:
 - Criando usuários;
 - Definindo e alterando senha;
 - Gerenciando login e logoff.
- Módulo messages:
 - Envio de mensagens da view para o template.


settings.py

- Vamos criar um app chamado authentication:
 - `py manage.py startapp authentication`
- Habilite o app em settings.py:

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'members',  
41     'plans',  
42     'authentication',  
43 ]
```



urls.py do projeto



```
19 urlpatterns = [  
20     path('', include('members.urls')),  
21     path('', include('authentication.urls')),  
22     path('admin/', admin.site.urls),  
23 ]
```

urls.py

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('login/', views.login_page, name='login_page'),
6     path('register/', views.register_page, name='register_page'),
7     path('logout/', views.logout_page, name='logout_page'),
8     path('change_password/', views.password_page, name='password_page'),
9 ]
```

login.html

- Não esqueça de criar a pasta 'templates';
- Tela onde o usuário se autentica para entrar no sistema.

```
1 {% extends "master.html" %}
2
3 {% block title %}
4     - Acesso
5 {% endblock %}
6
7 {% block content %}
8
9     <div class="card mt-2">
10
11         <div class="card-header">
12             Acesso
13         </div>
14         <div class="card-body">
15
16             {% if messages %}
17             {% for message in messages %}
18                 {% if message.tags == 'error' %}
19                     <div class="alert alert-danger" role="alert">{{ message }}</div>
20                 {% else %}
21                     <div class="alert alert-primary" role="alert">{{ message }}</div>
22                 {% endif %}
23             {% endfor %}
24             {% endif %}
25
26             <form action="/login/" method="post" enctype="multipart/form-data">
27                 {% csrf_token %}
28
29                 Usuário:<br>
30                 <input name="username">
31                 <br><br>
32                 Senha:<br>
33                 <input name="password" type="password">
34                 <br><br>
35                 <input type="submit" value="Submit">
36             </form>
37
38             <br>
39             <p>Não tem conta? <a href="/register/">Registre-se!</a></p>
40
41         </div>
42     </div>
43
44 {% endblock %}
```



127.0.0.1:8000/login/



Home members

Acesso

Usuário:

Senha:

Submit

Não tem conta? [Registre-se!](#)

register.html

```
24 <form action="/register/" method="post" enctype="multipart/form-data">
25     {% csrf_token %}
26
27     Primeiro Nome:<br>
28     <input name="first_name" required>
29     <br><br>
30     Sobrenome:<br>
31     <input name="last_name" required>
32     <br><br>
33     Usuário:<br>
34     <input name="username" required>
35     <br><br>
36     Senha:<br>
37     <input name="password" type="password" required>
38     <br><br>
39     <input type="submit" value="Submit">
40 </form>
```




127.0.0.1:8000/register/



[Home](#) [members](#)

Registro

Primeiro Nome:

Sobrenome:

Usuário:

Senha:

Já tem conta? [Acesse!](#)

views.py

```
1 from django.contrib.auth import authenticate, login, logout
2 from django.shortcuts import render, redirect
3 from django.contrib import messages
4 from django.contrib.auth.models import User
5
6 def login_page(request): 1 usage
7     if request.method == "POST":
8         username = request.POST.get('username')
9         password = request.POST.get('password')
10
11         if not User.objects.filter(username=username).exists():
12             messages.error(request, message: 'Usuário inválido')
13             return redirect('/login/')
14
15         user = authenticate(username=username, password=password)
16
17         if user is None:
18             messages.error(request, message: "Senha inválida")
19             return redirect('/login/')
20         else:
21             login(request, user)
22             return redirect('/members/')
23     return render(request, template_name: 'login.html')
```

```
25 def register_page(request): 1 usage
26     if request.method == 'POST':
27         first_name = request.POST.get('first_name')
28         last_name = request.POST.get('last_name')
29         username = request.POST.get('username')
30         password = request.POST.get('password')
31
32         user = User.objects.filter(username=username)
33
34         if user.exists():
35             messages.error(request, message: "Usuário já existe!")
36             return redirect('/register/')
37
38         user = User.objects.create_user(
39             first_name=first_name,
40             last_name=last_name,
41             username=username
42         )
43
44         user.set_password(password)
45         user.save()
46
47         messages.info(request, message: "Usuário registrado com sucesso!")
48         return redirect('/register/')
49
50     return render(request, template_name: 'register.html')
```

navbar.html dos outros módulos

- Mostra se o usuário está autenticado:

```
31      {% if user.is_authenticated %}  
32      <p>Bem-vindo, <a href="/change_password/">{{ user.username }}</a>. <a href="/logout/">(Sair)</a></p>  
33      {% endif %}  
34  
35      </div>  
36 </nav>
```



127.0.0.1:8000/members/



[Home](#) [members](#)

Bem-vindo, [jpescola2](#). ([Sair](#))

Members



[Emil Refsnes](#)



[Tobias Refsnes](#)



[Primeiro](#)

[1](#)

[2](#)

[3](#)

[4](#)

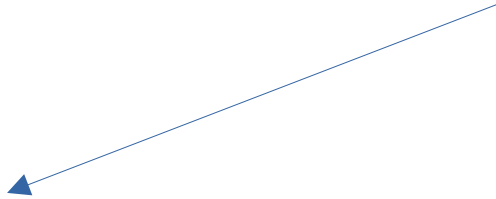
[5](#)

[6](#)

[Último](#)

[Add member](#)

Se o usuário clicar em (Sair)



```
52 def logout_page(request): 1 usage
53     logout(request)
54     messages.info(request, message: "Você saiu do sistema.")
55     return redirect('/login')
```



[Home](#) [members](#)

Acesso

Você saiu do sistema.

Usuário:

Senha:

Submit

Não tem conta? [Registre-se!](#)

password.html

```
16      {% if messages %}
17      {% for message in messages %}
18          {% if message.tags == 'error' %}
19              <div class="alert alert-danger" role="alert">{{ message }}</div>
20          {% else %}
21              <div class="alert alert-primary" role="alert">{{ message }}</div>
22          {% endif %}
23      {% endfor %}
24      {% endif %}
25
26      <form action="/change_password/" method="post" enctype="multipart/form-data">
27          {% csrf_token %}
28
29          Senha atual:<br>
30          <input name="actual_password" type="password" required>
31          <br><br>
32          Senha nova:<br>
33          <input name="new_password" type="password" required>
34          <br><br>
35          Repita a Senha nova:<br>
36          <input name="repeated_password" type="password" required>
37          <br><br>
38          <input type="submit" value="Submit">
39      </form>
40
41      <br>
42      <p>Não tem conta? <a href="/register/">Registre-se!</a></p>
```




127.0.0.1:8000/change_password/



[Home](#) [members](#)

Bem-vindo, [jpescola2](#). ([Sair](#))

Alteração de senha

Senha atual:

Senha nova:

Repita a Senha nova:


Submit

Não tem conta? [Registre-se!](#)

```
57 def password_page(request): 1 usage
58     if request.method == "POST":
59         password = request.POST.get('actual_password')
60         if authenticate(username=request.user.username, password=password):
61             p1 = request.POST.get('new_password')
62             p2 = request.POST.get('repeated_password')
63             if p1 == p2:
64                 request.user.set_password(p1)
65                 request.user.save()
66                 messages.info(request, message: "Senha alterada com sucesso.")
67             else:
68                 messages.error(request, message: "Senhas não coincidem.")
69         else:
70             messages.error(request, message: "Senha atual não é válida.")
71         return redirect('/change_password')
72     else:
73         return render(request, template_name: 'password.html')
```

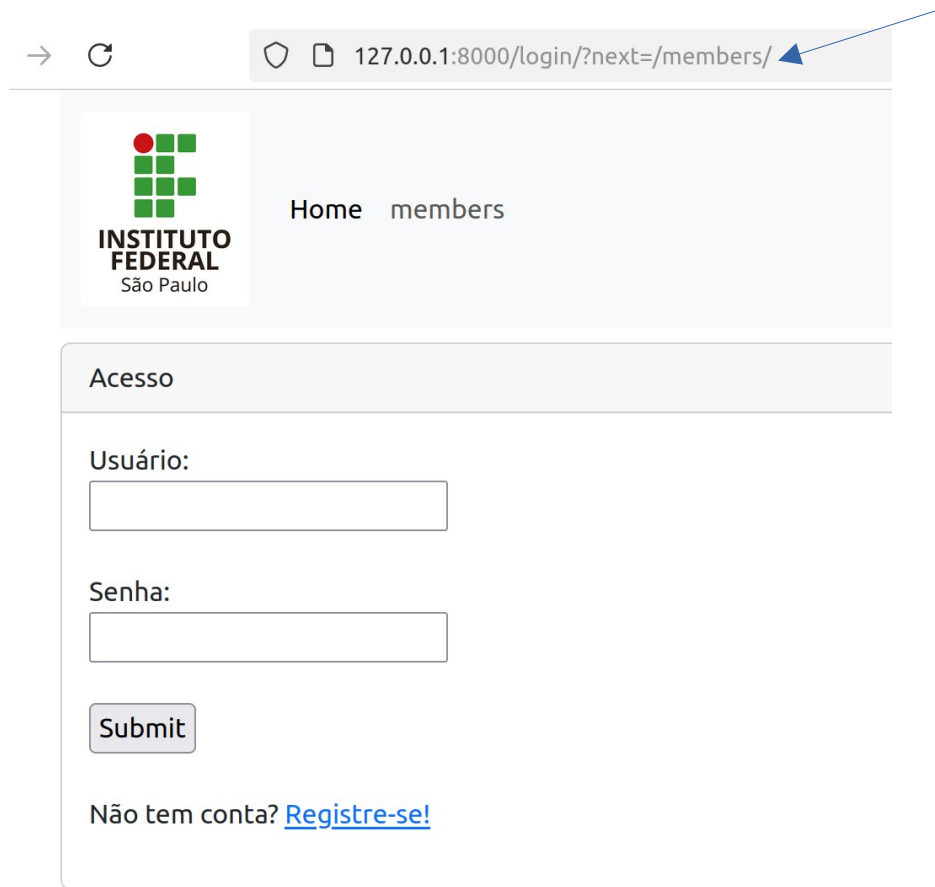
members/views.py

- from django.contrib.auth.decorators import login_required




```
10  @login_required(login_url="/login/")
11  def members(request):
12      mymembers = Member.objects.all()
13      template = loader.get_template('all_members.html')
14
15      paginator = Paginator(mymembers, per_page=2)
16      page_number = request.GET.get("page")
17      page_obj = paginator.get_page(page_number)
18
19      context = {
20          'mymembers': mymembers,
21          "page_obj": page_obj,
22      }
23      return HttpResponse(template.render(context, request))
```

A autenticação agora é obrigatória



→ ↻ 127.0.0.1:8000/login/?next=/members/

 [Home](#) [members](#)

**INSTITUTO
FEDERAL**
São Paulo

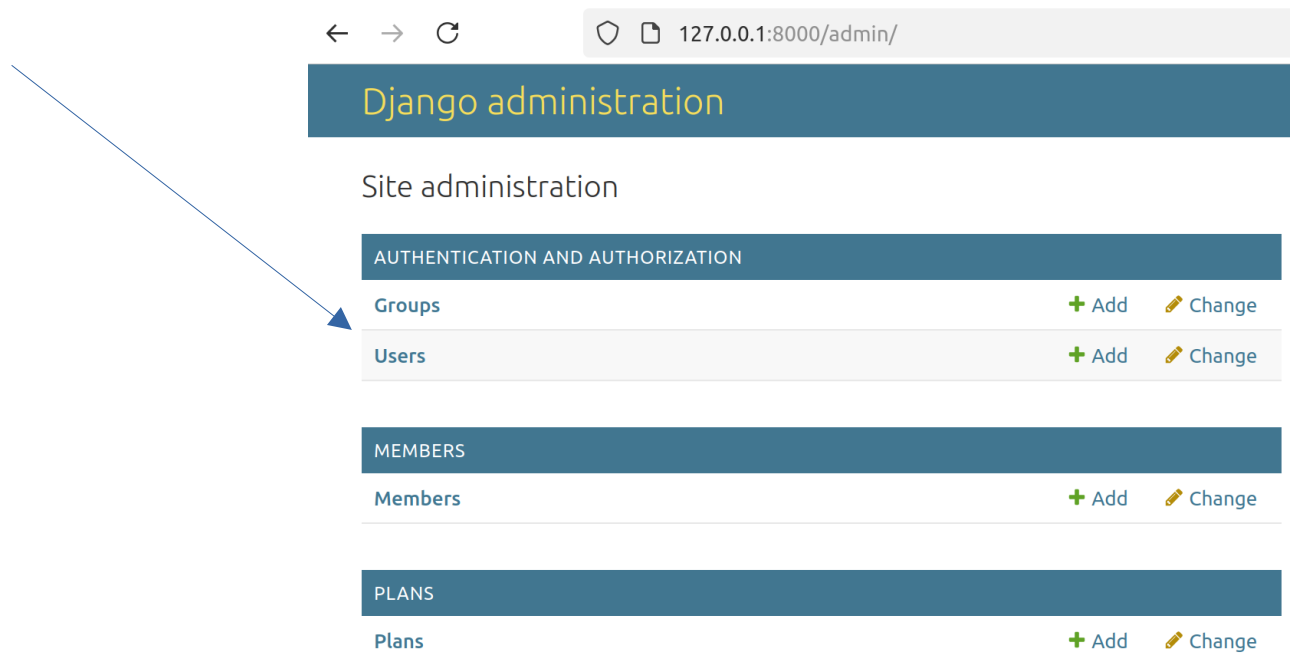
Acesso

Usuário:

Senha:

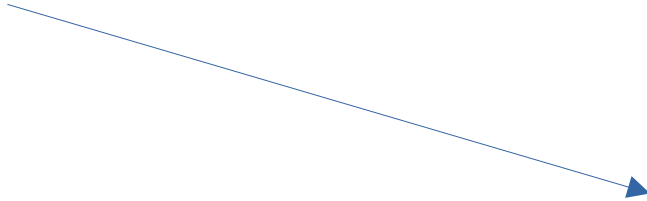
Não tem conta? [Registre-se!](#)

Gerenciando usuários pelo admin



Lista de usuários cadastrados

- Usuários podem ser criados ou autenticados pelo módulo admin.



Select user to change

Search

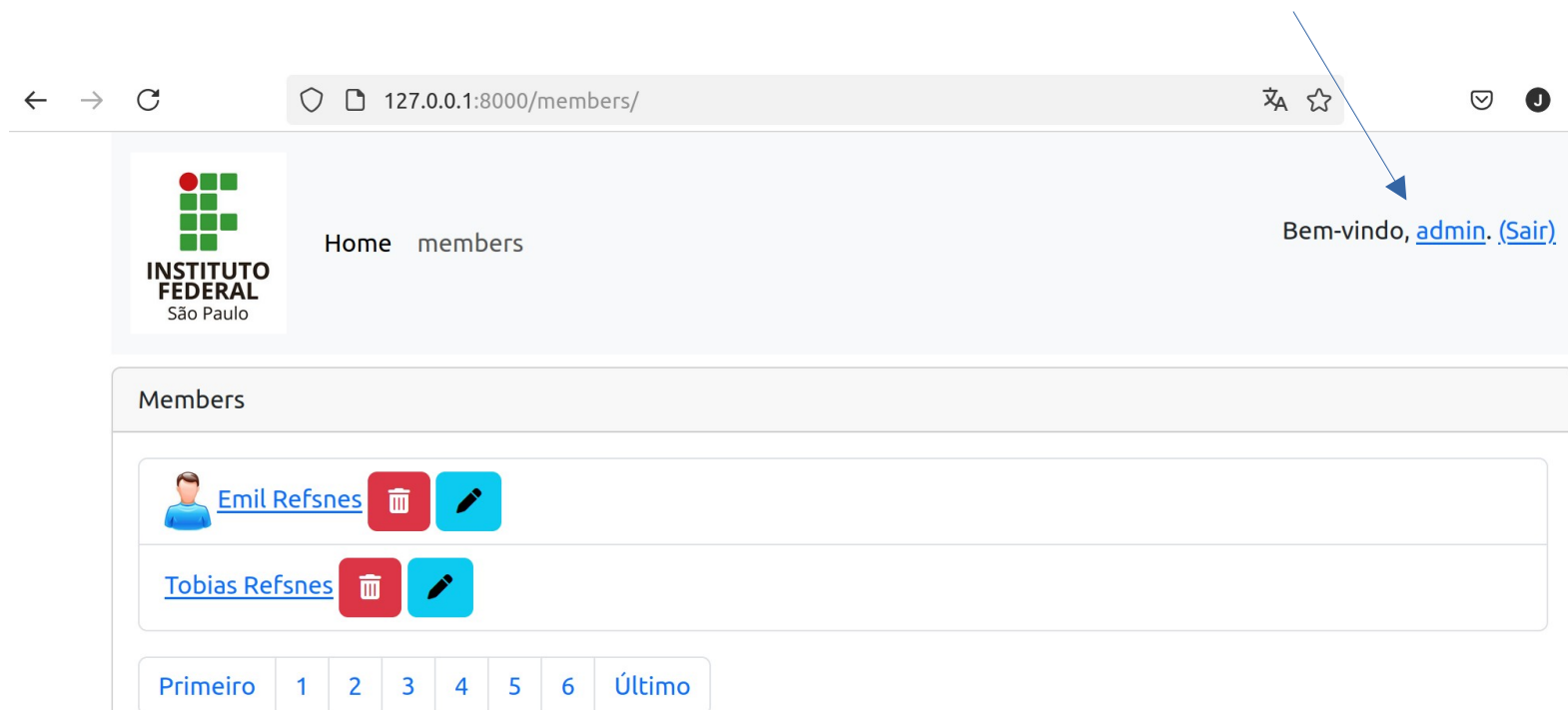
Action: ----- Go 0 of 4 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME
<input type="checkbox"/>	0	-	0	0
<input type="checkbox"/>	admin	admin@gmail.com	-	-
<input type="checkbox"/>	jpescola	jpescola@gmail.com	-	-
<input type="checkbox"/>	jpescola2	-	João	Escola

4 users

Admin logado

- Superusuário 'admin' criado por linha de comando para acesso ao módulo admin.



O que vimos?

- Sistema de autenticação do Django:
 - Criando usuários;
 - Definindo e alterando senha;
 - Gerenciando login e logoff.
- Módulo messages:
 - Envio de mensagens da view para o template.

Referências

- <https://www.geeksforgeeks.org/user-authentication-system-using-django/>
- <https://docs.djangoproject.com/en/5.1/topics/auth/default/>
- <https://docs.djangoproject.com/en/5.1/ref/contrib/messages/>