

PROGRAMAÇÃO BACK-END

django

The web framework for
perfectionists with deadlines.

PROGRAMAÇÃO BACK-END

Django Authentication

PROGRAMAÇÃO BACK-END

Autenticação no Django

A **autenticação** no Django é um dos componentes principais do **framework** e faz parte do sistema de **segurança** oferecido pela plataforma.

Ele fornece **funcionalidades** para autenticação de usuários, controle de permissões e grupos, gerenciamento de sessões e autorização.

PROGRAMAÇÃO BACK-END

Componentes principais:

User Model (Modelo de Usuário): Django já vem com um modelo de usuário pronto, chamado **User**, que armazena informações básicas como:

- ✓ **Nome de usuário (username)**
- ✓ **Senha (armazenada de forma criptografada)**
- ✓ **Email**
- ✓ **Nome e sobrenome**
- ✓ **Se o usuário é um administrador (superuser)**
- ✓ **Se a conta está ativa (ativo)**

PROGRAMAÇÃO BACK-END

Componentes principais (continuação):

Esse modelo pode ser utilizado diretamente ou estendido para adicionar novos campos e funcionalidades personalizadas.

Use Pgadmin4 e verifique as tabelas criados pelo Django no banco.

- >  auth_group
- >  auth_group_permissions
- >  auth_permission
- >  auth_user
- >  auth_user_groups
- >  auth_user_user_permissions
- >  django_admin_log
- >  django_content_type
- >  django_migrations
- >  django_session

PROGRAMAÇÃO BACK-END

Autenticação de Usuário:

Django usa sessões para gerenciar usuários autenticados. O sistema de autenticação básico é feito através de credenciais (nome de usuário e senha), mas também pode ser adaptado para funcionar com autenticação de terceiros, como OAuth (por exemplo, autenticação com Google, Facebook, etc.).

Para autenticar um usuário, normalmente se usa a função `authenticate`, seguida da função `login`.

PROGRAMAÇÃO BACK-END

Login e Logout:

O método **login** inicia a sessão do usuário, ou seja, associa o usuário autenticado à sessão atual do navegador.

O método **logout** finaliza essa sessão, removendo a associação do usuário à sessão do navegador.

PROGRAMAÇÃO BACK-END

Permissões e Grupos:

Django possui um sistema de permissões embutido que permite atribuir permissões a usuários e grupos.

Permissões: São definidas por padrão em modelos do Django, onde você pode controlar quem pode criar, alterar, deletar ou visualizar certos dados.

Grupos: Permitem agrupar usuários e atribuir permissões de maneira coletiva, ou seja, todos os usuários em um grupo terão as mesmas permissões.

PROGRAMAÇÃO BACK-END

Restrição de Acesso a Views:

Django também facilita a restrição de acesso a views específicas usando diretivas como `@login_required`.

Isso garante que apenas usuários autenticados possam acessar uma determinada view.

PROGRAMAÇÃO BACK-END

Formulários de Autenticação:

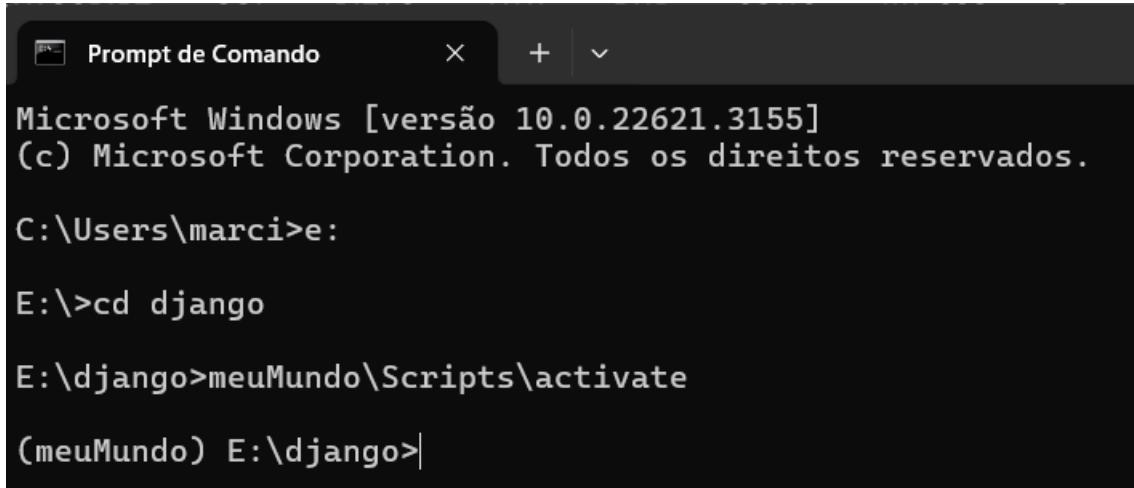
Django oferece formulários prontos para login ([AuthenticationForm](#)) e para criação de novos usuários ([UserCreationForm](#)), facilitando a criação dessas funcionalidades.

PROGRAMAÇÃO BACK-END

Iniciando

PROGRAMAÇÃO BACK-END

Rodar o script activate.bat



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.3155]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\marci>e:

E:\>cd django

E:\django>meuMundo\Scripts\activate

(meuMundo) E:\django>
```

Este passo é importante para configurar as variáveis de ambiente.

PROGRAMAÇÃO BACK-END

Iniciar Servidor Postgresql

PROGRAMAÇÃO BACK-END

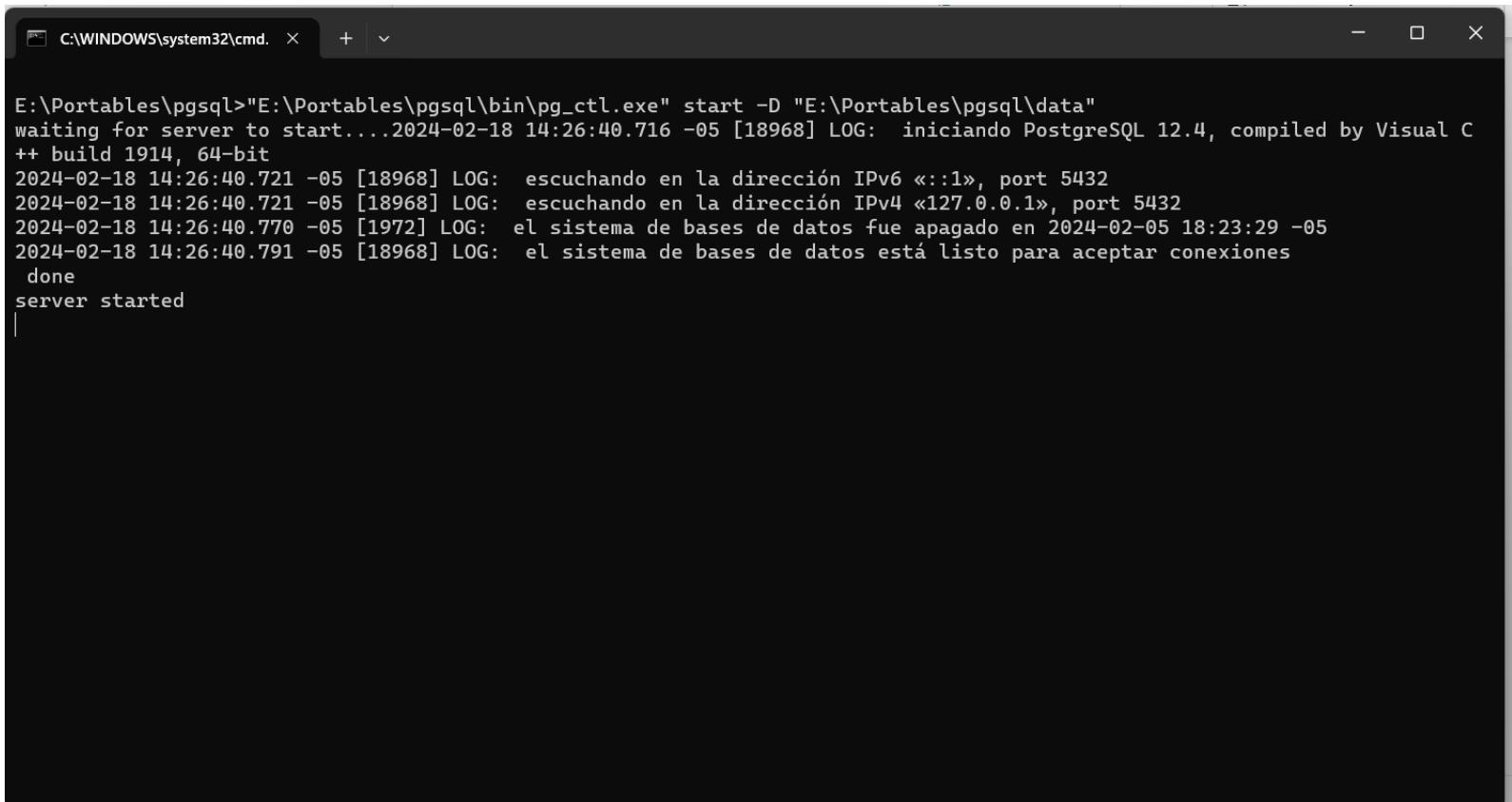
Na pasta `../pgsql` execute o arquivo `PostgreSQL-Start.bat` para iniciar o servidor



Nome	Data de modificação	Tipo	Tamanho
bin	11/08/2020 08:41	Pasta de arquivos	
data	18/02/2024 16:26	Pasta de arquivos	
doc	11/08/2020 08:41	Pasta de arquivos	
include	11/08/2020 08:41	Pasta de arquivos	
lib	11/08/2020 08:41	Pasta de arquivos	
pgAdmin 4	11/08/2020 08:41	Pasta de arquivos	
share	11/08/2020 08:41	Pasta de arquivos	
StackBuilder	11/08/2020 08:41	Pasta de arquivos	
Make Cluster.bat	09/01/2017 14:12	Arquivo em Lotes ...	1 KB
PgAdmin4.bat	28/04/2017 22:29	Arquivo em Lotes ...	1 KB
PostgreSQL-Restart.bat	06/04/2012 09:42	Arquivo em Lotes ...	1 KB
PostgreSQL-Start.bat	28/04/2017 21:46	Arquivo em Lotes ...	1 KB
PostgreSQL-Stop.bat	06/04/2012 09:43	Arquivo em Lotes ...	1 KB
Service - Delete.bat	29/08/2020 23:25	Arquivo em Lotes ...	1 KB
Service - Maker.bat	29/08/2020 23:26	Arquivo em Lotes ...	1 KB

PROGRAMAÇÃO BACK-END

Será aberta uma janela em Texto informando que o servidor iniciou (server started**).
Não feche esta janela**



A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.". The window contains the following text output from the command "E:\Portables\pgsql>E:\Portables\pgsql\bin\pg_ctl.exe start -D "E:\Portables\pgsql\data"":

```
E:\Portables\pgsql>"E:\Portables\pgsql\bin\pg_ctl.exe" start -D "E:\Portables\pgsql\data"
waiting for server to start....2024-02-18 14:26:40.716 -05 [18968] LOG:  iniciando PostgreSQL 12.4, compiled by Visual C
++ build 1914, 64-bit
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv6 «::1», port 5432
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv4 «127.0.0.1», port 5432
2024-02-18 14:26:40.770 -05 [1972] LOG:  el sistema de bases de datos fue apagado en 2024-02-05 18:23:29 -05
2024-02-18 14:26:40.791 -05 [18968] LOG:  el sistema de bases de datos está listo para aceptar conexiones
  done
server started
|
```

PROGRAMAÇÃO BACK-END

Cadastro de usuários

PROGRAMAÇÃO BACK-END

Templates

PROGRAMAÇÃO BACK-END

Utilizando a App **dvdrentalapp**. Na pasta **templates** crie os arquivos **salva.html** e **cadastro.html** digite o seguinte conteúdo:

salva.html

```
dvdrentalapp > templates > salva.html > html
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Usuario Salvo com sucesso</h1>
6
7
8  </body>
9  </html>
```

PROGRAMAÇÃO BÁSICA

cadastro.html

```
dvdrentalapp > templates > cadastro.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Cadastro de usuários</title>
6  </head>
7  <body>
8
9  <form method="post">
10     {% csrf_token %}
11
12     <div class="form-group">
13         {{form_usuario.username.errors}}
14         <label>Username: </label>
15         {{form_usuario.username}}
16     </div>
17
18     <div class="form-group">
19         {{form_usuario.password1.errors}}
20         <label>Senha: </label>
21         {{form_usuario.password1}}
22     </div>
23
24     <div class="form-group">
25         {{form_usuario.password2.errors}}
26         <label>Confirmação de senha: </label>
27         {{form_usuario.password2}}
28     </div>
29
30     <div class="form-group">
31         <input type="submit" value="Cadastrar" class="btn btn-primary">
32     </div>
33 </form>
34
35 </body>
36 </html>
37
```

PROGRAMAÇÃO BACK-END

Arquivos views.py e urls.py

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `views.py`

```
7  from django.contrib.auth.forms import UserCreationForm
8
9  def cadastrar_usuario(request):
10     if request.method == "POST":
11         form_usuario = UserCreationForm(request.POST)
12         if form_usuario.is_valid():
13             form_usuario.save()
14             return redirect('salva')
15     else:
16         form_usuario = UserCreationForm()
17     return render(request, 'cadastro.html', {'form_usuario': form_usuario})
18
19 def salva(request):
20     return render(request, 'salva.html')
```

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `urls.py`

```
10     path('cadastrar_usuario', views.cadastrar_usuario, name="cadastrar_usuario"),
11     path('salva', views.salva, name="salva"),
```

PROGRAMAÇÃO BACK-END

Arquivos settings.py

PROGRAMAÇÃO BACK-END

Na pasta `my_ads` verifique que o seguinte conteúdo no arquivo `settings.py`

```
99  # Password validation
100 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-passwordValidators
101
102 AUTH_PASSWORD_VALIDATORS = [
103     {
104         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
105     },
106     {
107         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
108     },
109     {
110         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
111     },
112     {
113         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
114     },
115 ]
```

Este conteúdo é muito importante para exigir uma senha forte.

PROGRAMAÇÃO BACK-END

Vamos alterar o conteúdo da seção **AUTH_PASSWORD_VALIDATORS** do arquivo **settings.py** para exigir uma senha fraca

```
99  # Password validation
100 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-passwordValidators
101
102 AUTH_PASSWORD_VALIDATORS = [
103     ]
104 ]
```

Um projeto em produção, deve ser mantido ou alterado com o conteúdo constando no slide anterior.

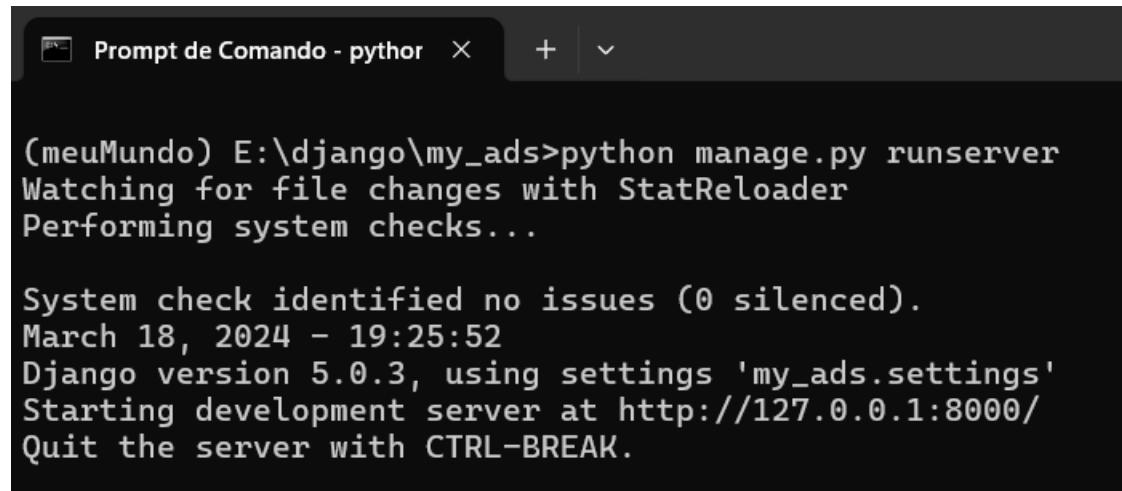
PROGRAMAÇÃO BACK-END

Iniciando o servidor
e testando

PROGRAMAÇÃO BACK-END

Execute na pasta **my_ads** o comando:

python manage.py runserver



```
(meuMundo) E:\django\my_ads>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2024 - 19:25:52
Django version 5.0.3, using settings 'my_ads.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Vá ao navegador e digite:

http://127.0.0.1:8000/cadastrar_usuario/

PROGRAMAÇÃO BACK-END

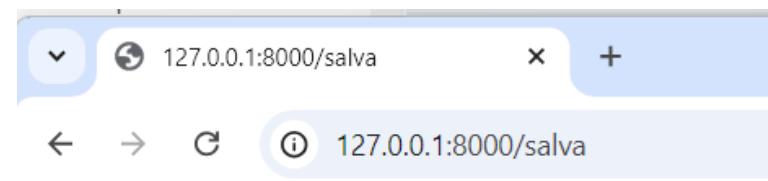
O resultado da página será o formulário para cadastrar um novo usuário.

The screenshot shows a web browser window with the following details:

- Title bar: Cadastro de usuários
- Address bar: 127.0.0.1:8000/cadastrar_usuario
- Form fields:
 - Username: [input field]
 - Senha: [input field]
 - Confirmação de senha: [input field]
- Action button: Cadastrar

A blue arrow points from the text "Digite os dados do novo usuário e acione o botão Cadastrar" to the "Cadastrar" button.

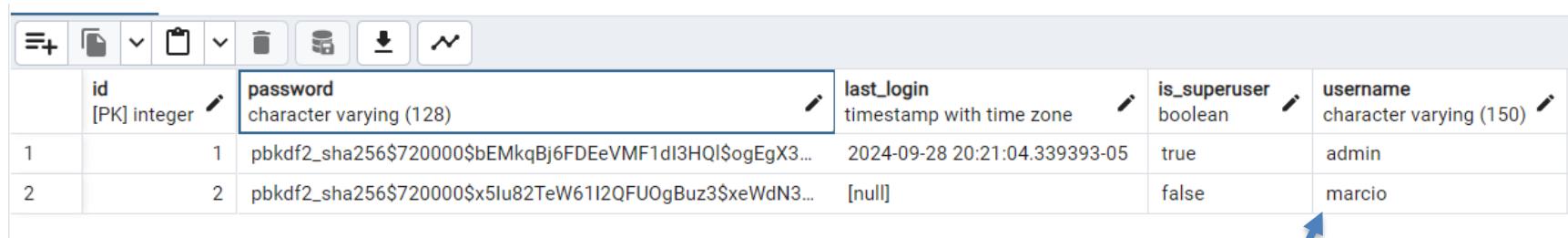
Digite os dados do novo usuário e acione o botão **Cadastrar**



Usuario Salvo com sucesso

PROGRAMAÇÃO BACK-END

Utilizando o Pgadmin4 verifique na tabela auth_user o novo usuário.



	id [PK] integer	password character varying (128)	last_login timestamp with time zone	is_superuser boolean	username character varying (150)
1	1	pbkdf2_sha256\$720000\$bEMkqBj6FDEeVMF1dl3HQI\$ogEgX3...	2024-09-28 20:21:04.339393-05	true	admin
2	2	pbkdf2_sha256\$720000\$x5lu82TeW61I2QFU0gBuz3\$xeWdN3...	[null]	false	marcio

PROGRAMAÇÃO BACK-END

Autenticando usuários

PROGRAMAÇÃO BACK-END

Templates

PROGRAMAÇÃO BACK-END

Utilizando a App **dvdrentalapp**. Na pasta **templates** crie os arquivos **logado.html** e **login.html** digite os seguintes conteúdos:

logado.html

```
dvdrentalapp > templates > logado.html > ...
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Usuario logado com sucesso</h1>
6
7
8  </body>
9  </html>
```

PROGRAMAÇÃO

login.html

```
dvdrentalapp > templates > login.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <title>Title</title>
6   </head>
7   <body>
8   <form method="post">
9       {% csrf_token %}
10
11      <div class="form-group">
12          {{form_login.username.errors}}
13          <label>Username: </label>
14          {{form_login.username}}
15      </div>
16
17      <div class="form-group">
18          {{form_login.password.errors}}
19          <label>Senha: </label>
20          {{form_login.password}}
21      </div>
22
23      <div class="form-group">
24          <input type="submit" value="Enviar" class="btn btn-primary">
25      </div>
26  </form>
27  </body>
28  </html>
```

PROGRAMAÇÃO BACK-END

Arquivos views.py e urls.py

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `views.py`

```
7  from django.contrib.auth.forms import UserCreationForm, AuthenticationForm
8  from django.contrib.auth import authenticate, login
9
10 def logar_usuario(request):
11     if request.method == "POST":
12         username = request.POST["username"]
13         password = request.POST["password"]
14         usuario = authenticate(request, username=username, password=password)
15         if usuario is not None:
16             login(request, usuario)
17             return redirect('logado')
18         else:
19             form_login = AuthenticationForm()
20     else:
21         form_login = AuthenticationForm()
22     return render(request, 'login.html', {'form_login': form_login})
23
24 def logado(request):
25     return render(request, 'logado.html')
```

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `urls.py`

```
10     path('cadastrar_usuario', views.cadastrar_usuario, name="cadastrar_usuario"),
11     path('salva', views.salva, name="salva"),
12     path('logar_usuario', views.logar_usuario, name="logar_usuario"),
13     path('logado', views.logado, name="logado"),
```

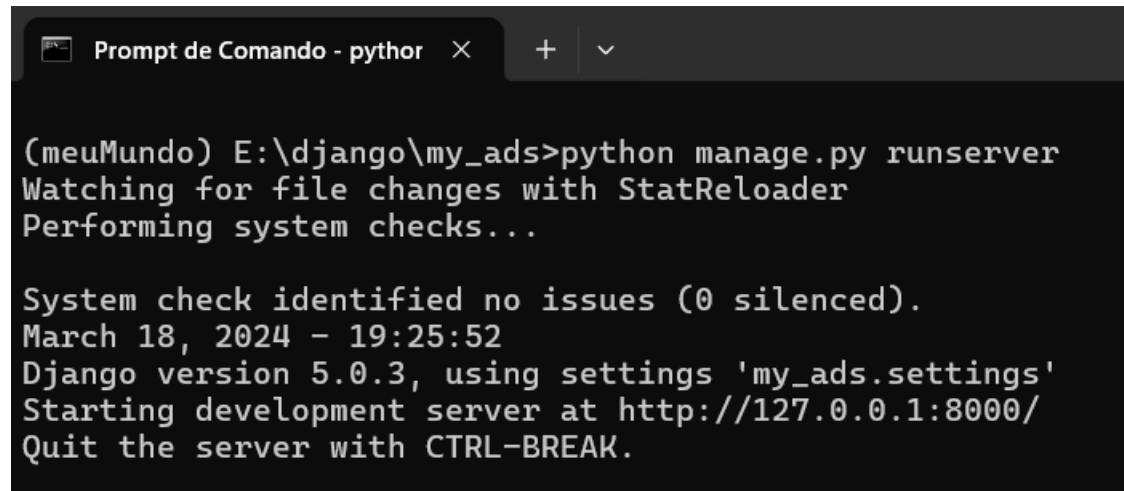
PROGRAMAÇÃO BACK-END

Iniciando o servidor
e testando

PROGRAMAÇÃO BACK-END

Execute na pasta **my_ads** o comando:

python manage.py runserver



```
(meuMundo) E:\django\my_ads>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2024 - 19:25:52
Django version 5.0.3, using settings 'my_ads.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Vá ao navegador e digite:

http://127.0.0.1:8000/logar_usuario/

PROGRAMAÇÃO BACK-END

O resultado da página será o formulário para logar.

The image shows two adjacent browser windows. The left window has a title bar 'Title' and a URL bar '127.0.0.1:8000/logar_usuario'. It contains a login form with fields for 'Username' and 'Senha', and a 'Enviar' button. The right window has a title bar '127.0.0.1:8000/logado' and a URL bar '127.0.0.1:8000/logado'. It displays the message 'Usuario logado com sucesso'. A blue arrow points from the text 'Envie' in the first slide to the 'Enviar' button in the screenshot.

Digite os dados do usuário e acione o botão **Enviar**

Usuario logado com sucesso

PROGRAMAÇÃO BACK-END

Pode-se verificar no Google Chrome a **sessionid**.

The screenshot shows the Google Chrome DevTools interface with the "Application" tab selected. On the left, the "Storage" section is expanded, showing "Session storage" and "Cookies". Under "Cookies", there is a entry for the domain "http://127.0.0.1:8000". The main area displays a table of cookies:

Name	Value	Do...	Path	Ex...	Size	Htt...	Se...	Sa...	Pa...	Cr...
csrfToken	VVjfVzew6iPMjCLb2TX7Exv...	12...	/	20...	41			Lax		
sessionid	gzlcnh78b8zjqufs0yarstvm9...	12...	/	20...	41	✓		Lax		

At the bottom, a message reads: "Select a cookie to preview its value".

PROGRAMAÇÃO BACK-END

**Logout de usuários
e alteração de senha**

PROGRAMAÇÃO BACK-END

Templates

PROGRAMAÇÃO BACK-END

Utilizando a App **dvdrentalapp**. Na pasta **templates** crie/altere os arquivos **home.html**, **alterar_senha.html**, **login.html** e **logado.html** digitando os seguintes conteúdos:

PROGRAMAÇÃO BACK-END

home.html

```
dvdrentalapp > templates > home.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Página Inicial</title>
6
7 </head>
8 <body>
9     <div class="container">
10        <h1>Bem-vindo à Aplicação Django</h1>
11        <div class="buttons">
12            <a href="{% url 'cadastrar_usuario' %}"><button>Cadastrar Usuário</button></a><br>
13            <a href="{% url 'logar_usuario' %}"><button>Logar</button></a>
14        </div>
15    </div>
16 </body>
17 </html>
```

PROGRAMAÇÃO BÁSICA

alterar_senha.html

```
dvdrentalapp > templates > alterar_senha.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Alterar senha</title>
6  </head>
7  <body>
8      <h1>Alterar Senha</h1>
9  <form method="post">
10     {% csrf_token %}
11
12     <div class="form-group">
13         {{form_senha.old_password.errors}}
14         <label>Senha antiga: </label>
15         {{form_senha.old_password}}
16     </div>
17
18     <div class="form-group">
19         {{form_senha.new_password1.errors}}
20         <label>Nova senha: </label>
21         {{form_senha.new_password1}}
22     </div>
23
24     <div class="form-group">
25         {{form_senha.new_password2.errors}}
26         <label>Confirmação de senha: </label>
27         {{form_senha.new_password2}}
28     </div>
29
30     <div class="form-group">
31         <input type="submit" value="Alterar" class="btn btn-primary">
32     </div>
33 </form>
34
35 </body>
36 </html>
```

PROGRAMAÇÃO BACK-END

login.html

```
dvdrentalapp > templates > login.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Title</title>
6  </head>
7  <body>
8  |   <h1>Logar no Sistema</h1>
9  <form method="post">
10 |   {% csrf_token %}
11
12 <div class="form-group">
13 |   {{form_login.username.errors}}
14 |   <label>Username: </label>
15 |   {{form_login.username}}
16 </div>
17
18 <div class="form-group">
19 |   {{form_login.password.errors}}
20 |   <label>Senha: </label>
21 |   {{form_login.password}}
22 </div>
23
24 <div class="form-group">
25 |   <input type="submit" value="Enviar" class="btn btn-primary">
26 </div>
27 </form>
28 </body>
29 </html>
```

PROGRAMAÇÃO BACK-END

logado.html

```
dvdrentalapp > templates > logado.html > ...
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Usuario logado com sucesso</h1>
6  <div class="buttons">
7      <a href="#">{% url 'alterar_senha' %}"><button>Alterar Senha</button></a><br>
8      <a href="#">{% url 'deslogar_usuario' %}"><button>Logout</button></a>
9  </div>
10
11 </body>
12 </html>
```

PROGRAMAÇÃO BACK-END

Arquivos views.py e urls.py

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `views.py`

```
 7  from django.contrib.auth.forms import UserCreationForm, AuthenticationForm
 8  from django.contrib.auth import authenticate, login, logout
 9  from django.contrib.auth.forms import PasswordChangeForm
10  from django.contrib.auth import update_session_auth_hash
11
12  def deslogar_usuario(request):
13      logout(request)
14      return redirect('home')
15
16  def alterar_senha(request):
17      if request.method == "POST":
18          form_senha = PasswordChangeForm(request.user, request.POST)
19          if form_senha.is_valid():
20              user = form_senha.save()
21              update_session_auth_hash(request, user)
22              return redirect('home')
23          else:
24              form_senha = PasswordChangeForm(request.user)
25          return render(request, 'alterar_senha.html', {'form_senha': form_senha})
26
27  def home(request):
28      return render(request, 'home.html')
```

PROGRAMAÇÃO BACK-END

Na pasta `dvdrentalapp` adicione o seguinte conteúdo no arquivo `urls.py`

```
10     path('cadastrar_usuario/', views.cadastrar_usuario, name="cadastrar_usuario"),
11     path('salva/', views.salva, name="salva"),
12     path('logar_usuario/', views.logar_usuario, name="logar_usuario"),
13     path('logado/', views.logado, name="logado"),
14     path('alterar_senha/', views.alterar_senha, name='alterar_senha'),
15     path('deslogar_usuario/', views.deslogar_usuario, name="deslogar_usuario"),
16     path('home/', views.home, name="home"),
```

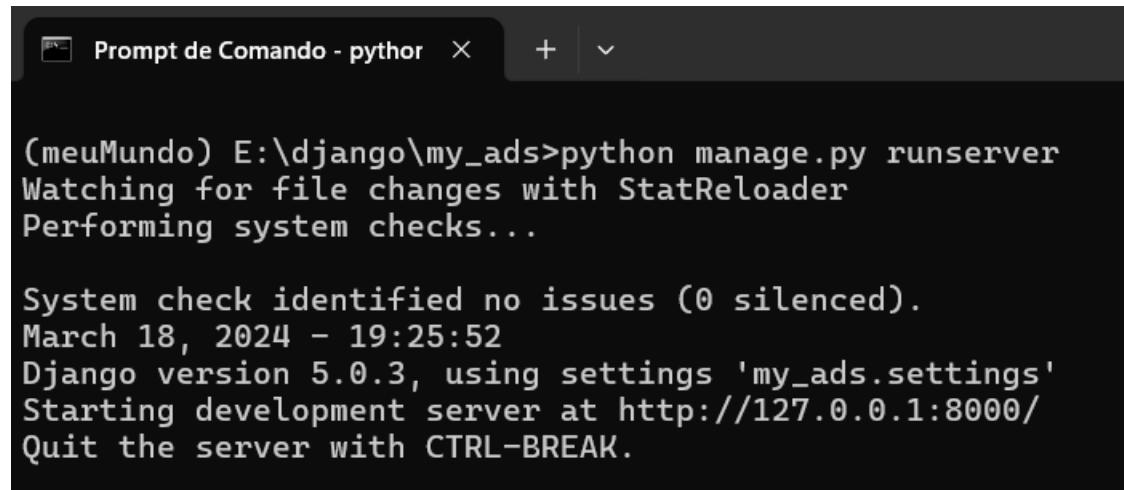
PROGRAMAÇÃO BACK-END

Iniciando o servidor
e testando

PROGRAMAÇÃO BACK-END

Execute na pasta **my_ads** o comando:

python manage.py runserver



```
(meuMundo) E:\django\my_ads>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2024 - 19:25:52
Django version 5.0.3, using settings 'my_ads.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Vá ao navegador e digite:

http://127.0.0.1:8000/home/

PROGRAMAÇÃO BACK-END

O resultado da página será o formulário home.

A screenshot of a web browser window titled "Página Inicial". The address bar shows the URL "127.0.0.1:8000/home/". The main content area displays the text "Bem-vindo à Aplicação Django" and two buttons: "Cadastrar Usuário" and "Logar".

A screenshot of a web browser window titled "Cadastro de usuários". The address bar shows the URL "127.0.0.1:8000/cadastrar_usuario/". The main content area displays the text "Cadastro de Usuário" and three input fields: "Username", "Senha", and "Confirmação de senha", followed by a "Cadastrar" button.

A screenshot of a web browser window titled "Title". The address bar shows the URL "127.0.0.1:8000/logar_usuario/". The main content area displays the text "Logar no Sistema" and three input fields: "Username", "Senha", and "Enviar".

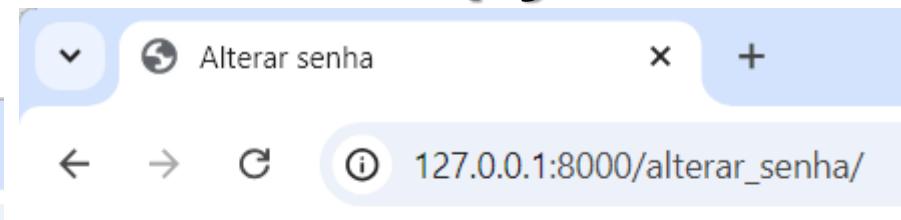
A screenshot of a web browser window titled "Title". The address bar shows the URL "127.0.0.1:8000/logar_usuario/". The main content area displays the text "Logar no Sistema" and three input fields: "Username", "Senha", and "Enviar".

PROGRAMAÇÃO BACK-END

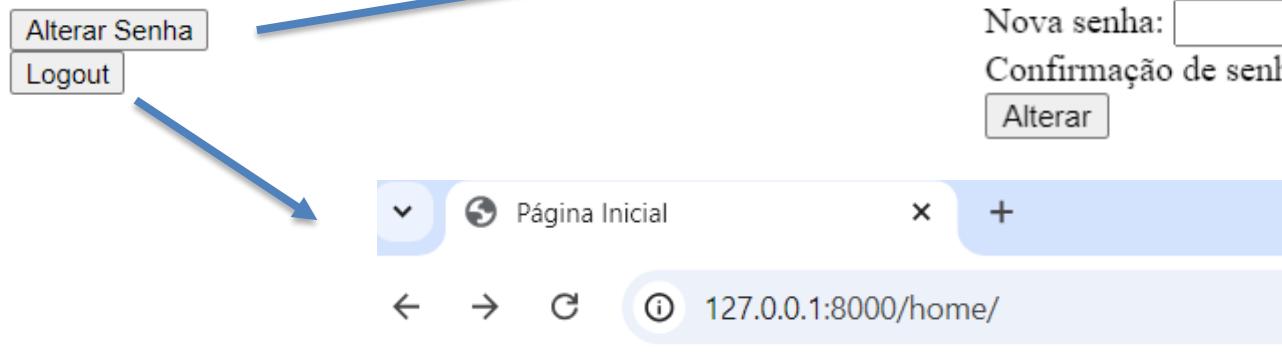
Ao **logar** no sistema teremos as opções na seguinte tela:



Usuario logado com sucesso



Alterar Senha



Bem-vindo à Aplicação Django