

## Mãos na massa: Autorização de usuários

Siga os passos abaixo para implementar o que foi visto nos vídeos!

Clicando no link a seguir, você consegue fazer o [download dos arquivos da aula 05 \(https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/05/aula5.zip\)](https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/05/aula5.zip).

1) Modifique a função **novo** para checar se o usuário está logado, veja como ela deve ficar abaixo:

```
@app.route('/novo')
def novo():
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect('/login')
    return render_template('novo.html', titulo='Novo Jogo')
```

2) Na mesma função, indique o **próximo** caminho que se deve seguir após o login, repare no **redirect**:

```
@app.route('/novo')
def novo():
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect('/login?proxima=novo')
    return render_template('novo.html', titulo='Novo Jogo')
```

3) Agora na função **login**, altere-a conforme abaixo:

```
@app.route('/login')
def login():
    proxima = request.args.get('proxima')
    return render_template('login.html', proxima=proxima)
```

4) Abra o arquivo **login.html** e adicione um **input** abaixo do **form**, veja como ficará abaixo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Jogoteca</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
  </head>
  <body>
    <div class="container">
      {% with messages = get_flashed_messages() %}
        {% if messages %}
          <ul id="messages" class="list_unstyled">
            {% for message in messages %}
              <li class="alert alert-sucess">{{ message }}</li>
            {% endfor %}
          </ul>
        {% endif %}
      {% endwith %}
    </div>
  </body>
</html>
```

```

        {% endfor %}
    </ul>
    {% enfif %}
{% endwith %}

<h1>Faça seu login</h1>
<form method="POST" action="/autenticar">
    <input type="hidden" name="proxima" value="{{ proxima }}">
    <p><label>Nome de usuário:</label> <input class="form-control" type="text" name="usu
    <p><label>Senha:</label> <input class="form-control" type="password" name="senha" r
    <p><button class="btn btn-primary" type="submit">Entrar</button></p>
</form>
</div>
</body>
</html>

```

5) Altere a função da rota de `/autenticar` e modifique o `redirect` removendo a rota **hardcoded**, veja a função modificada abaixo:

```

@app.route('/autenticar', methods=['POST', ])
def autenticar():
    if 'mestra' == request.form['senha']:
        session['usuario_logado'] = request.form['usuario']
        flash(request.form['usuario'] + ' logou com sucesso!')
        proxima_pagina = request.form['proxima']
        return redirect('/{}'.format(proxima_pagina))
    else:
        flash('Não logado, tente novamente!')
        return redirect('/login')

```

6) Teste as modificações em seu navegador!

7) Se acessarmos diretamente a rota de login, temos um problema, nossa app nos leva para uma rota `/None`, para resolver isto modifique o `input type hidden` que foi colocado e teste a aplicação. Veja abaixo como deve ficar:

```

<input type="hidden" name="proxima" value="{{ proxima or '/' }}">

```

8) Modifique as funções de rota para utilizar o `url_for`, retirando as partes **hardcoded** (lembre-se de importar o `url_for` no topo do código)

```

from flask import Flask, render_template, request, redirect, session, flash, url_for

app = Flask(__name__)
app.secret_key = 'alura'

class Jogo:
    def __init__(self, nome, categoria, console):
        self.nome = nome
        self.categoria = categoria
        self.console = console

jogo1 = Jogo('Super Mario', 'Ação', 'SNES')

```

```

jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

@app.route('/')
def index():
    return render_template('lista.html', titulo='Jogos', jogos=lista)

@app.route('/novo')
def novo():
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect(url_for('login', proxima=url_for('novo')))
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar', methods=['POST',])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    lista.append(jogo)
    return redirect(url_for('index'))

@app.route('/login')
def login():
    proxima = request.args.get('proxima')
    return render_template('login.html', proxima=proxima)

@app.route('/autenticar', methods=['POST', ])
def autenticar():
    if 'mestra' == request.form['senha']:
        session['usuario_logado'] = request.form['usuario']
        flash(request.form['usuario'] + ' logou com sucesso!')
        proxima_pagina = request.form['proxima']
        return redirect(proxima_pagina)
    else:
        flash('Não logado, tente novamente!')
        return redirect(url_for('login'))

@app.route('/logout')
def logout():
    session['usuario_logado'] = None
    flash('Nenhum usuário logado!')
    return redirect(url_for('index'))

app.run(debug=True)

```

9) No arquivo **login.html**, modifique a tag **form** e o **input type hidden**, veja abaixo:

```

<form method="POST" action="{ url_for('autenticar') }">
<input type="hidden" name="proxima" value="{ proxima or url_for('index') }">
...
</form>

```

10) Agora no arquivo **novo.html**, modifique também a tag **form** para utilizar o **url\_for**:

```
<form action="{ url_for('criar') }}" method="post">
...
</form>
```

11) Teste sua app e veja se está tudo ok!

12) Crie uma classe Usuario para permitir multiplos usuários logados, faça um teste instanciando alguns usuários e crie uma lista para armazená-los, veja o código inteiro abaixo:

```
from flask import Flask, render_template, request, redirect, session, flash, url_for
```

```
app = Flask(__name__)
app.secret_key = 'alura'
```

```
class Jogo:
    def __init__(self, nome, categoria, console):
        self.nome = nome
        self.categoria = categoria
        self.console = console
```

```
class Usuario:
    def __init__(self, id, nome, senha):
        self.id = id
        self.nome = nome
        self.senha = senha
```

```
usuario1 = Usuario('luan', 'Luiz Antonio Marques', '1234')
usuario2 = Usuario('Nico', 'Nico Steppat', '7a1')
usuario3 = Usuario('flavio', 'flavio Almeida', 'javascript')
```

```
usuarios = {usuario1.id: usuario1,
             usuario2.id: usuario2,
             usuario3.id: usuario3}
```

```
jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]
```

```
@app.route('/')
def index():
    return render_template('lista.html', titulo='Jogos', jogos=lista)
```

```
@app.route('/novo')
def novo():
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect(url_for('login', proxima=url_for('novo')))
    return render_template('novo.html', titulo='Novo Jogo')
```

```
@app.route('/criar', methods=['POST',])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
```

```

jogo = Jogo(nome, categoria, console)
lista.append(jogo)
return redirect(url_for('index'))

@app.route('/login')
def login():
    proxima = request.args.get('proxima')
    return render_template('login.html', proxima=proxima)

@app.route('/autenticar', methods=['POST', ])
def autenticar():
    if 'mestra' == request.form['senha']:
        session['usuario_logado'] = request.form['usuario']
        flash(request.form['usuario'] + ' logou com sucesso!')
        proxima_pagina = request.form['proxima']
        return redirect(proxima_pagina)
    else:
        flash('Não logado, tente novamente!')
        return redirect(url_for('login'))

@app.route('/logout')
def logout():
    session['usuario_logado'] = None
    flash('Nenhum usuário logado!')
    return redirect(url_for('index'))

app.run(debug=True)

```

13) Modifique a função **autenticar** para checar se esse usuário existe na lista criada:

```

from flask import Flask, render_template, request, redirect, session, flash, url_for

app = Flask(__name__)
app.secret_key = 'alura'

class Jogo:
    def __init__(self, nome, categoria, console):
        self.nome = nome
        self.categoria = categoria
        self.console = console

class Usuario:
    def __init__(self, id, nome, senha):
        self.id = id
        self.nome = nome
        self.senha = senha

usuario1 = Usuario('luan', 'Luiz Antonio Marques', '1234')
usuario2 = Usuario('Nico', 'Nico Steppat', '7a1')
usuario3 = Usuario('flavio', 'flavio Almeida', 'javascript')

usuarios = {usuario1.id: usuario1,
            usuario2.id: usuario2,
            usuario3.id: usuario3}

```

```
jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

@app.route('/')
def index():
    return render_template('lista.html', titulo='Jogos', jogos=lista)

@app.route('/novo')
def novo():
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect(url_for('login', proxima=url_for('novo')))
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar', methods=['POST',])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    lista.append(jogo)
    return redirect(url_for('index'))

@app.route('/login')
def login():
    proxima = request.args.get('proxima')
    return render_template('login.html', proxima=proxima)

@app.route('/autenticar', methods=['POST', ])
def autenticar():
    if request.form['usuario'] in usuarios:
        usuario = usuarios[request.form['usuario']]
        if usuario.senha == request.form['senha']:
            session['usuario_logado'] = usuario.id
            flash(usuario.nome + ' logou com sucesso!')
            proxima_pagina = request.form['proxima']
            return redirect(proxima_pagina)
    flash('Não logado, tente novamente!')
    return redirect(url_for('login'))

@app.route('/logout')
def logout():
    session['usuario_logado'] = None
    flash('Nenhum usuário logado!')
    return redirect(url_for('index'))

app.run(debug=True)
```

