

02

## Criando tela de Login

### Transcrição

Nessa aula, nós iremos implementar um pouco de segurança na nossa aplicação, fazendo com que somente quem tiver um usuário possa criar novos jogos. Assim, sempre que a rota `/novo` for acessada, verificaremos de alguma forma se essa pessoa está logada. Se não, ela será enviada para a tela de login (`/login`), na qual preencherá os dados **usuário** e **senha** para fazer a autenticação, que será enviada para o servidor.

O Flask verificará se esses dados condizem com aqueles no servidor, e o usuário finalmente retornará para a rota `/novo`, na qual poderá fazer alterações na lista.

E é isso que iremos fazer, sempre com pequenos passos. O primeiro deles é criar uma nova rota, cujo caminho será `/login`. Nela, teremos a função `login()`, que deve renderizar um *template* com o formulário de login — que ainda não criamos, mas que podemos deixar definido.

```
@app.route('/login')
def login():
    return render_template('login.html')
```

Clicando com o botão direito do mouse em "templates" e, em seguida, em "New > HTML File", criaremos o arquivo `login.html`. O código que usaremos nessa página está disponível 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">
      <h1>Faça seu login</h1>
      <form method="POST" action="/autenticar">
        <p><label>Nome de usuário:</label> <input class="form-control" type="text" name="usuario">
        <p><label>Senha:</label> <input class="form-control" type="password" name="senha" required>
        <p><button class="btn btn-primary" type="submit">Entrar</button></p>
      </form>
    </div>
  </body>
</html>
```

Nesse código, temos um `<form>` que coloca os dados desse formulário dentro de um método `POST`, e eles são enviados para o servidor por meio de uma rota chamada `/autenticar`, que não foi criada ainda. Nessa rota, serão verificados os `inputs` `usuario` e `senha`. Logo abaixo, temos o botão `Entrar`, que fará a submissão dos dados.

No navegador, ao acessar a página <http://127.0.0.1:5000/login> (<http://127.0.0.1:5000/login>), somos direcionados a uma tela com a mensagem "Faça seu login" e um formulário com os campos "Nome de usuário" e "Senha".

## Faça seu login

Nome de usuário:

Senha:

**Entrar**

Se tentarmos entrar com um login e senha qualquer, o navegador não encontrará nada, pois ainda não temos uma página `/autenticar`. Antes de criarmos essa rota, vamos refletir: no que consiste uma autenticação?

Basicamente, primeiro faremos uma verificação se o usuário e a senha existem. Nesse caso, por enquanto, verificaremos somente a senha. Em seguida, temos que direcionar esse usuário para algum lugar.

Por exemplo, se ele veio da página `/novo`, ele deve passar por `/login` e retornar à página `/novo`. Para simplificar, por enquanto vamos mandá-lo para a página de lista (`/`).

Agora vamos criar a nossa rota `/autenticar`, que contém uma função com o mesmo nome. Para simplificar, vamos definir que todos usuários sabem a senha — no caso, `mestra`. Se `mestra` for igual à senha passada pelo usuário, que pegaremos no campo `senha` do objeto `request` (`request.form['senha']`), ele deve ser redirecionado para `/`.

No entanto, se a senha estiver errada, esse usuário deve retornar para a página de login. Você já deve ter reparado que esquecemos algo, certo? Para não recebermos um erro "*Method Not Allowed*", temos que passar o método `POST` para a rota `/autenticar`. Assim, teremos:

```
@app.route('/autenticar', methods=['POST'])
def autenticar():
    if 'mestra' == request.form['senha']:
        return redirect('/')
    else :
        return redirect ('/login')
```

Acessando a URL <http://127.0.0.1:5000/login> (<http://127.0.0.1:5000/login>) e entrando com qualquer usuário e a senha "mestra", seremos redirecionados para a tela inicial (<http://127.0.0.1:5000/>). Por outro lado, se utilizarmos uma senha qualquer, seremos redirecionados para <http://127.0.0.1:5000/login> (<http://127.0.0.1:5000/login>) novamente. Porém, em qualquer um desses casos, não temos nenhum *feedback* de que estamos logados ou não. Ou seja, a usabilidade não está boa.

Vamos melhorar isso, adiante.