

Lista 7

Gerado com IA

Exercício: Adicionando Cadastro e Listagem de Usuários

Objetivo: Expandir a aplicação anterior para permitir o cadastro de múltiplos usuários e listar todos os usuários cadastrados, garantindo que apenas usuários logados possam acessar a lista.

Passo 1: Atualizando o Arquivo `app.py`

1. Configure o armazenamento dos usuários:

- Modifique ou adicione o seguinte código no início do arquivo:

```
usuarios_registrados = [] # Lista para armazenar usuários cadastrados
```

2. Adicione a rota `/cadastro`:

- Implemente a funcionalidade de cadastro:

```
@app.route('/cadastro', methods=['GET', 'POST'])
def cadastro():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        nome = request.form['nome']

        # Verificar se o usuário já existe
        for usuario in usuarios_registrados:
            if usuario['username'] == username:
                return render_template('cadastro.html', erro='Usuário
já cadastrado.')

        # Adicionar novo usuário à lista
        usuarios_registrados.append({'username': username, 'password':
password, 'nome': nome})
        return redirect(url_for('login'))

    return render_template('cadastro.html')
```

3. Atualize a rota `/login`:

- Modifique a rota de login para validar o usuário contra os dados cadastrados:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        # Validar credenciais
        for usuario in usuarios_registrados:
            if usuario['username'] == username and usuario['password']
            == password:
                session['username'] = username
                resposta =
                make_response(redirect(url_for('dashboard')))
                resposta.set_cookie('username', username,
                max_age=60*60*24)
                return resposta

        return render_template('login.html', erro='Usuário ou senha
        inválidos.')

    if 'username' in session:
        return redirect(url_for('dashboard'))

    return render_template('login.html')
```

4. **Objetivo deste passo:** Permitir que novos usuários sejam cadastrados e utilizem essas credenciais para fazer login.

Passo 2: Criando os Templates

1. Template de Cadastro (`templates/cadastro.html`):

- Crie o arquivo `cadastro.html` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Cadastro</title>
</head>
<body>
    <h1>Cadastro de Usuário</h1>
    <form action="{ url_for('cadastro') }" method="post">
        <label for="nome">Nome Completo:</label>
        <input type="text" id="nome" name="nome" required><br><br>
        <label for="username">Usuário:</label>
        <input type="text" id="username" name="username" required><br>
    <br>
        <label for="password">Senha:</label>
        <input type="password" id="password" name="password" required>
```

```
<br><br>
    <button type="submit">Cadastrar</button>
</form>
{% if erro %}
    <p style="color:red;">{{ erro }}</p>
{% endif %}
<a href="{{ url_for('login') }}">Já possui uma conta? Faça
login</a>
</body>
</html>
```

2. Template de Listagem de Usuários (templates/usuarios.html):

- Crie o arquivo `usuarios.html`:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Usuários Registrados</title>
</head>
<body>
    <h1>Usuários Registrados</h1>
    <ul>
        {% for usuario in usuarios %}
            <li>{{ usuario.nome }} ({{ usuario.username }})</li>
        {% endfor %}
    </ul>
    <a href="{{ url_for('dashboard') }}">Voltar ao Dashboard</a>
</body>
</html>
```

3. Objetivo deste passo: Criar as interfaces de cadastro e listagem de usuários.

Passo 3: Adicionando a Rota de Listagem

1. Adicione a rota `/usuarios`:

- Atualize o `app.py` com a funcionalidade de listagem:

```
@app.route('/usuarios')
def listar_usuarios():
    if 'username' not in session:
        return redirect(url_for('login'))

    return render_template('usuarios.html',
        usuarios=usuarios_registrados)
```

2. Adicione um link para o cadastro e listagem no template de login e dashboard:

- No `dashboard.html`, inclua:

```
<a href="{{ url_for('listar_usuarios') }}">Listar Usuários</a><br>
```

- No `login.html`, inclua:

```
<a href="{{ url_for('cadastro') }}">Não tem uma conta? Cadastre-se</a>
```

Passo 4: Testando a Aplicação

1. Inicie a aplicação:

- Execute o servidor com:

```
python app.py
```

2. Teste o cadastro e login:

- Acesse `http://127.0.0.1:5000/cadastro` para registrar novos usuários.
- Faça login com o usuário recém-cadastrado.
- Acesse `http://127.0.0.1:5000/usuarios` e verifique que a lista exibe todos os usuários cadastrados.

3. Verifique a proteção da rota:

- Tente acessar `/usuarios` sem fazer login e confirme que você é redirecionado para a página de login.

Conclusão do Exercício

Este exercício abordou:

1. Cadastro de múltiplos usuários.
2. Validação de credenciais no login.
3. Proteção de rota para listar usuários cadastrados.

Desafio Extra:

1. Adicione uma validação para evitar usernames duplicados no cadastro.
2. Crie um campo opcional "Função" no cadastro (e.g., `admin` ou `usuário`) e mostre apenas usuários com a mesma função ao listar.