

# Material de Estudo: Flask + Flask-Login (PSI)

## Sistema Operacional:

Windows

Fonte: Arial 12, com títulos em negrito e ampliados para melhor leitura.

## 1. Ambiente Virtual e Instalação

Para criar o ambiente virtual no Windows:

1. Abra o terminal (cmd ou PowerShell).
2. Execute: `python -m venv venv`
3. Ative com: `venv\Scripts\activate`
4. Instale os pacotes com: `pip install flask flask-login`

## 2. Estrutura Básica

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def home():
```

```
    return "Olá, Flask!"
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

## 3. Rotas e Métodos HTTP

```
@app.route('/form')
```

```
def form():
```

## Material de Estudo: Flask + Flask-Login (PSI)

```
return "<form action='/resultado' method='post'>
    <input name='nome'>
    <input type='submit'>
</form>"
```

```
@app.route('/resultado', methods=['POST'])
def resultado():
    nome = request.form['nome']
    return f"Nome: {nome}"
```

### 4. request e make\_response

```
@app.route('/resposta')
def resposta():
    nome = request.args.get('nome', 'Visitante')
    resposta = make_response(f"Olá, {nome}")
    resposta.headers['Custom-Header'] = 'Valor'
    return resposta
```

### 5. Cookies e Sessão

```
@app.route('/setcookie')
def setcookie():
    resp = make_response("Cookie salvo!")
    resp.set_cookie('usuario', 'Livia')
    return resp
```

```
@app.route('/getcookie')
def getcookie():
    usuario = request.cookies.get('usuario')
```

## Material de Estudo: Flask + Flask-Login (PSI)

```
return f"Usuário: {usuario}"
```

```
app.secret_key = 'segredo-super-seguro'
```

```
@app.route('/login')
```

```
def login():
```

```
    session['usuario'] = 'Lívia'
```

```
    return 'Logado!'
```

```
@app.route('/perfil')
```

```
def perfil():
```

```
    return f"Usuário atual: {session.get('usuario')}"
```

### 6. url\_for

```
@app.route('/inicio')
```

```
def inicio():
```

```
    return redirect(url_for('home'))
```

### 7. Templates (Jinja2)

base.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title>{% block title %}{% endblock %}</title></head>
```

```
<body>
```

```
    {% include 'navbar.html' %}
```

```
    {% with messages = get_flashed_messages() %}
```

```
        {% if messages %}
```

## Material de Estudo: Flask + Flask-Login (PSI)

```
<ul>{% for m in messages %}<li>{{ m }}</li>{% endfor %}</ul>
{% endif %}
{% endwith %}
{% block content %}{% endblock %}
</body>
</html>
```

home.html:

```
{% extends "base.html" %}

{% block title %}Página Inicial{% endblock %}
{% block content %}
    <h1>Bem-vinda!</h1>
{% endblock %}
```

```
@app.route('/mensagem')
def mensagem():
    flash("Você acessou a página com sucesso!")
    return redirect(url_for('home'))
```

### 8. Flask-Login

```
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'
```

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

## Material de Estudo: Flask + Flask-Login (PSI)

```
usuarios = {'admin': {'senha': '123'}}
```

```
@login_manager.user_loader
```

```
def load_user(user_id):  
    return Usuario(user_id)
```

```
@app.route('/login', methods=['GET', 'POST'])
```

```
def login():  
    if request.method == 'POST':  
        nome = request.form['usuario']  
        senha = request.form['senha']  
        if nome in usuarios and usuarios[nome]['senha'] == senha:  
            user = Usuario(nome)  
            login_user(user)  
            return redirect(url_for('protegida'))  
        else:  
            flash("Credenciais inválidas!")  
    return render_template('login.html')
```

```
@app.route('/logout')
```

```
@login_required
```

```
def logout():  
    logout_user()  
    return redirect(url_for('login'))
```

```
@app.route('/protegida')
```

```
@login_required
```

```
def protegida():  
    return "Você está logado e pode ver isso!"
```

## 9. Resumo Final

## Material de Estudo: Flask + Flask-Login (PSI)

- Flask: framework web com rotas, sessões e templates.
- `@app.route()`: cria rotas (URLs).
- `request`: acesso a dados enviados.
- `make_response()`: personaliza resposta.
- `session`: guarda dados do usuário.
- `url_for()`: gera URLs dinamicamente.
- `render_template()`: carrega templates HTML.
- `flash()`: mensagens de uma vez (feedback).
- Flask-Login: sistema de autenticação.
- `login_user()`, `logout_user()`: controle de login.
- `@login_required`: exige login.
- `@login_manager.user_loader`: define como recuperar o usuário.