

# Avaliação Prática - Flask e Flask-Login



## Perguntas Teóricas

### Pergunta 1 (10 pontos)

Sobre o uso de sessões em Flask, qual a sua função?

Alternativas

- A) Armazenar todo o histórico de requisições HTTP feitas pelo navegador
- B) Guardar dados temporariamente que identificam o usuário entre diferentes requisições
- C) Gerar autenticação de chave simétrica de criptografia de ponta a ponta
- D) Controlar exclusivamente o fluxo de dados para páginas restritas
- E) Reduzir o tempo de carregamento de páginas, evitando requisições

✓ Resposta correta: B)

Explicação:

As sessões em Flask servem para armazenar dados temporários que identificam o usuário entre diferentes requisições. Por exemplo, quando um usuário faz login, sua identificação pode ser armazenada na sessão para manter o acesso.

---

### Pergunta 2 (10 pontos)

Em Flask-Login, para que serve a função `@login_required`?

Alternativas

- A) Verificar se o usuário já fez login anteriormente sem necessidade de novo login
- B) Carregar automaticamente os dados do usuário quando ele acessa qualquer rota
- C) Carregar o objeto do usuário a partir do identificador salvo na sessão
- D) Impedir que diferentes sessões sejam criadas com dados inválidos
- E) Criar novas credenciais para o usuário sem necessidade de senha

✓ Resposta correta: C)

Explicação:

A função `@login_required` é usada para proteger rotas. Ela carrega o usuário a partir do identificador salvo na sessão. Caso não exista um usuário autenticado, redirecionado para a página de login.



## Código Flask Completo (app.py)

```
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_login import LoginManager, UserMixin, login_user, login_required,
logout_user, current_user
from werkzeug.security import generate_password_hash, check_password_hash

app = Flask(__name__)
app.secret_key = "chave_secreta_123"

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = "login"

# "Banco" em memória
usuarios = {}

class User(UserMixin):
    def __init__(self, matricula, email, senha_hash):
        self.id = matricula
        self.email = email
        self.senha_hash = senha_hash

@login_manager.user_loader
def load_user(user_id):
    for user in usuarios.values():
        if user.id == user_id:
            return user
    return None

@app.route("/")
def home():
    return redirect(url_for("login"))

@app.route("/cadastro", methods=["GET", "POST"])
def cadastro():
    if request.method == "POST":
        matricula = request.form["matricula"]
        email = request.form["email"]
        senha = request.form["senha"]

        if not matricula or not email or not senha:
            flash("Todos os campos são obrigatórios.")
            return redirect(url_for("cadastro"))

        if email in usuarios:
            flash("Email já cadastrado.")
            return redirect(url_for("cadastro"))
```

```

        senha_hash = generate_password_hash(senha)
        user = User(matricula, email, senha_hash)
        usuarios[email] = user

        flash("Usuário cadastrado com sucesso!")
        return redirect(url_for("login"))
    return render_template("cadastro.html")

@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        matricula = request.form["matricula"]
        senha = request.form["senha"]

        for user in usuarios.values():
            if user.id == matricula and check_password_hash(user.senha_hash,
senha):
                login_user(user)
                flash("Login realizado com sucesso!")
                return redirect(url_for("area_restrita"))

        flash("Login inválido.")
        return redirect(url_for("login"))
    return render_template("login.html")

@app.route("/area_restrita")
@login_required
def area_restrita():
    return render_template("area_restrita.html", matricula=current_user.id,
email=current_user.email)

@app.route("/logout")
@login_required
def logout():
    logout_user()
    flash("Logout realizado com sucesso.")
    return redirect(url_for("login"))

if __name__ == "__main__":
    app.run(debug=True)

```

## Modelos

### modelos/base.html

```

<!DOCTYPE html>
<html lang="pt-br">
<head>

```

```

    <meta charset="UTF-8">
    <title>{% block title %}Sistema{% endblock %}</title>
</head>
<body>
    <nav>
        <a href="{{ url_for('login') }}">Login</a> |
        <a href="{{ url_for('cadastro') }}">Cadastro</a> |
        <a href="{{ url_for('area_restrita') }}">Área Restrita</a> |
        <a href="{{ url_for('logout') }}">Logout</a>
    </nav>
    <hr>
    {% with messages = get_flashed_messages() %}
    {% if messages %}
        <ul>
            {% for message in messages %}
                <li>{{ message }}</li>
            {% endfor %}
        </ul>
    {% endif %}
    {% endwith %}
    {% block content %}{% endblock %}
</body>
</html>

```

## modelos/cadastro.html

```

{% extends 'base.html' %}

{% block title %}Cadastro{% endblock %}

{% block content %}
<h2>Cadastro</h2>
<form method="post">
    Matrícula: <input type="text" name="matricula"><br><br>
    Email: <input type="email" name="email"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <button type="submit">Cadastrar</button>
</form>
{% endblock %}

```

## modelos/login.html

```

{% extends 'base.html' %}

{% block title %}Login{% endblock %}

```

```
{% block content %}
<h2>Login</h2>
<form method="post">
    Matrícula: <input type="text" name="matricula"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <button type="submit">Entrar</button>
</form>
{% endblock %}
```

---

## modelos/area\_restrita.html

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

{% block title %}Área Restrita{% endblock %}

{% block content %}
<h2>Área Restrita</h2>
<p>Bem-vindo(a), {{ matricula }} ({{ email }})!</p>
{% endblock %}
```

---



## Estrutura recomendada

```
/meu_app
├── app.py
├── templates/
│   ├── base.html
│   ├── cadastro.html
│   ├── login.html
│   └── area_restrita.html
```