✅ O que é o Flask-Login?

É uma extensão do Flask que **ajuda a gerenciar sessões de usuários**, ou seja: fazer login, logout, lembrar quem está logado, proteger rotas, etc.

🧠 Conceitos principais do Flask-Login:

LoginManager

Inicializa o controle de login:

python
CopiarEditar

```
login_manager = LoginManager()
login_manager.init_app(app)
```

1.

User Loader

Diz ao Flask como carregar o usuário a partir do ID salvo na sessão:

python
CopiarEditar

```
@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

2.

Classe de usuário

Sua classe User (geralmente vem de um modelo do banco de dados) precisa **herdar UserMixin**, que dá os métodos is_authenticated, is_active, get_id(), etc.

Exemplo:

python
CopiarEditar

```
from flask_login import UserMixin
```

```
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password = db.Column(db.String(120), nullable=False)
```

3.

Fazer login

Depois de verificar o usuário e a senha:

python
CopiarEditar

```
from flask_login import login_user

login_user(user)
```

4.

Proteger rotas

Para exigir login numa rota:

python
CopiarEditar

```
from flask_login import login_required

@app.route('/painel')
@login_required
def painel():
    return 'Você está logado!'
```

5.

Fazer logout

python
CopiarEditar

```
from flask_login import logout_user
```

```
@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('login'))
```

6.

Acessar o usuário logado

```
 python
CopiarEditar
from flask_login import current_user

@app.route('/perfil')
def perfil():
    return f'Olá, {current_user.username}'
```

7.

Se quiser, posso fazer:

- Um **resumo em formato de mapa mental**
- Uma **simulação de questão**
- Ou revisar contigo linha por linha de um código com Flask-Login

✅ Questão 1

Pergunta: Qual é o comando correto para manter o usuário logado?
Sua resposta: B) login_required(user)
✅ **Correta: C) login_user(user)**

🔍 **Explicação:**

- login_user(user) é a função que *loga* o usuário na sessão.

- login_required é um *decorator* usado para proteger rotas, não para fazer login.

✅ Questão 2

Pergunta: Como proteger a rota /painel?

Sua resposta: is_authenticated
✅ **Correta (esperada):** adicionar o decorator @login_required

👉 Então o código correto seria:

python
CopiarEditar

```
from flask_login import login_required

@app.route('/painel')
@login_required
def painel():
    return "Bem-vindo ao painel!"
```

🔍 **Explicação:**

- is_authenticated é usado dentro da rota, se você quiser fazer um if.

- Mas para proteger diretamente a rota, usa-se @login_required.

✅ Questão 3

Pergunta: O que esse load_user faz?

Sua resposta: D) Verificar se o usuário está autenticado
✅ **Correta: C) Carregar um usuário da sessão para o banco de dados**

🔍 **Explicação:**

- Quando o usuário está logado, o Flask salva seu user_id na sessão.

- O load_user pega esse user_id e consulta no banco de dados (User.query.get) para recuperar o objeto User.

Questão 1 — Respostas completas:

a) **( ) A função login_required serve para autenticar um usuário.**
➡️ **Falso** — Ela apenas protege rotas, não autentica.

b) **( ) A função logout_user() encerra a sessão do usuário atual.**
➡️ **Verdadeiro** — Ela faz logout do usuário.

c) **( ) O user_loader é necessário para o Flask saber como encontrar um usuário logado.**
➡️ **Verdadeiro** — O Flask-Login usa essa função para carregar o usuário a partir do id salvo na sessão.

d) **( ) current_user.is_authenticated retorna True se o usuário estiver logado.**
➡️ **Verdadeiro** — Essa propriedade indica se o usuário está autenticado.

Resumo Flask-Login — Parte 1

1. O que é Flask-Login?
É uma extensão do Flask que ajuda a gerenciar autenticação dos usuários: login, logout, manter sessão e proteger páginas que precisam que o usuário esteja logado.

2. Configuração inicial

- Crie um objeto LoginManager e inicialize com a aplicação Flask:

python
CopiarEditar

```
login_manager = LoginManager()
login_manager.init_app(app)
```

3. User Loader
Função que diz ao Flask como carregar o usuário a partir do ID salvo na sessão:

python
CopiarEditar

```
@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

Beleza! Aqui vai a continuação:

Resumo Flask-Login — Parte 2

4. Classe User
Sua classe de usuário deve herdar UserMixin para ter métodos essenciais como is_authenticated, get_id(), etc.

```
from flask_login import UserMixin

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True)
    password = db.Column(db.String(120))
```

5. Fazer login
Depois de verificar usuário e senha, use:

```
from flask_login import login_user

login_user(user)
```

6. Proteger rotas
Para exigir que o usuário esteja logado para acessar uma rota, use o decorator:

```
from flask_login import login_required

@app.route('/painel')
@login_required
def painel():
    return "Área protegida"
```

Quer a próxima parte?

Aqui vai a última parte do resumo, já já você estará fera no Flask-Login! 🔥

Resumo Flask-Login — Parte 3

7. Fazer logout
Para encerrar a sessão do usuário logado:

```
from flask_login import logout_user

@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('login'))
```

8. Acessar o usuário atual
Dentro das rotas, você pode acessar o usuário logado com:

```
from flask_login import current_user

print(current_user.username)
```

current_user.is_authenticated retorna True se o usuário está logado.

Funções principais:

Função / Decorator	O que faz
login_user(user)	Loga o usuário na sessão
logout_user()	Encerra a sessão do usuário
login_required	Protege rota para só permitir logados
user_loader (decorator)	Carrega usuário a partir do id
current_user	Acessa o usuário logado na sessão

Dica para lembrar na prova:

Pense no Flask-Login como um clube:

- **login_user** entrega o cartão de sócio
- **logout_user** recolhe o cartão
- **login_required** é a porta que só deixa passar quem tem cartão
- **user_loader** é quem confere na lista de sócios quem está com cartão válido