

Exercícios Práticos — Troca de Mensagens com RPC (Python)

Objetivo: Compreender como processos diferentes podem trocar mensagens usando RPC (Remote Procedure Call), no caso com XML-RPC em Python.

Exercício 1 — Subindo o servidor

- 1 Escolha um aluno para rodar o server.py (o servidor de mensagens).
- 2 Esse servidor deve ficar rodando para a turma inteira.
- 3 Confirme que o servidor responde:

```
import xmlrpc.client
proxy = xmlrpc.client.ServerProxy("http://127.0.0.1:9000")
print(proxy.ping()) # deve responder "pong"
```

Exercício 2 — Enviando e recebendo mensagens

- 1 Cada aluno escolhe um nome de usuário (ex.: 'joao', 'maria', 'gabriel').
- 2 Um aluno roda o receptor:

```
proxy = xmlrpc.client.ServerProxy("http://127.0.0.1:9000")
for i in range(5):
    msgs = proxy.get_messages("gabriel")
    for m in msgs:
        print(f"[de {m['from']}] {m['body']}")
```
- 1 Outro aluno envia:

```
proxy = xmlrpc.client.ServerProxy("http://127.0.0.1:9000")
proxy.send_message("maria", "gabriel", "Oi Gabriel, tudo bem?")
```

Exercício 3 — Conversa em dupla

- 1 Formem duplas.
- 2 Cada aluno abre um receptor com seu nome.
- 3 Cada um envia 3 mensagens para o colega.
- 4 Registrem as mensagens recebidas no caderno.

Exercício 4 — Grupo de bate-papo

- 1 Formem grupos de 3 ou 4 alunos.
- 2 Todos configuram o receptor com seus nomes.
- 3 Escolham um moderador, que manda a primeira mensagem.
- 4 Cada aluno responde ao moderador.
- 5 Observem que cada mensagem vai para a 'caixa' individual.

Exercício 5 (Desafio) — Mural coletivo

- 1 Modifiquem o servidor para aceitar um destinatário especial 'all'.
 - 2 Nesse caso, a mensagem é enviada para todos os usuários conectados.
- ```
def send_message(sender, recipient, body):
 if recipient == "all":
```

```
 for user in state["messages"]:
 state["messages"][user].append({"from": sender, "body": body})
 else:
 box = state["messages"].setdefault(recipient, [])
 box.append({"from": sender, "body": body})
 return True
```