

# Laboratório 4 (parte 2)

**Breno Rosa Tomé - 113026430**

## Atividade 1 - Implementação

### Funcionamento

Para executar o chat é necessário iniciar o server, depois quantos clientes quiser

```
$ python3 server.py  
$ python3 client.py
```

Ao iniciar o client você deverá escolher o nome de usuário, após escolher será apresentado aos comandos disponíveis

```
> list : lista usuários ativos  
> active : marca o usuário como ativo  
> inactive : marca o usuário como inativo  
> pm user_name message : envia mensagem para usuário  
> fim : saí da aplicação
```

### list

Exibe os usuários ativos, é permitido mesmo sem estar ativo

### active

Fica ativo, permitindo receber e enviar mensagens

### inactive

Fica inativo, não poderá enviar nem receber mensagens

### pm

Envia uma mensagem direta a outro usuário ativo ou a si mesmo

**fim**

Encerra aplicação

## Atividade 2 - Testes

Como o chat não tem suporte a conversas em grupo, os testes relacionados a comportamento em grupo não foram feitos

Todos os demais funcionaram de acordo com o esperado

```
breno@pop-os:~/Cloud/megaSync/fundao/_dist/sistemas-distribuidos/lab4_2/client$ python3 client.py
Escolha um nome de usuário:Breno

    Bem vindo Breno, comandos disponiveis:

    > list : lista usuários ativos
    > active : marca o usuário como ativo
    > inactive : marca o usuário como inativo
    > pm user_name message : envia mensagem para usuário
    > fim : sai da aplicação

list
Não tem ninguém aqui :/
list
User1
user2
active
Pronto para receber mensagens
pm user2 oi
pm User1 oi
de User1:
Olá
de user2:
Como vai ?
oqejwcekqpwoke
Comando não encontrado
pm nome_errado mensagem
Destinatário não encontrado
inactive
Você não receberá mais mensagens, para reativar o recebimento use o comando 'active'
pm user2 oi
Você deve estar ativo para enviar mensagens
```

## Atividade 3 - Relatório

### Adaptações

Eu removi as funcionalidades de grupo que no projeto chamei de canais e movi a validação para dentro dos handlers

# Observações

Subestimei um pouco o prazo para a funcionalidade de grupos, não pensei em alguns requisitos como tratar a saída do dono do grupo, permissões de remover ou adicionar usuários, etc. os requisitos não são exatamente complicados de implementar, mas como só encontrei a necessidade dessas funcionalidades durante a implementação acabei não completando no prazo e preferi remover a funcionalidade.

# Perguntas

Aqui respondo as perguntas feitas sobre a parte 1

**enviar mensagem para usuário ou canal significa a mesma coisa? Ou serão duas opções distintas de comunicação?**

Os canais são equivalentes a grupos, originalmente a ideia era que se você enviar uma mensagem a um canal, todos usuários inscritos no canal receberiam

**Criar um canal será a operação básica para enviar mensagem para uma usuário? Uma canal envolverá dois usuários ou mais? Inscrever-se em um canal será a operação básica para receber mensagens de um usuário?**

O canal só seria necessário para comunicação em grupos, toda mensagem recebida por um canal seria repassada para todos inscritos nele.

**qual estilo arquitetural será usado? os canais ficarão no servidor?**

O estilo escolhido foi o em camadas, e os canais ficariam no servidor

**Por que optou-se pela arquitetura cli/ser?**

Escolhi a arquitetura cli/ser pela simplicidade, acredito que foi mais fácil de implementar que um sistema p2p

**Haverá uma conexão apenas entre o cliente e o servidor por onde passarão todas as trocas de mensagens daquele usuário?**

Sim, cada cliente faz uma conexão com o servidor por onde envia e recebe as mensagens