



Universidade Federal de Viçosa - Campus Florestal

Trabalho Prático III CarnaTroca

Professor: Thais Regina de Moura Braga Silva

Disciplina: Sistemas Distribuídos e Paralelos (CCF 355)

Arthur Marciano - 3019
Vinicius Julio - 3495

**Florestal, MG
2022**

ÍNDICE

1. Introdução	3
2. Desenvolvimento	3
Comunicação via Socket	3
Threads	4
Banco de dados	4
operacoes_server.py	5
Exemplos de utilização do sistema	
4. Conclusão	5

1. Introdução

O objetivo deste trabalho é a implementação do sistema de escambo CarnaTroca, utilizando sockets como sistema de comunicação entre o cliente e o servidor.

O sistema, desenvolvido em Python e com persistência em SQLite, permite que o usuário cadastre uma conta, crie anúncios de fantasias de carnaval, visualize anúncios de outros usuários podendo fazer propostas de trocas, e também visualizar e aceitar ou recusar propostas para as suas fantasias. Na versão atual o sistema possui interface de linha de comando, com destaques de texto colorido, com o intuito de tentar deixar mais amigável ao usuário, no entanto, futuras implementações possuirão interface gráfica. Nos arquivos da entrega há um README.md contendo instruções para executar o sistema e login de alguns usuários pré-cadastrados.

2. Desenvolvimento

Comunicação via Socket

No exemplo mostrado na imagem abaixo, o que esse código faz é enviar uma solicitação POST para a URL relacionada à função cadastraUser. Estamos usando a função .post() aqui, mas a Requests permite que você utilize outras funções, como .get() e .put() para também enviar essas solicitações.

As mensagens são trocadas de forma simples. Sempre que um cliente necessita executar algo, ele envia uma mensagem codificada *JSON* que é decodificada pelo servidor, e a partir do conteúdo desta mensagem ele executa a tarefa necessária e envia uma resposta de *erro* ou de *aprovado* de volta para o cliente.

Toda mensagem enviada pelo módulo cliente possui um control como mostrado na figura abaixo, este control serve para que o servidor saiba qual ação ele deve executar. Além do control, também são passadas outras informações para o servidor. Essa mensagem é enviada em bytes, sendo enviada pelos buffers da rede invocados pelos métodos recv e send, para ser possível traduzir um string para o formato binário de envio e vice-versa, utiliza-se o método encode(), no envio, e decode(), no recebimento, em todas as mensagens que passam pelo canal de comunicação.

```
user = {
    'control': ('1'),
    'nome': input('\nEscolha seu nome de usuario:'),
    'senha': input('Escolha uma senha para sua conta:')
}

envio = json.dumps(user)
# envia a mensagem para o servidor
client.sendall(envio.encode('utf-8'))
data = client.recv(1024)
```

Figura 1: Envio de mensagens pelo cliente

Threads

```
def threads():
    while True:
        try:
            client, ip = server.accept() # server.accept (método para aceitar a conexão)
            thread1 = threading.Thread(target=novo_cliente, args=[client, ip])
            thread1.start()
        except KeyboardInterrupt:
            print(f'Desligando o servidor!')
            server.close()

threads()
```

Figura 2: Threads

Sempre que um novo cliente é iniciado no servidor, uma nova thread é iniciada. Logo, o sistema suporta vários clientes conectados ao mesmo tempo no servidor.

Banco de dados

Para o banco de dados, foi utilizado o SQLite3 como forma de armazenar as informações utilizadas na aplicação.

Para que o sistema funcionasse corretamente foi necessário a criação de 4 tabelas, sendo elas:

- **user** → Tabela usada para armazenar os dados dos usuários. Neste caso possui apenas duas colunas:
 - *user_name* → Contém os nomes de usuário (não podem se repetir)
 - *senha* → Contém a senha para que os usuários façam login no sistema.
- **fantasias** → Tabela que armazena todos os dados das fantasias:
 - *ID* → Cada fantasia tem um ID único;
 - *user_userName* → Nome de usuário do dono da fantasia;
 - *nome_fantasia*;
 - *descricao*;
 - *tamanho*;
 - *disponivel* → 1 se a fantasia está disponível para troca e 0 se não;
- **trocas_pendentes** → Tabela contendo as informações das propostas de trocas:
 - *ID_troca* → Cada proposta possui um ID único;
 - *ID_anunciante* → ID da fantasia do anunciante;
 - *ID_proponente* → ID da fantasia do proponente;

- *nome_proponente*;
- **controle_trocas** → Tabela que contém as informações de todas as trocas.
 - *ID_troca* → Cada troca possui um ID, que após a troca ser realizada (ou não) é passado para tabela de controle;
 - *ID_fantasia_anunciante* → ID da fantasia que participou da troca;
 - *ID_fantasia_proponente* → ID da outra fantasia que participou da troca;
 - *resultado_troca* → 1 se a troca foi aceita, 0 se foi recusada;

operacoes_server.py

Comentaremos em especial sobre esse arquivo uma vez que os outros contêm principalmente a interface do usuário, enquanto esse contém a implementação das operações que o usuário pode realizar, sendo elas:

- **cadastra_user** → Verifica se o usuário já existe no banco de dados, se não, realiza o cadastro;
- **faz_login** → Verifica se as credenciais inseridas existem no banco de dados e caso sim, efetua o login;
- **cria_anuncio** → Cadastra a fantasia no banco de dados e a coloca como disponível;
- **lista_fantasias** → Lista todas as fantasias disponíveis, exceto as do usuário;
- **lista_meus_anuncios** → Lista todas as fantasias cadastradas do usuário;
- **propor_troca** → Usuário deve informar o ID das fantasias que deseja trocar, a proposta é cadastrada na tabela *trocas_pendentes*;
- **monitorar_trocas** → Mostra para o usuário as ofertas de troca das fantasias que ele possui, sendo possível visualizar as informações de ambas as fantasias, o nome do proponente e o ID da troca;
- **aceita_trocas** → Aceita a troca informada por meio da variável *ID_troca*, as fantasias mudam de usuário no banco de dados, seu campo “disponível” muda de 1 para 0, a proposta é removida da tabela *trocas_pendentes*;
- **recusa_trocas** → Recusa a troca informada em *ID_troca*, a proposta é removida de *trocas_pendentes*;

Exemplos de utilização do sistema

```
-----  
Bem vindo ao CarnaTroca!!  
-----
```

```
1: Fazer Cadastro
```

```
2: Fazer Login
```

```
3: Sair
```

```
Escolha uma das opcoes acima para continuar:█
```

Figura 3: Tela de login do sistema

```
Fazer o cadastro
```

```
Escolha seu nome de usuario:vinicio
```

```
Escolha uma senha para sua conta:cachorro123█
```

Figura 4: Efetuando o cadastro

```
-----  
Olá, você está logado no sistema CarnaTroca.  
-----
```

```
1: Anunciar itens para troca
```

```
2: Listar todos os itens disponiveis para troca
```

```
3: Ver minhas fantasias
```

```
4: Sair do sistema!
```

```
Cadastro realizado com sucesso!!
```

```
Escolha uma das opcoes acima para continuar:█
```

Figura 5: Tela inicial do sistema após login

```
-----  
Olá, você está logado no sistema CarnaTroca.  
-----
```

```
1: Anunciar itens para troca
```

```
2: Listar todos os itens disponiveis para troca
```

```
3: Ver minhas fantasias
```

```
4: Sair do sistema!
```

```
Cadastro realizado com sucesso!!
```

```
Escolha uma das opcoes acima para continuar:1
```

```
Anunciar fantasias para troca!
```

```
Nome da fantasia (obrigatório):edward maos de tesoura
```

```
Descrição (opcional):fiel ao filme, usada 2 vezes
```

```
Tamanho da fantasia (obrigatório):M█
```

Figura 6: Anunciando fantasia

```

Lista de fantasias disponiveis para troca:
-----
ID: 1
Nome do anunciante: Regis
Nome da fantasia: fantasia do shrek
Descrição: fantasia em otimo estado, novinha
Tamanho: G
Disponivel: 1
-----
ID: 8
Nome do anunciante: Regis
Nome da fantasia: Menina super poderosa
Descrição: muito confortavel
Tamanho: M
Disponivel: 1
-----
ID: 9
Nome do anunciante: marciano
Nome da fantasia: Uber
Descrição: Novinha, usada uma vez
Tamanho: G
Disponivel: 1
-----

Escolha uma opção para continuar.
-----
1: Propor troca.
2: Listar as propostas de troca
3: Voltar à tela do sistema
4: Sair
Escolha uma das opcoes acima para continuar:

```

Figura 7: Visualizando todos os anúncios disponíveis

```

Escolha uma opção para continuar.
-----
1: Propor troca.
2: Listar as propostas de troca
3: Voltar à tela do sistema
4: Sair
Escolha uma das opcoes acima para continuar:1
PROPOR TROCA!

ID da fantasia que você deseja:9
ID da fantasia que você deseja propor em troca: 10

```

Figura 8: Propondo a troca

```

Listando suas propostas de troca:
Minha fantasia:
ID: 9
Nome do anunciante: marciano
Nome da fantasia: Uber
Descrição: Novinha, usada uma vez
Tamanho: G
ID_Troca: 8
Fantasia proposta em troca:
ID: 10
Nome do anunciante: vinicio
Nome da fantasia: edward maos de tesoura
Descrição: fiel ao filme, usada 2 vezes
Tamanho: M
ID_Troca: 8
-----
Deseja aceitar ou recusar a(s) troca(s)?
1 - Aceitar
2 - Recusar
Escolha uma das opcoes acima para continuar:

```

Figura 9: Usuário visualizando a proposta de troca recebida

```

Deseja aceitar ou recusar a(s) troca(s)?
1 - Aceitar
2 - Recusar
Escolha uma das opcoes acima para continuar:1

Informe o ID da troca que deseja aceitar: 8

```

Figura 10: Aceitando a proposta de troca

vinicio@vinicio-VirtualBox: ~/Dow...	vinicio@vinicio-VirtualBox: ~/Dow...
<pre> Minhas fantasias: ----- ID: 10 Nome do anunciante: marciano Nome da fantasia: edward maos de tesoura Descrição: fiel ao filme, usada 2 vezes Tamanho: M Disponível: 0 ----- Escolha uma opção para continuar. ----- 1: Propor troca. 2: Listar as propostas de troca 3: Voltar à tela do sistema 4: Sair Escolha uma das opcoes acima para continuar: </pre>	<pre> Minhas fantasias: ----- ID: 9 Nome do anunciante: vinicio Nome da fantasia: Uber Descrição: Novinha, usada uma vez Tamanho: G Disponível: 0 ----- Escolha uma opção para continuar. ----- 1: Propor troca. 2: Listar as propostas de troca 3: Voltar à tela do sistema 4: Sair Escolha uma das opcoes acima para continuar: </pre>

Figura 11: Visualização das fantasias pós-troca

4. Conclusão

A realização do trabalho serviu como uma forma de mostrar realmente como é feita a comunicação via Sockets. A parte da comunicação em si se mostrou um pouco complicada em alguns momentos, como por exemplo para enviar o conteúdo do banco de dados do servidor para o cliente, mas com algumas pesquisas essas complicações foram superadas. A verdadeira dificuldade esteve na implementação da interface gráfica, e por esse motivo não foi realizada.

Portanto pode-se concluir que o trabalho foi de grande proveito, já que o grupo conseguiu ter uma experiência prática com a implementação da comunicação com Sockets, servindo como uma revisão e fixação do conteúdo visto previamente em sala de aula.