



Universidade Federal de Viçosa – *Campus* Florestal
CCF 355 - Sistemas Distribuídos

Trabalho Prático III
Álbum de Figurinhas

Professor: : Thais Regina de Moura Braga Silva

Guilherme Aguiar Milanez - 3496

Lucas Cunha Barbosa - 3493

Florestal, MG

2020

1. Introdução

O objetivo deste trabalho consiste na implementação do programa do álbum de figurinha implementado com suporte de comunicação via socket. Para tal, foi implementado, um banco de dados em MySQL para suportar todos os dados do sistema, um módulo para o servidor contendo a conexão com o banco de dados e uma API integrada capaz de traduzir comandos do cliente para informações presentes no banco de dados, um módulo de cliente operado em thread para envio e recebimento de mensagens via socket e manutenção de operações de um colecionador.

As seções a seguir descrevem alguns detalhes acerca das operações citadas acima, enfatizando escolhas de projeto, limitações e recursos.

1) Tecnologias Utilizadas

- a) MySQL
- b) Python
- c) socket (Python biblioteca)
- d) threading (Python biblioteca)
- e) os (Python biblioteca)
- f) subprocess (Python biblioteca)
- g) multiprocessing (Python biblioteca)
- h) queue (Python biblioteca)
- i) signal (Python biblioteca)
- j) MySQLdb (Python biblioteca)
- k) pyngrok (Python biblioteca)
- l) json (Python biblioteca)
- m) sys (Python biblioteca)

2. Comunicação via Socket

A comunicação via socket no projeto é feita por dois módulos, um módulo cliente e um módulo servidor. Para ser realizada é necessário duas informações principais: um endereço IP e um número de porta. Para os teste, usa-se como base os seguintes parâmetros:

ip: '127.0.0.1'

porta: 90

Sempre que um cliente é instanciado, inicia-se uma thread em que é a invocação da função do menu de uso do programa. Para cada tarefa seguinte executada pelo cliente uma mensagem é enviada para o servidor que irá processá-la da maneira correta.

As mensagens, nas implementações de socket são construídas pelo desenvolvedor, para isso utiliza uma cadeia de caracteres de base que será encaminhada para o outro elemento conectado. Essa cadeia de caracteres é dada 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 strings que viajaram pelos canais de comunicação.

3) Banco de Dados

Utilizou-se um banco de dados relacional (MySQL) para armazenar todas as informações usadas no sistema. Tabelas construídas:

- 1) Collector ⇒ Tabela usada para armazenar os dados referentes aos colecionadores. Tendo os seguintes atributos:
 - a) id
 - b) nome
 - c) email
 - d) password
 - e) coins
 - f) IdAlbum
- 2) Inventory_cards ⇒ Tabela usada para armazenar os dados referentes ao inventário dos colecionadores. Tendo os seguintes atributos:
 - a) id
 - b) idcard

- c) quantity
 - d) idCollector
- 3) Card ⇒ Tabela usada para armazenar os dados referentes às cartas. Tendo os seguintes atributos:
- a) id
 - b) name
 - c) description
 - d) picture
- 4) Album ⇒ Tabela usada para armazenar os dados referentes ao álbum do colecionador. Tendo os seguintes atributos:
- a) id
 - b) QuantityCards
 - c) QuantityStickeredCards
- 5) Collection_card ⇒ Tabela usada para armazenar as cartas que estão relacionadas ao álbum do jogador. Ou seja, armazenar as cartas coladas. Tendo os seguintes atributos:
- a) id
 - b) idAlbum
 - c) idCard
- 6) Store_cards ⇒ Tabela usada para armazenar os dados referentes às cartas compradas na loja. Tendo os seguintes atributos:
- a) id
 - b) price
 - c) name
 - d) category
 - e) icCollector
 - f) idCards
- 7) Exchange ⇒ Tabela usada para armazenar os dados referentes às trocas de cartas entre os usuários. Tendo os seguintes atributos:
- a) id
 - b) IdCollectorOwner
 - c) IdCollecotorTarget
 - d) idCard
 - e) idCardReceived

4) Operações Disponíveis

Um usuário pode logar ou solicitar um cadastro no programa como primeira ação dentro do sistema. Após ser autenticado, em sua tela, será exibido um menu com todas as opções de ações disponíveis no programa. Tais como, visualizar perfil, visualizar o álbum, visualizar o inventário ou realizar alguma compra de figurinhas.

Cada uma das operações recebe como passagem de parâmetro o cliente já em execução em uma das threads. Ao ser acionada a funcionalidade envia uma mensagem para o servidor como primeira requisição. Essa mensagem é utilizada para que seja possível identificar no servidor qual informação será coletada do banco de dados e posteriormente retornada para a interface com o usuário.

Cada uma das operações implementadas pela API, realiza consultas no banco de dados retornando uma lista de dicionário com as coletas feitas no banco de dados. Essa lista então é novamente decodificada na interface do usuário para que, depois de ser tratada, o usuário possa analisar informações relevantes de acordo com a operação solicitada.

5) Recompensas

O usuário só pode comprar novas figurinhas com dinheiro em sua conta. Pensando nisso, projetou-se um sistema capaz de combinar a necessidade da movimentação de dinheiro e a usabilidade. O sistema utiliza um esquema de gamificação para recompensar o usuário da seguinte maneira: quanto mais tempo logado o usuário obtiver mais recompensas, em dinheiro, o mesmo terá. No banco de dados o usuário terá anotado o tempo de logIn e o tempo de logOut.

6) Troca

Para o sistema de troca há uma tabela especial no banco de dados usada para manter informações sobre as trocas em trânsito, negadas e aceitas. Foi implementado o seguinte esquema de troca:

- 1) Usuário solicita a troca ;
- 2) É solicitado ao usuário que escolha uma das seguintes opções:

- a) Iniciar nova troca
 - b) Monitorar troca
- 3) Para iniciar uma nova troca:
- a) Usuário escolhe o id da carta que deseja disponibilizar para a transação e escolhe o id da carta que deseja receber;
 - b) Usuário seleciona o id do usuário que deseja realizar a troca
 - c) A troca é criada
 - d) Basta que o usuário solicitado aprove ou desaprove a troca
- 4) Para monitorar uma nova troca:
- a) O sistema exibe os id de todas as trocas ativas
 - b) O usuário seleciona um dos ids
 - c) O usuário terá duas opções
 - i) Caso o usuário seja o solicitante, poderá tanto cancelar a proposta de troca quanto manter.
 - ii) Caso o usuário seja o solicitado, poderá tanto aprovar quanto negar a troca.

A cada passo, mensagens são trocadas entre os processos clientes e o processo servidor, todas as opções selecionadas durante a troca são enviadas para o servidor pois, majoritariamente, cada passo realiza um tipo de consulta diferente no banco de dados. Ao final o usuário terá ciência do estado final de sua solicitação.