

FUZZY.ME

Autores: Israel Efraim de Oliveira, José Carlos Zancanaro.

O Fuzzy.Me é uma rede social simples dedicada a postagem de textos junto à imagens. O nome da rede era porque pretendíamos mexer na imagem e postar uma versão alterada dela (algoritmos genéticos)... fuzzy.

O projeto apresenta implementação de relação Cliente/Servidor (ServerSocket/Socket), com comunicação feita através de JSON, interface gráfica em JavaFX, persistência de dados com o JDBC utilizando o MariaDB (no Arch Linux), uso de threads e padrões de projeto.

De modo geral, o trabalho foi dividido em 4 projetos: core, que contém definições comuns a todos os projetos; ClienteFuzzy, que abstrai o processo de comunicação com o servidor, implementando as operações necessárias; ServidorFuzzy, que escuta e atende sockets e cria um gerenciador de conexão para cada um; FuzzyMeFX, que contém a implementação disto com uma interface gráfica. Em suma, é um grande projeto.

Com relação aos requerimentos do projeto, as threads foram implementadas para possibilitar que o Servidor atenda a mais de um cliente e utilizadas na UI para trabalhar sem ocupar a Thread principal do JavaFX, salvo em casos onde há necessidade de executar código na thread dele utilizando o Platform.runLater(). Os listeners, como requisitos de padrão de projeto, foram implementados no Servidor, para permitir saber o que está acontecendo na execução e, para o mesmo propósito, no socket de cliente. O projeto do Servidor utiliza estes para informar no console ações de recepção, finalização e exceção de sockets e a interface gráfica e cliente utilizam para saber se uma operação foi realizada com sucesso. Foram implementados “adapters” para cada um dos listeners, para possibilitar ao programador escolher implementar menos métodos da interface. Há a existência de uma instância singleton no projeto também, para gerenciamento do banco de dados.

Os protocolos de comunicação via Json seriam também fábricas simples e abstratas, mas acabamos optando por remover os outros tipos de comunicação e deixando somente um protocolo padrão. Os serializadores e deserializadores são instanciados através da classe Protocolo, que possui dois métodos estáticos.

