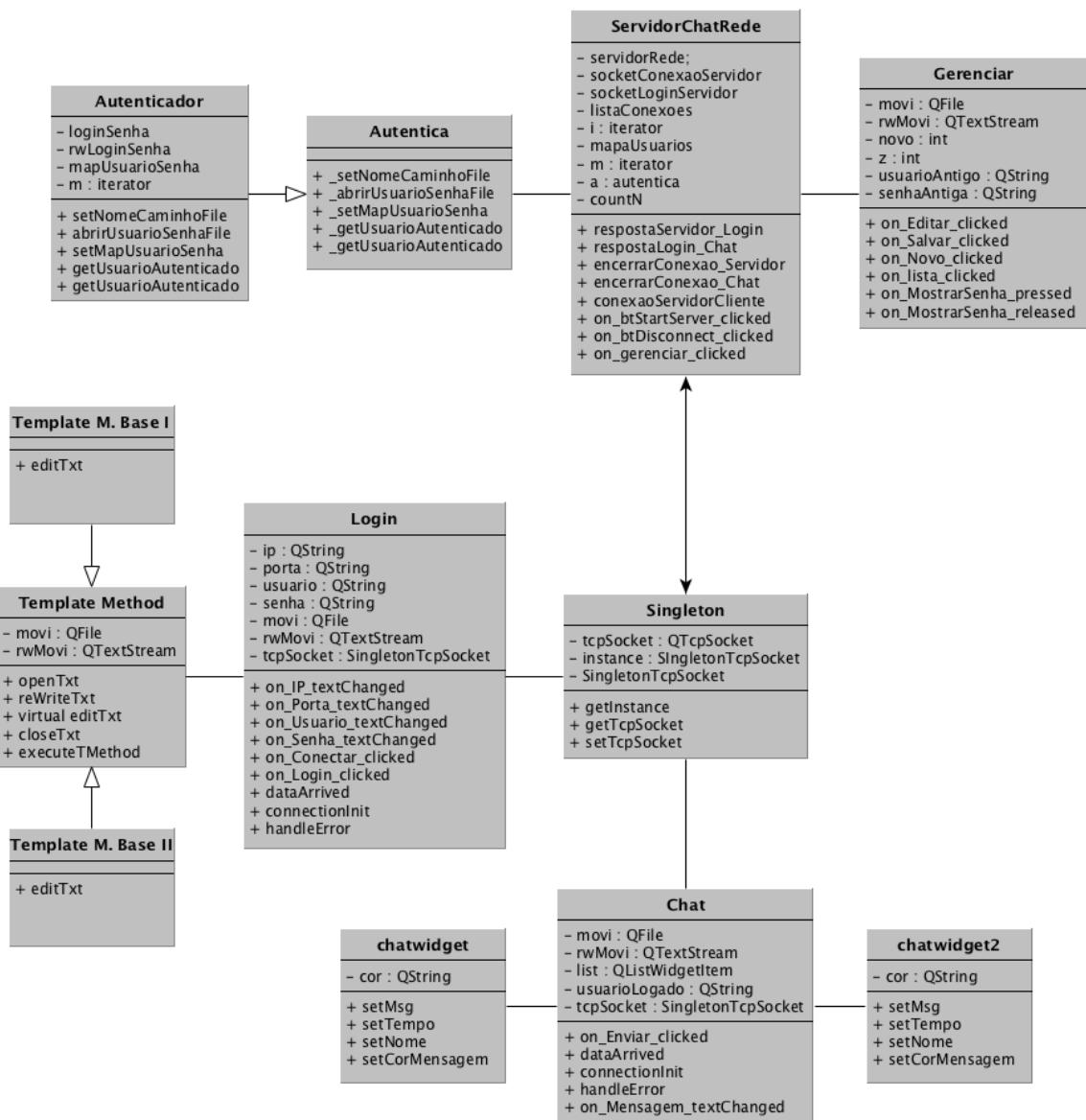


Documentação – Trabalho do Grau B

Integrantes: Eduardo Forneck, Guilherme Gotardo, Iago Carraro dos Santos

Descrição geral: Software que permite comunicação através de rede local por protocolo TCP, é composto por uma aplicação servidor e múltiplas instâncias de uma aplicação cliente, a comunicação se dá como um chat público, as mensagens de todos podem ser vistas por todos, apesar de publico, para poder enviar mensagem é necessário realizar o *login* com usuário e senha válidos, estes são gerenciados pelo administrador do sistema, que têm controle sobre os usuários através da aplicação servidora, que possibilita a adição de novos usuário e a edição de usuário e senha.



UML geral do software descrito

Descrição das Classes da Aplicação Cliente:

Login: Permite ao usuário realizar login, armazena informações de conexão ao servidor em um arquivo de texto para quando a aplicação for executada as informações de conexão ao servidor (da ultima conexão) já estarão preenchidas nos campos.
Se o Login for executado com sucesso, irá chamar a classe Chat.

Chat: Se conecta ao servidor e permite enviar e receber mensagens, ao receber mensagens as exibe juntamente com o usuário que as enviou.

Chatwidget: Classe que exibe um widget gráfico do tipo QListWidgetItem para ser exibido como um objeto de mensagem na classe Chat em um objeto QListWidgetItem, este objeto possui layout e configurações para exibir as mensagens enviadas pelo usuário *logado* na instância do sistema.

Chatwidget2: Possui função similar a Chatwidget, porém possui layout e configurações diferenciadas para exibição das mensagens recebidas de outros usuários do sistema.

Singleton: Classe responsável por instanciar apenas um objeto do tipo QTcpSocket, que é utilizado para conexão com o servidor pela classe Chat e Login. Implementa o padrão Singleton.

Template Method: Esta classe é responsável pela leitura e escrita de um arquivo de texto.

Ela consiste em 4 métodos de uso comum e um método virtual que cada classe base reimplementará:

openTxt: abre um arquivo de texto e retorna 1 se foi aberto corretamente e 0 se não foi aberto corretamente.

reWriteTxt: reescreve um arquivo.

editTxt: classe que será reimplementada pelas classes bases

closeTxt: fecha o arquivo de texto

executeTMethod: método para executar a sequencia do programa

Template Method Base I: Esta classe reimplementa o método *editTxt* de modo que ela retorne (através de um ponteiro de um objeto do tipo QString) o endereço IP e a porta do arquivo de configurações

Template Method Base II: Esta classe reimplementa o método *editTxt* de modo que ela escreva o IP e a porta em um arquivo (o parâmetro passado é um ponteiro de um objeto do tipo QString).

Descrição das Classes da Aplicação Servidor:

ServidorChatRede: Realiza comunicação com as aplicações clientes (com a classe Chat), recebe as mensagens enviadas e as redireciona para os outros usuários, enviando junto com a mensagem o *username* do usuário que a enviou, para que este possa ser exibido pela classe Chat.

Autentica: Esta classe implementa um Adaptador para a classe autenticador, a classe ServidorChatRede se comunica com a classe Autenticador através da Autentica.

Autenticador: Realiza comunicação com as aplicações clientes no momento de login (com a classe Login), realiza autenticação no momento de login, retornando a classe login se o usuário ou senha estão corretos, para que assim a classe Login possa chamar a classe Chat em caso positivo. Realiza a autenticação com base em um arquivo local de usuários e senhas. Esta classe implementa o padrão Façade.

Gerenciar: Permite a adição de um novo usuário e a edição de usuário e senha dos usuários do sistema, esta classe é localizada na aplicação servidora para que possa ser usada pelos administradores do sistema.

Utilização de Padrões de Projeto:

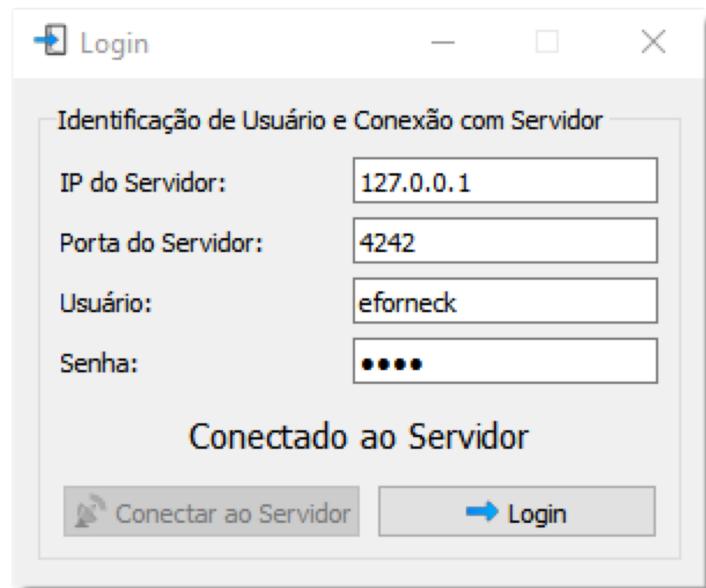
Façade: É utilizado na classe Autenticador, esta classe realiza o processo de autenticação do usuário no momento de login, criando uma interface para a classe ServidorChatRede

Singleton: É utilizado para instanciar apenas um objeto do tipo QTcpSocket na aplicação cliente, assim a classe Login e a classe Chat utilizam o mesmo socket para conexão com o servidor.

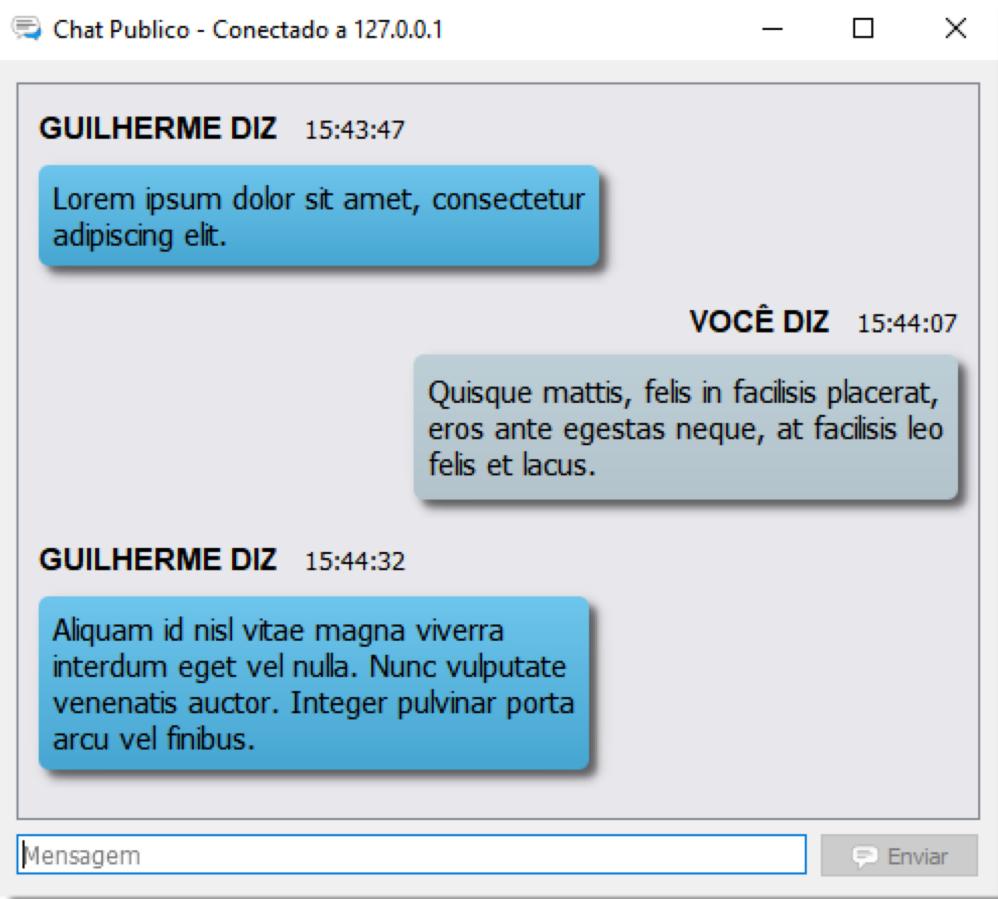
Template Method: Tem a função de executar um código sequencial sendo que nessa sequencia, há códigos comum para os objetos e códigos que se diferem.

Adapter: É utilizado para adaptar a classe Autenticador para uma versão antiga da classe cuja interface é incompatível com os métodos atuais.

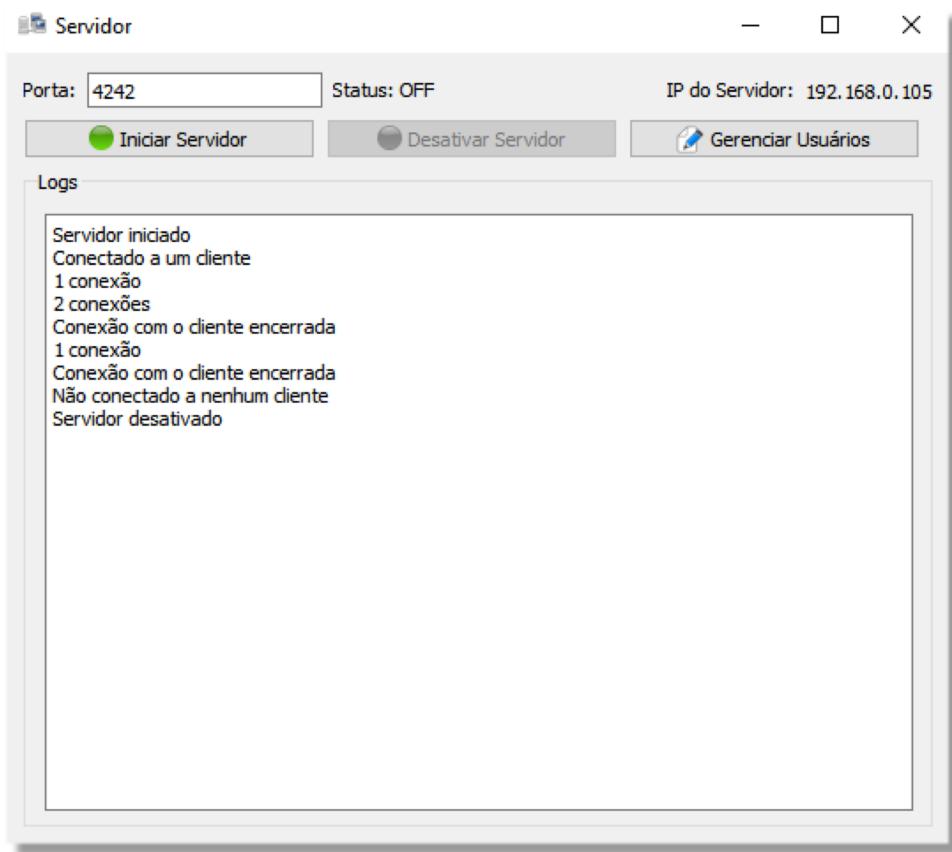
Screenshots:



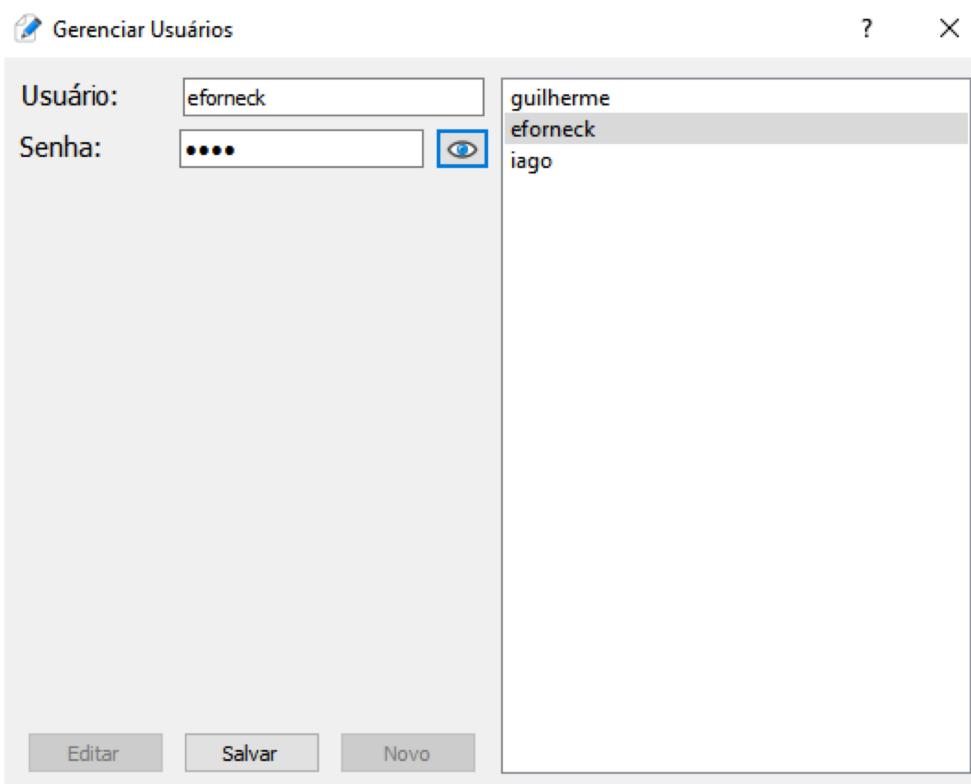
Tela de Login



Tela do Chat



Tela do Servidor



Tela de Gerenciamento de Usuários