U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Internet Relay Chat

ChaTime

1º Projeto de TDIN 2019/2020

Regente: António Miguel Pontes Pimenta Monteiro

Mestrado Integrado em Engenharia Informática e Computação

João Pedro Viveiros Franco - up201604828

Tomás Nuno Fernandes Novo - up201604503

Índice

1 – Introdução

No âmbito da unidade curricular Tecnologias de Distribuição e Integração, foi-nos proposto como 1ª trabalho prático o desenvolvimento de um sistema de ***Internet Relay Chat*** em **C#** baseado em ***.NET Remoting*****.** Para tal, usufruímos do ***Visual Studio*****,** que nos permitiu a criação de uma interface gráfica **(GUI)** que facilita a interação dos utilizadores, sendo que esta foi criada com ***.NET Windows Forms***.

Para utilizar a aplicação que desenvolvemos, cada utilizador precisa de se **registar** no servidor que é comum a todos os utilizadores, sendo que é necessário fornecer alguma informação, nomeadamente o seu ***nickname,*** **nome real** e ***password.*** Posteriormente, o *user* necessita de efetuar ***login*** para poder desfrutar da aplicação. Com o fim de proteger a informação do utilizador, antes de ser guardada na base de dados, a *password* é sujeita a uma função de **hash**: **SHA256**.

Cada utilizador usa uma instância da aplicação ***client****.*

2 – Arquitetura

2.1 – *Cliente*

Este módulo inclui não só as interfaces nas quais os utilizadores se podem registar ou autenticar, como também a janela principal na qual é possível visualizar todos os utilizadores (*online* e *offline*) e a janela na qual acontece a troca de mensagens. Uma instância deste módulo contém um objeto da classe ***Client*** que fornece uma interface simples e intuitiva para utilizar o protocolo de conversação. Contém *delegates* para eventos como ***OnlineUsersChanged***, ***ChatAsked***, ***ChatFinalized*** e ***MessageReceived*** de maneira a que a *UI* possa receber e tratar destes acontecimentos.

```
//---------------------------------Local methods---------------------------------

public bool Connect(string ServerIP, int port, string endpoint)...

public string Register(string username, string password, string RealName)...

public string Login(string username, string password)...

public bool Logout()...

public void Ping()...

public List<string> GetUsers()...

public int StartChat(string username)...

public int StartChat(List<string> usernames)...

public void AcceptChatRequest(int roomId)...

public void RejectChatRequest(int roomId)...

public bool SendMessage(int roomId, string message)...

public bool SendFile(int roomId, string message, byte[] file)...
```

Fig. 1 – Client methods

```
public delegate void OnlineUsersChangeEventHandler(object source, OnlineUsersEventArgs e);
public event OnlineUsersChangeEventHandler OnlineUsersChanged;

public delegate void AskForChatEventHandler(object source, AskForChatEventArgs e);
public event AskForChatEventHandler ChatAsked;

public delegate void ChatFinalizedEventHandler(object source, ChatFinalizedEventArgs e);
public event ChatFinalizedEventHandler ChatFinalized;

public delegate void MessageReceivedEventHandler(object source, MessageReceivedEventArgs e);
public event MessageReceivedEventHandler MessageReceived;
```

Fig. 2 – Client events

2.2 – *Server*

Por sua vez, o módulo do servidor estabelece a comunicação com a base de dados, sendo responsável por efetivar a autenticação e registo dos utilizadores. Assim, quando um utilizador se autentica ou desconecta, o servidor é o responsável por notificar os clientes desta mudança de estado, objetivando que o módulo ***client*** atualize a interface (evento ***OnlineUsersChanged***).

Também é responsabilidade do servidor controlar a criação das conversações, notificando os clientes da proposta de conversação (evento ***ChatAsked***) bem como registar as respostas e notificar os *users* do "resultado" deste pedido (evento ***ChatFinalized***).

É apenas mantido um servidor de .NET Remoting (no servidor do chat) que disponibiliza um objeto da classe ***Server*** (*singleton*). Os objetos da classe ***Client*** conectam-se por ***.NET Remoting*** ao servidor e chamam várias funções do servidor. Este objeto mantém cópias dos objetos ***Client*** dos outros utilizadores. Deste modo, o envio e a receção das mensagens é feita chamando um método remoto no próprio objeto do ***Client***.

```
public bool SendMessage(int roomId, string message)
{
    try
    {
        List<Client> clients = chatRooms[roomId].clients;

        for (int i = 0; i < clients.Count; i++)
        {
            clients[i].OnMessageSend(roomId, this.UserName, message);
        }
    }
    catch(Exception e)
    {
        Console.WriteLine(e.Message);
        return false;
    }

    return true;
}
```

Fig. 3 – Send message method of the client

2.3 – Base de Dados

Dada a necessidade de guardar informação relativa a cada utilizador (***nickname***, **nome real** e ***password***) e aos logs das conversações, foi criada e integrada no projeto uma **Base de Dados MongoDB.**

O código desenvolvido relativo à DB encontra-se no ficheiro **Database.cs**.

Para correr a *database*, foi criada a *script* **openDB.bat**, sendo necessário corrê-la antes de iniciar o servidor.

2.4 – *Remote Objects, Eventos e Interações*

Interações:

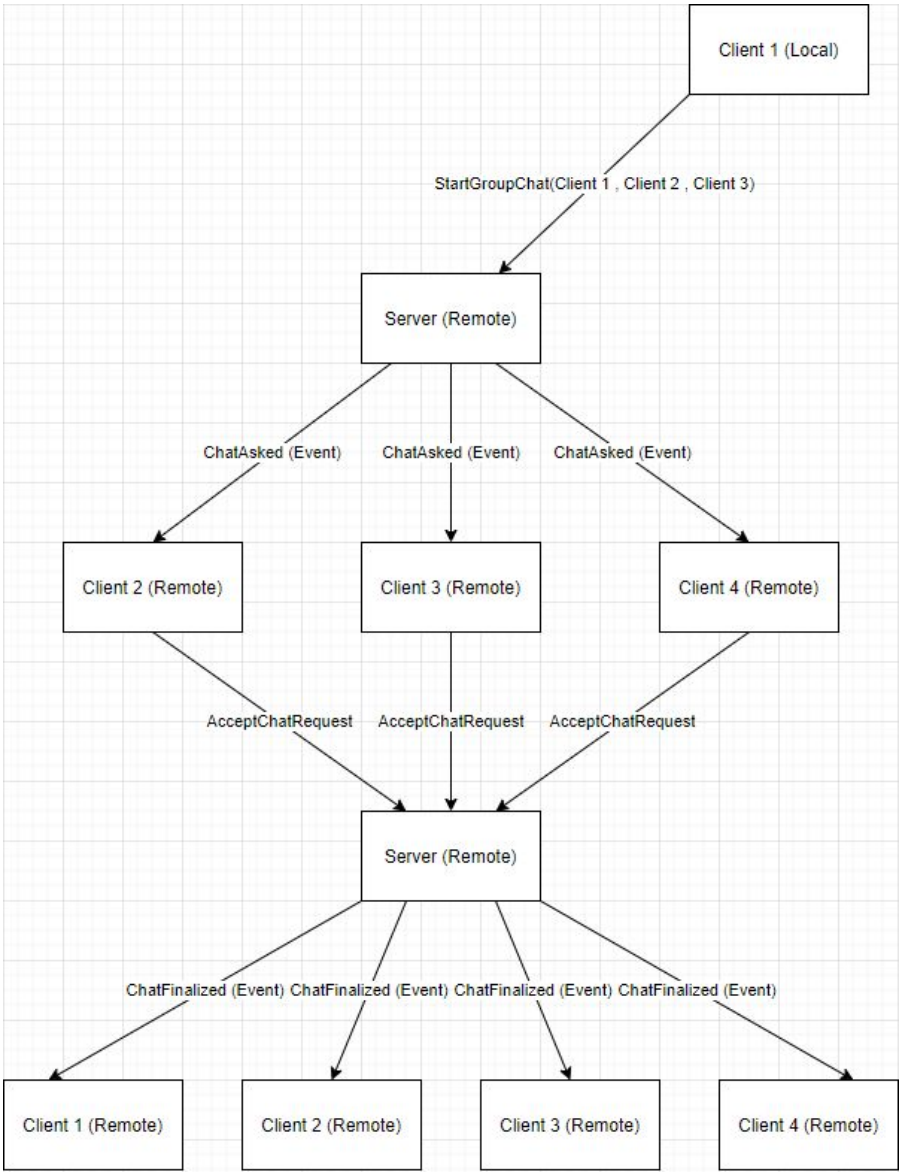


Fig. 4 – Interactions of remote objects on case of group chat request

Relativamente aos eventos, usufruímos bastante deste tipo de elementos para situações distintas, dadas a sua utilidade e facilidade de manipulação. Desta forma, usamos eventos para:

- notificar os utilizadores do *login* ou *logout* de um *user*
- enviar e receber propostas de conversações
- enviar e receber respostas acerca das referidas propostas
- enviar e receber mensagens
- enviar e receber ficheiros
- alterar o nome do *chat* nas diversas janelas de uma mesma sala

3 – Funcionalidades

3.1 – Registo

Para usufruir da aplicação, o utilizador precisa de se registar, sendo que para tal é necessária a introdução de um ***nickname***, que será a forma de os outros utilizadores o poderem identificar, o seu nome real e uma ***password*** e confirmação da mesma.

Esta funcionalidade encontra-se preparada para lidar com diversos erros, como por exemplo, caso o utilizador se pretenda registar-se com um ***nickname*** que já exista na base de dados, não permitindo que o mesmo ***nickname*** seja registado duas vezes. Um alerta também será lançado casa a ***password*** não coincida com a respetiva confirmação da mesma.

Com o fim de conferir confidencialidade ao utilizador, no campo onde a palavra passe é introduzida, cada carater é substituído por um *****. Antes de ser guardada na base de dados, como referido na introdução e objetivando proteger as informações do *user,* a *password* é sujeita a uma função de **hash**, nomeadamente à função **SHA256**.

Após efetuar, o registo, o utilizador é redirecionado para a página de ***login.***

3.2 – Autenticação

O ecrã inicial da aplicação é o da autenticação, na qual o utilizador pode autenticar-se introduzindo o seu ***nickname*** e respetiva ***password.*** Se os dados se encontrarem incorretos é exibida uma mensagem de erro. Caso contrário, esta janela fecha e é aberta a janela na qual o utilizador pode visualizar outros utilizadores que se encontrem *online* e estabelecer comunicações. Quando o utilizador faz ***login***, ele passa de desconectado a *online* para os outros utilizadores.

Nesta janela, o utilizador também possui um botão na qual pode efetuar o registo caso ainda não possua uma conta.

3.3 – *Comunicação entre dois utilizadores*

Após dar *login*, o utilizador é reencaminhado para uma janela na qual ele consegue observar todos os utilizadores do servidor, sendo que são priorizados na visualização os utilizadores que se encontram *online*. A partir do momento em que o utilizador se encontra

nesta janela, este pode iniciar ou receber uma proposta de conversa. Para iniciar uma conversa, apenas é necessário carregar no nome do utilizador com o qual deseja estabelecer um diálogo. Ao realizar esta ação, o outro utilizador receberá uma notificação e poderá aceitar, ou não, conversar com o utilizador inicialmente referido. Caso a proposta seja aceite, aparecerá um ícone de *chat* na janela do utilizador. Se o ícone referido for premido, aparecerá uma nova janela na qual é estabelecida a conversa entre os 2 utilizadores.

3.4 – *Comunicação entre múltiplos utilizadores*

Tal como no ponto anterior, após se autenticar, o utilizador consegue observar todos os utilizadores que se encontram *online*. Para iniciar uma conversa de grupo, apenas é necessário carregar no botão *"**Start Group Chat**"*. Após premir o referido botão, o utilizador pode selecionar os utilizadores com os quais deseja estabelecer uma comunicação em grupo. Ao serem selecionados pelo utilizador, a cor destes na interface do *user* mudará. Quando tiver selecionado todos, o utilizador apenas precisa de carregar no botão "***Accept***". Tal como na comunicação entre dois *users,* os utilizadores selecionados receberão uma notificação e poderão aceitar, ou não, conversar com o utilizador inicialmente referido. Se a proposta for aceite, aparecerá um ícone de *chat* na janela do utilizador. Se o mesmo ícone for premido, aparecerá uma nova janela na qual é estabelecida a conversa entre os múltiplos utilizadores, sendo que no canto inferior direito dessa janela é possível visualizar que utilizadores se encontram no grupo e respetiva sala.

3.5 – Várias conversações simultaneamente

Dado que cada sala de *chat* é representada por um ícone na janela onde é possível observar os utilizadores e *chats*, os vários ícones podem ser premidos e desta forma surgirão várias janelas que consistirão nas diversas conversas que o utilizador possui, permitindo assim manter diversas comunicações ao mesmo tempo. Para minimizar ou abrir as conversas, apenas é necessário carregar no ícone respetivo a cada conversação.

3.6 – Perseverança das salas de *chat*

Com o fim de garantir a melhor experiência possível aos utilizadores, as mensagens de cada sala de *chat* são guardadas na base de dados. Assim, caso o utilizador termine sessão, as respetivas mensagens não desaparecerão e continuarão intactas da próxima vez que iniciar sessão.

Contudo, os ficheiros não são armazenados na base de dados.

3.7 – Personalização sala de *chat*

Em qualquer sala de *chat*, quer de dois utilizadores quer de grupo, o utilizador pode escolher as cores da mensagem de cada utilizador presente na sala, inclusive do mesmo. Para tal, no canto inferior direito, ao lado do painel das mensagens, cada utilizador possui associada uma cor. Para a alterar, apenas é necessário carregar duas vezes no nome e aparecerá um novo painel que permite a alteração desta cor.

Além das cores, é possível alterar o nome do *chat* carregando na *textbox* que contém o nome. O nome será alterado em todas as janelas de todos os *users* que pertençam ao *chat* em que o nome foi alterado.

3.8 – *Envio de ficheiros*

Numa sala de chat, no painel onde o utilizador escreve a mensagem, no canto inferior direito, encontra-se um ícone com um ficheiro. Caso este seja premido, aparecerá uma janela na qual é possível escolher um ficheiro para enviar. Após a escolha, o ficheiro será enviado.

Quando um ficheiro é enviado, os restantes utilizadores do *chat* recebem uma mensagem enviada pelo *user* que enviou que contém "*Username: Sent file with name filename*". Ao premir duas vezes esta mensagem, aparecerá uma janela na qual se pode definir o destino e nome desejado para o ficheiro, sendo que este é guardado no computador dos utilizadores que receberam o ficheiro.

3.9 – *Logout*

Para efetivar o *logout,* o utilizador necessita de estar autenticado, sendo que apenas precisa de carregar no botão de fechar a janela. Quando se desconecta, os outros utilizadores que se encontram *online* visualizam a passagem do utilizador de *online* para *offline*.

4 – Modo de Funcionamento

Para usufruir da nossa aplicação, é necessário inicializar a base de dados e o servidor, bem como gerar diversas instâncias de clientes.

Quando o ***Client*** é inicializado, a aplicação abre com o ecrã de *login*.

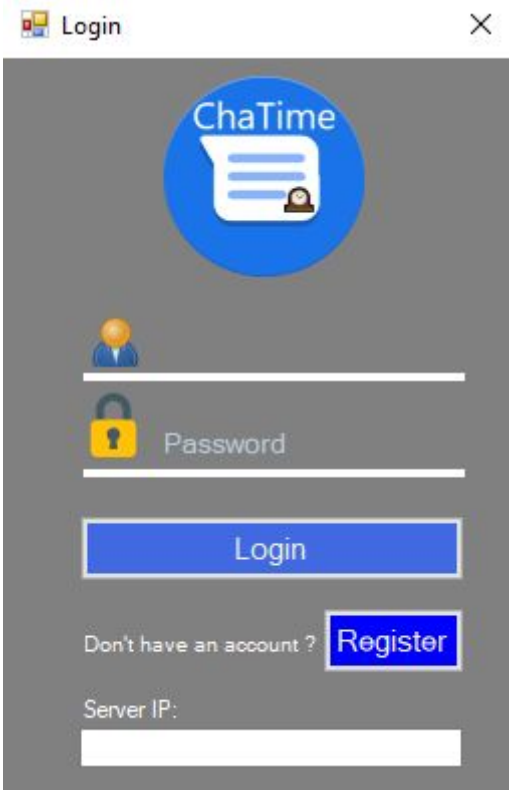
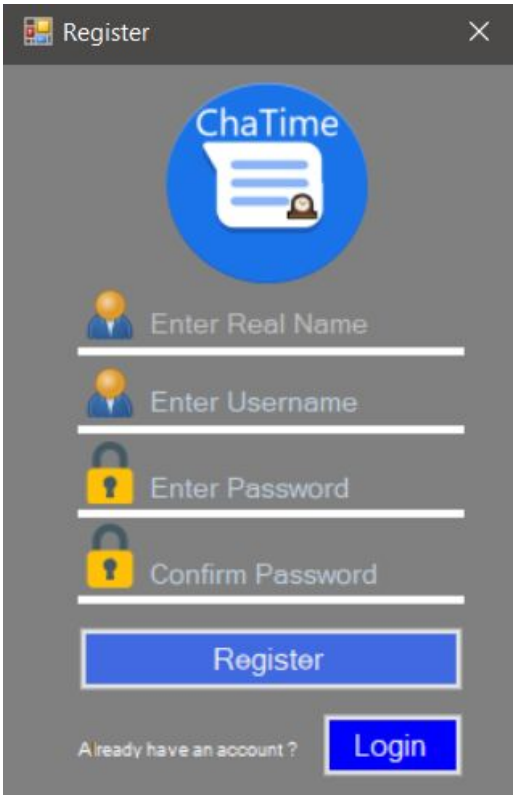


Fig. 4 – Ecrã de Login

No caso de ainda não estar registado, o utilizador pode facilmente criar uma conta carregando no botão de ***Register***. Ao carregar neste botão, o cliente é reencaminhado para o ecrã onde pode efetivar o seu registo. Para se registar, o cliente necessita de preencher os quatro campos presentes na figura **5**, sendo eles ***nickname, nome real, password*** e confirmação da ***password***. Cada carater dos últimos dois campos é substituído pelo carater ***** com o fim de conferir confidencialidade ao *user*. Além disso, como referido anteriormente, antes de ser submetida na base de dados, a palavra-passe é sujeita a uma função de **hash.**



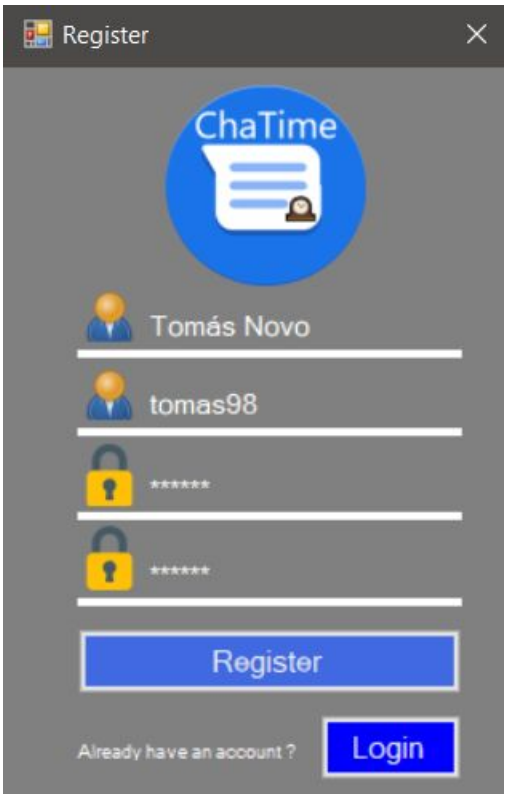
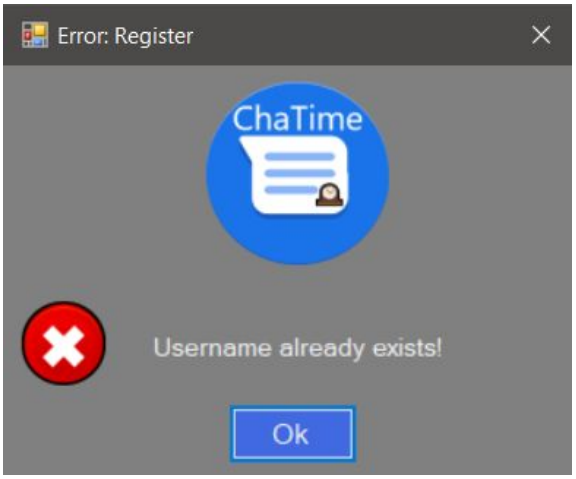


Fig. 5 e 6 – Ecrã de Registo por preencher/ Ecrã de registo preenchido

Se o ***nickname*** desejado já existir na base de dados ou se as ***passwords*** não coincidirem, serão exibidas mensagens de erro que não permitirão ao cliente efetuar o registo com sucesso.



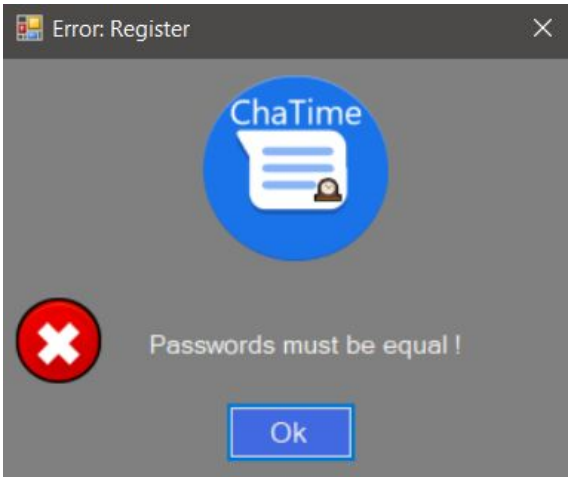


Fig. 7 e 8 – Erro de username existente / Erro password e confirmação diferentes

No caso de o registo ocorrer sem erros, o cliente é redirecionado para o ecrã de

login.

Neste ecrã, o cliente pode autenticar-se preenchendo os campos ***username*** e ***password.*** O campo ***IP,*** por definição é *localhost,* sendo apenas necessário ser introduzido caso o servidor se encontre noutra máquina.

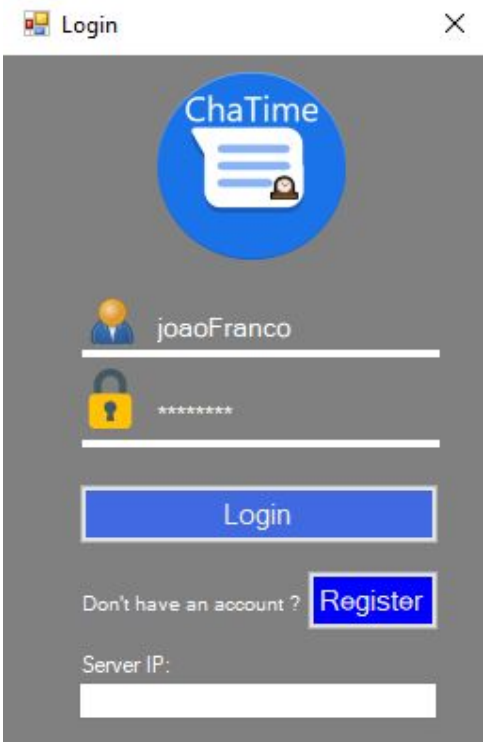


Fig. 9 – Ecrã de Login preenchido

Na eventualidade de os dados inseridos nos campos não serem válidos, isto é, se não estiverem de acordo com o que foi registado previamente na base de dados, surgirá uma mensagem de erro que impedirá o cliente de efetivar a autenticação.

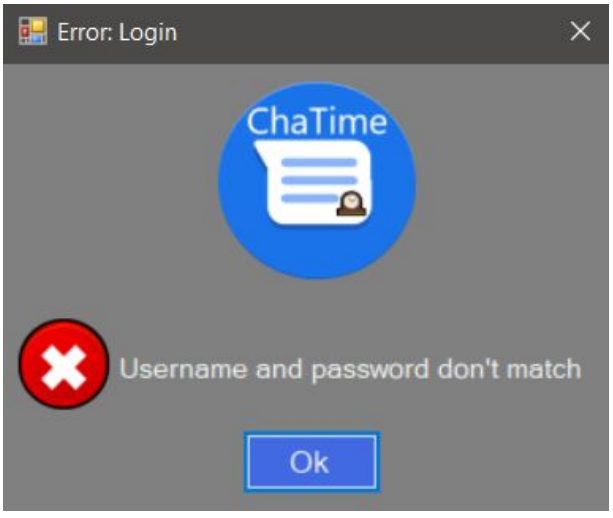


Fig. 10 – Erro validação da informação submetida

Surgirá também uma mensagem de erro caso haja uma tentativa de autenticação para um utilizador que já se encontra online.

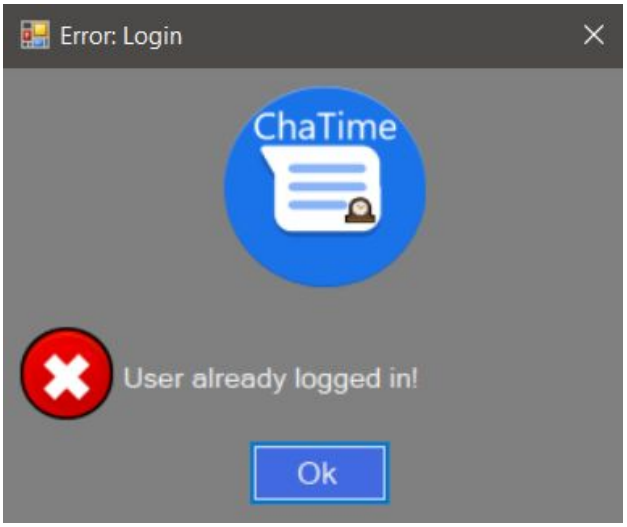


Fig. 11 – Erro validação da informação submetida

Se a autenticação for realizada com sucesso, o utilizador é reencaminhado para a janela principal, na qual pode visualizar todos os utilizadores, sendo priorizados na visualização aqueles que se encontram *online*. Acreditamos que o uso das bolinhas verdes e cinzentas para representar utilizadores *online* e *offline,* respetivamente, é simples e intuitivo. É também nesta janela que o utilizador pode enviar ou receber propostas de conversações, quer de grupo quer com outro *user.*

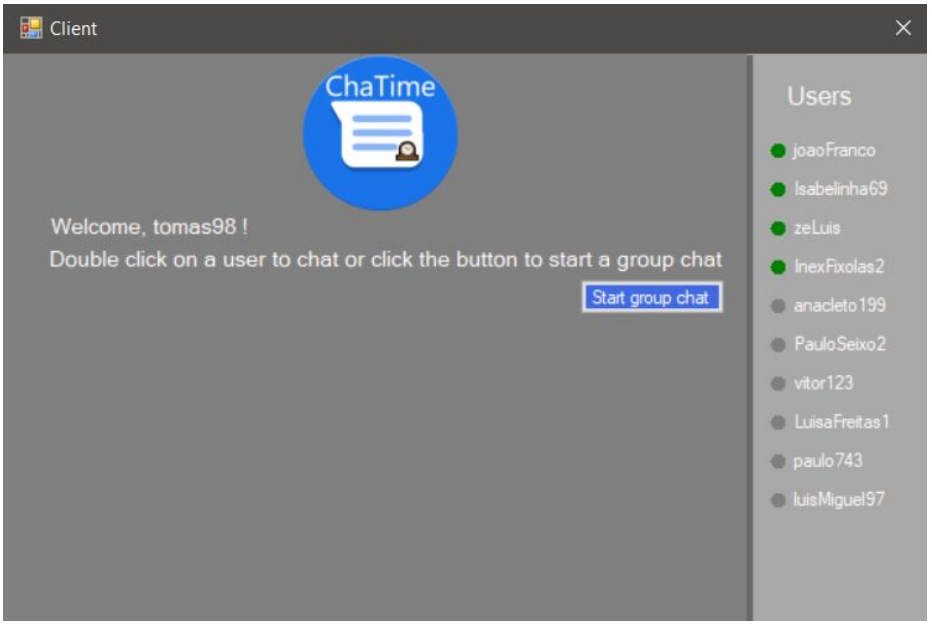


Fig. 12 – Ecrã principal

Neste ecrã, o cliente consegue visualizar no painel da direita os utilizadores que estão online.

Desta forma, o utilizador pode iniciar uma conversa com um utilizador (carregando duas vezes no nome de um utilizador) ou uma conversa de grupo (carregando no botão "*Start group chat*".

Principiando pela conversa com um *user,* para enviar uma proposta de *chat* ao utilizador desejado é apenas necessário fazer *double click* no seu nome e que ele esteja online. O utilizador recebe a proposta contida na figura **13.**

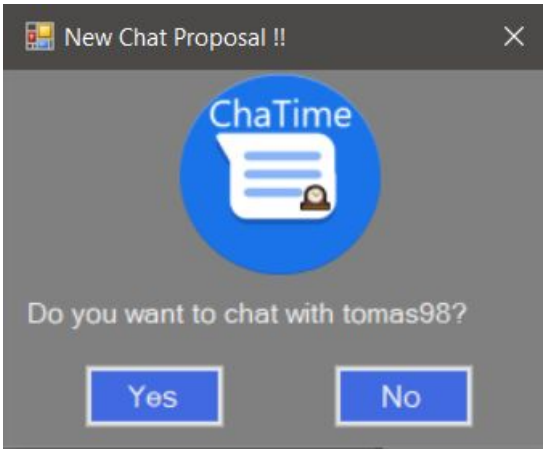
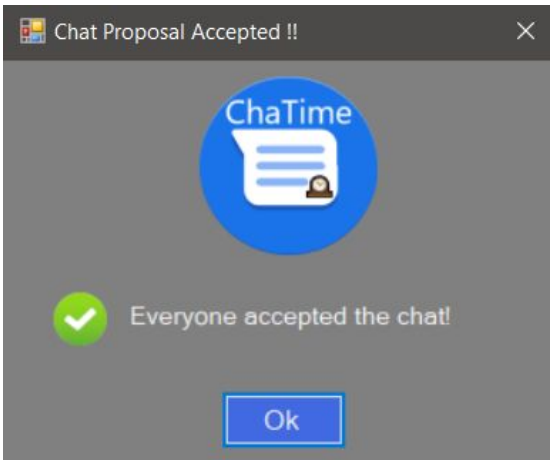


Fig. 13 – Proposta de conversação recebida

O utilizador que recebe a proposta pode então aceitar ou recusar a proposta de chat. No caso de aceitação, ambos os utilizadores receberão a notificação presente na figura **14,** caso contrário, visualizarão a notificação da figura **15.**



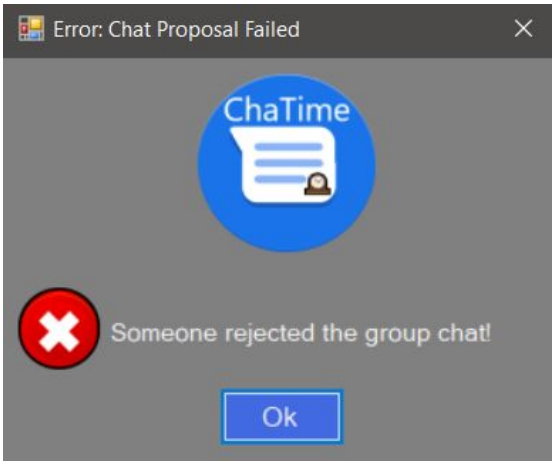


Fig. 14 e 15 – Proposta aceite / Proposta rejeitada

Caso seja premido o botão que tem como fim principiar uma conversa de grupo, a janela alterar-se-á com o fim de permitir ao utilizador escolher com que utilizadores pretende estabelecer uma conversação (figura **16**). Para escolher com quem deseja estabelecer comunicação, o *user* apenas necessita de carregar nos nomes dos utilizadores que se encontram online, sendo que os selecionados passarão a possuir cor **azul**. Quando todos os utilizadores desejados estiverem selecionados, ao carregar no botão "Accept", será enviada uma proposta para os *users* selecionados, como demonstra a figura **17**.

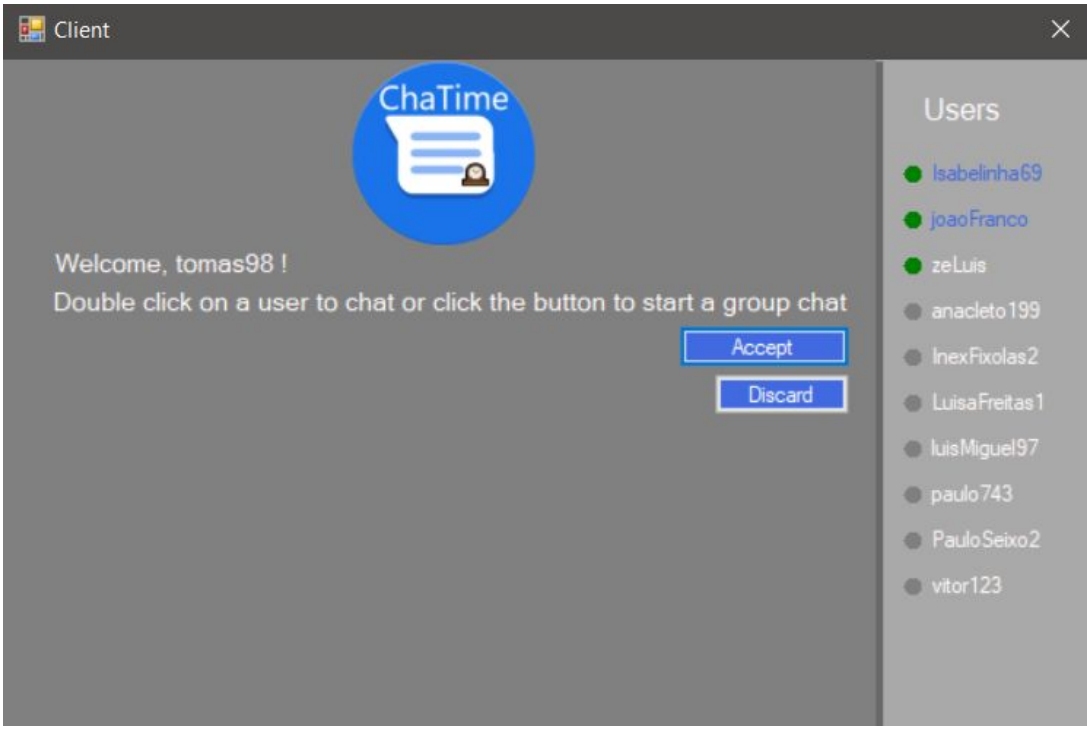


Fig. 16 – Escolha utilizadores chat de grupo

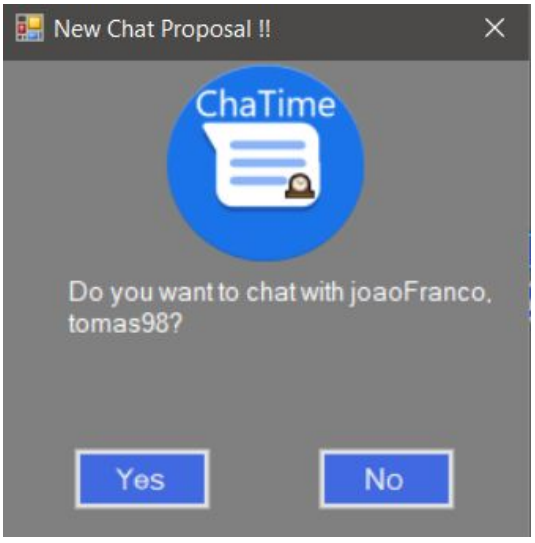


Fig. 17 – Proposta de conversação em grupo recebida

O *chat* de grupo apenas será criado se todos os utilizadores aceitarem. Na aceitação ou recusa da proposta recebida, de forma semelhante às comunicações de dois utilizadores, serão visualizadas, respetivamente, as notificações das figuras **14** e **15**.

Tanto nas propostas de grupo como individuais, no caso de aceitação, surgirá um ícone que representará o *chat* que acabou de ser aceite pelos diversos utilizadores.

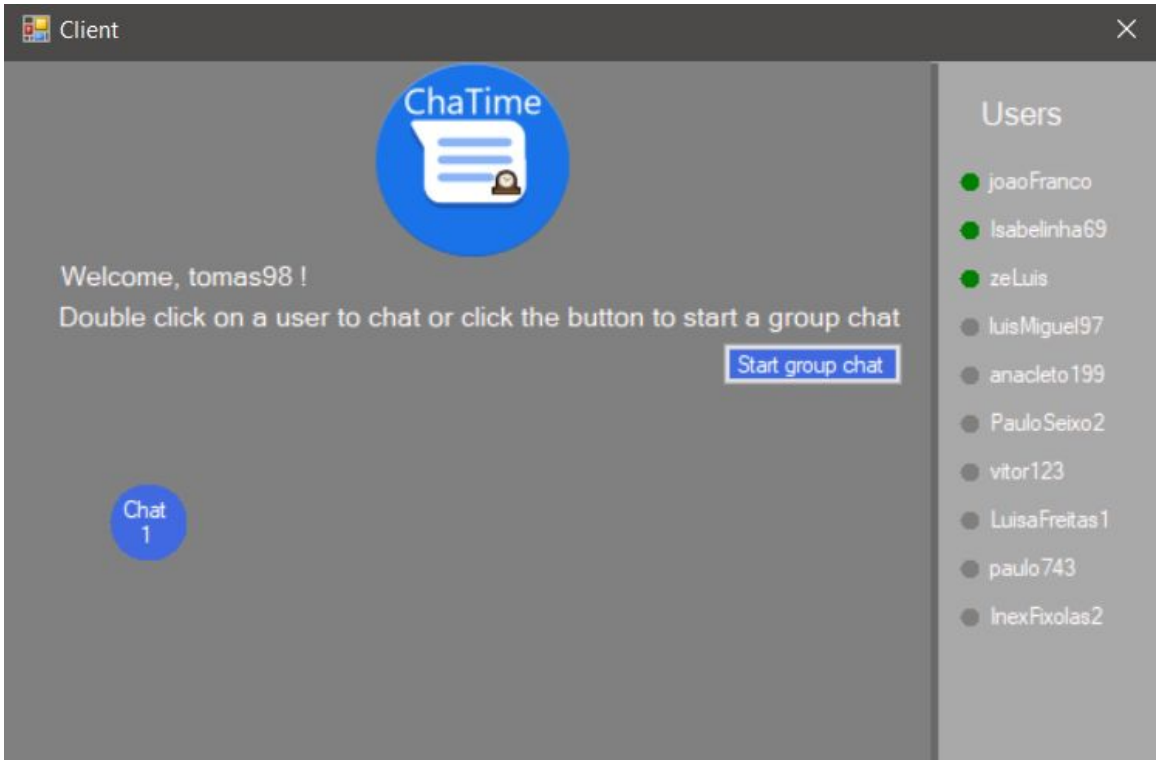


Fig. 17 – Ícone de chat

Este ícone abre ou minimiza a sala de *chat* que foi criada. A figura **18** representa a janela que surge quando o ícone é premido, ou seja, a sala de conversa em si. Nesta figura é possível visualizar diversos elementos:

- o nome do Chat (que pode ser alterado clicando e escrevendo no lugar do nome)
- os utilizadores que se encontram presentes no chat (Users In Chat)
- o painel onde é estabelecida a troca de mensagens
- o painel onde é escrita a mensagem para enviar e o botão que efetiva o envio
- um ícone onde ao clicar é possível escolher um ficheiro para enviar

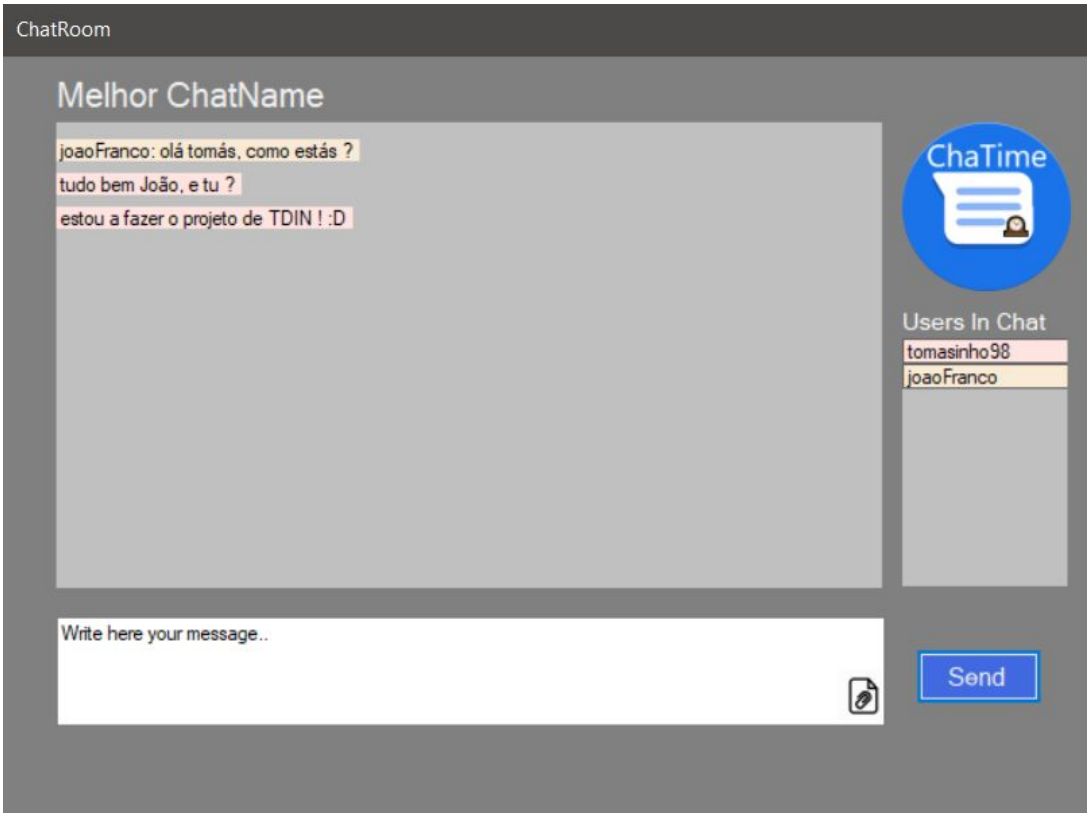


Fig. 18 – Exemplo sala de conversação

Como referido nas funcionalidades, caso seja premida duas vezes a *label* onde se encontram os utilizadores presentes no chat, surgirá uma janela que permite escolher uma nova cor para as mensagens do utilizador desejado.

Fig. 19 – Escolha de cores para as mensagens de um user

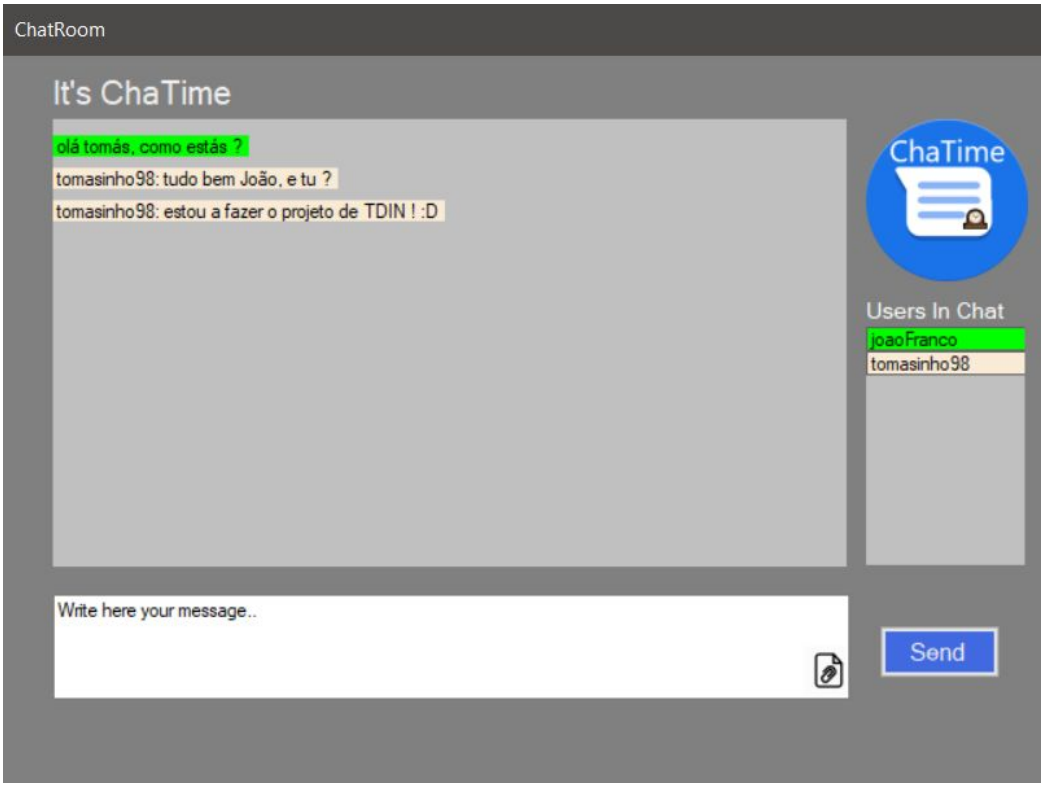


Fig. 20 – Alteração das cores das mensagens de um user

O envio de ficheiros é efetivado carregando no ícone presente no canto inferior direito do painel de envio de mensagens. Para fazer download do ficheiro, apenas é necessário carregar duas vezes na mensagem que informa que um utilizador enviou um *file*.

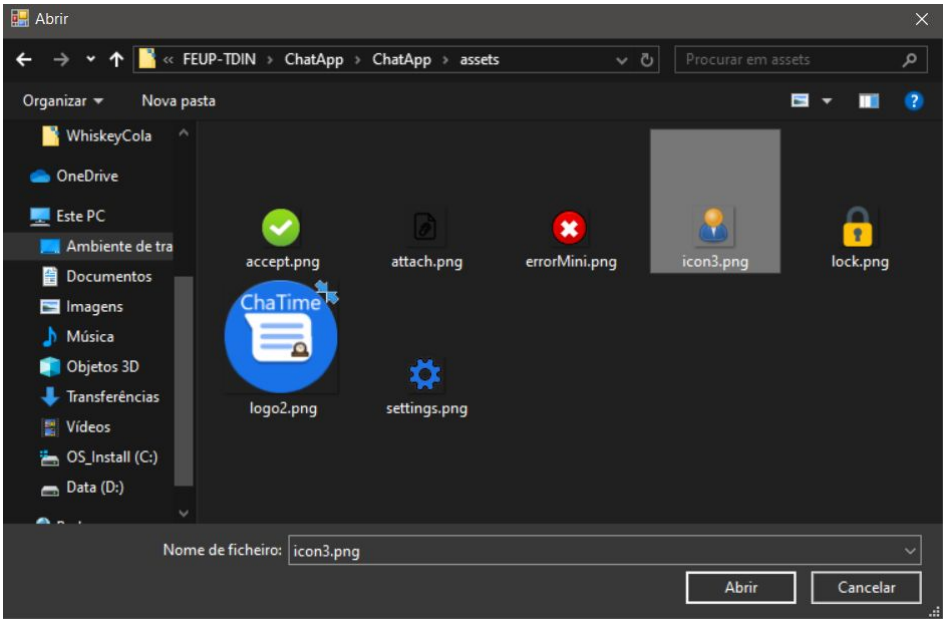


Fig. 20 – Escolha de ficheiro a enviar

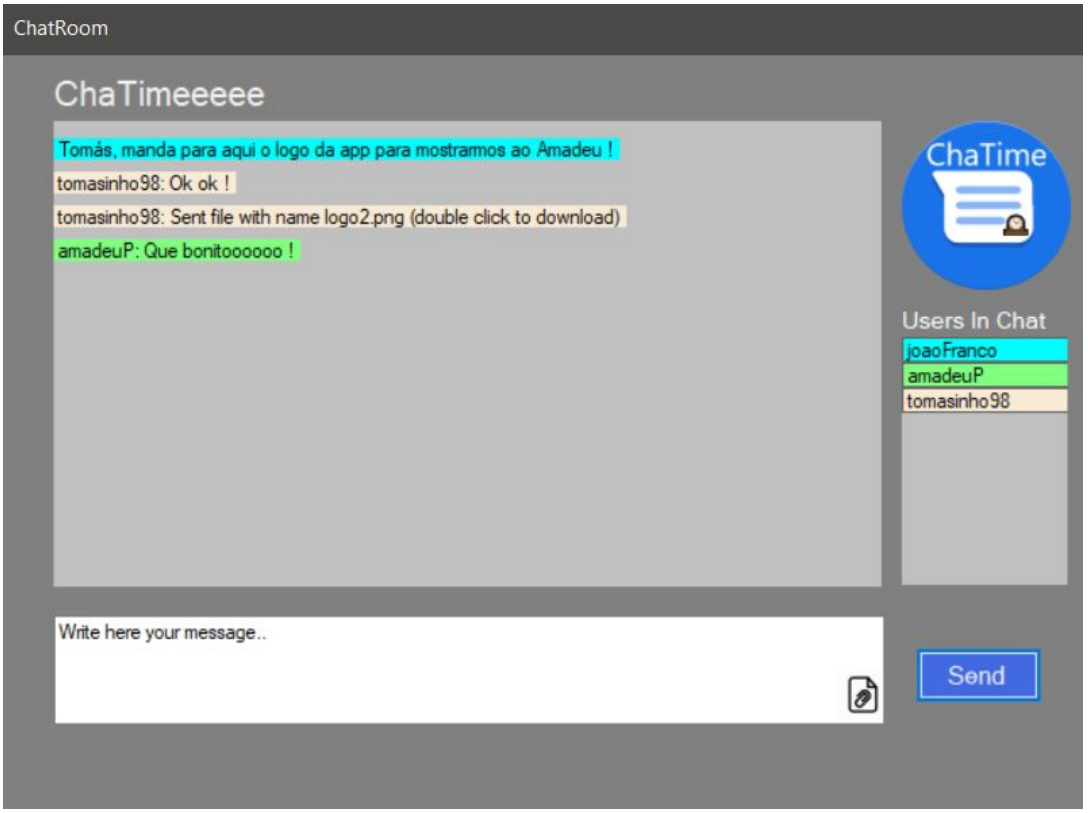


Fig. 21 – Notificação de envio de ficheiro

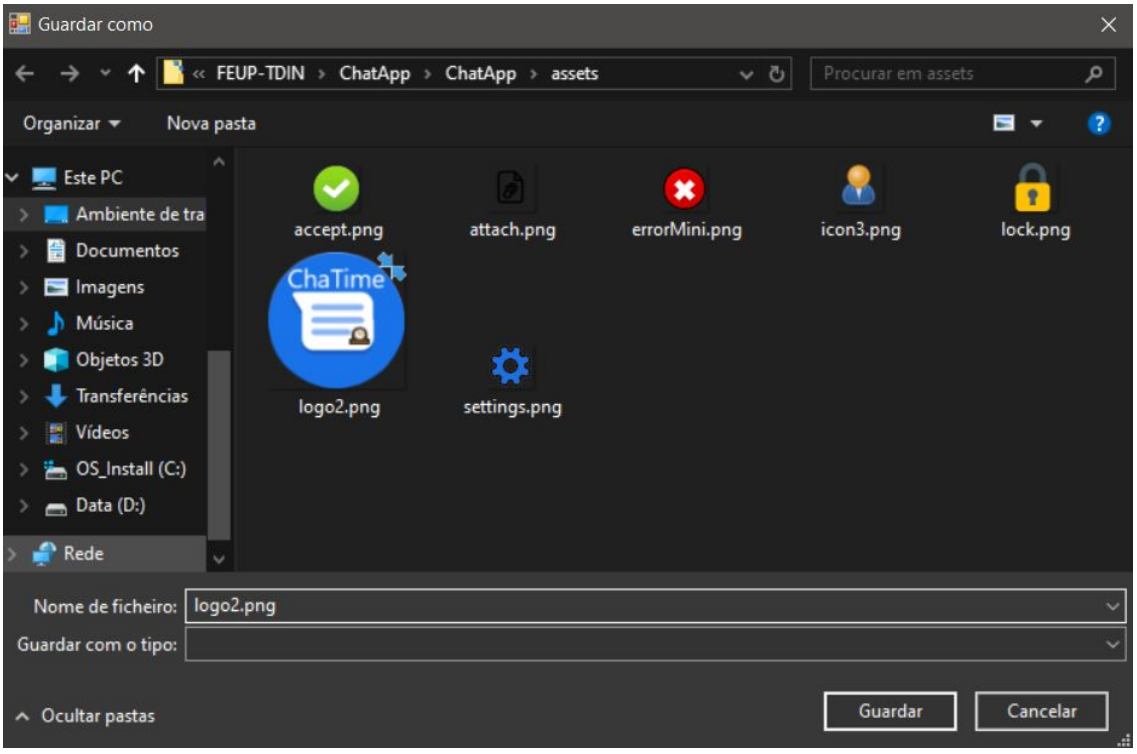


Fig. 20 – Escolha do destino do ficheiro recebido

5 – Testes Efetuados

5.1 – Registo

O teste da funcionalidade de registar utilizadores consistiu na introdução de diversas combinações de *nicknames* e *passwords.* Verificou-se que caso o utilizador não existisse na base de dados a redirecção para o ecrã de *login* era efetuada sem nenhum problema. No caso de o utilizador já existir, uma mensagem de erro é apresentada, não permitindo que o utilizador se registe com sucesso. Também é apresentada uma mensagem de erro na eventualidade de aquando do registo a *password* e respetiva confirmação da mesma não coincidirem.

5.2 – Autenticação

Objetivando testar esta funcionalidade, foram inseridas diferentes combinações de *nicknames* e *passwords.* Os resultados apurados provam que se as credenciais inseridas estiverem de acordo com as que estão guardadas na base de dados para aquele *nickname* o *login* ocorrerá sem qualquer tipo de problema, caso contrário, será apresentada uma mensagem de erro. Também foi possível observar que uma mensagem de erro surge caso um utilizador que esteja *online* se tente autenticar com o *nickname* do utilizador já autenticado.

Verificou-se também que após a autenticação com sucesso, os outros *users* conseguiram aperceber-se que o utilizador que deu *login* passou a estar *online* e, por consequência, disponível para conversar.

5.3 – *Comunicação entre dois utilizadores*

Com o fim de testar as conversações entre dois utilizadores, foram geradas diversas instâncias com diferentes clientes e foram lançadas propostas de comunicação que foram aceites. Verificou-se, como previsto, que o ícone representativo do *chat* não aparecerá, ou seja, a conversa não poderá ocorrer se o utilizador que recebe a proposta não aceitar a mesma. A troca de mensagens ocorreu sem qualquer erro tanto no envio como na receção, sendo possível visualizar ambas.

5.4 – *Comunicação entre múltiplos utilizadores*

De forma semelhante ao teste referido no ponto anterior, diversas instâncias com diferentes clientes foram geradas. Foram elaboradas diversas propostas de comunicação em grupo, verificando-se que apenas aderia ao grupo quem de facto aceita-se a proposta de adesão. Na eventualidade de aceitar, o ícone do *chat* respetivo surge na janela após se efetuar o login e é possível estabelecer uma conversação com os diversos membros do grupo. A troca de mensagens ocorreu sem erros: todos os membros conseguem enviar e receber mensagens, bem como visualizá-las.

5.5 – Várias conversações simultaneamente

Com o objetivo de testar esta funcionalidade, foram criadas várias instâncias de clientes diferentes e foram lançadas diversas propostas de comunicação tanto de dois como de múltiplos utilizadores. Conclui-se, como esperado, que apenas apareceram na janela os ícones de propostas aceites. Cada ícone representa uma conversação, e ao serem premidos diversos ícones, foi possível observar que surgiram diversas janelas que representavam salas de *chat* diferentes. Deste modo, o teste foi feito com sucesso dado que é possível efetuar diversas comunicações em janelas diferentes e de forma simultânea.

5.6 – Perseverança das salas de *chat*

Com a meta de testar se as salas de conversa de cada *user* e respetivas mensagens eram guardadas sem qualquer entrave, foram geradas várias instâncias de vários *clients,* bem como diversas salas de *chat.* Várias mensagens foram trocadas e após o término de sessão dos clientes, nas autenticações seguintes, tudo se mantinha igual a quando o utilizador fez *logout.* Para ir mais longe nos testes que efetuámos, no caso de o servidor encerrar e voltar a ser aberto, os *chats* e mensagens mantêm-se inalterados.

5.7 – Personalização sala de *chat*

Com a finalidade de testar esta funcionalidade, em diversas janelas que contêm as salas de *chat* foram alteradas as cores da mensagem de cada utilizador. Verificou-se que, como desejado, a cor da mensagem de cada utilizador apenas altera na janela em que as cores foram modificadas, permitindo uma personalização simples e eficaz sem afetar as preferências dos outros utilizadores que também podem alterar as cores de cada

mensagem conforme desejem.

Já para testar a alteração do nome do *chat,* mudou-se o nome várias vezes em várias janelas dos vários utilizadores pertencentes ao chat em *questão*. O nome foi alterado em todas as janelas de todos os *users,* concluindo-se que esta funcionalidade está a funcionar como desejado.

5.8 – *Envio de ficheiros*

Inquirindo acerca do funcionamento desta funcionalidade, em diversas salas de *chat* foi premido o ícone presente no painel de envio de mensagens com o fim de enviar diversos ficheiros. Foram enviados com sucesso ficheiros de diversos tipos, como .*png*'s, .*txt*'s e .*docx* , verificando-se que aquando do *download,* que ocorre quando a mensagem que contém o ficheiro é premida duas vezes, o ficheiro também é recebido e guardado com êxito no destino que o utilizador que recebe o ficheiro deseje.

5.9 – *Logout*

Com o objetivo de aferir acerca do desempenho desta funcionalidade, quando o utilizador fecha a janela principal (a que permite visualizar os outros utilizadores e os ícones de chat, bem como enviar propostas de comunicação), este deixa de estar *online* no servidor e os restantes *users* conseguem visualizar a passagem do utilizador que se desconectou para *offline*.

6 – Conclusão

Em suma, a realização deste projeto foi bastante positiva para os elementos do grupo devido à aquisição de conhecimentos relativamente ao ***.NET Remoting*** e à manipulação da linguagem **C#.**

A aplicação que desenvolvemos não só permite que diversos utilizadores estabeleçam uma ligação a um servidor centralizado como também permite que os *users* consigam identificar que outros utilizadores também se encontram *online***,** concedendo a hipótese de estabelecer uma comunicação ou rejeitar uma proposta de um/vários utilizador/utilizadores que desejem comunicar.

As informações de registo são arquivadas, o que permite ao utilizador a reutilização da sua conta, não sendo necessário criar uma nova cada vez que deseje usufruir dos serviços que a nossa aplicação oferece.

Acreditamos que o produto final vai de encontro ao objetivado sendo que desenvolvemos com sucesso um sistema **Internet Relay Chat** com uma interface intuitiva, permitindo aos utilizadores estabelecerem uma comunicação sem qualquer dificuldade. A aplicação não só possui todos os requisitos mínimos como também possibilita manter diversas conversas em janelas separadas, o envio de ficheiros e a criação de chats de grupo.