

2024-2025

P.PORTO

**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

SISTEMAS DISTRIBUÍDOS

TRABALHO PRÁTICO – AVALIAÇÃO CONTÍNUA

Trabalho elaborado por:

Grupo 9

8220337 – Hugo Ricardo Almeida Guimarães

8220307 – Pedro Marcelo Santos Pinho

Índice

Índice de Figuras	3
Introdução	4
a. Contextualização	4
b. Apresentação do Caso Estudo	4
1. Base de Dados	5
a. Utilizadores	5
b. Canais de Comunicação	5
c. Mensagens entre Canais.....	6
d. Mensagens privadas entre Utilizadores	6
e. Ativação de Comunicações de Emergência	6
f. Distribuição de Recursos de Emergência.....	7
g. Evacuações em Massa	7
1) Manual do Protocolo implementado	8
a) Comando de Autenticação	8
b) Comandos de Aprovação de Utilizadores.....	8
c) Comandos de Evacuação em Massa.....	8
d) Comandos de Distribuição de Recursos de Emergência.....	8
e) Comandos de Ativação de Comunicações de Emergência	8
f) Comandos de Canais	9
g) Comandos de Mensagens Privadas.....	9
h) Comandos de Notificações	9
2) IPs e portas usadas	9
3) Funcionalidades	10
a) Login	10
b) Registo	10
c) Evacuações em Massa	11
d) Distribuição de Recursos de Emergência.....	12
e) Ativação de Comunicações de Emergência	12
f) Canais de Comunicação	13
g) Mensagens privadas.....	14
h) Notificações	15
i) Gestão Hierárquica.....	16
j) Envio periódico de relatórios	18
k) Logs.....	19
4) Como Executar a Aplicação	19

5)	Perfis Criados	19
----	----------------------	----

Índice de Figuras

Figura 1 - Modelo da base de dados para os utilizadores	5
Figura 2 - Modelo da base de dados para os Canais de Comunicação	5
Figura 3 - Modelo da base de dados Para as Mensagens enviadas em Canais de Comunicação ..	6
Figura 4 - Modelo da base de dados para as Mensagens Privadas	6
Figura 5 - Modelo da base de dados para a ativação de comunicações DE EMERGÊNCIA.....	6
Figura 6 - Modelo da base de dados para a Distribuição de Recursos de Emergência.....	7
Figura 7 - Modelo da base de dados para Evacuações em Massa.....	7
Figura 8 - Ecrã de Login do cliente	10
Figura 9 -Ecrã de Registo do cliente.....	11
Figura 10 - Ecrã para a aprovação ou negas pedidos de Evacuações em Massa	11
Figura 11 - Ecrã para aprovar ou negar pedidos de Distribuição de Recursos de Emergência ...	12
Figura 12 - Ecrã para aprovar ou negar pedidos de Ativação de Comunicação de Emergência .	13
Figura 13 - Ecrã para criar ou entrar em Canais de Comunicação.....	13
Figura 14 - Ecrã para envio e receção de mensagens em um canal de comunicação	14
Figura 15 - Ecrã para a selecionar o utilizador com que se quer mandar e receber mensagens	14
Figura 16 - Ecrã para envio e receção de mensagens privadas	15
Figura 17 - Ecrã para visualizar notificações.....	16
Figura 18 - Ecrã principal para utilizadores do tipo geral (ALL)	17
Figura 19 - Ecrã principal para utilizadores do tipo Baixo (LOW)	17
Figura 20 - Ecrã principal para utilizadores do tipo médio (MEDIUM)	18
Figura 21 Ecrã principal para utilizadores do tipo alto (HIGH)	18
Figura 22 - Mensagem Broadcast de relatórios com estatísticas	19
Figura 23 - Ficheiro de logs no servidor	19

Introdução

a. Contextualização

Este projeto foi desenvolvido para a disciplina de Sistemas Distribuídos da Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto, e tem como objetivo utilizar os conhecimentos adquiridos no decorrer das aulas teóricas e práticas para a implementação de um protocolo de comunicação, e a implementação do mesmo para a comunicação de um servidor com vários clientes.

b. Apresentação do Caso Estudo

Neste projeto pretende-se desenvolver uma aplicação distribuída que auxilie a coordenação e comunicação entre diferentes entidades durante operações de emergência em cenários de catástrofes naturais, como terremotos ou inundações. A aplicação deve incluir funcionalidades que garantam a troca eficiente de mensagens e a gestão hierárquica de autorizações.

1. Base de Dados

Como o principal foco do projeto é a comunicação entre os clientes e o servidor entre sockets, decidiu-se que a base de dados seria um conjunto de ficheiros binários guardados no servidor, esses ficheiros estão guardados na pasta */database/files*.

A estrutura usada para guardar os dados em cada ficheiro é a seguinte:

a. Utilizadores

A estrutura do utilizador contém todos os dados que um utilizador precisa de ter para usar a aplicação. Neste modelo, o campo o usertype especifica o tipo de utilizador, se é um utilizador geral, de baixo, médio ou alto acesso, sendo que todos os utilizadores que não são de nível geral precisam de ser previamente aprovados por um utilizador de nível mais alto, por isso que existem os campos *isApproved* e *approvedBy*, eles identificam se um utilizador já foi aceite ou não, e se já foi, vai identificar por quem foi.

```
public class User implements Serializable {  
  
    private int id;  
    private String username;  
    private String email;  
    private String password;  
    private UserTypes userType;  
    private boolean isApproved;  
    private int approvedBy;  
}
```

FIGURA 1 - MODELO DA BASE DE DADOS PARA OS UTILIZADORES

b. Canais de Comunicação

Esta estrutura contém a informação sobre um canal, um canal é onde vários utilizadores podem juntar-se e trocarem mensagens uns com os outros, sendo que apenas utilizadores de nível médio e alto podem criar canais.

Quando um canal é criado, é gerada uma porta para o mesmo, as portas começam a partir do 1000 e vão até ao 65535. Para que todos os utilizadores recebam as mensagens em simultâneo, é criado um socket multicast para cada uma das salas, sendo o endereço do socket o 225.0.0.1, e é na porta onde serão identificados os canais para onde as mensagens serão enviadas.

```
public class Channel implements Serializable {  
    private final int id;  
    private final int ownerId;  
    private String name;  
    private int port;  
    private final List<Integer> participants;  
}
```

FIGURA 2 - MODELO DA BASE DE DADOS PARA OS CANAIS DE COMUNICAÇÃO

c. Mensagens entre Canais

Esta estrutura contém toda a informação que uma mensagem que é enviada num canal precisa de ter, ela identifica não só o canal onde foi enviada, como também quem a mandou, e a data e hora em que foi mandada.

```
public class ChannelMessage implements Serializable {  
    private int id;  
    private int channelId;  
    private int senderId;  
    private String content;  
    private String timestamp;
```

FIGURA 3 - MODELO DA BASE DE DADOS PARA AS MENSAGENS ENVIADAS EM CANAIS DE COMUNICAÇÃO

d. Mensagens privadas entre Utilizadores

Para simplificar o acesso às mensagens na base de dados, e na implementação da lógica de quem vai receber as mensagens, decidiu-se separar a estrutura de mensagens dos canais, e a estrutura de mensagens privadas entre utilizadores.

```
public class UserMessage implements Serializable {  
    private int id;  
    private int senderId;  
    private int receiverId;  
    private String content;  
    private String timestamp;
```

FIGURA 4 - MODELO DA BASE DE DADOS PARA AS MENSAGENS PRIVADAS

e. Ativação de Comunicações de Emergência

Da mesma forma que se separou as mensagens de canais e as mensagens privadas, decidiu-se separar também as estruturas de ativação de comunicações de emergência, distribuição de emergência de recursos e as evacuações em massa em estruturas diferentes, assim o acesso à base de dados fica mais simples. Ao separar-se também possibilita a implementação de lógica distinta nos atributos de cada estrutura, coisa que não seria possível se as mesmas estivessem juntas.

```
public class ActivatingEmergencyCommunications implements Serializable {  
    private final int id;  
    private final String message;  
    private int approverId;
```

FIGURA 5 - MODELO DA BASE DE DADOS PARA A ATIVAÇÃO DE COMUNICAÇÕES DE EMERGÊNCIA

f. Distribuição de Recursos de Emergência

```
public class EmergencyResourceDistribution implements Serializable {  
    private final int id;  
    private final String message;  
    private int approverId;
```

FIGURA 6 - MODELO DA BASE DE DADOS PARA A DISTRIBUIÇÃO DE RECURSOS DE EMERGÊNCIA

g. Evacuações em Massa

```
public class MassEvacuation implements Serializable {  
    private final int id;  
    private final String message;  
    private int approverId;
```

FIGURA 7 - MODELO DA BASE DE DADOS PARA EVACUAÇÕES EM MASSA

1) Manual do Protocolo implementado

Este manual descreve os comandos disponíveis no protocolo implementado, bem como os respetivos parâmetros associados.

a) Comando de Autenticação

- **LOGIN** «email» «password»
- **REGISTER** "«username»" «email» «password» «usertype»
 - usertype:
 - 0 - Utilizador aberto a todos, **all**
 - 1 - utilizador de nível **baixo**
 - 2 - utilizador de nível **médio**
 - 3 - utilizador de nível **alto**

b) Comandos de Aprovação de Utilizadores

- **APPROVE** «userForValidationId»
- **DENY** «userForDenyId»
- **PENDING_APPROVALS**
- **GET_USERS**

c) Comandos de Evacuação em Massa

- **MASS_EVACUATION** "«message»"
- **GET_MASS_EVACUATION_PENDING_APPROVALS**
- **APPROVE_MASS_EVACUATION** «massEvacuationIdToApprove»
- **DENY_MASS_EVACUATION** «massEvacuationIdToDeny»

d) Comandos de Distribuição de Recursos de Emergência

- **EMERGENCY_RESOURCE_DISTRIBUTION** "«message»"
- **GET_EMERGENCY_RESOURCE_DISTRIBUTION**
- **APPROVE_EMERGENCY_RESOURCE_DISTRIBUTION** «emergencyToApprove»
- **DENY_EMERGENCY_RESOURCE_DISTRIBUTION** «emergencyIdToDeny»

e) Comandos de Ativação de Comunicações de Emergência

- **ACTIVATING_EMERGENCY_COMMUNICATIONS** "«message»"
- **GET_ACTIVATING_EMERGENCY_COMMUNICATIONS**
- **APPROVE_ACTIVATING_EMERGENCY_COMMUNICATIONS**
«emergencyToApprove»
- **DENY_ACTIVATING_EMERGENCY_COMMUNICATIONS** «emergencyIdToDeny»

f) Comandos de Canais

- **GET_CHANNELS**
- **CREATE_CHANNEL** "«channelName»"
- **DELETE_CHANNEL** «channelId»
 - Nota: Só o dono do canal pode apagá-lo
- **JOIN_CHANNEL** «channelId»
- **LEAVE_CHANNEL** «channelId»
 - Nota: Quando o criador do canal sai do canal, o canal é apagado
- **GET_CHANNEL_MESSAGES** «channelId»
- **SEND_CHANNEL_MESSAGE** «channelId» "«message»"

g) Comandos de Mensagens Privadas

- **GET_MESSAGES_FROM_USER** «receiverUserId»
- **SEND_MESSAGE_TO_USER** «receiverId» "«message»"

h) Comandos de Notificações

- **GET_NOTIFICATIONS**

2) IPs e portas usadas

Para que o cliente funcione devidamente, este precisa de estar ligado a diversos sockets, fora o socket do servidor, sendo os IPs desses sockets:

- **IP de Broadcast para emergências (244.0.0.1)** -> para o cliente poder receber mensagens em Broadcast, este precisa de ter um thread a ouvir este socket multicast, a porta é 4321;
- **IP de relatórios (224.0.0.2)** -> Periodicamente, o servidor envia em Broadcast relatórios com estatísticas, para o cliente poder receber estes relatórios, tal como no Broadcast para emergências, precisa de ter uma thread com um socket permanentemente ligado a esse IP, a porta é 4322;
- **IP para canais de comunicação (255.0.0.1)** -> Quando um utilizador conecta-se a um canal, o programa precisa de abrir uma thread que fica ligada neste IP, a porta que a mesma usa varia de canal em canal;
- **Porta para chats privados (5600)** -> O servidor quando recebe um pedido de um utilizador para mandar uma mensagem privada a outro utilizador, este vai procurar o IP do destinatário da mensagem na lista de clientes ligados, como nessa lista é guardado o socket que faz a ligação com o cliente, facilmente consegue-se extrair o endereço de IP, e com este envia uma mensagem para a porta 5600, assim, basta o utilizador abrir um socket com o IP do servidor e essa porta para poder receber mensagens.

3) Funcionalidades

a) Login

O cliente conta com uma interface de login, a partir dela pode autenticar-se e entrar na página, ou ir para a página de registo. Uma funcionalidade que acrescenta valor no fator de segurança, é que quando o utilizador abre um socket para falar com o servidor, esse socket permanece ligado até ao cliente ou o servidor desligarem, logo, o servidor sabe sempre quem está a fazer os pedidos, logo o cliente não precisa de mandar o seu id para poder saber se pode ou não fazer determinadas operações, o servidor guarda o id do mesmo quando este faz login, assim não existe possibilidade do utilizador burlar o sistema e dizer que é outro utilizador que tem mais permissões.

Um lado negativo desta abordagem é que se o servidor for reiniciado, o cliente não poderá fazer mais ações até voltar para a página de login e autenticar-se novamente, mas como um servidor é algo que dificilmente é reiniciado, então acaba por ser um problema na parte de desenvolvimento, e não tanto na parte de produção.

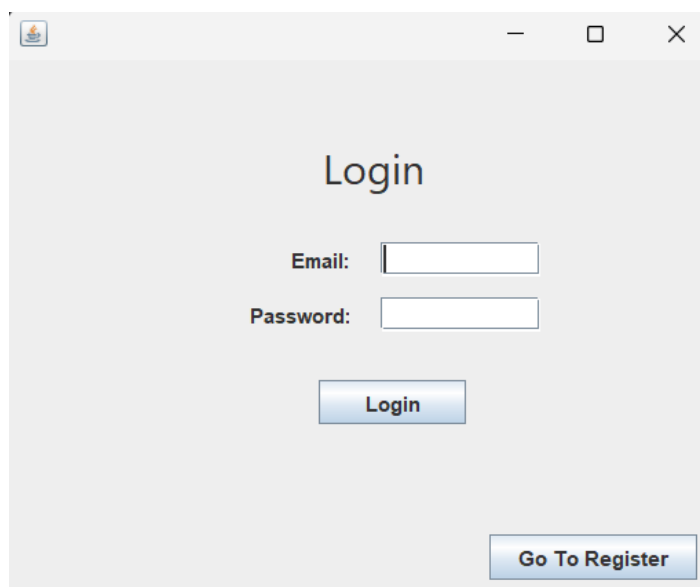
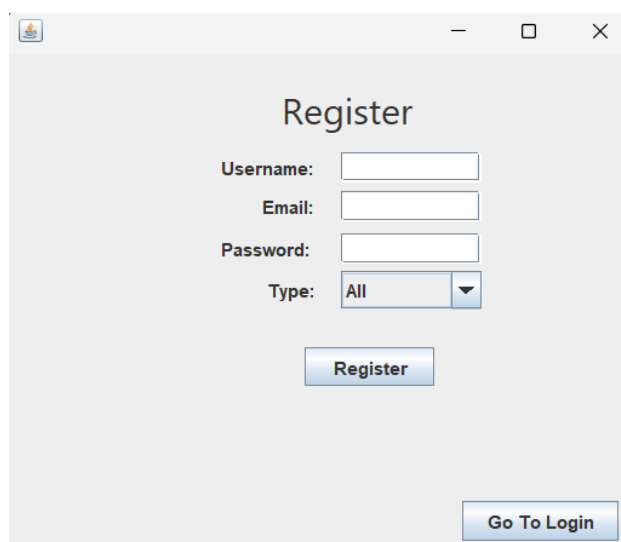


FIGURA 8 - ECRÃ DE LOGIN DO CLIENTE

b) Registo

A partir da página de login, o utilizador pode ir para a página de registo, onde o mesmo pode registar-se com um dos quatro tipos de utilizadores que existem: *All*; *Low*; *Medium* ou *High*.

Caso o utilizador registe-se como um utilizador do tipo *All*, este pode fazer login sem nenhum tipo de constrangimento, no entanto, se o mesmo fizer login com algum dos outros utilizadores, irá aparecer uma mensagem a dizer que ele foi registado com sucesso, mas que terá de aguardar que o perfil dele seja aprovado para que possa fazer login na aplicação.



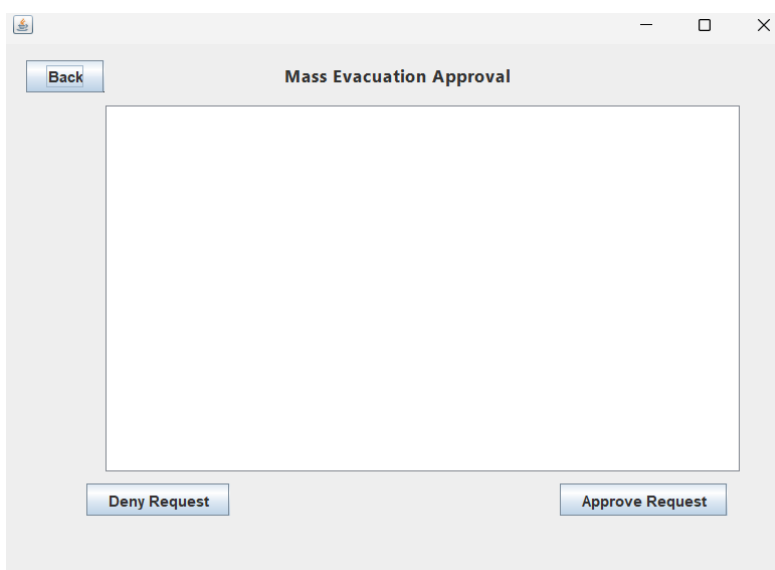
The screenshot shows a web browser window with a title bar containing a small icon and standard minimize, maximize, and close buttons. The main content area is titled "Register" in a large, bold font. Below the title, there are four input fields: "Username:", "Email:", "Password:", and "Type:". The "Type:" field is a dropdown menu currently showing "All". Below these fields is a blue "Register" button. At the bottom right of the form area is a blue "Go To Login" button.

FIGURA 9 - ECRÃ DE REGISTO DO CLIENTE

c) Evacuações em Massa

A partir da aplicação, quando já se está autenticado, um utilizador do tipo Medio ou Alto pode fazer pedidos de Evacuação em massa, caso o pedido seja realizado por um utilizador do tipo Alto, será automaticamente enviada uma mensagem em Broadcast com o conteúdo da operação de evacuação em Massa. Caso seja um utilizador de nível médio a realizar o pedido, irá aparecer uma janela a dizer que o pedido foi realizado com sucesso, mas que precisará de ser aceite por um utilizador com mais permissões.

Para responder um pedido de evacuação em massa, basta um utilizador de nível Alto entrar na aplicação, entrar na secção de aprovação ou rejeição de pedidos de evacuação em massa e aceitar ou rejeitar o pedido, quando este for respondido, o utilizador que realizou o pedido irá receber uma notificação com a resposta. Caso seja aprovado, para além da notificação, uma mensagem em Broadcast e uma notificação também será enviada para todos os utilizadores.



The screenshot shows a web browser window with a title bar. The main content area is titled "Mass Evacuation Approval". In the top left corner of the content area is a blue "Back" button. Below the title is a large, empty rectangular box, likely for displaying details of the request. At the bottom of the content area are two blue buttons: "Deny Request" on the left and "Approve Request" on the right.

FIGURA 10 - ECRÃ PARA A APROVAÇÃO OU NEGAS PEDIDOS DE EVACUAÇÕES EM MASSA

d) Distribuição de Recursos de Emergência

Na operação de distribuição de recursos de emergência, utilizadores de nível Geral (All) ou superior podem realizar pedidos. No caso de utilizadores de nível Geral, os pedidos necessitam de aprovação por parte de um utilizador de nível superior para que a mensagem em Broadcast seja transmitida. Após a aprovação, todos os utilizadores recebem uma notificação da operação de distribuição de recursos de emergência.

Semelhante à operação de evacuação em massa, quando um pedido de distribuição de recursos de emergência é aprovado ou rejeitado, o utilizador que submeteu o pedido recebe uma notificação com o resultado.

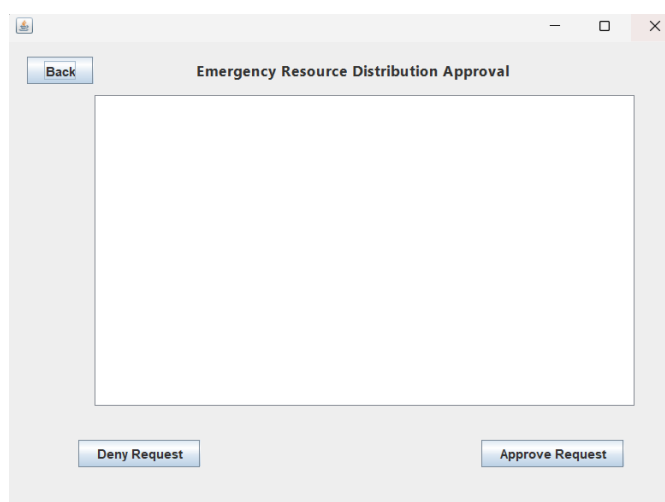


FIGURA 11 - ECRÃ PARA APROVAR OU NEGAR PEDIDOS DE DISTRIBUIÇÃO DE RECURSOS DE EMERGÊNCIA

e) Ativação de Comunicações de Emergência

A operação de ativação de comunicações de emergência é similar às operações de evacuação em massa e a operação de distribuição de recursos de emergência, a diferença é que esta operação não pode ser realizada por utilizadores de nível geral, apenas por utilizadores de nível baixo (low) para cima, sendo que os pedidos realizados por utilizadores de nível baixo precisam de ser aprovadas por utilizadores de nível médio ou alto para que a mensagem em Broadcast seja enviada.

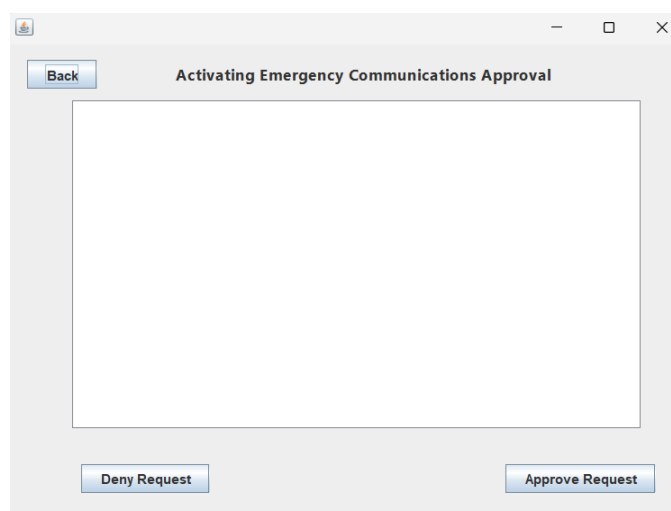


FIGURA 12 - ECRÃ PARA APROVAR OU NEGAR PEDIDOS DE ATIVAÇÃO DE COMUNICAÇÃO DE EMERGÊNCIA

f) Canais de Comunicação

Os canais de comunicação são grupos criados por utilizadores para facilitar o envio de mensagens entre os membros. Qualquer utilizador pode juntar-se a um canal existente e participar nas conversas. No entanto, apenas o criador do canal tem permissão para removê-lo.

A criação de canais é restrita a utilizadores de nível médio ou superior.

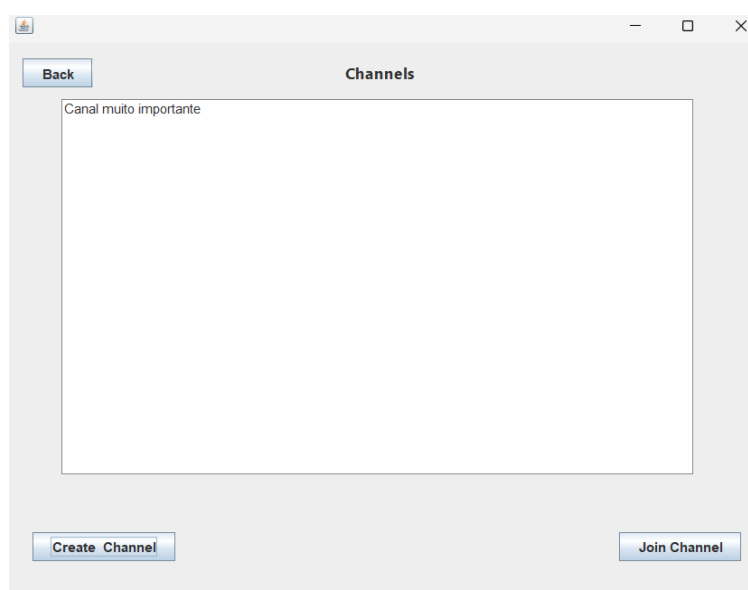


FIGURA 13 - ECRÃ PARA CRIAR OU ENTRAR EM CANAIS DE COMUNICAÇÃO

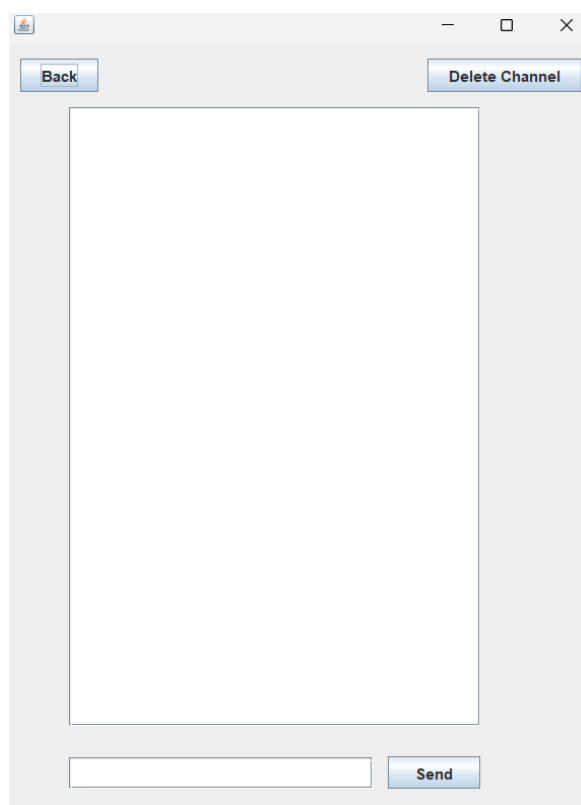


FIGURA 14 - ECRÃ PARA ENVIO E RECEÇÃO DE MENSAGENS EM UM CANAL DE COMUNICAÇÃO

g) Mensagens privadas

Da mesma forma que um utilizador pode enviar mensagens para grupos de utilizadores, o mesmo também pode mandar mensagens privadas para utilizadores específicos, basta clicar no botão que diz “Mensagens” no ecrã principal, escolher o utilizador que pretende falar no ecrã de seleção, e um chat abrirá para que o mesmo possa enviar mensagens.

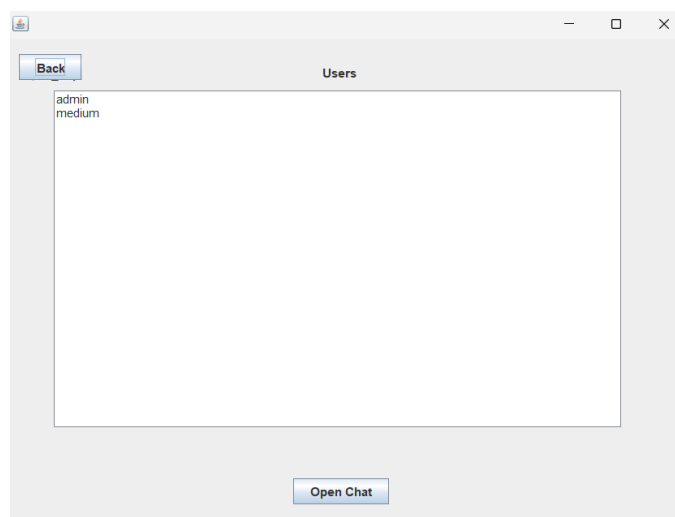


FIGURA 15 - ECRÃ PARA A SELECIONAR O UTILIZADOR COM QUE SE QUER MANDAR E RECEBER MENSAGENS

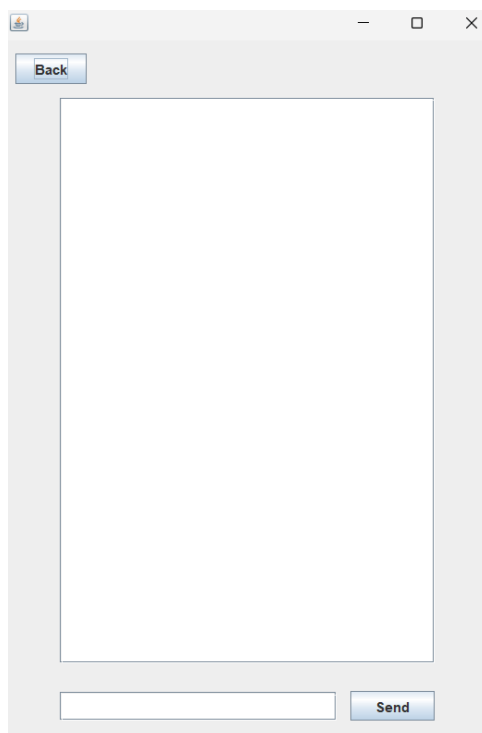


FIGURA 16 - ECRÃ PARA ENVIO E RECEÇÃO DE MENSAGENS PRIVADAS

h) Notificações

A aplicação inclui um sistema de notificações para manter os utilizadores informados sobre eventos importantes que possam ocorrer enquanto não têm a aplicação aberta. Notificações são enviadas nas seguintes situações:

- Sempre que uma mensagem Broadcast é transmitida relativamente a operações como pedidos de distribuição de recursos de emergência, ativação de comunicações de emergência ou evacuação em massa;
- Quando pedidos individuais relacionados com estas operações (distribuição de recursos de emergência, ativação de comunicações de emergência ou evacuação em massa) são aceites ou rejeitados. No caso de rejeição, apenas o utilizador que submeteu o pedido é notificado.

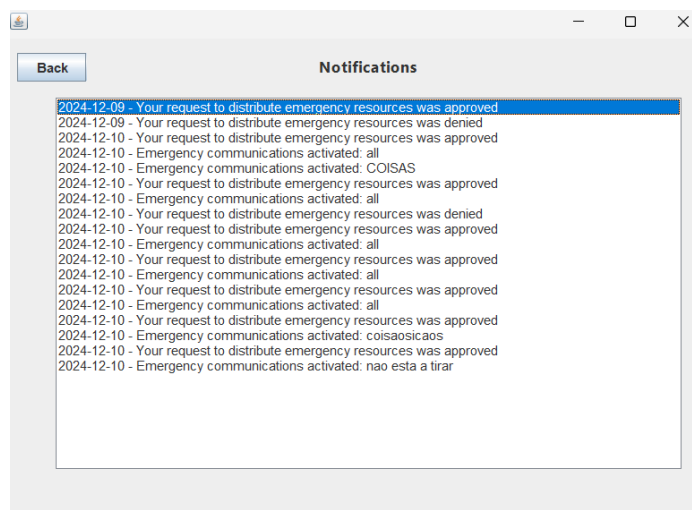


FIGURA 17 - ECRÃ PARA VISUALIZAR NOTIFICAÇÕES

i) Gestão Hierárquica

A aplicação implementa um sistema de gestão hierárquica que organiza os utilizadores em diferentes níveis de permissões, garantindo que apenas ações autorizadas possam ser realizadas. Existem quatro tipos de utilizadores:

1. **All:**

- Pode realizar pedidos de **distribuição de recursos de emergência**.
 1. Utilizadores do tipo **Low** ou superior têm permissão para aprovar estes pedidos.

2. **Low:**

- Pode realizar pedidos de **ativação de comunicações de emergência**.
 1. Estes pedidos podem ser aprovados por utilizadores de níveis superiores (Medium, ou High).

3. **Medium:**

- Pode realizar pedidos de **evacuações em massa**.
 1. Estes pedidos só podem ser aprovados por utilizadores do tipo **High**.
- Podem criar canais de comunicação

4. **High:**

- Tem permissões completas, podem realizar e aprovar qualquer tipo de operação.

Regras para permissões:

- Funcionalidades e operações não autorizadas para o nível do utilizador não aparecem na interface da aplicação;
- No servidor, todas as ações são verificadas e bloqueadas se o utilizador não tiver permissão para as realizar.

Aprovação de Utilizadores:

O sistema também gere o registo e aprovação de utilizadores:

- **Utilizadores do tipo All** podem registar-se sem necessidade de aprovação.
- Para os restantes tipos de utilizadores, o registo requer aprovação, que pode ser realizada apenas por utilizadores do mesmo nível ou de níveis superiores.

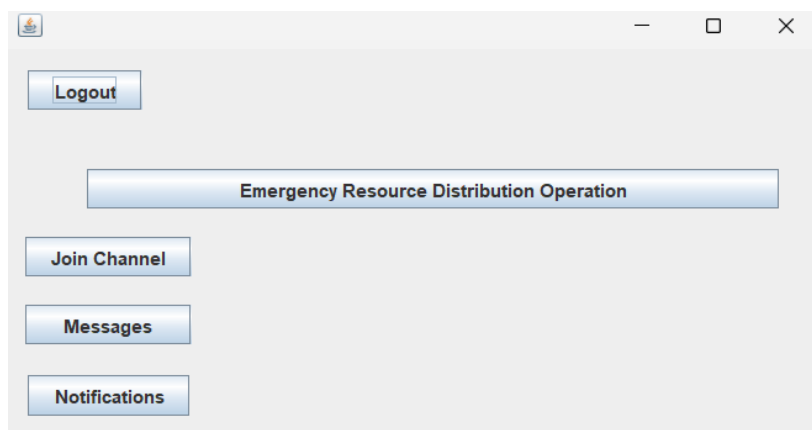


FIGURA 18 - ECRÃ PRINCIPAL PARA UTILIZADORES DO TIPO GERAL (ALL)



FIGURA 19 - ECRÃ PRINCIPAL PARA UTILIZADORES DO TIPO BAIXO (LOW)



FIGURA 20 - ECRÃ PRINCIPAL PARA UTILIZADORES DO TIPO MÉDIO (MEDIUM)



FIGURA 21 ECRÃ PRINCIPAL PARA UTILIZADORES DO TIPO ALTO (HIGH)

j) Envio periódico de relatórios

De cinco em cinco minutos o servidor irá enviar uma mensagem Broadcast para todos os clientes com estatísticas, as estatísticas são: número de utilizadores, número de canais criados; número de mensagens enviadas em canais e em privado.

Nota: Quando um cliente recebe o Broadcast com estatísticas, irá abrir uma janela a exibí-las, enquanto essa janela estiver aberta, ela irá bloquear a thread principal, sendo necessário fechar a mesma para poder voltar a usar a aplicação, isso não seria um problema, no entanto, existem ocasiões em que essa janela abre por debaixo de outras, podendo dar a impressão que o programa bloqueou, o que muito provavelmente não é o caso.

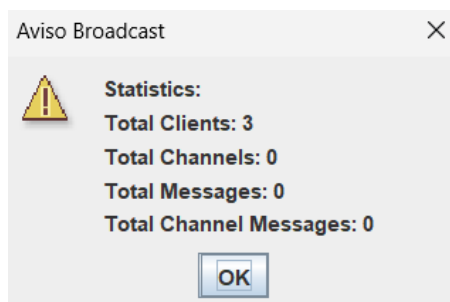


FIGURA 22 - MENSAGEM BROADCAST DE RELATÓRIOS COM ESTATÍSTICAS

k) Logs

Tanto o servidor como o cliente utilizam a biblioteca “Simple Logging Facade 4 Java” para gerar logs, portanto, enquanto os dois programas estiverem a executar, será possível visualizar os logs da execução dos mesmos no terminal onde a aplicação foi executada. No caso do servidor, os logs são guardados num ficheiro de texto (“/logs/application.log”).

```
2024-12-09 09:58:37 [Server] INFO app.estg.Server - Server started
2024-12-09 09:58:37 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 0 | Total Channels: 0
2024-12-09 09:59:01 [Server] INFO app.estg.Server - New client connected
2024-12-09 09:59:01 [server.WorkerThread] INFO i.estg.commands.auth.RegisterCommand - User isuchsidsf registered successfully
2024-12-09 09:59:01 [server.WorkerThread] INFO app.estg.threads.WorkerThread - Client disconnected
2024-12-09 09:59:36 [Server] INFO app.estg.Server - Server started
2024-12-09 09:59:36 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 1 | Total Channels: 0
2024-12-09 09:59:56 [Server] INFO app.estg.Server - New client connected
2024-12-09 09:59:56 [server.WorkerThread] INFO i.estg.commands.auth.LoginCommand - User with id 1 logged in
2024-12-09 09:59:56 [server.WorkerThread] INFO app.estg.threads.WorkerThread - Client disconnected
2024-12-09 10:04:36 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 1 | Total Channels: 0
2024-12-09 10:34:59 [Server] INFO app.estg.Server - Server started
2024-12-09 10:34:59 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 0 | Total Channels: 0
2024-12-09 10:35:44 [Server] INFO app.estg.Server - New client connected
2024-12-09 10:35:51 [server.WorkerThread] INFO i.estg.commands.auth.RegisterCommand - User dgfdgd registered successfully
2024-12-09 10:35:51 [server.WorkerThread] INFO app.estg.threads.WorkerThread - Client disconnected
2024-12-09 10:39:46 [Server] INFO app.estg.Server - New client connected
2024-12-09 10:39:48 [server.WorkerThread] ERROR app.estg.threads.WorkerThread - Error while running the server: Connection
2024-12-09 10:39:48 [server.WorkerThread] INFO app.estg.threads.WorkerThread - Client disconnected
2024-12-09 10:40:27 [Server] INFO app.estg.Server - Server started
2024-12-09 10:40:27 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 1 | Total Channels: 0
2024-12-09 10:41:05 [Server] INFO app.estg.Server - New client connected
2024-12-09 10:41:14 [server.WorkerThread] INFO i.estg.commands.auth.RegisterCommand - User low registered successfully
2024-12-09 10:41:14 [server.WorkerThread] INFO app.estg.threads.WorkerThread - Client disconnected
2024-12-09 10:42:33 [Server] INFO app.estg.Server - Server started
2024-12-09 10:42:33 [Timer-0] INFO app.estg.Server - Broadcasted message: Statistics: Total Clients: 2 | Total Channels: 0
2024-12-09 10:43:33 [Server] INFO app.estg.Server - New client connected
```

FIGURA 23 - FICHEIRO DE LOGS NO SERVIDOR

4) Como Executar a Aplicação

Para executar a aplicação, é preciso ter dois terminais abertos, um com o servidor e outro com o cliente, e realizar o comando “gradle run” no servidor e no cliente.

Nota: O comando deverá ser executado primeiramente no servidor, e só depois no cliente.

5) Perfis Criados

Para facilitar o acesso à aplicação, o servidor conta com quatro tipos de utilizadores já criados, e prontos a usar, sendo eles:

- Email: high@high.com Password: high
- Email: medium@medium.com Password: medium
- Email: low@low.com Password: low
- Email: all@all.com Password: all