

UNIVERSIDADE SALVADOR - UNIFACS

GUIA DE USO PARA APLICATIVO DE CHAMADAS

EQUIPE:

ALEXSANDRO ARAÚJO;

DEYSE GUEDES;

GILEADE TEIXEIRA;

RAMON OLIVEIRA;

SÉRGIO PIMENTEL.

FEIRA DE SANTANA

2021

SUMÁRIO

INTRODUÇÃO AO SISTEMA	3
Repositório	3
Objetivo	3
Linguagem	3
Ambiente de desenvolvimento	3
Módulos	3
Pré-requisitos	3
Executáveis	4
Procedimentos para executar arquivos .jar	4
FUNCIONAMENTO DO APLICATIVO	4
O Servidor	5
Turmas	6
Clientes	7
O Professor	7
O Aluno	10
DICIONÁRIO DE TERMOS	12
FONTES DE PESQUISA	16

INTRODUÇÃO AO SISTEMA

1. Repositório

<https://github.com/adsfsa/app-chamada>

2. Objetivo

Realizar comunicações cliente-servidor utilizando sockets e multithreads.

3. Linguagem

Java (JDK 17).

4. Ambiente de desenvolvimento

IntelliJ IDEA Community Edition 2021.2.2.

5. Módulos

- Arquivos .java → /src.
- Arquivos .class → /out/production/app-chamada.
- Pacotes: /classes, /servidor e /clientes.

6. Pré-requisitos

- **Instalar e configurar JDK** (versão atual = 17);

Link útil (JDK atual): <https://www.oracle.com/java/technologies/downloads/>

- **Windows:** <https://www.youtube.com/watch?v=Cq7gdAVPIF4>
- **Mac:** <https://www.youtube.com/watch?v=N3U-xdUxr7U> (inglês)
- **Linux:** <https://www.youtube.com/watch?v=jARiy3DZdwg>

7. Executáveis

Para realizar o download dos arquivos .jar , acesse a pasta “**jar**”, clique em um arquivo e procure a opção “Download”.

- Servidor: baixar arquivo “**Servidor.jar**” em nosso repositório;
- Professor: baixar arquivo “**ClienteProfessor.jar**” em nosso repositório;
- Aluno: baixar arquivo “**ClienteAluno.jar**” em nosso repositório.

8. Procedimentos para executar arquivos .jar

- **Windows:** shorturl.at/cyDQV (português)
- **Mac:** <https://www.youtube.com/watch?v=IV3npU1NbHQ> (inglês)
- **Linux:** <https://www.youtube.com/watch?v=DVD9DIxs9Lo> (inglês)

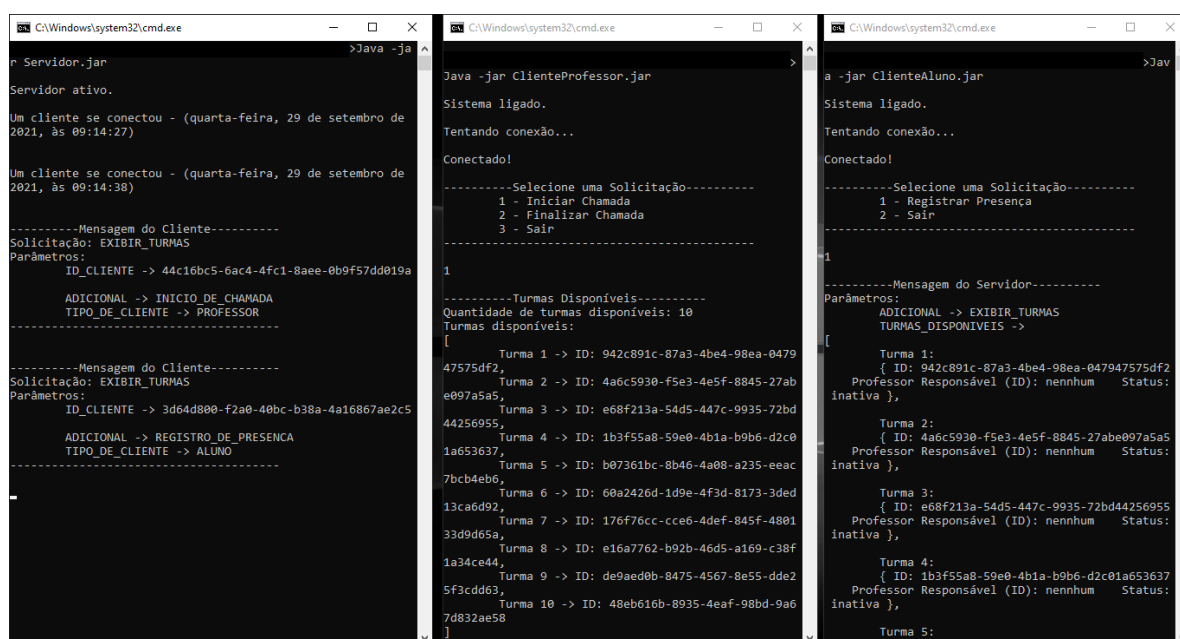
FUNCIONAMENTO DO APLICATIVO

Este aplicativo de chamadas foi criado na linguagem Java, utilizando os conceitos de sockets e multithreads. O contexto de execução é dividido em três partes: Servidor, ClienteProfessor, e ClienteAluno, cada uma representando, respectivamente, o servidor, um professor, e um aluno. Vale ressaltar que todas as classes internas são essenciais para o funcionamento correto e esperado do aplicativo.

O ServerSocket inicia uma nova rede de comunicação na porta 5555 do localhost. as comunicações se dão através de Mensagens¹, e todas as partes aplicam a Regra De Desconexão*.

A tecla Enter é a condição de validação para inputs de dados (0+Enter, 1+Enter).

¹ Para todos os termos com asterisco (*) consulte o Dicionário de Termos.



```

C:\Windows\system32\cmd.exe
>Java -jar Servidor.jar

Servidor ativo.

Um cliente se conectou - (quarta-feira, 29 de setembro de 2021, às 09:14:27)

Um cliente se conectou - (quarta-feira, 29 de setembro de 2021, às 09:14:38)

-----Mensagem do Cliente-----
Solicitação: EXIBIR_TURMAS
Parâmetros:
ID_CLIENTE -> 44c16bc5-6ac4-4fc1-8aee-0b9f57dd019a
ADICIONAL -> INICIO_DE_CHAMADA
TIPO_DE_CLIENTE -> PROFESSOR

-----Mensagem do Cliente-----
Solicitação: EXIBIR_TURMAS
Parâmetros:
ID_CLIENTE -> 3d64d800-f2a0-40bc-b38a-4a16867ae2c5
ADICIONAL -> REGISTRO_DE_PRESENÇA
TIPO_DE_CLIENTE -> ALUNO

C:\Windows\system32\cmd.exe
>Java -jar ClienteProfessor.jar

Sistema ligado.
Tentando conexão...
Conectado!

-----Seleção uma Solicitação-----
1 - Iniciar Chamada
2 - Finalizar Chamada
3 - Sair

1

-----Turmas Disponíveis-----
Quantidade de turmas disponíveis: 10
Turmas disponíveis:
[
Turma 1 -> ID: 942c891c-87a3-4be4-98ea-047947575df2,
Turma 2 -> ID: 4a6c5930-f5e3-4e5f-8845-27abe097a5a5,
Turma 3 -> ID: e68f213a-54d5-447c-9935-72bd44256955,
Turma 4 -> ID: 1b3f55a8-59e0-4b1a-b9b6-d2c01a653637,
Turma 5 -> ID: b07361bc-8b46-4a08-a235-eeac7bcb4eb6,
Turma 6 -> ID: 60a2426d-1d9e-4f3d-8173-3ded13ca6d92,
Turma 7 -> ID: 176f76cc-cce6-4def-845f-480133d9d65a,
Turma 8 -> ID: e16a7762-b92b-46d5-a169-c38f1a34ce44,
Turma 9 -> ID: de9aed0b-8475-4567-8e55-dde25f3cdd63,
Turma 10 -> ID: 48eb616b-8935-4eaf-98bd-9a67d832ae58
]

C:\Windows\system32\cmd.exe
>Java -jar ClienteAluno.jar

Sistema ligado.
Tentando conexão...
Conectado!

-----Seleção uma Solicitação-----
1 - Registrar Presença
2 - Sair

1

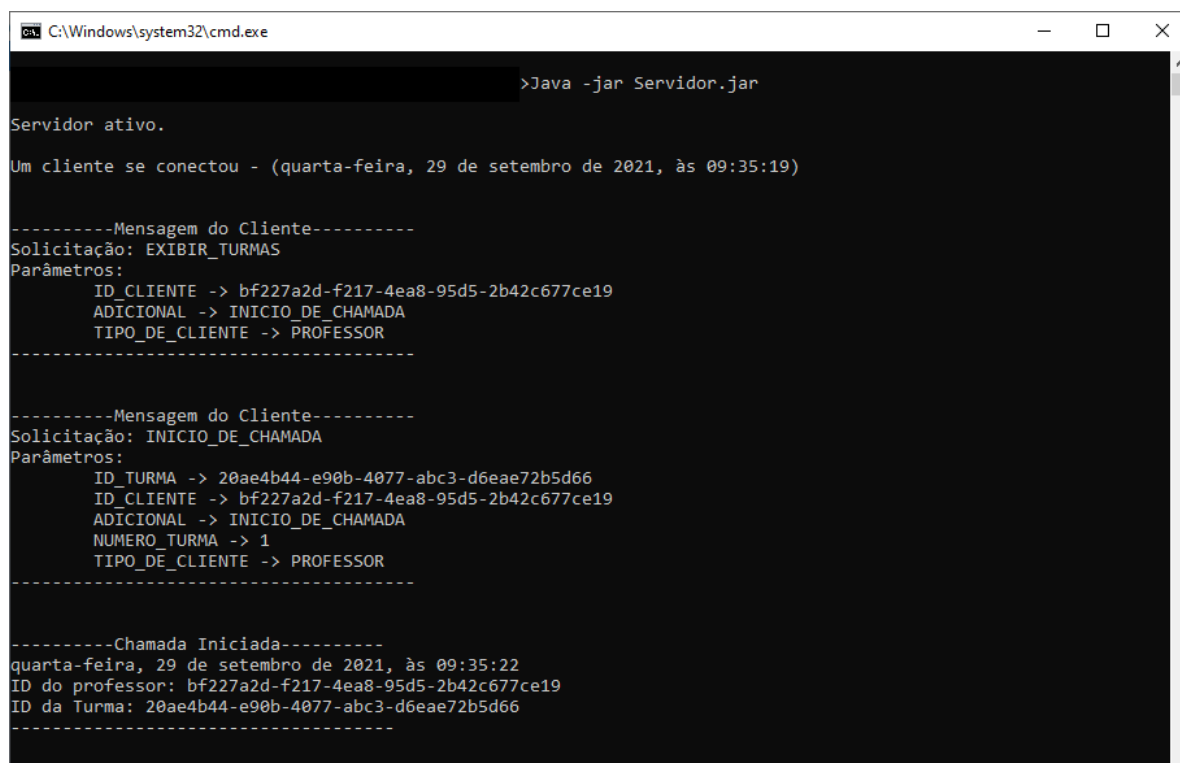
-----Mensagem do Servidor-----
Parâmetros:
ADICIONAL -> EXIBIR_TURMAS
TURMAS_DISPONIVEIS ->
[
Turma 1:
{ ID: 942c891c-87a3-4be4-98ea-047947575df2
Professor Responsável (ID): nennhum Status: inativa },
Turma 2:
{ ID: 4a6c5930-f5e3-4e5f-8845-27abe097a5a5
Professor Responsável (ID): nennhum Status: inativa },
Turma 3:
{ ID: e68f213a-54d5-447c-9935-72bd44256955
Professor Responsável (ID): nennhum Status: inativa },
Turma 4:
{ ID: 1b3f55a8-59e0-4b1a-b9b6-d2c01a653637
Professor Responsável (ID): nennhum Status: inativa },
Turma 5:

```

Imagem 1: Funcionamento do aplicativo com as três entidades conectadas.

1. O Servidor

Uma vez iniciado, aguarda a conexão de clientes (professores e alunos). O servidor registra conexões, desconexões, erros de conexão e recebimento de solicitações, e envia respostas para as solicitações recebidas.



```

C:\Windows\system32\cmd.exe
>Java -jar Servidor.jar

Servidor ativo.

Um cliente se conectou - (quarta-feira, 29 de setembro de 2021, às 09:35:19)

-----Mensagem do Cliente-----
Solicitação: EXIBIR_TURMAS
Parâmetros:
ID_CLIENTE -> bf227a2d-f217-4ea8-95d5-2b42c677ce19
ADICIONAL -> INICIO_DE_CHAMADA
TIPO_DE_CLIENTE -> PROFESSOR

-----Mensagem do Cliente-----
Solicitação: INICIO_DE_CHAMADA
Parâmetros:
ID_TURMA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
ID_CLIENTE -> bf227a2d-f217-4ea8-95d5-2b42c677ce19
ADICIONAL -> INICIO_DE_CHAMADA
NUMERO_TURMA -> 1
TIPO_DE_CLIENTE -> PROFESSOR

-----Chamada Iniciada-----
quarta-feira, 29 de setembro de 2021, às 09:35:22
ID do professor: bf227a2d-f217-4ea8-95d5-2b42c677ce19
ID da Turma: 20ae4b44-e90b-4077-abc3-d6eae72b5d66

```

Imagem 2: Servidor ativo, se comunicando com um professor.

Características

- Quando o arquivo referente ao servidor é executado, o servidor é iniciado, mas ele só é encerrado “manualmente” (encerrando a aplicação do terminal), simulando um servidor real, onde não é necessário perguntar a um usuário x se é ou não é preciso esperar mais um cliente;
- Existem n professores e alunos enviando solicitações para o servidor, por isso, a cada solicitação recebida, ele envia uma resposta para TODOS os clientes, e cabe à conexão do cliente identificar se a resposta é, ou não, para o seu cliente;
- O arquivo referente ao servidor não precisa ser iniciado antes dos arquivos dos clientes, os clientes estarão esperando o servidor ser ativado, e quando ele for, a comunicação é iniciada.

2. Turmas

Quando o servidor é iniciado, uma lista de turmas é criada automaticamente. Cada turma tem um ID*, um número de identificação (turma 1, turma 2, etc.), um status (ativa ou inativa) e um professor relacionado (o professor que abriu a chamada para a turma) (imagem 3). Uma turma também tem uma lista de matrículas, que é preenchida à medida em que os alunos vão registrando presenças (uma presença só é removida quando a turma é desativada, resetando a lista inteira).

Visando testes práticos e rápidos, a Capacidade Da Turma* foi definida para dois alunos. Se você quiser simular uma chamada mais realista (com 30, 40 alunos), clone o projeto completo, mude as configurações no arquivo “Turma.java” e faça os testes numa IDE.

```

public class Turma implements Serializable {
    /*variaveis globais*/
    private String id;
    private int numero;
    private String idProfessor;
    private String status;
    private ArrayList<Integer> listaDeMatricula = new ArrayList<>(Arrays.asList(new Integer[2]));
    private String dataHoraInicioChamada;
    private String dataHoraFimChamada;

    /*construtores*/
    public Turma(int numero) {
        this.id = UUID.randomUUID().toString();
        this.numero = numero;
        this.idProfessor = null;
        this.status = "inativa";
        this.dataHoraInicioChamada = null;
        this.dataHoraFimChamada = null;
    }
}

```




Imagem 3: Início do arquivo “Turma.java”. Para permitir mais alunos por turma, mude o valor indicado pela seta em vermelho.

3. Clientes

A depender do tipo de cliente, o menu de solicitações varia para gerar as Permissões* e Interações Específicas*. A solicitação de exibição de turmas é comum e obrigatória para todos os clientes. Cada cliente tem um ID, e uma identificação adicional se for necessário (a Matrícula* de um aluno, por exemplo).

3.1. O Professor

Quando o sistema do professor é executado, iniciam-se sucessivas tentativas de conexão com o servidor, e, ao conseguir se conectar, é exibido o menu de solicitações. O menu dá a possibilidade de um professor iniciar uma chamada para uma turma, encerrar uma chamada ativa e sair do programa.

```

C:\Windows\system32\cmd.exe
>Java -jar ClienteProfessor.jar

Sistema ligado.
Tentando conexão...
Conectado!

-----Selecione uma Solicitação-----
1 - Iniciar Chamada
2 - Finalizar Chamada
3 - Sair
-----

1

-----Turmas Disponíveis-----
Quantidade de turmas disponíveis: 10
Turmas disponíveis:
[
  Turma 1 -> ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66,
  Turma 2 -> ID: 77d3dc6a-6799-48d9-93d1-c602029f7a6b,
  Turma 3 -> ID: f1af4ca2-a629-484a-91ac-65e58a3eddff,
  Turma 4 -> ID: 358116c0-ca15-4dc2-91a8-ad412d8b725d,
  Turma 5 -> ID: 7e46d46c-c07d-48f9-a2d7-4969614dca3a,
  Turma 6 -> ID: e631c066-cb19-4033-9198-72766caea3a2,
  Turma 7 -> ID: a0ba5528-b269-4c14-88d3-718dc82fa677,
  Turma 8 -> ID: 6c18fd80-667c-45db-80e0-3579254d7be1,
  Turma 9 -> ID: 0fd0b668-38fd-47d0-8a7e-4c1852b73078,
  Turma 10 -> ID: 02ff134f-6fcd-42f6-92df-a9e348599756
]

-----
Insira o número da turma (ex.: Turma 1 -> 1), ou 0 para cancelar.
1

Turma Selecionada: { Turma 1, ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 }

-----Mensagem do Servidor-----
Parâmetros:
ADICIONAL -> INICIO_DE_CHAMADA
STATUS_DA_CHAMADA -> ativa
TURMA_REFERENCIA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
CLIENTE_REFERENCIA -> bf227a2d-f217-4ea8-95d5-2b42c677ce19 (Você)
TAMANHO_DA_TURMA -> 2
DATAHORA_INICIO_CHAMADA -> quarta-feira, 29 de setembro de 2021, às 09:35:22
NUMERO_TURMA -> 1
QUANTIDADE_DE_ALUNOS_PRESENTES -> 0

-----
Aguardando alunos...
Alunos Presentes: 0/2.

```

Imagem 4: Um professor se conectou, escolheu uma turma, e está esperando alunos.

Solicitações

- **Iniciar Chamadas:** o professor envia uma solicitação de exibição de turmas, e, após receber a resposta do servidor, é exibida a lista de turmas (é importante dizer que o servidor se encarrega de enviar apenas as turmas inativas, para facilitar a visualização e interpretação do professor. Depois disso, o sistema do professor se encarrega de exibir apenas os dados importantes, como ID e status da turma). É possível cancelar a solicitação digitando 0. Quando uma turma válida for selecionada, o sistema envia a solicitação para abertura de chamada para a turma selecionada, recebe a confirmação de abertura da chamada e aguarda a confirmação de presença dos alunos. Uma confirmação de presença é capturada quando um aluno solicita presença (o servidor envia uma resposta para todos os cliente conectados, então se a mensagem de resposta for uma confirmação de presença para a turma que o professor está esperando confirmações, a presença é exibida).


```
C:\Windows\system32\cmd.exe

Turma Selecionada:      { Turma 1,      ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 }

-----Mensagem do Servidor-----
Parâmetros:
  ADICIONAL -> INICIO_DE_CHAMADA
  STATUS_DA_CHAMADA -> ativa
  TURMA_REFERENCIA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
  CLIENTE_REFERENCIA -> bf227a2d-f217-4ea8-95d5-2b42c677ce19 (Você)
  TAMANHO_DA_TURMA -> 2
  DATAHORA_INICIO_CHAMADA -> quarta-feira, 29 de setembro de 2021, às 09:35:22
  NUMERO_TURMA -> 1
  QUANTIDADE_DE_ALUNOS_PRESENTES -> 0

-----
Aguardando alunos...

Alunos Presentes: 0/2.

-----Presença Registrada-----
Aluno (Matrícula): 841705982
-----

Alunos Presentes: 1/2.

-----Presença Registrada-----
Aluno (Matrícula): 344330081
-----

Alunos Presentes: 2/2.

Todos os alunos confirmaram presença.
```

Imagem 5: Uma chamada foi ativada e todos os alunos confirmaram presença. Após isso, o sistema retorna para o menu.

- **Encerrar Chamadas:** da mesma maneira, o professor envia e recebe uma mensagem a respeito da exibição de turmas (se este professor não tiver nenhuma turma aberta, não serão exibidas turmas disponíveis, e sim uma mensagem informando que a solicitação é impossível). A diferença aqui, é que, após selecionar uma turma válida, a turma é encerrada. Paralelamente, os alunos não poderão mais registrar presença na turma encerrada.

```

C:\Windows\system32\cmd.exe

Todos os alunos confirmaram presença.

-----Selecione uma Solicitação-----
1 - Iniciar Chamada
2 - Finalizar Chamada
3 - Sair
-----

2

-----Turmas Disponíveis-----
Quantidade de turmas disponíveis: 1
Turmas disponíveis:
[
    Turma 1 -> ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66
]
-----

Insira o número da turma (ex.: Turma 1 -> 1), ou 0 para cancelar.

1

Turma Selecionada:      { Turma 1,      ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 }

A chamada para a turma 1 será encerrada...

-----Mensagem do Servidor-----
Parâmetros:
ADICIONAL -> ENCERRAMENTO_DE_CHAMADA
STATUS_DA_CHAMADA -> inativa
DATAHORA_FIM_CHAMADA -> quarta-feira, 29 de setembro de 2021, às 10:21:52
TURMA_REFERENCIA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
CLIENTE_REFERENCIA -> bf227a2d-f217-4ea8-95d5-2b42c677ce19 (Você)
ALUNOS_PRESENTES -> [841705982, 344330081]
DATAHORA_INICIO_CHAMADA -> quarta-feira, 29 de setembro de 2021, às 09:35:22
NUMERO_TURMA -> 1
QUANTIDADE_DE_ALUNOS_PRESENTES -> 2
-----

A chamada para a turma 1 foi encerrada.

```

Imagem 6: Encerramento de chamada após uma turma ser preenchida. O sistema retornará para o menu.

```

C:\Windows\system32\cmd.exe

-----Selecione uma Solicitação-----
1 - Iniciar Chamada
2 - Finalizar Chamada
3 - Sair
-----

2

-----Turmas Disponíveis-----
Quantidade de turmas disponíveis: 0
Só é possível encerrar chamadas de turmas que você abriu.
-----

Solicitação Cancelada.

-----Selecione uma Solicitação-----
1 - Iniciar Chamada
2 - Finalizar Chamada
3 - Sair
-----

```

Imagem 7: Encerramento de chamada invalidada.

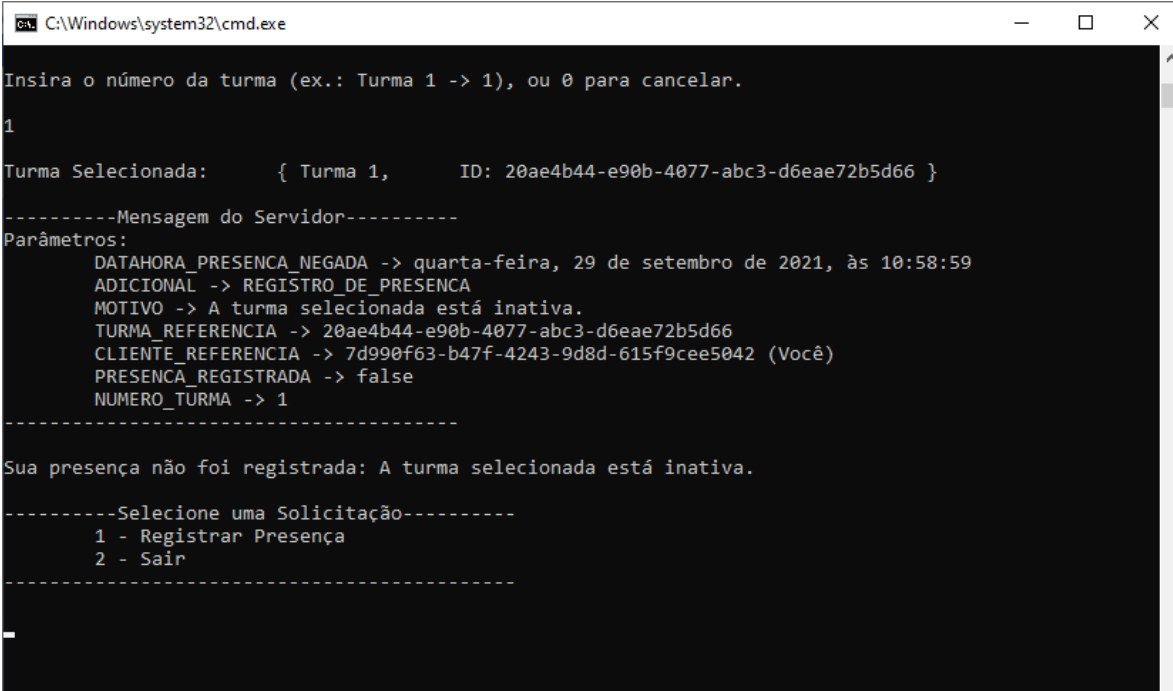
3.2. O Aluno

Assim como o professor, o sistema do aluno também tenta se conectar com o servidor inúmeras vezes quando é executado. Quando ele se conectar será exibido

um menu que possibilita o aluno registrar presença em uma turma, ou sair da aplicação.

Solicitações

- Registrar Presença: o sistema envia a solicitação referente, e espera a resposta relacionada, possibilitando a seleção da turma. No caso do aluno, todas as turmas criadas pelo servidor são exibidas, cabendo, portanto, ao aluno decidir registrar sua presença numa turma ativa (caso exista), tentar se registrar numa turma inativa (e o registro ser invalidado), ou cancelar a solicitação digitando 0. Se o previsto der certo, a confirmação de registro é arquivada no servidor e exibida para o aluno.



```
C:\Windows\system32\cmd.exe

Insira o número da turma (ex.: Turma 1 -> 1), ou 0 para cancelar.
1
Turma Selecionada:      { Turma 1,      ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 }

-----Mensagem do Servidor-----
Parâmetros:
    DATAHORA_PRESENCA_NEGADA -> quarta-feira, 29 de setembro de 2021, às 10:58:59
    ADICIONAL -> REGISTRO_DE_PRESENCA
    MOTIVO -> A turma selecionada está inativa.
    TURMA_REFERENCIA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
    CLIENTE_REFERENCIA -> 7d990f63-b47f-4243-9d8d-615f9cee5042 (Você)
    PRESENCA_REGISTRADA -> false
    NUMERO_TURMA -> 1

-----
Sua presença não foi registrada: A turma selecionada está inativa.

-----Selecione uma Solicitação-----
    1 - Registrar Presença
    2 - Sair
-----
```

Imagem 8: Presença negada.

```

C:\Windows\system32\cmd.exe
-----Selecione uma Solicitação-----
1 - Registrar Presença
2 - Sair
-----
1
-----Mensagem do Servidor-----
Parâmetros:
ADICIONAL -> EXIBIR_TURMAS
TURMAS_DISPONIVEIS ->
[
Turma 1:
{ ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 Professor Responsável (ID): bf227a2d-f217-4ea8-95d5-2b42c677ce19 Status: ativa },
Turma 2:
{ ID: 77d3dc6a-6799-48d8-93d1-c602029f7a6b Professor Responsável (ID): nenhum Status: inativa },
Turma 3:
{ ID: f1af4ca2-a629-484a-91ac-65e50a3eddfc Professor Responsável (ID): nenhum Status: inativa },
Turma 4:
{ ID: 358116c0-ca15-4dc2-91a8-ad412d8b725d Professor Responsável (ID): nenhum Status: inativa },
Turma 5:
{ ID: 7e46d46c-c07d-48f9-a2d7-4969614dca3a Professor Responsável (ID): nenhum Status: inativa },
Turma 6:
{ ID: e631c066-cb19-4033-9198-72766caea3a2 Professor Responsável (ID): nenhum Status: inativa },
Turma 7:
{ ID: a0ba5528-b269-4c14-88d3-718dc82fa677 Professor Responsável (ID): nenhum Status: inativa },
Turma 8:
{ ID: 6c18fd80-607c-45db-80e0-3579254d7be1 Professor Responsável (ID): nenhum Status: inativa },
Turma 9:
{ ID: 0fd6b668-38fd-47d0-8a7e-4c1852b73678 Professor Responsável (ID): nenhum Status: inativa },
Turma 10:
{ ID: 02ff134f-6fcd-42f6-92df-a9e348599756 Professor Responsável (ID): nenhum Status: inativa }
]
CLIENTE_REFERENCIA -> 7d990f63-b47f-4243-9d8d-615f9cee5042 (Você)
QUANTIDADE DE TURMAS DISPONIVEIS -> 10
-----
Insira o número da turma (ex.: Turma 1 -> 1), ou 0 para cancelar.
1
Turma Selecionada: { Turma 1, ID: 20ae4b44-e90b-4077-abc3-d6eae72b5d66 }
-----Mensagem do Servidor-----
Parâmetros:
DATAHORA_PRESENCA_REGISTRADA -> quarta-feira, 29 de setembro de 2021, às 10:20:59
ADICIONAL -> REGISTRO_DE_PRESENCA
MATRICULA_DO_ALUNO -> 34f230081
ID_PROFESSOR -> bf227a2d-f217-4ea8-95d5-2b42c677ce19
TURMA_REFERENCIA -> 20ae4b44-e90b-4077-abc3-d6eae72b5d66
CLIENTE_REFERENCIA -> 7d990f63-b47f-4243-9d8d-615f9cee5042 (Você)
PRESENCA_REGISTRADA -> true
NUMERO_TURMA -> 1
-----
Sua presença foi registrada.
-----Selecione uma Solicitação-----
1 - Registrar Presença
2 - Sair
-----

```

Imagem 9: Processo de registro de presença. A presença foi registrada.

DICIONÁRIO DE TERMOS

ID: Identificações únicas, do tipo String, construídas com a classe UUID.

```
String idProfessor = UUID.randomUUID().toString();
```

Imagem 10: Construção do ID de um professor.

Interações Específicas: Dependendo do tipo de cliente, o menu exibe opções diferentes (professores não marcam presença em chamadas, e alunos não iniciam/encerram chamadas).

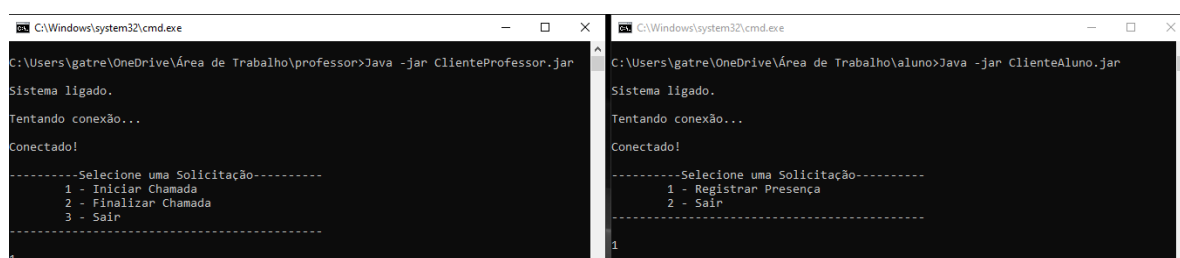


Imagem 11: Menu do professor à esquerda, e menu do aluno à direita.

Matrícula: Identificação adicional para alunos, representada por um número inteiro de nove dígitos, construído com a classe Random.

```
int matriculaAluno = new Random().nextInt( bound: 900000000) + 100000000;
```

Imagem 12: Construção da matrícula de um aluno.

Mensagem: Classe composta por uma solicitação do tipo String, e uma lista de parâmetros no formato chave-valor.

- **Solicitação:** Existem solicitações normais e exclusivas. As exclusivas não são completamente interpretadas por uma das partes do sistema (um cancelamento de solicitação só ocorre durante a escolha de uma turma, do lado do cliente, e, portanto, não precisa ser recebido no servidor. Da mesma forma, a interface do cliente só precisa exibir os parâmetros da resposta do servidor, e exibir a mensagem de uma forma mais amigável e legível). Por padrão, uma nova mensagem começa com uma solicitação CRIAR_MENSAGEM, mas ao final de todas as verificações, essa solicitação será modificada e enviada para seu destino.

```
private String solicitacao;
/*Solicitações:
 * CRIAR_MENSAGEM -> (default) (exclusiva) criar nova mensagem;
 * CANCELAR_SOLICITACAO -> (exclusiva) cancelar solicitação;
 * MENSAGEM_DO_SERVIDOR -> (exclusiva) resposta do servidor;
 * EXIBIR_TURMAS -> solicitação para exibir turmas;
 * INICIO_DE_CHAMADA -> solicitação para iniciar chamada;
 * ENCERRAMENTO_DE_CHAMADA -> solicitação para encerrar chamada;
 * REGISTRO_DE_PRESENCA -> solicitação para registrar presença;
 */
```

Imagem 13: Solicitações.

- Parâmetros: Informações da mensagem. São de diferentes tipos, e geralmente são definidos apenas de um lado da aplicação (são definidas de um lado, mas processadas pelos dois lados). Parâmetros são classificados em três níveis:
 - Nível 1: Parâmetros de extrema importância para o processamento de mensagens (definem as estruturas condicionais e loops), definindo o rumo da comunicação entre as partes (quem enviou, quem receberá, referente a qual solicitação, etc.);
 - Nível 2: Parâmetros importantes para estruturas condicionais, mas que não definem o rumo da comunicação;
 - Nível 3: Parâmetros para exibição.

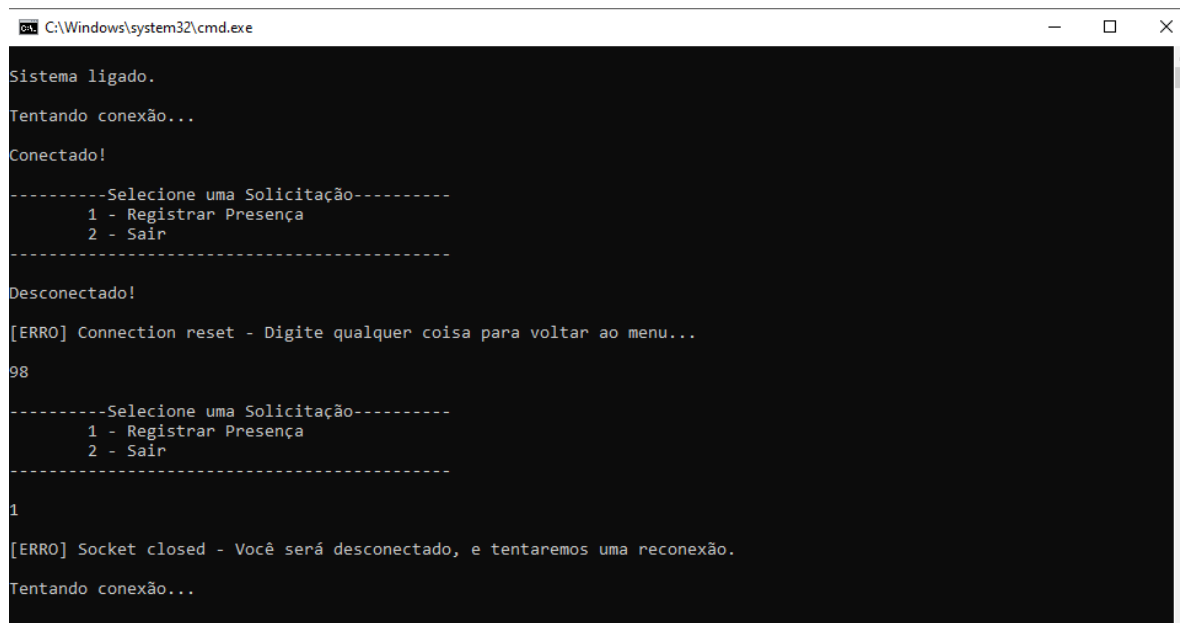
```
private Map<String, Object> params;
/* "chave" : Object valor
Parâmetros:
* ID_TURMA -> (nível 1) (String) (cliente/servidor) id da turma;
* ID_CLIENTE -> (nível 1) (String) (cliente) id do cliente;
* TIPO_DE_CLIENTE -> (nível 1) (String) (cliente) tipo de cliente;
* CLIENTE_REFERENCIA -> (nível 1) (String) (servidor) backup para o id do cliente;
* ADICIONAL -> (nível 1) (String) (servidor) informação adicional (geralmente usada como backup da solicitação; solicitação em resposta à adicional);
* MATRICULA_DO_ALUNO -> (nível 2) (Integer) (cliente) matricula do aluno;
* STATUS_DA_CHAMADA -> (nível 2) (String) (servidor) chamada ativa ou inativa;
* TURMAS_DISPONIVEIS -> (nível 2) (ListaDeTurmas extends ArrayList<Turma>) (servidor) array de turmas ativas;
* QUANTIDADE_DE_TURMAS_DISPONIVEIS -> (nível 2) (Integer) (servidor) quantidade de turmas ativas;
* ALUNOS_PRESENTES -> (nível 2) (ArrayList<Integer>) (servidor) array com matriculas dos alunos que registraram presença;
* QUANTIDADE_DE_ALUNOS_PRESENTES -> (nível 2) (Integer) (servidor) quantidade de alunos que registraram presença;
* TURMA_REFERENCIA -> (nível 3) (String) (servidor) backup para o id da turma;
* NUMERO_TURMA -> (nível 3) (Integer) (cliente/servidor) numero da turma selecionada (é mais importante para o cliente, para ele identificar a turma selecionada mais rapido);
* ID_PROFESSOR -> (nível 3) (String) (cliente) backup para o id do professor (geralmente usado por alunos);
* DATAHORA_INICIO_CHAMADA -> (nível 3) (String) (servidor) data e hora em que a chamada foi iniciada (formato: "EEEE, dd 'de' MMMM 'de' yyyy, 'às' HH:mm:ss" -> nome do dia, dia de mês de ano, às horas:minutos:segundos);
* DATAHORA_FIM_CHAMADA -> (nível 3) (String) (servidor) data e hora em que a chamada foi encerrada;
* PRESENCA_REGISTRADA -> (nível 3) (Boolean) (servidor) a chamada foi ou não foi registrada;
* DATAHORA_PRESENCA_REGISTRADA -> (nível 3) (String) (servidor) data e hora em que a presença foi registrada;
* DATAHORA_PRESENCA_NEGADA -> (nível 3) (String) (servidor) data e hora em que a presença foi registrada;
* MOTIVO -> (nível 3) (String) (servidor) explicação para negações de chamada (geralmente utilizada para explicar negações de presença registrada);
*/
```

Imagem 14: Parâmetros.

Permissões: Relacionado às Interações Específicas. O servidor garante que alunos não executem solicitações de professores, e vice-versa.

Regra De Desconexão: Lógica que simula a prevenção de desconexões (se o servidor cair o cliente desconecta, mas não fecha o programa, da mesma forma, se

o cliente cair, o servidor registra a desconexão, mas continua ativo). Por esse motivo não existe uma ordem de execução de arquivos.



```
C:\Windows\system32\cmd.exe

Sistema ligado.
Tentando conexão...
Conectado!

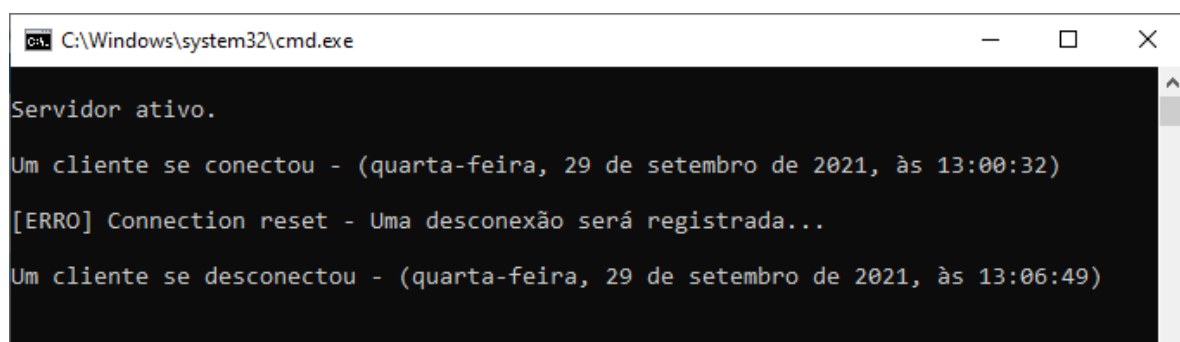
-----Selecione uma Solicitação-----
      1 - Registrar Presença
      2 - Sair
-----

Desconectado!
[ERRO] Connection reset - Digite qualquer coisa para voltar ao menu...
98

-----Selecione uma Solicitação-----
      1 - Registrar Presença
      2 - Sair
-----

1
[ERRO] Socket closed - Você será desconectado, e tentaremos uma reconexão.
Tentando conexão...
```

Imagem 15: O servidor caiu, mas o sistema continua ligado.



```
C:\Windows\system32\cmd.exe

Servidor ativo.

Um cliente se conectou - (quarta-feira, 29 de setembro de 2021, às 13:00:32)
[ERRO] Connection reset - Uma desconexão será registrada...
Um cliente se desconectou - (quarta-feira, 29 de setembro de 2021, às 13:06:49)
```

Imagem 16: Um cliente caiu, mas o servidor continua ativo.

FONTES DE PESQUISA

Stack Overflow. Disponível em: <https://stackoverflow.com/>.

BERNADI, ÉLDER. **Programação cliente-servidor com sockets em Java.** 2020. Disponível em: shorturl.at/bjyRW.

GLIDERMAN. **Java Ep31 - Networking Pt1 - Initial connection.** 2016. Disponível em: <https://www.youtube.com/watch?v=JV3YsR9wiq0>.

GLIDERMAN. **Java Ep32 - Networking Pt2 - Multithreaded server.** 2016. Disponível em: <https://www.youtube.com/watch?v=RQ2v0CSV4tY>.

GLIDERMAN. **Java Ep33 - Networking Pt3 - Multithreaded client.** 2016. Disponível em: <https://www.youtube.com/watch?v=0YvPY9gXUXg&t=2s>.

Java change color of text printed. **Grepper**, 2020. Disponível em: shorturl.at/uzDFJ. Acesso em: 26 set. 2021.

ARUCHAMY, VIKRAM. *How to Export a JAR from IntelliJ.* **Lightrun**, 2021. Disponível em: <https://lightrun.com/java/how-to-export-a-jar-from-intellij/>. Acesso em: 27 set. 2021.