

---

# **A3 - Sistemas distribuídos e mobile**

**Prof. Gustavo Torres Custódio**

---

---

# Integrantes:

**12522161564 - João dos Santos Cardoso de Jesus**

**12522167011 - Brener Meyer Mayorga Quiroz**

**12522170274 - Rafael Tetti Estevam**

**12522158236 - Omar Turkmani**

**12522159109 - Gabriel Paz De Oliveira Carvalho**

**12522211771 - Gustavo Rafael Pereira de Souza**

**12522172069 - Nicollas Oliveira Rocha**

---

---

# **Temática abordada:**

## **Chat em grupo**

---

---

# Como usar:

**Nosso projeto consiste em um sistema de chat em grupo com suporte de múltiplos clientes, utilizando sockets em Java.**

## **Como executar:**

**Execute primeiramente a Classe Server, assim que a mensagem de confirmação da ativação do servidor for recebida, pode-se iniciar múltiplos clientes. Após iniciar a classe cliente, será necessário digitar um nome de usuário para que o chat será liberado, após isso, é só conversar entre os clientes!**

---

---

# Classes:

**Agora falaremos um pouco sobre as classes do projeto e algumas das suas funcionalidades e características.**

---

---

# Server:

- **Classe responsável por ser a "host" dos clientes, que é executado localmente na porta 1234 (futuramente dinâmica), aceitando conexões dos clientes/usuários.**
  - **Além disto, ele exibe as conexões realizadas neste serviço exibindo o id do cliente na console.**
-

---

# Client:

- **Classe responsável por estabelecer conexão com o servidor, informar o nome do cliente/usuário, enviar e receber mensagens.**
  - **Também é responsável por realizar a validação de nome de usuário permitido.**
  - **O cliente também tem a opção de se desconectar da sessão ativa do servidor.**
  - **Além disso, notifica o cliente caso o servidor não esteja ativo.**
-

---

# ClientHandler:

- **Classe responsável por receber as mensagens de um determinado cliente, e envia-las aos outros clientes conectados por Broadcast.**
  - **Também foi adicionado uma capacidade técnica de possibilitar o envio de mensagens para o próprio cliente (no caso de usuário desconectado), e mensagem para um cliente específico (exemplo de bem vindo ao chat).**
  - **Recebe o nome do cliente/usuário e realiza a liberação do "chat", notificando por meio de Broadcast que um novo usuário se conectou, além de também notificar quando outros clientes se desconectam do chat room.**
-



---

# Base:

- **Uma classe abstrata que contém métodos comuns quando se fala em conexão via Socket, além de outras funcionalidades, como por exemplo: Buffer, Log, etc.**
  - **Além disso possui também uma funcionalidade que exibe uma mensagem de erro especificando nome da classe, métodos utilizados e detalhes do erro.**
-

---

# ServerUtils:

- **Classe estática/singleton cuja funcionalidade é validar usuários únicos conectados ao servidor através do uso de um `ArrayList<String>` ThreadSafe.**

---

# Dúvidas

