

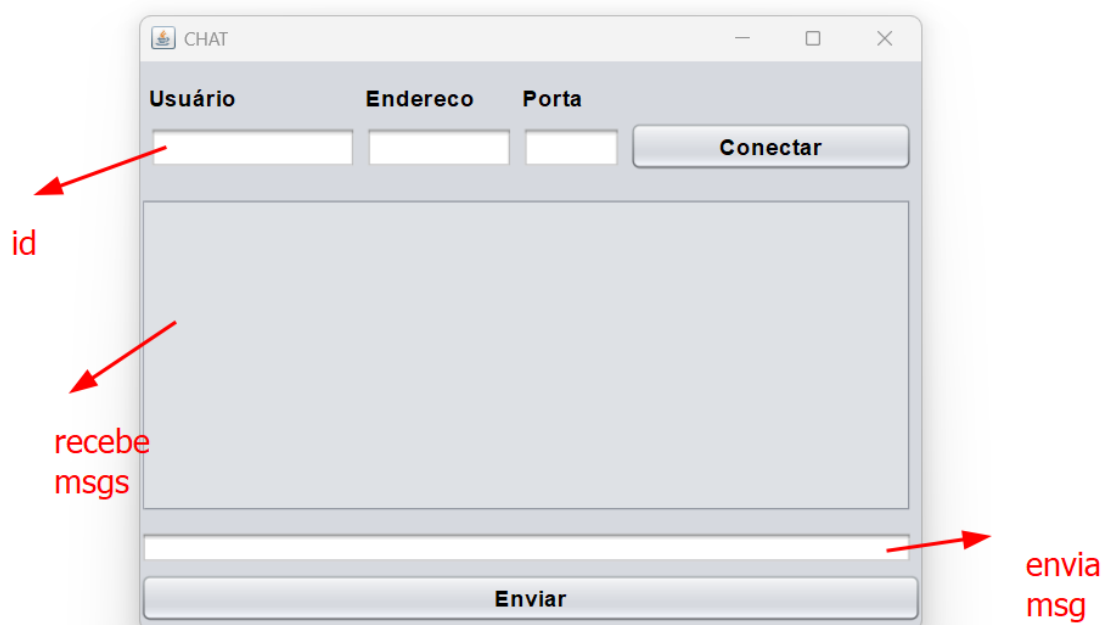
Enunciado da atividade

Faça um chat com componentes gráficos do Java utilizando a lógica disponibilizada no BatePapoOriginal.java e formulário do BatePapo.java ambos disponíveis no Atividade 7.zip anexo nesta atividade.

Regras para construir o programa

Trata-se de incluir no programa BatePapo.java a lógica do BatePapoOriginal.java – explicada na aula de UDP2.

Lembrando que o programa “BatePapo” deverá ser executado tantas vezes quanto o número de usuários.



Utilize o endereço **224.1.1.20** para a conexão multicast. A porta deverá ser uma porta efêmera, **50000**.

Não implementar nada mais do que o solicitado, tal como botões “*clique aqui para iniciar o servidor*” ou outras funcionalidades não solicitadas.

Tanto o servidor (envia mensagens) como o cliente (recebe mensagens) compartilharão a mesma tela, **portanto o servidor deverá ser iniciado primeiro.**

- O seu projeto deverá ter **seu nome como autor**, em um comentário. A classe que não tiver seu nome como autor, renderá desconto global no exercício.

Exemplo:

```
// @author Nome e Sobrenome
```

- O seu projeto deverá ser entregue com a package **local.redes**

```
package local.redes;
```

O aplicativo que for entregue com *package* diferente, receberá desconto.

- **Atenção:** para este exercício, o servidor **NÃO DEVERÁ FECHAR A CONEXÃO** e sim permanecer em estado bloqueante. Só terminará a conexão quando o usuário enviar a mensagem “Sair”
- **Implementar Threads**, baseado em videoaula de apoio. **Se não houver a implementação de threads, o programa receberá a nota zero (0).**
- A *thread* deverá ser implementada **somente no servidor**, como já demonstrado na disciplina – e – **não no cliente** (embora isso também já foi demonstrado).

Dicas importantes para concluir esta atividade

Dica 01: Este exercício pode ser feito com apenas 1 arquivo, **BatePapo.java** (cliente e servidor).

Dica 02: Lembre-se da criação de formulários na disciplina de **Linguagem de Programação Java I**.

A IDE facilita muito a criação desses formulários, e depois, você incorpora as ações/eventos nesses campos/botões.

Regras da Entrega

Anexe na atividade, o arquivo fonte **.java** e eventuais arquivos *.form*.

É permitido criar mais classes, porém, **02 classes bastam** para construir esse pequeno programa.

Não compactar nenhum arquivo. O envio compactado, como é exigido nas outras disciplinas, mas não nesta, acarretará desconto.

Submeta **os arquivos** no ambiente de ensino. **Não deixar a tarefa em Modo Rascunho.**
Clique no botão **Enviar Tarefa por Definitivo**.

Atenção para as penalidades no exercício

Quando o exercício recair em uma dessas situações e acumular erros, **os descontos abaixo serão aplicados**. A critério do avaliador, alguns descontos poderão ser isentos, dependendo do julgamento do prejuízo acumulado de nota.

Situação de descontos cumulativos	Desconto
Classes (ou uma delas) não possuem um comentário indicando sua autoria.	10,00
Entregou o exercício com <i>package</i> diferente.	15,00

Entregou o exercício em formato ZIP ou RAR ou outro, que é pedido nas outras disciplinas, mas não nesta .	15,00
Foi utilizada porta baixa ou porta reservada – o programa só funciona com a troca da porta.	20,00
Foi utilizado um IP de LAN – e não um endereço Multicast	20,00
O aluno não usou threads no programa	30,00
Servidor envia mensagens para o chat mas clientes não recebem.	60,00
Servidor recebe mensagem de si mesmo	60,00
O código não compila. O código é incompreensível.	100,00
Cópia de outro aluno, não importando se tudo está correto.	100,00