# Redes de Computadores
# Projeto de Programação – Comunicação não confiável

*Revisão 1: 11/01/2024.*

## 1. Definição do Sistema

Neste exercício deverá implementar um componente que gere uma comunicação não confiável entre dois computadores, um cliente e um servidor. A comunicação deverá ser intermediada pelo componente, instalado em cada computador, que aplicará diversas propriedades ao envio ou recebimento de cada segmento UDP.

## 2. Recomendação Inicial

Assistir o vídeo do link https://www.youtube.com/watch?v=nysfXweTI7o e implementar os exemplos mostrados em Java. Só assistir o vídeo não lhe será de utilidade quando tiver que implementar funcionalidades mais complexas.

## 3. Funcionalidades do componente (denominado Canal)

1. Receber pedidos de envio.
2. Criar o segmento UDP e enviar.
3. Receber o segmento UDP, verificando sua integridade.
4. Parametrizar (e ler) do arquivo CONFIG todos os valores abaixo. O formato do arquivo pode ser json ou outro que permita interoperabilidade (e.g., yaml).

O componente Canal deve aplicar ao envio ou recebimento do segmento UDP as seguintes propriedades:

5. Eliminar a mensagem com probabilidade 4%.
6. Gerar um atraso (*delay*) de 80 millisegundos.
7. Duplicar o segmento com probabilidade 3%.
8. Corromper somente 1 byte com probabilidade 2%.
9. Cortar os bytes do segmento maiores a 1024 bytes.

## 4. Funcionalidades do Cliente e do Servidor

O cliente:

- Inicialização: Deverá obter o IP da máquina e ler do teclado: a porta do Cliente; IP e porta do Servidor ao qual contatar. Deixe como default o IP 127.0.0.1 (só apertando a tecla *enter* já considerará esse IP)
- Deverá perguntar quantas mensagens enviar e se quer enviar de forma sequencial ou paralela. Ler o valores, e enviar essa quantidade de mensagens, usando o componente Canal.

- Deverá mostrar na console a consolidação das propriedades mencionadas após o envio de todas as mensagens. Um formato possível pode ser:
    Total de mensagens enviadas: 100.
    Total de mensagens eliminadas: 2
    Total de mensagens atrasadas: 20
    Total de mensagens duplicadas: 3
    Total de mensagens corrompidas: 1 (erro na integridade)
    Total de mensagens cortadas: 5 (erro na integridade)

O servidor:

- Inicialização: Deverá obter o IP da máquina e ler do teclado: a porta do Servidor.
- Deverá responder para o cliente um ACK do segmento recebido usando o componente Canal.
- Deverá mostrar na console a consolidação das propriedades mencionadas após o recebimento de todas as mensagens. A consolidação no recebimento deverá conter as mesmas informações mostradas no envio, mas separado por cliente.

## 5. Mensagens (prints) apresentadas na console

Na console deverão ser apresentadas, via print, as propriedades adicionadas a cada segmento (enviado e recebido).

## 6. Teste a ser realizado

### 6.1. Teste Local

O professor compilará o código e abrirá 4 consoles. Uma delas será o *servidor* e as outras três serão os *clientes*. Exemplo das consoles pode ser observado no link: https://www.youtube.com/watch?v=FOwKxw9VYqI

Os testes a serem realizados serão:

a) Executar para cada cliente 100 mensagens sequenciais, mostrando no final a consolidação.
b) Executar para cada cliente 100 mensagens paralelas, mostrando no final a consolidação.

Cabe destacar que:

- Você não precisará de 4 computadores para realizar a atividade. Basta usar 4 consoles (que correspondem a 4 processos) em 1 computador.

### 6.2. Teste Remoto

Mostre duas consoles, uma será o servidor e a outra será o cliente. O servidor e o cliente deverão estar separados obrigatoriamente por uma WAN.

Os testes a serem realizados serão:

a) Mostre tanto o IP quanto a localização geográfica da máquina.
b) Realize os mesmos testes sequenciais e paralelos da seção 6.1.

## 7. Código fonte

- Deverá criar somente as classes Cliente, Servidor e Canal. Caso envie novas classes, serão descontados 3 pontos da nota final.
    - A única exceção é a criação das classes para Threads, mas elas deverão ser criadas dentro de uma das classes acima (e.g, classes aninhadas).

- O código fonte deverá apresentar claramente (usando comentários) os trechos de código que realizam as funcionalidades mencionadas na Seção 3.

- Independente da linguagem, o uso de bibliotecas que realizem parte das funcionalidades pedidas não será aceito. Caso tenha dúvidas de alguma específica, pergunte ao professor.

## 8. Entrega

A entrega é individual e consistirá em: (a) um relatório com nome SeuRA.pdf; (b) o código fonte do programa com as pastas. A entrega será como definida no plano de ensino, não sendo aceita por outras formas.

- Caso tenha utilizado uma biblioteca permitida pelo professor, envie-a também.

O relatório deverá ter obrigatoriamente as seguintes seções:

a) Nome e RA do participante.

b) Link do vídeo do funcionamento (screencast). O vídeo do screencast deverá conter no máximo 5 minutos, mostrando a compilação, o funcionamento do código no teste local e no teste remoto. **Não envie o vídeo do screencast**, envie só o link do vídeo, o qual pode disponibilizar no Youtube ou em outro lugar semelhante (como vimeo). Lembre-se de dar permissão para visualizá-lo.

c) No vídeo do screencast do ponto anterior, deve realizar e mostrar os testes (local e remoto) da Seção 6.

d) Para cada funcionalidade do componente, uma breve explicação em “alto nível” de como foi realizado o tratamento da requisição. Na explicação DEVE mencionar as linhas do código fonte que fazem referência. Cada estrutura de dados criada deve ser explicada (para que serve, o que armazena, etc).

e) Explicação do uso das threads, mencionado as linhas do código fonte que fazem referência.

f) Explique quais foram as diferenças nas informações apresentadas nos tipos de testes local e remoto. Caso não tenha percebido diferenças, modifique os parâmetros do arquivo CONFIG. Cabe destacar que os mesmos parâmetros deverão ser usados em ambos tipos de testes.

g) Links dos lugares de onde baseou seu código (caso aplicável). Prefiro que insira os lugares a encontrar na Internet algo similar.

## 9. Observações importantes sobre a avaliação

A seguir mencionam-se alguns assuntos que descontarão a nota.

- Código fonte não compila ou usa bibliotecas externas sem aprovação do professor (nota zero).
- Não enviou o relatório (menos 2 pontos).
- Não enviou o link do vídeo, o link não está disponível ou o vídeo não mostra o funcionamento, separado nas consoles (menos 2 pontos).
- Enviou outros arquivos do código fonte além dos citados na Seção “Código fonte”, por exemplo os .class ou outras classes (menos 3 pontos).
- Código fonte sem comentários que referenciem as funcionalidades da Seção 3 (menos 2 pontos)

## 9. Links recomendados (para Java)

- Para baixar a JDK 8
  https://www.oracle.com/br/java/technologies/javase/javase-jdk8-downloads.html

- Vídeo explicativo sobre programação UDP, TCP e Threads:
  https://www.youtube.com/watch?v=nysfXweTI7o

- Informações sobre programação com UDP podem ser encontradas em:
  https://www.baeldung.com/udp-in-java
  https://www.geeksforgeeks.org/working-udp-datagramsockets-java/
  https://docs.oracle.com/javase/tutorial/networking/datagrams/index.html

- Informações sobre Threads, que permitem que o servidor ou peer receba e envie informações de forma simultânea:
  https://www.baeldung.com/a-guide-to-java-sockets (Seção 6: TCP com muitos clientes)
  https://www.tutorialspoint.com/java/java_multithreading.htm

## 10. Ética

Cola, fraude, ou plágio implicará na nota zero a todos os envolvidos em todas as avaliações e exercícios programáticos da disciplina.