

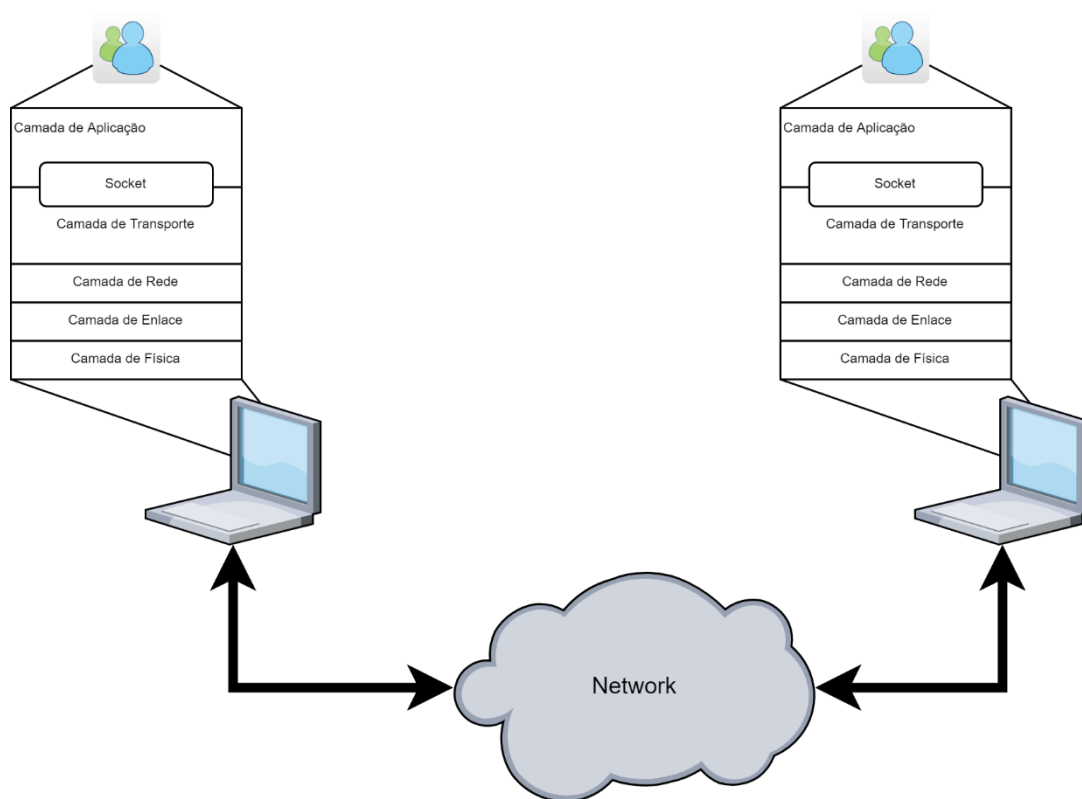
### **Instruções**

---

1. Esta avaliação deve ser feita individualmente ou em até 3 pessoas.
2. Data de entrega: **25/10/2022 até 18:59**. Não serão aceitos trabalhos em atraso.
3. Esta avaliação tem por objetivo consolidar o aprendizado sobre algoritmos usados para a construção da camada de transporte.
4. O uso de bibliotecas de terceiros deverá ser explicado com uma justificativa válida que ficará ao critério do professor aceitar ou não.
5. Poderá ser feito nas linguagens C/C++, Rust, Python, Java, C#, Javascript, Kotlin, Flutter ou a seu critério. Abstrações por meio de bibliotecas poderão ser feitas mediante autorização do professor. A disponibilidade/existência de sockets para uso de TCP/UDP na linguagem escolhida é de responsabilidade do aluno(s).
6. O sistema deve ser entregue funcionando corretamente.
7. Cópias de outros alunos ou da internet, mesmo que parcial, quando detectada implicará em nota zero para todos os envolvidos.
8. Deve ser apresentado um relatório eletrônico em formato PDF (não será aceito em outro formato e a nota relativa a esta parte será zero) que contenha:
  - a. Identificação do autor e do trabalho.
  - b. Enunciado do projeto.
  - c. Explicação e contexto da aplicação para compreensão do problema tratado pela solução.
  - d. Resultados obtidos com as simulações.
  - e. Códigos importantes da implementação.
  - f. Resultados obtidos com a implementação (tabelas, gráficos e etc).
  - g. Análise e discussão sobre os resultados finais.
9. Deve ser disponibilizado os códigos da implementação juntamente com o relatório. O código deverá ser feito entregue via repositório no Github (Gitlab). O repositório deve ser fechado e deve ser aberto somente no dia da entrega (o(s) aluno(s) podem optar por adicionar o professor no Github – Vielf). Também deve ser apresentado o trabalho em aula para o professor, apresentando todo o código e compilando no memento da apresentação. Na apresentação do trabalho não é necessário utilizar slides, apenas apresentar o código e sua execução (compilar na apresentação).

### (7,5) Projeto 1

Nesse projeto, como forma a estimular a integração dos conceitos da M1 e do conteúdo relativo à camada de transporte, você(s) irá(ão) desenvolver um cliente para chat de mensagens (podem chamar de “MRC1”). Para o trabalho, vocês deverão desenvolver essa aplicação que permita que pelo menos dois usuários conversem (troquem mensagens). Os dados a serem trocados podem se resumir a apenas texto (podem expandir para troca de imagens). O cliente poderá usar qualquer dos dois tipos de protocolos de transporte (UDP ou TCP), lembrando que vocês devem justificar suas escolhas, além disso, pode-se utilizar o protocolo QUIC (há o link para uma biblioteca Python em Materiais Extras).



Requisitos para o desenvolvimento:

- Para a comunicação entre os dois clientes, pode usar: localhost, Máquina Virtual (ex: VirtualBox), rede local (se possuir dois computadores na casa).
- Deve ser permitido identificar o usuário, além do uso da porta que é usada. Não há um requisito de porta, mas você deve justificar a escolha do número usado.
- Use um marcador de hora-minuto que a mensagem foi enviada (pode adicionar segundo).
- Ao receber uma mensagem, você deve exibir o nome do usuário e o momento em que a mensagem foi recebida.
- Os dois clientes devem conseguir enviar mensagens no momento que quiserem, não sendo necessário esperar uma iteração de um iniciador.

- O uso do protocolo de transporte fica a critério do desenvolvedor, mas a escolha deve ser justificada. Podem usar variações de protocolo, passando parte do QoS (Quality of Service) para a camada de aplicação (ex: QUIC).
- Deve-se adicionar um dos dois aspectos (ou os dois) na implementação:
  - Segurança – criptografia ou técnica similar
  - Notificação de entrega e leitura

## **(2,5) Projeto 2**

Nessa segunda parte, você deve expandir o conceito para 3 clientes (2 clientes no mesmo host) e utilizar os recursos para que mais de um processo utilize a mesma porta. Para isso você pode usar um dos dois comandos:

- `SO_REUSEPORT` ou `SO_REUSEADDR` (ou equivalente na linguagem utilizada)

Pesquise sobre a utilização dos mesmos e como usar. Caso a linguagem/sistema operacional que você use não ofereça suporte, você deve dissertar os motivos que levaram a essa questão. Caso contrário, descreva os resultados obtidos com os experimentos.

## **Observação para os dois projetos**

Em ambos os projetos, você deve usar o Wireshark para avaliar a comunicação dos seus clientes. Um roteiro e explicação sobre como usar o Wireshark para TCP/UDP pode ser encontrado na pasta Roteiros no material da disciplina.