MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
Curso de Bacharel Sistemas de Informação

REDES DE COMPUTADORES II

Segunda Avaliação de Redes de Computadores II de 2025-2: Valor 10 pontos.

Objetivo

Implementar um servidor web sequencial e um concorrente a fim de ressaltar pontos de vantagem e desvantagem de cada abordagem.

Objetivos específicos:

- Preparar ambiente de teste.
- Estruturar mensagens HTTP de requisição e resposta.
- Definir métricas de desempenho.
- Elaborar testes.
- Gerar gráficos e conhecimentos estatísticos.
- Análise e avaliação das ações empregadas.
- Gerar relatórios para avaliação.

Projeto

Consiste no desenvolvimento de um servidor *web*: (a) interativo síncrono e (b) concorrente assíncrono. A implementação deve usar o modelo **sockets** de programação para rede e protocolo **TCP**. Não pode usar nenhuma biblioteca que implementa algum servidor *web* ou paradigma de comunicação de alto nível como Flask, FastAPI ou Django.

- As mensagens entre o cliente e o servidor devem seguir a estrutura do protocolo HTTP. O objetivo do trabalho é avaliar a performance dos dois tipos de servidores para diferentes tipos de comunicação.
- Faz parte da avaliação a descoberta de quais são os tipos de cenários em que um servidor é melhor do que o outro.

- O aluno deve simular a rede através do uso do *Docker*. Os contêineres devem ser instâncias da imagem Ubuntu.
- As avaliações dos testes devem ser ilustrada através de gráficos. Gere média e desvio padrões para cada teste e no mínimo 10 execuções para cada avaliação. As métricas da quantificação da qualidade dos dos tipos de servidores faz parte da descoberta que cada aluno deve elaborar/projetar.
- Defina que o endereçamento IP dos hosts no ambiente Docker deve ser baseado nos quatro últimos algarismos da matrícula. P.ex., seja a matrícula 20219015499, então os quatro últimos são 5499, logo a subrede deve ser 54.99.00.01, 54.99.00.02, …, 54.99.00.254.
- Inclusão de Cabeçalho HTTP Único e Calculado: A mensagem HTTP deve incluir no cabeçalho da requisição o campo personalizado e obrigatório, **X-Custom-ID: [VALOR]**. Este valor deve ser o resultado de uma função criptográfica simples (como um Hash MD5 ou SHA-1) calculada sobre a matrícula, espaço, e o nome do aluno. Utilizar alguma biblioteca de criptografia para esta função.
- As primitivas em que o servidor deve responder será definida pelo projetista (aluno). Cabe ao aluno definir de acordo com seus objetivos quais primitivas o servidor deverá responder (GET, POST, DELETE, PUT, etc..etc.)

Tecnologias obrigatórias

- **Python** para o desenvolvimento da lógica dos servidores e clientes.
- Mensagens **HTTP**, protocolo **TCP** e IP.
- **Docker** para criar e simular os diferentes elementos da rede (cada *host* deve rodar em seu próprio *container*)
- Uso de **Threads** ou **Multiprocess** para implementação de paralelismo.

Critérios de Avaliação (10 Pontos)

Critério	Descrição	Pontos

1. Estrutura da Rede em Docker	Criação correta de containers para cada cliente e servidor, com subredes (seguir a numeração da matrícula) corretamente configuradas com DOCKERFILE.	**0.5**
2. Mensagens estruturadas segundo o protocolo HTTP.	As requisições e respostas seguem a estrutura de mensagens do HTTP. Cada mensagem possui o **X-Custom-ID** explicado no projeto.	**0.5**
3. Configuração da rede.	Demonstração da configuração da rede e checagem da comunicação entre o/os cliente(s) e servidor. *O IP segue o rigor da definição explicada no projeto.*	**0.5**
4. Implementação do servidor Sequencial.	Servidor sequencial aceita e valida mensagens HTTP na porta 80 e responde de acordo com as primitivas das requisições no formato HTTP.	**1.0**
5. Implementação do servidor Concorrente	Servidor concorrente aceita e valida mensagens HTTP na porta 80 e responde de acordo com as primitivas das requisições no formato HTTP.	**1.5**
6. Métricas	Elaboração e definição com o formalismo matemático para avaliação das abordagens.	**1.0**
7. Abordagem de avaliação	Descrição, justificativa e aplicação de como as avaliações serão aplicadas, garantindo a re-aplicabilidade dos testes.	**1.0**
8. Teste	Apresentação, com todo rigor estatístico, dos resultados, uso de tabelas e gráficos adequados.	**2.0**
9. Relatório	Documentação do projeto, incluindo todos os pontos destacados nas seções deste projeto.	**1.0**

10. Vídeo	Vídeo de 15 a 20 minutos explicando as decisões tomadas para concretização do projeto ressaltando: (a) a mensagens; (b) modelo de interação; (c) mensuradores e métricas; (d) testes; (resultados); (e) conclusão.	**1.0**

Entrega Esperada

- Código-fonte completo em Python (No GITHUB)
- Dockerfile(s) e docker-compose.yml
- README.md explicando:
    - Como executar o projeto
    - Link do GITHUB (se por algum motivo não puder usar o GITHUB, então do Google Drive). Lembre-se de compartilhar: raynergomes@gmail.com e rayner@ufpi.edu.br.
    - Link do Youtube.
- Vídeo demonstrando o funcionamento segundo os critérios de avaliação.
- Relatório com explicação do projeto.
    - Seguir o modelo de artigo do SBC.
    - Descrever as métricas
    - Descrever os testes
    - Descrever os resultados
    - Considerações finais sobre os resultados
- Respostas:
    - Quais pontos que o servidor sequencial é melhor? Por que?
    - Quais pontos que o servidor concorrente é melhor? Por que?
    - Qual a vantagem e desvantagem de sua abordagem?
- **Data**: 29/10/2025

- Trabalho individual
- Agendamento de apresentações se necessário conforme o professor.
    - Trabalhos idênticos ou muito similares
    - Falhas na execução
    - Explicação comprometida
    - Detecção de IA.

Penalidades

A seguir segue uma lista de itens que anulará as avaliações enviadas:

- Sem o relatório PDF.
- Relatório sem nome do aluno.
- Sem o *link* do GitHub ou Google Drive.
- Código idêntico já enviado.
- Sem vídeo.
- Não desenvolvido em Python.
- Comunicação entre o cliente e servidor sem ser por Socket.
- Uso de *frameworks* ou bibliotecas para paralelismo e concorrência (p.ex. Dark)
- Código detectado por ferramentas de plágio e Código Gerado por IA (Explicações após agendamento)

Boa sorte!