

**Desenvolvimento de um Sistema Distribuído com Comunicação Cliente-Servidor
usando Sockets e RPC/gRPC**

Objetivo:

- Aplicar os conceitos de sistemas distribuídos, modelo cliente-servidor, e middlewares (RPC/gRPC) para implementar um sistema que simula um serviço de consulta a um catálogo de produtos distribuído.

Cenário:

Você foi contratado por uma empresa para desenvolver um sistema distribuído que permita consultar informações de produtos armazenados em um servidor. O sistema deve atender os seguintes requisitos:

1. Modelo Cliente-Servidor:

- O servidor armazena um catálogo de produtos (nome, preço, e quantidade em estoque).
- O cliente pode realizar as seguintes operações:
 - Consultar produtos pelo nome.
 - Consultar produtos com preço inferior a um valor fornecido.
 - Atualizar o estoque de um produto específico.

2. Sockets ou RPC/gRPC:

- Utilize **Sockets** para estabelecer a comunicação entre cliente e servidor ou **RPC/gRPC** para realizar chamadas remotas.
- O middleware escolhido deve ser configurado adequadamente para garantir a interação entre cliente e servidor.

3. Execução:

- O sistema deve ser implementado em Python, Java ou PHP.
- O cliente deve ser capaz de se conectar remotamente ao servidor, realizar as operações e receber as respostas.

Tarefas a Realizar:

Parte 1: Em Laboratório (Hoje)

- **Desenvolver com Sockets:**
 - Crie o servidor que gerencia o catálogo de produtos (pode ser armazenado em uma estrutura como lista ou dicionário).
 - Implemente o cliente que envia solicitações ao servidor (por exemplo, consultar produtos e atualizar estoque).
 - Teste localmente a comunicação entre cliente e servidor.

Parte 2: Em Casa

- **Implementar com RPC/gRPC:**
 - Transforme o sistema desenvolvido em laboratório para usar RPC/gRPC.
 - Defina o **.proto** com as mensagens e serviços necessários para implementar as operações de consulta e atualização.
 - Configure o servidor e cliente para realizar chamadas remotas utilizando RPC/gRPC.

Entrega (Até 11/12/2024):

1. Código-fonte do sistema (Sockets e RPC/gRPC).
2. Documentação breve explicando:
 - Como executar o sistema.
 - Comparação entre a abordagem com Sockets e gRPC.
3. **Opcional:** Um diagrama simples que ilustre o fluxo de comunicação entre cliente e servidor.

Critérios de Avaliação (1 Pontos):

- Implementação funcional com Sockets (0,4 pontos).
- Implementação funcional com RPC/gRPC (0,4 pontos).
- Documentação clara e comparativa entre Sockets e RPC/gRPC (0,2 pontos).

Dicas para a Implementação:

- Certifique-se de utilizar portas disponíveis para a comunicação.
- Teste as conexões em diferentes máquinas para simular cenários distribuídos.
- No gRPC, utilize o comando `python -m grpc_tools.protoc` (ou equivalente) para gerar os arquivos de cliente e servidor.

Respostas contendo o código-fonte do sistema e documentação devem ser enviadas via email: felipe_silva@ifba.edu.br **com a devida identificação do aluno, disciplina e turma.**