

# Atividade: Comunicação entre Processos com Sockets em C e Python

**Objetivo:** Os alunos deverão implementar e executar dois sistemas cliente-servidor utilizando sockets, um exemplo totalmente em C e outro em Python. Os sistemas serão testados em um ambiente distribuído, onde o servidor será executado no Google Cloud e o cliente na máquina local.

---

## Descrição da Atividade

### 1. Objetivo Geral

Compreender a comunicação entre processos em sistemas distribuídos, implementando dois sistemas cliente-servidor, um totalmente em C e outro em Python, utilizando sockets e o protocolo TCP/IP.

### 2. Contexto do Ambiente Distribuído

- O servidor será executado em uma máquina virtual (VM) configurada no Google Cloud.
- O cliente será executado localmente na máquina do aluno, conectando-se ao servidor via internet.

### 3. Cenários a Serem Implementados

1. **Exemplo 1:** Sistema cliente-servidor totalmente implementado em **C**.
  2. **Exemplo 2:** Sistema cliente-servidor totalmente implementado em **Python**.
- 

## Novos Requisitos de Alteração do Código

Os alunos deverão modificar o código inicial do cliente e do servidor para incluir a seguinte funcionalidade:

### 1. Cliente:

- O cliente enviará múltiplas mensagens para o servidor, com entradas fornecidas pelo usuário.
- A interação será encerrada quando o usuário digitar a palavra `exit`.
- O cliente exibirá as respostas do servidor no terminal após cada envio.

### 2. Servidor:

- O servidor será adaptado para aceitar e processar múltiplas mensagens do cliente durante a mesma conexão.
  - Quando o cliente enviar `exit`, o servidor encerrará a conexão de forma apropriada.
- 

## Instruções

### 1. Configuração do Ambiente no Google Cloud:

- Configure uma VM no Google Cloud com Linux (ex.: Ubuntu).
- Instale as dependências necessárias:
  - Para o exemplo em C: GCC ou outro compilador C.
  - Para o exemplo em Python: a versão apropriada do Python.
- Configure as regras de firewall para abrir a porta TCP necessária (ex.: 5000) para ambos os exemplos.

### 2. Execução do Código:

- Suba o código original do servidor no Google Cloud e modifique-o conforme os novos requisitos.
- Execute o código do cliente na máquina local com as modificações para enviar múltiplas mensagens.

### 3. Testes:

- Realize testes enviando mensagens variadas ao servidor, incluindo `exit`, para verificar o funcionamento correto.

---

## Requisitos de Entrega

### 1. Documentação:

- Explique as alterações feitas nos códigos do cliente e servidor.
- Detalhe os passos realizados para configurar o ambiente no Google Cloud.
- Apresente logs e/ou capturas de tela da execução do cliente e servidor.

### 2. Teste:

- Demonstre a comunicação entre cliente e servidor, com múltiplas mensagens enviadas e processadas corretamente.

### 3. Análise Crítica:

- Compare as implementações em C e Python, destacando vantagens e desvantagens de cada abordagem.

---

## Critérios de Avaliação

- Funcionamento correto dos dois sistemas cliente-servidor.
- Implementação da funcionalidade de envio e recepção de múltiplas mensagens.
- Capacidade de configurar o ambiente no Google Cloud.
- Qualidade da documentação e análise crítica das implementações.
- Clareza na apresentação dos resultados e logs.

---

## Sugestão para os Alunos

- No **cliente**: Utilize um loop que solicita entradas do usuário e envia cada mensagem ao servidor até que o usuário digite `exit`.
- No **servidor**: Implemente um loop que leia continuamente as mensagens do cliente e responda até receber `exit`.

### **server.c**

*Copie e cole o código aqui:*

### **cliente.c**

*Copie e cole o código aqui:*

### **server.py**

Copie e coleo código aqui:

**server.py**

Copie e coleo código aqui: