

Ciência da Computação – Programação Paralela e Distribuída		
Aluno (a):		
Turma:	Bimestre: 2º	Data da Entrega: 11/06/25 – 16h

Projeto de Computação em Nuvem

Instruções da atividade avaliativa:

- Esta atividade deve ser desenvolvida em até 3 pessoas.
- A atividade deve ser entregue no tempo estabelecido.
- A identificação de **plágio resulta em anulação da prova** (nota 0).
- **Qualquer ferramenta consultada deve ser devidamente referenciada e justificada.**

1. Contextualização

O objetivo deste trabalho prático é aprofundar conceitos-chave de sistemas distribuídos, computação em nuvem e comunicação entre processos, estimulando a aplicação de tecnologias de infraestrutura como código e protocolos de comunicação modernos. Cada grupo deverá projetar, implementar e documentar uma solução que combine ao menos as seguintes temáticas:

- **Arquitetura de Nuvem**
- **AWS CloudFormation**
- **Terraform**
- **gRPC**
- **Middleware (Message Broker ou RPC intermediário)**
- **Sockets (TCP/UDP)**

Observação importante:

- Caso o grupo opte por usar **AWS CloudFormation**, não é obrigatório empregar **Terraform** e vice-versa. O mesmo serve para gRPC, Middleware e Sockets.
- A escolha das tecnologias (por exemplo, CloudFormation vs. Terraform; gRPC vs. Sockets; tipo de middleware) deve estar justificada no relatório.

2. Objetivos Específicos

1. **Projetar uma arquitetura de nuvem** que suporte um sistema distribuído de curto porte (por exemplo, micro-serviços ou aplicação cliente-servidor escalável).
2. **Automatizar o provisionamento da infraestrutura** usando AWS CloudFormation ou Terraform (ou ambos, desde que justificado).

3. **Implementar comunicação entre componentes** de forma eficiente, utilizando gRPC, middleware de mensageria ou sockets nativos, demonstrando paralelismo e/ou concorrência.
4. **Documentar e testar** a solução completa, evidenciando aspectos de escalabilidade, tolerância a falhas e monitoramento (mesmo que simples).

3. Descrição Geral do Trabalho

Cada grupo deverá entregar:

1. **Relatório Técnico** (em PDF ou Markdown) contendo:
 - **Introdução e motivação** do problema escolhido.
 - **Arquitetura proposta** (diagramas de alto nível mostrando VPC, sub-redes, balanceadores, instâncias, serviços gerenciados, etc.).
 - **Justificativa das escolhas tecnológicas**, indicando quais temas da lista foram utilizados e por quê.
 - **Detalhamento da infraestrutura como código** (fragmentos ou explicações de stacks/templates CloudFormation e/ou arquivos Terraform).
 - **Descrição da comunicação entre componentes** (fluxo de chamadas gRPC, configuração de filas ou tópicos de middleware, ou implementação de sockets).
 - **Estratégias de paralelismo e/ou escalabilidade** (por exemplo, múltiplas threads/processos para atender conexões, auto-scaling groups, ou workers concorrentes que consomem mensagens de um broker).
 - **Testes realizados** (casos de uso funcionais, testes de carga ou concorrência, resultados obtidos).
 - **Considerações finais e possíveis melhorias** (o que poderia ser expandido caso houvesse mais tempo).
2. **Repositório de Código-Fonte** hospedado no github contendo:
 - **Código de infraestrutura** (templates YAML/JSON para CloudFormation e/ou arquivos *.tf para Terraform).
 - **Código da aplicação distribuída** (services, scripts de configuração, scripts de deploy, etc.).
 - **Scripts de build/deploy** (por exemplo, Dockerfiles caso usem containers, instruções de compilação, etc.).
 - **Instruções para execução local e em nuvem** (README detalhado explicando como inicializar a infraestrutura, fazer deploy dos serviços e rodar testes).
3. **Apresentação em vídeo** (no máximo 20 minutos) abordando:
 - Visão geral da solução.
 - Demonstração (live ou vídeo) do provisionamento automático e do funcionamento dos serviços.
 - Principais desafios encontrados e lições aprendidas.