

RESPOSTA CHECAGEM_03 - FLASK MONTE CARLO COM DOCKER E KUBERNETES

- **Professor:** Luis Vinicius Costa Silva
- **Aluno:** João Vitor de Souza Gonçalves
- **Data de avaliação:** 23/09/2025

1. Apresentação

Documento Markdown com o objetivo de detalhar cada uma das atividades realizadas para conclusão da Checagem sobre Flask Monte Carlo com Docker e Kubernetes. O objetivo principal da atividade é desenvolver uma aplicação Flask que realiza uma simulação Monte Carlo, empacotá-la em Docker e executá-la em um Cluster Kubernetes local fazendo o uso de Minikube ou cluster local. A aplicação será construída com os seguintes endpoints:

- /docker-info: informações do container Docker;
- /montecarlo/: simulação Monte Carlo em CPU única;
- /montecarlo-distributed/: simulação distribuída em múltiplos pods.

Na atividade deve-se realizar o deploy da aplicação usando Docker e Kubernetes, testar todos os endpoints e modificar o código para adicionar um novo endpoint responsável por executar uma variação da simulação Monte Carlo (exemplo: estimativa de área de uma função ou outra forma geométrica). Por fim, deverá utilizar o script "montecarlo_aggregator.py" para realizar a criação de uma lógica própria para agregação de resultados distribuídos.

- Critérios de avaliação:

- Correto deploy da aplicação em Docker e Kubernetes;
- Funcionamento correto de todos os endpoints;
- Capacidade de modificação do código para criar um novo endpoint funcional;
- Clareza e organização do código e testes;
- Uso correto das ferramentas de container e cluster (Docker, Minikube/Kubernetes).

2. Requisitos

- SO Ubuntu 22.04 (VM com WSL)
- Python (Bibliotecas: os, socket, random, math, jsonify, request)
- Flask
- Docker
- Minikube
- kubectl

3. Passo a passo

3.1 Docker

- *Instalar dependências*

```
sudo apt-get install -y ca-certificates curl gnupg lsb-release
```

- *Chave GPG do Docker*

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

- *Adicionar repositório*

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- *Instalar Docker*

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

3.2 kubectl

- *Instalação kubectl*

```
curl -LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

3.3 Minikube

- *Instalar binário do Minikube*

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

3.4 Startar cluster

- *Iniciar cluster*

```
minikube start --driver=docker  
eval $(minikube docker-env)  
docker build -t flask-montecarlo:latest .
```

3.5 YAML

- *Usar código .yaml*

```
kubectl apply -f kube-flask-montecarlo.yaml
```

3.6 Iniciar a aplicação

- *Dar início na aplicação*

```
minikube service flask-montecarlo-service --url
```

4. Resolução de erros

Um dos problemas notáveis foi o mal funcionamento de uma das rotas, sendo ela `"/montecarlo-distributed/"`. Para o problema em questão não foi encontrada uma solução exata para o erro apresentado, porém sabe-se que o problema persiste por causa de uma das configurações realizadas pelo arquivo `.yaml` original enviado junto com a atividade.

5. Conclusão Final

O trabalho abordou diversos assuntos referentes ao desenvolvimento de aplicações com o uso de Docker e Kubernetes. Foi realizada uma simulação prática de como os conceitos de containers e sua orquestração funcionam e podem ser aplicados em um cenário específico, como é o caso de Monte Carlo.

Apesar de conseguir criar e executar o Docker, aplicar o deployment no Kubernetes e testar os endpoints tanto localmente quanto no Minikube, alguns problemas técnicos ainda precisam ser resolvidos, especialmente em relação à execução dos testes e à distribuição dos pods no Kubernetes.

LINKS:

- Caminho repositório geral:

https://github.com/LuisVCSilva/clusterizacao_servidores_2025

- Caminho repositório da checagem:

[https://github.com/LuisVCSilva/clusterizacao_servidores_2025/tree/master/Ciclo 2/checagem 3/Kubernetes checagem 3](https://github.com/LuisVCSilva/clusterizacao_servidores_2025/tree/master/Ciclo%202/checagem_3/Kubernetes_checagem_3)