DESAFIO PRÁTICO #101

C++, gstreamer, tensorRT, CUDA, UI, GIT, docker

Queremos construir um container docker: ubuntu 20.04, que contenha uma aplicação usando gstreamer que recebe imagens desde uma câmera web.

- A aplicação gstreamer deve ter um source desde uma câmera e um sink para visualização.
- A rede Yolo para detecção de pessoas e outros objetos está amplamente usada em muitos sistemas computacionais, nós queremos usar uma rede Yolo v5 desenvolvida para o dataset COCO (https://cocodataset.org/) tal red tem sido tratada e já construída em https://github.com/ultralytics/yolov5, um cojunto de pesos treinados de tal rede está disponível em: https://download.stereolabs.com/sample_custom_objects/yolov5s_v6.0.wts.zip
- Para este desafio pode usar qualquer rede de tamanho S que use COCO.
- É necessário o uso de GIT para Taguear o desenvolvimento do desafio, e é necessário marcar os níveis alcançados com tags no GIT.
- Deve ser entregue um arquivo Zip contendo todo o projeto desenvolvido.
- Deve ser entregue uma documentação em pdf, breve mas esclarecedora de cada parte desenvolvida. A documentação deve incluir testes em cada nível com capturas de tela explicadas, e mostrar a evolução do projeto tagueado por níveis, assim como os commits realizados.

Níveis do desafio

- **Nível Zero:** Deve ser criado um "**container docker**":
  - Ubuntu 20.04
  - Suporte gstreamer
  - Preparado para usar tensorRT com CUDA dentro do "**container docker**"
  - Arquivos Dockerfile e docker-compose.yml são os entregáveis desse nível.
- **Primeiro Nível:** Desenvolver uma pipeline gstreamer dentro do "**container docker**":
  - Deve estar completamente em C++
  - Deve ler os dados desde uma câmera
  - Deve mostrar uma saída gráfica das imagens da câmera em tempo real.
- **Segundo Nível:** Incrementar um módulo na pipeline gstreamer (C++) previa:
  - Para lançar uma rede COCO de tamanho S, usando usando tensorRT e CUDA da máquina,
  - Mostrar as inferências em texto no terminal (bound boxes).
- **Terceiro Nível:** Desenvolver um pequeno controlador em C++ dentro do "**container docker**" que permita:
  - Iniciar e parar a pipeline gstreamer.
  - Trocar a rede COCO.
- **Quarto Nível:**
  - Desenvolver uma interfaz de usuario em C++ dentro do "**container docker**" que permita manipular o funcionamento da pipeline gstreamer:
    - Iniciar e parar a pipeline gstreamer.
    - Trocar a rede de inferência.
    - Incrementar/diminuir brilho/contraste na imagem mostrada em tempo real.
    - Desenhar os bound boxes em tempo real.
