UFPI – CCN – DC
Arquitetura de Computadores 2025.1
Professor: Ivan Saraiva Silva
Grupo: Davi Soares de Macedo & Enrico da Rocha Santos Teixeira
Maio de 2025

Introdução

Este trabalho propõe a implementação de códigos para cálculo de histograma para imagens no formato QCIF em dois níveis de abstração: alto nível, utilizando Python, e baixo nível, utilizando Assembly RISC-V no simulador RARS. O objetivo é demonstrar as etapas de desenvolvimento de cada solução, detalhar as metodologias adotadas e, ao final, comparar a qualidade dos histogramas gerados e o desempenho de execução em ambas as abordagens.

Metodologia

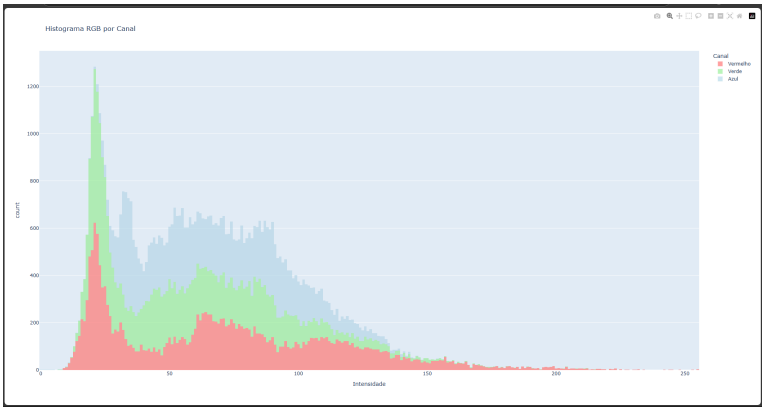
Na implementação em alto nível foi desenvolvido o script *histograma.py*. Inicialmente, o código lê a imagem QCIF, calcula os histogramas dos canais R, G e B, aplica equalização e salva os valores obtidos em um arquivo texto na pasta *result_python*. Em seguida, gera e exibe um gráfico contendo a distribuição de pixels para cada canal.

Para a versão em RISC-V, a imagem QCIF foi convertida para o formato bruto (.raw) e processada pelo programa *riscv1.asm* no simulador RARS via linha de comando. O Assembly lê byte a byte, conta as intensidades de cada canal e grava o resultado em *out_rgb_riscv.txt*. Por fim, um pequeno script Python lê esse arquivo e plota gráficos comparativos dos histogramas para os 3 canais RGB..

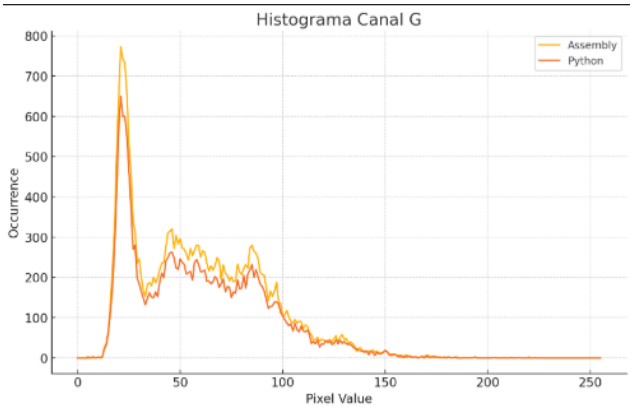
Resultados e Análise

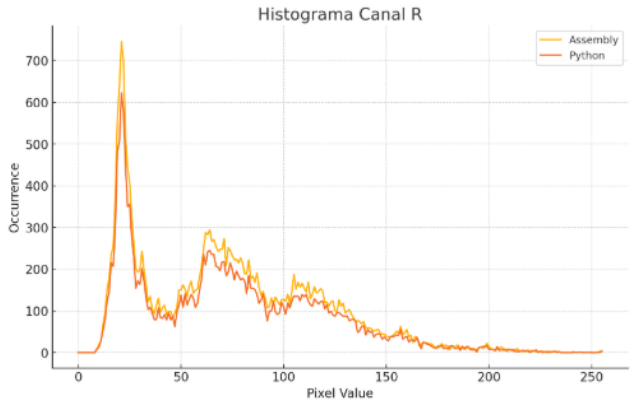
Os histogramas obtidos com ambas as implementações apresentaram curvas praticamente coincidentes, com algumas pequenas diferenças em seus valores.

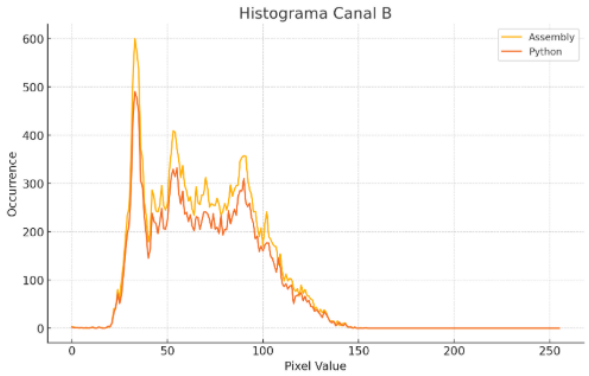
Comparativo entre os pixels dos 3 canais em alto nível



Comparativos entre cada canal tanto para alto quanto para baixo nível







Conclusão

Conclui-se que, baseado na experiência e em várias tentativas que , o desempenho, velocidade de processamento, em assembly é melhor que python, no entanto a implementação em alto nível é muito menos trabalhosa.