Universidade Federal do Piauí

Centro de Ciências da Natureza

Departamento de Computação

Disciplina: Arquitetura de computadores

Professor: Ivan Silva Saraiva

Alunos: Davi Sousa Soares e Victor Kauan da Silva Miranda

Conteúdo

1. Introdução

2. Metodologia

2.1. Implementação em Assembly RISC-V

2.2. Implementação em Python

3. Resultados

3.1. Comparação Visual

3.2 Análise

4. Conclusão

## **1. Introdução**

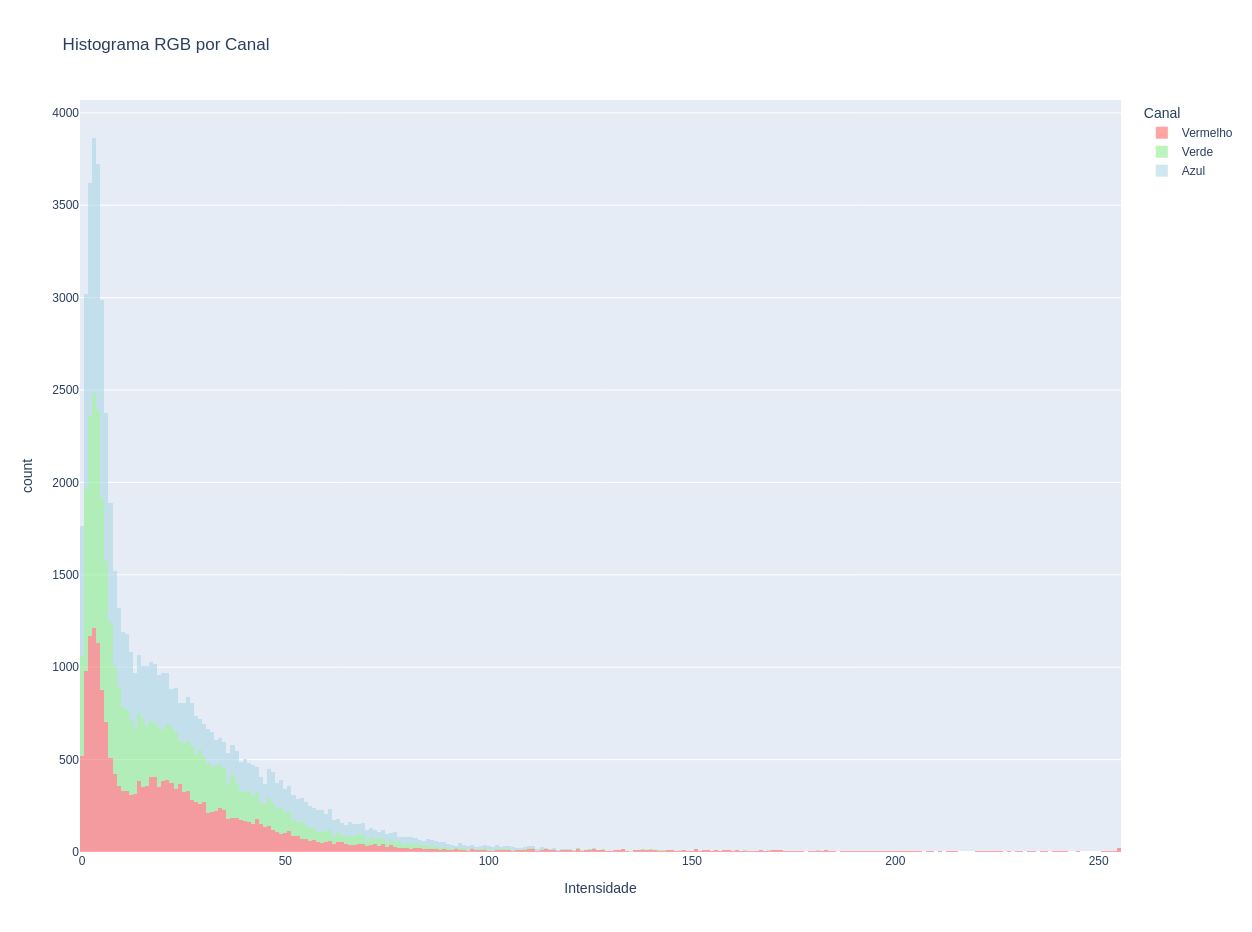
Este trabalho da disciplina arquitetura de computadores consiste em equalizar o histograma dos pixels de uma imagem. A atividade deve ser feita tanto em risc-v utilizando o simulador RARS(Assembler and Runtime Simulator) quanto em uma linguagem de alto nível para comparação de resultados, nesse caso a linguagem de alto nível escolhida foi python usando as bibliotecas plotly, pillow e pandas

Imagem a ser equalizada histograma da imagem feito com plotly

(histograma\_rgb.py)

## **2. Metodologia**

### **2.1 Implementação em RISC-V**

O código “equalizador.asm” consiste em um macro que calcula o histograma de um arquivo binário, equaliza seus valores e armazena o histograma equalizado em um txt no formato “pixel x - ocorrência y” e armazena os novos valores dos pixels em um outro binário binário. Esse marco recebe 3 endereços: o endereço do binário que contém os pixel, o endereço do txt que receberá o histograma equalizador e o endereço do binário onde será escrito os novos valores dos pixels. Com isso o código recebe 3 binários contendo os canais vermelho, verde e azul e cria 3 arquivos txts com seus respectivos histogramas e 3 arquivos binários com seus respectivos pixels equalizados. Todos os binários estão armazenados na pasta “bins”, os que possuem o sufixo “channel” são os binários de entrada e os que possuem o prefixo “pixel\_bytes” são os equalizados. Na pasta “result\_rars” estão os txts gerados pelo rars

### **2.2 Implementação em Python**

O código “bin\_all.py” é responsável por criar o binário de cada canal da imagem para o rars ler, “bin\_to\_jpg.py” pega os binarios equalizados da pasta bin e transforma em uma imagem e usa a função plotly.histogram para calcular o histograma da imagem equalizada e mostrar no navegador, “histograma\_rgb.py” mostra o histograma da imagem original no navegador, “cria\_imagem.py” é um modulo para criar imagens a partir de pixels e “equalizaçã[o.py](http://o.py)” realiza a equalização da imagem, constrói seu histograma, mostra no navegador e salva em um txt na pasta “result\_python”

## **3. resultados**



imagem equalizada via rars histograma da imagem equalizada

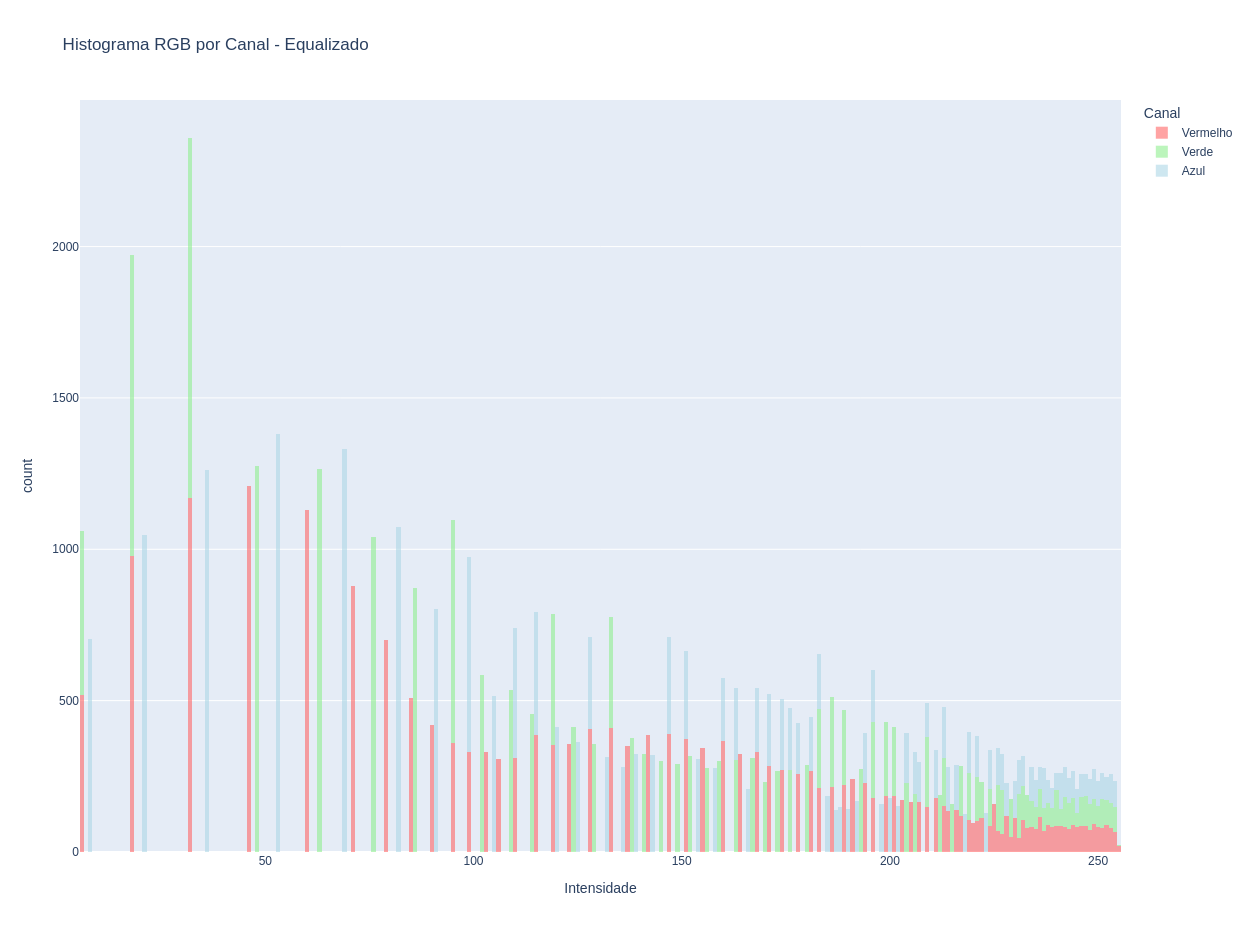


imagem equalizada via python histograma da imagem

## **4. conclusão**

Os resultados idênticos comprovam a precisão do código fonte em assembly para conferir os resultados com mais detalhes é recomendado verificar-los pelo próprio navegador