

# Sistemas Embarcados CSW41

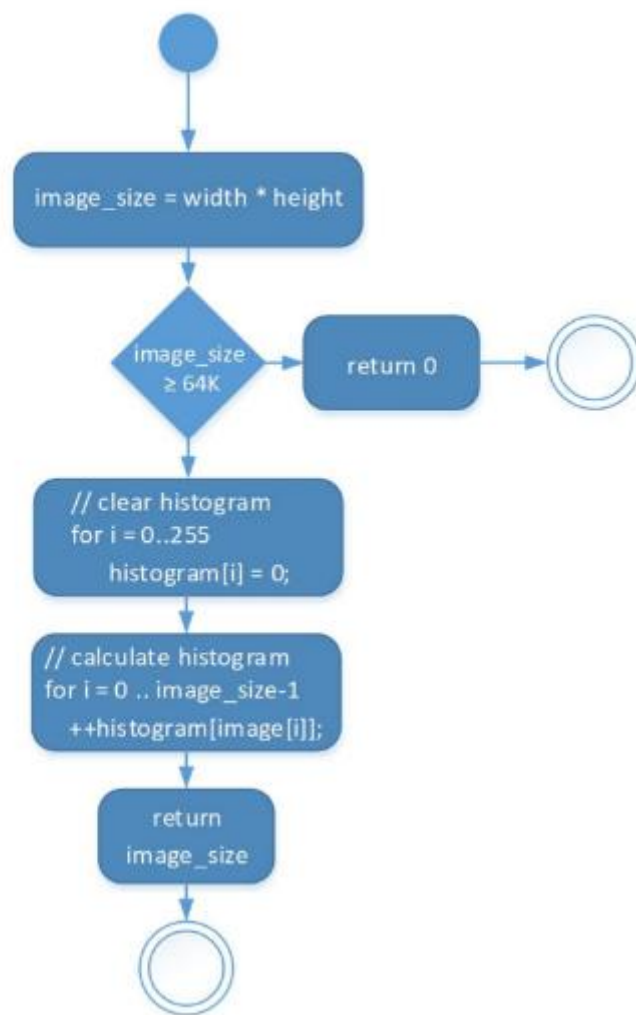
Matheus Mattos

## LAB 3

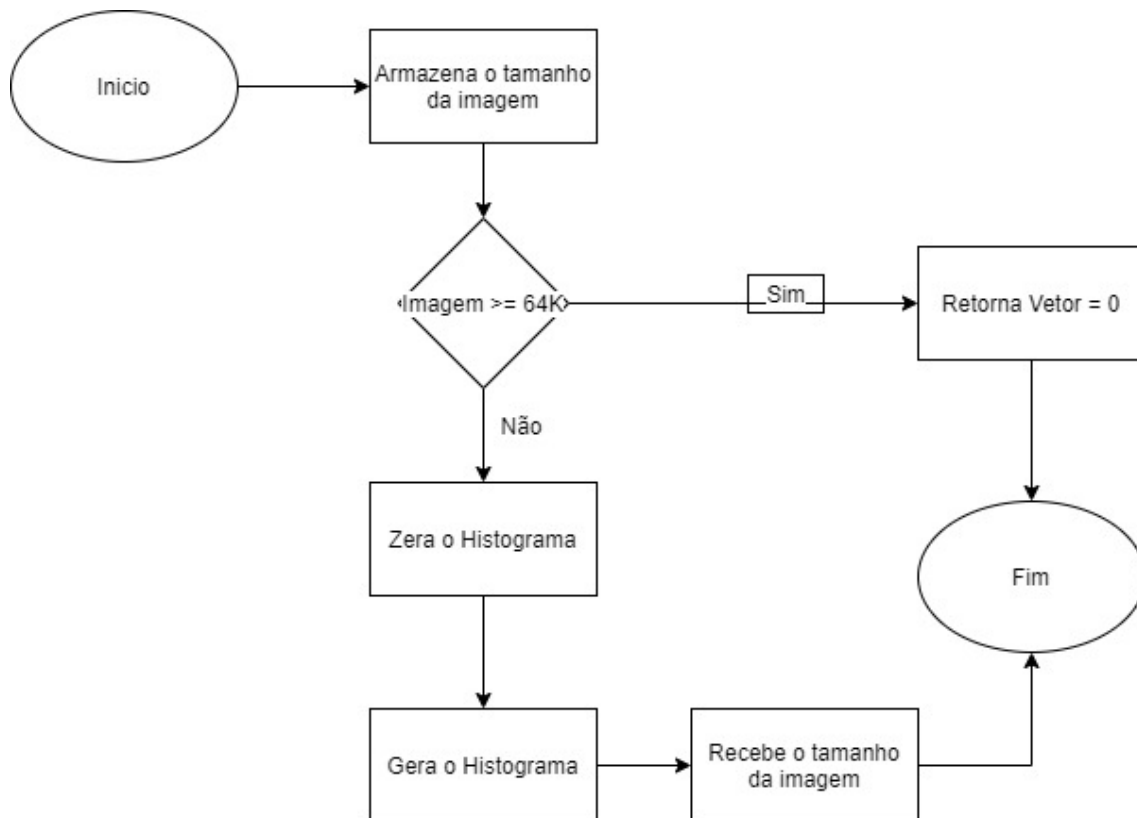
O Laboratório 3 consiste em criar uma solução que ao ler uma imagem em tons de cinza seja possível gerar um histograma, com isso é necessário mapear os pixels em forma de matriz que foi fornecido pelo professor, cada pixel possui um valor entre 0 referente a cor preta e 255 referente a cor branco. A solução deste projeto foi desenvolvida em Assembly e C++.



Para o desenvolvimento inicial do projeto foi feito um estudo do fluxograma fornecido pelo professor, no qual este fluxograma apresentava o funcionamento da solução geral.



Em cima do fluxograma fornecido pelo professor, foi necessário desenvolver a solução em assembly para o problema proposto, ficando desta maneira a solução do fluxograma geral.



Para a criação do código em Assembly foi dividido em três etapas.

## EightBitHistogram

Esta função é responsável por salvar o contexto e verificar se o número de elementos é menor ou igual a 64K, caso o número de elementos exceda o limite permitido, ela pede para o programa retornar zero para o Registrador R0 (indicação de erro). Caso a função esteja dentro do limite permitido de elementos, ela pula para label Zerar, onde ira zerar os vetores e os registradores.

Vale se destacar nesta função os seguintes trechos do código:

- O registrador onde é armazenado o tamanho da imagem.

```
EightBitHistogram
    MUL R6, R0, R1
```

- Instrução condicional de bloco, onde é feito a verificação se a imagem é maior que 64k, caso seja retorna zero.

```
    ITT GT
    MOVGT R0, #0
    BXGT LR
```

- E as chamadas para inserir e remover da pilha.

```
    PUSH {LR}
    BL Zerar
    POP {LR}
    MOV R7, #0
    PUSH {LR}
    BL GerarHistograma
    POP {LR}
```

## Zerar

Percorre todo o vetor e vai salvando 0 na posição da memória em R8, com o registrador R7 começando no zero, cada vez que ele percorre uma posição do vetor ele vai incrementando este registrador até chegar no tamanho final do vetor (256), após feito isso ele zera novamente o registrador R7 e começa a gerar o histograma.

## GerarHistograma

Vale se destacar nesta função os seguintes trechos do código:

- Para gerar o histograma, pega a posição inicial do vetor  
`GerarHistograma`  
`MOV R3, R9`
- Lê o valor da posição atual em que se encontra o vetor  
`LDR R11, [R3, R8]`
- Armazena na memória o valor lido  
`STR R11, [R3]`
- Compara se já chegou na última posição do vetor, se não continua o loop.  
`CMP R7, R6`  
`BNE GerarHistograma`  
`BX LR`

## Terminal I/O

