

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CURSO: ENGENHARIA DA COMPUTAÇÃO**

**DISCIPLINA: ESTRUTURA DE DADOS**  
**PROFESSOR: OBERLAN CHRISTO ROMÃO**

**RELATÓRIO**  
**EP1 – FotoXop**

**FELIPE CAMILO SOARES**  
**JULHO DE 2021**

## INTRODUÇÃO

O relatório a seguir contém exemplos de execução referentes ao Exercício Programa 1, onde usando os conhecimentos adquiridos na disciplina, deveríamos implementar funções responsáveis por operar um Tipo Abstrato de Dados (TAD). Neste Exercício uma imagem no formato ppm é utilizada para execução do programa, e nele devemos aplicar diferentes filtros de imagem. Filtros esses que faziam operações simples e operações complexas para assim, modificar o conteúdo da imagem. É extremamente importante lembrar de fazer as alocações corretamente e consequentemente as desalocações, pois cada função precisa de uma cópia da imagem original para operar, sendo necessário liberar a cópia no fim da função.

## IMPLEMENTAÇÃO

Para a implementação foi necessário estudar cada função separadamente, analisando quais as dependências de cada função para assim completar as funções. Para começar aplicar os filtros deveríamos garantir que as funções de alocação, cópia e liberação estivessem corretamente implementadas. A seguir os detalhes da implementação de cada filtro:

### 1. Escurecer imagem

Para esta opção apenas salve o valor digitado pelo usuário para assim aplicá-lo nos referentes pixels da imagem. Trata-se de uma operação simples, onde cada pixel apenas é subtraído pelo valor digitado. Não podemos esquecer que os valores dos pixels estão limitados entre 0 e 255, e para realizar as alterações foram utilizados operadores ternários em cada pixel.

### 2. Clarear imagem

Trata-se da operação inversa da função anterior, nesse caso o valor salvo é utilizado para acrescentar o valor de cada pixel. Também é importante lembrar que o valor é limitado entre 0 e 255, sendo necessário a utilização de operadores ternários.

### 3. Escala de cinza

Para esta opção os valores dos pixels são calculados a partir da média ponderada dos valores de cada cor de um pixel, sendo 30% de vermelho, 59% de verde e 11% de azul. Dessa forma há uma distinção mais clara em relação às três cores principais.

### 4. Filtro de Sobel

Um dos filtros mais complexos deste programa, é utilizado para destacar os contornos da imagem, sendo necessário fazer operações mais complexas em cada pixel. Nessa implementação foram aplicadas duas matrizes de convolução em cada pixel, onde foi calculada a média entre as duas matrizes e assim aplicando esse valor em cada banda de cor dos pixels.

### 5. Detecção de bordas de Laplace

Trata-se de outra matriz de convolução aplicada em cada banda de cor da cada pixel, onde o resultado é um destaque nas bordas da imagem. A implementação é muito parecida com a do filtro de Sobel, mudando apenas a matriz utilizada.

### 6. Meu Filtro

Para esse filtro utilizei o site [https://docs.gimp.org/2.8/pt\\_BR/plugin-convmatrix.html](https://docs.gimp.org/2.8/pt_BR/plugin-convmatrix.html) como base para reproduzir o filtro. Trata-se de uma matriz de convolução

que destaca relevos na imagem. Essa matriz é aplicada da mesma forma que as anteriores e o resultado é muito bom.

## EXEMPLOS DE EXECUÇÃO

Finalmente chegamos na parte mais legal do código: a execução.  
Vamos testar com a imagem meme.ppm:

C++ developer  
learning Python

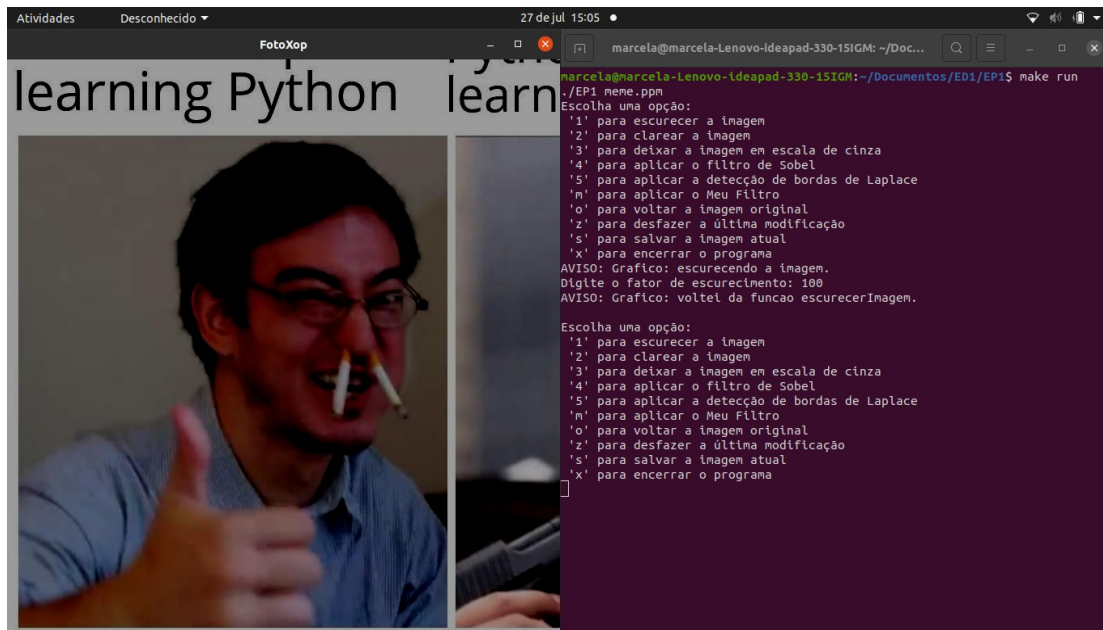
Python developer  
learning C++



Assim que executamos o programa e passamos a imagem como parâmetro temos as seguintes mensagens:

```
Atividades Terminal 27 de jul 15:10
marcela@marcela-Lenovo-ideapad-330-15IGM: ~/Document...
marcela@marcela-Lenovo-ideapad-330-15IGM:~/Documentos/ED1/EP1$ make run
./EP1 meme.ppm
Escolha uma opção:
'1' para escurecer a imagem
'2' para clarear a imagem
'3' para deixar a imagem em escala de cinza
'4' para aplicar o filtro de Sobel
'5' para aplicar a detecção de bordas de Laplace
'm' para aplicar o Meu Filtro
'o' para voltar a imagem original
'z' para desfazer a última modificação
's' para salvar a imagem atual
'x' para encerrar o programa
```

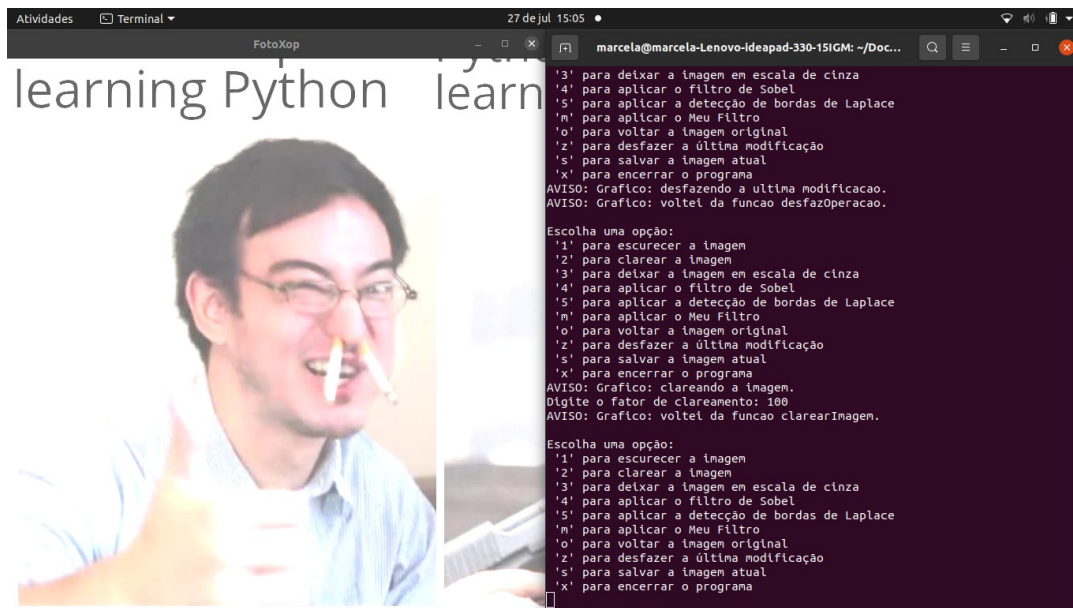
Digamos que queremos escurecer a imagem com um fator 100.



Essa será a visão no terminal e ao lado a exibição da imagem já editada. Veja abaixo:



Após isso vamos desfazer essa alteração, para prosseguir com as próximas.

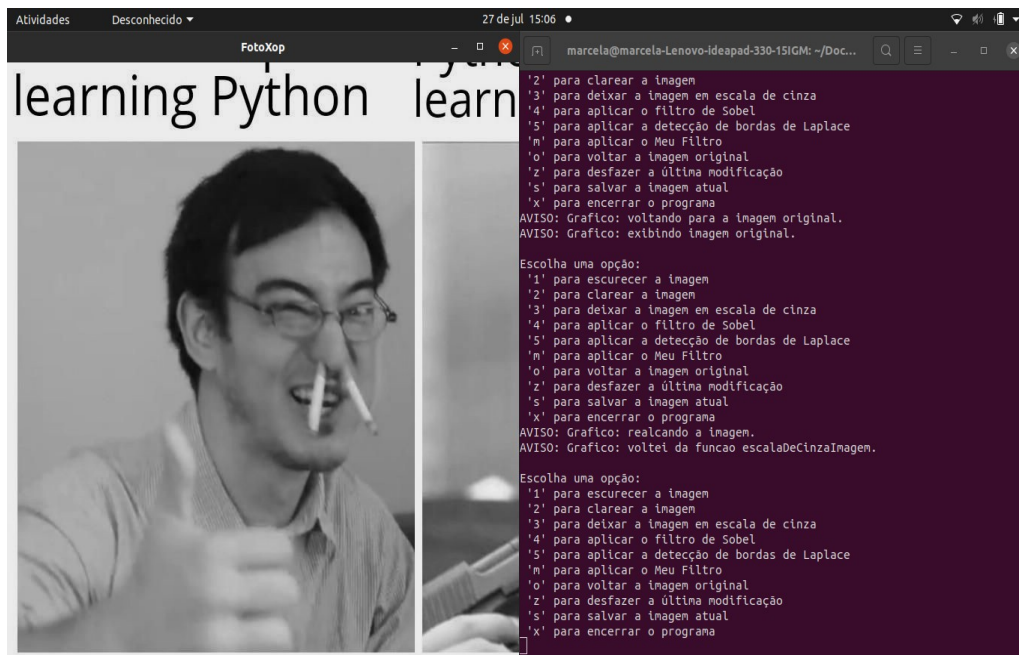


No terminal digitamos 'z' e o programa desfez a última alteração, logo após aplicamos um clareamento de fator 100. Veja o resultado:

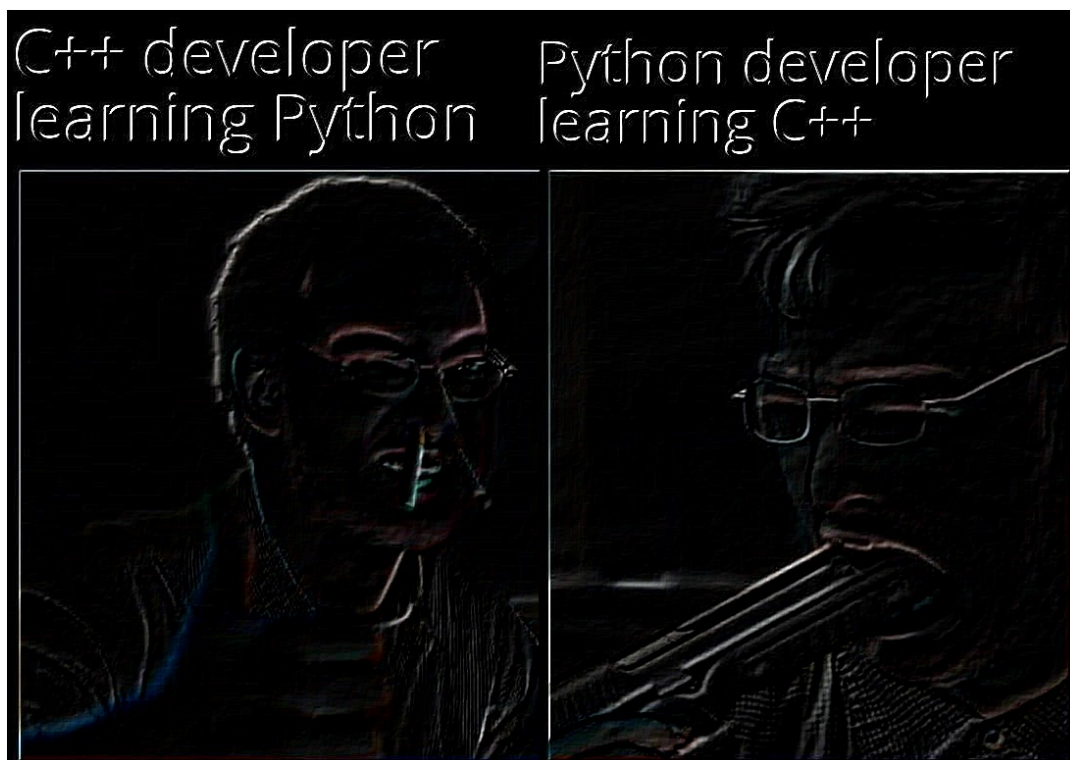


Veja um exemplo para a escala de cinza:

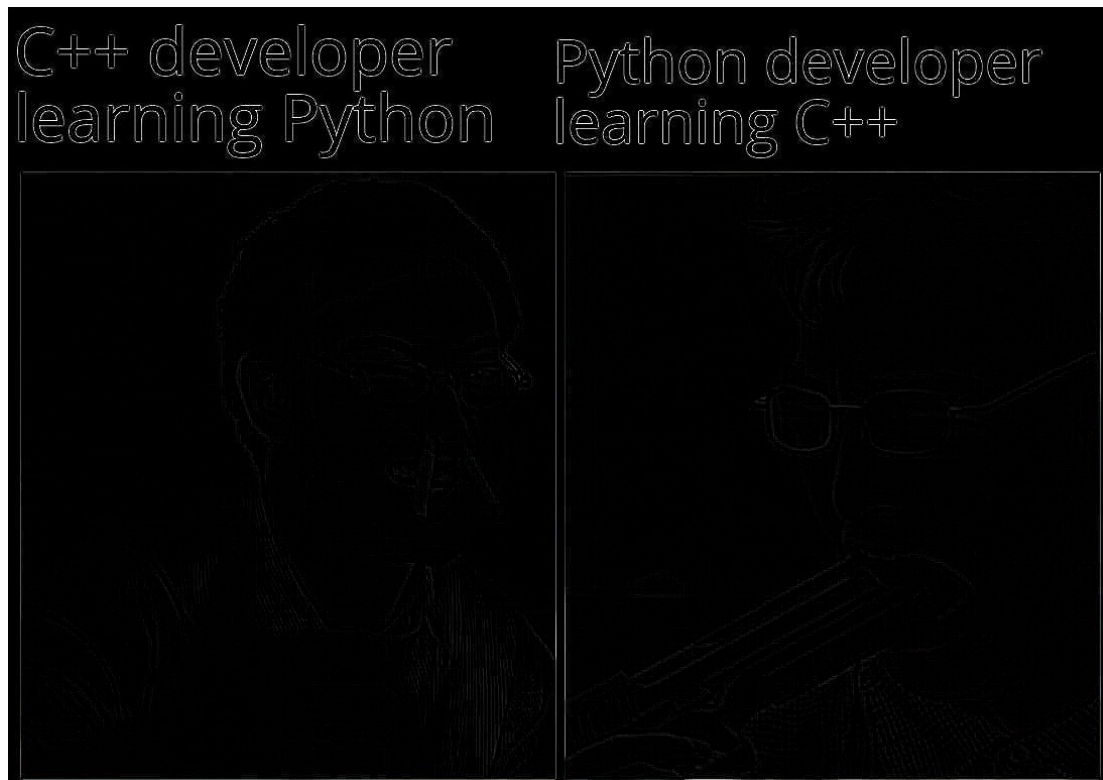




À partir daqui todas as alterações são desfeitas antes de avançarmos para outro filtro. Veja como aplicamos os filtros de Sobel, Laplace e meu próprio filtro.



A imagem acima refere – se ao filtro de Sobel. Note como as bordas são destacadas.



A detecção de bordas de Laplace são um pouco menos visíveis nessa imagem utilizada para os testes.



Acima temos um exemplo da aplicação do meu filtro, onde os relevos são destacados, aumentando um pouco a nitidez da imagem.

## CONSIDERAÇÕES FINAIS

Esse Exercício sem dúvidas pôs à prova meus conhecimentos com TAD e alocação dinâmica de memória. Foi muito bom implementar um programa assim, pois não fazia nem ideia de que os filtros de imagem eram aplicados dessa forma e a experiência adquirida será muito útil para as próximas atividades.