

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO LUIZ GRAVE GROSS
180171

Relatório – Laboratório 5

Trabalho da Disciplina de Fundamentos de
Processamento de Imagens

Prof. Manuel Menezes de Oliveira Neto

Porto Alegre, 7 de dezembro de 2011.

Questão 2. Estas imagens encontram-se representadas no modelo de cores RGB. Assim, cada uma delas corresponde a uma matrix com três dimensões ($N \times M \times 3$), onde N e M representam os números de linhas e colunas, respectivamente, da matrix. Escreva um procedimento para separar cada um dos canais (R, G e B) como imagens independentes:

- (a) Leia a image “Chateau_small.jpg” usando o commando **imread**.
- (b) Obtenha as dimensões da imagem lida utilizando o comando size (e.g., [Rows, Cols, Channels] = **size**(<nome da matrix onde foi lida a imagem>));
- (c) Utilizando dois laços aninhados, salve os conteúdos dos canais R, G e B em três matrizes distintas (com dimensões $N \times M$) chamadas *red*, *green* e *blue*, respectivamente.
- (d) Exiba a imagem original, bem como *red*, *green* e *blue* em uma janela definida pelo comando **subplot(2,2,...)**.

Código em MATLAB:

```
% 2.a - Carregando imagem
PS = imread('Paradise_small.jpg');

% 2.b - Obtendo dimensões da imagem
[h, w, c] = size(PS);

% 2.c - obtendo canais R G B
PS_red      = zeros(h, w, c);
PS_green    = zeros(h, w, c);
PS_blue     = zeros(h, w, c);

for i = 1:h
    for j = 1:w
        PS_red(i,j,1)   = PS(i,j,1);
        PS_green(i,j,2) = PS(i,j,2);
        PS_blue(i,j,3)  = PS(i,j,3);
    end;
end;
```

```
% 2.d - Exibindo imagem original e canais R G B
```

```
figure(1);
```

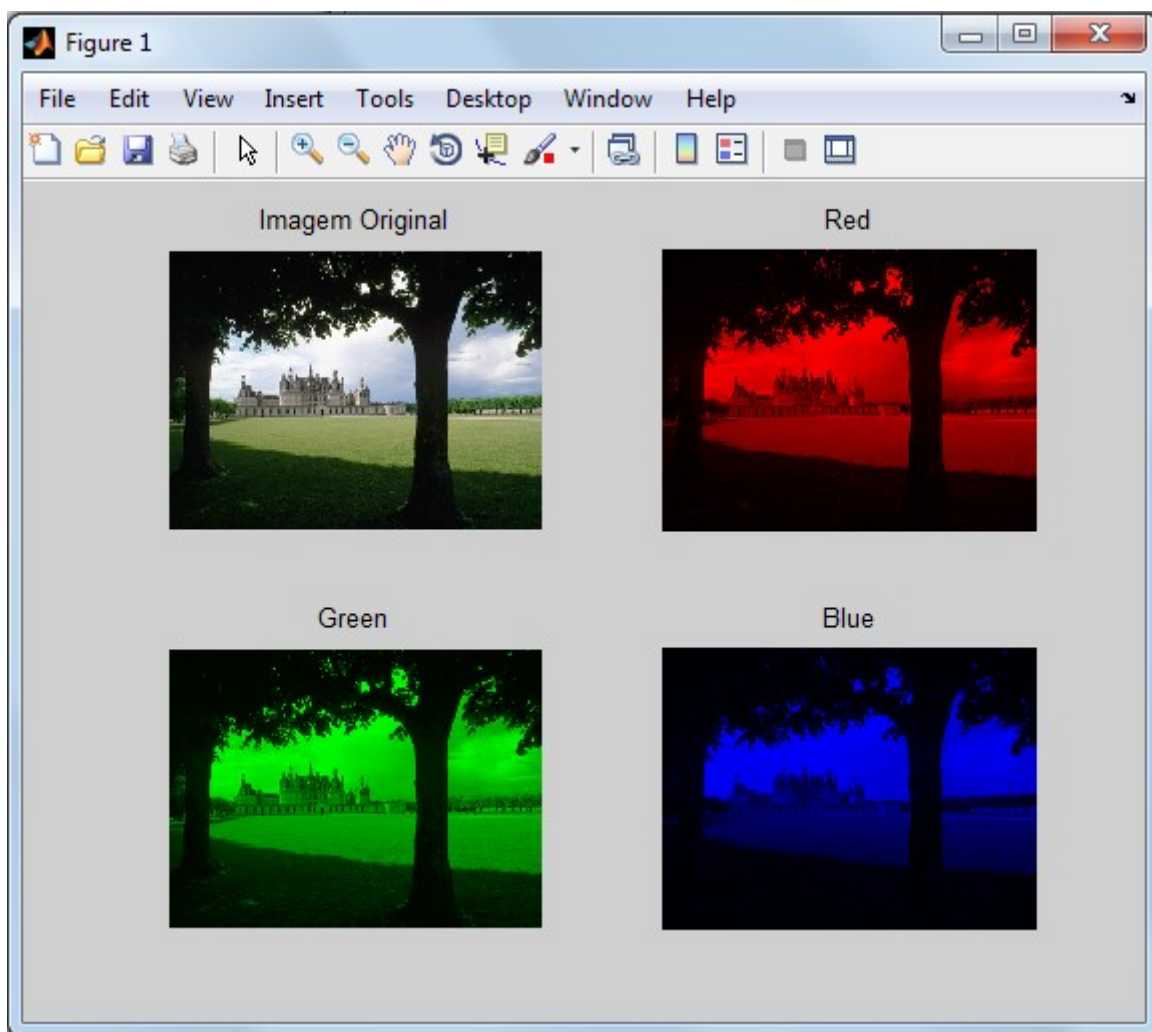
```
subplot(2,2,1); imshow(uint8(PS)); title('Imagem Original');
```

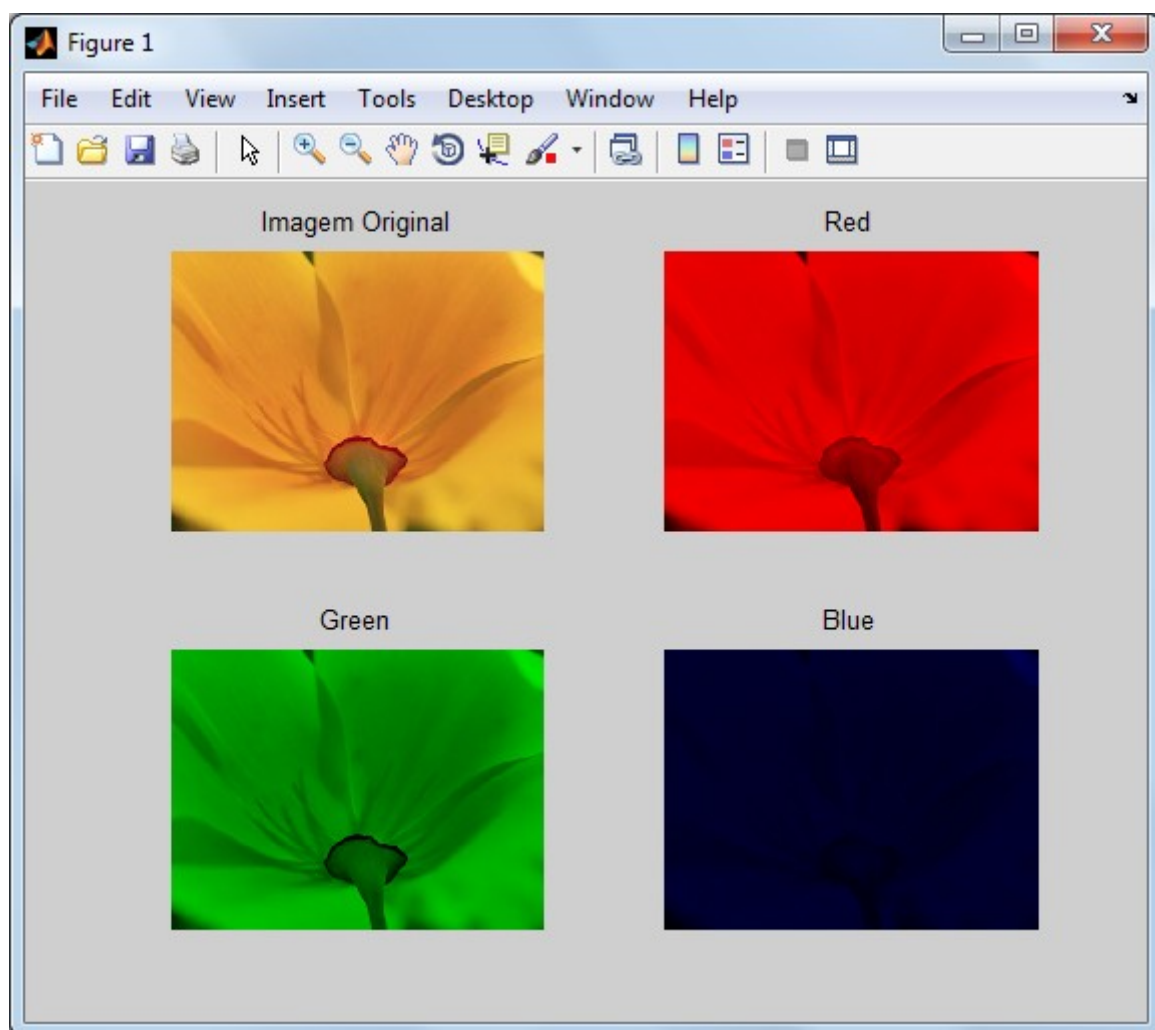
```
subplot(2,2,2); imshow(uint8(PS_red)); title('Red');
```

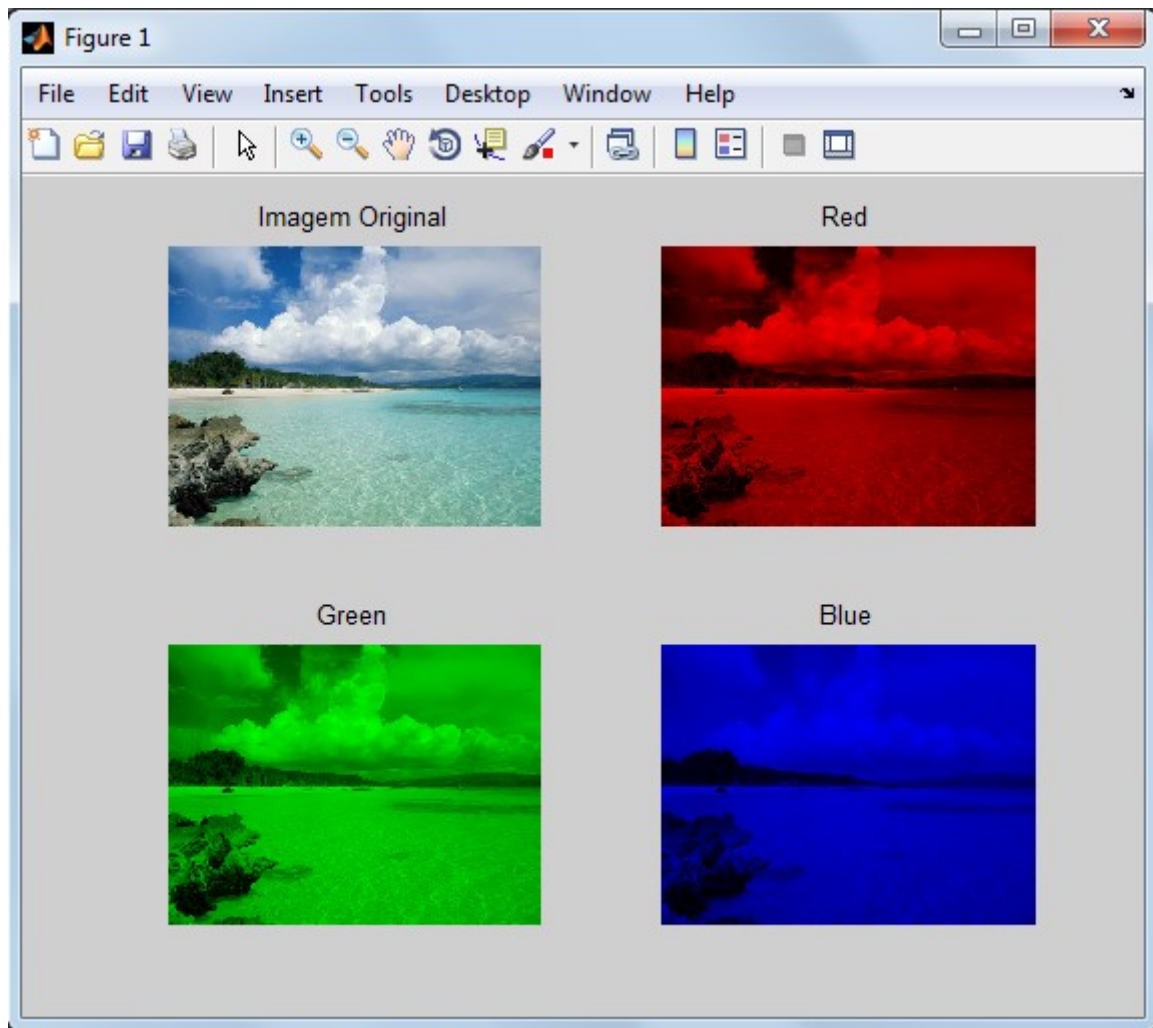
```
subplot(2,2,3); imshow(uint8(PS_green)); title('Green');
```

```
subplot(2,2,4); imshow(uint8(PS_blue)); title('Blue');
```

O trecho acima refere-se à imagem `Paradise_small.jpg`, porém para as demais imagens utilizaram o mesmo código.







Questão 3. Combine, duas a duas, as imagens dos canais R, G e B obtidas anteriormente, criando as imagens red_green (canais R e G, com B = 0), red_blue (canais R e B, com G = 0) e green_blue (canais G e B, com R = 0). Lembre-se que estas imagens terão dimensões $N \times M \times 3$. Inicialize cada uma delas com o comando `uint8(zeros(Rows, Cols, Channels))`. Exiba a imagem original, e as três imagens obtidas por meio destas combinações em uma janela definida pelo comando `subplot(2,2,...)`.

```
% 3 - Combinado os canais dois a dois

PS_red_green    = zeros(h,w,c);
PS_red_blue     = zeros(h,w,c);
PS_green_blue   = zeros(h,w,c);

for i = 1:h
    for j = 1:w
        PS_red_green(i,j,1)    = PS_red(i,j,1);
```

```

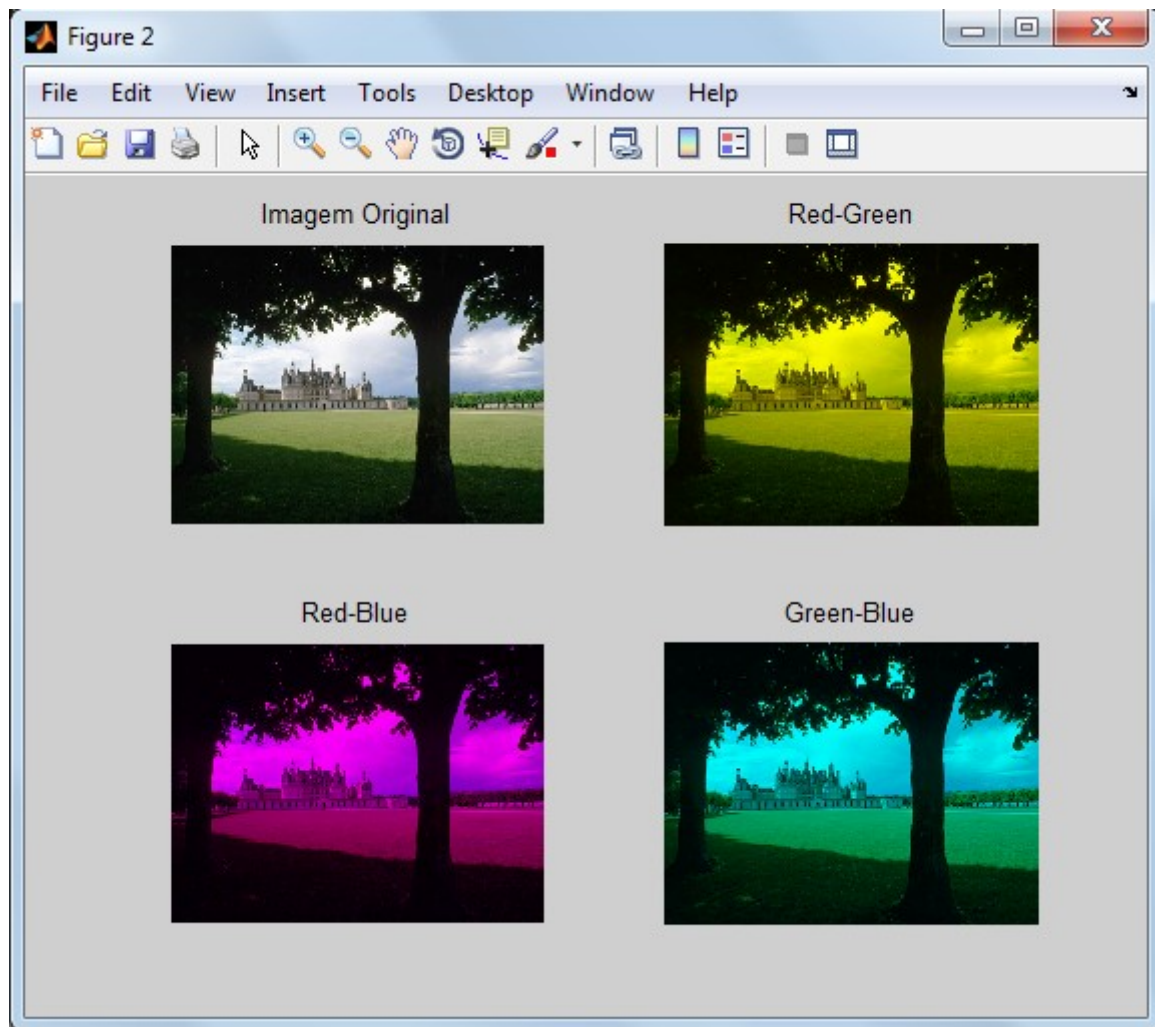
PS_red_green(i,j,2)      = PS_green(i,j,2);
PS_red_blue(i,j,1)       = PS_red(i,j,1);
PS_red_blue(i,j,3)       = PS_blue(i,j,3);
PS_green_blue(i,j,2)     = PS_green(i,j,2);
PS_green_blue(i,j,3)     = PS_blue(i,j,3);

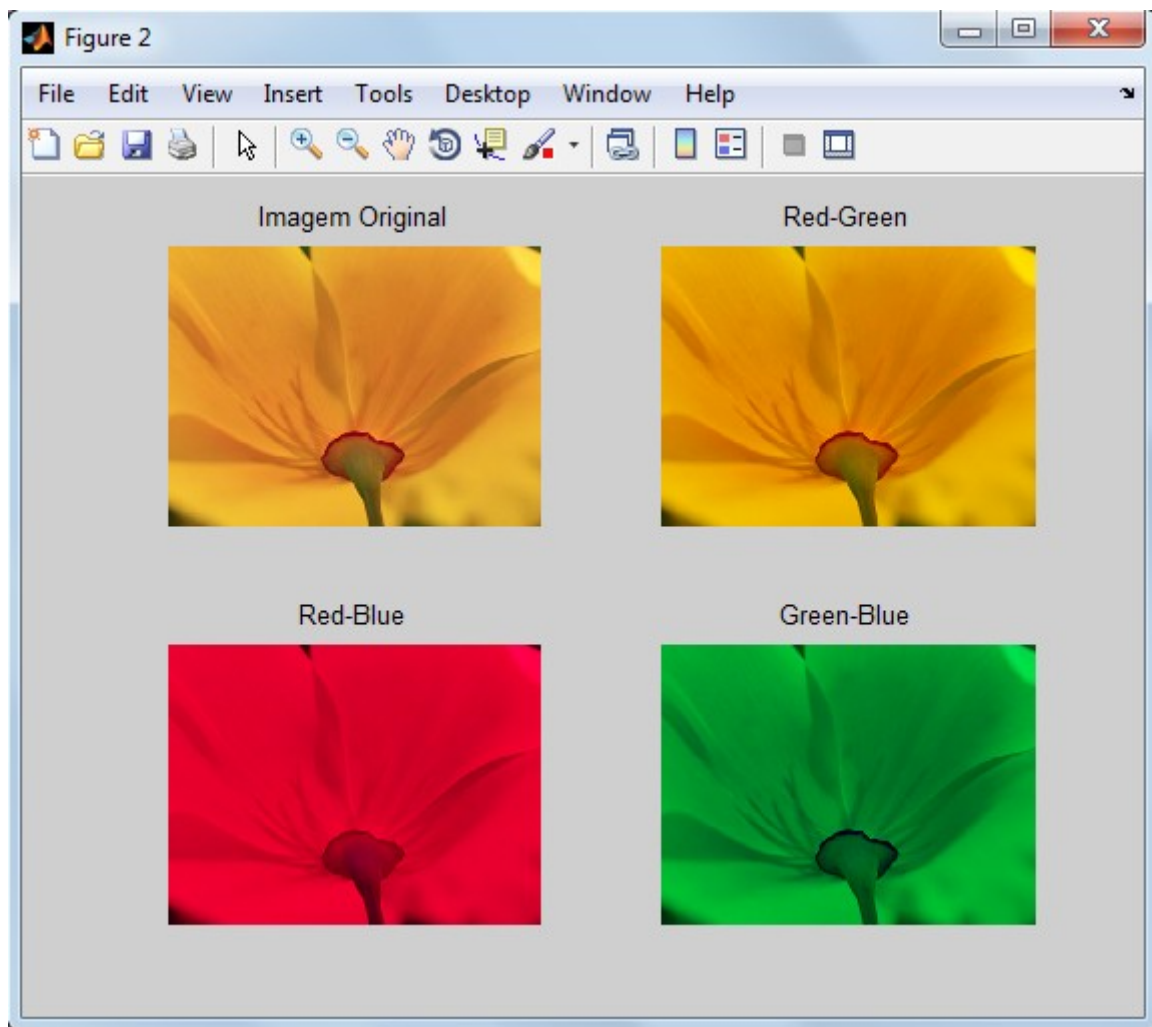
end;

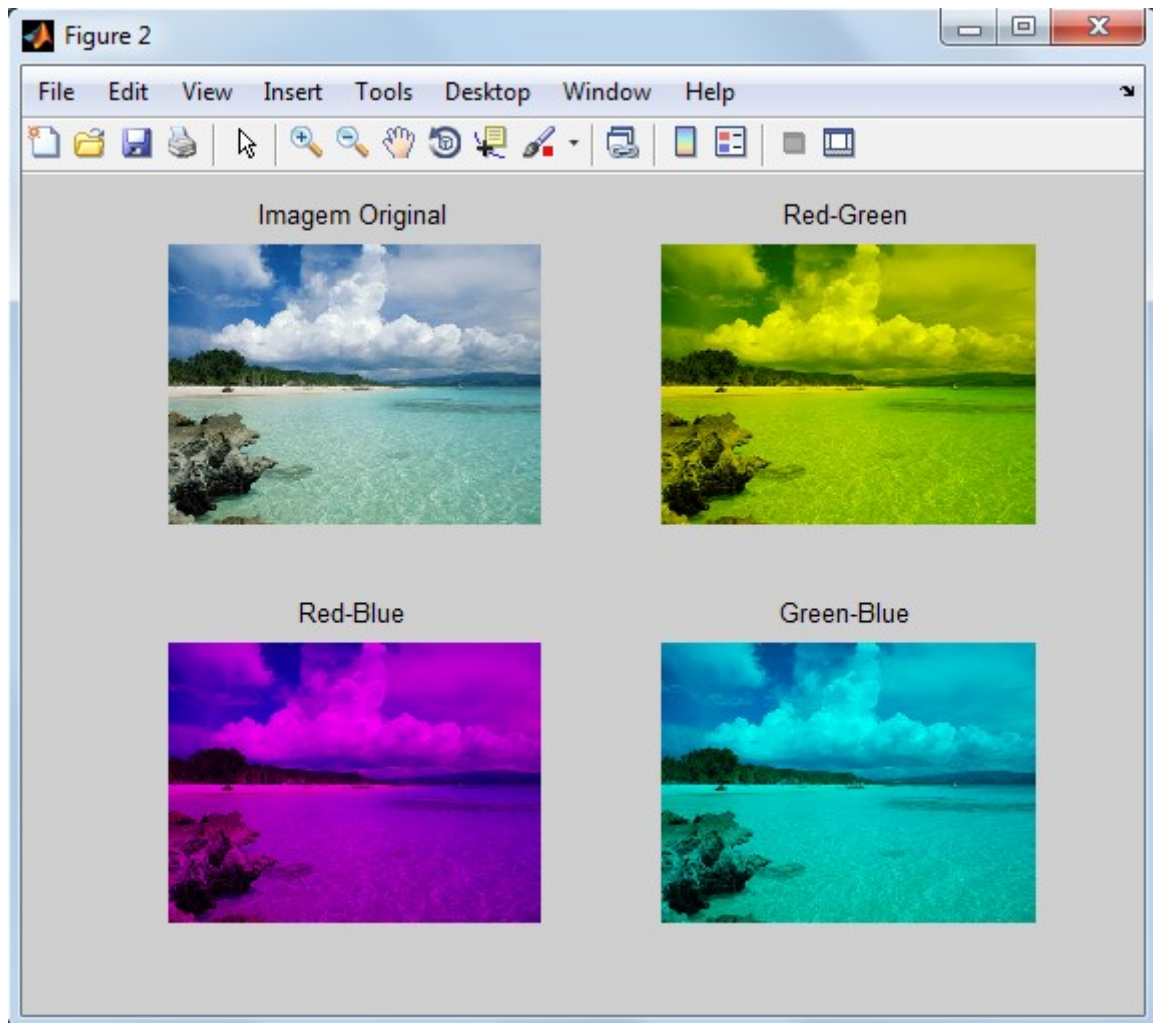
end;

figure(2);
subplot(2,2,1); imshow(uint8(PS)); title('Imagem Original');
subplot(2,2,2); imshow(uint8(PS_red_green)); title('Red-Green');
subplot(2,2,3); imshow(uint8(PS_red_blue)); title('Red-Blue');
subplot(2,2,4); imshow(uint8(PS_green_blue)); title('Green-Blue');

```







Questão 4. Calcule o negativo (complemento) da imagem original ($255 - \text{imagem_original}$), bem como o complemento de cada um dos canais (R, G e B) da imagem original individualmente. Exiba a imagem complemento da imagem original, bem como os complementos dos três canais em uma janela definida pelo comando **subplot(2,2,...)**.

```
% 4 - Calculando negativo da imagem original e dos canais
```

```
PS_red_neg = zeros(h, w, c);
```

```
PS_green_neg = zeros(h, w, c);
```

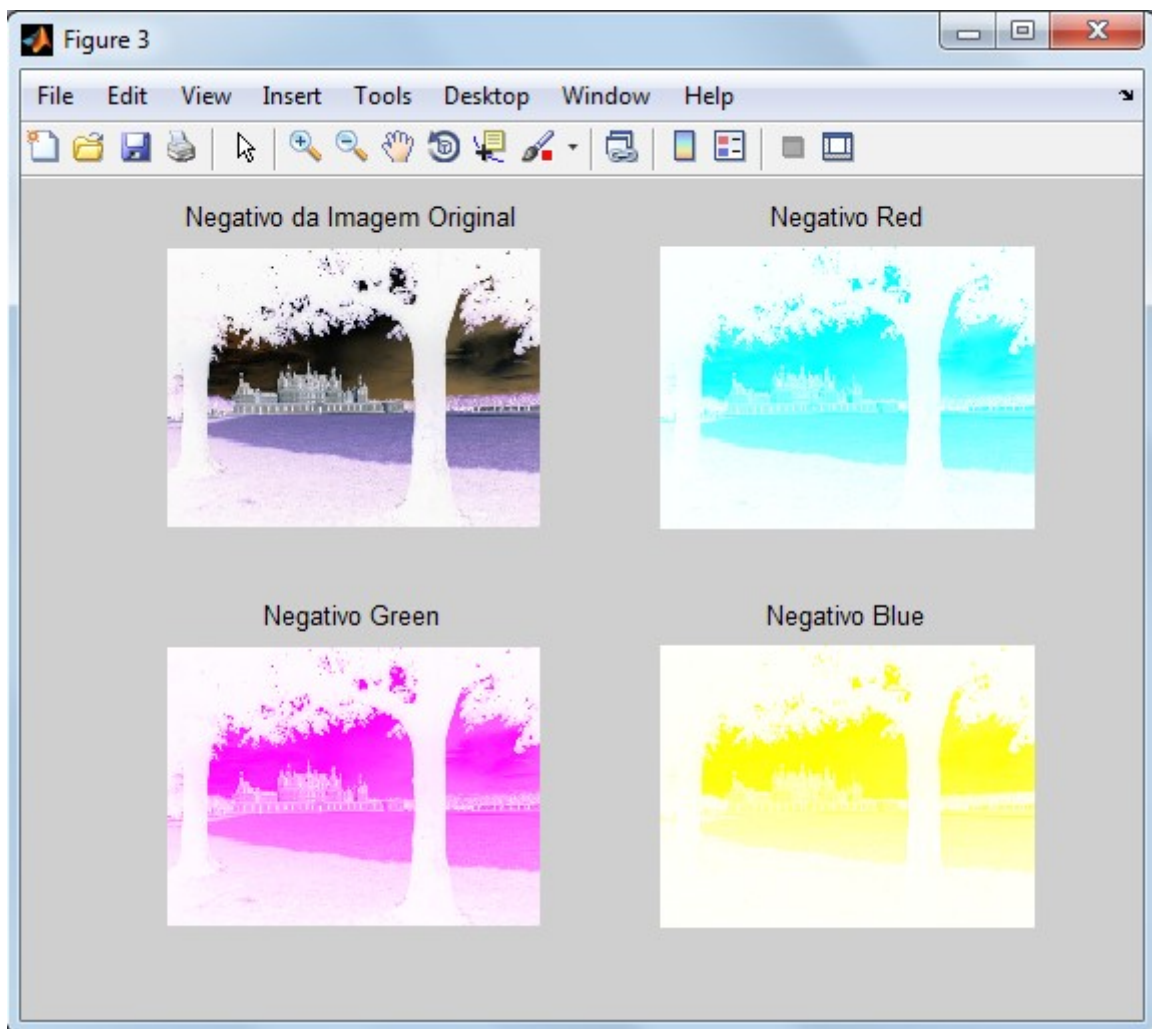
```
PS_blue_neg = zeros(h, w, c);
```

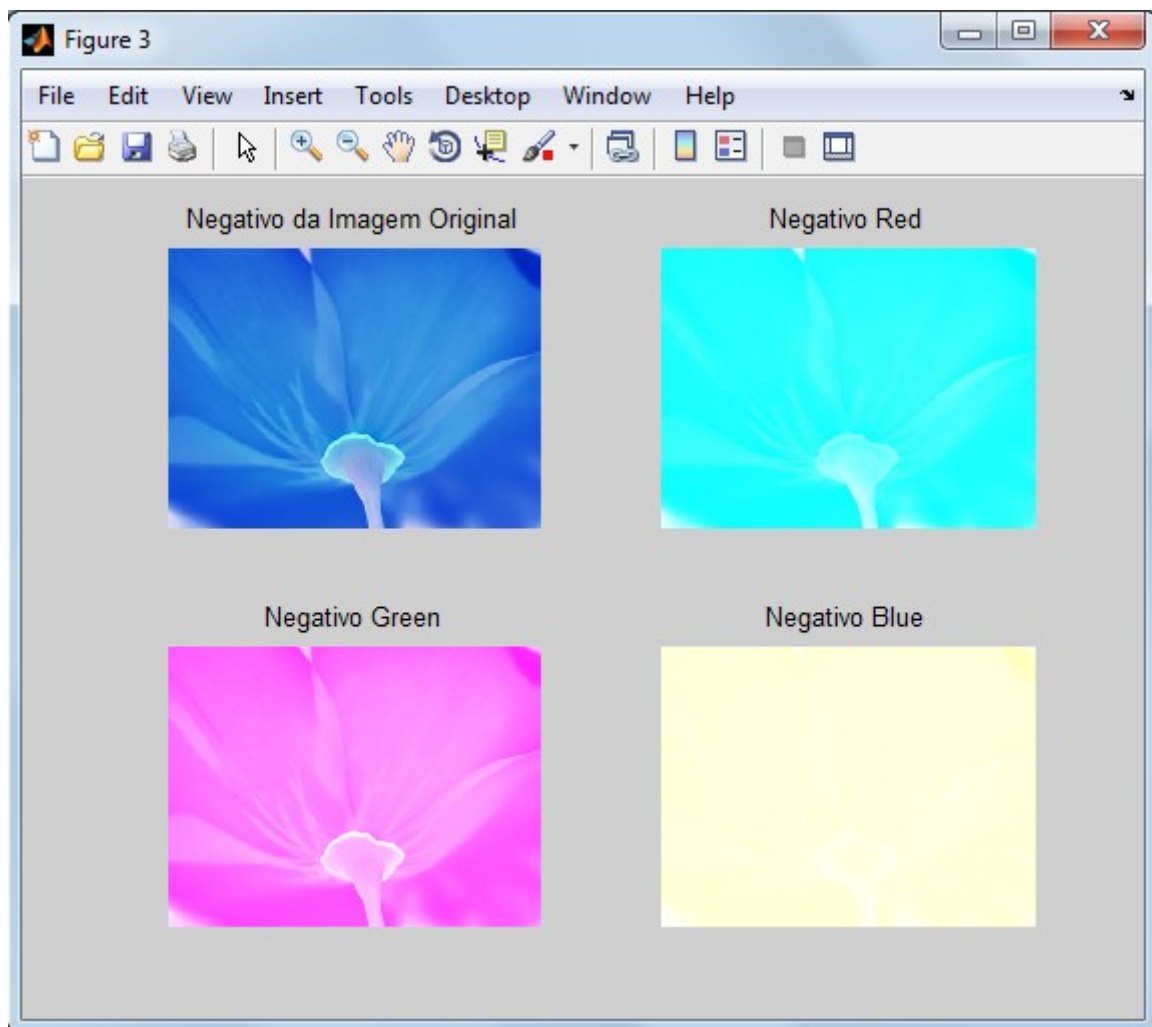
```
PS_ned = zeros(h, w, c);
```

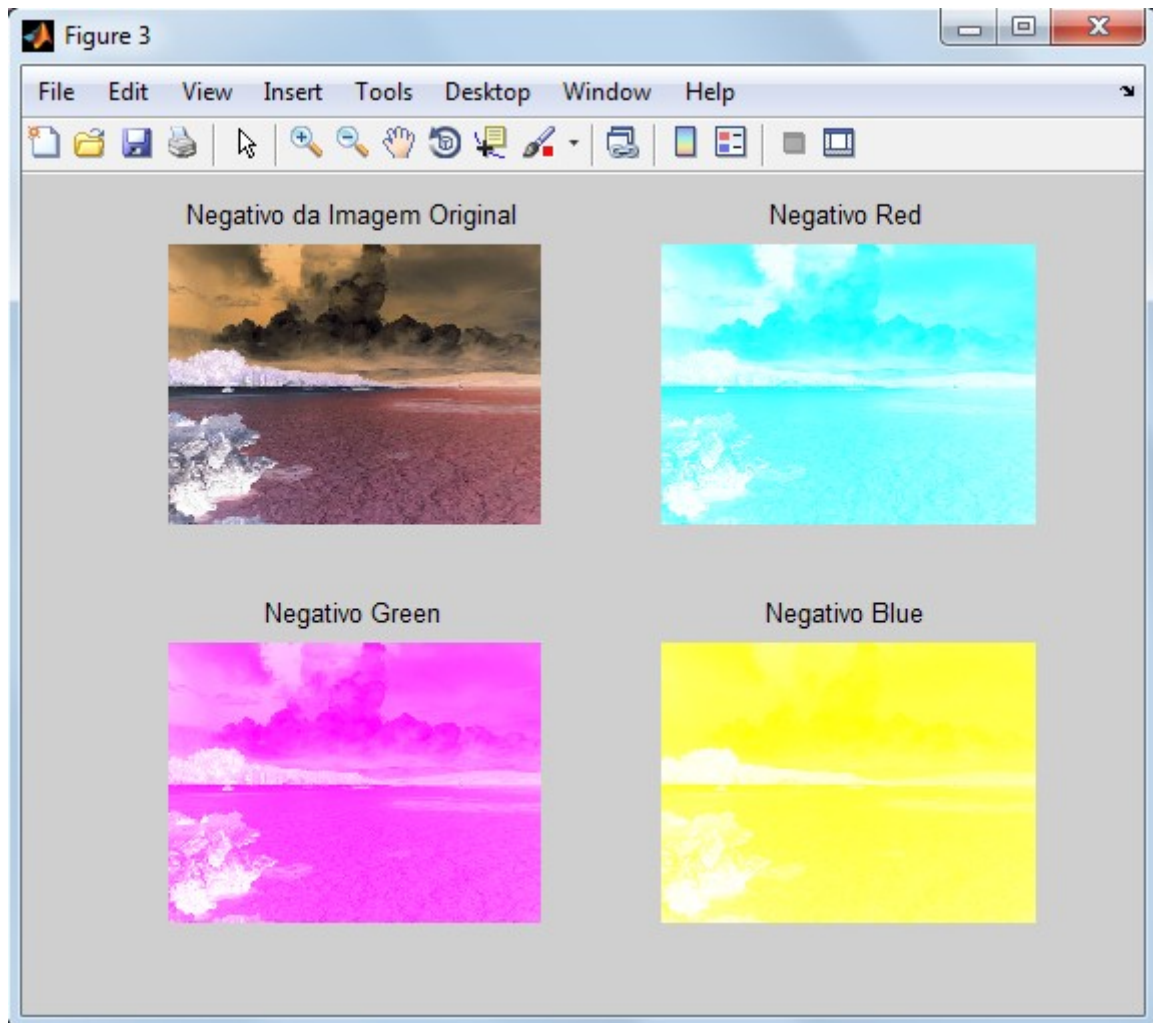
```
PS_neg = 255 - PS;
```



```
PS_red_neg    = 255 - double(PS_red);  
PS_green_neg  = 255 - double(PS_green);  
PS_blue_neg   = 255 - double(PS_blue);  
  
figure(3);  
subplot(2,2,1); imshow(uint8(PS_neg)); title('Negativo da Imagem Original');  
subplot(2,2,2); imshow(uint8(PS_red_neg)); title('Negativo Red');  
subplot(2,2,3); imshow(uint8(PS_green_neg)); title('Negativo Green');  
subplot(2,2,4); imshow(uint8(PS_blue_neg)); title('Negativo Blue');
```







Questão 5. Calcule a luminância (versão em tons de cinza da imagem colorida original) definida pela expressão $Luminância = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$. Exiba a imagem original e sua versão em tons de cinza, em uma janela definida pelo comando **subplot**(1, 2,...).

```
% 5 - Calculando Gray Scale da imagem original
```

```
PS_gray = zeros(h, w, c);
```

```
for i = 1:h
```

```
    for j = 1:w
```

```
        PS_gray(i,j,1) = 0.299 .* PS(i,j,1) + 0.587 .* PS(i,j,2) + 0.114 .* PS(i,j,3);
```

```
        PS_gray(i,j,2) = 0.299 .* PS(i,j,1) + 0.587 .* PS(i,j,2) + 0.114 .* PS(i,j,3);
```

```

        PS_gray(i,j,3) = 0.299 .* PS(i,j,1) + 0.587 .* PS(i,j,2) + 0.114 .*
        PS(i,j,3);

    end;

end;

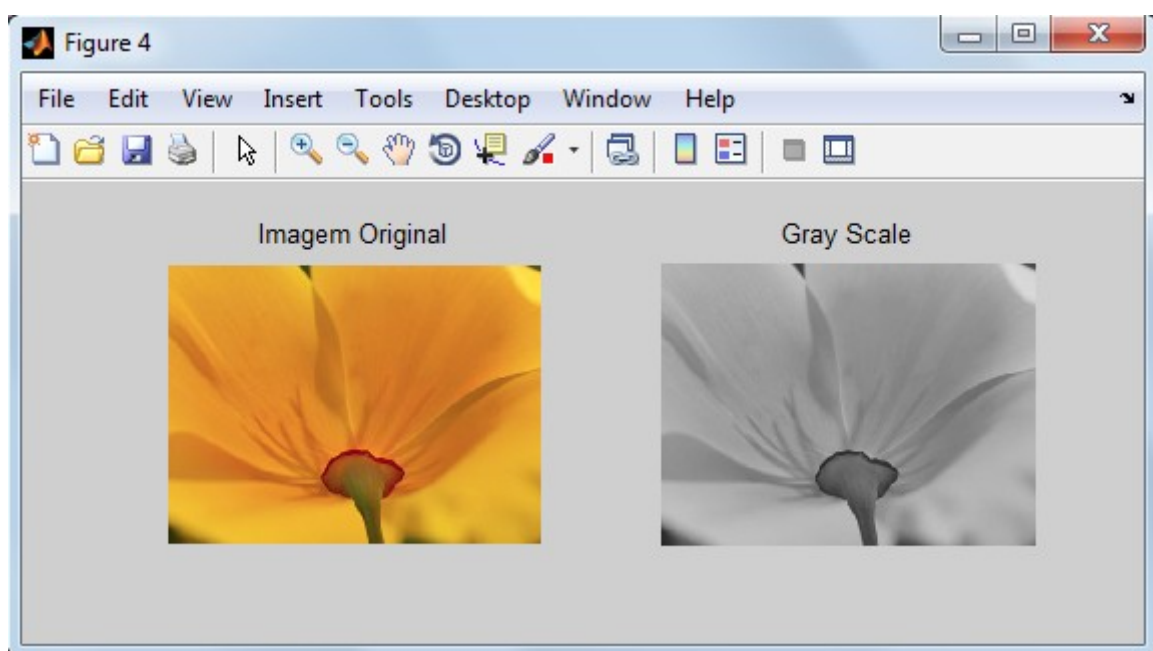
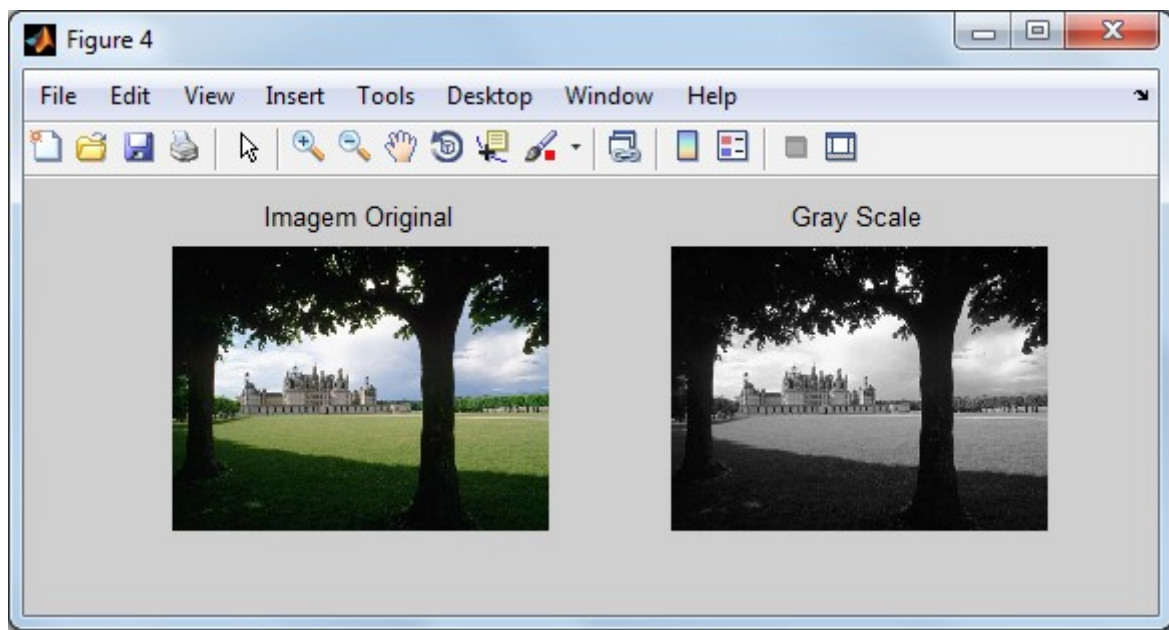
```

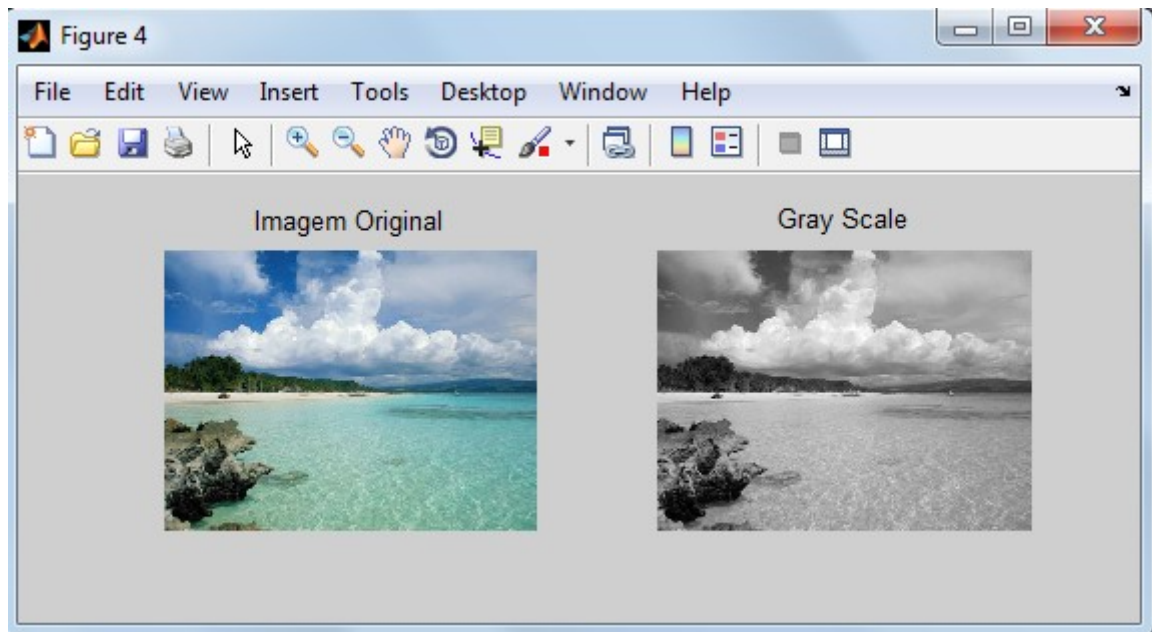
```

figure(4);

subplot(1,2,1); imshow(uint8(PS)); title('Imagem Original');
subplot(1,2,2); imshow(uint8(PS_gray)); title('Gray Scale');

```





Questão 6. Defina um kernel Gaussiano $G = [0.0625 \ 0.125 \ 0.0625; 0.125 \ 0.25 \ 0.125; 0.0625 \ 0.125 \ 0.0625]$ e obtenha o resultado da convolução da imagem colorida com este kernel. Você consegue aplicar a convolução diretamente sobre a imagem? Exiba a imagem original e sua filtrada, em uma janela definida pelo comando **subplot(1, 2,...)**.

A função *conv2* não pôde ser utilizada, pois ela faz convolução de uma matriz de uma dimensão apenas. Logo utilizei a função *convn*, que realiza a convolução em uma matriz n-dimensional, dado um kernel.

```
% 6 - Aplicando filtro gaussiano na imagem original

G = [0.0625 0.125 0.0625; 0.125 0.25 0.125; 0.0625 0.125 0.0625];

PS_gauss = convn(double(PS), G);

figure(5);

subplot(2,2,1); imshow(uint8(PS)); title('Imagem Original');

subplot(2,2,2); imshow(uint8(PS_gauss)); title('Filtro Gaussiano x1');

PS_gauss = convn(double(PS_gauss), G);

PS_gauss = convn(double(PS_gauss), G);

PS_gauss = convn(double(PS_gauss), G);

subplot(2,2,3); imshow(uint8(PS_gauss)); title('Filtro Gaussiano x4');
```



```
PS_gauss = convn(double(PS_gauss), G);  
PS_gauss = convn(double(PS_gauss), G);  
PS_gauss = convn(double(PS_gauss), G);  
PS_gauss = convn(double(PS_gauss), G);  
subplot(2,2,4); imshow(uint8(PS_gauss)); title('Filtro Gaussiano x8');
```

