

APOSTILA PARA OFICINA:

O ENSINO DOS CONCEITOS FUNDAMENTAIS DE MATRIZES UTILIZANDO
COMPUTAÇÃO GRÁFICA

PROFa. ANA PAULA SALSA BERNARDO

Aplicação Didática

Título	O ensino dos conceitos fundamentais de matrizes utilizando computação gráfica.
Nível de Ensino	Médio.
Tipo de atividade	Expositiva e prática podendo ser em grupo ou individual dependendo dos materiais disponíveis e número de participantes.
Duração	5 aulas de 3 horas.
Objetivos	<ul style="list-style-type: none"> • Tornar a prática docente mais motivadora e rica de elementos desafiadores; • Obter e/ou ampliar conhecimento em computação gráfica por meio de algum software matemático; • Desenvolver a capacidade de associar conceitos envolvendo matemática em situações do cotidiano; • Tornar a aula mais dinâmica e participativa, fazendo como que o aluno seja protagonista de seu aprendizado; • Fazer com que o aluno aprenda os conceitos básicos de matrizes por meio da utilização de um problema contextualizado convertendo matrizes em imagens; • Tornar o aluno capaz de interpretar os conteúdos aprendidos a uma situação real; • Proporcionar uma experiência que dê um significado àquilo que o aluno estiver aprendendo; • Maior interação entre o estudante e as novas tecnologias e informação e comunicação; • Oportunizar ao estudante expor suas ideias e opiniões sobre a forma como o conteúdo de matrizes foi apresentado; • Despertar o senso investigativo, bem como a curiosidade naquilo que se está aprendendo.
Conteúdo abordado	Matrizes e operações entre matrizes; Linguagem computacional GNU Octave (online); Conversão de Matrizes em Imagens.
Material utilizado	Notebook para o professor;

	Chromebook para o aluno (verificar a disponibilidade na escola) e/ou outro computador que o aluno possa desenvolver as atividades.
--	--

Metodologia

Esta atividade didática, consiste em apresentar para os estudantes do Ensino Médio, uma forma alternativa para compreender os conceitos fundamentais envolvendo as matrizes e sua aplicabilidade.

Nossa proposta tem como objetivo, proporcionar aos estudantes uma vivência que seja cooperativa ao trabalharem em grupo e que contribua conectar os conceitos de matrizes com situações da vida real por meio de ferramentas computacionais.

A seguir apresentamos as atividades que foram realizadas em cada encontro.

Aula 1: Apresentação da oficina e conteúdos teóricos matriciais implementados no Octave.

Apresentação da proposta destacando os objetivos para, em seguida, propor atividades contendo a explicação teórica sobre matrizes utilizando o GNU Octave online disponível em <https://octave-online.net/> de modo que os alunos desenvolvam as atividades simultaneamente, conforme descrito abaixo:

Atividade 1: Primeiro contato com o GNU Octave. Entenda como funciona. Aprenda a Salvar seus arquivos.

- Escolha do diretório para salvar seus arquivos (área de trabalho, documentos, etc);
- No Octave, podemos usar a Janela de Comandos ou o Editor. A diferença entre eles está em como interagimos e desenvolvemos nossos Scripts, conforme descrito a seguir:

Janela de Comandos: nessa janela podemos executar os comandos diretamente. Cada linha é executada após ser inserida. É ideal para execuções rápidas e interativas.

Editor: é mais apropriado para o desenvolvimento de scripts completos e estruturados, pois permite salvar o código para reutilizarmos depois. Ideal para quando precisarmos compilar o código várias vezes ou para compartilhar;

- Usaremos a janela de comandos para fazer os scripts da aula 1 por serem conteúdos teóricos em que podemos visualizar os resultados rapidamente.

Atividade 2: Como criar uma matriz no Octave

Criamos uma matriz de qualquer ordem (tamanho). Chamamos de matriz uma tabela de elementos dispostos em linhas e colunas. Por exemplo, considerando os dados referentes a altura, peso e idade de um grupo de 4 pessoas, podemos sistematizar esses dados de acordo com o quadro a seguir:

	Altura (m)	Peso (kg)	Idade (anos)
Pessoa 1	1,70	70	23
Pessoa 2	1,75	60	45
Pessoa 3	1,60	52	25
Pessoa 4	1,81	72	30

Esse quadro contendo as informações das quatro pessoas pode ser representado por meio de uma matriz denotada por

$$A = \begin{pmatrix} 1,70 & 70 & 23 \\ 1,75 & 60 & 45 \\ 1,60 & 52 & 25 \\ 1,81 & 72 & 30 \end{pmatrix}.$$

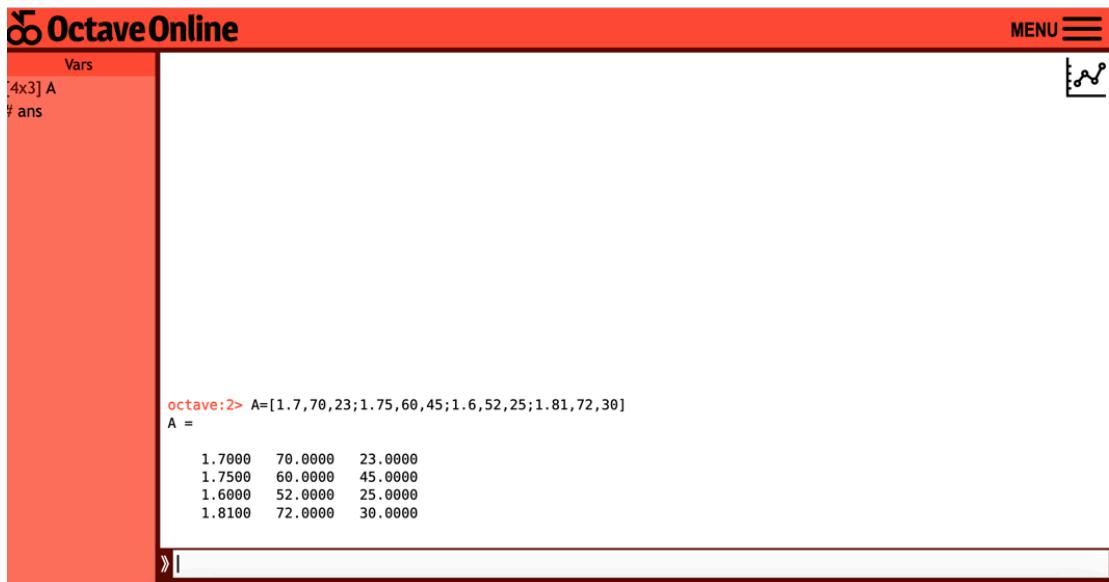
Estendemos a ideia passada por esse exemplo de uma maneira mais geral, em que consideramos uma matriz de m linhas e n colunas conforme definição a seguir.

Definição 1: Sejam $m \geq 1$ e $n \geq 1$ dois números inteiros. Uma matriz real $m \times n$ é uma dupla sequência de números reais, distribuídos em m linhas (horizontal) e n colunas (vertical), formando uma tabela que se indica por:

$$A_{m \times n} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = (a_{ij})_{m \times n}$$

Indicamos por $A_{m \times n} = (a_{ij})_{m \times n}$ a matriz A com m linhas e n colunas onde i indica a linha e j indica a coluna. Cada elemento a_{ij} da matriz, ocupa a i -ésima linha e j -ésima coluna da matriz.

No Octave utilizamos os seguintes comandos para exibir a matriz A :



```

Octave Online

Vars
4x3] A
# ans

octave:2> A=[1.7,70,23;1.75,60,45;1.6,52,25;1.81,72,30]
A =
1.7000 70.0000 23.0000
1.7500 60.0000 45.0000
1.6000 52.0000 25.0000
1.8100 72.0000 30.0000

```

Atividade 3: Tipos de matrizes (retangular, quadrada, coluna, linha, identidade, nula, constante e transposta).

Existem alguns tipos de matrizes que devido a sua importância recebem nomes especiais, como é o caso da matriz identidade e da matriz nula. Definimos a matriz identidade (ou matriz unidade) como sendo uma matriz quadrada ($m = n$), dada por $I_n = (a_{ij})_{n \times n}$, tal que

$$a_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad \forall i, j \in 1, 2, 3, \dots, n.$$

Uma propriedade importante da matriz identidade é que para qualquer matriz do tipo $n \times n$ teremos $A \cdot I_n = I_n \cdot A = A$.

Já a matriz nula, é uma matriz de ordem $m \times n$ em que todos os seus elementos $a_{ij} = 0$, $\forall i \in 1, 2, 3, \dots, m$ e $j \in 1, 2, 3, \dots, n$.

Outro tipo de matriz que usaremos em nossas atividades é a matriz transposta. Chamamos de matriz transposta de uma matriz $A = (a_{ij})_{m \times n}$ a matriz dada por $A^t = (b_{ij})_{n \times m}$, cujas linhas são as colunas de A , isto é, $b_{ij} = a_{ji}$.

A partir do momento em que compreendemos o significado matemático de uma matriz e também temos uma noção do que ela pode representar no mundo real, podemos avançar para as suas operações.

```

Vars
[3x1] C
[2x2] CT
[3x3] I
[1x3] L
[3x2] N
[2x2] Q
[2x3] R
[3x2] Rt
# ans

octave:1> R=[1,2,3;4,5,6] % Matriz retangular 2x3
R =
1 2 3
4 5 6

octave:2> Q=[-1,4;2,-3] % Matriz quadrada 2x2
Q =
-1 4
2 -3

octave:3> C=[-10;2;7] % Matriz coluna 3x1
C =
-10
2
7

octave:4> L=[2,-7,12] % Matriz linha 1x3
L =
2 -7 12

octave:5> I=[1,0,0;0,1,0;0,0,1] % Matriz Identidade 3x3
I =
1 0 0
0 1 0
0 0 1

octave:6> N=[0,0;0,0;0,0] % Matriz nula 3x2
N =
0 0
0 0
0 0

octave:7> CT=[7,7;7,7] % Matriz constante 2x2
CT =
7 7
7 7

octave:8> Rt=[1,4;2,5;3,6] % Matriz transposta de R
Rt =
1 4
2 5
3 6

```

Atividade 4: Matriz oposta e operação de soma e subtração de matrizes.

Sejam $A = (a_{ij})_{m \times n}$ e $B = (b_{ij})_{m \times n}$, matrizes $m \times n$. A adição de A com B , indicada $A + B$, será a matriz $C = A + B = (a_{ij} + b_{ij})_{m \times n} = (c_{ij})_{m \times n}$.

A operação de adição possui suas propriedades sendo elas:

Comutativa	$A + B = B + A$
Associativa	$A + (B + C) = (A + B) + C$
Existência da matriz que representa o elemento neutro	Existe uma matriz nula denominada por O , $a_{ij} = 0$, $\forall i \in 1, 2, 3, \dots, m$ e $j \in 1, 2, 3, \dots, n$, de mesma ordem da matriz A tal que: $A + O = A = O + A$
Existência da matriz oposta	Dada uma matriz A , existe uma matriz $B = -A$, tal que $A + -(-A) = O$

A operação de Subtração deriva da adição ao somarmos uma matriz A com a matriz oposta de uma matriz B , resultando em $A + (-B) = A - B$.

```

Octave Online
Vars
[2x3] A
[2x3] B
[2x3] ans

octave:1> A=[1,5,3;-2,4,-6] % Matriz 2x3
A =
1 5 3
-2 4 -6

octave:2> B=-A % Matriz oposta de A
B =
-1 -5 -3
2 -4 6

octave:3> A+B %Soma de matrizes (devem ter o mesmo tamanho)
ans =
0 0 0
0 0 0

octave:4> A-B % Subtração de matrizes (A + (-B))
ans =
2 10 6
-4 8 -12

```

Atividade 5: Multiplicação de uma matriz por um escalar e multiplicação entre matrizes.

Dados a matriz $A = (a_{ij})_{m \times n}$ e um número real α , o produto de α por A é a matriz real $m \times n$ dada por:

$$\alpha \cdot A = \begin{pmatrix} \alpha \cdot a_{11} & \alpha \cdot a_{12} & \dots & \alpha \cdot a_{1n} \\ \alpha \cdot a_{21} & \alpha \cdot a_{22} & \dots & \alpha \cdot a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha \cdot a_{m1} & \alpha \cdot a_{m2} & \dots & \alpha \cdot a_{mn} \end{pmatrix}$$

Suas propriedades são dadas por:

Associativa	$(\alpha\beta)A = \alpha(\beta A)$; $\alpha\beta$ são escalares reais.
Distributiva	$(\alpha + \beta)A = \alpha A + \beta A$ $\alpha(A + \beta) = \alpha A + \beta A$
Multiplicação do número zero por uma matriz A	$0 \cdot A = 0$; 0 é a matriz nula.

Sejam $A = (a_{ij})_{m \times n}$ e a matriz $B = (b_{ij})_{n \times p}$. O produto $A \cdot B$ (também indicado AB) é matriz $C = (c_{ij})_{m \times p}$ cujo termo geral c_{ij} é obtido somando-se a multiplicação ordenada dos elementos da i -ésima linha pela j -ésima coluna. Matematicamente:

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}; \forall i \in \{1, \dots, m\} \text{ e } j \in \{1, \dots, p\}$$

Vale lembrar que, só definimos (ou só existe) o produto de duas matrizes A e B , quando o número de colunas de A for igual ao número de linhas de B . Além disso, $AB = O$ (matriz nula), somente se $A = O$ ou $B = O$. As propriedades para esta operação são:

Elemento neutro multiplicativo	$AI = IA = A; I$ é a matriz identidade.
Distributiva à esquerda e à direita, respectivamente, da multiplicação em relação à soma	$A(B + C) = AB + AC$ $(A + B)C = AC + BC$
Associativa	$(AB)C = A(BC)$
A transposta da multiplicação entre duas matrizes	$(AB)^t = B^t A^t$
Multiplicação do número zero por uma matriz A	$0 \cdot A = O; O$ é a matriz nula.

Em geral, $AB \neq BA$, entretanto, se $AB = BA$ dizemos que as matrizes A e B são comutáveis.

No octave os comandos são dados por:

```

octave:1> A=[-1,0,4;5,6,-2;3,2,1] % Matriz 3x3
A =
-1   0   4
 5   6  -2
 3   2   1

octave:2> 2*A % Multiplicação da Matriz A pelo escalar 2
ans =
-2   0   8
10  12  -4
 6   4   2

octave:3> B=[5,6,-1;0,0,1;3,0,2] % Matriz 3x2
B =
 5   6  -1
 0   0   1
 3   0   2

octave:4> A*B % Multiplicação entre A e B (A.B)
ans =
 7  -6   9
19  30  -3
18  18   1

octave:5> B*A % Nesse caso é possível fazer o produto B.A. Em geral não vale a comutatividade para multiplicação
ans =
22  34   7
 3   2   1
 3   4  14

```

Atividade 6: Matriz inversa

Dada uma matriz quadrada A , de ordem n , se existir uma matriz quadrada B , de mesma ordem, que satisfaz a condição: $AB = BA = I$ diremos que B é a inversa de A e se representa por A^{-1} . Assim, $AA^{-1} = A^{-1}A = I$.

```
octave:1> A=[1,2;4, -2] % Matriz 2x2
A =
1 2
4 -2

octave:2> B=inv(A) % B é a matriz inversa de A
B =
0.2000  0.2000
0.4000 -0.1000

octave:3> I = A*B % O resultado do produto entre A e sua inversa é a identidade I
I =
1 0
0 1
```

Aula 2: Aprendendo criar imagens no Octave fazendo a conversão de matriz para imagem e vice-versa (imagem bidimensional x matriz binária).

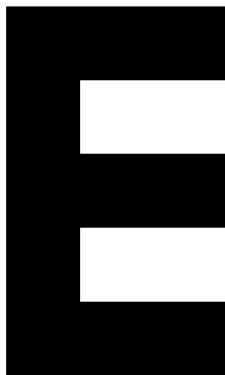
Nesta aula iniciamos a conversão de matrizes em imagens e também, alguns efeitos que são possíveis aplicar nas imagens.

Atividade 1: Obtenha a imagem da letra E por meio de uma matriz que a represente. Lembre-se de atribuir o valor 0 para o preto e 1 para o branco. Para converter uma matriz em uma imagem temos que usar o comando “uint8”. Veja o código a seguir.

```
octave:1> E=
[1,0,0,0,1;1,0,1,1,1;1,0,0,0,1;1,0,1,1,1;1,0,0,0,1]
E =
1 0 0 0 1
1 0 1 1 1
1 0 0 0 1
1 0 1 1 1
1 0 0 0 1

octave:3> E_image = uint8(E*255)
E_image =
255 0 0 0 255
255 0 255 255 255
255 0 0 0 255
255 0 255 255 255
255 0 0 0 255
```

```
octave:4> imshow(E_image)
```



Atividade 2: Obtenha a imagem do gato por meio de uma matriz que a represente. Lembre-se de atribuir o valor 0 para o preto e 1 para o branco. Para converter uma matriz em uma imagem temos que usar o comando “uint8”. Veja o código a seguir.

Código do Gato com fundo branco:

```

111111000000000000000000010111;1111111001111110001
11000000000100111;11111111001111101111101111011100
1001111;111111110011111001111110010011111;111
11111110001111000000000000111111;111111111111110
0000111110000011111111;111111111111110000000000
001111111111];
imwrite(G,"imagem.png");
imshow(G)

```

Código do Gato com fundo preto:

Atividade 3: Debate com perguntas

O que é uma imagem digital?

Resposta: Uma imagem digital é uma representação de uma imagem visual em um formato que pode ser armazenado, processado e exibido por dispositivos eletrônicos, como computadores, smartphones e câmeras digitais. Em outras

palavras, uma imagem digital é a representação numérica de uma imagem bidimensional constituída por uma matriz binária.

O que é uma imagem binária?

Resposta: Uma imagem binária corresponde a uma imagem bidimensional composta por zero (preto) e um (branco).

O que é o pixel?

Resposta: A imagem digital é composta por uma grade de pequenos pontos, chamados pixels, cada um representando uma cor ou nível de brilho específico. O pixel pode ser entendido assim, como a menor unidade de uma imagem digital. Cada pixel possui uma cor e uma intensidade que são determinadas por valores numéricos.

Qual o tamanho da matriz usada para representar a letra E?

Resposta: A matriz que representa a letra E tem tamanho 5x5.

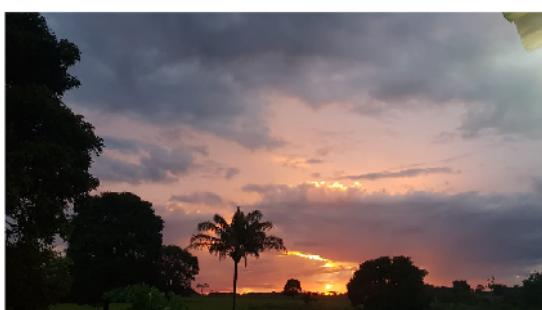
Atividade 4: Aprendendo a importar e salvar uma imagem no Octave.

No Octave Online, clique em "Upload file", selecione a imagem desejada do seu computador nos formatos compatíveis que pode ser em JPG ou PNG, e ela ficará disponível para uso no ambiente.

Vejamos um exemplo: Para importar a imagem intitulada "Anoitecer", devemos importar a imagem que foi enviada para o ambiente online, usar o comando "imread('Anoitecer.jpg')" e, em seguida, exibir a imagem por meio do comando "imshow(imagem)", conforme ilustrado no código abaixo.

```
octave:1> % Importa a imagem que foi enviada para o ambiente online
          
imagem = imread('Anoitecer.jpg');

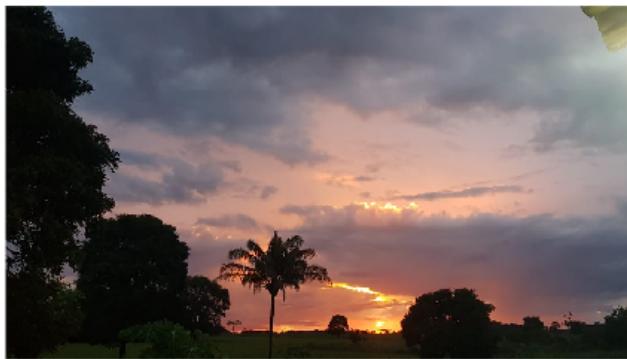
% Exibe a imagem carregada para garantir que foi importada corretamente
imshow(imagem);
```



Para salvar a imagem em outro formato, usamos o código abaixo:

```
octave:1> % Importa a imagem que foi enviada para o
ambiente online
imagem = imread('Anoitecer.jpg');

% Exibe a imagem carregada para garantir que foi
importada corretamente
imshow(imagem);
% Salvar a imagem no formato PNG
imwrite(imagem, 'imagem_processada.png');
```



Aula 3: Aprendendo a converter uma imagem em escala de cinza e identificar os efeitos obtidos em imagens por meio de operações matriciais

Atividade 1: Dada uma imagem, faça a conversão dessa imagem para escala de cinza.

Exemplo 1: Para converter uma imagem colorida em escala de cinza, escolhemos a imagem intitulada “Ipeamarelo” em formato png e usamos os seguintes comandos:



Octave Online MENU

Files 4

Anoitecer.jpg
cafemaisorquidea.jpg
ipeamarelo.png
my_script1.m

Drop Files Here
to Upload

my_script1.m

RUN ▶

```
1  imagem = imread('ipeamarelo.png'); % Carrega a imagem
2  imagem_pb = rgb2gray(imagem); % Converte para preto e branco
3  imshow(imagem_pb); % Exibe a imagem preto e branco
4  imwrite(imagem_pb, 'imagem_pb.png'); % Salva a nova imagem
5
```

Vars (1600x777x3)
(1600x777)

octave:9> source("my_script1.m")

Exemplo 2: Faremos o mesmo para a imagem “rosabranca”.



OctaveOnline MENU

Files + ↻

my_script1.m

Jiboa2.png
anoitecer.png
cafezamioculca.png
ceu.png
cocacola.png
imagem_pb.png
ipeamarelo.png
jiboa.png
my_script1.m
my_script2.m
pordosol.png
rosabranca.png
sítioarqueológico.png

Drop Files Here to Upload

```
1 imagem = imread('rosabranca.png'); % Carrega a imagem
2 imagem_pb = rgb2gray(imagem); % Converte para preto e branco
3 imshow(imagem_pb); % Exibe a imagem preto e branco
4 imwrite(imagem_pb, 'imagem_pb.png'); % Salva a nova imagem
```

RUN ▶

Vars

```
# ans
(1045x778x3)
(1045x778) tf
(1600x1200x3)
```

octave:7> source("my_script1.m")

Atividade 2: Debate com perguntas

Qual o significado de uma imagem em escala de cinza?

Resposta: Uma imagem em escala de cinza é uma imagem que representa apenas variações de intensidade de luz, indo do preto ao branco. Em vez de usar cores como em imagens RGB (vermelho, verde e azul), cada pixel é codificado com um valor que indica a sua luminosidade, sendo o preto o valor mínimo (geralmente 0) e o branco o máximo (geralmente 255 em imagens de 8 bits). Isso resulta em uma transição contínua de tons de cinza.

Qual a variação das tonalidades de cinza para cada pixel no Octave?

Resposta: No Octave, em imagens de 8 bits (que são as mais comuns), a tonalidade de cinza de cada pixel varia de 0 a 255. Esse intervalo representa a intensidade da luz, onde:

- 0 corresponde ao preto (ausência de luz);
- 255 corresponde ao branco (máxima intensidade de luz);

- E os valores entre 0 e 255 representam diferentes tons de cinza, com 128, por exemplo, sendo um tom intermediário.

Se a imagem tiver maior profundidade de bits (por exemplo, 16 bits), o intervalo será maior (0 a 65535).

Atividade 3: Escolha uma imagem convertendo-a em escala de cinza. Em seguida, some a imagem obtida com uma matriz constante em escala de cinza.

a) Conversão da imagem "rosabranca" em escala de cinza (preto e branco):

```
octave:1> imagem = imread('rosabranca.png');
octave:2> imshow(imagem);
```



```
octave:3> imagem = imread('rosabranca.png'); %
Carrega a imagem
imagem_pb = rgb2gray(imagem); % Converte para preto
e branco
imshow(imagem_pb); % Exibe a imagem preto e branco
imwrite(imagem_pb, 'imagem_pb.png'); % Salva a nova
imagem
```



b) Soma da imagem em escala de cinza com uma matriz constante também em escala de cinza.

Aos escolhermos uma imagem em escala de cinza, podemos somar esta imagem a uma outra, cuja matriz que a representa é uma matriz constante de fator 150, também em escala de cinza. Como resultado, teremos um clareamento da imagem.

OctaveOnline MENU

Files RUN ▶ Vars Why this ad? ⓘ

```
# ans
[1045x778] c
# const_value
<1045x778x3>
<1045x778> i
<1045x778x3>
<1045x778> i
# m
# n
```

octave:7> source("my_script2.m")

Matriz constante



Drop Files Here to Upload

my_script2.m

```
1 % Carregar a imagem e converter para escala de cinza (se necessário)
2 img = imread('rosabranca.png'); % Substitua 'minha_imagem.jpg' pelo nome ou caminho da imagem
3 img_gray = rgb2gray(img); % Converter para escala de cinza (caso a imagem seja colorida)
4
5 % Obter as dimensões da imagem
6 [m, n] = size(img_gray);
7
8 % Criar uma matriz constante com o mesmo tamanho da imagem
9 const_value = 150; % Valor constante desejado (entre 0 e 255)
10 const_matrix = const_value * ones(m, n); % Criar uma matriz de tamanho [m x n] com o valor 150
11
12 % Exibir a matriz constante
13 imshow(uint8(const_matrix));
14 title('Matriz constante');
```

15

OctaveOnline MENU

Files RUN ▶ Vars Why this ad? ⓘ

```
# ans
[1045x778] c
# const_value
<1045x778x3>
<1045x778> i
<1045x778x3>
<1045x778> i
# m
# n
(1045x778) r
```

octave:8> source("my_script3.m")

Imagen somada com matriz constante



Drop Files Here to Upload

my_script3.m

```
1 % Passo 1: Carregar a imagem e converter para escala de cinza
2 img = imread('rosabranca.png'); % Substitua 'minha_imagem.jpg' pelo caminho da sua imagem
3 img_gray = rgb2gray(img); % Converter para escala de cinza, se for colorida
4
5 % Passo 2: Criar uma matriz constante com o mesmo tamanho da imagem
6 [m, n] = size(img_gray); % Obter as dimensões da imagem
7 const_matrix = 150 * ones(m, n); % Crie uma matriz constante (ajuste o valor conforme necessário, de 0 a 255)
8
9 % Passo 3: Somar a imagem em escala de cinza com a matriz constante
10 result_img = img_gray + const_matrix;
11
12 % Garantir que o valor da soma esteja entre 0 e 255
13 result_img = uint8(min(result_img, 255)); % Limita os valores acima de 255
14
15 % Passo 4: Exibir a imagem resultante
16 imshow(result_img);
17 title('Imagen somada com matriz constante');
18
19 % Passo 5: Salvar a imagem resultante
20 imwrite(result_img, 'imagem_resultante.jpg');
```

21

O resultado do clareamento é dado pela soma das imagens R e G:

Imagen original em escala de cinza (R)



Imagen em escala de cinza (G)

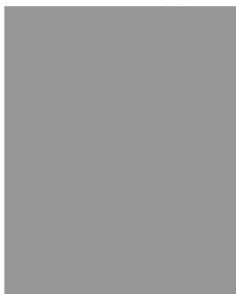


Imagen clareada (C)
 $R + G = C$



Responda:

O que acontece com a imagem resultante?

Resposta: A imagem resultante fica mais clara, com possíveis áreas claras saturadas em branco, enquanto a estrutura original é mantida, porém com redução de contraste.

Atividade 4: Escolha uma imagem convertendo-a em escala de cinza. Em seguida, subtraia dessa imagem uma imagem obtida a partir da matriz constante em escala de cinza.



The screenshot shows the Octave Online interface. On the left, the 'Files' sidebar lists various image files. The main workspace shows a script named 'my_script4.m' with the following code:

```
1 % Passo 1: Carregar a imagem e converter para escala de cinza
2 img = imread('rosabranca.png'); % Substitua 'minha_imagem.jpg'
3 img_gray = rgb2gray(img); % Converter para escala de
4 % cinza, se for colorida
5
6 [m, n] = size(img_gray); % Obter as dimensões da
7 % imagem
8 const_value = 150; % Valor constante a ser
9 % subtraído (ajuste conforme necessário)
10 const_matrix = const_value * ones(m, n); % Criar uma matriz
11 % constante de tamanho [m x n]
12
13 % Passo 3: Subtrair a matriz constante da imagem em escala de
14 result_img = img_gray - const_matrix;
15
16 % Garantir que o valor da subtração esteja entre 0 e 255
17 result_img = uint8(max(result_img, 0)); % Limita os valores
18 % abaixo de 0
19
20 % Passo 4: Exibir a imagem resultante
21 imshow(result_img);
22 title('Imagen apds subtração com matriz constante');
23
24 % Passo 5: Salvar a imagem resultante
25 imwrite(result_img, 'imagem_subtraida.jpg');
```

The 'RUN' button is highlighted. To the right, the 'Vars' panel shows variables like 'ans' and 'const_matrix'. The output window shows the command 'octave:2> source("my_script4.m")' and the resulting image titled 'Imagen apds subtração com matriz constante', which is a dark, almost black image of a rose.

O resultado do escurecimento é dado pela subtração das imagens R e G:

Imagen original em escala de cinza (R)



Imagen em escala de cinza (G)



Imagen clareada (C)
 $R - G = E$



Responda:

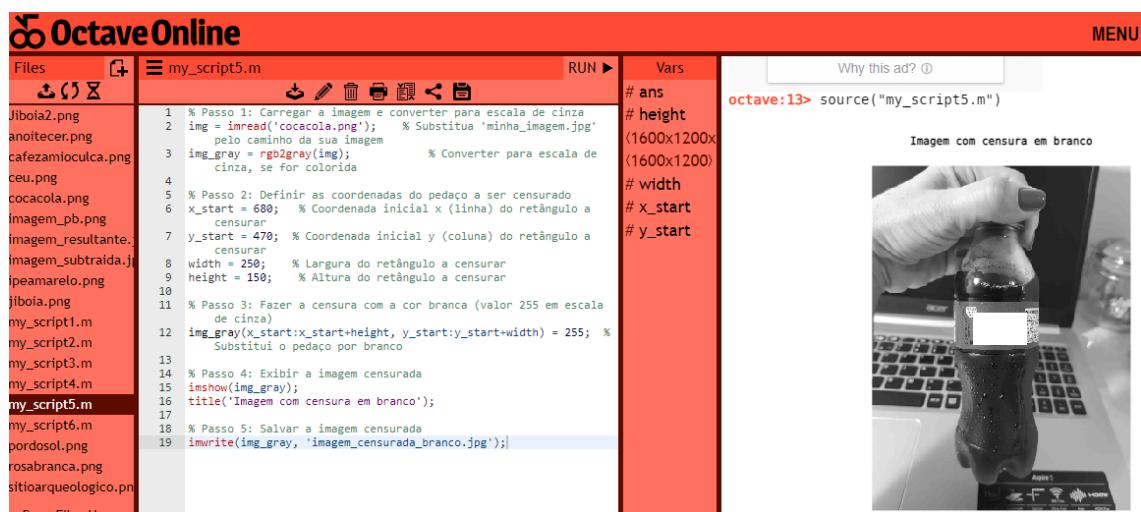
O que acontece com a imagem resultante?

Resposta: A imagem resultante fica mais escura, com áreas escuras podendo se saturar em preto, mantendo a estrutura, mas com menos visibilidade nos detalhes devido à redução de contraste nas áreas claras.

Aula 4: Outros efeitos interessantes utilizando as operações matriciais (ainda para imagens em escala de cinza)

Atividade 1: Escolha uma imagem convertendo-a em escala de cinza. Faça uma censura em sua imagem, escondendo um pedaço que você não quer que apareça. Primeiro, faça sua censura com a cor branco e depois com o preto.

Censura com a cor branco:



The screenshot shows the Octave Online interface with a script named 'my_script5.m' open. The script reads a Coca-Cola bottle image, converts it to grayscale, and then performs a rectangular censoring operation on a specific region. The censored area is then displayed. The image shows a hand holding a Coca-Cola bottle over a laptop keyboard, with the censored region visible on the bottle's label.

```
1 % Passo 1: Carregar a imagem e converter para escala de cinza
2 img = imread('cocacola.png'); % Substitua 'minha_imagem.jpg' pelo caminho da sua imagem
3 img_gray = rgb2gray(img); % Converter para escala de cinza, se for colorida
4
5 % Passo 2: Definir as coordenadas do pedaço a ser censurado
6 x_start = 680; % Coordenada inicial x (linha) do retângulo a censurar
7 y_start = 470; % Coordenada inicial y (coluna) do retângulo a censurar
8 width = 250; % Largura do retângulo a censurar
9 height = 150; % Altura do retângulo a censurar
10
11 % Passo 3: Fazer a censura com a cor branca (valor 255 em escala de cinza)
12 img_gray(x_start:x_start+height, y_start:y_start+width) = 255; % Substitui o pedaço por branco
13
14 % Passo 4: Exibir a imagem censurada
15 imshow(img_gray);
16 title('Imagen com censura em branco');
17
18 % Passo 5: Salvar a imagem censurada
19 imwrite(img_gray, 'imagem_censurada_branco.jpg');
```

Código completo:

Passo 1: Carregar a imagem e converter para a escala de cinza
img = imread('cocacola.png'); (carregamos a imagem)

```
img_gray = rgb2gray(img); (convertemos em escala de cinza, se necessário)
```

Passo 2: Definir as coordenadas da área da imagem a ser censurada.

```
x_start = 680; (coordenada inicial x (linha) do retângulo a censurar)
```

```
y_start = 470; (coordenada inicial y (coluna) do retângulo a censurar)
```

```
width = 250; (largura do retângulo a censurar)
```

```
height = 150; (altura do retângulo a censurar)
```

Passo 3: Fazer a censura com a cor branca (valor 255 em escala de cinza)

```
img_gray(x_start:x_start+height, y_start:y_start+width) = 255; (Substituímos a área desejada por branco)
```

Passo 4: Exibir a imagem censurada

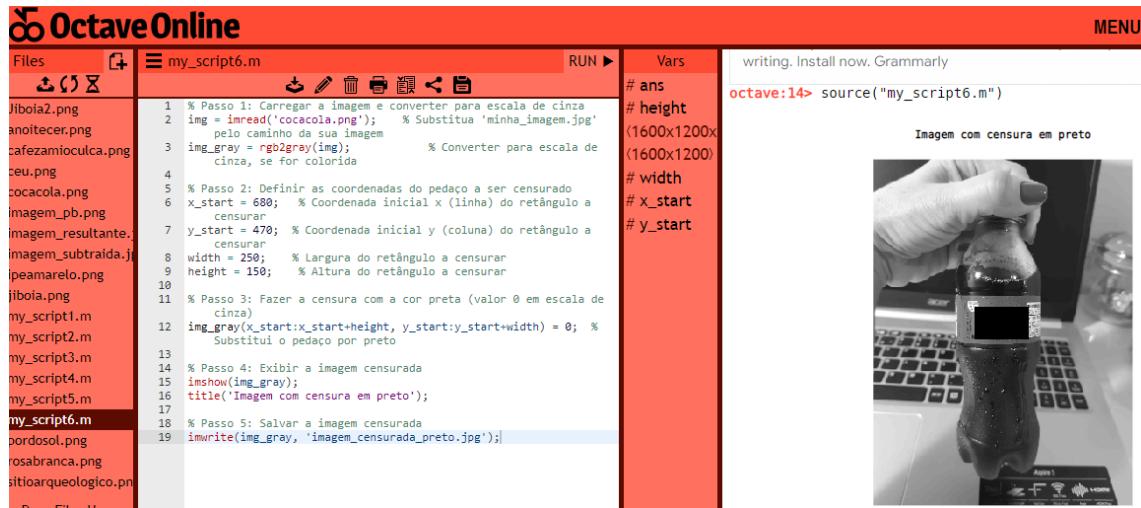
```
imshow(img_gray); (exibimos a imagem resultante)
```

```
title('Imagem com censura em branco');
```

Passo 5: Salvar a imagem censurada (resultante)

```
imwrite(img_gray, 'imagem_censurada_branco.jpg');
```

Censura com a cor preto:



The screenshot shows the Octave Online interface with a script named 'my_script6.m'. The code is as follows:

```
1 % Passo 1: Carregar a imagem e converter para escala de cinza
2 img = imread('cocacola.png'); % Substitua 'minha_imagem.jpg'
3 img_gray = rgb2gray(img); % Converter para escala de cinza, se for colorida
4
5 % Passo 2: Definir as coordenadas do pedaço a ser censurado
6 x_start = 680; % Coordenada inicial x (linha) do retângulo a censurar
7 y_start = 470; % Coordenada inicial y (coluna) do retângulo a censurar
8 width = 250; % Largura do retângulo a censurar
9 height = 150; % Altura do retângulo a censurar
10
11 % Passo 3: Fazer a censura com a cor preta (valor 0 em escala de cinza)
12 img_gray(x_start:x_start+height, y_start:y_start+width) = 0; % Substitui o pedaço por preto
13
14 % Passo 4: Exibir a imagem censurada
15 imshow(img_gray);
16 title('Imagem com censura em preto');
17
18 % Passo 5: Salvar a imagem censurada
19 imwrite(img_gray, 'imagem_censurada_preto.jpg');
```

The right side of the interface shows the resulting image, which is a photograph of a hand holding a Coca-Cola bottle with a black rectangular redaction box covering the middle portion of the bottle.

Código completo:

Passo 1: Carregar a imagem e converter para a escala de cinza

```
img = imread('cocacola.png'); (carregamos a imagem)
```

```
img_gray = rgb2gray(img); (convertemos em escala de cinza, se necessário)
```

Passo 2: Definir as coordenadas da área da imagem a ser censurada.

```
x_start = 680; (coordenada inicial x (linha) do retângulo a censurar)
```

```
y_start = 470; (coordenada inicial y (coluna) do retângulo a censurar)
width = 250; (largura do retângulo a censurar)
height = 150; (altura do retângulo a censurar)
```

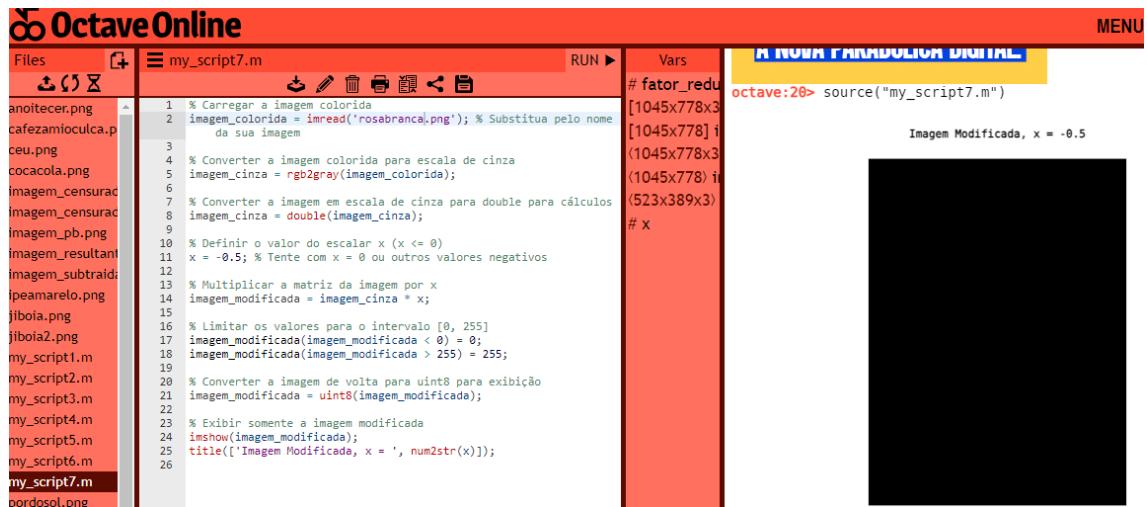
Passo 3: Fazer a censura com a cor branca (valor 0 em escala de cinza)
 $\text{img_gray}(\text{x_start}:\text{x_start}+\text{height}, \text{y_start}:\text{y_start}+\text{width}) = 0;$ (Substituímos a área desejada por preto)

Passo 4: Exibir a imagem censurada
 $\text{imshow}(\text{img_gray});$ (exibimos a imagem resultante)
 $\text{title}('Imagem com censura em preto');$

Passo 5: Salvar a imagem censurada
 $\text{imwrite}(\text{img_gray}, \text{'imagem_censurada_preto.jpg'});$

Atividade 2: Considere uma imagem em escala de cinza. Em seguida, multiplique essa matriz por um escalar real x .

- a) Tomando um valor de $x \leq 0$, por exemplo $x = -0.5$ a imagem fica inteiramente na cor preto.



```
Octave Online
my_script7.m
1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png'); % Substitua pelo nome
3 % da sua imagem
4 % Converter a imagem colorida para escala de cinza
5 imagem_cinza = rgb2gray(imagem_colorida);
6 % Converter a imagem em escala de cinza para double para cálculos
7 imagem_cinza = double(imagem_cinza);
8 % Definir o valor do escalar x (x <= 0)
9 x = -0.5; % Tente com x = 0 ou outros valores negativos
10 % Multiplicar a matriz da imagem por x
11 imagem_modificada = imagem_cinza * x;
12 % Limitar os valores para o intervalo [0, 255]
13 imagem_modificada(imagem_modificada < 0) = 0;
14 imagem_modificada(imagem_modificada > 255) = 255;
15 % Converter a imagem de volta para uint8 para exibição
16 imagem_modificada = uint8(imagem_modificada);
17 % Exibir somente a imagem modificada
18 imshow(imagem_modificada);
19 title(['Imagem Modificada, x = ', num2str(x)]);
20
```

Código:

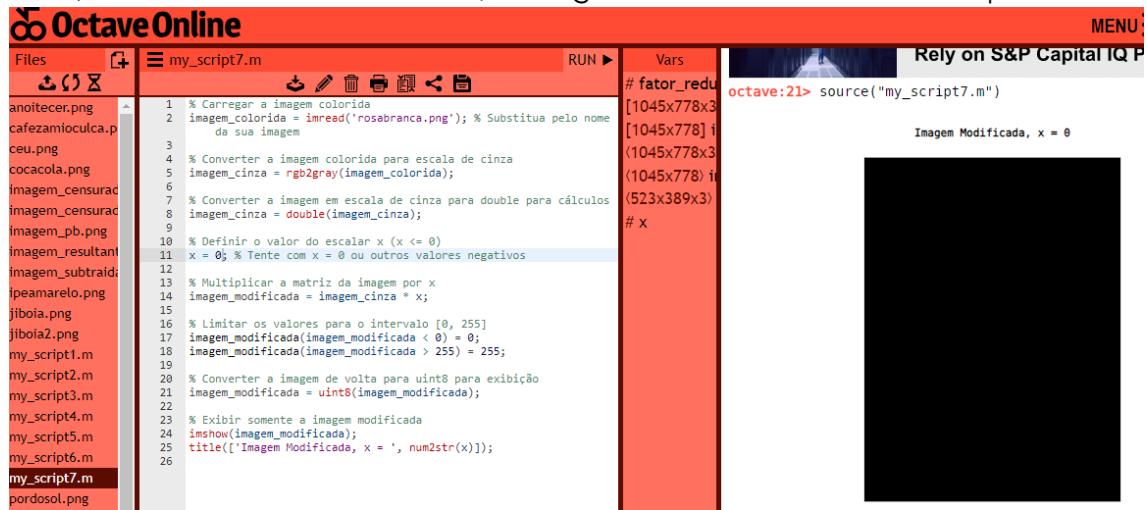
```
imagem_colorida = imread('rosabranca.png'); (carregamos a imagem)
imagem_cinza = rgb2gray(imagem_colorida); (convertemos a imagem em escala de cinza, se necessário)
imagem_cinza = double(imagem_cinza); (usamos o comando double para realizar cálculos)
x = -0.5; (definimos o valor do escalar  $x \leq 0$ )
```

```

imagem_modificada = imagem_cinza * x; (multiplicamos a matriz da imagem
por x)
imagem_modificada(imagem_modificada < 0) = 0; (limitamos os valores para o
intervalo [0,255])
imagem_modificada(imagem_modificada > 255) = 255; (limitamos os valores
para o intervalo [0,255])
imagem_modificada = uint8(imagem_modificada); (exibimos a imagem por meio
do comando uint8)
imshow(imagem_modificada); (exibimos a imagem modificada)
title(['Imagen Modificada, x = ', num2str(x)]);

```

b) Tomando o valor de $x = 0$, a imagem fica inteiramente na cor preta.



The screenshot shows the Octave Online interface. On the left, the 'Files' sidebar lists various image files and the script 'my_script7.m'. The main workspace shows the script code and its execution results. The command `source("my_script7.m")` is run, and the output shows the variable `imagem_modificada` is a 1045x778x3 grayscale image. The command `imshow(imagem_modificada)` is run, and the resulting image is a solid black square.

```

my_script7.m
1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png'); % Substitua pelo nome
   da sua imagem
3
4 % Converter a imagem colorida para escala de cinza
5 imagem_cinza = rgb2gray(imagem_colorida);
6
7 % Converter a imagem em escala de cinza para double para cálculos
8 imagem_cinza = double(imagem_cinza);
9
10 % Definir o valor do escalar x (x <= 0)
11 x = 0; % Tente com x = 0 ou outros valores negativos
12
13 % Multiplicar a matriz da imagem por x
14 imagem_modificada = imagem_cinza * x;
15
16 % Limitar os valores para o intervalo [0, 255]
17 imagem_modificada(imagem_modificada < 0) = 0;
18 imagem_modificada(imagem_modificada > 255) = 255;
19
20 % Converter a imagem de volta para uint8 para exibição
21 imagem_modificada = uint8(imagem_modificada);
22
23 % Exibir somente a imagem modificada
24 imshow(imagem_modificada);
25 title(['Imagen Modificada, x = ', num2str(x)]);
26

```

Código:

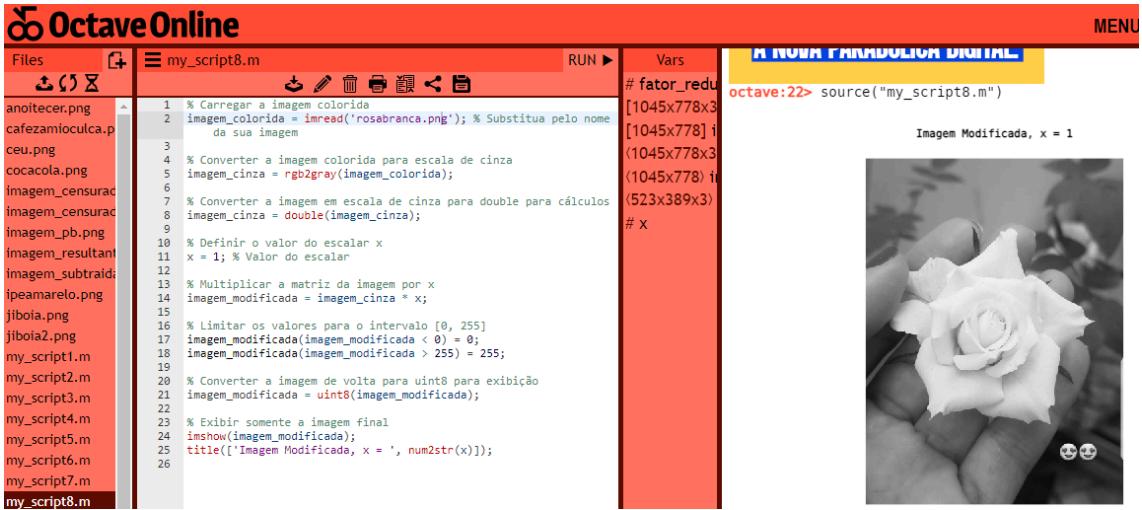
```

imagem_colorida = imread('rosabranca.png'); (carregamos a imagem)
imagem_cinza = rgb2gray(imagem_colorida); (convertemos a imagem para
escala de cinza, se necessário)
imagem_cinza = double(imagem_cinza); (usamos o comando double para
realizar cálculos)
x = 0; (definimos o valor do escalar x)
imagem_modificada = imagem_cinza * x; (multiplicamos a matriz da imagem
pelo escalar x)
imagem_modificada(imagem_modificada < 0) = 0; (limitamos os valores para o
intervalo [0,255])
imagem_modificada(imagem_modificada > 255) = 255; (limitamos os valores
para o intervalo [0,255])
imagem_modificada = uint8(imagem_modificada); (usamos o comando uint8
para exibir a imagem)
imshow(imagem_modificada); (exibimos a imagem modificada)
title(['Imagen Modificada, x = ', num2str(x)]);

```

Assim, ao multiplicarmos uma imagem em escala de cinza por um escalar $x \leq 0$, teremos como resultado, uma imagem na cor preto.

c) O que acontece se $x = 1$?



The screenshot shows the Octave Online interface. On the left, a file list shows various image files. The central area contains the script code 'my_script8.m'. The code reads a white rose image, converts it to grayscale, and then multiplies it by 1. The output window shows the result of the multiplication and the title 'Imagen Modificada, x = 1'.

```

Octave Online
my_script8.m
1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png'); % Substitua pelo nome
   da sua imagem
3
4 % Converter a imagem colorida para escala de cinza
5 imagem_cinza = rgb2gray(imagem_colorida);
6
7 % Converter a imagem em escala de cinza para double para cálculos
8 imagem_cinza = double(imagem_cinza);
9
10 % Definir o valor do escalar x
11 x = 1; % Valor do escalar
12
13 % Multiplicar a matriz da imagem por x
14 imagem_modificada = imagem_cinza * x;
15
16 % Limitar os valores para o intervalo [0, 255]
17 imagem_modificada(imagem_modificada < 0) = 0;
18 imagem_modificada(imagem_modificada > 255) = 255;
19
20 % Converter a imagem de volta para uint8 para exibição
21 imagem_modificada = uint8(imagem_modificada);
22
23 % Exibir somente a imagem final
24 imshow(imagem_modificada);
25 title(['Imagen Modificada, x = ', num2str(x)]);
26

```

Código:

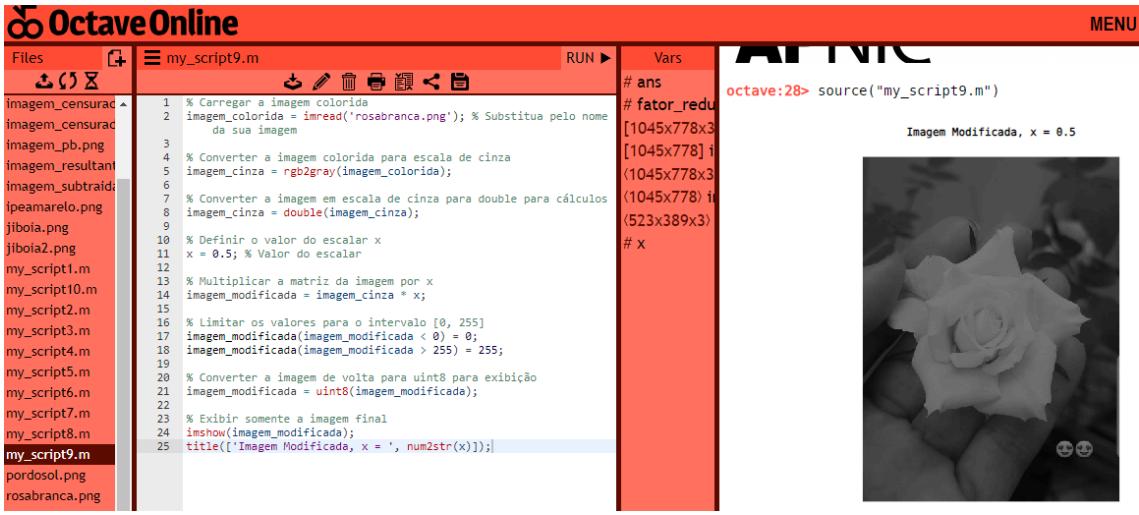
```

imagem_colorida = imread('rosabranca.png'); % (carregamos a imagem)
imagem_cinza = rgb2gray(imagem_colorida); (convertemos a imagem em escala
de cinza, se necessário)
imagem_cinza = double(imagem_cinza); (usamos o comando double para
realizar cálculos)
x = 1; (definimos o valor do escalar x)
imagem_modificada = imagem_cinza * x; (multiplicamos a matriz da imagem por
x)
imagem_modificada(imagem_modificada < 0) = 0; (limitamos os valores para o
intervalo [0,255])
imagem_modificada(imagem_modificada > 255) = 255; (limitamos os valores
para o intervalo [0,255])
imagem_modificada = uint8(imagem_modificada); (usamos o comando uint8
para exibir a imagem)
imshow(imagem_modificada); (exibimos a imagem final)
title(['Imagen Modificada, x = ', num2str(x)]);

```

Multiplicar uma imagem em escala de cinza por um escalar $x = 1$ mantém a imagem inalterada.

d) O que acontece se $0 < x < 1$?



The screenshot shows the Octave Online interface. On the left, a file list shows various image files and the script 'my_script9.m'. The main area displays the script code:

```

1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png'); % Substitua pelo nome
   da sua imagem
3
4 % Converter a imagem colorida para escala de cinza
5 imagem_cinza = rgb2gray(imagem_colorida);
6
7 % Converter a imagem em escala de cinza para double para cálculos
8 imagem_cinza = double(imagem_cinza);
9
10 % Definir o valor do escalar x
11 x = 0.5; % Valor do escalar
12
13 % Multiplicar a matriz da imagem por x
14 imagem_modificada = imagem_cinza * x;
15
16 % Limitar os valores para o intervalo [0, 255]
17 imagem_modificada(imagem_modificada < 0) = 0;
18 imagem_modificada(imagem_modificada > 255) = 255;
19
20 % Converter a imagem de volta para uint8 para exibição
21 imagem_modificada = uint8(imagem_modificada);
22
23 % Exibir somente a imagem final
24 imshow(imagem_modificada);
25 title(['Imagem Modificada, x = ', num2str(x)]);

```

On the right, the output shows the result of running the script, titled 'Imagem Modificada, x = 0.5'. The image is a grayscale rose, but it is significantly darker than the original, illustrating the effect of multiplying by a value between 0 and 1.

Código:

```

imagem_colorida = imread('rosabranca.png'); (carregamos a imagem)
imagem_cinza = rgb2gray(imagem_colorida); (convertemos a imagem para
escala de cinza, se necessário)
imagem_cinza = double(imagem_cinza); (usamos o comando double para
realizar os cálculos)
x = 0.5; (definimos o valor do escalar x)
imagem_modificada = imagem_cinza * x; (multiplicamos o valor do escalar x pela
matriz da imagem)
imagem_modificada(imagem_modificada < 0) = 0; (limitamos os valores para o
intervalo [0, 255])
imagem_modificada(imagem_modificada > 255) = 255; (limitamos os valores
para o intervalo [0, 255])
imagem_modificada = uint8(imagem_modificada); (usamos o comando uint8
para exibir a imagem)
imshow(imagem_modificada); (exibimos a imagem final)
title(['Imagem Modificada, x = ', num2str(x)]);

```

Quando multiplicamos uma imagem por um número real maior do que zero e menor do que 1, realizamos um escurecimento na imagem, ou seja, o brilho da imagem é reduzido tornando-a mais escura.

e) O que acontece se $x > 1$?

The screenshot shows the Octave Online interface. On the left, a file browser lists various image files. The main area is a script editor with the following code:

```

1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png'); % Substitua pelo nome
   da sua imagem
3
4 % Converter a imagem colorida para escala de cinza
5 imagem_cinza = rgb2gray(imagem_colorida);
6
7 % Converter a imagem em escala de cinza para double para cálculos
8 imagem_cinza = double(imagem_cinza);
9
10 % Definir o valor do escalar x
11 x = 1.5; % Valor do escalar
12
13 % Multiplicar a matriz da imagem por x
14 imagem_modificada = imagem_cinza * x;
15
16 % Limitar os valores para o intervalo [0, 255]
17 imagem_modificada(imagem_modificada < 0) = 0;
18 imagem_modificada(imagem_modificada > 255) = 255;
19
20 % Converter a imagem de volta para uint8 para exibição
21 imagem_modificada = uint8(imagem_modificada);
22
23 % Exibir somente a imagem final
24 imshow(imagem_modificada);
25 title(['Imagen Modificada, x = ', num2str(x)]);

```

The command window shows the output of the script: "octave:27> source("my_script9.m")" and "Imagen Modificada, x = 1.5". To the right, a preview image shows a white rose flower held in a hand, with a small emoji at the bottom right.

Código:

```

imagem_colorida = imread('rosabranca.png'); (carregamos a imagem)
imagem_cinza = rgb2gray(imagem_colorida); (convertemos a imagem colorida
para escala de cinza)
imagem_cinza = double(imagem_cinza); (usamos o comando double para
realizar os cálculos)
x = 1.5; % Valor do escalar (definimos o valor do escalar x)
imagem_modificada = imagem_cinza * x; (multiplicamos a matriz da imagem por
x)
imagem_modificada(imagem_modificada < 0) = 0; (limitamos os valores para o
intervalo [0, 255])
imagem_modificada(imagem_modificada > 255) = 255; (limitamos os valores
para o intervalo [0, 255])
imagem_modificada = uint8(imagem_modificada); (usamos o comando uint8
para exibir a imagem)
imshow(imagem_modificada); (exibimos a imagem final)
title(['Imagen Modificada, x = ', num2str(x)]);

```

Quando o número real x assume valores maiores do que 1, temos um aumento no brilho da imagem, tornando-a mais clara.

Atividade 3: Considere as matrizes A e B que representam as letras L e T, respectivamente. Faça a multiplicação entre as matrizes e converta para uma imagem binária. Responda:

- Há alguma relação entre a imagem resultante com as imagens L e T? Justifique sua resposta.

```

>> A=[0,1,1,1,1;0,1,1,1,1;0,1,1,1,1;0,1,1,1,1;0,0,0,0,1]
A =
 0   1   1   1   1
 0   1   1   1   1
 0   1   1   1   1
 0   1   1   1   1
 0   0   0   0   1

>> imshow(A)
>> B=[0,0,0,0,0;1,1,0,1,1;1,1,0,1,1;1,1,0,1,1;1,1,0,1,1]
B =
 0   0   0   0   0
 1   1   0   1   1
 1   1   0   1   1
 1   1   0   1   1
 1   1   0   1   1

>> imshow(B)
>> A*B
ans =
 4   4   0   4   4
 4   4   0   4   4
 4   4   0   4   4
 4   4   0   4   4
 1   1   0   1   1

>> imshow(A*B)

```



Primeiramente, temos que observar se o produto entre as duas matrizes é possível. Para que seja, o número de colunas da primeira matriz tem que ser igual ao número de linhas da segunda matriz. Em nosso caso o produto é possível, porém, a imagem resultante do produto não tem nenhuma conexão com o conteúdo das imagens L e T.

Isso ocorre porque nem todas as imagens podem ser multiplicadas entre si (devido a incompatibilidade do tamanho das matrizes para efetuar o produto), e também, se ambas as matrizes possuírem valores binários 0 e 1, é muito provável que o produto exceda o 1, fornecendo uma imagem totalmente branca. O mesmo ocorre se tomarmos valores entre 0 e 255 em escala de cinza.

Aula 5: Efeito de zoom e fechamento da oficina com entrevista e questionário.

Atividade 1: Carregar uma imagem e aplicar o efeito zoom nessa imagem.

```

Octave Online
File my_script1.m RUN ▶ Vars R$19,499 R$20,735
1 % Carregar a imagem colorida
2 imagem_colorida = imread('rosabranca.png');
3
4 % Converter a imagem para escala de cinza
5 A = rgb2gray(imagem_colorida);
6
7 % Obter as dimensões da imagem
8 [n_linhas, n_colunas] = size(A);
9
10 % Definir o fator de zoom (exemplo: 2x zoom)
11 fator_zoom = 2;
12
13 % Calcular as novas dimensões para o recorte central
14 nova_altura = round(n_linhas / fator_zoom);
15 nova_largura = round(n_colunas / fator_zoom);
16
17 % Calcular as coordenadas do recorte central
18 inicio_linha = round((n_linhas - nova_altura) / 2);
19 inicio_coluna = round((n_colunas - nova_largura) / 2);
20
21 % Fazer o recorte central da imagem
22 imagem_recortada = A(inicio_linha:(inicio_linha + nova_altura - 1), ...
23 [ ] [ ] [ ] inicio_coluna:(inicio_coluna + nova_largura - 1));
24
25 % Redimensionar a imagem recortada para o tamanho original
26 imagem_zoom = imresize(imagem_recortada, [n_linhas, n_colunas]);
27
28 % Exibir a imagem original e a imagem com zoom
29 subplot(1, 2, 1);
30 imshow();
31 title('Imagen Original');

```

Atividade 2: Aplicação de questionário,

Após realização de mais algumas atividades envolvendo imagens e matrizes, realizamos a aplicação de questões envolvendo as atividades da oficina além de um debate para que os participantes pudessem expor pontos de vista, opiniões e sugestões acerta da temática e sua aplicação em sala de aula.

Questões discutidas e respondidas pelos participantes:

- 1) Você já teve contato com o software GNU Octave antes da oficina?
- 2) O uso de uma aplicação envolvendo imagens ajudou você compreender melhor os conceitos de matrizes?
- 3) Foi fácil utilizar o software Octave na versão online?
- 4) As explicações para utilizar o software Octave foram claras?
- 5) A prática com o software Octave ajudou a compreender melhor os conceitos de matrizes?
- 6) As atividades práticas com o software Octave foram úteis?
- 7) O software Octave fez você se interessar mais por matrizes e imagens?
- 8) A prática com o software Octave fez com que o estudo de matrizes se tornasse mais interessante ou desafiador para você?
- 9) Você gostaria que o software Octave tivesse sido utilizado como recurso nas aulas regulares em sala de aula, para auxiliar no aprendizado de conteúdos como matrizes por exemplo?

