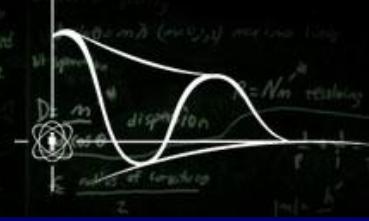


# X Escola do CBPF

## 13 a 24 de julho de 2015



# Processamento de Imagens

**Prof. Marcelo Portes de Albuquerque**  
**Prof. Márcio Portes de Albuquerque**

**Colaboradores:** Clécio R. De Bom (CBPF)  
**André Persechino (CBPF)**

**X Escola do CBPF – 2015**

# Organização do Curso

---

**G14 - Aulas – 13 a 24 de Julho**

**Seg. – Introdução a Análise de Imagens**

**Ter . – Segmentação e Morfologia Matemática**

**Qua. – Reconhecimento de Padrões**

**Qui. – Laboratório (ImageJ e MATLAB)**

**Sex. – Laboratório (MATLAB)**

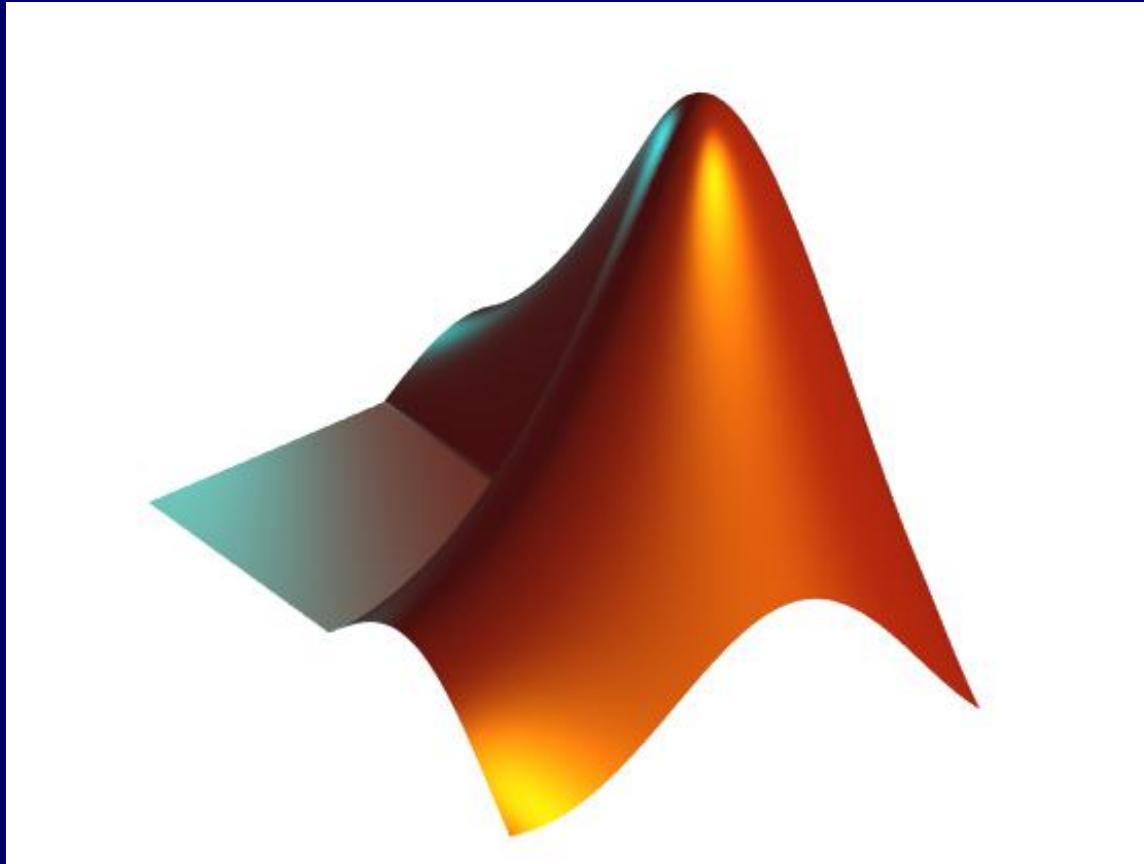
**Site Web:**

Notas de Aula: <http://mesonpi.cat.cbpf.br/escola2015>

# Informações X Escola CBPF

---

- A Escola fornecerá certificado de **frequência** aos alunos inscritos que tiverem assistido **no mínimo 8 horas** de aula de um determinado curso de graduação, e no mínimo 12 horas, no caso dos cursos de pós-graduação.
- Além disto, a Escola **exige freqüência integral** dos estudantes que **beneficiam de auxílio financeiro** (hospedagem e alimentação), em 2 cursos no caso dos estudantes de graduação e em 3 cursos, dos estudantes de pós-graduação.
- Pelas duas razões acima, é necessário que os alunos **assinem** as listas de presença a cada aula.



# **Tópicos em Processamento de Imagens com MATLAB®**

**André Persechino**

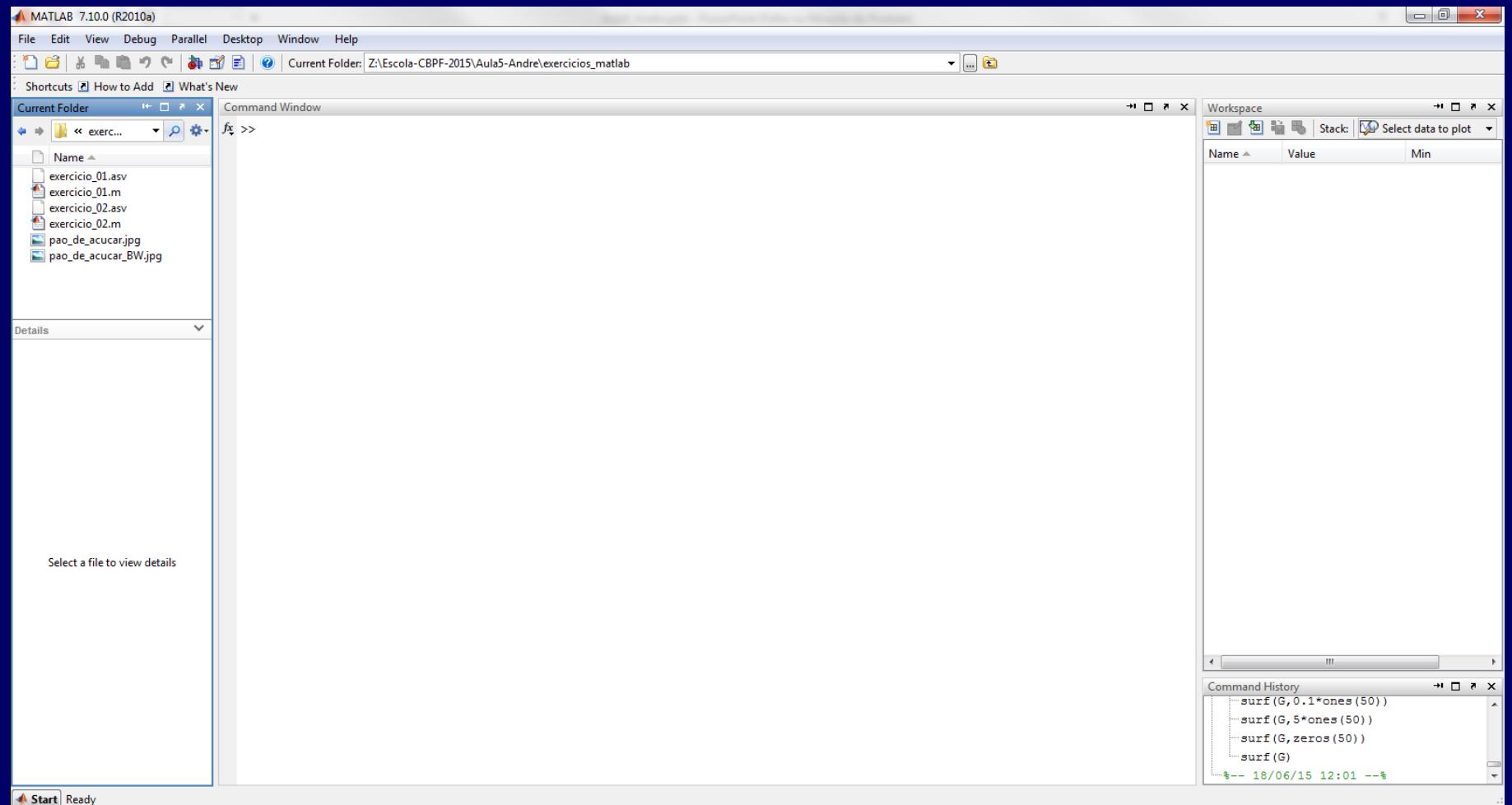
aamerico@cbpf.br

X Escola do CBPF - Julho 2015

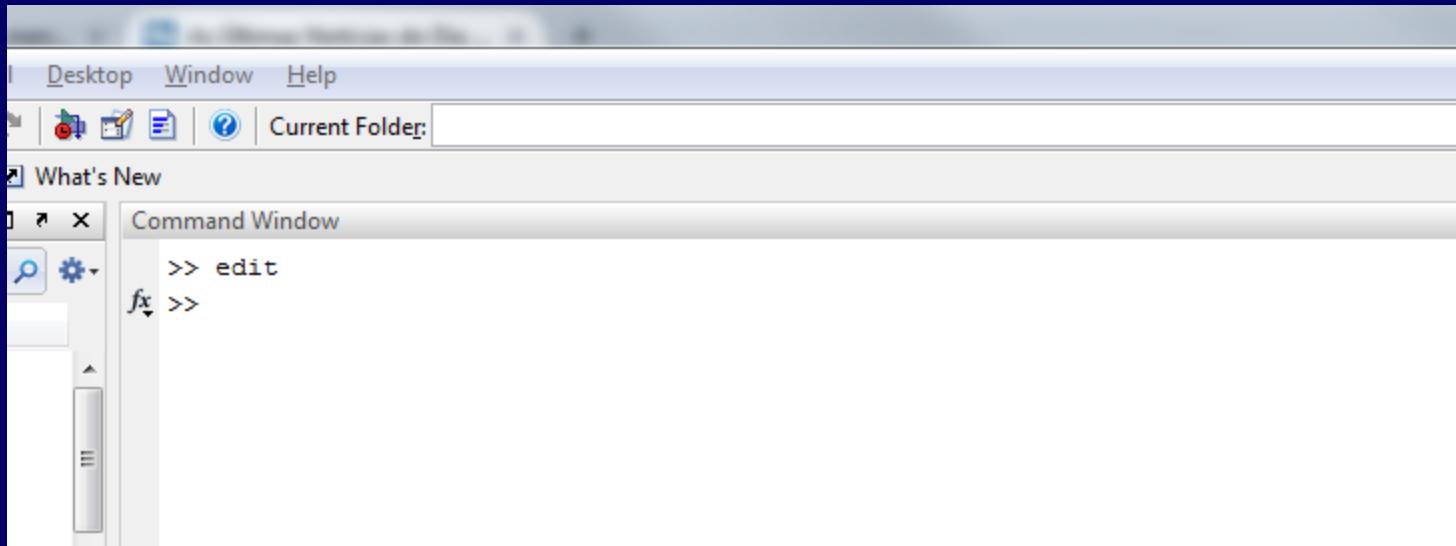
# Tópicos em PDI com MATLAB

- Noções preliminares
  - Entrada de imagens no workspace
  - Manipulação básica das entradas da imagem
  - obtenção de histograma
- Filtragem
  - FFT e filtragem no domínio das frequências
  - Filtragem no domínio espacial

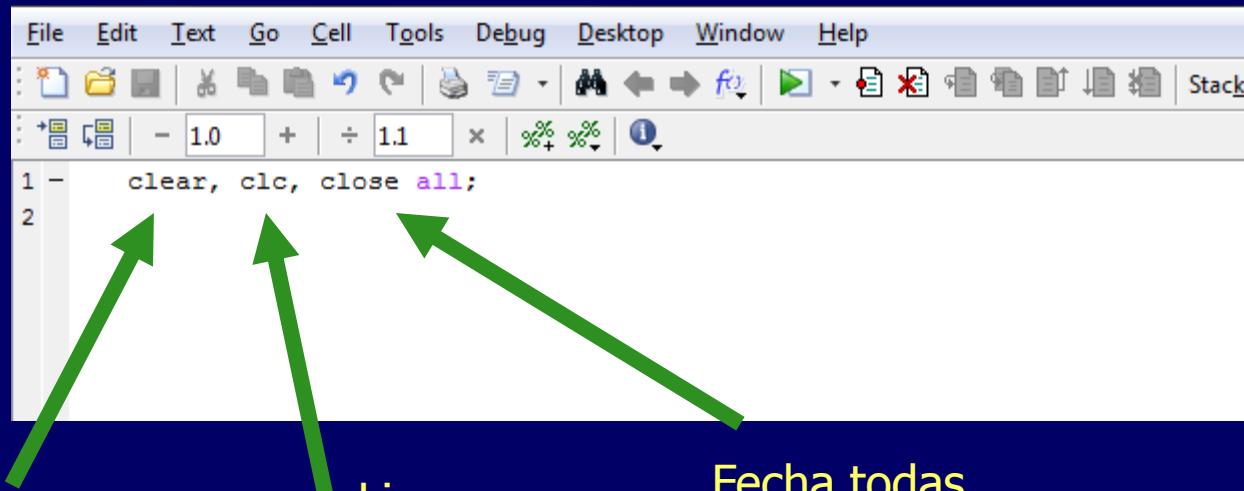
# Noções preliminares



- Embora seja possível programar no workspace, é preferível sempre usar o editor.



- Como boa prática de programação em MATLAB, devemos sempre:
  - Aniquilar quaisquer variáveis existentes;
  - Fechar quaisquer figuras abertas;
  - Limpar o workspace.



Aniquila todas as variáveis armazenadas

Limpa o workspace

Fecha todas as janelas

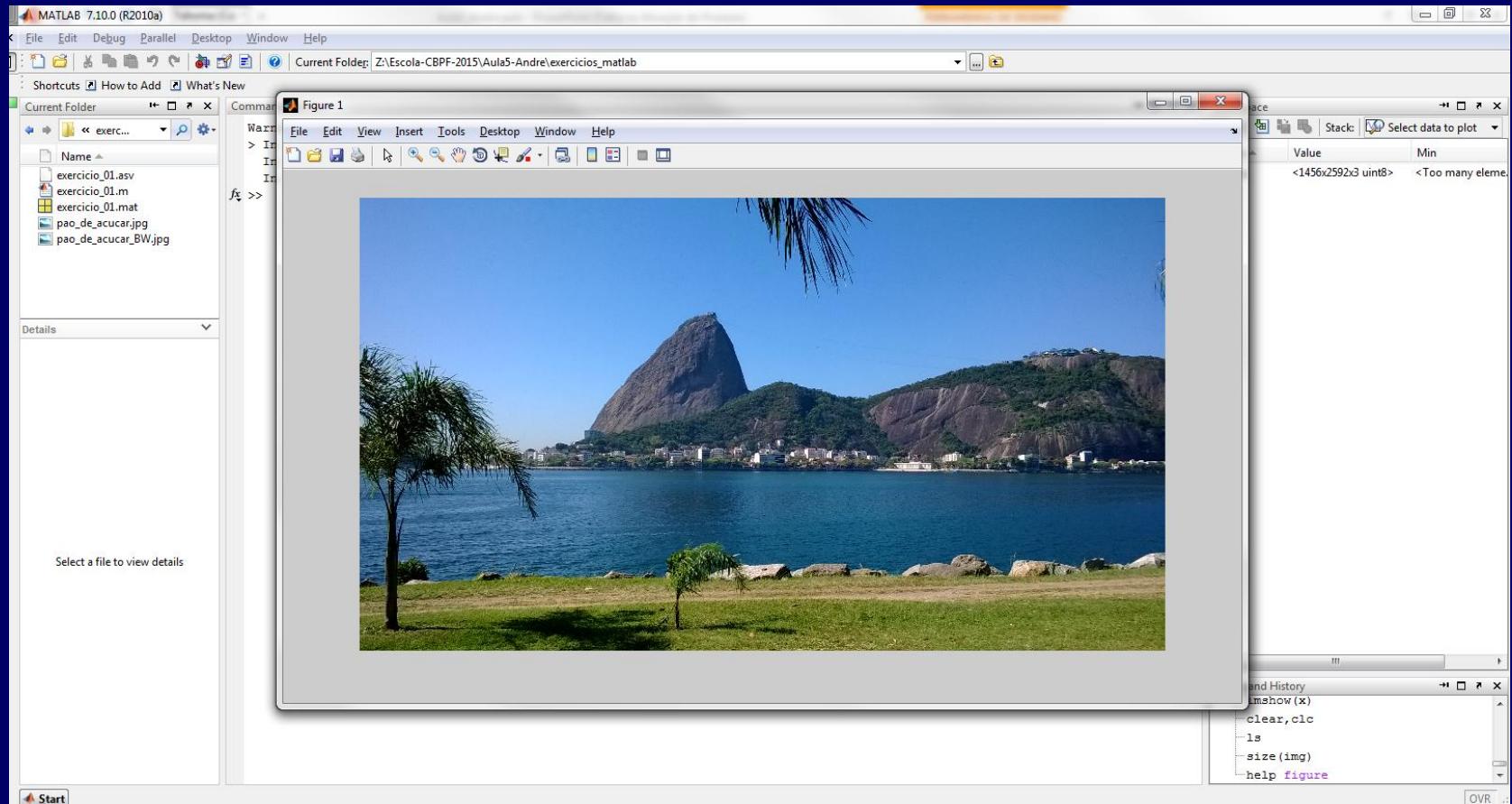
- Para abrir a nossa imagem teste usamos o comando `imread()`, cujo argumento de entrada é uma string com o nome do arquivo.

**TAREFA 01:**

abra o editor, inicie-o com o escopo mostrado e abra a imagem `'pao_de_acucar.jpg'`.

```
clear, clc, close all;  
  
img = imread('pao_de_acucar.jpg');
```

- Para exibir a imagem carregada, usamos o comando `imshow()`:



Com a imagem no workspace, podemos começar a trabalhar sobre ela.

Mas antes de tudo, é preciso conhecê-la...

Queremos responder as seguintes perguntas sobre img:

- Quais suas dimensões?
- Qual o intervalo de quantização usado?
- Qual a forma de seu histograma?

- As dimensões espaciais e o intervalo de quantização usado podem ser obtidos por meio do comando `whos()`, cujo argumento de entrada é uma *string* com o nome da variável.

```
clear, clc, close all;

img = imread('pao_de_acucar.jpg');

figure; imshow(img);

% ----- OBTENCAO DAS INFORMACOES BASICAS DA IMAGEM -----
info    = whos('img');

disp(info);
```

```
name: 'img'  
    size: [1456 2592 3]  
    bytes: 11321856  
    class: 'uint8'  
    global: 0  
    sparse: 0  
    complex: 0  
    nesting: [1x1 struct]  
    persistent: 0
```

- O acesso às entradas da imagem/matriz ocorrem por meio do operador dois-pontos:

```
img (Li:Lf, Ci:Cf, Chi:Chf)
```

**OBSERVAÇÃO:**

Para acessar *todas* as linhas, colunas ou canais, usa-se : sem definir os extremos do intervalo.

**Exemplos:**

```
>> img(:,1:100,1);  
  
>> img(5:505,:,:,:);  
  
>> img(:,:,:,:);
```

- O histograma pode ser visualizado por meio da função `imhist()`, cujo parâmetro de entrada é a imagem a ser analisada.

**OBSERVAÇÃO:**

O histograma deve ser obtido para cada canal da imagem, individualmente.

```
clear, clc, close all;

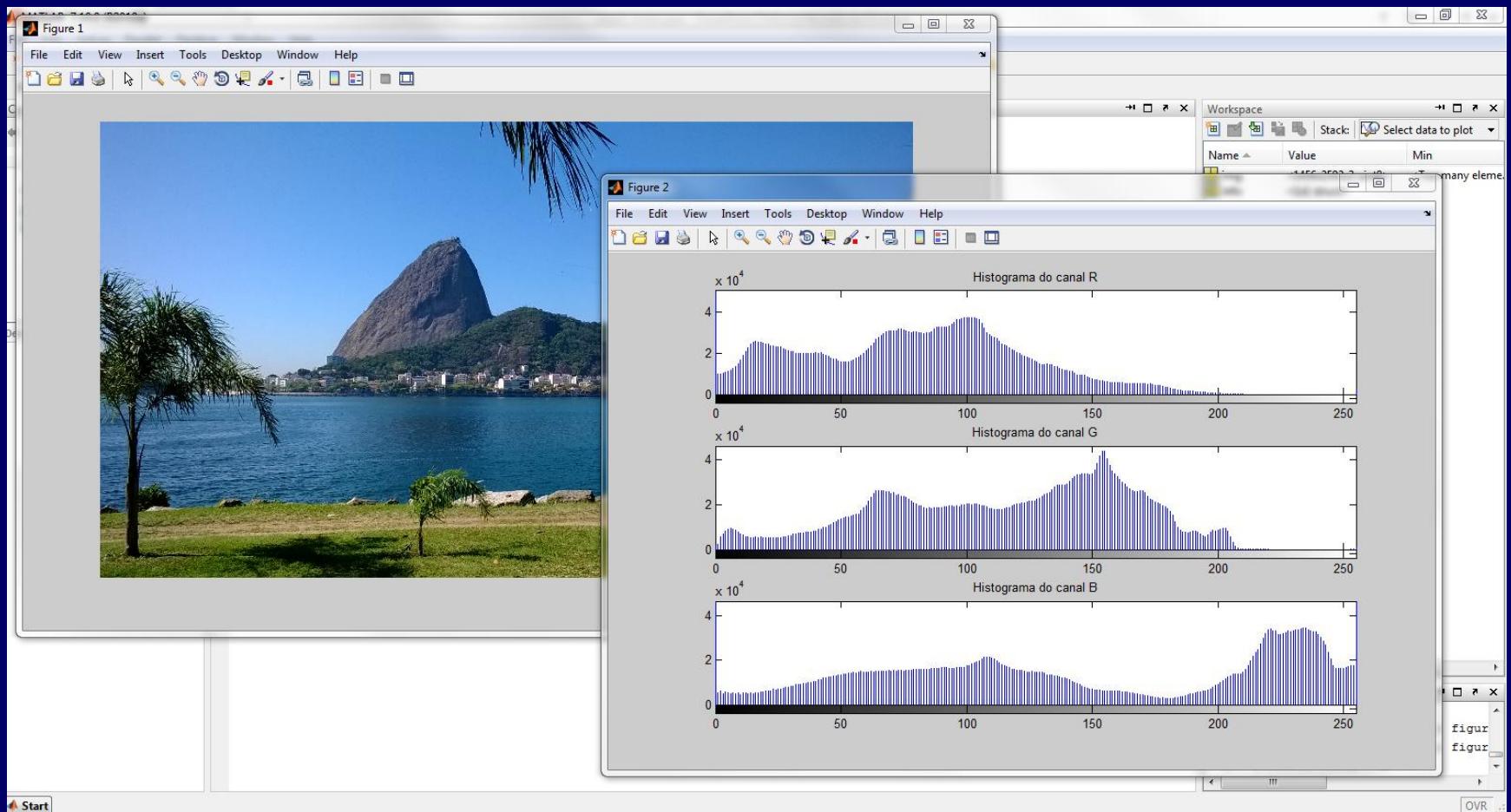
img = imread('pao_de_acucar.jpg');

figure; imshow(img);

% ----- OBTENCAO DAS INFORMACOES BASICAS DA IMAGEM -----
info = whos('img');

disp(info);

% % ----- OBTENCAO DO HISTOGRAMA DA IMAGEM -----
figure;
subplot(3,1,1)
imhist(img(:,:,1)); title('Histograma do canal R');
subplot(3,1,2)
imhist(img(:,:,2)); title('Histograma do canal G');
subplot(3,1,3)
imhist(img(:,:,3)); title('Histograma do canal B');
```



# Filtragem

A base da filtragem no domínio das frequências consiste a obtenção da transformada discreta de Fourier da imagem e do filtro, bem como suas inversas:

$$F(k_x, k_y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi j \left( \frac{k_x}{M} x + \frac{k_y}{N} y \right)}$$

DFT

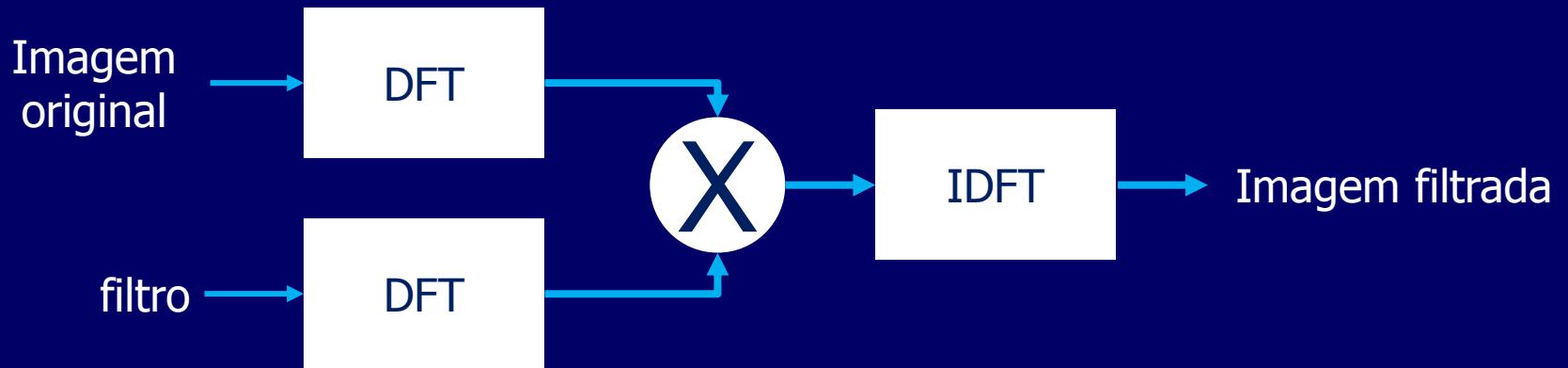
$$f(x, y) = \frac{1}{MN} \sum_{k_x=0}^{M-1} \sum_{k_y=0}^{N-1} F(k_x, k_y) e^{2\pi j \left( \frac{x}{M} k_x + \frac{y}{N} k_y \right)}$$

IDFT

$$f(x, y) * h(x, y) \leftrightarrow F(k_x, k_y) H(k_x, k_y)$$

Teorema da convolução

# Dinâmica básica da filtragem no domínio das frequências



- A DFT de uma imagem pode ser obtida por meio da função `fft2()`.
- A IDFT pode ser obtida por meio da função `ifft2()`.

**OBSERVAÇÃO:**

Antes de tomar a `fft2` da imagem, é preciso convertê-la para o formato `double`. Isso é feito por meio do comando `im2double()`.

## TAREFA 02 :

- [1] Initialize um novo programa chamado `estudo_fft.m`, em que seja aberta a imagem `pao_de_acucar_BW.jpg`. Converta-a para o formato `double`.
- [2] Obtenha a DFT da imagem.
- [3] Obtenha a IDFT da DFT da imagem e a exiba na tela.

```
clear, clc, close all;  
  
img = imread('pao_de_acucar_BW.jpg');  
img = im2double(img);  
  
FT = fft2(img);  
IFT = ifft2(FT);  
  
figure; imshow(IFT);
```

- Podemos visualizar o espectro de potência da imagem:

$$|F(k_x, k_y)| = \sqrt{\Re(F(k_x, k_y))^2 + \Im(F(k_x, k_y))^2}.$$

- Para isso, fazemos uso da função `abs()`.

```
clear, clc, close ;
imshow(log(SPEC+1), []);  
img = imread('pao_de_acucar_BW.jpg');
img = im2double(img);  
  
FT = fft2(img);
SPEC = abs(FT);
  
figure; imshow(log(SPEC+1), []);
```

## OBSERVAÇÃO 1:

Para ser exibido de maneira mais perceptível, geralmente mostramos o *logaritmo* do espectro.

## OBSERVAÇÃO 2:

O que vemos na imagem do espectro corresponde a uma versão não-centralizada dele. Para visualizar o espectro centralizado, usamos a função `fftshift()` sobre a DFT.



```

clear, clc, close all;

img = imread('pao_de_acucar_BW.jpg');
img = im2double(img);

% OBTENCAO DAS DIMENSOES DA IMAGEM
s = size(img);
% OBTENCAO DA MAIOR DIMENSAO DA IMAGEM
DimMax = max(s);

% ESTIMATIVA DA MENOR POTENCIA DE 2 MAIOR QUE DimMax
M = log2(DimMax);
M = 2^ceil(M);

% OBTENCAO DAS DFTs E DOS ESPECTROS DE POTENCIA
FT = fft2(img,M,M);
IFT = ifft2(FT);

SPEC_img = abs(fftshift(FT));

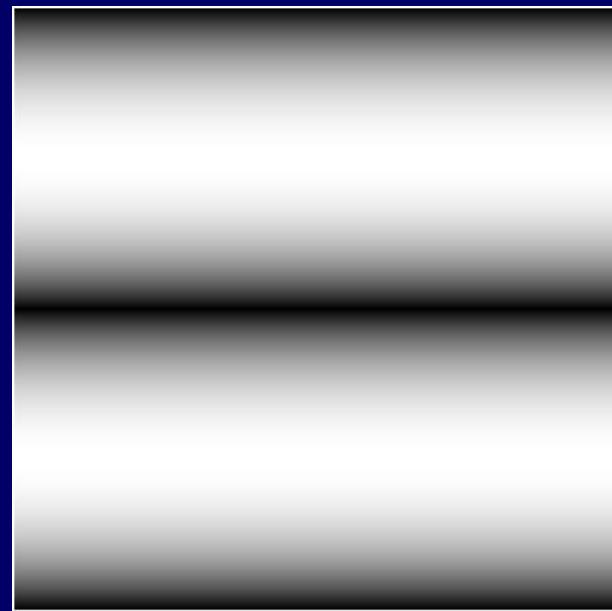
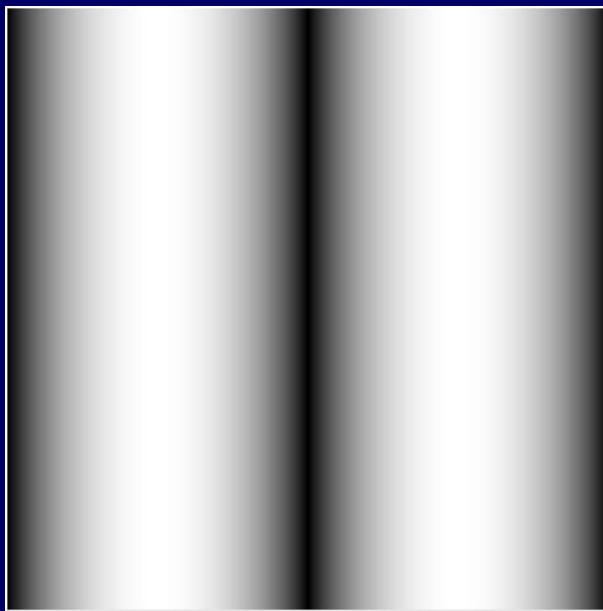
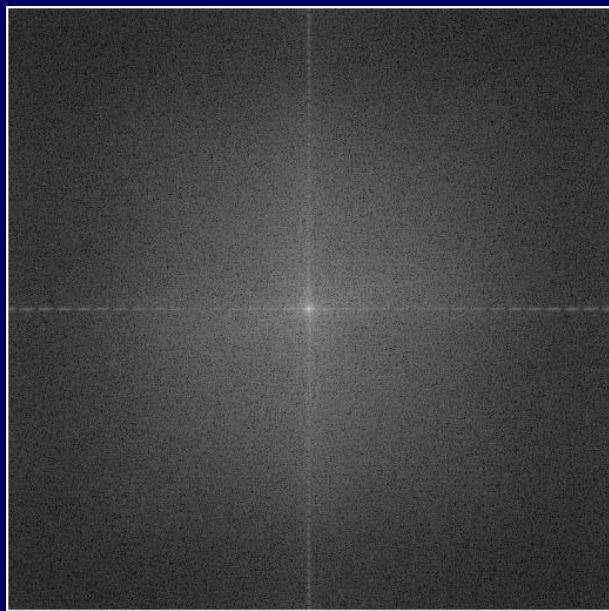
figure; imshow(log(SPEC_img+1),[]); title('Espectro da imagem');
figure; imshow(IFT,[]); title('IDFT');

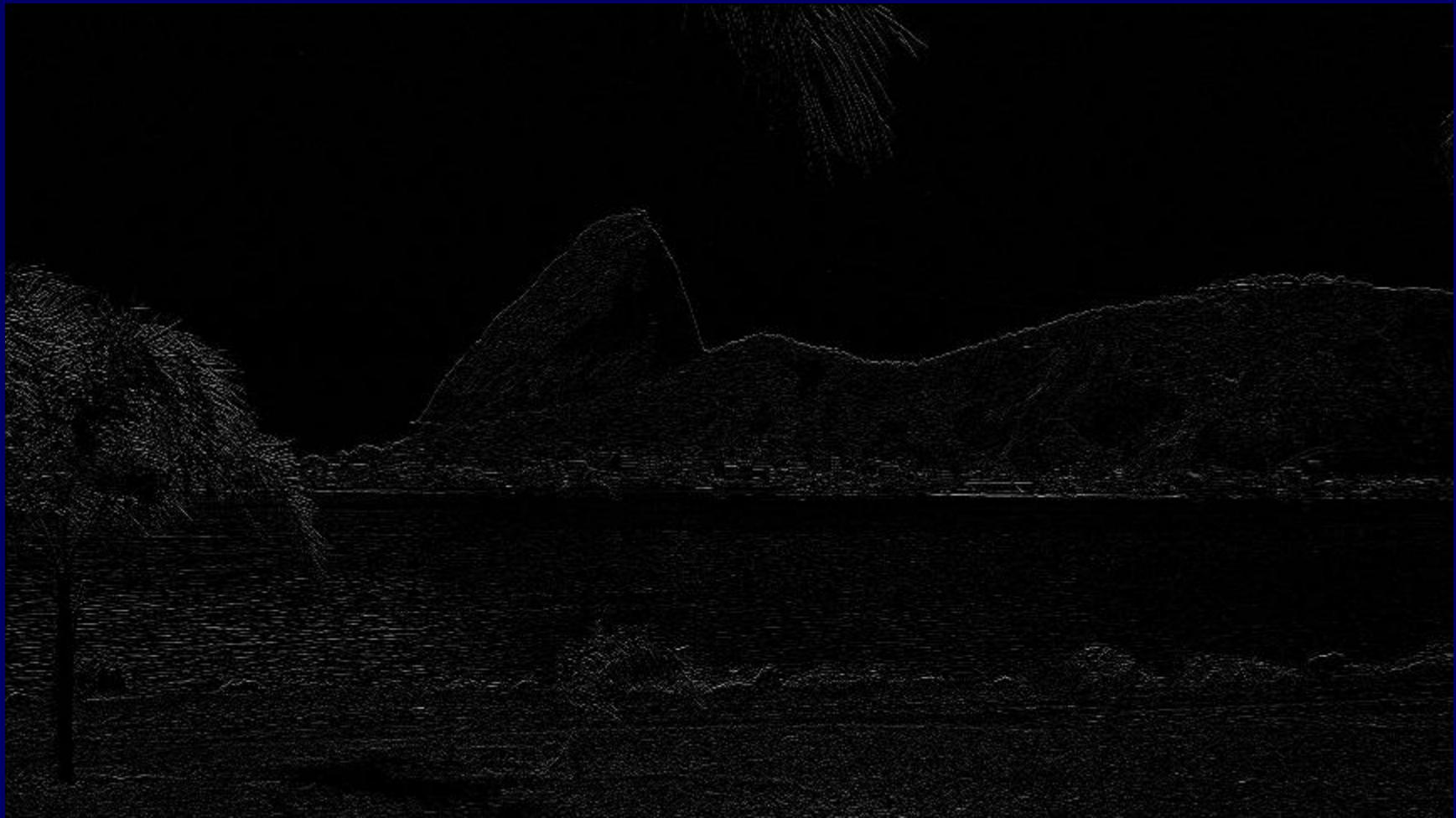
```



**TAREFA:**

- [1] Abra a imagem `pao_de_acucar_BW.jpg` e a converta em double.
- [2] Construa os filtros diferenciadores  $h = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$  e  $v = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ .
- [3] Obtenha os espectros (centralizados) da imagem e dos dois filtros.
- [4] Realize o produto ponto-a-ponto das DFTs da imagem e dos filtros.
- [5] Obtenha as IDFTs dos produtos e verifique a ação dos filtros. O que ocorreu?







- Os filtros  $h$  e  $v$  detectam bordas nas duas direções da imagem.
- Podemos estimar a *magnitude do gradiente a imagem*, obtendo assim informações sobre suas bordas:

$$\|\vec{\nabla}f(x, y)\| = \sqrt{(\partial_x f(x, y))^2 + (\partial_y f(x, y))^2}$$



```

clear, clc, close all;

img = imread('pao_de_acucar_BW.jpg');
img = im2double(img);

% DEFINICAO DOS FILTROS
h = [0 0 0;-1 0 1;0 0 0]; v = [0 -1 0;0 0 0; 0 1 0];

% OBTENCAO DAS DIMENSOES DA IMAGEM
s = size(img);
% OBTENCAO DA MAIOR DIMENSAO DA IMAGEM
DimMax = max(s);

% ESTIMATIVA DA MENOR POTENCIA DE 2 MAIOR QUE DimMax
M = log2(DimMax);
M = 2^ceil(M);

% OBTENCAO DAS DFTs E DOS ESPECTROS DE POTENCIA
FT = fft2(img,M,M); FTH = fft2(h,M,M); FTV = fft2(v,M,M);

SPEC_img = abs(fftshift(FT));
SPEC_h = abs(fftshift(FTH));
SPEC_v = abs(fftshift(FTV));

figure; imshow(log(SPEC_img+1),[]); title('Espectro da imagem');
figure; imshow(log(SPEC_h+1),[]); title('Espectro de h');
figure; imshow(log(SPEC_v+1),[]); title('Espectro de v');

% FILTRAGEM REALIZADA PELO PRODUTO PONTO-A-PONTO DAS TRANSFORMADAS
Y_h = FT.*FTH; Y_v = FT.*FTV;

y_h = ifft2(Y_h); y_h = y_h(1:s(1),1:s(2));
y_v = ifft2(Y_v); y_v = y_v(1:s(1),1:s(2));

% ESTIMATIVA DA MAGNITUDE DO GRADIENTE
MagGrad = sqrt(y_v.^2+y_h.^2);

figure; imshow(y_h);
figure; imshow(y_v);
figure; imshow(MagGrad);

```

## TAREFA:

Defina novos filtros e realize o processo de filtragem no domínio das frequências tal como feito nos exemplos anteriores.

## EXEMPLOS DE FILTROS:

Laplaciano:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Média aritmética (raio  $n$ ):

$$M_n = \frac{1}{n^2} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{n \times n}$$

Gaussiano (raio  $n$  e desvio padrão  $\sigma$ ):

$$G_n = \frac{1}{2^{2n}} [g_{ij}] ; g_{ij} = e^{\left(-\frac{i^2+j^2}{2\sigma^2}\right)} ; \text{ com } i, j \in \left[-\frac{n-1}{2}, \frac{n-1}{2}\right]$$

- Vários filtros podem ser obtidos por meio do comando `fspecial()`, cujo argumento de entrada é uma string com o nome do filtro.

```
clear, clc, close all;

img = imread('pao_de_acucar_BW.jpg');
img = im2double(img);

L = fspecial('laplacian',0);
G = fspecial('gaussian',10,1);
P = fspecial('prewitt');
S = fspecial('sobel');
```

- Para filtros lineares, a filtragem da imagem pode ocorrer diretamente sobre o domínio espacial.

$$f(x, y) * h(x, y) \leftrightarrow F(k_x, k_y)H(k_x, k_y)$$

- Isso é feito por meio do comando `imfilter()`, cujos argumentos de entrada são a imagem e o filtro.

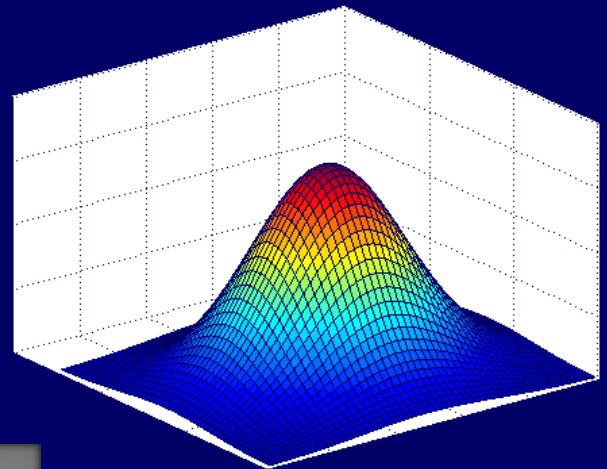
```
clear, clc, close all;

img = imread('pao_de_acucar_BW.jpg');
img = im2double(img);

G = fspecial('gaussian',50,10);

y = imfilter(img,G);

figure; imshow(y); title('Imagen filtrada');
figure; surf(G); title('Representacao do filtro');
```



- Obviamente, o MATLAB proporciona muito mais recursos em PDI.
- Há ferramentas para filtragem não-linear, filtros estatísticos, morfologia matemática etc.
- Aliado a todo esse suporte específico, há uma poderosa linguagem de programação para ambiente científico.

# ■ Alguns problemas atacados via PDI por nós no CBPF:

## ■ Caracterização de amostras geológicas

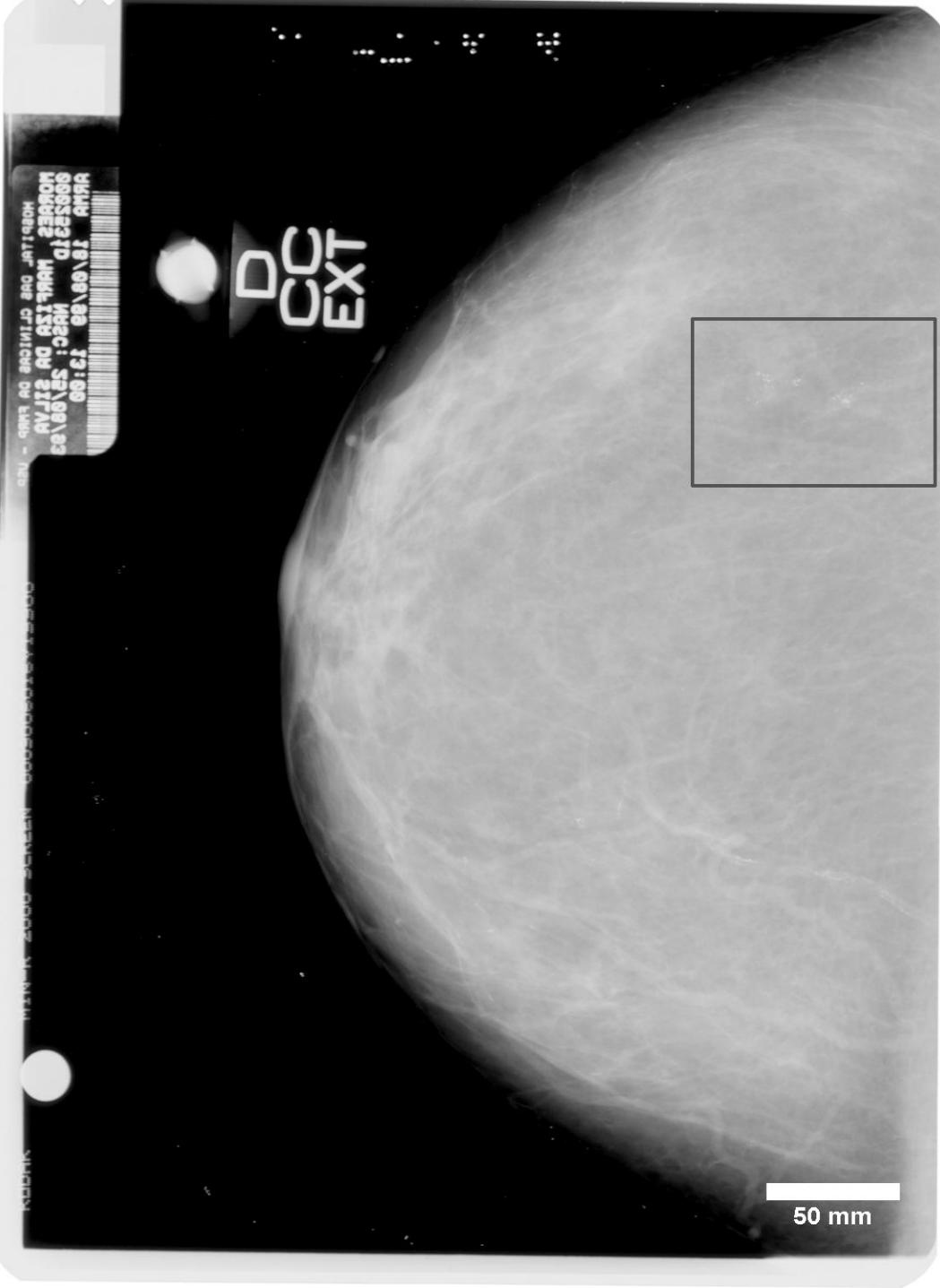
De Bom, C. R. et al. *Multiscale matching of micro-CT images using pattern recognition and Hu moments*. Notas Técnicas do CBPF. v. 4. n. 1. 2014.

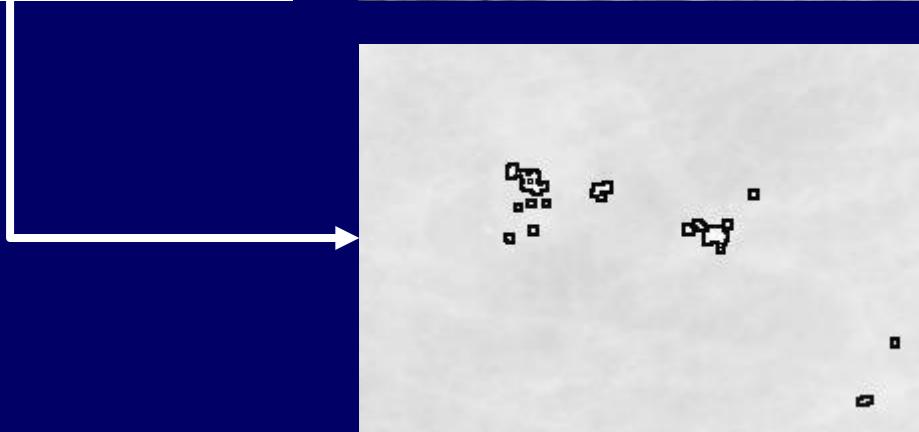
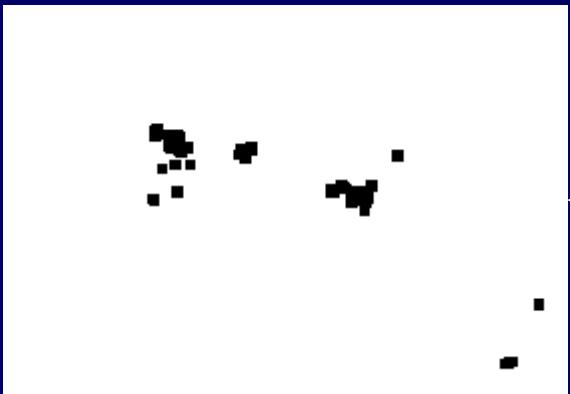
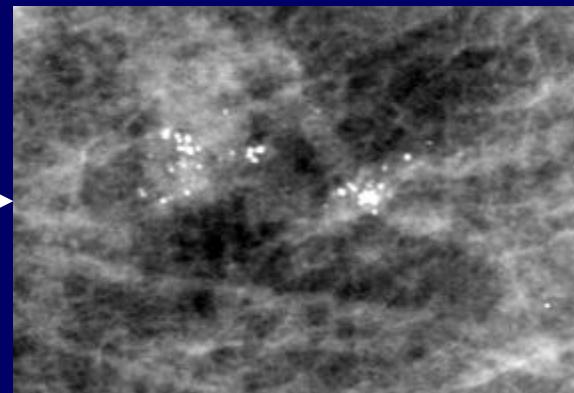
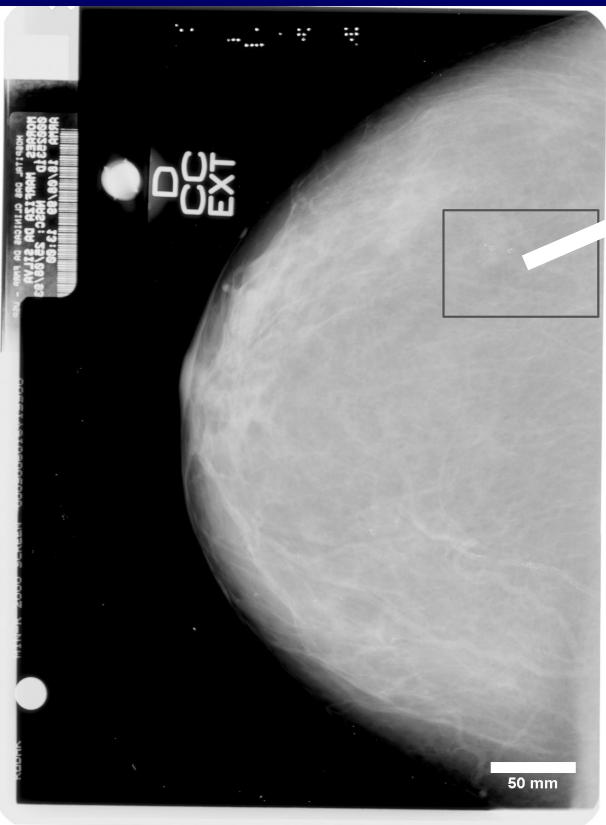
## ■ Processamento de imagens em hardware e aplicações a sequências de alta frequência de aquisição

da Silva, F. S. et al. *Desenvolvimento de ferramentas via FPGA para leitura e processamento de imagens em tempo real*. Notas Técnicas do CBPF. v. 4. n. 2. 2014.

## ■ Processamento de imagens mamográficas com vistas à detecção de estruturas suspeitas

Persechino, A. et al. *Evidenciação de estruturas em imagens por meio de equação de difusão*. Notas Técnicas do CBPF. v. 4. n. 2. 2014.





Se você tem interesse em trabalhar em nível de IC, mestrado ou doutorado nas áreas de

- Instrumentação científica em física;
- Física estatística / teoria da informação;
- Petrofísica;
- Imagens médicas,

entre em contato conosco.



## Laboratório de Processamento Digital de Sinais e Imagens

lpdsi@cbpf.br

Coordenação de Atividades Técnicas – CAT

Rua Dr. Xavier Sigaud, 150. 3º andar. Ala C.  
Urca, Rio de Janeiro – RJ. CEP: 22290-180