

### Atividade #03

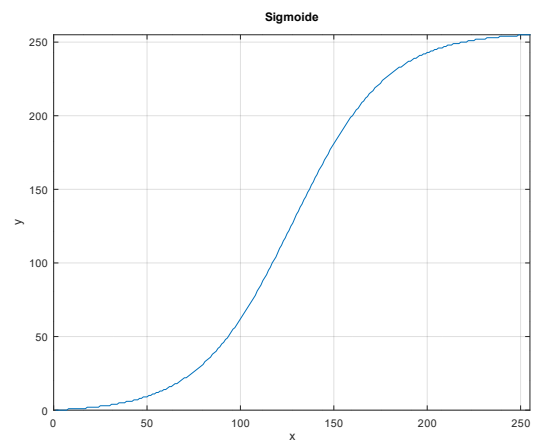
Vale nota, individual ou em dupla, observar prazo e instruções de entrega no moodle

#### Arquivos necessários

1. vpfig.png [adaptada de Volnei A. Pedroni, Eletrônica Digital Moderna e VHDL, 2020, Fig. 4.6]
2. gDSC04422m16.png

#### 3.1) Funções de transformação dos níveis de cinza (sigmoide usando intlut)

```
%Sigmoid
%Aloca uint8
%para depois usar funcao intlut (y1 é a LUT)
%Equação da sigmoide
slope = 0.05;
inflec = 127;
x = 0:1:255;
y1 = 1./(1 + exp(-slope*(x - inflec)));
y1n = mat2gray(y1);
y1n = uint8(y1n.*255);
%Display
figure, plot(y1n)
xlim([0 255]), ylim([0 255])
grid on
title('Sigmoide')
xlabel('x'), ylabel('y')
```



A função sigmoide também pode ser utilizada para o contrast stretching.

Aplice a função sigmoide para contrast stretching especificada anteriormente (código exemplo) na imagem *vpfig.png*. Use a função `intlut` do MATLAB. Teste e mostre os resultados com diferentes valores para a variável 'slope', que determina a inclinação da sigmoide ('slope' maior: sigmoide mais 'ingreme').

Nome do .m: atv03\_01.m

#### 3.2) Equalização do histograma na unha

Fazer a equalização do histograma de uma imagem na unha. Pode usar a função `cumsum`. Mostrar a imagem *gDSC04422m16.png* processada com o seu programa e com o `histeq(I,256)` do MATLAB e o histograma de cada uma. Os passos para a equalização do histograma estão descritos abaixo.

Nome do .m: atv03\_02.m

1. Obter o histograma da imagem original.
2. Normalizar este histograma [dividir por  $M \times N$  (número de pixels da imagem)].
3. Obter a *cumulative distribution function* (cdf) [acumular o histograma do passo 2, isto é,  $p(i) = \sum_{j=0 \text{ até } i} p(j)$ ].
4. Transformar a cdf em *níveis de cinza arredondados* [multiplicar por 255 e transformar em `uint8`].
5. Aplicar o vetor do passo 4 como uma *função de transformação* sobre a imagem original, usando a função `intlut` do MATLAB.