

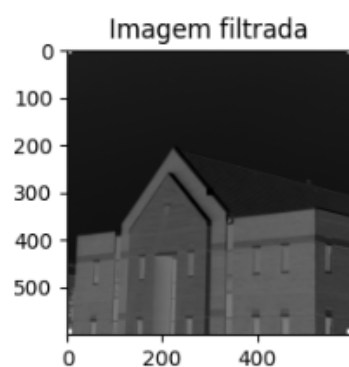
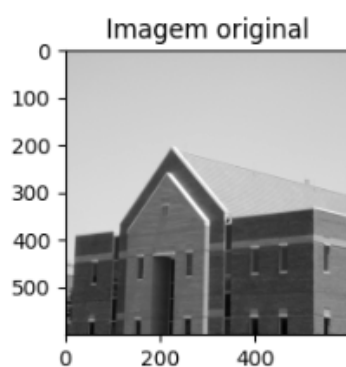
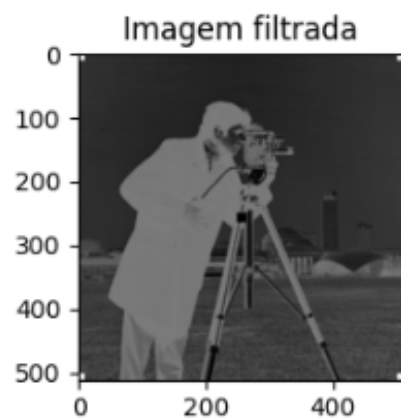
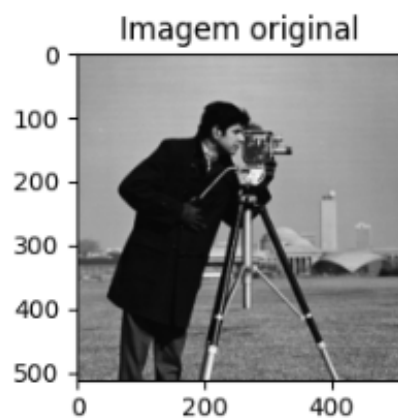
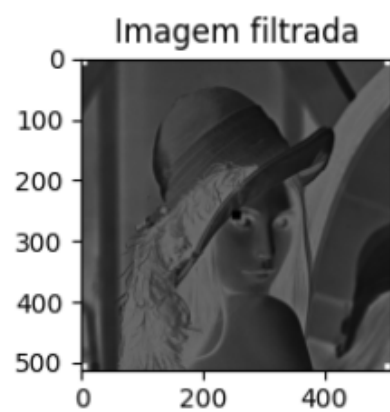
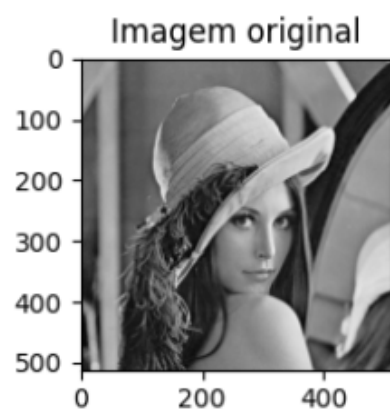
## Fundamentos

1) Operação ponto a ponto.

### Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosPontoaPonto.py>

### Resultados:



Operações ponto a ponto em Processamento Digital de Imagens (PDI) são aquelas em que cada pixel na imagem de entrada é processado independentemente dos outros pixels. Isso significa que o resultado da operação em um determinado pixel não depende dos valores dos pixels vizinhos. As operações ponto a ponto são fundamentais no PDI e são frequentemente utilizadas para realizar manipulações simples nos valores dos pixels de uma imagem.

São eficientes e, muitas vezes, são implementadas usando operações vetorizadas em bibliotecas como NumPy para melhorar o desempenho. Elas são a base para muitos algoritmos de processamento de imagens mais complexos.

## 2) Operação por vizinhança.

### **Código:**

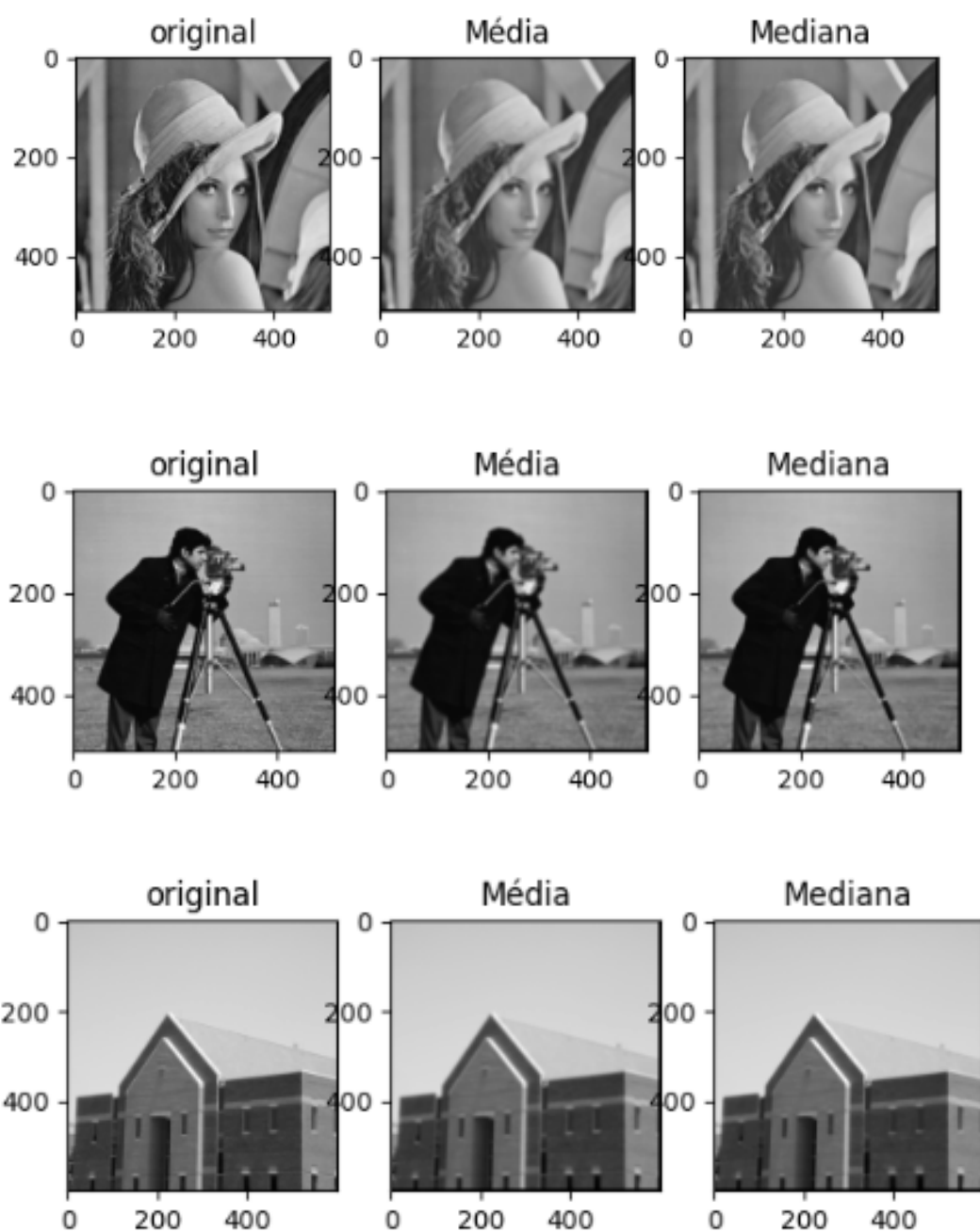
<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosOpPorVizinhanca.py>

Nas operações por vizinhança o valor de um pixel em uma imagem pode ser alterado com base nos valores dos pixels vizinhos. A vizinhança de um pixel refere-se aos pixels que o rodeiam em uma determinada região. Existem várias operações por vizinhança comuns, sendo algumas delas:

- Filtragem Média:
- Filtragem de Ponto Médio:
- Detecção de Bordas:
- Operações de Máscara:
- Erosão e Dilatação:
- Segmentação:

A vizinhança pode ser definida de várias maneiras, como 4-vizinhança (considerando pixels acima, abaixo, à esquerda e à direita) ou 8-vizinhança (considerando também os pixels nas diagonais).

## Resultado:



### 3) Zoom in

#### Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosZoomIn.py>

O *Zoom In* em uma imagem digital é um processo que amplia uma parte específica da imagem, tornando-a mais visível e ocupando uma porção maior da tela. Existem várias maneiras de implementar o *zoom in* em imagens digitais sendo a escolha do método de zoom dependente das necessidades específicas da aplicação e da qualidade desejada na imagem ampliada.

#### Resultado:



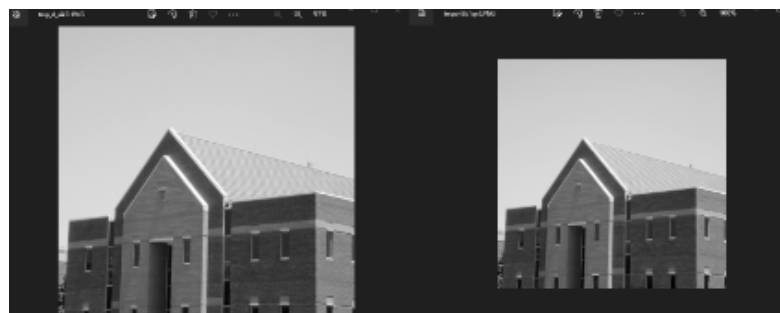
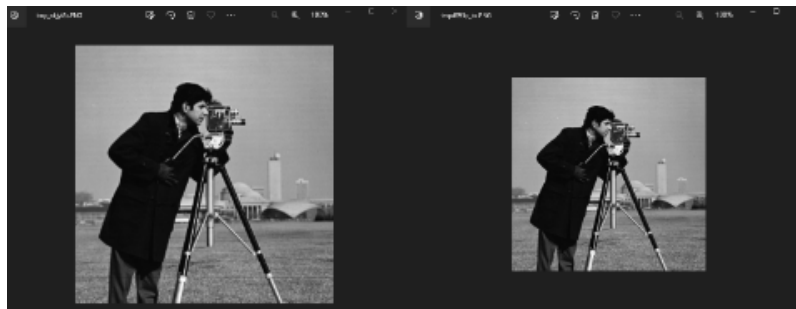
#### 4) Zoom out

##### Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosZoomOut.py>

Sendo o processo inverso do *Zoom In*. Ele envolve a redução do tamanho da imagem, mostrando a imagem original em uma porção menor da tela. Assim como no caso do zoom in, há várias abordagens para implementar o zoom out e o método escolhido também depende da aplicação.

##### Resultado:



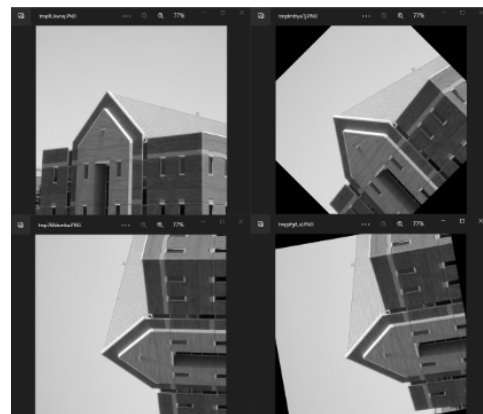
## 5) Rotação

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosRotacao.py>

A rotação de uma imagem envolve a transformação dos pixels da imagem para mudar sua orientação angular sendo a escolha do método dependente dos requisitos específicos da aplicação. Rotações pequenas podem ser realizadas eficientemente por métodos simples, enquanto rotações maiores ou mais complexas podem exigir técnicas mais avançadas.

Resultado:



## 6) Translação

Código:

<https://github.com/LuizHenriqueCervantes/pdi/blob/d2ab1fc56fd812a5a6123477e11036c9534db521/Fundamentos/fundamentosTranslacao.py>

Envolve o deslocamento de todos os pixels de uma imagem em direções específicas ao longo dos eixos x e y. A ideia básica é mover a imagem de uma certa quantidade de pixels em uma direção específica, criando a ilusão de que a imagem foi movida.

Resultado:

