Matrizes_Metodo_de_Gauss

```
In [1]: ## Matrizes_Metodo_de_Gauss
        '''Determina as solucoes de matrizes baseando-se no metodo de Gauss-Jordan (Eliminacao
        '''

        import numpy as np
        import copy

        def gauss_Elimin():
            print("Esta funcao usa dois argumentos (A,b) , Ax = b\n O array de saida sao as sol
            print(".........................................................")
            y = eval(input("Introduza os coef. de A, ex: [4,-2,1], [-2,4,-2], [1,-2,4] \n"))
            print(".........................................................")
            t = eval(input("Introduza os coef. de b, ex: [11,-16,17] \n"))
            print(".........................................................\n")
            # Matrizes .......> Ax=b; A==a
            a=np.array(y)
            b = np.array(t)

            n = len(b)
            # Fase de eliminacao
            for k in range(0,n-1):
                for i in range(k+2,n):
                    if a[i,k]!= 0.0:
                        lam = a[i,k]/a[k,k]
                        a[i,k+1] = a[i,k+1] - lam*a[k,k+1]
                        b[i] = b[i] - lam*b[k]
            # Back substitution
            print("........... As solucoes sao:\n")
            for k in range(n-1,-1,-1):
                b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]
            return b

In [2]: # Chamar a funcao para imprimir as solucoes!
        gauss_Elimin()

Esta funcao usa dois argumentos (A,b) , Ax = b
 O array de saida sao as solucoes das incognitas do seu sistema de equacoes
```

```
...
Introduza os coef. de A, ex: [4,-2,1], [-2,4,-2], [1,-2,4]
[4,-2,1], [-2,4,-2], [1,-2,4]
...
Introduza os coef. de b, ex: [11,-16,17]
[11,-16,17]
...

... As solucoes sao:



Out[2]: array([ 1, -2,  3])
```

Reparem que tambem podemos usar a algebra linear de numpy

```python
In [3]: from numpy import linalg as lin
        def Gauss():
            print("Esta funcao usa dois argumentos (A,b) , Ax = b\n O array de saida sao as sol
            print(".....................................................")
            y = eval(input("Introduza os coefs. de A, ex: [4,-2,1], [-2,4,-2], [1,-2,4] \n"))
            t = eval(input("Introduza os coefs. de b, ex: [11,-16,17] \n"))
            a=np.array(y)
            b=np.array(t)
            solucao = lin.solve(a,b)
            return solucao
```

```
In [4]: Gauss()

Esta funcao usa dois argumentos (A,b) , Ax = b
 O array de saida sao as solucoes das incognitas do seu sistema de equacoes

...
Introduza os coefs. de A, ex: [4,-2,1], [-2,4,-2], [1,-2,4]
[4,-2,1], [-2,4,-2], [1,-2,4]
Introduza os coefs. de b, ex: [11,-16,17]
[11,-16,17]


Out[4]: array([ 1., -2.,  3.])
```