

```
In [1]: import math as m
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import IPython.display as ipd
import sympy as sy
import sympy.physics.quantum.constants as const

x, y, z = sy.symbols('x y z')
sy.init_printing()
radianos = m.radians
```

## LISTA 4 - QUESTÃO 1

```
In [2]: ipd.Image(filename='L4Q1.png')
```

Out[2]:

**1ª Questão)** A pessoa, na Figura 1, possui 195 libras e sobe a escada, mas de repente para na posição mostrada quando sente que a escada está prestes a escorregar. Suponha que a parede em B seja completamente lisa, o centro de gravidade para o homem esteja indicado por G e desconsidere o peso da escada, qual deve ser inclinação da escada se o coeficiente de atrito estático entre a almofada de atrito A e o chão é  $\mu_s = 0,35$ ?

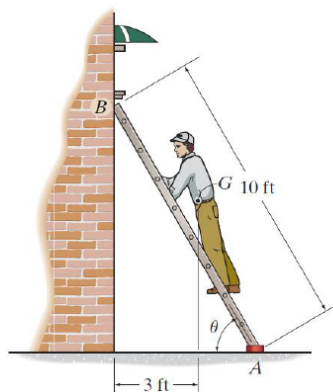


Figura 1.

### Dados necessários:

```
In [3]: massa_escada = 195 # [lbs]
coeficiente_atrito = 0.35 # [-]
g = sy.physics.units.g # [m/s^2]
g.convert_to(sy.physics.units.ft/sy.physics.units.s**2)
Lx = 3 # [ft]
Ly = 10 # [ft]
```

Elementos a serem calculados:

```
In [4]: normal_B, normal_A, Fat_A, theta = sy.symbols('normal_B normal_A Fat_A theta')
x = 10 * sy.cos(theta) - 3
y = 10 * sy.sin(theta)
```

### Calculando:

somatório das forças em x:

```
In [5]: Fat_A = coeficiente_atrito * normal_A
somatorio_fx = sy.Eq(normal_B - Fat_A, 0)
somatorio_fx
```

Out[5]:  $-0.35normal_A + normal_B = 0$

Somatório das forças em Y:

```
In [6]: somatorio_fy = sy.Eq(- massa_escada * g + normal_A, 0)
somatorio_fy
```

Out[6]:  $-195g + normal_A = 0$

relação entre as forças em x e y:

```
In [7]: solucao = sy.solve([somatorio_fx, somatorio_fy], [normal_B, normal_A, Fat_A])
solucao
```

Out[7]:  $\{normal_A : 195.0g, normal_B : 68.25g\}$

Somatório dos momentos em relação ao ponto A:

```
In [8]: somatorio_Ma = sy.Eq(- normal_A * x + normal_B * y, 0)
somatorio_Ma
```

Out[8]:  $-normal_A (10 \cos(\theta) - 3) + 10normal_B \sin(\theta) = 0$

Combinação das equações:

valores das normais (solucao) na eq dos momentos (somatorio\_Ma):

```
In [9]: eq_final = sy.simplify(somatorio_Ma.subs(solucao))
eq_final
```

Out[9]:  $(682.5 \sin(\theta) - 1950.0 \cos(\theta) + 585.0)g = 0$

a equação na linha anterior descreve a relação entre as medidas de forças em x, y e os momentos em relação ao ponto A para cada ângulo, o que nos dá a solução final em radianos:

```
In [10]: sy.solve(eq_final, theta, dict=True)
```

Out[10]:  $\{\theta : -1.62038638219529\}, \{\theta : 0.947036743421837\}$

salvando a solução em uma lista:

```
In [11]: theta_valores = sy.solve(eq_final, theta)
```

Isolar as soluções da equação com a função solve e salvar no vetor\_solucao:

```
In [12]: theta_graus = []
for angle in theta_valores:
    theta_graus.append(m.degrees(angle))
data1 = pd.DataFrame({
```

```
'theta em rad': theta_valores,  
'theta em graus': theta_graus  
})  
data1
```

Out[12]:

	theta em rad	theta em graus
0	-1.62038638219529	-92.841301
1	0.947036743421837	54.261208

In [13]: `print('o ângulo correto é: {:.3f} graus, pois o valor é positivo para a função seno`  
o ângulo correto é: 54.261 graus, pois o valor é positivo para a função seno

In [13]: