

Conversão de Unidades - Exercícios

Gustavo J. V. Meira Filho

Table of contents

Exercício 1: pH de um tampão	3
Exercício 2: Constante de equilíbrio pela energia livre	4
Exercício 3: Verificando a conversão da reação	5
Exercício 4: Faixa segura de temperatura de operação	6
Exercício 5: Conversão de uma lista de temperaturas	7
Exercício 6: Verificação de pressão com while loop	8
Exercício 7: Tabela de massas molares	9
Exercício 8: Conversor simples de unidades com dicionário	10
Exercício 9: Função para calcular a taxa de reação (lei de velocidade)	11
Exercício 10: Cálculo da constante de velocidade pela Equação de Arrhenius	13
Exercício 11: Conversão automática de temperatura	14
Exercício 12: Cálculo do Coeficiente Global de Transferência de Calor (U)	15

Exercício 1: pH de um tampão

A equação de Henderson–Hasselbalch relaciona o pH de uma solução tampão com o pKa do ácido e a razão entre base conjugada e ácido:

$$pH = pKa + \log_{10} \left(\frac{[base]}{[acido]} \right)$$

Sabendo que:

- pKa = 4.76
- [base] = 0.1 mol/L
- [ácido] = 0.01 mol/L

Calcule o pH do tampão.

Opcional: não use nenhuma operação de multiplicação ou divisão!

```
import math

pKa = 4.76
base = 0.1
acido = 0.01

pH = pKa + math.log10(base / acido)

print(f"O pH do tampão é {pH:.2f}")
```

O pH do tampão é 5.76

```
pH = pKa + math.log10(base) - math.log10(acido)

print(f"O pH do tampão é {pH:.2f}")
```

O pH do tampão é 5.76

Exercício 2: Constante de equilíbrio pela energia livre

A relação entre a variação da energia livre padrão (ΔG°) e a constante de equilíbrio (K) é dada por:

$$K = e^{-\frac{\Delta G^\circ}{RT}}$$

Sabendo que:

- $\Delta G^\circ = -5000 \text{ J/mol}$
- $R = 8.314 \text{ J/(mol} \cdot \text{K)}$
- $T = 298 \text{ K}$

Calcule a constante de equilíbrio K .

```
import math

# Dados
dG = -5000 # J/mol
R = 8.314 # J/(mol·K)
T = 298 # K

# Equação de equilíbrio
K = math.exp(-dG / (R * T))

print(f"A constante de equilíbrio é K = {K:.3f}")
```

A constante de equilíbrio é $K = 7.524$

Exercício 3: Verificando a conversão da reação

A **conversão** X em uma reação é dada por:

$$X = \frac{C_{A0} - C_A}{C_{A0}}$$

onde:

- C_{A0} é a concentração inicial do reagente *mol/L*
- C_A é a concentração final *mol/L*

O processo é considerado:

- **Excelente** se $X \geq 0.9$
- **Aceitável** se $0.7 \leq X < 0.9$
- **Ruim** se $X < 0.7$

```
Ca0 = 2.0 # mol/L
Ca = 0.3 # mol/L

X = (Ca0 - Ca) / Ca0

if X >= 0.9:
    print(f"Conversão = {X:.1%} → Reação excelente!")
elif X >= 0.7:
    print(f"Conversão = {X:.1%} → Reação aceitável.")
else:
    print(f"Conversão = {X:.1%} → Reação ruim.")
```

Conversão = 85.0% → Reação aceitável.

Exercício 4: Faixa segura de temperatura de operação

Um reator deve operar entre **350 K** e **600 K**. Crie um programa que leia a temperatura T e indique:

- “Temperatura segura” se estiver dentro da faixa;
- “Temperatura muito baixa” se menor que 350 K;
- “Temperatura muito alta” se maior que 600 K.

```
T = 420 # K

if T < 350:
    print("Temperatura muito baixa!")
elif T > 600:
    print("Temperatura muito alta!")
else:
    print("Temperatura segura ")
```

Temperatura segura

Exercício 5: Conversão de uma lista de temperaturas

Você tem uma lista de temperaturas em °C e quer convertê-las para K usando a relação:

$$T_K = T_C + 273.15$$

Escreva um programa que percorra a lista e imprima todas as temperaturas em Kelvin.

```
temps_c = [25, 50, 75, 100]

for T in temps_c:
    T_k = T + 273.15
    print(f"{T} °C = {T_k:.2f} K")
```

```
25 °C = 298.15 K
50 °C = 323.15 K
75 °C = 348.15 K
100 °C = 373.15 K
```

Exercício 6: Verificação de pressão com while loop

Um sistema começa a operar a **100 kPa**, e a pressão aumenta **20 kPa por ciclo**. Escreva um programa que use um **while loop** para imprimir a pressão a cada ciclo até atingir **200 kPa**.

```
P = 100 # kPa
while P <= 200:
    print(f"Pressão atual: {P} kPa")
    P += 20 # aumenta 20 kPa a cada iteração
```

```
Pressão atual: 100 kPa
Pressão atual: 120 kPa
Pressão atual: 140 kPa
Pressão atual: 160 kPa
Pressão atual: 180 kPa
Pressão atual: 200 kPa
```


Exercício 7: Tabela de massas molares

Crie um **dicionário** com as massas molares g/mol de algumas substâncias e use-o para calcular a **massa de uma amostra**.

Sabendo que:

$$m = n \times M$$

onde

- \$ n \$ = número de mols
- \$ M \$ = massa molar g/mol

```
molar_mass = {  
    "H2": 2.016,  
    "O2": 32.00,  
    "CO2": 44.01,  
    "CH4": 16.04  
}  
  
n = 0.5 # mols de CO2  
m = n * molar_mass["CO2"]  
  
print(f"Massa de {n} mol de CO2 = {m:.2f} g")
```

Massa de 0.5 mol de CO2 = 22.00 g

Exercício 8: Conversor simples de unidades com dicionário

Monte um dicionário com **fatores de conversão de pressão** e use-o para converter **1 atm** para outras unidades.

$$1 \text{ atm} = 101325 \text{ Pa} = 1.01325 \text{ bar} = 760 \text{ mmHg}$$

```
pressao = {  
    "Pa": 101325,  
    "bar": 1.01325,  
    "mmHg": 760  
}  
  
valor_atm = 1 # atm  
for unidade, fator in pressao.items():  
    print(f"{valor_atm} atm = {valor_atm * fator} {unidade}")
```

```
1 atm = 101325 Pa  
1 atm = 1.01325 bar  
1 atm = 760 mmHg
```

Exercício 9: Função para calcular a taxa de reação (lei de velocidade)

A lei de velocidade para uma reação genérica



pode ser escrita como:

$$r = k \cdot [A]^n$$

onde:

- r é a taxa de reação $\text{mol} \cdot \text{L}^{-1} \cdot \text{s}^{-1}$,
- k é a constante de velocidade,
- $[A]$ é a concentração do reagente,
- n é a ordem da reação.

“E se dobrarmos $[A]$? A taxa dobra também? Ou quadruplica?”

```
def taxa_reacao(k, A, n):  
    """  
    Calcula a taxa de reação pela lei de velocidade  $r = k \cdot [A]^n$ .  
    Retorna o valor de  $r$ .  
    """  
    r = k * (A ** n)  
    return r  
  
# Exemplo de uso:  
r1 = taxa_reacao(0.02, 0.5, 1)    # 1ª ordem  
r2 = taxa_reacao(0.005, 0.5, 2)  # 2ª ordem  
  
print(f"Reação de 1ª ordem: r = {r1:.4f} mol/L.s")  
print(f"Reação de 2ª ordem: r = {r2:.4f} mol/L.s")
```

Reação de 1ª ordem: $r = 0.0100 \text{ mol/L.s}$

Reação de 2ª ordem: $r = 0.0013 \text{ mol/L.s}$

Agora dobrando a concentração de A:

```
# Dobrando a quantidade de [A] <3  
r1_duplicado = taxa_reacao(0.02, 0.5*2, 1)    # 1ª ordem  
r2_duplicado = taxa_reacao(0.005, 0.5*2, 2)  # 2ª ordem  
  
print(f"Reação de 1ª ordem: r = {r1_duplicado:.4f} mol/L.s")  
print(f"Reação de 2ª ordem: r = {r2_duplicado:.4f} mol/L.s")
```

Reação de 1ª ordem: $r = 0.0200 \text{ mol/L}\cdot\text{s}$
Reação de 2ª ordem: $r = 0.0050 \text{ mol/L}\cdot\text{s}$

Avaliando as proporções...

```
print(f"A taxa para r1 fica {(r1_duplicado / r1)} vezes maior")  
print(f"A taxa para r2 fica {(r2_duplicado / r2)} vezes maior")
```

A taxa para r1 fica 2.0 vezes maior
A taxa para r2 fica 4.0 vezes maior

Aqui a gente nota a diferença entre primeira e segunda ordem de uma reação!

Exercício 10: Cálculo da constante de velocidade pela Equação de Arrhenius

A Equação de Arrhenius relaciona a constante de velocidade (k) com a temperatura (T):

$$k = A \cdot e^{-\frac{E_a}{R \cdot T}}$$

onde:

- A = fator pré-exponencial s^{-1}
- E_a = energia de ativação J/mol
- R = constante dos gases $8.314 J/mol \cdot K$
- T = temperatura K

“E se aumentarmos a temperatura para 400 K? O que acontece com k ? Por que isso acelera a reação?”

```
import math

def arrhenius(A, Ea, T):
    """
    Calcula a constante de velocidade (k) pela equação de Arrhenius.
    Retorna k em s^-1.
    """
    R = 8.314 # J/mol·K
    k = A * math.exp(-Ea / (R * T))
    return k

# Exemplo de uso:
A = 1.2e7 # s^-1
Ea = 75000 # J/mol
T = 350 # K

k = arrhenius(A, Ea, T)
print(f"Constante de velocidade k = {k:.4e} s^-1")
```

Constante de velocidade $k = 7.6849e-05 \text{ s}^{-1}$

```
k = arrhenius(A, Ea, T=400)
print(f"Constante de velocidade k = {k:.4e} s^-1")
```

Constante de velocidade $k = 1.9268e-03 \text{ s}^{-1}$

Ou seja, um aumento de 50 K faz k crescer cerca de 25×, o que ilustra lindamente o efeito da temperatura sobre a velocidade das reações

Exercício 11: Conversão automática de temperatura

Crie uma **função** que receba o valor e a unidade de temperatura ("C", "K", "F") e retorne um **dicionário** com as três equivalentes.

```
def converte_temp(T, unidade):
    if unidade == "C":
        C = T
        K = T + 273.15
        F = T * 9/5 + 32
    elif unidade == "K":
        C = T - 273.15
        K = T
        F = C * 9/5 + 32
    elif unidade == "F":
        C = (T - 32) * 5/9
        K = C + 273.15
        F = T
    else:
        return "Unidade inválida!"
    return {"C": round(C,2), "K": round(K,2), "F": round(F,2)}

print(converte_temp(300, "K"))
```

```
{'C': 26.85, 'K': 300, 'F': 80.33}
```

Exercício 12: Cálculo do Coeficiente Global de Transferência de Calor (U)

(com conversão automática de unidades)

Em engenharia térmica, uma forma empírica americana de estimar o **coeficiente global de transferência de calor** (U) é:

$$U = \frac{C \cdot Q^{0.8}}{A^{0.2}}$$

onde:

- U = coeficiente global ($\text{BTU/h} \cdot \text{ft}^2 \cdot ^\circ\text{F}$)
- Q = taxa de calor transferido (BTU/h)
- A = área de troca térmica (ft^2)
- C = constante empírica do sistema americano ($C = 0.023$ para água-água**)

Tarefa:

1. O usuário fornece **Q em kW**, **A em m²**, e quer o resultado de **U em W/m² · K**.
2. O programa deve:
 - Converter **Q e A** para unidades do **sistema americano**,
 - Calcular **U (em BTU/h · ft² · °F)**,
 - Converter o resultado final de volta para **W/m² · K**.

Variações possíveis para deixar mais rica a discussão:

1. **Adicionar parâmetro de fluido:** Diferentes valores de (C):

- Água-água $\rightarrow 0.023$
- Óleo-água $\rightarrow 0.017$
- Ar-água $\rightarrow 0.012$

2. **Usar entrada de dados real:**

```
tipo = input("Fluido (agua, oleo, ar): ").lower()
if tipo == "agua":
    C = 0.023
elif tipo == "oleo":
    C = 0.017
else:
    C = 0.012
```

3. Caso queiram adiantar as próximas aulas: **Fazer gráfico U vs Q** para mostrar comportamento exponencial.

```

# Conversões base
KW_TO_BTU_H = 3412.14      # 1 kW = 3412.14 BTU/h
M2_TO_FT2 = 10.7639       # 1 m² = 10.7639 ft²
BTU_HFT2F_TO_W_M2K = 5.678 # 1 BTU/h·ft²·°F = 5.678 W/m²·K

def calc_U(Q_kW, A_m2, C=0.023):
    """
    Calcula o coeficiente global U usando fórmula empírica americana.
    Converte internamente as unidades e retorna em W/m²·K.
    """
    # 1. Converter para sistema americano
    Q_btu_h = Q_kW * KW_TO_BTU_H
    A_ft2 = A_m2 * M2_TO_FT2

    # 2. Calcular U em BTU/h·ft²·°F
    U_btu = (C * (Q_btu_h ** 0.8)) / (A_ft2 ** 0.2)

    # 3. Converter de volta para W/m²·K
    U_w = U_btu * BTU_HFT2F_TO_W_M2K
    return U_w

# Exemplo de uso:
U = calc_U(Q_kW=150, A_m2=50)
print(f"Coeficiente global U = {U:.2f} W/m²·K")

```

Coeficiente global U = 1370.98 W/m²·K

Há um motivo físico e numérico de por que essas conversões são necessárias.

Essa equação empírica não funcionaria diretamente em unidades SI (comuns)! Ela **só é válida no sistema americano**, porque o valor da constante (C) foi ajustado empiricamente **para aquelas unidades**.

O que você faz	O que acontece
Usa unidades americanas + C original	Funciona, valores coerentes
Usa unidades SI com o mesmo C	Dá resultado fisicamente incorreto
Usa SI e recalibra C	Funciona, mas você precisa medir novos dados

Esse foi um problema no PIBIC do Gugu Meira... então fica aqui minha contribuição de sabedoria rs.