

Por favor lea cuidadosamente las siguientes instrucciones:

- Este ejercicio en clase es **individual**.
- Debe entregar esta hoja al final de la sesión.
- En preguntas *abiertas*, debe escribir su respuesta de manera clara y legible, aplicando la sintaxis y los conceptos de Python estudiados en el curso. Se provee una cuadrícula para facilitar la indentación de su código; úsela de manera que quede claro cuando una instrucción está indentada. **Las respuestas ilegibles o inadecuadamente indentadas, en un examen real, pueden ser calificadas como incorrectas.**
- Si tiene dudas, puede ir al puesto del profesor a solicitar su ayuda.

Compromiso de integridad:

Al proporcionar sus datos a continuación, usted confirma su compromiso de completar este ejercicio de acuerdo con las normas de integridad académica de la Universidad de los Andes, incluyendo acatar el Reglamento General de Estudiantes de Pregrado y todas las disposiciones relacionadas con evitar el *Fraude académico*¹, establecidas en el Reglamento Disciplinario para Estudiantes de Pregrado.

SOLUCIÓN

¹ En el artículo 4 del Reglamento Disciplinario para Estudiantes de Pregrado, se define el *Fraude académico* como “el comportamiento del estudiante que infringe las reglas de la Universidad o las reglas establecidas por el evaluador, en desarrollo de una actividad académica”.

CupiQuímica

Los profesionales de una compañía química utilizan diversas fórmulas en sus procesos. Este gremio lo ha contratado para desarrollar un programa en Python, denominado **CupiQuímica**, el cual debe calcular las siguientes fórmulas:

1. Fórmula de **cantidad de sustancia en moles**:

$$n = \frac{m}{M}$$

Donde:

n = Moles

m = Masa de la sustancia (g)

M = Masa molar (g/mol)

2. Relación de Avogadro:

$$N = n \cdot N_A$$

Donde:

- **N** = Número de partículas (átomos, moléculas o iones).
- **n** = Cantidad de sustancia en moles.
- **N_A** = Constante de Avogadro (6.022×10²³ partículas/mol).

3. Fórmula de la **molaridad**:

$$M = \frac{n}{V}$$

Donde:

M = Molaridad (mol/L).

n = Cantidad de sustancia en moles.

V = Volumen de solución en litros.

4. Ecuación de los gases ideales, aquí para obtener las moles de un gas:

$$n = \frac{P \cdot V}{R \cdot T}$$

Donde:

n = Moles del gas.

P = Presión (atm).

V = Volumen (L).

R = Constante de los gases ideales (0.0821 L·atm·mol⁻¹·K⁻¹).

T = Temperatura (K).

Pregunta 1: Implemente una función en Python que permita calcular la cantidad de sustancia en moles. Debe seguir las buenas prácticas del curso (BP-1 a BP-14). Note que para esta función ya se provee la signatura, docstring y 4 doctests (falta la implementación). Debe redondear el resultado final a 3 cifras decimales.

```
def calcular_moles_sustancia(masa: float, masa_molar: float) -> float:
    """
    Calcula la cantidad de moles.

    Args:
        masa (float): Masa de la sustancia en gramos.
        masa_molar (float): Masa molar de la sustancia en g/mol.

    Returns:
        float: Cantidad de moles de la sustancia, redondeada a tres decimales.

    >>> calcular_moles_sustancia(18, 18) # Agua: 18 g con M = 18 g/mol
    1.0
    >>> calcular_moles_sustancia(44, 44) # CO2: 44 g con M = 44 g/mol
    1.0
    >>> calcular_moles_sustancia(10, 58.44) # 10 g de NaCl
    0.171
    >>> calcular_moles_sustancia(98, 49) # 98 g de H2SO4 (M = 49 g/mol aprox.)
    2.0
    """
    return round(masa / masa_molar, 3)
```

Pregunta 2: Implemente una función en Python que permita calcular la relación de Avogadro. Debe seguir las buenas prácticas del curso (BP-1 a BP-14). Note que para esta función ya se provee 4 doctests (falta la signatura, docstring e implementación). Note que el número de Avogadro, se puede representar en Python así: 6.022e23. Debe redondear el resultado final a 3 cifras decimales.

```
def calcular_relacion_avogadro(masa_sustancia: float, masa_molar: float) -> float:
    """
    Calcula el número de partículas (N) usando la relación de Avogadro,
    a partir de la masa de la sustancia y su masa molar:

    
$$N = (m / M) * N_A$$


    Args:
        masa_sustancia (float): Masa de la sustancia en gramos.
        masa_molar (float): Masa molar de la sustancia en g/mol.

    Returns:
        float: Número de partículas, redondeado a tres decimales.

    >>> calcular_relacion_avogadro(18, 18) # 1 mol -> N_A
    6.022e+23
    >>> calcular_relacion_avogadro(9, 18) # 0.5 mol -> N_A/2
    3.011e+23
    >>> calcular_relacion_avogadro(36, 18) # 2 mol -> 2*N_A
    1.2044e+24
    >>> calcular_relacion_avogadro(0.018, 18) # 1e-3 mol -> 1e-3*N_A
    6.022e+20
    """
    NUMERO_AVOGADRO = 6.022e23
    moles = calcular_moles_sustancia(masa_sustancia, masa_molar)
    return round(moles * NUMERO_AVOGADRO, 3)
```

Pregunta 3: Implemente una función en Python que permita calcular la molaridad. Debe seguir las buenas prácticas del curso (BP-1 a BP-14). Debe redondear el resultado final a 2 cifras decimales. Puede usar calculadora para obtener los resultados de los 4 doctests que debe proponer.

```
def calcular_molaridad(masa_sustancia: float, masa_molar: float, volumen: float) -> float:
    """
    Calcula la molaridad de una solución.

    Args:
        masa_sustancia (float): Masa de la sustancia en gramos.
        masa_molar (float): Masa molar de la sustancia en g/mol.
        volumen (float): Volumen de la solución en litros.

    Returns:
        float: Molaridad de la solución en mol/L, redondeada a tres decimales.

    >>> calcular_molaridad(18, 18, 1) # 18 g de agua (1 mol) en 1 L
    1.0
    >>> calcular_molaridad(36, 18, 1) # 36 g de agua (2 mol) en 1 L
    2.0
    >>> calcular_molaridad(9, 18, 2) # 9 g de agua (0.5 mol) en 2 L
    0.25
    >>> calcular_molaridad(0.1, 18, 0.005) # 0.1 g en 5 mL -> 1.111 mol/L
    1.2
    """
    moles = calcular_moles_sustancia(masa_sustancia, masa_molar)
    return round(moles / volumen, 2)
```

Pregunta 4: Implemente una función en Python que permita calcular los moles de un gas ideal usando la ecuación de los gases ideales. Debe seguir las buenas prácticas del curso (BP-1 a BP-14). Debe redondear el resultado final a 2 cifras decimales. Puede usar calculadora para obtener los resultados de los 4 doctests que debe proponer.

```
def calcular_moles_gas_ideal(presion: float, volumen: float, temperatura: float) -> float:
    """
    Calcula los moles de un gas ideal usando la ecuación:  $n = (P * V) / (R * T)$ .

    Args:
        presion (float): Presión del gas en atm.
        volumen (float): Volumen del gas en litros (L).
        temperatura (float): Temperatura absoluta en kelvin (K).

    Returns:
        float: Cantidad de moles (n), redondeada a tres decimales.

    >>> calcular_moles_gas_ideal(1, 24.63, 300) #  $P*V = R*T \rightarrow 1 \text{ mol}$ 
    1.0
    >>> calcular_moles_gas_ideal(2, 24.63, 300) # Duplicar P duplica n (V,T fijos)
    2.0
    >>> calcular_moles_gas_ideal(0.5, 10, 250) # Caso con n pequeño
    0.24
    >>> calcular_moles_gas_ideal(1, 1, 298) # 1 L a 298 K
    0.04
    """
    CONSTANTE_GASES_IDEALES = 0.0821
    return round((presion * volumen) / (CONSTANTE_GASES_IDEALES * temperatura), 2)
```