

Programación para Física y Astronomía

Departamento de Física.

Corodinadora: C Loyola

Profesoras/es C Loyola / C Femenías / Y Navarrete / C Ruiz / F Bugini

Primer Semestre 2025

Universidad Andrés Bello

Departamento de Física y Astronomía



Introducción y Repaso

Revisión de Conceptos Clave

Aplicación de Conceptos

Ejercicio a Evaluar // Tarea Semanal

Conclusiones

Introducción y Repaso

- **Semana 2, Sesión 1 (Sesión 3)** se centró en:
 - Sintaxis básica de Python (indentación, palabras reservadas, comentarios).
 - Reforzar tipos de datos y operaciones (prioridades, conversión, etc.).
 - Ejemplos interactivos en Google Colab.
 - Actividad grupal: *Mini-calculadora*.
- **Objetivo de hoy:** Aplicar estos conceptos en problemas un poco más elaborados.

Objetivos de la Sesión 4

- **Consolidar** el manejo de variables, operaciones y sintaxis en Python.
- **Ejercitar** la resolución de problemas más complejos y colaborativos.
- **Fomentar** el razonamiento algorítmico (secuencia, condición, repetición).
- **Preparar** el terreno para estructuras de control (**if**, **while**, etc.).

Revisión de Conceptos Clave

- Las variables se crean con una asignación: `x = 10`.
- **Operaciones:**
 - Suma, resta, multiplicación, división, división entera, exponente.
 - Uso de paréntesis para priorizar operaciones.
- Python es **dinámico** en tipos: `x = 3` (int), luego `x = 3.14` (float).
- **Comentarios** ayudan a documentar el código (`#`, `"""..."""`).

```
1 user_input = input("Dame un número: ") # str
2 num = float(user_input)                # convierto a float
3 print(f'Tu número + 10 es: {num + 10}')
```

- `input()` siempre regresa una cadena (`str`).
- Para obtener números, se hace `int()` o `float()`.
- **Sugerencia:** Manejar excepciones (ej.: `ValueError`) en casos avanzados.

- **Bloques** de código definidos por sangría (generalmente 4 espacios).
- Usar dos puntos (:) tras `if`, `for`, `while`, etc.
- Ejemplo simple:

```
if x > 0:  
    print("x es positivo")
```

- **Cuidado:** Mezclar tabulaciones y espacios puede causar errores.

Aplicación de Conceptos

Actividad Central: Problemas Paso a Paso

- Realizaremos 2-3 **problemas guiados** de dificultad gradual.
- Objetivo: Integrar variables, operaciones y lógica básica.
- Cada ejercicio se abordará primero en **colaboración** y luego se compartirá la solución.

Ejercicio 1: Ecuación de Movimiento en 1D

Enunciado

- Datos:
 - x_0 : posición inicial (m).
 - v_0 : velocidad inicial (m/s).
 - a : aceleración constante (m/s^2).
- Calcular la posición $x(t)$ en un tiempo t usando:

$$x(t) = x_0 + v_0 \cdot t + \frac{1}{2}at^2$$

Tips:

- Pedir los datos al usuario.
- Imprimir la respuesta con un mensaje que indique la unidad (por ej. metros).

Ejercicio 2: Promedio y Varianza (Intro)

Enunciado

- Pedir al usuario **3 valores** (pueden representar mediciones físicas).
- Calcular el **promedio** (\bar{x}) y la **varianza** muestral.
- Fórmula varianza (para 3 datos):

$$\sigma^2 = \frac{\sum_{i=1}^3 (x_i - \bar{x})^2}{3 - 1}$$

- Mostrar ambos resultados.

Discusión: Manejo de potencias y restas, y cuidado con divisiones.

Ejercicio 3: Factor de Conversión

Enunciado

- Crear un programa que solicite:
 - Un valor numérico (**valor**).
 - Una unidad origen (por ej. "cm", "m", "km").
 - Una unidad destino ("cm", "m", "km").
- Convertir **valor** de la unidad origen a la unidad destino.
- Imprimir el resultado final.

Idea: Reutilizar factores como: $1 \text{ m} = 100 \text{ cm}$, $1 \text{ km} = 1000 \text{ m}$, etc.

- Formar **grupos de 2-3 estudiantes**.
- Seleccionar 1-2 ejercicios propuestos (o intentar todos).
- Editar un **notebook compartido** en Google Colab.
- **Objetivo:** Discutir soluciones, anotar dudas y resolver en conjunto.

- ¿Cuál de los ejercicios fue el más complejo?
- ¿En qué parte surgieron errores recurrentes?
- ¿Cómo podría hacerse un **diseño modular** (dividir el problema en funciones)?

Comparte tus experiencias con la clase.

Solución Propuesta: Ecuación de Movimiento 1D

```
1 x0_str = input("x0 (m): ")
2 v0_str = input("v0 (m/s): ")
3 a_str  = input("a (m/s^2): ")
4 t_str  = input("t (s): ")
5
6 x0 = float(x0_str)
7 v0 = float(v0_str)
8 a  = float(a_str)
9 t  = float(t_str)
10
11 x_t = x0 + v0*t + 0.5*a*(t**2)
12
13 print(f'La posición en el tiempo t es: {x_t} m')
```

Solución Propuesta: Promedio y Varianza

```
1 vals = []
2 for i in range(1, 4):
3     val_str = input(f"Ingrese valor {i}: ")
4     val = float(val_str)
5     vals.append(val)
6
7 # Promedio
8 mean = sum(vals) / 3
9
10 # Varianza muestral (n=3)
11 var = 0
12 for v in vals:
13     var += (v - mean)**2
14 var = var / (3 - 1)
15
16 print(f'Promedio = {mean}')
17 print(f'Varianza = {var}')
```

Solución Propuesta: Conversión de Unidades

```
1 valor_str = input("Valor: ")
2 unidad_origen = input("Unidad origen (cm, m, km): ")
3 unidad_destino = input("Unidad destino (cm, m, km): ")
4
5 valor = float(valor_str)
6 # Convertir primero a metros
7 if unidad_origen == "cm":
8     valor_m = valor / 100.0
9 elif unidad_origen == "m":
10     valor_m = valor
11 elif unidad_origen == "km":
12     valor_m = valor * 1000
13 else:
14     valor_m = None
15 # Convertir metros a destino
16 if unidad_destino == "cm":
17     result = valor_m * 100.0
18 elif unidad_destino == "m":
19     result = valor_m
20 elif unidad_destino == "km":
21     result = valor_m / 1000
22 else:
23     result = None
24
25 print(f'Resultado: {result} {unidad_destino}')
```

- **For Loops** y **range**: usados en Ej. 2 para leer múltiples valores.
- **if/elif/else** (Ej. 3): Estructura para manejar casos y evitar errores de usuario.
- **Validaciones**: No cubrimos todas, pero en casos avanzados se revisan entradas inválidas.
- **Reutilización de código**: Cómo podríamos organizarlo en funciones (tema futuro).

Ejercicio a Evaluar // Tarea Semanal

Calculadora Físico-Química

- Pedir al usuario que seleccione:
 - (1) Calcular densidad ($\rho = \frac{m}{V}$)
 - (2) Calcular fuerza ($F = m \cdot a$)
 - (3) Calcular energía cinética ($E_c = \frac{1}{2}mv^2$)
- Con base en la elección, se piden los datos correspondientes y se muestra el resultado.
- Se puede repetir o terminar el programa tras cada operación.

- ¿Alguno de los ejercicios resultó especialmente difícil?
- ¿Cómo han manejado los **mensajes de error** (entradas inválidas)?
- **Sugerencia:** Documentar mejor tus funciones para cuando construyamos proyectos más grandes.

Recuerden

El grupo debe entregar el resultado en la plataforma CANVAS

Conclusiones

- Ampliamos ejercicios que involucran:
 - Operaciones matemáticas en escenarios reales (movimiento, varianza, conversiones).
 - Manejo de datos desde el usuario.
 - Estructuración básica (**if**, **for**) sin profundizar en la teoría de control (eso vendrá pronto).
- Vimos la importancia de la **colaboración** y la **discusión** para resolver problemas.

- Próximamente: **Unidad III** del Syllabus (Controladores **if**, **while**, **break**, **continue**, etc.).
- **Recomendación:**
 - Revisar cómo usar **while** para repeticiones indefinidas.
 - Entender la diferencia entre **while** y **for**.
- **Recordar:** Python 3.x es la versión recomendada. Fíjate en **/** vs **//**.

- [Python Docs - Control Flow](#) (para un vistazo previo).
- [LearnPython.org](#) (ejercicios básicos).
- [Real Python - Python Basics](#) (material de repaso).

- **Practicar todos los días:** sesiones cortas pero frecuentes.
- **Explorar ejemplos reales:** si te gusta la astronomía, intenta con datos de planetas o estrellas.
- **Comentar y Documentar** tu código: te ayudará a recordar qué hiciste la próxima vez.

¡Muchas gracias!

- Recuerda guardar tu trabajo en Google Drive.
- La próxima sesión profundizaremos en `if`, `while` y estructuras de control.
- ¡Sigán practicando con los ejercicios adicionales!