

Programación para Física y Astronomía

Departamento de Física.

Coordinadora: C Loyola

Profesores C Femenías / F Bugini / D Basantes

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 1

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.

- **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

Revisión de Conceptos Clave \ni Variables y Operaciones: Recordatorio

- 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.

Revisión de Conceptos Clave \ni Estructura Condicional: if, elif, else

```
1 edad = int(input("Ingresa tu edad: "))
2
3 if edad >= 18:
4     print("Eres mayor de edad")
5 elif edad >= 13:
6     print("Eres adolescente")
7 else:
8     print("Eres menor de edad")
```

- **if**: Evalúa una condición verdadera/falsa
- **elif**: Condición alternativa (opcional)
- **else**: Se ejecuta si ninguna condición anterior es verdadera (opcional)
- Operadores de comparación: ==, !=, <, >, <=, >=

```
1  # Repetir 5 veces
2  for i in range(5):
3      print(f"Iteración número: {i}")
4
5  # Rango con inicio y fin
6  for numero in range(1, 6): # 1, 2, 3, 4, 5
7      print(f"Número: {numero}")
8
9  # Rango con paso
10 for par in range(0, 11, 2): # 0, 2, 4, 6, 8, 10
11     print(f"Número par: {par}")
```

- `range(n)`: genera números de 0 a $n-1$
- `range(inicio, fin)`: de inicio a $fin-1$
- `range(inicio, fin, paso)`: con incremento específico

Revisión de Conceptos Clave \ni FOR: Trabajando con Listas y Acumuladores

```
1  # Crear una lista y recorrerla
2  numeros = [2, 5, 8, 3, 9]
3  for num in numeros:
4      print(f"El número es: {num}")
5
6  # Acumulador para suma
7  suma_total = 0
8  for i in range(1, 6): # suma de 1+2+3+4+5
9      suma_total += i   # equivale a: suma_total = suma_total + i
10
11  print(f"La suma total es: {suma_total}")
12
13  # Crear listas dinámicamente
14  cuadrados = []
15  for x in range(1, 6):
16      cuadrados.append(x**2) # [1, 4, 9, 16, 25]
```

Conceptos clave: Acumulación, listas dinámicas, append()

Aplicación de Conceptos

Aplicación de Conceptos \ni Actividad Central: Problemas Paso a Paso

- Realizaremos 5 **ejercicios progresivos** que integran:
 - Variables y operaciones matemáticas
 - Estructuras condicionales (**if**)
 - Bucles de repetición (**for**)
 - Aplicaciones en contexto físico
- Cada ejercicio se abordará primero en **colaboración** y luego se compartirá la solución.
- Objetivo: Consolidar el uso de estructuras de control en problemas reales.

Aplicación de Conceptos \ni Ejercicio 1:

Ecuación de Movimiento en 1D



Enunciado

- Dados los siguientes parámetros físicos:
 - x_0 : posición inicial (m)
 - v_0 : velocidad inicial (m/s)
 - a : aceleración constante (m/s²)
 - t : tiempo (s)
- Calcular la posición final usando la ecuación cinemática:

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

- Mostrar el resultado con unidades apropiadas.

Conceptos: Ecuaciones cinemáticas, variables de entrada, cálculos secuenciales.

Física relevante: Movimiento rectilíneo uniformemente acelerado (MRUA).

Aplicación de Conceptos \ni Ejercicio 2:

Promedio y Varianza de Mediciones



Enunciado

- Solicitar al usuario **3 mediciones** físicas (pueden ser temperaturas, distancias, etc.).
- Utilizar un bucle **for** para recopilar los datos.
- Calcular el **promedio** (\bar{x}) y la **varianza** muestral.
- Fórmula de varianza muestral:

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

- Mostrar ambos resultados con formato apropiado.

Conceptos: Bucles **for**, listas, acumuladores, estadística básica.

Física relevante: Análisis estadístico de mediciones experimentales.

Aplicación de Conceptos \ni Ejercicio 3:

Conversión de Unidades con Condicionales



Enunciado

- Crear un programa que solicite al usuario:
 - Un valor numérico
 - Una unidad origen: "cm", "m", "km"
 - Una unidad destino: "cm", "m", "km"
- Usar estructuras **if-elif-else** para determinar la conversión.
- Calcular y mostrar el resultado con las unidades correspondientes.
- Manejar casos de unidades inválidas con mensajes de error.

Conceptos: Condicionales múltiples, validación de entrada, factores de conversión.

Física relevante: Sistema métrico de unidades, conversiones de longitud.

Aplicación de Conceptos \ni Ejercicio 4:

Tabla de Multiplicar con Bucles



Enunciado

- Solicitar al usuario un número entero.
- Usar un bucle **for** para generar la tabla de multiplicar de ese número.
- Mostrar los resultados del 1 al 10 en formato: "**N x i = resultado**".
- Agregar una validación para verificar que el número ingresado sea positivo.

Conceptos: Bucles **for**, validación con **if**, formato de salida.

Física relevante: Relaciones proporcionales, escalado de magnitudes.

Aplicación de Conceptos \ni Ejercicio 5:

Clasificador de Temperaturas



Enunciado

- Solicitar al usuario una lista de temperaturas en grados Celsius.
- Usar un bucle **for** para procesar cada temperatura.
- Clasificar cada temperatura usando **if-elif-else**:
 - Menor a 0°C: "Congelación"
 - 0°C a 25°C: "Frío"
 - 25°C a 35°C: "Templado"
 - Mayor a 35°C: "Calor"
- Mostrar un resumen final con la cantidad de temperaturas en cada categoría.

Conceptos: Bucles, condicionales anidados, contadores, procesamiento de listas.

Física relevante: Estados de la materia, escalas de temperatura.

- Formar **grupos de 2-3 estudiantes**.
- Seleccionar 2-3 ejercicios de los 5 propuestos (según el tiempo disponible).
- Editar un **notebook compartido** en Google Colab.
- **Estrategia recomendada:**
 - Ejercicios 1-2: Fundamentales (variables, operaciones, bucles básicos)
 - Ejercicios 3-4: Intermedios (condicionales, validaciones)
 - Ejercicio 5: Avanzado (integración de conceptos)
- **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.

Aplicación de Conceptos \ni Solución 1 de Referencia:

Ecuación de Movimiento en 1D



```
1  # Solicitar datos al usuario con unidades claras
2  x0 = float(input("Posición inicial x0 (m): "))
3  v0 = float(input("Velocidad inicial v0 (m/s): "))
4  a = float(input("Aceleración a (m/s²): "))
5  t = float(input("Tiempo t (s): "))
6
7  # Aplicar la ecuación cinemática
8  x_final = x0 + v0 * t + 0.5 * a * (t**2)
9
10 # Mostrar resultado con formato claro
11 print(f"La posición final es: {x_final:.2f} m")
```

Discusión: Uso directo de `float()` para simplificar entrada, formato de decimales en salida.

Aplicación de Conceptos \ni Solución 2 de Referencia:

Promedio y Varianza de Mediciones



```
1  # Recopilar datos usando bucle for
2  mediciones = []
3  for i in range(1, 4):
4      valor = float(input(f"Ingrese medición {i}: "))
5      mediciones.append(valor)
6
7  # Calcular promedio
8  promedio = sum(mediciones) / len(mediciones)
9
10 # Calcular varianza muestral
11 suma_diferencias = 0
12 for valor in mediciones:
13     suma_diferencias += (valor - promedio)**2
14
15 varianza = suma_diferencias / (len(mediciones) - 1)
16
17 # Mostrar resultados
18 print(f"Promedio: {promedio:.3f}")
19 print(f"Varianza muestral: {varianza:.3f}")
```

Aplicación de Conceptos \ni Solución 3 de Referencia:

Conversión de Unidades con Condicionales



```
1  # Solicitar datos al usuario
2  valor = float(input("Ingrese el valor numérico: "))
3  unidad_origen = input("Unidad origen (cm, m, km): ").lower()
4  unidad_destino = input("Unidad destino (cm, m, km): ").lower()
5
6  # Convertir primero todo a metros (unidad base)
7  if unidad_origen == "cm":
8      valor_metros = valor / 100
9  elif unidad_origen == "m":
10     valor_metros = valor
11  elif unidad_origen == "km":
12     valor_metros = valor * 1000
13  else:
14     print("Unidad de origen no válida")
15     valor_metros = None
16
17  # Convertir de metros a unidad destino
18  if valor_metros is not None:
19     if unidad_destino == "cm":
```


Aplicación de Conceptos \ni Solución 4 de Referencia:

Tabla de Multiplicar con Bucles



```
1  # Solicitar número al usuario
2  numero = int(input("Ingrese un número para su tabla de
   ↪ multiplicar: "))
3
4  # Validar que sea positivo
5  if numero > 0:
6      print(f"Tabla de multiplicar del {numero}:")
7      print("-" * 25)
8
9      # Generar tabla usando bucle for
10     for i in range(1, 11):
11         resultado = numero * i
12         print(f"{numero} x {i} = {resultado}")
13 else:
14     print("Por favor, ingrese un número positivo.")
```

Discusión: Validación con if, bucle for con range, formato de salida organizado.

Aplicación de Conceptos \ni Solución 5 de Referencia:

Clasificador de Temperaturas



```
1  # Solicitar temperaturas
2  temperaturas = []
3  num_temp = int(input("¿Cuántas temperaturas desea clasificar?
   ↪  "))
4
5  for i in range(num_temp):
6      temp = float(input(f"Temperatura {i+1} (°C): "))
7      temperaturas.append(temp)
8
9  # Contadores para cada categoría
10 congelacion = frio = templado = calor = 0
11
12 # Clasificar cada temperatura
13 for temp in temperaturas:
14     if temp < 0:
15         print(f"{temp}°C: Congelación")
16         congelacion += 1
17     elif temp <= 25:
18         print(f"{temp}°C: Frío")
```

- **Estructuras de control integradas:**
 - `for` loops con `range()`: Ejercicios 2, 4 y 5
 - `if-elif-else`: Ejercicios 3 y 5
 - Validaciones: Ejercicios 4 y 5
- **Conceptos de programación aplicados:**
 - Acumuladores: varianza (Ej. 2), contadores (Ej. 5)
 - Listas dinámicas: `append()` en múltiples ejercicios
 - Formato de salida: `f-strings` con decimales
- **Buenas prácticas observadas:**
 - Comentarios explicativos en cada sección
 - Validación de entrada de usuario
 - Nombres de variables descriptivos

Ejercicio a Evaluar: Tarea Semanal 1

Ejercicio a Evaluar: Tarea Semanal 1 \ni Actividad Extra 1:

Calculadora Físico-Química Interactiva



Enunciado

- Crear un programa que presente un menú al usuario con las opciones:
 - (1) Calcular densidad: $\rho = \frac{m}{V}$ (kg/m³)
 - (2) Calcular fuerza: $F = m \cdot a$ (N)
 - (3) Calcular energía cinética: $E_c = \frac{1}{2}mv^2$ (J)
 - (4) Salir del programa
- Usar **if-elif-else** para procesar la selección del usuario.
- Solicitar los datos necesarios y mostrar el resultado con unidades.
- Permitir al usuario realizar múltiples cálculos hasta que elija salir.

Objetivo: Integrar menús, condicionales, bucles y cálculos físicos en una aplicación completa.

Ejercicio a Evaluar: Tarea Semanal 1 \ni Retroalimentación Colectiva

- ¿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.
 - Estructuras de control (**if**, **for**) aplicadas en contextos físicos.
- Vimos la importancia de la **colaboración** y la **discusión** para resolver problemas.
- Integramos conceptos de programación con aplicaciones físicas reales.

- Próximamente: **Unidad III** del Syllabus (Controladores **while**, **break**, **continue**, etc.).
- **Recomendación:**
 - Revisar cómo usar **while** para repeticiones indefinidas.
 - Entender la diferencia entre **while** y **for**.
 - Practicar validaciones de entrada con bucles.
- **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 **while** y estructuras de control avanzadas.
- ¡Sigán practicando con los ejercicios adicionales!