

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

Recordatorio: Estructuras de Control

Tarea Semanal

Conclusiones

Introducción y Repaso

- **Semana 3, Sesión 1 (Sesión 5)** se enfocó en:
 - Introducir las estructuras de control fundamentales en Python:
 - `if`, `elif`, `else` para condicionales.
 - `while` para bucles basados en condiciones lógicas.
 - Ejemplos sencillos de decisiones y repeticiones.
 - Manejo de la indentación y bloques de código.
- **Objetivo de hoy:** Realizar un **laboratorio práctico** con ejercicios enfocados en la aplicación de condicionales y bucles.

- **Aplicar** las estructuras de control en la resolución de problemas concretos.
- **Fomentar** el trabajo en parejas o grupos para compartir estrategias.
- **Profundizar** en el uso de `if-elif-else` y `while`.
- **Fortalecer** la comprensión de la lógica condicional y bucles a través de experimentación en Colab.

Recordatorio: Estructuras de Control

Condicionales en Python

```
1 if condicion1:  
2     # Bloque de código si condicion1 es True  
3 elif condicion2:  
4     # Bloque de código si condicion2 es True  
5 else:  
6     # Bloque de código si ninguna de las anteriores
```

- Cada condición se evalúa en orden.
- Solo se ejecuta el primer bloque que resulte **True**.
- **Uso común:** validaciones, menús, decisiones.

Bucle While en Python

```
1 while condicion:
2     # bloque que se repite
3     # mientras la condicion sea True
4
5 # Al salir, la condicion es False (o se rompió el bucle con
   ↪ break)
```

- Útil cuando no se sabe cuántas iteraciones exactas serán necesarias.
- **break**: fuerza la salida del bucle.
- **continue**: salta a la siguiente iteración.

Ejemplo Rápido: Menú Interactivo con While

```
1 op = ""
2 while op != "q":
3     print("Menú:")
4     print("(1) Saludar")
5     print("(2) Despedir")
6     print("(q) Salir")
7     op = input("Opción: ")
8
9     if op == "1":
10        print("Hola!")
11    elif op == "2":
12        print("Adiós!")
13    elif op == "q":
14        print("Saliendo...")
15    else:
16        print("Opción inválida")
```

Tarea Semanal

- Se plantearán **tres problemas** de complejidad progresiva.
- El grupo deberá resolver al menos 2 de estos problemas, para obtener la calificación máxima.
- Cada problema requiere el uso de condicionales y/o bucles **while**.
- Trabajaremos en **parejas** o **pequeños grupos**.
- Al final, el grupo deberá adjuntar sus resultados en CANVAS.

Problema 1: Adivina el Número

Enunciado

- El programa genera un número entero aleatorio entre 1 y 50.
- El usuario debe adivinar el número.
- Se le indica si su intento es **muy alto** o **muy bajo**, hasta acertar.
- Cuando acierta, se imprime cuántos intentos utilizó.

Indicaciones:

- Usar `while` para repetir hasta acertar.
- módulo `random`: `import random;`
`random.randint(1,50).`

Problema 2: Calculadora de Calificaciones

Enunciado

- Pedir repetidamente **notas** de estudiantes en el rango [1.0 - 7.0].
- Acumular la suma y el conteo de notas.
- Si se ingresa **-1**, termina la captura de datos.
- Imprimir el **promedio final** de las notas ingresadas (o mensaje si no se ingresó ninguna).

Puntos Clave:

- **while** para la repetición.
- Validar que la nota esté en [1.0, 7.0] o sea **-1** para salir.

Problema 3: Simulación de Movimiento Discreto

Enunciado

- Imaginemos un objeto en la posición $x=0$ de una línea de números.
- Cada paso de tiempo, el usuario ingresa un comando:
 - R (mover +1)
 - L (mover -1)
 - S (mantenerse)
 - Q (terminar)
- El programa muestra la posición actualizada tras cada comando.
- Al final, imprime cuántos pasos se realizaron.

Sugerencia:

- Usar un **while** infinito y **break** al recibir "Q".
- Contar la cantidad de pasos totales (excluyendo el comando "Q").

- Dividir la clase en **parejas** o **tríos**, según tamaño del grupo total.
- Cada equipo elige al menos dos problemas a resolver:
 - Problema 1: Adivina el Número.
 - Problema 2: Calculadora de Calificaciones.
 - Problema 3: Simulación de Movimiento Discreto.
- **Objetivo:** Implementar y probar en Google Colab, comentando el código. Enviar resultado a CANVAS.

Lineamientos para Resolver los Problemas

- Crear un **notebook** específico para esta sesión.
- Explicar brevemente la lógica al inicio de cada solución.
- Probar **múltiples casos** (valores límites, valores inválidos, etc.).
- Anotar cualquier dificultad o bug encontrado, junto con la forma de resolverlo.

Conclusiones

- **Condicionales** son la base de la *toma de decisiones* en un programa.
- **Bucle While** posibilita la *repetición basada en una condición*, útil para menús interactivos o lectura indefinida.
- Aprendimos la importancia de **validar** datos y usar **break/continue** en casos convenientes.

- Practica **problemas pequeños** con **if** y **while**, como minijuegos o menús.
- Lee la documentación de **Python** sobre control de flujo (if, while, for).
- Experimenta con valores **fuera de rango** para entender mejor la lógica y los errores potenciales.

- Siguiente tema: **Bucle For** y manejo de listas (Unidad III).
- Revisaremos **break** y **continue** con más detalle.
- Pronto veremos estructuras de datos (listas, arreglos) e interacción con bucles para manipular secuencias.

- Python Docs - Control Flow
- Learn Python - Conditions
- Real Python - While Loops
- Foros y Comunidades: Stack Overflow, Reddit /r/learnpython.

¡Gracias por su atención!

- Aseguren de **guardar** sus notebooks de hoy.
- ¡Nos vemos en la próxima sesión para seguir avanzando con **for**, listas y más ejemplos!