

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



Ejercicios Organizados por Estructuras de Control

Introducción

Ejercicios con IF-ELIF-ELSE

Ejercicios con FOR/WHILE

Ejercicios Combinados

Ejercicios Extra Desafiantes

Resumen y Conclusiones

Introducción

- **Separación por estructura de control:**
 - Ejercicios con `if-elif-else` (condicionales)
 - Ejercicios con `for/while` (bucles)
 - Ejercicios combinados (ambas estructuras)
- **Dificultad progresiva:** De simple a complejo
- **Contexto físico:** Aplicaciones en física y matemáticas
- **Soluciones completas:** Código funcional y documentado

Ejercicios con IF-ELIF-ELSE

Ejercicios con IF-ELIF-ELSE \ni Ejercicio 1:

Clasificación de Estados del Agua



Enunciado

- Solicitar al usuario la temperatura del agua en grados Celsius.
- Usar **if-elif-else** para clasificar el estado:
 - Menor a 0°C: "Sólido (hielo)"
 - Entre 0°C y 100°C: "Líquido"
 - Mayor a 100°C: "Gaseoso (vapor)"
- Mostrar el estado y el nivel de energía molecular.

Conceptos: Condicionales básicas, comparaciones numéricas.

Física relevante: Estados de la materia y cambios de fase.

Ejercicios con IF-ELIF-ELSE \ni Solución 1 de Referencia:

Clasificación de Estados del Agua



```
1  # Solicitar temperatura al usuario
2  temperatura = float(input("Temperatura del agua (°C): "))
3
4  # Clasificar estado usando if-elif-else
5  if temperatura > 100:
6      estado = "gaseoso (vapor)"
7      energia = "alta"
8  elif temperatura >= 0:
9      estado = "líquido"
10     energia = "media"
11 else: # temperatura < 0
12     estado = "sólido (hielo)"
13     energia = "baja"
14
15 # Mostrar resultados
16 print(f"A {temperatura}°C, el agua está en estado {estado}")
17 print(f"Nivel de energía cinética molecular: {energia}")
```

Ejercicios con IF-ELIF-ELSE \ni Ejercicio 2:

Clasificador de Velocidades



Enunciado

- Pedir al usuario la velocidad de un objeto (m/s) y su masa (kg).
- Clasificar según rangos físicos:
 - $v < 1$ m/s: "Movimiento lento"
 - $1 \leq v < 10$ m/s: "Movimiento moderado"
 - $10 \leq v < 100$ m/s: "Movimiento rápido"
 - $v \geq 100$ m/s: "Movimiento muy rápido"
- Calcular y mostrar la energía cinética si se proporciona la masa.

Conceptos: Múltiples condicionales, validación de entrada.

Física relevante: Escalas de velocidad y energía cinética.

Ejercicios con IF-ELIF-ELSE ⇒ Solución 2 de Referencia:

Clasificador de Velocidades



```
1  # Entrada de datos
2  velocidad = float(input("Velocidad del objeto (m/s): "))
3  masa = float(input("Masa del objeto (kg, 0 si no aplica): "))
4
5  # Clasificación de velocidad
6  if velocidad < 1:
7      categoria = "Movimiento lento"
8  elif velocidad < 10:
9      categoria = "Movimiento moderado"
10 elif velocidad < 100:
11     categoria = "Movimiento rápido"
12 else:
13     categoria = "Movimiento muy rápido"
14
15 print(f"Velocidad: {velocidad} m/s → {categoria}")
16
17 # Cálculo de energía cinética si se proporciona masa
18 if masa > 0:
19     energia_cinetica = 0.5 * masa * velocidad**2
```

Ejercicios con IF-ELIF-ELSE \ni Ejercicio 3:

Conversión de Unidades con Validación



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 completa, manejo de errores.

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

Ejercicios con IF-ELIF-ELSE \ni Solución 3 de Referencia:

Conversión de Unidades con Validación



```
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":
```

Ejercicios con FOR/WHILE

Ejercicios con FOR/WHILE \ni Ejercicio 4: Tabla de Multiplicar



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** básicos, validación simple.

Física relevante: Relaciones proporcionales, escalado de magnitudes.

Ejercicios con FOR/WHILE \ni Solución 4 de Referencia:

Tabla de Multiplicar



```
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: Bucle `for` con `range()`, validación con `if`.

Ejercicios con FOR/WHILE \ni Ejercicio 5: Suma de Números Pares



Enunciado

- Calcular la suma de todos los números pares desde 2 hasta un número n dado por el usuario.
- Usar un bucle **for** con **range()**.
- Mostrar cada número par que se suma y el total final.
- Verificar el resultado usando la fórmula: $\sum_{k=1}^{n/2} 2k = n(n/2 + 1)$ para n par.

Conceptos: Bucles **for**, acumuladores, verificación matemática.

Física relevante: Sumas de series, análisis de datos.

Ejercicios con FOR/WHILE \ni Solución 5 de Referencia:

Suma de Números Pares



```
1  # Entrada de datos
2  n = int(input("Ingrese el número límite: "))
3
4  # Inicializar acumulador
5  suma_pares = 0
6
7  print(f"Números pares desde 2 hasta {n}:")
8
9  # Bucle para sumar números pares
10 for numero in range(2, n + 1, 2): # Desde 2, hasta n+1, de 2 en
    ↪ 2
11     suma_pares += numero
12     print(f"  + {numero}")
13
14 print(f"Suma total de números pares: {suma_pares}")
15
16 # Verificación con fórmula (solo si n es par)
17 if n % 2 == 0:
18     formula_resultado = n * (n // 2 + 1)
```


Ejercicios con FOR/WHILE \ni Ejercicio 6:

Aproximación a la Raíz Cuadrada



Enunciado

- Calcular la raíz cuadrada de un número usando el método de Newton-Raphson.
- Usar la fórmula iterativa: $x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n})$
- Usar un bucle **while** que continúe hasta que la precisión sea menor a 0.0001.
- Mostrar cada iteración y el resultado final.
- Verificar comparando con el valor real.

Conceptos: Bucles **while**, precisión numérica, métodos iterativos.

Física relevante: Métodos numéricos en simulaciones físicas.

Ejercicios con FOR/WHILE \ni Solución 6 de Referencia:

Aproximación a la Raíz Cuadrada



```
1  # Entrada de datos
2  numero = float(input("Número para calcular raíz cuadrada: "))
3  aproximacion = numero / 2  # Estimación inicial
4  tolerancia = 0.0001
5  iteracion = 0
6
7  print(f"Calculando raíz cuadrada de {numero}...")
8
9  # Método de Newton-Raphson
10 while abs(aproximacion**2 - numero) > tolerancia:
11     iteracion += 1
12     aproximacion = (aproximacion + numero/aproximacion) / 2
13     print(f"Iteración {iteracion}: {aproximacion:.6f}")
14
15 print(f"Raíz cuadrada  $\approx$  {aproximacion:.6f}")
16 print(f"Verificación: {aproximacion}2 = {aproximacion**2:.6f}")
17 print(f"Error: {abs(aproximacion**2 - numero):.6f}")
```

Ejercicios Combinados

Ejercicios Combinados \ni Ejercicio 7:

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/s^2)
 - 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: Entrada de datos, cálculos directos.

Física relevante: Movimiento rectilíneo uniformemente acelerado (MRUA).

Ejercicios Combinados \ni Solución 7 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: Aplicación directa de fórmulas físicas, formato de salida.

Ejercicios Combinados \ni Ejercicio 8: Promedio y Varianza de Mediciones



Enunciado

- Solicitar al usuario **3 mediciones** físicas (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.

Ejercicios Combinados \ni Solución 8 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}")
```

Ejercicios Combinados \ni Ejercicio 9:

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.

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

Ejercicios Combinados \ni Solución 9 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")
```

Ejercicios Combinados \ni Ejercicio 10:

Juego de Adivinanza con Física



Enunciado

- El programa "piensa" en la velocidad de la luz en el vacío (299,792,458 m/s).
- El usuario debe adivinar este número.
- Usar un bucle **while** que continúe hasta que el usuario acierte.
- Dar pistas usando **if-elif-else**: "muy alto", "alto", "bajo", "muy bajo".
- Contar el número de intentos y mostrar estadísticas finales.

Conceptos: Bucles **while**, condicionales complejas, contadores.

Física relevante: Constantes físicas fundamentales.

Ejercicios Combinados \ni Solución 10 de Referencia:

Juego de Adivinanza con Física



```
1  # Número secreto: velocidad de la luz en m/s
2  numero_secreto = 299792458
3  intentos = 0
4
5  print("Adivina la velocidad de la luz en el vacío (m/s)")
6  print("Pista: es un número de 9 dígitos")
7
8  while True:
9      intento = int(input("Tu estimación: "))
10     intentos += 1
11
12     diferencia = abs(intento - numero_secreto)
13
14     if intento == numero_secreto:
15         print(f";Correcto! La velocidad de la luz es
16         ↪ {numero_secreto} m/s")
17         print(f"Lo lograste en {intentos} intentos")
18         break
19     elif diferencia < 10000000: # Dentro de 1 millón
```

Ejercicios Extra Desafiantes



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.
- Usar **while** para permitir múltiples cálculos hasta que elija salir.

Objetivo: Integrar menús, condicionales, bucles y cálculos físicos.

Ejercicios Extra Desafiantes \ni Actividad Extra 2: Simulador de Caída Libre



Enunciado

- Simular la caída libre de un objeto usando: $y(t) = y_0 - \frac{1}{2}gt^2$
- Pedir altura inicial y_0 y gravedad g (9.81 m/s² por defecto).
- Usar un bucle **while** para calcular la posición cada 0.1 segundos.
- Detenerse cuando el objeto toque el suelo ($y \leq 0$).
- Mostrar tabla con tiempo, altura y velocidad en cada paso.
- Usar **if** para detectar cuando toca el suelo.

Objetivo: Bucles **while** con condiciones físicas y simulación temporal.

Resumen y Conclusiones

Ejercicios IF-ELIF-ELSE

- **Básico:** Estados del agua (1 condicional simple)
- **Intermedio:** Clasificador de velocidades (múltiples condiciones + validación)
- **Avanzado:** Conversión de unidades (validación completa + manejo de errores)

Ejercicios FOR/WHILE

- **Básico:** Tabla de multiplicar (for simple)
- **Intermedio:** Suma de pares (for con acumulador)
- **Avanzado:** Raíz cuadrada (while con precisión)

Ejercicios Combinados

- **Básico:** Ecuación de movimiento (cálculo directo)
- **Intermedio:** Promedio y varianza (for + estadística)
- **Avanzado:** Clasificador de temperaturas (for + if anidados)

- **Estructuras de control integradas:**
 - `if-elif-else`: Toma de decisiones múltiples
 - `for` loops: Iteración con conteo conocido
 - `while` loops: Iteración con condición variable
- **Conceptos de programación aplicados:**
 - Acumuladores y contadores
 - Validación de entrada de usuario
 - Manejo de errores y casos especiales
 - Listas dinámicas y procesamiento de datos
- **Aplicaciones físicas:**
 - Ecuaciones cinemáticas y dinámicas
 - Análisis estadístico de mediciones
 - Constantes físicas y conversiones de unidades
 - Métodos numéricos y simulaciones

Para cada ejercicio, evaluar:

- **Funcionalidad (40%):** ¿El código ejecuta sin errores?
- **Uso correcto de estructuras (30%):** ¿Se usan las estructuras apropiadas?
- **Legibilidad y comentarios (20%):** ¿Es fácil entender el código?
- **Manejo de casos especiales (10%):** ¿Se validan las entradas?

Recomendaciones

- Comenzar con ejercicios básicos antes de avanzar
- Probar con diferentes valores de entrada
- Documentar el razonamiento en comentarios
- Verificar resultados físicos cuando sea posible

- **Práctica recomendada:**
 - Comenzar con ejercicios básicos (1-4)
 - Avanzar gradualmente a ejercicios intermedios (5-8)
 - Desafiarse con ejercicios avanzados (9-10)
 - Intentar las actividades extra como proyectos
- **Temas siguientes:**
 - Funciones y modularización de código
 - Listas y estructuras de datos
 - Bibliotecas científicas (NumPy, Matplotlib)
- **Recursos adicionales:**
 - Practicar en plataformas como HackerRank o LeetCode
 - Explorar aplicaciones en física computacional
 - Documentar soluciones en un portafolio personal

¡Excelente trabajo!

- Ya dominan las **estructuras de control fundamentales**
- Pueden aplicar programación a **problemas físicos complejos**
- Están preparados para **conceptos más avanzados**
- Recuerden: **la práctica constante es clave**

¡Continúen explorando y aprendiendo!