


Kata: Estructuras de Control (Bucles)




Descripción General

¡Bienvenido a esta kata de programación! 

En esta práctica, fortalecerás tu comprensión de **bucles** (for, while) y **manipulación de datos** en Python . A través de ejercicios progresivos, aprenderás a:





- Automatizar tareas repetitivas.
- Filtrar y transformar datos.
- Implementar lógica condicional en ciclos.
- Mejorar tu eficiencia al resolver problemas.

Cada ejercicio incluye:


-  **Objetivo:** Qué lograrás.
-  **Instrucciones:** Pasos concretos para resolverlo.
-  **Preguntas de reflexión:** Para profundizar en el aprendizaje.

Requisitos

Antes de comenzar, asegurate de tener lo siguiente listo:

-  Python instalado (versión 3.8 o superior)
 -  Editor de código (recomendado: VS Code)
 -  Muchas ganas de aprender y divertirse resolviendo desafíos 
-

Ejercicio 1: Bucle for para números pares

 **Objetivo:** Imprimir números pares usando un bucle for.


 **Instrucciones:**

1. Escribe un bucle for que imprima los números pares del **2 al 20** (inclusive).
2. Usa un condicional o el paso del rango para lograrlo.

 **Preguntas de reflexión:**

- ¿Cómo modificarías el código para imprimir solo impares?
 - ¿Qué pasa si el rango fuera de 2 a 20 con paso 3?
-

Ejercicio 2: Bucle while con suma acumulativa

 **Objetivo:** Usar un bucle while para controlar una condición de salida.


 **Instrucciones:**

1. Pide al usuario que ingrese números hasta que la suma supere **100**.
2. Imprime la suma total al final.

 **Preguntas de reflexión:**

- ¿Qué ocurre si el primer número ingresado es mayor que 100?
 - ¿Cómo evitarías errores si el usuario ingresa texto?
-

Ejercicio 3: Filtrar palabras por letra inicial

 **Objetivo:** Iterar sobre una lista y aplicar filtros.


 **Instrucciones:**

1. Dada una lista de palabras (ej: ["apple", "banana", "avocado"]), imprime solo las que empiezan con **"a"**.

 **Preguntas de reflexión:**

- ¿Cómo harías que la comparación sea case-insensitive (ej: "Apple" también se cuente)?
 - ¿Qué método de strings es útil para esto?
-

Ejercicio 4: Tabla de multiplicar del 7

 **Objetivo:** Usar un bucle para generar patrones.

 **Instrucciones:**

1. Imprime la tabla de multiplicar del **7** (desde $7 \times 1 = 7$ hasta $7 \times 10 = 70$).

 **Preguntas de reflexión:**

- ¿Cómo adaptarías el código para que el usuario elija la tabla?
 - ¿Qué estructura usarías para almacenar los resultados?
-

Ejercicio 5: Contador de vocales

 **Objetivo:** Contar caracteres específicos en un string.

 **Instrucciones:**

1. Pide al usuario una cadena de texto.
2. Cuenta y muestra cuántas **vocales** (a, e, i, o, u) contiene.

 **Preguntas de reflexión:**

- ¿Cómo manejarías las vocales acentuadas (á, é)?
 - ¿Qué estructura de datos te ayudaría a optimizar el código?
-

Ejercicio 6: Números repetidos en una lista

 **Objetivo:** Filtrar elementos duplicados manteniendo el orden.


 **Instrucciones:**

1. Dada una lista (ej: [3, 1, 3, 5, 1]), crea una nueva lista con los números que aparecen **más de una vez** (en este caso: [3, 1]).

 **Preguntas de reflexión:**

- ¿Por qué es importante mantener el orden de aparición?
 - ¿Cómo resolverías esto sin usar estructuras adicionales?
-

Ejercicio 7: FizzBuzz

 **Objetivo:** Implementar lógica condicional en bucles.

 **Instrucciones:**

1. Imprime números del **1 al 100**, pero:
 - Para múltiplos de 3 → "Fizz".
 - Para múltiplos de 5 → "Buzz".
 - Para múltiplos de ambos → "FizzBuzz".

 **Preguntas de reflexión:**

- ¿Por qué el orden de los condicionales es crucial aquí?
 - ¿Cómo extenderías el juego a otros números (ej: 7 → "Bazz")?
-

Ejercicio 8: Frecuencia de palabras

 **Objetivo:** Usar diccionarios para contar elementos.


 **Instrucciones:**

1. Dada una cadena (ej: "hola hola mundo"), devuelve un diccionario con la frecuencia de cada palabra (ej: {"hola": 2, "mundo": 1}).

 **Preguntas de reflexión:**

- ¿Cómo ignorarías signos de puntuación (ej: "hola," vs "hola")?
 - ¿Qué método de strings ayuda a separar palabras?
-

✳ Ejercicio 9: Filtrar consonantes

 **Objetivo:** Manipular strings con condiciones.


 **Instrucciones:**

1. Dada una cadena, crea una nueva cadena que solo contenga sus **consonantes** (ej: "Hola" → "Hl").

 **Preguntas de reflexión:**

- ¿Cómo manejarías las mayúsculas/minúsculas?
 - ¿Qué estructura usarías para definir las consonantes?
-

Ejercicio 10: Números primos

 **Objetivo:** Implementar un algoritmo con bucles anidados.

 **Instrucciones:**

1. Pide al usuario un número entero positivo.
2. Imprime todos los **números primos** menores o iguales a ese número.

 **Preguntas de reflexión:**

- ¿Cómo optimizarías la verificación de primos?
 - ¿Qué ventaja tiene usar `range(2, int(n**0.5) + 1)`?
-

 **Bonus:**

- **Diagrama de flujo:** Elige un ejercicio y dibuja su diagrama de flujo.
- **Reto extra:** Modifica un ejercicio para usar `break` o `continue`.