

Actividad: Calidad en Códigos

En esta actividad se desarrollan dos de los tres ejercicios propuestos en el documento "Calidad en códigos.pdf". Cada solución incluye una explicación de su funcionamiento, el código fuente con buenas prácticas y el análisis de complejidad algorítmica con notación Big-O.

Ejercicio 1: Detección de duplicados en una lista

El programa recibe varias listas de números enteros y determina si contienen elementos duplicados.

```
def detectar_duplicados(listas):
    for lista in listas:
        vistos = set()
        duplicado = False
        for num in lista:
            if num in vistos:
                duplicado = True
                break
            vistos.add(num)
        if duplicado:
            print("Duplicados encontrados")
        else:
            print("Sin duplicados")

# Ejemplo de uso
pruebas = [
    [1, 2, 3, 4, 5],
    [10, 20, 10, 30],
    [7, 8, 9]
]
detectar_duplicados(pruebas)
```

Complejidad Algorítmica: $O(n)$

Justificación: Se recorre cada lista una sola vez. Las operaciones de búsqueda e inserción en un conjunto tienen un costo promedio constante $O(1)$, por lo que el tiempo total crece linealmente con el tamaño de la lista.

Ejercicio 2: Número más frecuente (moda)

El programa determina cuál es el número que aparece con mayor frecuencia en una lista. En caso de empate, selecciona el número menor entre los empatados.

```
def numero_mas_frecuente(listas):
    for lista in listas:
        frecuencia = {}
        for num in lista:
            frecuencia[num] = frecuencia.get(num, 0) + 1
        max_freq = max(frecuencia.values())
        candidatos = [num for num, freq in frecuencia.items() if freq == max_freq]
        print(min(candidatos), max_freq)

# Ejemplo de uso
pruebas = [
    [2, 2, 3, 3, 1],
    [10, 10, 10, 5]
]
numero_mas_frecuente(pruebas)
```

Complejidad Algorítmica: $O(n)$

Justificación: Se recorre la lista una sola vez para contar las frecuencias ($O(n)$), y luego se determina

el máximo en el diccionario, lo que también es lineal en el tamaño de la lista. Por tanto, la complejidad total es $O(n)$.