



## ISIS-1221

# INTRODUCCIÓN A LA PROGRAMACIÓN

### Nivel 3 – Laboratorio 3

### Recorridos sobre listas

#### Objetivos

1. Desarrollar habilidades para procesar y manipular listas de números.
2. Practicar el uso de ciclos para buscar patrones en secuencias de datos.
3. Aprender a trabajar con estructuras de datos anidadas (listas de diccionarios).

#### Preparación del ambiente de trabajo

Para este laboratorio se le proporcionará un módulo de consola (`consola.py`) que contiene un menú interactivo para probar sus funciones.

Usted debe crear un nuevo módulo de funciones llamado `modulo.py` en el que escribirá las funciones correspondientes a las actividades 1 a 3 de este laboratorio. La consola importará automáticamente sus funciones desde este módulo.

#### Actividad 1: Buscar elementos iguales consecutivos

Escriba una función que reciba como parámetro una lista de números enteros y retorne la suma de las posiciones de la última ocasión en la que dos números seguidos tienen el mismo valor. Si no hay dos valores seguidos iguales, debe retornar -1.

Si el problema se puede solucionar mediante un recorrido parcial de la lista, debe hacerlo de esa manera.

**E.g.:**

- `[1, 2, 3, 3, 4, 5] → 5` (solo hay un par de consecutivos iguales, en las posiciones 2 y 3)
- `[1, 2, 3, 4, 5] → -1` (no hay elementos consecutivos iguales)
- `[5, 5, 1, 2, 5, 5, 5, 2] → 11` (la última vez que dos números seguidos tienen el mismo valor es en las posiciones 5 y 6)

## Actividad 2: Mejor estudiante por materia

Escriba una función que reciba como parámetros una lista de estudiantes y el nombre de una materia, y retorne el código del estudiante que tiene la mejor nota en esa materia.

La lista de estudiantes contiene diccionarios con la siguiente estructura:

- `codigo`: El código del estudiante (cadena de caracteres).
- `nombre`: El nombre del estudiante (cadena de caracteres).
- `IP`: La nota de Introducción a la Programación (número decimal).
- `calculo`: La nota de Cálculo (número decimal).
- `escritura`: La nota de Escritura Universitaria (número decimal).
- `constitucion`: La nota de Constitución (número decimal).

Si dos o más estudiantes tienen la misma nota máxima en la materia especificada, debe retornar el código del estudiante cuyo nombre sea alfabéticamente mayor (último en orden alfabético).

**E.g.:** Si la lista es:

```
[
  {"codigo": "202012345", "nombre": "Ana García", "IP": 4.5, "calculo": 3.8,
   "escritura": 4.2, "constitucion": 4.0},
  {"codigo": "202098765", "nombre": "Carlos López", "IP": 4.5, "calculo": 4.1,
   "escritura": 3.9, "constitucion": 4.3},
  {"codigo": "202055555", "nombre": "Beatriz Ruiz", "IP": 4.2, "calculo": 4.5,
   "escritura": 4.0, "constitucion": 3.7}
]
```

Para la materia "IP", la función debe retornar "202098765" (Carlos López), ya que tanto Ana como Carlos tienen 4.5, pero "Carlos López" es alfabéticamente mayor que "Ana García".

## Actividad 3: Contar apariciones de sublista

Escriba una función que reciba como parámetros una lista de números y otra lista de números más corta, y retorne la cantidad de veces en que la segunda lista (sublista) aparece consecutivamente en la primera lista.

No considere el caso en que aparezcan elementos intercalados. La sublista debe aparecer de forma consecutiva en la lista principal. Si el problema se puede solucionar mediante un recorrido parcial de la lista, debe hacerlo de esa manera.

**E.g.:**

- Lista: [1, 2, 3, 1, 4, 2, 3], Sublista: [1, 2, 3]  $\rightarrow$  1
- Lista: [1, 2, 3, 4, 5], Sublista: [6, 7]  $\rightarrow$  0
- Lista: [1, 1, 7, 1, 1, 2], Sublista: [1, 1]  $\rightarrow$  2

## Entrega

Entregue el archivo `modulo.py` a través de Brightspace en el laboratorio del Nivel 3 designado como "N3-L3: Recorridos sobre listas".

Puede usar el módulo `console.py` proporcionado para probar sus funciones antes de la entrega. Su archivo debe funcionar correctamente desde la consola.