

# Trabajo Práctico Integrador - Programación I

## Alumnos:

Ramiro García Villarroel – ramiro.garcia.villarroel@gmail.com -

Comisión 5

Leandro Daniel Schaberger – [leandroschaberger01@gmail.com](mailto:leandroschaberger01@gmail.com) -

Comisión 13

**Materia:** Programación I **Profesores:** Matías Santiago Torres – Cinthia Rigoni

## **Introducción:**

En esta sección se presentan los conceptos teóricos utilizados en el desarrollo del sistema, los cuales forman parte de los contenidos centrales de la materia Programación 1.

## **Listas:**

Las listas en Python son colecciones ordenadas y mutables de elementos, que pueden ser de diferentes tipos de datos, como números, cadenas de texto o booleanos. Se definen utilizando corchetes [] con los elementos separados por comas. Son muy versátiles, ya que permiten acceder a los elementos por índice, modificarlos, añadir o eliminar elementos y realizar operaciones como concatenar y replicar listas.

Ejemplo: lista\_paises = [pais1, pais2, pais3]

## **Diccionarios:**

Un diccionario de Python es una estructura de datos que almacena elementos en pares de clave-valor, donde cada clave es única e inmutable, y puede usarse para acceder a su valor correspondiente. Los diccionarios se definen utilizando llaves {} y son mutables y desordenados, permitiendo que los valores sean de cualquier tipo de dato.

Ejemplo: pais = {"nombre": "Argentina", "poblacion": 45376763, "superficie": 2780400, "continente": "América"}

## **Funciones:**

Una función en Python es un bloque de código reutilizable que realiza una tarea específica. Se definen usando la palabra clave `def`, un nombre, paréntesis y dos puntos. Las funciones ayudan a organizar el código, lo hacen más legible y facilitan la reutilización sin tener que escribir el mismo código varias veces.

Ejemplo: `def buscar_pais(lista, nombre):...`

### **Condicionales:**

Los condicionales en Python son instrucciones que permiten al programa tomar decisiones, ejecutando diferentes bloques de código dependiendo de si una condición es verdadera o falsa. Se utilizan para controlar el flujo de un programa, permitiéndole adaptarse a distintas situaciones y ejecutar acciones específicas basadas en la evaluación de expresiones lógicas u operadores de comparación. Las estructuras básicas son `if`, `elif` (else if) y `else`.

Ejemplo: `if poblacion > 0:...`

### **Ordenamientos:**

Los ordenamientos en Python son procesos para organizar datos en un orden específico, como ascendente o descendente. Se utilizan métodos como las funciones integradas `sorted()` y el método de lista `sort()`, así como algoritmos específicos como el ordenamiento burbuja, para hacer que los datos sean más comprensibles y fáciles de analizar o manipular.

Ejemplo: `lista_ordenada = sorted(lista_paises, key=lambda p: p["poblacion"])`

### **Estadísticas Básicas:**

En Python, las estadísticas básicas son herramientas para describir y resumir conjuntos de datos mediante el uso de funciones y bibliotecas como `statistics` (integrada) o `Pandas` (externa). Estas herramientas calculan medidas como la media, la mediana, la moda, la varianza y la desviación estándar para entender la tendencia central y la dispersión de los datos.

Ejemplo: `promedio = sum(p["poblacion"] for p in lista_paises) / len(lista_paises)`

### **Archivos CSV:**

Un archivo CSV (.csv, valores separados por comas) es un formato de texto plano para almacenar datos tabulares, donde cada línea representa una fila y los valores de esa fila están separados por comas. En Python, el módulo `csv` proporciona herramientas para leer y escribir estos archivos de manera estructurada, lo que facilita la manipulación de datos de hojas de cálculo y bases de datos.

Ejemplo: `import csv`

with open("paises.csv", newline="", encoding="utf-8") as archivo:

```
lector = csv.DictReader(archivo)
```

### Diagrama de Flujo:

