

## Trabajo Práctico Integrador (TPI)

### Gestión de Datos de Países en Python: filtros, ordenamientos y estadísticas

#### Objetivo

Desarrollar una aplicación en Python que permita gestionar información sobre países, aplicando **listas, diccionarios, funciones, estructuras condicionales y repetitivas, ordenamientos y estadísticas**. El sistema debe ser capaz de leer datos desde un archivo CSV, realizar consultas y generar indicadores clave a partir del dataset.

El objetivo principal es **afianzar el uso de estructuras de datos, modularización con funciones y técnicas de filtrado/ordenamiento**, aplicando los conceptos aprendidos en Programación 1.

#### Consignas generales

- Lenguaje: **Python 3.x**
- Estructuras: listas, diccionarios, funciones.
- Archivos: lectura desde **CSV**.
- Código claro, comentado y modularizado (una función = una responsabilidad).
- Validaciones de entradas y manejo básico de errores.
- Trabajo en **equipos de 2 personas**.

#### Dominio (dataset de países)

Cada país estará representado con los siguientes datos:

- **Nombre** (string)
- **Población** (int)
- **Superficie** en km<sup>2</sup> (int)
- **Continente** (string)

*Ejemplo de registro CSV:*

nombre,poblacion,superficie,continente

Argentina,45376763,2780400,América

Japón,125800000,377975,Asia

Brasil,213993437,8515767,América

Alemania,83149300,357022,Europa

#### Requerimientos técnicos

##### 1) Diseño (previo al código)

- Explicar en un **informe teórico** los conceptos aplicados:
  - Listas
  - Diccionarios
  - Funciones
  - Condicionales
  - Ordenamientos
  - Estadísticas básicas
  - Archivos CSV
- Definir el **flujo de operaciones principales** en un diagrama o esquema.

## 2) Funcionalidades mínimas del sistema

El programa debe ofrecer un **menú de opciones en consola** que permita:

- **Buscar un país** por nombre (coincidencia parcial o exacta).
- **Filtrar países** por:
  - Continente
  - Rango de población
  - Rango de superficie
- **Ordenar países** por:
  - Nombre
  - Población
  - Superficie (ascendente o descendente)
- **Mostrar estadísticas:**
  - País con mayor y menor población
  - Promedio de población
  - Promedio de superficie
  - Cantidad de países por continente

## 3) Validaciones

- Controlar errores de formato en el CSV.
- Evitar fallos al ingresar filtros inválidos o búsquedas sin resultados.
- Mensajes claros de éxito/error.

## Entregables (obligatorios)

### 1. Carpeta digital

- Marco teórico con fuentes bibliográficas.
- Código Python funcional, modular y comentado.
- Capturas de pantalla de ejecución de ejemplos.
- Conclusiones grupales sobre los aprendizajes.

### 2. Repositorio en GitHub

Debe incluir:

- Proyecto completo en Python.
- README.md con:
  - Descripción del programa.
  - Instrucciones de uso.
  - Ejemplos de entradas y salidas.
  - Participación de los integrantes.
- Archivo CSV con el dataset base.

### 3. Video tutorial (10–15 minutos)

- Explicación del problema planteado.
- Presentación de la estructura de datos utilizada.
- Demostración del programa funcionando.
- Reflexión final sobre el desarrollo del proyecto.

## Criterios de evaluación

- Correcta funcionalidad (búsquedas, filtros, ordenamientos, estadísticas).
- Uso correcto de estructuras de datos (listas y diccionarios).
- Calidad del código (modularización, legibilidad, comentarios).
- Documentación (README claro, informe teórico coherente).
- Presentación en video (tiempo adecuado, explicación técnica, participación equitativa).
- Entrega completa en GitHub con código, informe y CSV.