

UNIVERSIDAD TECNOLÓGICA NACIONAL

Tecnicatura Universitaria en Programación

Programación 1 – Trabajo Práctico Integrador Final

Integrantes:

Diego Fernandez – Comisión 5

Renzo Galetto – Comisión 5

Docente:

Profesora – Cinthia Rigoni

Profesor Matías Torres

Gestor de países – Proyecto programación 1

Este programa permite gestionar información sobre los países a partir de un archivo CSV.

Incluye funciones de búsqueda, filtrado, ordenamiento y estadísticas básicas, lo desarrollamos aplicando listas, diccionarios, funciones, condicionales y bucles

FUNCIONALIDADES

- Leer un archivo CSV con datos de países
- Buscar un país por nombre (parcial)
- Filtrar por:
 - Continente
 - Rango de población
 - Rango de superficie
- Ordenar países por:
 - Nombre
 - Población
 - Superficie (ascendente o descendente)
- Calcular estadísticas:
 - País con mayor y menor población
 - Promedio de población
 - Promedio de superficie

Cantidad de países por continente

CONCEPTOS APLICADOS:

Listas: para almacenar todos los países cargados en el CSV, las listas nos permiten almacenar múltiples elementos ordenados y modificables. En este proyecto utilizamos una lista principal llamada “países”, donde cada elemento representa un país.

Ejemplo:

Países =

```
[{"nombre": "Argentina", "población": 45376763, "superficie": 2780400, "continente": "América"}]
```

Diccionarios: Para representar cada país con sus atributos, en esta estructura almacenamos datos en pares *clave: valor*

Esto facilita el acceso rápido a los datos por nombre de campo

```
(P["nombre"], p["población"], etc)
```

Funciones: Cada acción del sistema está separada por funciones, cada bloque cumple una única función.

Ejemplo:

Cargar_paises (archivo) → lee el CSV y devuelve la lista de países

Buscar_pais (países, nombre) → busca coincidencias

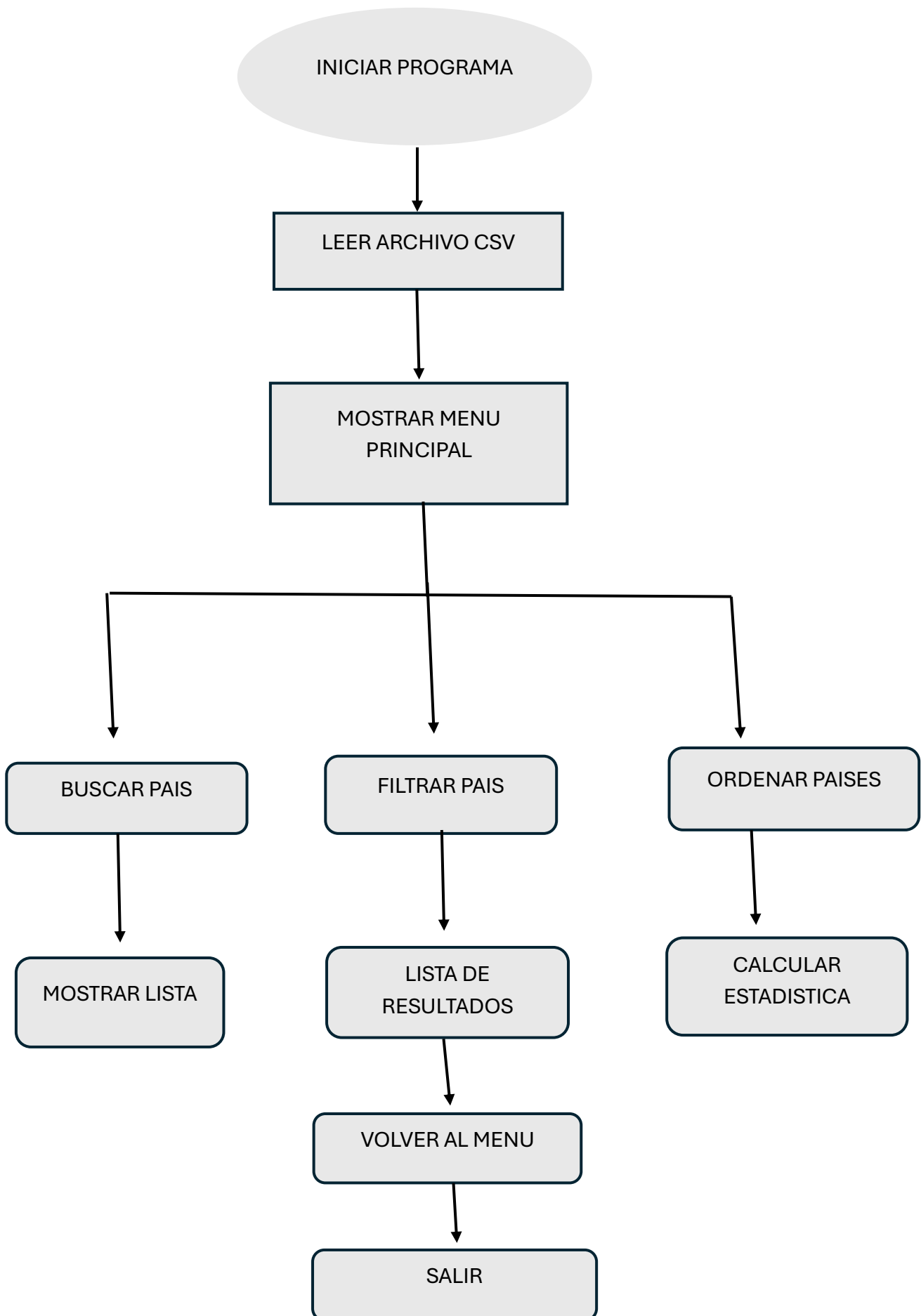
Estadísticas(países) → calcula promedios y máximos

Condicionales y bucles: Control del menú y validación de entradas, empleamos condicionales (if, else, elif) para controlar el flujo según las elecciones del usuario y validar entradas.

Utilizamos bucles for y while para recorrer listas y mantener activo el menú hasta que el usuario decida salir.

CSV: Lectura estructurada del dataset con el módulo CSV, Python nos permite leer datos desde un archivo separado por comas, cada fila se transforma en un diccionario que se agrega a la lista principal de países.

DIAGRAMA DE FLUJO – TRABAJO INTEGRADOR



ARCHIVO CSV (EJEMPLO)

NOMBRE	POBLACION	SUPERFICIE	CONTINENTE
Argentina	45376763	2780400	América
Japón	125800000	377975	Asia
Brasil	213993437	8515767	América
Alemania	83149300	357022	Europa

VALIDACIONES:

- Verificación de existencia del archivo CSV
- Control de formato de columnas y datos numéricos
- Evita fallos ante búsquedas sin resultados
- Muestra mensajes de error o éxito
- Permite reintentar operaciones o volver al menú

RESULTADOS ESPERADOS:

El sistema permite ejecutar consultas de manera ágil desde la consola, mostrando información clara y organizada.

El usuario puede comparar países, conocer promedio y analizar datos demográficos básicos.

CONCLUSIONES

El desarrollo de este proyecto nos permitió aplicar los conocimientos adquiridos en la materia programación 1, consolidando el uso de:

- Estructuras de datos (LISTAS Y DICCIONARIOS)
- Modularización de código mediante funciones
- Lectura de archivos externos y validaciones de formato
- Estructuras condicionales y repetitivas para control de flujo
- Procesamiento y análisis de datos básicos (ORDENAMIENTO Y ESTADÍSTICAS)

El trabajo contribuyó al fortalecimiento de la lógica algorítmica y la organización del código en programas reales, preparándonos para futuros proyectos más complejos en Python.