# INFORME TÉCNICO – GESTIÓN DE DATOS DE PAÍSES EN PYTHON



Integrantes:  
- Ezequiel Menéndez  
- Santiago Barretto  
- Nicolás Morón

Profesora: Cinthia Rigoni  
Materia: Programación 1  
Año: 2025

## ÍNDICE

1. Introducción

2. Marco Teórico

2.1. Listas y diccionarios en Python

2.2. Modularización y funciones

2.3. Validaciones y manejo de errores

2.4. Lectura y escritura de archivos CSV

2.5. Consumo de API externa

3. Diseño del Sistema

3.1. Representación de los datos

3.2. Estructura de archivos y módulos

3.3. Flujo principal del programa

3.4. Funcionalidades principales

3.5. Validaciones implementadas

3.6. Diagrama de flujo (descripción textual)

4. Implementación

4.1. Descripción de cada módulo

4.2. Ejemplo de ejecución

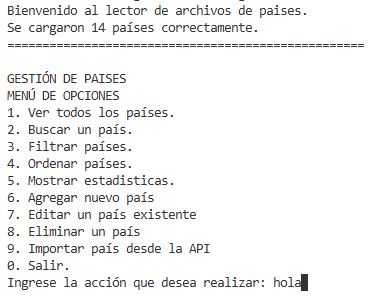
5. Conclusiones

6. Referencias Bibliográficas

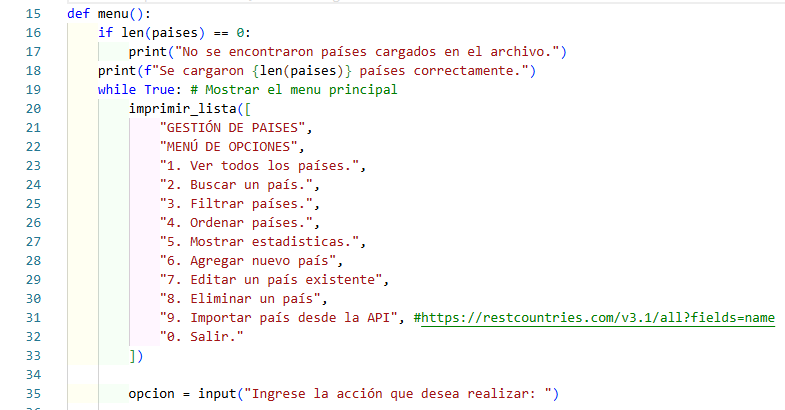
7. Anexos

## 1. Introducción

El presente informe técnico describe el desarrollo del proyecto “Gestión de Datos de Países”, realizado en Python como trabajo integrador final de la materia Programación 1.  
  
El objetivo principal del proyecto fue aplicar los conocimientos adquiridos sobre estructuras de datos, modularización, validaciones, manejo de archivos y consumo de API, desarrollando un sistema capaz de gestionar, analizar y actualizar información sobre países del mundo.  
  
El programa ofrece un menú interactivo en consola que permite al usuario realizar búsquedas, filtros, ordenamientos y cálculos estadísticos sobre los países, así como también agregar, modificar, eliminar o importar países desde una API externa.







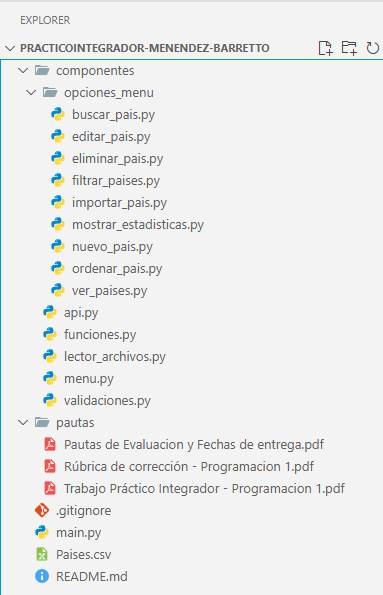
## 2. Marco Teórico

### 2.1. Listas y diccionarios en Python

Los diccionarios representan los países con sus atributos (nombre, población, superficie, continente), mientras que las listas almacenan el conjunto de países.

### 2.2. Modularización y funciones

El código se divide en módulos que agrupan funcionalidades relacionadas, mejorando la legibilidad y el trabajo en equipo.



### 2.3. Validaciones y manejo de errores

Se implementan validaciones para entradas de usuario y manejo de excepciones como ValueError y FileNotFoundError.

### 2.4. Lectura y escritura de archivos CSV

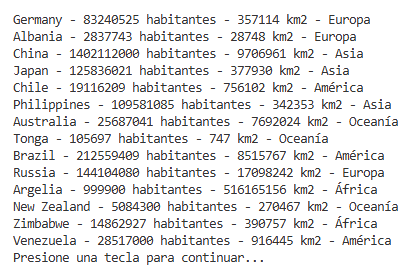
El sistema maneja un archivo Paises.csv validando su formato y convirtiendo los datos a diccionarios.

### 2.5. Consumo de API externa

Se utiliza el módulo api.py para conectarse a restcountries.com y actualizar la base local.

## 3. Diseño del Sistema

#### 3.1. Representación de los datos

Cada país se representa como un diccionario:

#### Todos los países se almacenan en una lista llamada paises.

#### 3.2. Estructura de archivos y módulos

El proyecto está organizado del siguiente modo:

* main.py: Punto de entrada del programa.
* menu.py: Controla la navegación del menú principal usando match-case.
* Carpeta opciones\_menu/: Contiene las funcionalidades (ver, buscar, filtrar, ordenar, estadísticas, agregar, editar, eliminar, importar desde API).
* funciones.py: Funciones reutilizables para entrada de datos y manejo de listas.
* lector\_archivos.py: Lectura y validación del archivo Paises.csv.
* validaciones.py: Validaciones de formato y datos.
* api.py: Comunicación con la API de países.

#### 3.3. Flujo principal del programa

1. Se carga el archivo Paises.csv.
2. Si el archivo no existe, se crea uno nuevo.
3. Se muestra el menú principal.
4. El usuario elige una opción y el sistema ejecuta la función correspondiente.
5. El ciclo continúa hasta seleccionar “Salir”.

#### 3.4. Funcionalidades principales

* Ver todos los países cargados.
* Buscar país por nombre.
* Filtrar por continente, población o superficie.
* Ordenar por nombre, población o superficie (asc/desc).
* Mostrar estadísticas globales (máximos, mínimos, promedios, conteos).
* Agregar, modificar o eliminar un país.
* Importar país desde API externa.

#### 3.5. Validaciones implementadas

* Detección de campos vacíos o inválidos.
* Verificación de duplicados.
* Control de rangos numéricos.
* Manejo de errores en lectura y escritura de archivos.

#### 3.6. Diagrama de flujo (descripción textual)

Inicio → Cargar datos desde Paises.csv  
→ Validar archivo  
→ Mostrar menú  
→ Según la opción seleccionada:

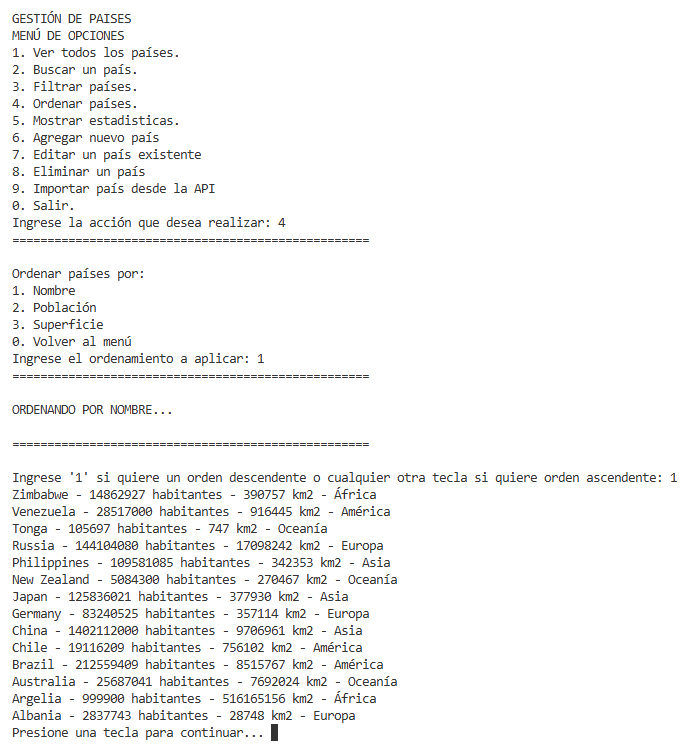
* Buscar país
* Filtrar países
* Ordenar países
* Mostrar estadísticas
* Agregar/editar/eliminar país
* Importar desde API  
  → Mostrar resultados  
  → Repetir hasta “Salir”.

### 4. Implementación

#### 4.1. Descripción de cada módulo

* **main.py:** inicia la ejecución del programa.
* **menu.py:** administra la navegación principal.
* **funciones.py:** contiene funciones comunes de entrada y visualización.
* **validaciones.py:** agrupa validaciones y manejo de datos.
* **lector\_archivos.py:** lee y guarda los países en Paises.csv.
* **api.py:** conecta con la API restcountries.com.
* **opciones\_menu/**: implementa cada funcionalidad del menú principal.

#### 4.2. Ejemplo de ejecución



## 5. Conclusiones

El desarrollo de este sistema permitió aplicar en un proyecto real los conceptos de estructuras de datos, validaciones, manejo de archivos y consumo de API.  
Gracias a la división modular, el código resultó claro, reutilizable y fácil de mantener.

Cada integrante cumplió un rol fundamental:

* Ezequiel Menéndez: estructura del proyecto, lectura de archivos y manejo de errores.
* Santiago Barretto: desarrollo del menú interactivo, filtrado y ordenamiento.
* Nicolás Morón: implementación de la API externa y gestión de datos dinámicos.

El proyecto logró cumplir con todos los objetivos propuestos y sentó una base sólida para futuros desarrollos en el área de programación y análisis de datos.

## 6. Referencias Bibliográficas

 Python Software Foundation. (2025). Python 3 Documentation. <https://docs.python.org/3/>

 Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. Green Tea Press.

 API REST Countries. https://restcountries.com/

 Repositorio oficial del proyecto:  
[https://github.com/EzequielMenendez/PracticoIntegrador-Menendez-Barretto-Moron](https://github.com/EzequielMenendez/PracticoIntegrador-Menendez-Barretto-Moron?utm_source=chatgpt.com)

### 7. Anexos

***7.1. Código fuente completo***  
Disponible en el repositorio GitHub.

***7.2. Archivo de datos Paises.csv***Contiene el dataset base con nombre, población, superficie y continente.

***7.3. Ejemplo de salida de estadísticas***