



# Instrucciones Miniproyecto 1

## Descripción

En este miniproyecto deberás completar un código que utilice estructuras secuenciales y no secuenciales para obtener información a partir de una base de datos entregada.

## Trabajo a Realizar

El objetivo de este trabajo es trabajar con un archivo “sucio” (es decir, desordenado y con filas repetidas), teniendo que procesar y guardar los datos para responder diversas consultas.

Para lograr esto, se entregará un archivo MP1.zip, con cuatro archivos:

- `main.py`, es el único archivo que debe modificar
- `pokemon.csv` es un documento en formato CSV. Este archivo no debe ser modificado.
- `input.txt` es un documento de texto plano que les servirá para probar su trabajo. Se entregará más información de este archivo en la sección “Método de Corrección”
- `output.txt` es un documento de texto plano que les servirá para probar su trabajo. Se entregará más información de este archivo en la sección “Método de Corrección”

El formato del archivo CSV entregado contiene líneas con valores separados coma (,). Los contenidos de cada línea son los siguientes:

`id,nombre,tipo_primario;tipo_secundario,HP,ataque,defensa,generacion`

Los significados de cada valor son:

- `id`. Id del pokemon (NO de la fila del dataset).
- `nombre`. Nombre del pokemon.
- `tipo_primario;tipo_secundario`. Es un string con 2 valores separados por punto y coma (;), el primer valor indica el tipo principal del pokemon, y el segundo valor indica su tipo secundario.
- `HP`. Es un valor numérico que indica la vida máxima del pokemon.
- `ataque`. Es un valor numérico con la fuerza de ataque del pokemon.
- `defensa`. Es un valor numérico con la capacidad de defensa del pokemon.
- `generacion`. Es un valor numérico que representa la generación a la que pertenece el pokemon

El código principal que se entrega contiene el flujo del menú principal, que recibe un input indicando que consulta hacer, y ejecuta la función correspondiente. Usted debe realizar las siguientes acciones.

1. **Cargar los datos.** Lo primero que debe hacer su código es cargar los datos del archivo .csv entregado a diversas estructuras. Deben estar las siguientes estructuras mínimas, pero pueden crear más si lo consideran necesario.

#### 1.1. **tipos\_de\_pokemon:**

Esta estructura debe almacenar todos los tipos de pokemon que aparecen en el archivo. No se hace diferenciación entre tipos primarios y secundarios, simplemente se debe guardar una vez cada tipo.

#### 1.2. **pokemon\_por\_tipo:**

Esta estructura debe asociar los ids de pokemon a sus tipos. Es decir, para cada tipo de pokemon, deben guardarse los ids de los pokemon que lo tienen como tipo primario y los que lo tienen como tipo secundario.

#### 1.3. **info\_pokemon:**

Esta estructura debe asociar toda la información del pokemon menos el tipo (es decir, nombre, HP, Ataque, defensa, generación.) al id de este.

2. **Completar las funciones de consultas.** En el código principal entregado notarás que hay varias definiciones de funciones que no están completas. Estas son las consultas que se le harán a los datos que obtuviste en el punto anterior. Las funciones son las siguientes:

#### 2.1 **filtrar\_y\_ordenar\_por(tipo\_pokemon, criterio)**

Esta función recibe un string que representa un tipo de pokemon. Debe filtrar los pokemon que tengan ese tipo (ya sea primario o secundario) y entregar los nombres de estos en una lista. Esa lista debe venir ordenada de mayor a menor según el criterio recibido por la función (que puede ser HP, Ataque o Defensa).

#### 2.2 **estadísticas(tipo\_pokemon, criterio)**

Esta función recibe un string que representa un tipo de pokemon. Debe analizar los pokemon que tengan ese tipo (ya sea primario o secundario) y obtener los estadísticos Máximo, Mínimo y Promedio del valor dado por criterio (que puede ser HP, Ataque o Defensa). Esto debe ser entregado en formato de diccionario, donde las llaves son el nombre del estadístico ("max", "min", "prom") y los valores deben ser justamente, los valores obtenidos.

#### 2.3 **tipos(nombre\_pokemon)**

Esta función recibe el nombre de un pokemon y debe retornar una tupla, donde el primer elemento es el tipo primario del pokemon y el segundo elemento es su tipo secundario. En caso de tener únicamente tipo primario, el segundo elemento debe ser un string vacío.

## Método de Corrección

El código se corregirá automáticamente mediante tests. **Es por eso que es muy importante que no toques el código principal ni las funciones de output**, y que respetes los nombres de las estructuras/variables/llaves entregadas en el enunciado, ya que la mínima diferencia puede hacer que tu código se caiga y no sea evaluado correctamente.

En este caso el input y el output son manejados por el código principal, pero es recomendable prestar atención a cómo se hace, ya que en futuros proyectos tendrán que manejarlo ustedes.

La base de datos es única para todos los tests. Pero se subirá el input.txt de ejemplo que podrán utilizar como input para probar su código. También se subirá el archivo output.txt que mostrará el output esperado para dicho input. Cabe volver a mencionar que los outputs deben ser **EXACTAMENTE IGUALES** para conseguir el puntaje de cada sección.