

Descripción.

Se realizaron avances en el proyecto, incluidas validaciones a la entrada del usuario, un menú de opciones, solicitudes y uso de expresiones regulares para validar las solicitudes al api, limpieza de datos, una nueva estructura de datos (lista de diccionarios) utilizada para facilitar el acceso a la información relevante de la api y, con esto último, se incluyó una forma de crear o reescribir archivos en formato .txt con dicha estructura, almacenando en ellos los datos solicitados a la api.

Métodos de extracción de datos y validaciones.

Utilizando el módulo “requests” y la función “get”, se pueden realizar consultas a la api en formato “json”, sin embargo, en el código se cambia de este formato de datos a diccionarios con la función “.json”; esto ya se había hecho con anterioridad en la primera versión del código, más en este agregamos un método de validación utilizando expresiones regulares con el módulo “re”. Se toma la entrada del usuario y se busca una coincidencia con el formato de ID que tiene la api, después, mediante un bucle “while not” se devuelve un mensaje de error y se le solicita de nuevo una ID al usuario si el formato no coincide.

Técnicas de limpieza.

Dado que ya se había realizado en el código anterior, no se requirió una nueva limpieza en esta actualización del archivo.

Estructura de datos.

Con el fin de organizar la información consultada, primero se creó un menú para el usuario, así, este podrá realizar una de 5 opciones, incluyendo buscar personajes, enemigos, ver la lista de enemigos/personajes o simplemente salir. Para realizar esta “lista de enemigos/personajes”, se creó una nueva estructura de los datos, siendo esta una lista de diccionarios; al inicio del código, solo se tomó cada personaje y enemigo como un diccionario de datos que incluía: nombre, juegos, etc. Ahora se almacena el personaje o enemigo solicitado a la api en una lista de diccionarios, cuya estructura consiste en dos listas vacías que sirven como variables globales en las cuales vamos almacenando los personajes y enemigos (diccionarios de datos con pares clave-valor) mediante un par de ciclos “for”; esto nos permite capturar datos de una manera más eficiente y ordenada.

Almacenamiento de datos.

Después de crear la estructura de datos actualizada, se pudo fácilmente crear un par de funciones propias del código que almacenan las listas de personajes y enemigos en dos archivos .txt que se crean en el dispositivo o, en caso de ya existir, se reescriben. Esto se logró gracias a funciones sencillas como “open” para crear el archivo y “write” para escribir en este.