

Desafío UTN Industries

Industrias UTN es principalmente una empresa de defensa que desarrolla y fabrica armas avanzadas y tecnologías militares.



Recientemente ha decidido ampliar su departamento de IT y para acceder a las entrevistas es necesario completar el siguiente desafío, el cual estará dividido en etapas. Cada semana recibiremos un nuevo pedido de parte de la empresa.



La empresa compartió con todos los participantes cierta información confidencial de un grupo de superhéroes y villanos. Y semana a semana enviará una lista con los nuevos requerimientos. Quien supere todas las etapas accederá a una entrevista con el director de la compañía.

Set de datos

La información a ser analizada se encuentra en el archivo data_utn.py (Crearla para que solo contenga la lista de personajes), por tratarse de una lista, bastará con incluir dicho archivo en el proyecto de la siguiente manera:

```
from data_utn import lista_personajes
```

Formato de los datos recibidos

```
lista_personajes = [  
    {  
        "id": 367,  
        "nombre": "John Constantine",  
        "identidad": "John Constantine",  
        "alias": "Hellblazer",  
        "empresa": "DC Comics",  
        "altura": 183.0,  
        "peso": 0.0,  
        "genero": "M",  
        "color_ojos": "Blue",  
        "color_pelo": "Blond",  
        "alineacion": "good",  
        "inteligencia": 63,  
        "fuerza": 10,  
        "poder": 54  
    },  
    {  
        "id": 686,  
        "nombre": "Vegeta",  
        "identidad": "Vegeta",  
        "alias": "Super Vegeta",  
        "empresa": "Shueisha",  
        "altura": 168.0,  
    }  
]
```

```
"peso": 73.0,  
"genero": "M",  
"color_ojos": "n/a",  
"color_pelo": "Black",  
"alineacion": "bad",  
"inteligencia": 81,  
"fuerza": 100,  
"poder": 100  
}
```

1

Desafío #02.5: (crear una biblioteca propia llamada: utn_biblioteca.py)

En base a lo practicado, deberás desarrollar esta pequeña App. Es por ello que te proponemos lo siguiente:

IMPORTANTE: *Para todas y cada una de las funciones creadas, documentarlas escribiendo que es lo que hacen, que son los parámetros que reciben (si es una lista, un string, etc y que contendrá) y que es lo que retorna la función!*

NOTA: Te ayudaremos con un paso a paso para la mayor parte de las funcionalidades de la app, en algunos casos con la experiencia obtenida deberás desarrollar las funciones de cero o modificar alguna función que hayas desarrollado previamente, guiandote en lo que hiciste y en otros casos podrás utilizar algunas funciones tal cual las desarrollaste para ahorrarte trabajo y así no repetir código. ¡Sos muy crack, se que vas a poder! 😎

- 1.1. Crear la función 'obtener_nombre' la cual recibirá por parámetro un diccionario el cual representara a un héroe y devolverá un string el cual contenga su nombre formateado de la siguiente manera:

Nombre: Rambo
- 1.2. Crear la función 'imprimir_dato' la cual recibirá por parámetro un string y deberá imprimirlo en la consola. La función no tendrá retorno.
- 1.3. Crear la función 'utn_imprimir_nombres_personajes' la cual recibirá por parámetro la lista de personajes y deberá imprimirla en la consola. Reutilizar las funciones hechas en los puntos 1.1 y 1.2. Validar que la lista no esté vacía para realizar sus acciones, caso contrario no hará nada y retornara -1.

[Usar esta función en el menú y probar su funcionamiento. Para cual/es opción/es del menú la usarias?]

2. Crear la función `'obtener_nombre_y_dato'` la misma recibirá por parámetro un diccionario el cual representara a un personaje y una key (string) la cual representará el dato que se desea obtener.
 - La función deberá devolver un string el cual contenga el nombre y dato (key) del personaje a imprimir. El dato puede ser 'altura', 'fuerza', 'peso' O CUALQUIER OTRO DATO.
 - El string resultante debe estar formateado de la siguiente manera: (suponiendo que la key es fuerza). Ejemplo:
Nombre: Venom | fuerza: 500
3. Crear la función `'utn_imprimir_nombres_alturas'` la cual recibirá por parámetro la lista de personajes, la cual deberá iterar e imprimir sus nombres y alturas USANDO la función creada en el punto 2. Validar que la lista de héroes no esté vacía para realizar sus acciones, caso contrario no hará nada y retornara -1.

[Usar esta función en el menú y probar su funcionamiento. Para cual/es opción/es del menú la usarias?]

- 4.1. Crear la función `'calcular_max'` la cual recibirá por parámetro la lista de personajes y una key (string) la cual representará el dato que deberá ser evaluado a efectos de determinar cuál es el máximo de la lista. Por ejemplo la función deberá poder calcular: el peso, la altura o fuerza máximas y retornar el personaje que tenga el dato más alto.

Ejemplo de llamada:

`calcular_max(lista, 'poder')` -> Retorna: El diccionario del personaje q cumpla

- 4.2. Crear la función `'calcular_min'` la cual recibirá por parámetro la lista de personajes y una key (string) la cual representará el dato que deberá ser evaluado a efectos de determinar cuál es el mínimo de la lista. Por ejemplo la

función deberá poder calcular: el peso, la altura o fuerza máximas y retornar el personaje que tenga el dato más bajo.

Ejemplo de llamada:

calcular_min(lista, 'poder') -> Retorna: El diccionario del personaje q cumpla

4.3. Crear la funcion '*calcular_max_min_dato*' la cual recibira tres parámetros:

- La lista de personajes
- El tipo de cálculo a realizar: es un valor string que puede tomar los valores '*maximo*' o '*minimo*'
- Un string que representa la key del dato a calcular, por ejemplo: '*altura*', '*peso*', '*inteligencia*', etc.

La función deberá retornar el personaje que cumpla con la condición pedida. Reutilizar las funciones hechas en los puntos 4.1 y 4.2

Ejemplo de llamada:

calcular_max_min_dato(lista, "maximo", "poder") -> Retorna: El diccionario del personaje q cumpla

4.4. Crear la función '*utn_calcular_imprimir_heroe*' la cual recibirá tres parámetros:

- La lista de personajes
- El tipo de cálculo a realizar: es un valor string que puede tomar los valores '*maximo*' o '*minimo*'
- Un string que representa la key del dato a calcular, por ejemplo: '*altura*', '*peso*', '*poder*', etc.

La función deberá obtener el personaje que cumpla dichas condiciones, imprimir su nombre y el valor calculado. Reutilizar las funciones de los puntos:

punto 1.2, punto: 2 y punto 4.3

Validar que la lista de personajes no esté vacía para realizar sus acciones, caso contrario no hará nada y retornara -1.

Ejemplo de llamada:

utn_calcular_imprimir_personaje (lista, "maximo", "poder")

Ejemplo de salida:

Mayor poder: Nombre: Godzilla | altura: 119000

[Usar esta función en el menú y probar su funcionamiento. Para cual/es opción/es del menú la usarias?]

- 5.1. Crear la función 'sumar_dato_personaje' la cual recibirá como parámetro una lista de personajes y un string que representara el dato/key de los personajes que se requiere sumar. Validar que cada personaje sea tipo diccionario y que no sea un diccionario vacío antes de hacer la suma. La función deberá retorna la suma de todos los datos según la key pasada por parámetro
- 5.2. Crear la función 'dividir' la cual recibirá como parámetro dos números (dividendo y divisor). Se debe verificar si el divisor es 0, en caso de serlo, retornar 0 (es matematicamente incorrecto, pero así lo pidió el cliente 🤖), caso contrario realizar la división entre los parámetros y retornar el resultado
- 5.3. Crear la función 'calcular_promedio' la cual recibirá como parámetro una lista de personajes y un string que representa el dato del personaje que se requiere calcular el promedio (ej: promedio de fuerza, altura, peso, etc). La función debe retornar el promedio (de entre todos los personajes) del dato pasado por parámetro

IMPORTANTE: hacer uso de las las funciones creadas en los puntos 5.1 y 5.2

- 5.4. Crear la función '**utn_calcular_imprimir_promedio_dato**' la cual recibirá una lista de personajes y un string que representara el dato/clave a calcular, utilizando la función del punto 5.3 calcula y mostrará la altura promedio.

Validar que la lista de personajes no esté vacía para realizar sus acciones, caso contrario no hará nada y retornara -1.

IMPORTANTE: hacer uso de las las funciones creadas en los puntos 1.2, 5.1 y 5.3

[Usar esta función en el menú y probar su funcionamiento. Para cual/es opción/es del menú la usarias?]

- 6.1. Crear la función "imprimir_menu" que imprima el menú de opciones por pantalla, el cual permite utilizar toda la funcionalidad ya programada. Se deberá reutilizar la función antes creada encargada de imprimir un string (1.2)
- 6.2. Crear la función "validar_entero" la cual recibirá por parámetro un string de número el cual deberá verificar que sea sea un string conformado únicamente por dígitos. Retornara True en caso de serlo, False caso contrario
- 6.3. Crear la función 'utn_menu_principal' la cual se encargará de imprimir el menú de opciones y le pedirá al usuario que ingrese el número de una de las opciones elegidas y deberá validarlo. En caso de ser correcto dicho número, lo retornara casteado a entero, caso contrario retorna -1. Reutilizar las funciones del ejercicio 6.1 y 6.2

[Usar esta función en el menú y probar su funcionamiento. Para cual/es opción/es del menú la usarias?]

Opciones del menú:

1. Recorrer la lista imprimiendo por consola el nombre de cada personaje
2. Recorrer la lista imprimiendo por consola nombre de cada personaje junto a la altura del mismo.
3. Recorrer la lista y determinar cuál es el personaje más alto (MÁXIMO).
4. Recorrer la lista y determinar cuál es el personaje más bajo (MÍNIMO).
5. Recorrer la lista y determinar la altura promedio de los personajes (PROMEDIO).
6. Informar cual es la identidad del personaje asociado a cada uno de los indicadores anteriores (Opcion 3 y 4).

7. Calcular e informar cual es el personaje más y menos pesado.
8. Calcular e informar cual es el personaje mas y menos poderoso.
9. Determinar el promedio de nivel de poder de todos los personajes e informar qué personaje/s están por debajo de ese poder.
10. Calcular e informar la cantidad total de personajes.
11. Calcular e informar cuántos personajes son de "DC Comics" y cuántos de "Marvel Comics"
12. Ordenar los personajes en orden descendente del que tiene mayor poder al que tiene menos, luego mostrarlos por consola con su nombre y poder.
13. Ordenar los personajes en orden ascendente del que tiene menos inteligencia al que tiene más inteligencia. Luego mostrarlos por consola su nombre e inteligencia.
14. Calcular e informar el promedio de poder de personajes femeninos.
15. Calcular e informar el promedio de poder de personajes masculinos.

7. Crear la función '**utn_personajes_app**' la cual recibirá por parámetro la lista de personajes y se encargará de la ejecución principal de nuestro programa. Utilizar if/elif o match según prefiera (match solo para los que cuentan con python 3.10+). Debe informar por consola en caso de seleccionar una opción incorrecta y volver a pedir el dato al usuario. Reutilizar las funciones con prefijo '**utn_**' donde crea correspondiente.

Esta es la función principal del programa, la cual dentro tendrá cada opción del menú