

## Desafio Stark

Industrias Stark 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 semana a semana enviará una lista con los nuevos requerimientos. Quien supere todas las etapas accederá a una entrevista con el director para de la compañía.

## Set de datos

La información a ser analizada se encuentra en el archivo `data_stark.py`, por tratarse de una lista, bastará con incluir dicho archivo en el proyecto de la siguiente manera:

```
from data_stark import lista_personajes
```

## Formato de los datos recibidos

```
lista_heroes =  
[  
    {  
        "nombre": "Howard the Duck",  
        "identidad": "Howard (Last name unrevealed)",  
        "empresa": "Marvel Comics",  
        "altura": "79.349999999999994",  
        "peso": "18.449999999999999",  
        "genero": "M",  
        "color_ojos": "Brown",  
        "color_pelo": "Yellow",  
        "fuerza": "2",  
        "inteligencia": "average"  
    },  
    {  
        "nombre": "Rocket Raccoon",  
        "identidad": "Rocket Raccoon",  
        "empresa": "Marvel Comics",  
        "altura": "122.77",  
        "peso": "25.73",  
        "genero": "M",  
        "color_ojos": "Brown",  
        "color_pelo": "Brown",  
        "fuerza": "5",  
        "inteligencia": "average"  
    }  
]
```

#### Desafío #04:

**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!*

1.1. Crear la función '**extraer\_iniciales**' que recibirá como parámetro:

- *nombre\_heroe*: un string con el nombre del personaje

La función deberá devolver a partir del parámetro recibido un nuevo string con las iniciales del nombre del personaje seguidos por un punto (.)

- En el caso que el nombre del personaje contenga el artículo '*the*' se deberá omitir de las iniciales
- Se deberá verificar si el nombre contiene un guión '-' y sólo en el caso que lo tenga se deberá reemplazar por un espacio en blanco

La función deberá validar:

- Que el string recibido no se encuentre vacío

Devolver '*N/A*' en caso de no cumplirse la validación

Ejemplo de la salida de la función para Howard the Duck:

***H.D.***

1.2. Crear la función '**definir\_iniciales\_nombre**' la cual recibirá como parámetro:

- *heroe*: un diccionario con los datos del personaje

La función deberá agregar una nueva clave al diccionario recibido como parámetro. La clave se deberá llamar *'iniciales'* y su valor será el obtenido de llamar a la función *'extraer\_iniciales'*

La función deberá validar:

- Que el dato recibido sea del tipo diccionario
- Que el diccionario contengan la clave *'nombre'*

En caso de encontrar algún error retornar False, caso contrario retornar True

1.3. Crear la función ***'agregar\_iniciales\_nombre'*** la cual recibirá como parámetro:

- *lista\_heroes*: lista de personajes

Se deberá validar:

- Que *lista\_heroes* sea del tipo lista
- Que la lista contenga al menos un elemento

La función deberá iterar la *lista\_heroes* pasándole cada héroe a la función *definir\_iniciales\_nombre*.

En caso que la función *definir\_iniciales\_nombre()* retorne False entonces se deberá detener la iteración e informar por pantalla el siguiente mensaje:

*'El origen de datos no contiene el formato correcto'*

La función deberá devolver True en caso de haber finalizado con éxito o False en caso de que haya ocurrido un error

1.3. Crear la función '**stark\_imprimir\_nombres\_con\_iniciales**' la cual recibirá como parámetro:

- *lista\_heroes*: la lista de personajes

La función deberá utilizar la función *agregar\_iniciales\_nombre* para añadirle las iniciales a los diccionarios contenidos en la *lista\_heroes*

Luego deberá imprimir la lista completa de los nombres de los personajes seguido de las iniciales encerradas entre paréntesis ()

Se deberá validar:

- Que *lista\_heroes* sea del tipo lista
- Que la lista contenga al menos un elemento

Delante de cada nombre se deberá agregar un asterisco '\*' (de forma de viñeta) seguido de un espacio.

Ejemplo de salida:

**\* Howard the Duck (H.D.)**

**\* Rocket Raccoon (R.R.)**

...

La función no retorna nada

---

2.1. Crear la función '**generar\_codigo\_heroe**' la cual recibirá como parámetros:

- *id\_heroe*: un entero que representa el identificador del héroe.
  - NOTA: el valor de *id\_heroe* lo vamos a generar recién el punto 2.3. Para probar la función podes pasarle cualquier entero
- *genero\_heroe*: un string que representa el género del héroe ( puede tomar los valores 'M', 'F' o 'NB')

La función deberá generar un string con el siguiente formato:

***GENERO-000...000ID***

Es decir, el género recibido por parámetro seguido de un '-' (guión) y por último el identificador recibido. Todos los códigos generados deben tener como máximo 10 caracteres (contando todos los caracteres, inclusive el guión). Se deberá completar con ceros para que todos queden del mismo largo

Algunos ejemplos:

**F-00000001**

**M-00000002**

**NB-0000010**

La función deberá validar:

- El identificador del héroe sea numérico.
- El género no se encuentre vacío y este dentro de los valores esperados ('M', 'F' o 'NB')

En caso de no pasar las validaciones retornar 'N/A'. En caso de verificarse correctamente retornar el código generado

2.2. Crear la función '**agregar\_codigo\_heroe**' la cual recibirá como parámetro:

- *heroe*: un diccionario con los datos del personaje
- *id\_heroe*: un entero que representa el identificador del héroe.
  - NOTA: el valor de *id\_heroe* lo vamos a generar recién el punto

2.3. Para probar la función podes pasarle cualquier entero

La función deberá agregar una nueva clave llamada '*codigo\_heroe*' al diccionario '*heroe*' recibido como parámetro y asignarle como valor un código utilizando la función '*generar\_codigo\_heroe*'

La función deberá validar:

- Que el diccionario recibido como parámetro no se encuentre vacío.
- Que el código recibido mediante *generar\_codigo\_heroe* tenga exactamente 10 caracteres

En caso de pasar las validaciones correctamente la función deberá retornar True, en caso de encontrarse un error retornar False

2.3. Crear la función '***stark\_generar\_codigos\_heroes***' la cual recibirá como parámetro:

- *lista\_heroes*: la lista de personajes

La función deberá iterar la lista de personajes y agregarle el código a cada uno de los personajes.

El código del héroe (*id\_heroe*) surge de la posición del mismo dentro de la *lista\_heroes* (comenzando por el 1).

Reutilizar la función *agregar\_codigo\_heroe* pasándole como argumentos el héroe que se está iterando y el *id\_heroe*

Una vez finalizado imprimir por pantalla un mensaje como el siguiente:  
(## representa la cantidad de códigos generados):

**Se asignaron ## códigos**

\* El código del primer héroe es:           M-00000001

\* El código del del último héroe es: M-00001224

La función deberá validar::

- La lista contenga al menos un elemento
- Todos los elementos de la lista sean del tipo diccionario
- Todos los elementos contengan la clave 'genero'

En caso de encontrar algún error, informar por pantalla: *'El origen de datos no contiene el formato correcto'*

La función no retorna ningún valor.

---

3.1. Crear la función '**sanitizar\_entero**' la cual recibirá como parámetro:

- numero\_str: un string que representa un posible número entero

La función deberá analizar el string recibido y determinar si es un número entero positivo. La función debe devolver distintos valores según el problema encontrado:

- Si contiene caracteres no numéricos retornar -1
- Si el número es negativo se deberá retornar un -2
- Si ocurren otros errores que no permiten convertirlo a entero entonces se deberá retornar -3

También deberá quitar los espacios en blanco de atrás y adelante del string en caso que los tuviese

En caso que se verifique que el texto contenido en el string es un número entero positivo, retornarlo convertido en entero



3.2. Crear la función '**sanitizar\_flotante**' la cual recibirá como parámetro:

- *numero\_str*: un string que representa un posible número decimal

La función deberá analizar el string recibido y determinar si es un número flotante positivo. La función debe devolver distintos valores según el problema encontrado:

- Si contiene caracteres no numéricos retornar -1
- Si el número es negativo se deberá retornar un -2
- Si ocurren otros errores que no permiten convertirlo a entero entonces se deberá retornar -3

También deberá quitar los espacios en blanco de atrás y adelante del string en caso que los tuviese

En caso que se verifique que el texto contenido en el string es un número flotante positivo, retornarlo convertido en flotante

3.3. Crear la función '**sanitizar\_string**' la cual recibirá como parámetro

- *valor\_str*: un string que representa el texto a validar
- *valor\_por\_defecto*: un string que representa un valor por defecto (parámetro opcional, inicializarlo con '-')

La función deberá analizar el string recibido y determinar si es solo texto (sin números). En caso de encontrarse números retornar "N/A"

En el caso que *valor\_str* contenga una barra '/' deberá ser reemplazada por un espacio

El espacio es un caracter válido

En caso que se verifique que el parámetro recibido es solo texto, se deberá retornar el mismo convertido todo a minúsculas

En el caso que el texto a validar se encuentre vacío y que nos hayan pasado un valor por defecto, entonces retornar el valor por defecto convertido a minúsculas

Quitar los espacios en blanco de atras y adelante de ambos parámetros en caso que los tuviese

3.4. Crear la función '**sanitizar\_dato**' la cual recibirá como parámetros:

- *heroe*: un diccionario con los datos del personaje
- *clave*: un string que representa el dato a sanitizar (la clave del diccionario). Por ejemplo *altura*
- *tipo\_dato*: un string que representa el tipo de dato a sanitizar. Puede tomar los valores: '*string*', '*entero*' y '*flotante*'

La función deberá sanitizar el valor del diccionario correspondiente a la clave y al tipo de dato recibido

Para sanitizar los valores se deberán utilizar las funciones creadas en los puntos 3.1, 3.2, 3.3 y 3.4

Se deberá validar:

- Que *tipo\_dato* se encuentre entre los valores esperados ('*string*', '*entero*', '*flotante*') la validación debe soportar que nos lleguen mayúsculas o minúsculas. En caso de encontrarse un valor no válido informar por pantalla: '*Tipo de dato no reconocido*'

- Que *clave* exista como clave dentro del diccionario *heroe*. En caso de no encontrarse, informar por pantalla: 'La clave especificada no existe en el héroe'. (en este caso la validación es sensible a mayúsculas o minúsculas)

Ejemplo de llamada a la función válida:

***sanitizar\_dato(dict\_personaje, "altura", "Flotante")***

La función deberá devolver True en caso de haber sanitizado algún dato y False en caso contrario.

3.5. Crear la función '***stark\_normalizar\_datos***' la cual recibirá como parámetros:

- *lista\_heroes*: la listas personajes

La función deberá recorrer la lista de héroes y sanitizar los valores solo de las siguientes claves: '*altura*', '*peso*', '*color\_ojos*', '*color\_pelo*', '*fuerza*' e '*inteligencia*'

- Un vez finalizado el proceso mostrar el mensaje 'Datos normalizados',
- Validar que la lista de héroes no esté vacía para realizar sus acciones, caso contrario imprimirá el mensaje: "Error: Lista de héroes vacía"
- La función no retorna nada
- Reutilizar la función *sanitizar\_dato*

---

4.1. Crear la función '***generar\_indice\_nombres***' la cual recibirá como parámetro:

- *lista\_heroes*: la lista de personajes

La función deberá iterar la lista de personajes y generar una lista donde cada elemento es cada palabra que componen el nombre de los personajes.

Por ejemplo la lista que se deberá retornar tiene la siguiente forma:

***['Howard', 'the', 'Duck', 'Rocket', 'Raccoon', 'Wolverine', ... ]***

La función deberá validar que:

- La lista contenga al menos un elemento
- Todos los elementos de lista\_heroes sean del tipo diccionario
- Todos los elementos contengan la clave *'nombre'*

En caso de encontrar algún error, informar por pantalla: *'El origen de datos no contiene el formato correcto'*

4.2. Crear la función ***'stark\_imprimir\_indice\_nombre'*** la cual recibirá como parámetro:

- lista\_heroes: la lista de personajes

La función deberá mostrar por pantalla el índice generado por la función *generar\_indice\_nombres* con todos los nombres separados con un guión.

Por ejemplo:

***Howard-the-duck-Rocket-Raccoon-Wolverine...***

---

5.1. Crear la función ***'convertir\_cm\_a\_mtrs'*** la cual recibirá como parámetro:

- *valor\_cm*: Un número que representa una medida en centímetros

La función deberá retornar el número recibido, pero convertido a la unidad metros. La función deberá validar que el número recibido sea un número flotante positivo, en caso de no serlo retornar -1

5.2. Crear la función '**generar\_separador**' la cual recibirá como parámetro

- *patron*: un carácter que se utilizará como patrón para generar el separador
- *largo*: un número que representa la cantidad de caracteres que va ocupar el separador.
- *imprimir*: un parámetro opcional del tipo booleano (por default definir en True)

La función deberá generar un string que contenga el patrón especificado repitiendo tantas veces como la cantidad recibida como parámetro (uno junto al otro, sin salto de línea)

Si el parámetro booleano recibido se encuentra en False se deberá solo retornar el separador generado. Si se encuentra en True antes de retornarlo, imprimirlo por pantalla

La función deberá validar:

- Que el parámetro *patrón* tenga al menos un carácter y como máximo dos
- Que el parámetro *largo* sea un entero entre 1 y 235 inclusive

En caso de no verificarse las validaciones devolver 'N/A'

Ejemplo de llamada:

***generar\_separador('\*', 10)***

Ejemplo de salida:

\*\*\*\*\*

5.3. Crear la función '**generar\_encabezado**' la cual recibirá como parámetro

- titulo: un string que representa el título de una sección de la ficha

La función deberá devolver un string que contenga el título envuelto entre dos separadores (estimar el largo requerido para tu pantalla).

Ejemplo de salida:

```
*****
PRINCIPAL
*****
```

La función deberá convertir el titulo recibido en todas letras mayúsculas

#### 5.4. Crear la función '*imprimir\_ficha\_heroe*' la cual recibirá como parámetro:

- *heroe*: un diccionario con los datos del héroe

La función deberá a partir los datos del héroe generar un string con el siguiente formato e imprimirlo por pantalla::

```
*****
PRINCIPAL
*****

NOMBRE DEL HÉROE:           Spider-Man (S.M.)

IDENTIDAD SECRETA:          Peter Parker

CONSULTORA:                  Marvel Comics

CÓDIGO DE HÉROE :           M-00000001

*****
FISICO
*****
```

ALTURA: 1,78 Mtrs.

PESO: 74,25 Kg.

FUERZA: 55 N

\*\*\*\*\*

### SEÑAS PARTICULARES

\*\*\*\*\*

COLOR DE OJOS: Hazel

COLOR DE PELO: Brown

5.5. Crear la función '**stark\_navegar\_fichas**' la cual recibirá como parámetros:

- *lista\_heroes*: la listas personajes

La función deberá comenzar imprimiendo la ficha del primer personaje de la lista y luego solicitar al usuario que ingrese alguna de las siguientes opciones:

[ 1 ] Ir a la izquierda      [ 2 ] Ir a la derecha      [ S ] Salir

- Si el usuario ingresa '1': se debe mostrar el héroe que se encuentra en la posición anterior en la lista (en caso de estar en el primero, ir al último)
- Si el usuario ingresa '2': se debe mostrar el héroe que se encuentra en la posición siguiente en la lista (en caso de estar en el último, ir al primero)
- Si ingresa 'S': volver al menú principal

- Si ingresa cualquier otro valor, volver a mostrar las opciones hasta que ingrese un valor válido

6.1. Crear la función *'imprimir\_menu'* que imprima las siguientes opciones por pantalla:

"""

1 - Imprimir la lista de nombres junto con sus iniciales

2 - Generar códigos de héroes

3 - Normalizar datos

4 - Imprimir índice de nombres

5 - Navegar fichas

S - Salir

---

"""

6.2. Crear la función *'stark\_menu\_principal'*. No recibe parámetros.

La función deberá imprimir el menú de opciones y le pedirá al usuario que ingrese una.

La función deberá retornar la respuesta del usuario

6.3. Crear la función *'stark\_marvel\_app\_3'* la cual recibirá como parámetro:

- *lista\_heroes*: la lista de personajes

La función 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 'stark\_' donde crea correspondiente.