



RETO 2 - FUNDAMENTOS DE PROGRAMACIÓN

La clínica veterinaria “San Francisco de Asís” ofrece atención en salud a las siguientes especies: Caninos, felinos, equinos, bovinos y roedores. El grupo de personas que está al frente de la administración de la clínica desea implementar un plan de acompañamiento especial en salud para aquellos animales que se clasifiquen como “adultos mayores” siempre y cuando los pacientes sean del tipo canino, felino, equino o bovino.

En la tabla a continuación se ilustra la guía que debe seguirse para clasificar como “adultos mayores” a los pacientes veterinarios, según su edad:

Tipo	Edad mínima en años
Canino o felino	9
Equino o bovino	16

La clínica cuenta con datos de pacientes veterinarios, distribuidos así:

- Una lista con los nombres de los pacientes, llamada “nombres”
- Una lista llamada “tipos” con los tipos de pacientes (canino, felino, equino, bovino, roedor) ordenada de acuerdo con los animales presentes en la lista de nombres
- Una lista llamada “edades” con las respectivas edades de los pacientes
- Una lista llamada “pesos” en las que están almacenados los pesos de cada uno de los pacientes, correspondiendo a los animales en la lista de los nombres

Usted ha sido delegado para:

- El reordenamiento de los datos en un primer diccionario, de tal manera que los datos no estén separados en listas distintas sino unidos por parejas clave:valor, donde la “clave” será un índice de tipo string (el





número 1 en string para el primer valor, el número 2 en string para el segundo valor, el número 3 en string para el tercer valor y así sucesivamente) y donde los valores serán listas que contengan la información consolidada de cada paciente veterinario en este orden: Su nombre, su tipo, su edad y su peso.

- La creación de un segundo diccionario donde las “claves” también serán índices de tipo string (el número 1 en string para el primer valor, el número 2 en string para el segundo valor, el número 3 en string para el tercer valor y así sucesivamente) y donde los valores serán listas que contengan los siguientes datos de los pacientes veterinarios caninos/felinos que CUMPLAN LA EDAD PARA BENEFICIARSE CON EL PLAN DE ACOMPAÑAMIENTO en este orden: Su nombre, su tipo y su peso.
- La creación de un tercer diccionario donde las “claves” también serán índices de tipo string (el número 1 en string para el primer valor, el número 2 en string para el segundo valor, el número 3 en string para el tercer valor y así sucesivamente) y donde los valores serán listas que contengan los siguientes datos de los pacientes veterinarios equinos/bovinos que CUMPLAN LA EDAD PARA BENEFICIARSE CON EL PLAN DE ACOMPAÑAMIENTO en este orden: Su nombre, su tipo y su peso.
- El cálculo del promedio de edad de los pacientes caninos/felinos que resulten elegibles para beneficiarse con el plan de acompañamiento.
- El cálculo del promedio de edad de los pacientes equinos/bovinos que resulten elegibles para beneficiarse con el plan de acompañamiento.

A modo ilustrativo, suponga la siguiente situación: La clínica veterinaria “San Francisco de Asís” le ha entregado a usted la siguiente información en listas de Python para procesarla tal como se indicó:

Nombres: [“Martín”, “Milú”, “Anastasia”, “Lupita”, “Tomas”, “Pelusa”, “Genoveva”, “Motita”]

Tipos: [“canino”, “canino”, “felino”, “felino”, “felino”, “canino”, “bovino”, “roedor”]

Edades: [12, 9, 10, 8, 9, 2, 14, 1]

Pesos: [33, 26, 4, 5, 5, 6, 106.4, 0.34]

Esta información debe entenderse así: En el caso de “Pelusa” (índice 5 de la lista de Nombres) se trata de un “canino” (índice 5 de la lista de Tipos) con una edad





de 2 años (índice 5 de la lista de Edades) y con un peso de 6 kilogramos (índice 5 de la lista de Pesos). Nótese que los pesos pueden ser valores reales (con decimales como el caso de “Genoveva” y “Motita”) mas las edades pueden ser enteros redondos o reales; los nombres y los tipos son datos de tipo string. Lo que quiere la veterinaria es que se reorganice dicha información y se usarán diccionarios de Python. Un fragmento del primer diccionario a crear sería así:

```
{“1” : [Martín, “canino”, 12, 33] , “2” : [“Milú”, “canino”, 9, 26] , “3” : [“Anastasia” ,
“felino” , 10, 4] , “4” : [“Lupita” , “felino” , 8 , 5]... }
```

Observe que las claves de este diccionario son números en tipo string. En el segundo diccionario a crear deben estar aquellos pacientes elegibles al plan de acompañamiento dado su rango de edad (caninos, felinos, equinos, bovinos) El segundo diccionario a crear sería así:

```
{“1” : [“Martín” , “canino” , 33] , “2” : [“Milú” , “canino” , 26] , “3”: [“Anastasia” ,
“felino” , 4] , “4” : [“Tomasas” , “felino” , 5]}
```

Nótese que en este segundo diccionario las claves también son strings de números y que sólo hay pacientes caninos/felinos con edades mayores o iguales a 9 años, por lo que ni “Lupita” ni “Pelusa” son calificables para el programa. El tercer diccionario a crear sería:

```
{}
```

Porque en este diccionario estarán contenidos los nombres de aquellos equinos/bovinos cuya edad los haga elegibles al programa que adelanta la veterinaria y el único animal de ese tipo es “Genoveva” cuya edad (14 años) no la hace elegible de acuerdo a la tabla en el inicio de este enunciado. A continuación, se debe entregar el cálculo de los promedios de edad de los pacientes elegibles para el programa en las dos categorías: canino/felino y equino/bovino. Los resultados a entregar serían:

10

None

El primer resultado es el promedio de las edades de “Martín, “Milú”, “Anastasia” y “Tomasas” (pacientes canino/felino elegibles para el programa) y el segundo resultado es de tipo None pues no hay pacientes equino/bovino elegibles para el programa de acompañamiento de la clínica veterinaria.





TAREAS

Realizar un programa en Python que le permita a la administración de la clínica veterinaria “San Francisco de Asís” tener en otro formato la información con la que inicialmente cuentan para poder identificar a los pacientes veterinarios que sean elegibles para su plan de acompañamiento. La función, escrita en Python, estará contenida en un archivo de nombre `solution.py` e internamente estará denotada así:

```
veterinaria(nombres, tipos, edades, pesos)
```

PARA LA IMPLEMENTACIÓN DE LA SOLUCIÓN NO DEBE USAR EL INGRESO DE INFORMACIÓN POR CONSOLA. Debe crear su programa de tal forma que toda la información que se usará para validar su solución (listas de nombres, de tipos, de edades y de pesos) se proporcione a través de los parámetros de entrada (nombres, tipos, edades, pesos) y de tal manera que el programa retorne tres diccionarios y dos cálculos de promedios.

NO OLVIDAR QUE LAS CLAVES EN LOS DICCIONARIOS SERÁN NÚMEROS DE TIPO STRING ORDENADOS DESDE EL 1. Si utiliza números de tipo entero o de tipo real, el calificador no podrá procesar su solución.

Controle la operación del algoritmo para algún escenario donde ningún paciente en las listas tenga la edad para ser elegible al plan de acompañamiento y por ende no se vaya a calcular ningún promedio; es decir, busque que el cálculo de promedios de edad ocurra cuando haya en las listas animales adultos mayores y que cuando no haya un cálculo de promedio de edad, la variable del promedio retorne un valor “None”

NOTA ACLARATORIA

Se recomienda desarrollar la prueba en un IDE como G Colab, VSCode, PyCharm, Spyder, etc. Al final debe copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:





Modo incorrecto:

Examen caracterización-estudiantes

NO SUBIR NINGÚN ARCHIVO

Descripción Entrega **Editar** Ver entrega

Entrega

Comentarios

Seleccione un archivo... Tamaño máximo para archivos nuevos: 5MB

solucion.py

Puede arrastrar y soltar archivos aquí para añadirlos

Enviar Cancelar

Modo correcto:

Área personal - Mis cursos - Examen caracterización-estudiantes - Reto de clasificación

Examen caracterización-estudiantes

Descripción Entrega **Editar** Ver entrega

LUGAR CORRECTO

solucion.py

```
1 NO ELIMINAR LAS SIGUIENTES IMPORTACIONES, sirven para probar tu código en consola, y el funcionamiento de la librería csv respectivamente
2 from test import tester
3 import csv
4
5 """NOTAS:
6 - PARA ESTE RETO PUEDES PROBAR TU PROGRAMA, DANDO CLICK EN LA NAVE ESPACIAL
7 - LA CONSOLA TE DIRÁ SI TU SOLUCIÓN ES CORRECTA O NO
8 - NO olvidar evaluar tu solución
9 """
10
11
12 """Inicio espacio para programar funciones propias"""
13 #En este espacio podrás programar las funciones que deseas usar en la función solución (ES OPCIONAL)
14
15
16
```

TRIPULANTE, ¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 2!