

""" Se requiere un sistema para registrar los turnos de un día para una veterinaria. La misma atiende de 9 a 19hs, los turnos son uno por hora.

Se debe contar con un menú tal:

1) Registrar turno: Se debe ingresar el nombre de la mascota, la edad, especie (Gato o Perro) y el horario. No dar dos turnos a la misma mascota. La carga Finaliza con mascota "FIN" (Se puede seguir cargando turnos)

2) Prioridad: Insertar a continuación de los turnos de las mascotas de más de 10 años, un 99 en turno y edad, y "PRIORIDAD" en nombre.

3) Ver turnos: Se muestra con formato adecuado y ordenados, los datos de cada turno (horario, mascota). Si la mascota tiene prioridad, mostrarlo.

4) Eliminar turno: Permite eliminar el turno de una mascota.

5) Ver estadísticas:

- * Mascota más joven (mostrar especie, nombre y edad)

- * Promedio de edad de cada especie.

- * Porcentaje de mascotas que superan el promedio de edad, sobre el total de cada especie.

6) Salir """

FUNCIONES

```
def seleccionar_opcion():
```

```
    ("Menu:")
```

```
    ("1) Registrar turno")
```

```
    ("2) Prioridad")
```

```
    ("3) Ver turnos")
```

```
    ("4) Eliminar turno")
```

```
    ("5) Ver estadísticas")
```

```
    ("6) Salir")
```

```
    opcion = int(input("Ingrese la opción del menú: "))
```

```
    while opcion < 1 or opcion > 6:
```

```
        opcion = int(input("Error... Ingrese la opción del menú: "))
```

```
    return opcion
```

```
def ingresar_nombre():
```

```
    mascota = input("Ingrese el nombre de la mascota: ").upper( )
```

```
    while mascota == "":
```

```
        mascota = input("Error... Ingrese el nombre de la mascota: "). upper( )
```

```
    return mascota
```

```
def ingresar_especie():
```

```
    e = input("Ingrese el tipo de mascota (P o G): ").upper( )
```

```
    while e != "P" and e != "G":
```

```
        e = input("Error... Ingrese el tipo de mascota (P o G): ").upper()
```

```
    return e
```

```
def ingresar_edad():
```

```
    edad = int(input("Ingrese el edad: "))
```

```
    while edad < 0 or edad > 20:
```

```
        edad = int(input("Error... Ingrese la edad: "))
```

```
    return edad
```

```
def ingresar_horario():
```

```
    horario = int(input("Ingrese el horario: "))
```

```
    while horario < 9 or horario > 19:
```

```
        horario = int(input("Error .. Ingrese el horario: "))
```

```
    return horario
```

```

def buscar_mascota(arr_nombres, nombre):

    i = 0
    while i < len(arr_nombres) and nombre != arr_nombres[i]:
        i += 1
    return i

def registrar_turnos(arr_nombres, arr_edades, arr_especie, arr_horario):
    nombre = ingresar_nombre()
    while nombre != "FIN":
        pos = buscar_mascota(nombres, nombre)
        if pos == len(nombres):
            especie = ingresar_especie()
            edad = ingresar_edad()
            horario = ingresar_horario()

            arr_nombres.append(nombre)
            arr_especie.append(especie)
            arr_edades.append(edad)
            arr_horario.append(horario)
        else:
            print(f"La mascota ya tiene turno hoy a las {arr_horario[pos]}")
            nombre = ingresar_nombre()

def registrar_prioridad(arr_nombres, arr_edades, arr_especie, arr_horario):
    """Inserta turnos de prioridad después de cada mascota mayor de 10 años."""
    i = 0
    print("Buscando mascotas mayores de 10 años para asignar prioridad...")

    while i < len(arr_nombres):
        es_prioridad = 0
        if arr_nombres[i] == "PRIORIDAD":
            es_prioridad = 1
        if es_prioridad == 0 and arr_edades[i] > 10:
            # Insertar los datos de prioridad *a continuación* (índice i+1)
            arr_nombres.insert(i + 1, "PRIORIDAD")
            arr_edades.insert(i + 1, 99)
            arr_horario.insert(i + 1, arr_horario[i])
            arr_especie.insert(i + 1, arr_especie[i])
            i += 1
            print(f"-> Prioridad asignada después de {arr_nombres[i-1]}")
        i += 1
    print("Asignación de prioridad finalizada.")

def mostrar_arreglos (arr_nombres, arr_edades, arr_especie, arr_horario):
    for i in range(len(arr_nombres)):
        if i+1 < len(arr_nombres) and arr_nombres[i+1] == "PRIORIDAD":
            print(f"{arr_horario[i]}: {arr_nombres[i]} - PRIORIDAD")

        elif arr_nombres[i] != "PRIORIDAD":
            print(f"{arr_horario[i]}: {arr_nombres[i]}")

def eliminar_turno(arr_nombres, arr_edades, arr_especie, arr_horario):
    nombre = ingresar_nombre()
    pos = buscar_mascota(arr_nombres, nombre)
    if pos != len(arr_nombres):
        arr_nombres.pop(pos)
        arr_edades.pop(pos)
        arr_especie.pop(pos)
        arr_horario.pop(pos)
    else:

```

```

        print("La mascota no tiene turno asignado. ")

def buscar_joven(arr_edades):
    pos_min = 0
    for i in range(len(arr_edades)):
        if arr_edades[i] < arr_edades[pos_min]:
            pos_min = i
    return pos_min

def calcular_promedio(arr_especie, arr_edades, tipo):
    acum = 0
    cont = 0
    promedio = -1
    for i in range(len(arr_edades)):
        if arr_especie[i] == tipo:
            acum += arr_edades[i]
            cont += 1

    if cont != 0:
        promedio = acum/cont

    return promedio

def calcular_porcentaje(arr_especie, arr_edades, tipo, promedio):
    cont_total = 0
    cont_supera = 0

    for i in range(len(arr_edades)):
        if arr_especie[i] == tipo:
            if arr_edades[i] > promedio:
                cont_supera += 1

        cont_total += 1
    porcentaje = cont_supera/cont_total * 100

    return porcentaje

def ver_estadisticas(arr_nombres, arr_edades, arr_especie, arr_horario):
    """ * Mascota más joven (mostrar especie, nombre y edad)
    * Promedio de edad de cada especie.
    * Porcentaje de mascotas que superan el promedio de edad, sobre el total de cada
    especie. """
    pos_mas_joven = buscar_joven(edades)
    promedio_gato = calcular_promedio(arr_especie, arr_edades, "G")
    promedio_perro = calcular_promedio(arr_especie, arr_edades, "P")

    print(f"La mascota mas joven es un {arr_especie[pos_mas_joven]} llamado
{arr_nombres[pos_mas_joven]} y tiene {arr_edades
[pos_mas_joven]} años")

    if promedio_gato != -1:
        print("El promedio de edades de los gatos es: ", promedio_gato)
        porcentaje = calcular_porcentaje(arr_especie, arr_edades, "G", promedio_gato)
        print("El porcentaje que supera dicha edad es:", porcentaje)
    if promedio_perro != -1:
        print("El promedio de edades de los perros es: ", promedio_perro)
        porcentaje = calcular_porcentaje(arr_especie, arr_edades, "P", promedio_perro)
        print("El porcentaje que supera dicha edad es:", porcentaje)

def ordenar_arreglos(arr_nombres, arr_edades, arr_especie, arr_horario):
    for i in range(len(arr_horario)):

```

```

        for j in range(len(arr_horario)):
            if arr_horario[i] < arr_horario[j]:
                intercambiar(arr_horario, i, j)
                intercambiar(arr_nombres, i, j)
                intercambiar(arr_edades, i, j)
                intercambiar(arr_especie, i, j)

def intercambiar(arreglo, i, j):
    aux = arreglo[i]
    arreglo[i] = arreglo[j]
    arreglo[j] = aux

# PROGRAMA PRINCIPAL
#Variables

nombres = []
edades = []
especie = []
horario = []

opcion = seleccionar_opcion()
while opcion != 6:

    if opcion == 1:
        registrar_turnos(nombres, edades, especie, horario)
    if opcion == 2:
        registrar_prioridad(nombres, edades, especie, horario)
    if opcion == 3:
        mostrar_arreglos(nombres, edades, especie, horario)
    if opcion == 4:
        eliminar_turno(nombres, edades, especie, horario)
    if opcion == 5:
        ver_estadisticas(nombres, edades, especie, horario)
    opcion = seleccionar_opcion()
print("Fin del programa.")

```