



Analizando Mortalidad de Aves (SECCION 2)

El servicio agrícola y ganadero (SAG) es el organismo oficial del Estado de Chile, encargado de apoyar el desarrollo de la agricultura, los bosques y la ganadería, a través de la protección y mejoramiento de la salud de los animales y vegetales.

La División de Protección Pecuaria es responsable de la protección, mantención e incremento del patrimonio zoonosanitario nacional, así como de la certificación zoonosanitaria y de aptitud de consumo humano de los productos pecuarios primarios de exportación.

La sanidad animal del país se aborda tanto desde la prevención del ingreso de enfermedades (evitando así la introducción al territorio de agentes causantes de enfermedades exóticas o endémicas de importancia económica), como mediante el control y erradicación de enfermedades presentes en el país. Ambas acciones son apoyadas por el Sistema de Vigilancia, el cual permite detectar precozmente el ingreso de enfermedades exóticas al país.

De esta forma, el SAG aborda la sanidad animal como pilar fundamental para asegurar la competitividad del sector pecuario y entregar las garantías necesarias a los mercados de destino de las exportaciones de productos animales.

Como sistema de vigilancia se analizan los casos de mortalidad de aves silvestres y se registran en archivos de texto para su posterior discusión.

Objetivo

En este proyecto usted debe escribir un programa en Python 3, que dé respuesta a lo siguiente:

- a. Que indique la cantidad de aves muertas por región
- b. Que indique la cantidad de aves muertas para el mes de enero del año 2023.
- c. Que indique la cantidad de muertes detectadas de la especie “Tagua” en la comuna de Cartagena.
- d. Que indique la cantidad de muertes detectadas de la especie “Piquero” reportada para el día 12 de febrero del año 2023.
- e. Que grafique la cantidad de muertes de las siguientes especies: Gaviota garuma, Piquero, Gaviota de Franklin, Pelicano y Guanay, considerando un gráfico de barras.

Entrada

Los datos se encuentran en un archivo llamado “mortalidad_aves.txt”, que se encuentra disponible para su descarga en la plataforma UCMVirtual del curso. Cada línea contiene la siguiente información: fecha del informe, nombre de la estación, región, comuna, lugar, sitio, especie y total de muertes. Cada dato en la línea viene separado por una coma. Como sugerencia cree un lista con el nombre de las regiones contenidas en el archivo (no deben repetirse). Esto mismo debe considerarlo como ayuda para determinar el nombre de las comunas por una región específica.

Salida

La salida del programa deberá quedar registrada en un archivo de texto llamado “resultadoS2.txt”, con el siguiente formato:

```
Autor(es): nnnnnnn - nnnnnnn  
  
Cantidad de aves muertas por región:  
  
    Nombre de la región: xxxx  
  
    Nombre de la región: xxxx  
  
    ...  
  
Casos aves muertas mes enero del año 2023: xxxx  
  
En la comuna de <nombre comuna> se detectaron xxxx Taguas  
muertas.  
  
Las muertes detectadas para el 12 de febrero del 2023 de la  
especie Gaviota garuma son: xxxx
```

Mostrar el grafico solicitado.

Rúbrica de Evaluación

	1	3	5	7	Total
Revisión del Código. (20%)	Código desordenado, sin uso de las normas de programación, no utiliza comentarios, nombres de variables inadecuados.	Código Ordenado, usa indentación. Usa identificadores de variables adecuado. Incorpora comentarios mínimos o existe sobresaturación de comentarios.	Código ordenado. Uso adecuado de variables (listas, string, etc.). Uso adecuado de estructuras algorítmicas Uso eficiente de los comentarios.	Cumple con todas las normas de programación.	
Solución Computacional. (40%)	No se ejecuta. Errores de Ejecución. Programa cae en Loop.	Ejecuta pero con errores de lógica. Ejecuta pero no llega a la solución.	Llega a una solución parcial. Tarda más de un minuto en llegar a la solución. No cumple con el formato de salida.	Llega a la solución en menos de un minuto cumpliendo exactamente con el formato de salida.	
Interrogación (40%)	No da respuesta a lo preguntado entrega respuesta errónea.			Da respuesta correcta a lo consultado	

Requerimiento base:

Como mínimo su programa debe presentar la estructura siguiente.

```
def lectura_datos(...): ...

def funcion_a(...): ...

def funcion_b(...): ...

def funcion_c(...): ...

def funcion_d(...): ...

def funcion_e(...): ...

def genera_salida(...): ...

if __name__ == "__main__": ...
    i? = funcion_a(i?) # Respuesta al punto a de la tarea
    i? = funcion_b(i?) # Respuesta al punto b de la tarea
    i? = funcion_c(i?) # Respuesta al punto c de la tarea
    i? = funcion_d(i?) # Respuesta al punto d de la tarea
    i? = funcion_e(i?) # Respuesta al punto e de la tarea
    genera_salida(i?)
```

Debe respetar estrictamente el nombre de las funciones, usted completar lo necesario en cada función.

INDICACIONES:

Debe ser desarrollado por exactamente por dos estudiantes (ni más ni menos), por lo tanto con fecha 13 de junio deben subir a la plataforma UCMvirtual un archivo word conteniendo la identificación de cada uno de los integrantes.

La fecha de entrega del proyecto es a través de la misma plataforma es el día viernes 21 de Junio hasta las 23:30 hrs.