

# Tarea #3 – Estudiando un Hábitat Natural

**Profesor: Francisco J. Gutiérrez**

Auxiliares: Mariela Cornejo, Matías Godoy

Esta tarea se puede realizar en grupos de hasta máximo dos personas. Se verificará que se cumpla con la Política de Colaboración.

Todas las funciones **DEBEN** tener receta de diseño completa: contrato, propósito, ejemplos y tests.

Se le sugiere trabajar con tiempo. **NO DEJE LA TAREA PARA EL ÚLTIMO DÍA.** En particular, no se aceptarán entregas fuera de plazo ni por otro medio distinto a U-Cursos.

Solo puede utilizar los contenidos vistos en el curso.

## 1. Introducción

En una reserva natural se está realizando un estudio del movimiento de los animales en su hábitat. Dentro de la reserva natural, hay distintos puestos de observación que permiten registrar la posición de los animales a lo largo del tiempo. La información que se va recopilando se almacena en archivos de texto. Cada línea de estos archivos de texto contiene la información de un avistamiento de un animal, que consiste en la información del puesto de observación, el identificador del animal (un número entero único), la fecha (en formato AAAA-MM-DD) y la hora del avistamiento (de 00:00 a 23:59). Todos estos valores están separados por comas. Un ejemplo de uno de estos archivos de texto es el siguiente (los ‘...’ indican más líneas de texto omitidas):

Puesto7,12,2025-10-13,17:45

Puesto1,3,2025-10-13,18:21

Puesto4,8,2025-10-14,09:17

Puesto9,12,2025-10-14,12:50

...

Existe otro archivo de texto de nombre ‘**animales.txt**’ que contiene la información del identificador del animal y su especie, valores que están separados por comas. Un ejemplo de este archivo de texto es el siguiente (los ‘...’ indican más líneas de texto omitidas):

1,ciervo

2,jabali

3,ciervo

...

12,huemul

...

El problema consiste en programar un sistema que permita identificar en qué zona (definida por cada puesto de observación) es más probable que se encuentre una especie de animal a una cierta hora del día. Para esto, considere lo siguiente:

1. Puede haber uno o más animales de cada especie habitando en la reserva natural.
2. El sistema sólo recibe la hora  $HR$  del día que se quiere estudiar, por lo que todo animal que haya sido avistado entre las  $HR:00$  y las  $HR:59$  en un punto de observación, cuenta como que estaba en esa zona en la hora relevante de estudio.
3. La zona más probable para una especie en una hora especificada, es la moda del puesto de observación donde se hayan avistado los animales de la especie estudiada en dicha hora a lo largo de todo el periodo del cual se tienen datos.

## 2. Trabajo Pedido

La entrega de esta tarea considera dos archivos: `tarea3A.py`, módulo que contendrá el diseño de estructuras y las funciones a implementar sobre la lógica del sistema usando Programación Imperativa, y `tarea3B.py`, módulo que contendrá el diseño de clases y métodos a implementar sobre la lógica del sistema usando Programación Orientada a Objetos.

### 2.1. Módulo `tarea3A` (3.0 pt)

En este archivo debe definir, **siguiendo la receta de diseño** vista en clases, las dependencias entre los conceptos principales del problema e implementar las funciones necesarias para resolver dicho problema. Diseñe tests adecuados para demostrar el correcto funcionamiento de sus funciones.

- (0.5 pt) Usando la función `mutable` del módulo `estructura`, diseñe la estructura de nombre `animal` que permita almacenar, al menos, los siguientes atributos: `identificador`, `especie`, y `listaAvistamientos`. Utilice una lista indexada (de Python) para guardar la información de todos los avistamientos de animales de la misma especie (inicialmente vacía). Luego, defina una variable de estado en donde se almacenarán la lista de todos los animales del hábitat (también vacía, inicialmente).
- (1.0 pt) Escriba el procedimiento `agregarAvistamientos(nombreArchivo) :: str → None`, tal que dado un String con el nombre del archivo que contiene la información de los avistamientos, modifique la variable de estado para agregar todos los avistamientos de este archivo. Note que se agregan avistamientos, así que si existía información previa en la variable de estado no debe borrarse.
- (1.0 pt) Escriba la función `zonaMasProbable(especieAnimal, hora) :: str int → str`, tal que dado el String `especieAnimal` correspondiente al nombre de una especie, devuelva el puesto de observación más probable donde se observarán animales de dicha especie para la hora especificada.
- (0.5 pt) Por un error en los datos producido por el cambio de hora, todas las horas posteriores al 19 de octubre de 2025 están adelantadas en una hora (por ejemplo, esta guardado 13:00 cuando debería ser 12:00). Escriba el procedimiento `corregirHora() :: None → None`, tal que corrija las horas incorrectas en la variable de estado.

Recuerde que para hacer testing con estructuras mutables, se requiere definir un estado conocido de las variables que son mutables, y luego hacer un programa que ejecute los tests y verifique que los cambios a las variables de estado son correctos y que las funciones retornan los valores esperados.

**Nota:** Puede definir cualquier función auxiliar que considere necesaria, recordando seguir la receta de diseño. Puede suponer que no hay identificadores de animales repetidos, y que los archivos de texto a procesar tienen la estructura definida.

## 2.2. Módulo tarea3B (3.0 pt)

En este módulo se debe implementar la misma funcionalidad que en la primera parte, pero usando Programación Orientada a Objetos. Para esto:

- **(0.5 pt)** Diseñe y escriba la clase **Especie**, donde almacene la información de una especie de la reserva natural (la misma información definida para la primera parte, más el conjunto de identificadores de animales de la especie junto con todos sus avistamientos en el tiempo), y con métodos que permitan acceder y actualizar los atributos de la clase. Luego, defina una variable de estado en donde se almacenarán la lista de todos los objetos correspondientes a especies del hábitat (vacía, inicialmente).
- **(1.0 pt)** Escriba el procedimiento `agregarAvistamientos(nombreArchivo) :: str → None`, tal que dado un String con el nombre del archivo que contiene la información de los avistamientos, vaya creando y modificando objetos de la clase **Especie**, y vaya manteniendo la variable de estado con la lista de objetos. Note que se agregan avistamientos, así que si existía información previa no debe borrarse.
- **(1.0 pt)** Escriba el método `zonaMasProbable(hora) :: int → str` de la clase **Especie**, que recibe una hora y devuelva el puesto de observación más probable donde se observarán animales de dicha especie para dicha hora especificada.
- **(0.5 pt)** Por un error en los datos producido por el cambio de hora, todas las horas posteriores al 19 de octubre de 2025 están adelantadas en una hora (por ejemplo, esta guardado 13:00 cuando debería ser 12:00). Escriba el procedimiento `corregirHora() :: None → None`, tal que corrija las horas incorrectas en todos los objetos almacenados en la variable de estado.

En esta parte, puede hacer el testing mediante un programa que pruebe todas las funcionalidades solicitadas.

## 3. Entrega de la Tarea

La entrega de la tarea deben ser los archivos `tarea3A.py` y `tarea3B.py`. Se recibirán entregas únicamente por U-Cursos en el espacio designado para ello. El plazo es hasta el **viernes 21 de noviembre a las 18:00 hrs**, permitiendo atrasos (sin descuento) hasta el **domingo 23 de octubre a las 23:59 hrs**. Una vez finalizado este plazo, NO se recibirán entregas.