

**CC1002-2 Introducción a la Programación****Profesora:** Bárbara Poblete**Auxiliares:** Cristóbal Sepúlveda  
Daniela Campos**Auxiliar 5**

P1. Considere que tiene la estructura `Animal` que tiene como atributos `Especie`, `Clase`<sup>1</sup>, `Dieta`<sup>2</sup> y `Edad`(en meses). Se le entregará una lista de Animales y se le solicita que realice lo siguiente:

- a) Utilizando un **Filtro**, obtenga por cada clase una lista de los animales de esa clase.
- b) Utilizando un **Mapa**, convierta las edades de todos los animales de la lista a años(puede truncar).
- c) Defina un función **frecuencia** que toma como argumento la lista de animales y retorna dos lista una que contiene las especies de animales que hay en la lista y otra que contiene las cantidades de animales de cada especie en la lista anterior.
- d) Haga lo mismo que en el item anterior, pero esta vez con la dieta.
- e) Cambie, para todos los animales de la lista, su dieta de por ejemplo Carnivoro a Come Carne.
- f) Retorne una lista con todos los animales de la lista que son Mamíferos y Carnívoros.

P2. Para representar la información de los alumnos y sus cursos se han definido las siguientes estructuras:

**curso:** `codigo(str) uds(int)`Por ejemplo: `c=curso("CC1002",10)`**alumno:** `nombre(str) cursosInscritos( lista(curso) )`Por ejemplo: `a=alumno("Juan",lista(c,lista(...)))`

Escribir la función de nombre **alumnoConMasUDs** que reciba una lista con todos los alumnos de la Escuela y entregue el (la) alumno(a) que inscribió más unidades docentes

**Notas:**

- el contrato de la función debe ser: `alumnoConMasUDs: lista(alumno) ->alumno`
- suponga que un solo alumno(a) tiene la mayor cantidad de unidades docentes inscritas.
- no es necesario escribir la receta de diseño, ni el testing, ni la plantilla
- puede escribir funciones adicionales
- puede usar directamente (sin el `import`) las funciones `cabeza`, `cola` y la estructura `lista`

P3. Asuma que tiene la estructura `Persona` la cual tiene como atributos `nombre`, `edad` y `genero`. Usted recibirá una lista de personas desordenada y su trabajo es ordenarla de manera lexicográfica de acuerdo a su nombre, indicar quien es la primera persona y la última. Para esto utilice un Arbol Binario de Búsqueda, siguiendo los siguientes pasos:

- a) Cree la estructura Arbol Binario vista en clases.
- b) Defina una función **insertarPersona** que toma como argumento un ABB y una Persona y retorna el ABB con la persona en el lugar que corresponde.
- c) Defina una función **insertarLista** que toma como argumento un ABB y una lista de personas y retorna el ABB con todas las personas de la lista.
- d) Defina una función **primero** que toma como argumento un ABB y retorna la persona que tiene el nombre menor lexicográficamente.
- e) Defina una función **ultimo** que toma como argumento un ABB y retorna la persona que tiene el nombre de mayor lexicográficamente.

---

<sup>1</sup>Clases de Animales: Mamífero, Pez, Ave, Anfibio y Reptil

<sup>2</sup>Dietas de Animales: Carnívora, Hervívora y Omnívora

- f) Defina una función **cantidadDe** que toma como argumento un ABB y un género y retorna la cantidad de personas que son de ese género.
- g) Defina una función **mayorQue** que toma como argumento un ABB y una edad y retorna la cantidad de personas que son mayores a esa edad.
- h) Defina una función **equilibrado** que toma como argumento un ABB y retorna True si es que este está equilibrado, False si no. Un AB está equilibrado cuando cada uno de sus sub-árboles izquierdos y derechos tienen la misma cantidad de elementos. Puede utilizar una función auxiliar pero no es necesario.
- i) Defina una función **aLista** que toma como argumento un ABB y retorna una lista de los elementos del ABB ordenados.

