

# Práctico 6: TDA Listas

## OBJETIVOS:

- Usar el TDA Lista (Independientemente de su forma de implementación: Listas Enlazadas o Listas con arreglos) para resolver problemas.

## PAUTAS:

- Este práctico debe ser implementado haciendo uso de las operaciones básicas ya implementadas en el TDA Lista.

## Ejercicio 1

En el Casino Campestre del departamento de Canelones existe una gran variedad de animales, algunos endémicos y otros traídos de otras latitudes. En el desarrollo de un sistema para gestionar los animales del recinto, los ingenieros plantearon el siguiente diseño de clases:

CasinoCampestre
# animales : ListaSE<Animal>
+ EnPeligroExtincion(especie: <b>string</b> ):lógico
...

Animal
# especie : <b>string</b>
# edad : <b>entero</b>
+ getEspecie() : <b>string</b>
+ getEdad() : <b>entero</b>
...

Implemente el método **EnPeligroExtincion** el cual recibe como parámetro una cadena de caracteres que representa el nombre de una especie, y retorna verdadero en caso de que la especie esté en peligro de extinción o falso en caso contrario. Una especie se encontraría en peligro de extinción si en el Casino Campestre sólo existiesen no más de 3 ejemplares de dicha especie y cada uno de estos animales exceda los 10 años de vida.

## Ejercicio 2

En un almacén de la ORT se desea auditar todos los insumos informáticos presentes en el mismo. Dicho proceso se precisa que sea automatizado y en ese sentido los desarrolladores del sistema plantearon el siguiente diseño de clases:

Almacen
# insumos : ListaSE<Insumo>
+ InsumoInsuficiente(tipo: <b>string</b> ):lógico
...

Insumo
# tipo : <b>string</b>
# serie : <b>entero</b>
# garantia : <b>entero</b>
+ getTipo() : <b>string</b>
+ getSerie() : <b>entero</b>
+ getGarantia() : <b>entero</b>
...

Implemente el método **InsumoInsuficiente** el cual recibe como parámetro una cadena de caracteres que representa el nombre de un tipo de insumo, y retorna verdadero en caso que la presencia de ese tipo de insumo sea insuficiente en el almacén. Un tipo de insumo se considera insuficiente en el almacén si sólo existiesen no más de 5 insumos de dicho tipo y cada uno de estos tuviera no más de 1 año de garantía.

## Ejercicio 3

Se desea desarrollar un asistente matemático para el trabajo con Conjuntos. Un conjunto se puede modelar a través de una lista que contenga los elementos que pertenecen al conjunto. Una característica fundamental de los conjuntos es que los elementos no se repiten dentro del mismo. Implemente la funcionalidad **Union**, **EsSubConjunto** e **Iguals** de la clase conjunto. Atendiendo a los requerimientos del sistema se obtuvo el siguiente diseño de clases:

Conjunto<T>
# elementos: ListaSE<T>
...
+ Conjunto()
+ Conjunto(elementos: ListaSE<T>)
+ Union(c: Conjunto<T>): Conjunto<T>
+ EsSubConjunto(c: Conjunto<T>): Lógico
+ Iguals(c: Conjunto<T>): Lógico
...