**CORTE 1 - TALLER 1**
**Estructura de datos lineales**
**Por: Guillermo Andres De Mendoza Corrales**

---

Implementación de los algoritmos para un ArrayList y un LinkedList, mediante POO

## Punto 1: ArrayList

Los estudiantes deberán realizar la implementación de cada uno de los siguientes métodos para la clase ArrayList, incluyendo las validaciones o gestión de errores necesarios:

```
class ArrayList:

    # size: initial capacity of the collection
    # initial_elements: allow the collection to start with some elements
    def __init__(self, size=100, initial_elements=[]):
        pass

    # return an str of the collection
    def __str__(self):
        pass

    # return the length of the elements in the collection
    def __len__(self):
        pass

    # return a boolean that implies if the collection is empty or not
    def isEmpty(self):
        pass

    # return the element of the collection in the index possition
    # Error: the index dont exist
    def __getitem__(self, index):
        pass

    # allow the collection to be called in a for loop
    def __iter__(self):
        pass

    # return a boolean value representing the existence of an element in the collection
    def __contains__(self, element):
        pass

    # add the element to the end of the collection
    def append(self, element):
        pass

    # add the element to the collection at the requested index
    # Error: non existing index in the collection
    def insert(self, index, element):
        pass

    # remove an element in the collection by its value
    # Error: the element dont exist in the collection
    def remove(self, element):
        pass

    # remove and return the element in the collection by its index
    def pop(self, index):
        pass

    # remove all elements in the collection
    def clear(self):
        pass
```

**Punto 2: LinkedList**

Los estudiantes deberán realizar la implementación de cada uno de los siguientes métodos para la clase LinkedList, incluyendo las validaciones o gestión de errores necesarios:

```
class LinkedList:

    # initial_elements: allow the collection to start with some elements
    def __init__(self, initial_elements=[]):
        pass

    # return an str of the collection
    def __str__(self):
        pass

    # return the length of the elements in the collection
    def __len__(self):
        pass

    # return the element of the collection in the index possition
    # Error: the index dont exist
    def __getitem__(self, index):
        pass

    # return a boolean that implies if the collection is empty or not
    def isEmpty(self):
        pass

    # allow the collection to be called in a for loop
    def __iter__(self):
        pass

    # return a boolean value representing the existence of an element in the collection
    def __contains__(self, element):
        pass

    # add the element to the end of the collection
    def append(self, element):
        pass

    # add the element to the collection at the requested index
    # Error: non existing index in the collection
    def insert(self, index, element):
        pass

    # remove an element in the collection by its value
    # Error: the element dont exist in the collection
    def remove(self, element):
        pass

    # remove and return the element in the collection by its index
    def pop(self, index):
        pass

    # remove all elements in the collection
    def clear(self):
        pass
```