# UNIVERSIDAD NACIONAL DE INGENIERIA FACULTAD DE CIENCIAS

**Escuela Profesional de Ciencias de la Computación Algoritmos y Estructuras de Datos**

# 2° Practica Calificada - Parte II

Fecha: 09.05.2025

Tiempo: 1h 30’

Total=12pts

Código

Apellidos y Nombres

***No modifique la pag.1 del formato de la prueba, consigne sus respuestas a partir de la pag.3***

*Todas las clases, interfaces que formen parte de su solución deben tener como encabezado su Código de estudiante.*

En el proyecto del curso: (<https://github.com/zcodlab/251CC232AProject>)

1. Implemente métodos:
   1. Que retorne la union de dos listas dadas. Suponga que las dos listas contienen objetos String. No hay repeticiones en cada una de las dos listas de entrada. Si las dos listas de entrada son, por ejemplo,

(“uno”,”dos”,”tres”,”cuatro”,”cinco”)

y (“uno”,”cinco”,”cero”,”cuatro”,”ocho”)

entonces su unión es (“uno”,”dos”,”tres”,”cuatro”,”cinco”,”cero”,”ocho”)

El prototipo del método es como sigue:

Public ListTDA<String> unión(ListTDA<String> list1, ListTDA<String> list2)

* 1. Que retorne la interseccion de dos listas dadas. por ejemplo,

(“uno”,”dos”,”tres”,”cuatro”,”cinco”)

y (“uno”,”cinco”,”cero”,”cuatro”,”ocho”)

entonces su interseccion es (“uno”,”cuatro”,”cinco”)

El prototipo del método es como sigue:

Public ListTDA<String> interseccion(ListTDA<String> list1, ListTDA<String> list2)

* 1. Que retorne la diferencia de dos listas dadas. por ejemplo,

(“uno”,”dos”,”tres”,”cuatro”,”cinco”)

y (“uno”,”cinco”,”cero”,”cuatro”,”ocho”)

entonces su diferencia es (“dos”,”tres”,”cuatro”)

El prototipo del método es como sigue:

Public ListTDA<String> diferencia(ListTDA<String> list1, ListTDA<String> list2)

-Implemente los métodos en las clases concretas LinkedListTDA y ArrayListTDA **(2pts c/uno por implementación completa, incluyendo menu)**

-Implemente un menú de opciones: Crear Lista 1, Crear Lista 2, Realizar Union, Realizar Intersección, Realizar Diferencia, Visualizar Lista.

-La opción visualizar es obligatorio para considerar el puntaje:

1. Los elementos en ListTDA están desordenados. Defina una nueva interface ListSortTDA donde los elementos están ordenados en orden ascendente. Los elementos que agregamos a una lista ordenada deben ser instancias de una clase que implementa la interfaz Comparable de modo que se puedan comparar. Los métodos para la interfaz ListSortTDA son esencialmente los mismos que los de la interface ListTDA excepto por el método add que especifica la posición índice de un elemento recientemente agregado. Con la interface ListSortTDA, solo hay un método add que agrega un elemento a la posición correcta, de modo que la lista resultante esta ordenada. En la lista ordenada no se permiten repeticiones. Lance un IllegalArgumentException cuando se intente agregar un valor repetido. Un elemento e1 es repetido si hay un e2 en la lista tal que e1.compareTo(e2)==0. Implemente la interface ListSortTDA, nombre la clase ArrayListSortTDA y LinkedListSortTDA.

Implemente un menú de opciones, la opción visualizar es obligatorio para considerar el puntaje:

a)Agregar (2pts)

b)Verificar (1pt)

c)Obtener ubicación (1pt)

d)Eliminar (2pts)

e)Visualizar

**Procedimiento de Entrega de la Prueba:**

Presentar el presente formato, consignando su Código, apellidos y nombre, capturas de pantalla que demuestre la resolución de cada uno de los enunciados. El presente formato y su código fuente, publíquelo en su cuenta Github (modo privado) agregando como colaborador del proyecto a la cuenta del docente del curso: **zcodlab**

Nombre del Proyecto en Github: **251CC232A<CODIGOESTUDIANTE>PC2**

**Asegúrese que el nombre de su cuenta Github lo identifique claramente: ApellidosNombres**

*\*Si se detecta 2 o más trabajos iguales se anula a todos los implicados*