

### **PARTE 3: Ejercicio de programación JAVA**

**Duración 60 minutos**

Este ejercicio comprende **2** pasos:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo de los casos de prueba ("test case", al menos uno) para verificar la corrección de la funcionalidad implementada.

#### **ESCENARIO**

La multinacional de la informática ORACUL™, diseñador del lenguaje JavaAED, desarrolla diversos algoritmos avanzados para ser aplicados en la industria.

Los ingenieros de la empresa investigaron ciertas situaciones en que concluyeron que sería conveniente contar con una buena, robusta y eficiente implementación del algoritmo "binsort" o "bucketsort" para ordenar datos.

La gerencia tecnológica se decide a realizar un llamado público a la comunidad para desarrollar esta implementación.

#### **PASO 1: Funcionalidad a desarrollar:**

Crear un paquete en NetBeans para desarrollar el programa, e importar todos los paquetes desarrollados para Clasificadores (incluir también los utilitarios para la lectura de archivos).

Crear una clase JAVA "**Parcial3**", que tendrá un método "**main**" para ejecutar el programa.

Agregar al **TClasificador** utilizado en las UT6 y UT7 un método que realice la ordenación por **binsort**, de un conjunto de datos pasado por parámetro en la forma ya habitual.

**Se requiere entonces:**

1. Leer un conjunto de datos provisto en el archivo "**clavesBinsort.txt**". Este contiene claves numéricas, entre **0** y **99**.
2. Seleccionar la cantidad de urnas a ser utilizadas, de acuerdo a las claves.
3. Utilizando clases de la api de colecciones de java (arraylist, linkedlist, etc) que resulten más convenientes desde el punto de vista de este algoritmo, implementar las urnas.
4. Completar y probar el algoritmo.
5. Escribir en un archivo "**salida.txt**" el conjunto de datos ordenado resultante de la ejecución del algoritmo.
6. Agregar un comentario en lenguaje natural que explique qué se pretende hacer en el método y cómo se usan las estructuras.

#### **PASO 2: TEST CASE.**

Implementa el o los **Casos de Prueba** necesarios para verificar el correcto funcionamiento del método desarrollado, es decir, el conjunto al final del algoritmo debe estar ordenado.

#### **ENTREGA**

Subir a la webasignatura, en la tarea "**PARCIAL3-PARTE3**" un archivo **zip**, con **TODO** el código utilizado y el archivo de salida.