Departamento de TIC
Algoritmos y Estructuras de Datos
Laboratorio Unidad 2
Algoritmos de Ordenamiento, Búsqueda y Persistencia.

Objetivos

Unidad 2: Algoritmos de Ordenamiento, Búsqueda y Persistencia

Al finalizar esta unidad, el estudiante estará en capacidad de:

OE3.3 Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera.

OE3.4 Manipular archivos de texto y utilizarlos para implementar requerimientos del cliente relacionados con persistencia.

OE4.1 Implementar algoritmos clásicos de ordenamiento de datos en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE4.2 Implementar algoritmos clásicos de búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE4.4 Calcular el tiempo de ejecución de un algoritmo por medio de las operaciones de tiempo del sistema

Preparación

- Lea cuidadosamente el enunciado, la documentación suministrada y cada uno de los puntos que debe desarrollar antes de empezar su desarrollo. Pregunte a su profesor cualquier duda respecto al enunciado o a los requerimientos funcionales que debe desarrollar.
- El trabajo debe ser realizado individualmente.
- El trabajo será entregado en la fecha y hora establecida en Moodle.
- Usted debe utilizar git para manejar el versionamiento de su trabajo desarrollado localmente y manejar como repositorio remoto **privado**, un repositorio en GitHub o un proyecto GitLab. Su trabajo debe ser gestionado con git desde el inicio del desarrollo, los commit deben hacerse regularmente, así como los push al remoto. Esto se verificará con las fechas de los commits. En el momento de la fecha de entrega máxima, usted debe hacer público su repositorio. Recuerde las convenciones de nombre y estructura de directorios indicadas en esta presentación de Git.

Enunciado

Un inversionista de la ciudad quiere construir un sistema de información para tener diferentes clubes de mascotas a nivel nacional, le ha solicitado que construya un programa para el registro de los clubes, de los dueños y de sus respectivas mascotas. La información que se necesita es la siguiente:

- De los clubes es necesario registrar la identificación del club, el nombre, la fecha de creación y tipos de mascotas.
- De los dueños se debe registrar la identificación del dueño, los nombres, los apellidos, la fecha de nacimiento y que tipo de mascota prefiere.
- Y de las mascotas se quiere registrar la identificación de la mascota, su respectivo nombre, fecha de nacimiento que se conozca, género y tipo de mascota.

Una de las necesidades detectadas, es que la información de los clubes debe ser almacenada para su gestión en un archivo plano y para la información de los dueños y las mascotas se debe usar archivos con serialización. El inversionista espera que existan diversos reportes en el sistema de información, estos son:

- Se debe poder generar listados ordenados de los clubes, los dueños y las mascotas por cualquier criterio de los campos solicitados.
- Listado ordenado de los dueños, según el número de mascotas.
- Listado ordenado de los clubes, según el número de dueños.
- El sistema deberá tener opciones de búsqueda por cualquier criterio de los campos solicitados, una solicitud del inversionista es que se realicen de dos formas (usando la manera tradicional y búsqueda binaria) y adicionalmente quiere poder ver en pantalla la comparación de los tiempos que gasta el sistema en realizar la búsqueda tradicional vs la búsqueda binaria.

Tenga en cuenta que cuando se registra una mascota, está será registrada al dueño que se encuentre actualmente en la pantalla, la aplicación deberá verificar que el dueño exista (este registrado en el sistema). El sistema también deberá considerar las siguientes reglas:

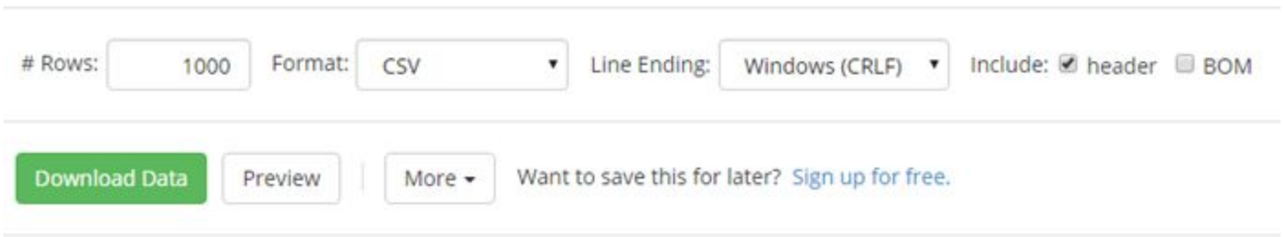
- Para registrar un dueño:
  - No deben existir dos dueños con la misma identificación.
- Para registrar una mascota:
  - Ninguna mascota para el mismo dueño se podrá llamar igual.
- Para eliminar un club, un dueño o una mascota, deberá hacerse por el número de identificación o el nombre.

UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Estructuras de Datos
Laboratorio Unidad 2
Algoritmos de Ordenamiento, Búsqueda y Persistencia.

- Implementar y utilizar los tres métodos de ordenamiento clásicos: burbuja, selección e inserción.
- Implementar y utilizar las dos estrategias de búsqueda clásicas: secuencial y binaria.
- Utilizar la interface Comparable.
- Utilizar la interface Comparator.
- Utilizar el método de ordenamiento de la clase Arrays utilizando:
  - Comparable.
  - Comparator.

Con el objetivo de realizar pruebas aceptables al sistema, se debe generar una cantidad de datos importante y sugerimos utilizar la herramienta web https://www.mockaroo.com que permite generar datos aleatorios de forma personalizada. Se debe generar el máximo de filas posible para hacer buenas pruebas y el formato en que se debe exportar debe ser CSV el cual delimita los archivos de texto con coma por defecto.



Ya que el máximo es 1000, realice pruebas del tiempo de la consulta para archivos de entrada mas grande como 100000, por ejemplo (generando 100 veces 1000 y uniendolos).

Entregables.
1. Requerimientos funcionales y no funcionales.
2. Diagrama de clases de modelo y control de la interfaz (no generado automáticamente).
3. Implementación completa de todos los requerimientos en Java.
4. Tabla de trazabilidad de requerimientos vs métodos (tabla con una columna de los requerimientos, tal que, por cada requerimiento se indica en la columna siguiente todos los métodos que contribuyen a resolverlo).
5. Diseño y pruebas de todas las funcionalidades no triviales.

Fecha de Entrega Máxima: la que está definida a través de Moodle. El laboratorio debe trabajarse y entregarse **individualmente**. Lo que usted debe entregar de su trabajo es la url de su repositorio en GitHub o proyecto en GitLab. Recuerde que el repositorio o proyecto debe ser privado durante el desarrollo del laboratorio y hacerse público solo en el momento justo de la entrega máxima indicada aquí.