

## PARCIAL I ALGORITMOS

### Integrantes:

NATALIA BETANCOURT  
MARIA JOSE POLANCO PUERTA  
LINA MARIA PERALTA DELGADO

### Brief

La fundación llamada **Peludos Sin Hogar** se encarga de recoger perros abandonados, tratarlos y ponerlos en adopción.

La fundación ha hecho una labor impresionante, ha logrado salvar una gran cantidad de perros, sin embargo, se ha vuelto un tema crítico de manejar. Por esta razón la fundación ha decidido desarrollar un aplicativo de escritorio que les ayude a:

1. Cargar la información de sus peludos, los cuales tiene los siguientes atributos: id numérico, nombre, edad, raza y fecha de nacimiento. La información de los perros se encuentra en dos archivos .txt . El archivo uno contiene el id y nombre. El archivo dos tiene id, raza y fecha de nacimiento del perro, deben tener cuidado con los id, no están en el mismo orden.
2. Se desea que el orden natural de la lista se dé por el id de cada perro.
3. El programa debe guardar un archivo .txt con la lista de los perros ordenados por cada atributo (un archivo con los perros ordenados por nombre, otro archivo ordenados por id y así por cada atributo). El nombre y la raza de los perros debe quedar en minúscula.
4. El programa debe tener una interfaz gráfica la cual muestre la información de los txt, la cual debe ordenarse según la interacción que ustedes escojan (presionando teclas, botones, etc) es libre de cada desarrollador, sin embargo, debe tener instrucciones de cómo organizar la lista por cada atributo.

### Bonus (10%)

1. Al seleccionar dos elementos de la lista o arrastrar un elemento a la posición del otro elemento estos se intercambian de posición.
2. Agregar imagen a cada perro según su raza.

### Instrucciones finales

- Para cargar los archivos están en la carpeta data/imports y para exportar los archivos data/exports

## Contexto

Se debe crear un programa para la fundación **Peludos Sin Hogar** para organizar la información de cada perro. Esta información se encuentra en dos archivos que contienen: el ID, el nombre, la fecha de nacimiento y la raza del perro. Se debe visualizar la información, poder ordenarla según cada atributo y guardarla organizadamente en un nuevo archivo, en este el nombre y la raza deben de ir en minúscula.

## Entidades

1. Main
2. Archivo TXT 1
3. Archivo TXT 2
4. Controller
5. Logic
6. Dog
7. DogName
8. DogDate
9. DogBreed

## Requerimientos Funcionales:

RF1	
Descripción	El programa debe cargar los TXT en arreglos.
Entradas	Archivo TXT.
Salidas	Arreglo string.
Precondición	Deben existir los TXT.
Postcondición	-

RF2	
Descripción	El programa debe pintar las listas con la información de cada perro en pantalla.
Entradas	Listas con la información de cada perro
Salidas	-
Precondición	La lista deberá filtrar la información de los perros
Postcondición	Lista aparece en pantalla

RF3	
Descripción	El programa debe tener como ordenamiento natural el ID del perro.
Entradas	Archivo TXT
Salidas	ID de los perros ordenados naturalmente
Precondición	Deben existir los TXT

<b>Postcondición</b>	-
----------------------	---

<b>RF4</b>	
<b>Descripción</b>	<b>El programa</b> debe guardar un TXT con la lista ordenada por ID
<b>Entradas</b>	Listas perros ordenada por ID.
<b>Salidas</b>	Archivo TXT.
<b>Precondición</b>	-
<b>Postcondición</b>	-

<b>RF5</b>	
<b>Descripción</b>	<b>El programa</b> debe guardar un TXT con la lista ordenada por nombre en minúscula.
<b>Entradas</b>	Listas perros ordenadas por nombre.
<b>Salidas</b>	Archivo TXT.
<b>Precondición</b>	-
<b>Postcondición</b>	-

<b>RF6</b>	
<b>Descripción</b>	<b>El programa</b> debe guardar un TXT con la lista ordenada por raza en minúscula.
<b>Entradas</b>	Listas perros ordenadas por raza.
<b>Salidas</b>	Archivo TXT.
<b>Precondición</b>	-
<b>Postcondición</b>	-

<b>RF7</b>	
<b>Descripción</b>	<b>El programa</b> debe guardar un TXT con la lista ordenada por fecha
<b>Entradas</b>	Listas perros ordenadas por fecha.
<b>Salidas</b>	Archivo TXT.
<b>Precondición</b>	-
<b>Postcondición</b>	-

<b>RF8</b>	
<b>Descripción</b>	<b>El programa</b> debe ordenar la información según la tecla presionada (keyPressed).
<b>Entradas</b>	Presionar la tecla correspondiente.
<b>Salidas</b>	Lista de perros ordenada.
<b>Precondición</b>	-
<b>Postcondición</b>	-

RF9	
<b>Descripción</b>	El programa debe mostrar en la interfaz gráfica la imagen correspondiente a cada raza.
<b>Entradas</b>	Raza del perro.
<b>Salidas</b>	Imagen del perro según su raza.
<b>Precondición</b>	Cargar imágenes formato png o jpg.
<b>Postcondición</b>	-

RF10	
<b>Descripción</b>	El programa debe mostrar en la interfaz las instrucciones para que el usuario sepa cómo ordenar los datos
<b>Entradas</b>	-
<b>Salidas</b>	-
<b>Precondición</b>	-
<b>Postcondición</b>	Texto en pantalla con las instrucciones.

RF11	
<b>Descripción</b>	El programa debe guardar un nuevo TXT con la información ordenada.
<b>Entradas</b>	Datos ordenados
<b>Salidas</b>	Datos ordenados impresos en un nuevo TXT
<b>Precondición</b>	-
<b>Postcondición</b>	-

RF12	
<b>Descripción</b>	Los archivos se deben cargar en la carpeta data/imports y para exportarlos se deben cargar en data/exports
<b>Entradas</b>	Archivo TXT inicial o Archivo TXT nuevo
<b>Salidas</b>	Archivos en la carpeta correspondiente
<b>Precondición</b>	-
<b>Postcondición</b>	-

### Requerimientos No Funcionales

- Debe ser lenguaje Java.
- Debe estar hecho con librería en Processing.
- Se debe de visualizar toda la información en la pantalla.
- Se debe especificar en la pantalla que teclas se usan para los ordenamientos.

**KeyPressed correspondientes a cada dato:**

- ID: tecla “i”
- Name: tecla “ n “
- Breed: tecla “ b “
- Date: tecla “ d “

## Diagrama de clases

