**Examen Parcial I**

**DCA – 2020-2**

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, nombre y edad del perro. 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**

5. Al seleccionar dos elementos de la lista se intercambia de posición.

1. **Análisis y diseño de software**

**Contexto:** Una base de datos de los peludos la cual muestre cada uno con las siguientes características: id numérico, nombre, edad, raza y fecha de nacimiento.

**Requerimientos funcionales**

|  |  |
| --- | --- |
| RF1 | |
| **Descripción:** | El programa debe cargar la información de los peludos desde dos txt teniendo en cuenta el ID |
| **Entradas:** | Dos archivos txt |
| **Salidas:** | La carga de información de los peludos como: id numérico, nombre, edad, raza y fecha de nacimiento. |
| **PreCondiciones:** | Debo tener los txt dentro de mis archivos del programa |
| **Post-Condiciones:** |  |

|  |  |
| --- | --- |
| RF2 | |
| **Descripción:** | El programa debe juntar la información partida en dos txt de cada uno de los peludos |
| **Entradas:** | Dos archivos txt |
| **Salidas:** | La carga de información de los peludos registrada correctamente en un arreglo de objetos Mascota |
| **PreCondiciones:** | Debo tener los txt dentro de mis archivos del programa |
|  |  |

|  |  |
| --- | --- |
| RF3 | |
| **Descripción:** | El programa debe poder ordenar de forma natural a los peludos por el ID |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | La lista de mascotas está ordenada naturalmente por ID |
| **PreCondiciones:** | Arraylist mascotas !=null, mi clase Mascota implemente la interfaz comparable |
|  |  |

|  |  |
| --- | --- |
| RF4 | |
| **Descripción:** | El programa debe guardar un archivo .txt con la lista de los perros ordenados por nombre |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | La lista de mascotas está ordenada por nombre, la lista ordena se guardó en un txt |
| **PreCondiciones:** | Arraylist mascotas !=null, tener una clase implementando comparator que compare el nombre de las mascotas |
|  |  |

|  |  |
| --- | --- |
| RF5 | |
| **Descripción:** | El programa debe guardar un archivo .txt con la lista de los perros ordenados por raza |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | La lista de mascotas está ordenada por raza, la lista ordena se guardó en un txt |
| **PreCondiciones:** | Arraylist mascotas !=null, tener una clase implementando comparator que compare la raza de las mascotas |

|  |  |
| --- | --- |
| RF6 | |
| **Descripción:** | El programa debe guardar un archivo .txt con la lista de los perros ordenados por fecha de nacimiento |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | La lista de mascotas está ordenada por fecha de nacimiento , la lista ordena se guardó en un txt |
| **PreCondiciones:** | Arraylist mascotas !=null,tener una clase implementando comparator que compare la fecha de nacimiento de las mascotas |

|  |  |
| --- | --- |
| RF7 | |
| **Descripción:** | El programa debe en todos los archivos guardados poner el nombre y la raza de los peludos en minúscula |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | todos los archivos guardados tienen el nombre y la raza de los peludos en minúscula |
| **PreCondiciones:** |  |

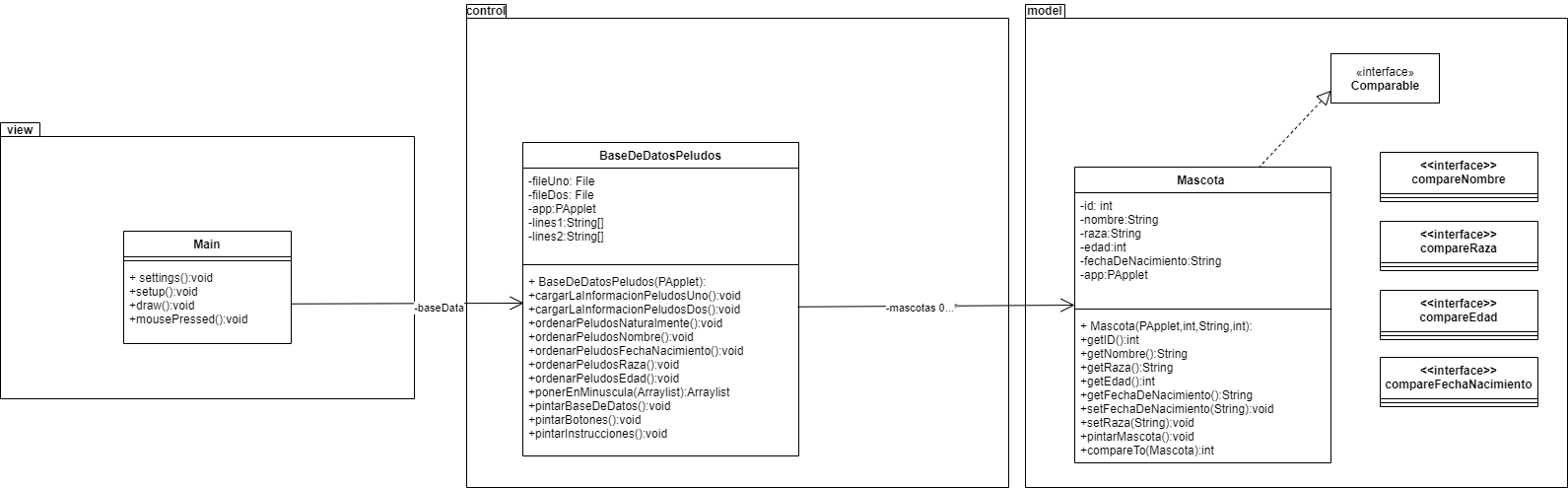
|  |  |
| --- | --- |
| RF8 | |
| **Descripción:** | El programa debe poder mostrar mediante una interfaz gráfica la información de los txt |
| **Entradas:** | Arraylist mascotas, los archivos txt |
| **Salidas:** | Los datos de las mascotas se han mostrado en la pantalla. |
| **PreCondiciones:** | mascotas!=null |

|  |  |
| --- | --- |
| RF9 | |
| **Descripción:** | El programa debe tener un botón para que el usuario ordene por cada uno de los atributos (id,nombre,raza,edad,fecha nacimiento) |
| **Entradas:** | Arraylist mascotas |
| **Salidas:** | Los datos de las mascotas se han mostrado en la pantalla ordenados según el botón presionado |
| **PreCondiciones:** | mascotas!=null, tener las clases comparadoras de cada uno de los atributos que implementan la interfaz comparator. |

**Requerimientos no funcionales**

|  |  |
| --- | --- |
| RNF 1 | |
| **Descripción:** | El programa debe tener instrucciones de cómo organizar la lista por cada atributo. |

**UML**

****

[**https://drive.google.com/file/d/1FU1yBxomqM\_9\_cW9oj2g5uYNtZ4hwGhH/view?usp=sharing**](https://drive.google.com/file/d/1FU1yBxomqM_9_cW9oj2g5uYNtZ4hwGhH/view?usp=sharing)