Manuel David Castaño Saldarriaga

A00358994

Requerimientos funcionales

|  |  |
| --- | --- |
| Nombre | R1. Registrar clubes, dueños y mascotas. |
| Resumen | Se requiere que el programa añada clubes, dueños y mascotas al sistema. |
| Entradas | Atributos del elemento por agregar |
| Salida | Ninguna |

|  |  |
| --- | --- |
| Nombre | R2. Ordenar clubes, dueños y mascotas. |
| Resumen | Se requiere que el programa sea capaz de ordenar los clubes, dueños y mascotas por cualquier criterio que el usuario escoja. |
| Entradas | <Ninguna> |
| Salida | Listado ordenado de los clubes, dueños o mascotas. |

|  |  |
| --- | --- |
| Nombre | R3. Buscar clubes, dueños y mascotas. |
| Resumen | Se requiere que el programa sea capaz de buscar los clubes, dueños y mascotas por cualquier criterio que el usuario escoja, de la manera tradicional y binaria, además de mostrarle al usuario el tiempo que se tardó en buscar en cada caso. |
| Entradas | Atributo por el que se quiere buscar el dueño, club o mascota. |
| Salida | Un mensaje diciendo si el elemento existe o no. |

|  |  |
| --- | --- |
| Nombre | R4. Eliminar clubes, dueños y mascotas. |
| Resumen | Se requiere que el programa elimine los clubes, dueños o mascotas. |
| Entradas | Identificación o nombre del elemento a eliminar. |
| Salida | Un mensaje diciendo si fue posible eliminarlo o no. |

Requerimientos no funcionales

* No deben existir dos dueños con la misma identificación dentro del programa.
* Un dueño no puede tener dos mascotas con el mismo nombre.
* Implementar y utilizar los tres métodos de ordenamiento clásicos.
* Utilizar las interfaces Comparable y Comparator.
* Guardar la información de los clubes en archivos planos y la información de los dueños y sus mascotas en archivos serializados.

Trazabilidad

|  |  |  |
| --- | --- | --- |
| Requerimiento | Clase | Método |
| R1. Registrar clubes, dueños y mascotas. | Program | +addClub(Club): void |
| +addOwner(Owner): void |
| +addPet(Pet): void |
| Club | +addOwner(Owner): void |
| +addPet(Pet): void |
| Owner | +addPet(Pet): void |
| R2. Ordenar clubes, dueños y mascotas. | Program | +orderClubBy\*(): void |
| +orderOwnerBy\*(): void |
| +orderPetBy\*(): void |
| Club | +orderOwnerBy\*(): void |
| +orderPetBy\*(): void |
| Owner | +orderPetBy\*(): void |
| R3. Buscar clubes, dueños y mascotas. | Program | +findSequentialClubBy\*(String): void  +findBinaryClubBy\*(String): void |
| +findSequentialOwnerBy\*(String): void  +findBinaryOwnerBy\*(String): void |
| +findSequentialPetBy\*(String): void  +findBinaryPetBy\*(String): void |
| Club | +findSequentialOwnerBy\*(String): void  +findBinaryOwnerBy\*(String): void |
| +findSequentialPetBy\*(String): void  +findBinaryPetBy\*(String): void |
| Owner | +findSequentialPetBy\*(String): void  +findBinaryPetBy\*(String): void |
| R4. Eliminar clubes, dueños y mascotas. | Program | +eliminateClub(String): void |
| +eliminateOwner(String): void |
| +eliminatePet(String): void |
| Club | +eliminateOwner(String): void |
| +eliminatePet(String): void |
| Owner | +eliminatePet(String): void |