1. Requerimientos Funcionales

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Registrar un club |
| **Resumen:** | Permitir registrar un club con el nombre, fecha de creación y tipos de mascotas |
| Entradas: | Nombre, creación fecha, tipos de mascotas |
|  |  |
| Resultados: | Club registrado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Registrar un dueño |
| **Resumen:** | Permitir registrar la identificación del dueño, los nombres, apellidos, la fecha de nacimiento y el tipo de mascota que prefiere  Cond: No deben existir dos dueños con la misma identificación. |
| Entradas: | Nombre, apellido, fecha de nacimiento y tipo de mascota que prefiere |
|  |  |
| Resultados: | Dueño registrado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Registrar una mascota |
| **Resumen:** | Permitir registrar una mascota con su respectivo nombre, fecha de nacimiento que se conozca, genero y tipo de mascota  Cond: Ninguna mascota para el mismo dueño se podrá llamar igual. |
| Entradas: | Nombre, fecha de nacimiento, genero y tipo de mascota |
|  |  |
| Resultados: | Mascota registrada |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #4. Almacenar la información |
| **Resumen:** | Permitir almacenar la información de los clubes en un archivo plano |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Información del club almacenada |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #5. Serializar información |
| **Resumen:** | Permitir serializar la información de los dueños y las mascotas |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Información serializada |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #6. Generar reportes |
| **Resumen:** | Permitir generar listados ordenados de los clubes, los dueños, y las mascotas por cualquier criterio de los campos solicitados |
| Entradas: | Ordenar por campo solicitado |
|  |  |
| Resultados: | Clubes, dueños y mascotas ordenados por un criterio determinado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #7. Generar listado ordenado de los dueños |
| **Resumen:** | Permitir generar un listado ordenado de los dueños según el número de mascotas |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Listado ordenado de los dueños |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #8. Generar listado ordenado de los clubes |
| **Resumen:** | Permitir generar un listado ordenado de los clubes según el número de dueños |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Listado ordenado de los clubes |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #9. Permitir opciones de búsqueda |
| **Resumen:** | Permitir opciones de búsqueda por cualquier criterio de los campos solicitados, ya sea de manera tradicional o 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 |
| Entradas: | Campos solicitados |
|  |  |
| Resultados: | Resultado búsqueda tradicional |
| s | Resultado búsqueda binaria |

|  |  |
| --- | --- |
| **Nombre:** | R. #10. Verificar que el dueño exista |
| **Resumen:** | Permitir verificar si existe un dueño para registrar una mascota |
| Entradas: | Ninguna |
|  |  |
| Resultados: | SI: Mascota registrada |
|  | No: Mascota no registrada |

|  |  |
| --- | --- |
| **Nombre:** | R. #11. Permitir eliminar un club, mascota o dueño |
| **Resumen:** | Permitir eliminar correctamente un club, mascota o dueño por número de identificación o nombre |
| Entradas: | Número de identificación o nombre |
|  |  |
| Resultados: | Club, mascota o dueño eliminado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | RNF. #1. Implementar métodos de ordenamiento clásicos |
| **Resumen:** | El programa debe implementar 3 métodos de ordenamiento clásico |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Programa con 3 métodos de ordenamiento clásico |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | RNF. #2. Implementar estrategias de búsqueda |
| **Resumen:** | El programa debe implementar estrategias de búsqueda clásicas |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Programa con 2 estrategias de búsqueda clásicas |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | RNF. #3. Utilizar la interface Comparable |
| **Resumen:** | El programa debe utilizar la interface Comparable |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Programa con interfaz Comparable |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | RNF. #4. Utilizar la interface Comparator |
| **Resumen:** | El programa debe utilizar la interface Comparator |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Programa con interfaz Comparator |
|  |  |

1. Diagrama de Clase

Imagen que contiene texto

Descripción generada automáticamente

1. Trazabilidad

|  |  |  |
| --- | --- | --- |
| Requerimiento | Método | Clase |
| R1 | loadClubs | CentralSistem |
| R2 | addOwner | CentralSistem |
| R3 | Pet() | Pet |
| R4 | loadClubs | CentralSistem |
| R5 | saveClubs | CentralSistem |
| R6 | toString()  sort\*By\*  search\*  searchLineal\*  serachBinary\* | CentralSistem |
| R7 | SortOwnerByKindOfPet | CentralSistem |
| R8 | sortByClubBy\*\* toString | CentralSistem |
| R9 | searchLineal  serachBinary | CentralSistem |
| R10 | loadOwners | CentralSistem |

1. Diseño de casos de pruebas unitarias

Configuración de los escenarios

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1 | CentralSistem | Se crea escenario con un nuevo Objeto tipo CentralSistem para pruebas correctas de datos recolectados por un texto plano |

Diseño de Casos de Prueba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Ordenar Club, Dueños, Cargar Dueños y Club | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| CentralSistem | loadClubs | setupScenary1 | Data/clubs.csv | True  Lee correctamente y inicializa todos los objetos generados en una página web |
| CentralSistem | loadOwner | setupScenary1 | Data/owners.csv | True Lee correctamente y inicializa todos los objetos generados por la página web |
| CentralSistem | sortByClubsById | setupScenary1 | Ninguno | True Ordena correctamente los clubes por número de Id |
| CentralSistem | sortByClubByName | setupScenary1 | Ninguno | True Ordena correctamente todos los objetos de tipo Club lexicográficamente |
| CentralSistem | sortByOwnerByName | setupScenary1 | Ninguno | True Ordena correctamente los objetos de tipo Owner lexicográficamente |
| CentralSistem | sortByOwnerByLastName | setupScenary1 | Ninguno | True Ordena correctamente los objetos tipo Owner lexicográficamente |