|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Businessman | ordeningClubById(); | Debe de ordenar la lista de clubs de acuerdo con el id | ------------------- | Es verdadero,  Puesto lo ordeno correctamente |
| Businessman | ordeningClubByName(); | Debe ordenar los clubs de acuerdo con el nombre | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Businessman | ordeningClubByClients (); | Debe ordenar los clubs de acuerdo con la cantidad de clientes que tengan | ---------------------- | Es verdadero, puesto que ordeno correctamente |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Club | ordeningClientId(); | Debe de ordenar la lista de clientes de acuerdo con el id | ------------------- | Es verdadero,  Puesto lo ordeno correctamente |
| Club | ordeningClientName(); | Debe ordenar los clientes de acuerdo con el nombre | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Club | ordeningClientsNuPets (); | Debe ordenar los clientes de acuerdo con la cantidad de mascotas que tengan | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Club | ordeningClientLasName(); | Debe ordenar los clientes de acuerdo con el apellido | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Club | ordeningClientBirthDate (); | Debe ordenar los clientes de acuerdo con la fecha de nacimiento | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Club | ordeningClientsFavTypePet (); | Debe ordenar los clientes de acuerdo con el tipo de mascota preferida | ---------------------- | Es verdadero, puesto que ordeno correctamente |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Client | ordenatioId(); | Debe de ordenar la lista de mascotas de acuerdo con el id | ------------------- | Es verdadero,  Puesto lo ordeno correctamente |
| Client | ordenationName(); | Debe ordenar las mascotas de acuerdo con el nombre | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Client | ordenationType(); | Debe ordenar las mascotas de acuerdo con el tipo de mascota que es | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Client | ordenationGender(); | Debe ordenar las mascotas de acuerdo con el geneto | ---------------------- | Es verdadero, puesto que ordeno correctamente |
| Client | ordenationDate (); | Debe ordenar las mascotas de acuerdo con la fecha de nacimiento | ---------------------- | Es verdadero, puesto que ordeno correctamente |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Pet | compareTo(); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero,  Puesto que dice que es menor |
| Pet | comparePetName(); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero, puesto que dijo que era mayor |
| Pet | comparePetName (); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero, puesto que las dos mascotas son iguales |
| Pet | comparePetBirth (); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero, puesto que la primera mascota es menor |
| Pet | compare(); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero, puesto que dijo que eran iguales |
| Pet | comparePetType(); | Debe decir cual mascota es mayor para ordenarla | --------------- | Es verdadero, puesto que es mayor |

Requerimientos

|  |  |
| --- | --- |
| RF1 |  |
| RESUMEN | Este requerimiento permite registrar un club en la información guardada |
| ENTRADA | * Identificación * Nombre * Fecha de creación * Tipo de mascotas |
| SALIDA | El club fue registrado correctamente |

|  |  |
| --- | --- |
| RF2 |  |
| RESUMEN | Este requerimiento permite registrar un dueño en la información guardada (no deben existir dos dueños con la misma identificación) |
| ENTRADA | * Identificación * Nombre * Apellido * Fecha nacimiento * Mascota preferida |
| SALIDA | El dueño fue registrado correctamente. |

|  |  |
| --- | --- |
| RF3 |  |
| RESUMEN | Este requerimiento permite registrar una mascota en la información guardada (ninguna mascota del mismo dueño podrá llamarse igual) |
| ENTRADA | * Identificación * Nombre * Fecha nacimiento * Genero * Tipo de mascota |
| SALIDA | La mascota fue registrada correctamente |

|  |  |
| --- | --- |
| RF4 |  |
| RESUMEN | Este requerimiento permite almacenar la información de los clubes en un archivo plano |
| ENTRADA | <ninguna> |
| SALIDA | La información se ha registrado correctamente. |

|  |  |
| --- | --- |
| RF5 |  |
| RESUMEN | Este requerimiento permite generar un listado ordenado de los clubes, dueños y mascotas por cualquiera de los campos solicitados por el usuario. |
| ENTRADA | * Campo solicitado por el usuario |
| SALIDA | El listado ordenado de los clubes, dueños y mascotas por el campo solicitado del usuario |

|  |  |
| --- | --- |
| RF6 |  |
| RESUMEN | Este requerimiento permite generar un listado ordenado de los dueños según el número de mascotas que cada de estos tiene. |
| ENTRADA | <ninguna> |
| SALIDA | El listado de los dueños ordenado correctamente. |

|  |  |
| --- | --- |
| RF7 |  |
| RESUMEN | Este requerimiento permite generar un listado ordenado de los clubes según el número de clientes. |
| ENTRADA |  |
| SALIDA | El listado de los clubes ordenado correctamente. |

|  |  |
| --- | --- |
| RF8 |  |
| RESUMEN | Debe poder buscar por cualquiera de los criterios de los campos solicitados (usando los métodos tradicional y binario) y adicional debe poder ver en la pantalla la comparación de tiempos que gasta el sistema en realizar la búsqueda por los dos métodos. |
| ENTRADA | * Campo solicitado por el usuario |
| SALIDA | La información buscada por el usuario y la comparación de los tiempos que gasta el sistema en realizar las dos formas de búsqueda. |

|  |  |
| --- | --- |
| RF9 |  |
| RESUMEN | Este requerimiento permite verificar que el dueño exista para que una mascota sea registrada a él . |
| ENTRADA | * Id del cliente. |
| SALIDA | La verificación se realizó correctamente |

|  |  |
| --- | --- |
| RF10 |  |
| RESUMEN | Este requerimiento funcional deberá eliminar a un club, dueño o mascota por medio del número de identificación o el nombre. |
| ENTRADA | * Número de identificación o el nombre |
| SALIDA | El club, dueño o mascota se ha eliminado correctamente |

**REQUERIMIENTOS NO FUNCIONALES**

|  |  |
| --- | --- |
| RNF1 |  |
| RESUMEN | El programa debe de utilizar los tres métodos de ordenamiento clásico: burbuja, selección e inserción para las diferentes funciones establecidas. |

|  |  |
| --- | --- |
| RNF2 |  |
| RESUMEN | El programa debe implementar las dos interfaces: comparable y comparator en las diferentes funciones establecidas. Este también debe de utilizarse en el ordenamiento de los Arrays. |