

Especificacion de requerimientos funcionales

|         |   |
|---------|---|
| Nombre  | RF1: Registrar los datos del club   |
| Resumen | Permite registrar Clubs de manera manual o agregandolos de una fuente externa |
| Entrada | Id del club, Nombre del club, Mascotas permitidas en el club                  |
| Salida  | Registra el club  |

|         |  |
|---------|--|
| Nombre  | RF2: Registrar los datos del dueño de una mascota  |
| Resumen | Permite registrar los datos de un dueño de una mascota, ademas, como el dueño es nuevo debe permitir registrarle una mascota |
| Entrada | Id del dueño, Nombre del dueño, apellido del dueño, fecha de nacimiento  |
| Salida  | Si los valores ingresados se salen de los parametros envia un mensaje de alerta  |

|         |  |
|---------|--|
| Nombre  | RF3: Registrar las mascotas  |
| Resumen | Permite registrar mascotas a un club y a un dueño deseado por el usuario, ademas se pueden agregar datos de manera externa al programa.              |
| Entrada | Id del cliente, id de la mascota, nombre de la mascota, dia, mes y año de nacimiento de la mascota, genero de la mascota, ingrese el tipo de mascota |
| Salida  | Registra mascotas  |

|         |   |
|---------|---|
| Nombre  | RF4: Generar listados ordenados de los clubes por cualquier criterio  |
| Resumen | Ordena los clubes por orden natural, por el criterio que el usuario escoja                                  |
| Entrada | N/A   |
| Salida  | Muestra el listado de los valores de los atributos de los clubs en el orden natural escogido por el usuario |

|         |  |
|---------|--|
| Nombre  | RF5: Generar listados ordenados de los dueños por cualquier criterio   |
| Resumen | Ordena el listado de los dueños por orden natural, por el criterio escogidos por el usuario                  |
| Entrada | N/A  |
| Salida  | Muestra el listado de los valores de los atributos de los dueños en el orden natural escogido por el usuario |

|         |  |
|---------|--|
| Nombre  | RF6: Generar listados ordenados de las mascotas por cualquier criterio   |
| Resumen | Ordena el listado de las mascotas por orden natural, por el criterio escogidos por el usuario                  |
| Entrada | N/A  |
| Salida  | Muestra el listado de los valores de los atributos de las mascotas en el orden natural escogido por el usuario |

|         |   |
|---------|---|
| Nombre  | RF7: Generar listado ordenado de dueños según el número de mascotas                                     |
| Resumen | Ordena los dueños por la cantidad de mascotas que tanga cada dueño de menor a mayor dueño con mascotas. |
| Entrada | N/A   |
| Salida  | Valores de los atributos de las lista de los dueños ordenados como deseo el usuario                     |

|         |   |
|---------|---|
| Nombre  | RF8: Generar listado de los clubes según el numero de dueños                                      |
| Resumen | Ordena los clubes por la cantidad de dueños que tanga cada club de menor a mayor club con dueños. |
| Entrada | N/A   |
| Salida  | Valores de los atributos de las lista de los clubes ordenados como deseo el usuario               |

|         |   |
|---------|---|
| Nombre  | RF9: Verificar que no existan dueños con l misma identificación   |
| Resumen | Verifica que a la hora de agregar un dueño no exista un dueño con una misma ID en un mismo club                           |
| Entrada | Id del dueño de la mascota  |
| Salida  | Si el id ya existe arroja un mensaje de alerta anunciando de que ya existe un usuario con esa id y que debe ingresar otra |

|         |  |
|---------|--|
| Nombre  | RF10: Verificar que no existan mascotas con el mismo nombre para un dueño  |
| Resumen | Verifica que a la hora de agregar mascotas a un dueño no existan mas mascotyas con el mismo nombre para el mismo dueño |
| Entrada | id del dueño al cual se le quiere agregar la mascota y nombre de la mascota para verificar que no exista               |

|         |  |
|---------|--|
| Nombre  | RF11: Eliminar un club con el número de identificación   |
| Resumen | Permite eliminar un club del listado de clubs con el numero de identificacion del club que el usuario desea eliminar |
| Entrada | Id del club del cual el usuario quiere eliminar  |

|         |  |
|---------|--|
| Nombre  | RF12: Eliminar un club con el nombre   |
| Resumen | Permite eliminar un club del listado de clubs con el nombre del club que el usuario desea eliminar |
| Entrada | Nombre del club que desea eliminar   |

|        |   |
|--------|---|
|        |   |
| Salida | Si la mascota tiene el mismo nombre, envia un mensaje de alerta al usuario de que ya existe una mascota con el mismo nombre |

|        |  |
|--------|--|
|        |  |
| Salida | Elimina el club deseado por el usuario |

|        |  |
|--------|--|
|        |  |
| Salida | Elimina el club deseado por el usuario |

|         |  |
|---------|--|
| Nombre  | RF13: Eliminar un dueño con el número de identificación  |
| Resumen | Elimina el dueño de una mascota con toda su informacion con el id de esta pero antes ingresando datos que permitan saber en que club se encuentra el dueño |
| Entrada | Id del club en cual se encuentra el dueño e id del dueño al cual se quiere eliminar  |
| Salida  | Elimina el dueño deseado por el usuario  |

|         |   |
|---------|---|
| Nombre  | RF14: Eliminar un dueño con el nombre   |
| Resumen | Elimina el dueño de una mascota con toda su informacion con el nombre de esta, pero antes ingresando datos que permitan saber en que club se encuentra el dueño |
| Entrada | id del club en cual se encuentra el dueño, nombre del dueño que se quiere eliminar  |
| Salida  | Elimina el dueño deseado por el usuario   |

|         |   |
|---------|---|
| Nombre  | RF15: Eliminar un animal con su identificación  |
| Resumen | Elimina una mascota de un dueño con su id, para esto necesita saber en que club se encuentra el dueño de esa mascota. |
| Entrada | Id del club, id del dueño y id de la mascota que se quiere eliminar   |
| Salida  | Elimina la mascota deseada por el usuario si no se encuentra el mensaje envia una alerta de error                     |

|         |   |
|---------|---|
| Nombre  | RF16: Eliminar un animal con su nombre  |
| Resumen | Elimina una mascota con el nombre de un dueño, para esto necesita saber en que club se encuentra el dueño de esa mascota. |
| Entrada | Id del club, id del dueño y nombre de la mascota que se quiere eliminar   |
| Salida  | Elimina la mascota deseada por el usuario si no se encuentra el mensaje envia una alerta de error                         |

|         |  |
|---------|--|
| Nombre  | RNF1: Realizar búsqueda de cualquier campo de manera tradicional:  |
| Resumen | Realiza una busqueda del campo requerido que lo tiene un objeto, comenzando desde la primera posición del arreglo del objeto al cual se le quiere buscar el campo, terminando hasta la posición en el cual encuentra el campo buscado, si el campo no es encontrado llegara hasta el final y lanzara una excepción |
| Entrada | Campo del objeto que se quiere buscar  |
| Salida  | Objeto que contenga el campo que el usuario deseaba  |

|         |   |
|---------|---|
| Nombre  | RNF2: Realizar búsqueda de cualquier campo de manera binaria  |
| Resumen | Realiza la busqueda de un campo que contenga un objeto que se encuentra en la lista, comienza desde la mitad de la lista y cmienza a validar si este se encontraba en la mitad o si esta a laizquierda o derecha de la lista despues de que defina si esta a la izquierda o derecha realiza el procedimiento anterior otra vez hasta que encuentre el objeto con el campo requerdo si no lo encuentra lanzara una excepcion |
| Entrada | Campo del objeto que se quiere buscar   |
| Salida  | Objeto que contenga el heeue que es usuario le solicito   |

|         |  |
|---------|--|
| Nombre  | RFN3: Mostrar en pantalla la comparativa de tiempos que gasta en realizar una búsqueda tradicional |
| Resumen | Permite calcular los tiempos en nano segundos el tiempo que tarda la busqueda tradicional          |

|         |   |
|---------|---|
| Nombre  | RFN4 Mostrar en pantalla la comparativa de tiempos que gasta en realizar una búsqueda binaria |
| Resumen | tiempo en nanosegundos que tarda en hacer la busqueda binaria                                 |

|         |  |
|---------|--|
| Nombre  | RFN 5: Implementar y utilizar los tres métodos de ordenamiento clásicos: burbuja, selección e inserción                |
| Resumen | Implementa los tres metodos de ordenamiento clasicos, los cuales ordenan los objetos o valores que el usuario solicite |

|         |   |
|---------|---|
| Entrada | N/A   |
| Salida  | Tiempo total que tardo en hacer la busqueda tradicional |

|         |   |
|---------|---|
| Entrada | N/A   |
| Salida  | Tiempo total que tarda en hacer la busqueda binaria |

|         |  |
|---------|--|
| Entrada | N/A  |
| Salida  | Lista ordenada qpor el campo que el usuario solicito |

|         |  |
|---------|--|
| Nombre  | RNF6:Implementar y utilizar las dos estrategias de búsqueda clásicas: secuencial y binario   |
| Resumen | Implementa los dos metodos de busqueda clasica, los cuales buscan los objetos que el usuario quiera por un campo que el usuario quiere |
| Entrada | Atributo o campo por el cual quiere buscar un objeto   |
| Salida  | Objeto que contiene el campo por el cual el usuario decidio buscar el objeto   |

|         |   |
|---------|---|
| Nombre  | RNF7 Utilizar la interface Comparable                       |
| Resumen | Permite hacer comparaciones por un orden que no sea natural |
| Entrada | Dos objetos   |
| Salida  | Compara los dos objetos                                     |

|         |  |
|---------|--|
| Nombre  | RNF 8 La información de los clubes debe ser almacenado en archivo plano            |
| Resumen | Permite almacenar la informacion de todos los clubes en un archivo de formato .TXT |
| Entrada | Atributos del club que se quiere almacenar   |
| Salida  | El archivo plano se guarda   |

|         |   |
|---------|---|
| Nombre  | RNF 9 Las mascotas deben ser almacenadas en archivos serializables  |
| Resumen | Almacena a los dueños de las mascotas en un archivo serializable que a la vez serializa las mascotas que tiene cada dueño |
| Entrada | Atributos del owner y la mascota de ese owner que se quiere serializar  |
| Salida  | El archivo se serializa   |

| Requerimiento funcional  | Metodo Utilizado   | Clase   |
|--|--|---|
| RF1: Registrar los datos del club                                      | 1 //addClub():String   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF2: Registrar los datos del dueño de una mascota                      | 1//searchClubToAdOwner():void<br>1//createOwner():String<br>2//getld():String getOwner():String  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF3: Registrar las mascotas  | 1//addPetToAnOwner()String<br>2//searchOwnerToAddApet():void<br>3//addPetToAnOwner():void  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF4: Generar listados ordenados de los clubes por cualquier criterio   | 1//organizeWithld():void<br>1//ShowArrayList():String<br>2//compareClubWithld():void<br>1//organizeWithName():void<br>1//ShowArrayList():String<br>2//compareClubWithDate():void<br>1//organizeWithld():void<br>1//ShowArrayList():String<br>2//compareClubWithDate():void<br>1//organizeWithld():void<br>1//ShowArrayList():String<br>2//compareClubWithld():void                                     | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF5: Generar listados ordenados de los dueños por cualquier criterio   | 1//organizeOwnerWithld():void<br>1//ShowArrayList():String<br>2//compareOwnerWithld():void<br>1//organizeOwnersWithName():void<br>1//ShowArrayList():String<br>2//compareClubWithDate():void<br>1//organizeOwnersWithLastName():void<br>1//ShowArrayList():String<br>2//compareClubWithDate():void<br>1//organizeOwnersWithbornDate():void<br>1//ShowArrayList():String<br>2//compareClubWithld():void | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF6: Generar listados ordenados de las mascotas por cualquier criterio | 3//organizeWithld():void<br>3//organizeWithName():void<br>3//organizeWithBornDate():void<br>3//organizeWithType():void<br>3//organizeWithGender():void   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF7: Generar listado ordenado de dueños según el número de mascotas    | 1//organizeOwnersByNumberOfPet():void<br>2//organizeOwnersByNumberOfPet():void<br>3//numberOfPets():void<br>1//ShowArrayListOwners():String  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF8: Generar listado de los clubes según el numero de dueños           | 1//organizeClubByNumberOfOwners():void<br>2//numberOfOwners():void<br>1//ShowArrayList():void  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF9: Verificar que no existan dueños con la misma identificación       | 1//askForlds():String<br>2//askForlds():String   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |

|  |  |   |
|--|--|---|
| RF10: Verificar que no existan mascotas con el mismo nombre para un dueño                          | 1//askForIds():String<br>2//askForIds():String<br>3//askForIds():String  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF11: Eliminar un club con el número de identificación   | 1//deleteClubWithId():String   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF12: Eliminar un club con el nombre   | 1//deleteClubWithName():String   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF13: Eliminar un dueño con el número de identificación  | 1//searchClubToDelete():String<br>2//searchOwnerToDeleteWithId():String  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF14: Eliminar un dueño con el nombre:   | 1//searchApetWithClubOwnerAndPetName():String<br>2//searchApetWithClubOwnerAndPetName():String<br>3//searchApetWithClubOwnerAndPetId():String<br>4//comparePetWithId():int | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF15: Eliminar un animal con su identificación   | 1//searchApetWithClubOwnerAndPetId():String<br>2//searchApetWithClubOwnerAndPetId():String<br>3//searchApetWithClubOwnerAndPetId():String<br>4//comparePetWithId():String  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RF16: Eliminar un animal con su nombre   | 1//searchApetWithClubOwnerAndPetName():String<br>2//searchApetWithClubOwnerAndPetName():String<br>3//searchApetWithClubOwnerAndPetName():String                            | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RNF1: Realizar búsqueda de cualquier campo de manera tradicional:                                  | 1//traitionalSearchWithId();   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RNF2: Realizar búsqueda de cualquier campo de manera binaria:                                      | 1//binarySearchingById();<br>2//compareClubWithId();   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RFN3: Mostrar en pantalla la comparativa de tiempos que gasta en realizar una búsqueda tradicional | 0//System.nanoTime();  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RFN4 Mostrar en pantalla la comparativa de tiempos que gasta en realizar una búsqueda binaria      | 0//System.nanoTime();  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |

|   |   |   |
|---|---|---|
| RFN 5: Implementar y utilizar los tres métodos de ordenamiento clásicos: burbuja, selección e inserción | Por efectos practicos, los diferentes tipos de ordenamiento se encuentran principalmente en 1 y 2 | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RNF6:Implementar y utilizar las dos estrategias de búsqueda clásicas: secuencial y binario              | Por efectos practicos, los diferentes tipos de ordenamiento se encuentran principalmente en 1 y 2 | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RFN7 Utilizar el método de ordenamiento de la clase Arrays utilizando Comparable.                       |   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RNF 8 La información de los clubes debe ser almacenado en archivo plano                                 | 1//setUpClub():void;  | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |
| RNF 9 Las mascotas deben ser almacenadas en archivos serializables                                      | 1//adderOfOwners():void;<br>1//adderOfOwners1():void;   | 0. Main<br>1. Holding<br>2. Club<br>3. OwnerOfPet<br>4. Pet |