

Nombre: Cristian Camilo Rivadeneira

Código: A00354996

1. Requerimientos Funcionales

- **Requerimiento Funcional R1.**

Nombre	Registrar un Club en el holding de Clubes.
Resumen	Permite que agregar un club a la lista de clubes que administra el sistema.
Entrada	Datos generales de cada club: Su identificación, su nombre asignado, una fecha de creación y el tipo de mascotas que admite el club.
Salida	Se agrega un nuevo Club a holding de Clubes.

- **Requerimiento Funcional R2.**

Nombre	Registrar un dueño en un Club
Resumen	Permite agregar un dueño a un club, siempre y cuando este no tenga la identificación de otro dueño ya existente dentro de la lista de dueños del club al que se desea agregar
Entrada	Dato necesarios para acceder al club: La identificación del Club Datos generales de los dueños: Su identificación, su nombre, su apellido, su fecha de Nacimiento y su tipo de mascota preferida
Salida	Se agrega un nuevo dueño a un club

- **Requerimiento Funcional R3.**

Nombre	Agregar una mascota a un dueño.
Resumen	Permite agregar una mascota a un dueño, siempre y cuando este no tenga registrada una mascota con el mismo nombre en su lista de mascotas.
Entrada	Dato necesarios para acceder al club: La identificación del Club Dato necesarios para acceder al dueño: La identificación del Dueño Datos generales de las mascotas: Su identificación, su nombre, su fecha de nacimiento, su género y su especie
Salida	Se agrega una nueva mascota a un dueño.

- **Requerimiento Funcional R4.**

Nombre	Generar listado ordenado de los clubes registrados en el holding
Resumen	Permite mostrar los clubes de manera ordenada (de menor a mayor lexicográficamente) en una lista según el criterio que el usuario elija. Los criterios por los cuales esto puede ser llevado a cabo son los siguientes: <ul style="list-style-type: none">• Por Identification.• Por Nombre.• Por fecha de creación.• Por tipo de mascota permitida.
Entrada	
Salida	Listado ordenado de los clubes según el criterio solicitado.

- **Requerimiento Funcional R5**

Nombre	Generar listado ordenado de los dueños de un Club.
Resumen	<p>Permite mostrar los dueños de un club de manera ordenada (de menor a mayor lexicográficamente) en una lista según el criterio que el usuario elija.</p> <p>Los criterios por los cuales esto puede ser llevado a cabo son los siguientes:</p> <ul style="list-style-type: none"> • Por Identification. • Por Nombre. • Por Apellido. • Por Fecha de nacimiento. • Por Tipo de mascota preferida.
Entrada	Dato necesarios para acceder al club: La identificación del Club
Salida	Listado ordenado de los dueños según el criterio solicitado.

- **Requerimiento Funcional R6.**

Nombre	Generar listado ordenado de las mascotas de un dueño.
Resumen	<p>Permite mostrar la mascotas de un dueño de manera ordenada (menor a mayor lexicográficamente) en una lista según el criterio que el usuario elija</p> <p>Los criterios por los cuales esto puede ser llevado a cabo son los siguientes:</p> <ul style="list-style-type: none"> • Por Identification. • Por Nombre de la mascota. • Por Fecha de nacimiento de la mascota. • Por Especie.
Entrada	<p>Dato necesarios para acceder al club: La identificación del Club</p> <p>Dato necesarios para acceder al dueño: La identificación del Dueño.</p>
Salida	Listado ordenado de las mascotas según el criterio seleccionado.

- **Requerimiento Funcional R7.**

Nombre	Generar listado ordenado de los clubes según el número de dueños.
Resumen	Permite mostrar los clubes de manera (de menor a mayor lexicográficamente) en lista de acuerdo con el número de dueños que tenga cada club.
Entrada	
Salida	Listado ordenado de acuerdo con la cantidad de dueños en cada uno de los clubes.

•Requerimiento Funcional R8.

Nombre	Generar listado ordenado de los dueños según el número de mascotas
Resumen	Permite mostrar los dueños de un club de manera ordenada (de menor a mayor lexicográficamente) de acuerdo con el número de mascotas que tenga cada dueño.
Entrada	Dato necesarios para acceder al club: La identificación del Club
Salida	Listado ordenado de acuerdo con la cantidad de mascotas que tenga cada dueño.

• Requerimiento Funcional R9.

Nombre	Buscar un club por medio del método de búsqueda secuencial.
Resumen	<p>Permite encontrar un Club que este registrado en el holding por medio del método de <i>búsqueda secuencial</i> y además muestra el tiempo que tardo en ejecutarse en el sistema.</p> <p>Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son:</p> <ul style="list-style-type: none"> • Por Identification. • Por Nombre. • Por fecha de creación. • Por tipo de mascota permitida.
Entrada	
Salida	Un mensaje con la información del club que se buscó y el tiempo que tardo en ejecutarse la instrucción.

• Requerimiento Funcional R10.

Nombre	Buscar un club por medio del método de búsqueda binaria.
Resumen	<p>Permite encontrar un Club que este registrado en el holding por medio del método de <i>búsqueda binaria</i> y además muestra el tiempo que tardo en ejecutarse en el sistema.</p> <p>Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son:</p> <ul style="list-style-type: none"> • Por Identification. • Por Nombre. • Por fecha de creación. • Por tipo de mascota permitida.
Entrada	
Salida	Un mensaje con la información del club que se buscó y el tiempo que tardo en ejecutarse la instrucción.

• Requerimiento Funcional R11.

Nombre	Buscar un dueño de un club por medio del método de búsqueda tradicional.
Resumen	Permite encontrar un dueño que este registrado en un club del holding por medio del método de <i>búsqueda tradicional</i> y además muestra el tiempo que tardo en ejecutarse en el sistema.

	Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son: <ul style="list-style-type: none"> • Por Identification. • Por Nombre. • Por Apellido • Por fecha de nacimiento. • Por tipo de mascota preferida.
Entrada	Dato necesarios para acceder al club: La identificación del Club
Salida	Un mensaje con la información del dueño que se buscó y el tiempo que tardo en ejecutarse la instrucción.

• Requerimiento Funcional R12.

Nombre	Buscar un dueño de un club por medio del método de búsqueda binaria.
Resumen	Permite encontrar un dueño que este registrado en un club del holding por medio del método de <i>búsqueda binaria</i> y además muestra el tiempo que tardo en ejecutarse en el sistema. Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son: <ul style="list-style-type: none"> • Por Identification. • Por Nombre. • Por Apellido • Por fecha de nacimiento. • Por tipo de mascota preferida.
Entrada	Dato necesarios para acceder al club: La identificación del Club
Salida	Un mensaje con la información del dueño que se buscó y el tiempo que tardo en ejecutarse la instrucción.

• Requerimiento Funcional R13.

Nombre	Buscar una mascota de un dueño por medio del método de búsqueda secuencial.
Resumen	Permite encontrar una mascota de un dueño que este registrado en un club del holding por medio del método de <i>búsqueda tradicional</i> y además muestra el tiempo que tardo en ejecutarse en el sistema. Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son: <ul style="list-style-type: none"> • Por Identification. • Por Nombre de la mascota. • Por fecha de nacimiento de la mascota. • Por Especie.
Entrada	Dato necesarios para acceder al club: La identificación del Club Dato necesarios para acceder al dueño: La identificación del Dueño.
Salida	Un mensaje con la información del dueño que se buscó y el tiempo que tardo en ejecutarse la instrucción.

• Requerimiento Funcional R14.

Nombre	Buscar una mascota de un dueño por medio del método de búsqueda binaria.
Resumen	Permite encontrar una mascota de un dueño que este registrado en un club del holding por medio del método de <i>búsqueda binaria</i> y además muestra el tiempo que tardo en ejecutarse en el sistema.

	Esta búsqueda se podrá realizar por diferentes criterios de acuerdo con el que el usuario elija los cuales son: <ul style="list-style-type: none"> • Por Identification. • Por Nombre de la mascota. • Por fecha de nacimiento de la mascota. • Por Especie.
Entrada	Dato necesarios para acceder al club: La identificación del Club Dato necesarios para acceder al dueño: La identificación del Dueño.
Salida	Un mensaje con la información del dueño que se buscó y el tiempo que tardo en ejecutarse la instrucción.

- **Requerimiento Funcional R15.**

Nombre	Eliminar un Club por medio de su identificación.
Resumen	Permite eliminar un club registrado en el holding por medio de su identificación.
Entrada	La identificación del club a eliminar.
Salida	Se elimina el club del holding.

- **Requerimiento Funcional R16.**

Nombre	Eliminar un Club por medio de su nombre.
Resumen	Permite eliminar un club registrado en el holding por medio de su nombre.
Entrada	El nombre del club a eliminar.
Salida	Se elimina el club del holding.

- **Requerimiento Funcional R17.**

Nombre	Eliminar un Dueño por medio de su identificación.
Resumen	Permite eliminar un dueño registrado en un club del holding por medio de su identificación.
Entrada	Dato necesarios para acceder al club: La identificación del Club. La identification del club a eliminar.
Salida	Se elimina el dueño del club.

- **Requerimiento Funcional R18.**

Nombre	Eliminar un Dueño por medio de su nombre.
Resumen	Permite eliminar un dueño registrado en un club del holding por medio de su nombre.
Entrada	Dato necesarios para acceder al club: La identificación del Club. El nombre del club a eliminar.
Salida	Se elimina el dueño del club.

- **Requerimiento Funcional R19.**

Nombre	Eliminar una mascota por medio de su identificación.
Resumen	Permite eliminar una mascota de un dueño registrado en un club del holding por medio de su identificación.

Entrada	Dato necesarios para acceder al club: La identificación del Club. Dato necesarios para acceder al dueño: La identificación del Dueño. La identificación de la mascota a eliminar.
Salida	Se elimina la mascota de su dueño.

- Requerimiento Funcional R20.

Nombre	Eliminar una mascota por medio de su nombre.
Resumen	Permite eliminar una mascota de un dueño registrado en un club del holding por medio de su nombre.
Entrada	Dato necesarios para acceder al club: La identificación del Club. Dato necesarios para acceder al dueño: La identificación del Dueño. El nombre de la mascota a eliminar.
Salida	Se elimina la mascota de su dueño.

2. Requerimientos Funcionales

- Requerimiento No Funcional R1.

Requerimiento	La información de los clubes debe ser guardada en un archivo plano para su gestión.
Importancia	Permite que la información de los clubes se pueda visualizar en el archivo de texto. El control de información de grandes volúmenes de datos se hace mucho mejor cuando se tiene esta información almacenada para su visualización y acceso en los sistemas que la administran.

- Requerimiento No Funcional R2.

Requerimiento	La información de los dueños y las mascotas se debe almacenar en un archivo serializable.
Importancia	Permite que la información de objetos del mundo sea mucho mas fácil de guardar cuando se tratan de listas con listas anidadas. Para este caso gestionar desde un sistema la lista de dueños y sus mascotas es mas sencillo si se pueden deserializar como una sola línea de información binaria

- Requerimiento No Funcional R3.

Requerimiento	Usar los métodos de ordenamiento clásicos: burbuja, selección e inserción.
Resumen	Permite ordenar datos de clubes, dueños y mascotas de acuerdo con los algoritmos base de los métodos de ordenamiento clásicos.

- Requerimiento No Funcional R4.

Requerimiento	Usar las interfaces Comparable y Comparator para realizar comparaciones entre criterios
Resumen	Permite ordenar datos de clubes, dueños y mascotas de acuerdo con los algoritmos base de los métodos de ordenamiento clásicos.

- Requerimiento No Funcional R5.

Requerimiento	Usar las interfaces Comparable y Comparator para realizar comparaciones entre criterios
Resumen	Permite ordenar datos de clubes, dueños y mascotas de acuerdo con los algoritmos base de los métodos de ordenamiento clásicos.

- Trazabilidad

Requerimiento Funcional	Método y clase usado para su resolución
R1. Registrar un Club en el holding de Clubes.	<ul style="list-style-type: none"> • Método: registerClub () Clase: ClubAdministration • Método: saveClubs () Clase: ClubAdministration • Método: registerClub() Clase: Menu • Método: public static void main Clase: Menu
R2. Registrar un dueño en un Club	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: writeOwners() Clase: ClubAdministration • Método: giveOwner() Clase: Club • Método: registerOwner() Clase: Club • Método: registerOwner() Clase: Menu • Método: public static void main Clase: Menu
R3. Agregar una mascota a un dueño.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: givePet() Clase: Owner • Método: registerPet() Clase: Owner • Método: registerPet() Clase: Menu • Método: public static void main Clase: Menu
R4. Generar listado ordenado de los clubes registrados en el holding	<ul style="list-style-type: none"> • Método: orderClubsById() Clase: ClubAdministration • Método: orderClubsByClubName() Clase: ClubAdministration • Método: orderClubsByDate() Clase: ClubAdministration • Método: orderClubsByPet() Clase: ClubAdministration • Método: orderClubs() Clase: Menu • Método: public static void main Clase: Menu
R5. Generar listado ordenado de los dueños de un Club.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: orderOwnersById Clase: Club • Método: orderOwnersByName Clase: Club • Método: orderOwnersByDate Clase: Club • Método: orderOwnersByPet Clase: Club • Método: orderOwners() Clase: Menu • Método: public static void main Clase: Menu
R6. Generar listado ordenado de las mascotas de un dueño.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: orderPetsById() Clase: Owner • Método: orderPetsByName() Clase: Owner • Método: orderPetsByDate() Clase: Owner • Método: orderPetsByPetType() Clase: Owner • Método: orderPets() Clase: Menu • Método: public static void main Clase: Menu
R7. Generar listado ordenado de los clubes según el número de dueños.	<ul style="list-style-type: none"> • Método: numberOwners() Clase: Club • Método: orderClubsByNumberOwners() Clase: ClubAdministration • Método: orderByNumberOwners() Clase: Menu • Método: public static void main Clase: Menu

R8. Generar listado ordenado de los dueños según el número de mascotas	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: numberPets() Clase: Owner • Método: orderOwnersByNumberPets() Clase: Club • Método: orderByNumberPets() Clase: Menu • Método: public static void main Clase: Menu
R9. Buscar un club por medio del método de búsqueda secuencial.	<ul style="list-style-type: none"> • Método: secuencialSearchById() Clase: ClubAdministration • Método: secuencialSearchByClubName() Clase: ClubAdministration • Método: secuencialSearchByClubDate() Clase: ClubAdministration • Método: secuencialSearchByPet() Clase: ClubAdministration • Método: searchClubs() Clase: Menu • Método: public static void main Clase: Menu
R10. Buscar un club por medio del método de búsqueda binaria.	<ul style="list-style-type: none"> • Método: binarySearchById() Clase: ClubAdministration • Método: binarySearchByClubName() Clase: ClubAdministration • Método: binarySearchByClubDate() Clase: ClubAdministration • Método: binarySearchByPet() Clase: ClubAdministration • Método: searchClubs() Clase: Menu • Método: public static void main Clase: Menu
R11. Buscar un dueño de un club por medio del método de búsqueda tradicional.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: secuencialSearchById() Clase: Club • Método: secuencialSearchByName() Clase: Club • Método: secuencialSearchByLastName() Clase: Club • Método: secuencialSearchByOwnerDate() Clase: Club • Método: secuencialSearchByPet() Clase: Club • Método: searchOwners() Clase: Menu • Método: public static void main Clase: Menu
R12. Buscar un dueño de un club por medio del método de búsqueda binaria.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: binarySearchById() Clase: Club • Método: binarySearchByName() Clase: Club • Método: binarySearchByLastName() Clase: Club • Método: binarySearchByDate() Clase: Club • Método: binarySearchByPet() Clase: Club • Método: searchOwners() Clase: Menu • Método: public static void main Clase: Menu
R13. Buscar una mascota de un dueño por medio del método de búsqueda secuencial.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: secuencialSearchById() Clase: Owner • Método: secuencialSearchByName() Clase: Owner • Método: secuencialSearchByDate() Clase: Owner • Método: secuencialSearchByPet() Clase: Owner • Método: searchPets() Clase: Menu • Método: public static void main Clase: Menu
R14. Buscar una mascota de un dueño por medio del método de búsqueda binaria.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: binarySearchById() Clase: Owner • Método: binarySearchByName() Clase: Owner • Método: binarySearchByDate() Clase: Owner • Método: binarySearchByPet() Clase: Owner • Método: searchPets() Clase: Menu • Método: public static void main Clase: Menu
R15. Eliminar un Club por medio de su identificación.	<ul style="list-style-type: none"> • Método: deleteClubById() Clase: ClubAdministration • Método: deleteClub() Clase: Menu • Método: public static void main Clase: Menu

R16. Eliminar un Club por medio de su nombre.	<ul style="list-style-type: none"> • Método: deleteClubByName() Clase: ClubAdministration • Método: deleteClub() Clase: Menu • Método: public static void main Clase: Menu
R17. Eliminar un Dueño por medio de su identificación.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: deleteOwnerById() Clase: Club • Método: deleteOwner() Clase: Menu • Método: public static void main Clase: Menu
R18. Eliminar un Dueño por medio de su nombre.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: deleteOwnerByName() Clase: Club • Método: deleteOwner() Clase: Menu • Método: public static void main Clase: Menu
R19. Eliminar una mascota por medio de su identificación.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: deletePetById() Clase: Owner • Método: deletePet() Clase: Menu • Método: public static void main Clase: Menu
R20. Eliminar una mascota por medio de su nombre.	<ul style="list-style-type: none"> • Método: searchClub() Clase: ClubAdministration • Método: searchOwner() Clase: Club • Método: deletePetByName() Clase: Owner • Método: deletePet() Clase: Menu • Método: public static void main Clase: Menu

