

Mateo Valdés Otero

10 de septiembre de 2019

Algoritmos y Programación II

Laboratorio 2

1. Requerimientos funcionales:

RF1: Registrar un club

RF2: Registrar un dueño

RF3: Registrar una mascota

RF4: Mostrar una lista ordenada de los clubes

RF5: Mostrar una lista ordenada de los dueños

RF6: Mostrar una lista ordenada de las mascotas

RF7: Almacenar la información de los clubes, dueños y mascotas

RF8: Eliminar un club especificado

RF9: Eliminar un dueño especificado

RF10: Eliminar una mascota especificada

RF11: Buscar clubes

RF12: Buscar dueños

RF13: Buscar mascotas

RF14: Cargar la información de los clubes, dueños y mascotas

Especificación requerimientos funcionales:

Nombre	RF1: Registrar un club
Resumen	El sistema permite registrar un nuevo club
Entrada	Nombre, número de identificación, fecha de creación y tipo de mascotas del club
Salida	El club ha sido registrado exitosamente

Nombre	RF2: Registrar un dueño
Resumen	El sistema permite registrar un nuevo dueño de mascotas en un club
Entrada	Nombres, apellidos, número de identificación, fecha de nacimiento y tipo de mascota de preferencia

Salida	El dueño ha sido registrado exitosamente
--------	--

Nombre	RF3: Registrar una mascota
Resumen	El sistema permite registrar una mascota que le pertenece a un dueño
Entrada	Nombre, número de identificación, fecha de nacimiento, género y tipo de mascota
Salida	La mascota ha sido registrada exitosamente

Nombre	RF4: Mostrar una lista ordenada de los clubes
Resumen	El sistema muestra una lista ordenada de los clubes dado un criterio de ordenamiento. Los criterios de ordenamiento son: <ul style="list-style-type: none"> • Ordenar por el nombre • Ordenar por el ID • Ordenar por la fecha de creación • Ordenar por el tipo de mascota del club • Ordenar por el número de dueños
Entrada	El criterio de ordenamiento
Salida	La lista ordenada de los clubes

Nombre	RF5: Mostrar una lista ordenada de los dueños
Resumen	El sistema muestra una lista ordenada de los dueños dado un criterio de ordenamiento. Los criterios de ordenamiento son: <ul style="list-style-type: none"> • Ordenar por los nombres • Ordenar por los apellidos • Ordenar por el ID • Ordenar por la fecha de nacimiento • Ordenar por tipo de mascota preferido
Entrada	El criterio de ordenamiento
Salida	La lista ordenada de los dueños

Nombre	RF6: Mostrar una lista ordenada de las mascotas
Resumen	El sistema muestra una lista ordenada de las mascotas dado un criterio de ordenamiento. Los criterios de ordenamiento son: <ul style="list-style-type: none"> • Ordenar por el nombre • Ordenar por el ID • Ordenar por la fecha de nacimiento • Ordenar por el género • Ordenar por el tipo de mascota

Entrada	El criterio de ordenamiento
Salida	La lista ordenada de las mascotas

Nombre	RF7: Almacenar la información de los clubes, dueños y mascotas
Resumen	El sistema permite almacenar la información de los clubes, dueños y mascotas. Los clubes son almacenados en un archivo plano (de tipo .txt) y los dueños y las mascotas son almacenados en un archivo serializado (de tipo .se).
Entrada	-
Salida	Los archivos con la información de los clubes, dueños y mascotas

Nombre	RF8: Eliminar un club especificado
Resumen	El sistema permite eliminar un club especificado
Entrada	El ID del club que se desea eliminar
Salida	El club ha sido eliminado exitosamente

Nombre	RF9: Eliminar un dueño especificado
Resumen	El sistema permite eliminar un dueño especificado
Entrada	El ID del dueño que se desea eliminar
Salida	El dueño ha sido eliminado exitosamente

Nombre	RF10: Eliminar una mascota especificada
Resumen	El sistema permite eliminar una mascota especificado
Entrada	El nombre de la mascota que se desea eliminar
Salida	La mascota ha sido eliminada exitosamente

Nombre	RF11: Buscar clubes
Resumen	<p>El sistema permite buscar uno o más clubes a partir de un criterio de búsqueda y una palabra clave. Los criterios de búsqueda son:</p> <ul style="list-style-type: none"> • Buscar por nombre • Buscar por ID • Busca por fecha de creación • Buscar por tipo de mascota • Buscar por número de dueños

Entrada	El criterio de búsqueda y la palabra clave
Salida	Todos los clubes que apliquen

Nombre	RF12: Buscar dueños
Resumen	El sistema permite buscar uno o más dueños a partir de un criterio de búsqueda y una palabra clave. Los criterios de búsqueda son: <ul style="list-style-type: none"> • Buscar por nombres • Buscar por apellidos • Buscar por ID • Buscar por fecha de nacimiento • Busca por tipo preferido de mascota
Entrada	El criterio de búsqueda y la palabra clave
Salida	Todos los clubes que apliquen

Nombre	RF13: Buscar mascotas
Resumen	El sistema permite buscar uno o más mascotas a partir de un criterio de búsqueda y una palabra clave. Los criterios de búsqueda son: <ul style="list-style-type: none"> • Buscar por nombre • Buscar por ID • Buscar por fecha de nacimiento • Buscar por género • Buscar por tipo de mascota
Entrada	El criterio de búsqueda y la palabra clave
Salida	Todos los clubes que apliquen

Nombre	RF14: Cargar la información previamente almacenada de los clubes, dueños y mascotas
Resumen	El sistema permite cargar la información previamente almacenada de los clubes, dueños y mascotas. La información de los clubes se carga de un archivo plano (de tipo .txt) y la información de los dueños y las mascotas se carga de un archivo serializado (de tipo .se)
Entrada	Los archivos planos/serializados que contienen la información de los clubes, los dueños y las mascotas
Salida	La información ha sido cargada exitosamente

Requerimientos no funcionales:

- El programa debe estar escrito en el lenguaje de programación Java
- Se deben hacer pruebas en JUnit para asegurarse de que el programa

- Se deben hacer generar varios datos aleatorios para evaluar la funcionalidad del programa
- Se debe usar los interfaces Comparable y Comparator del lenguaje de programación Java
- Se debe usar algoritmos de búsqueda secuenciales y binarios

2. Diagrama de clases:

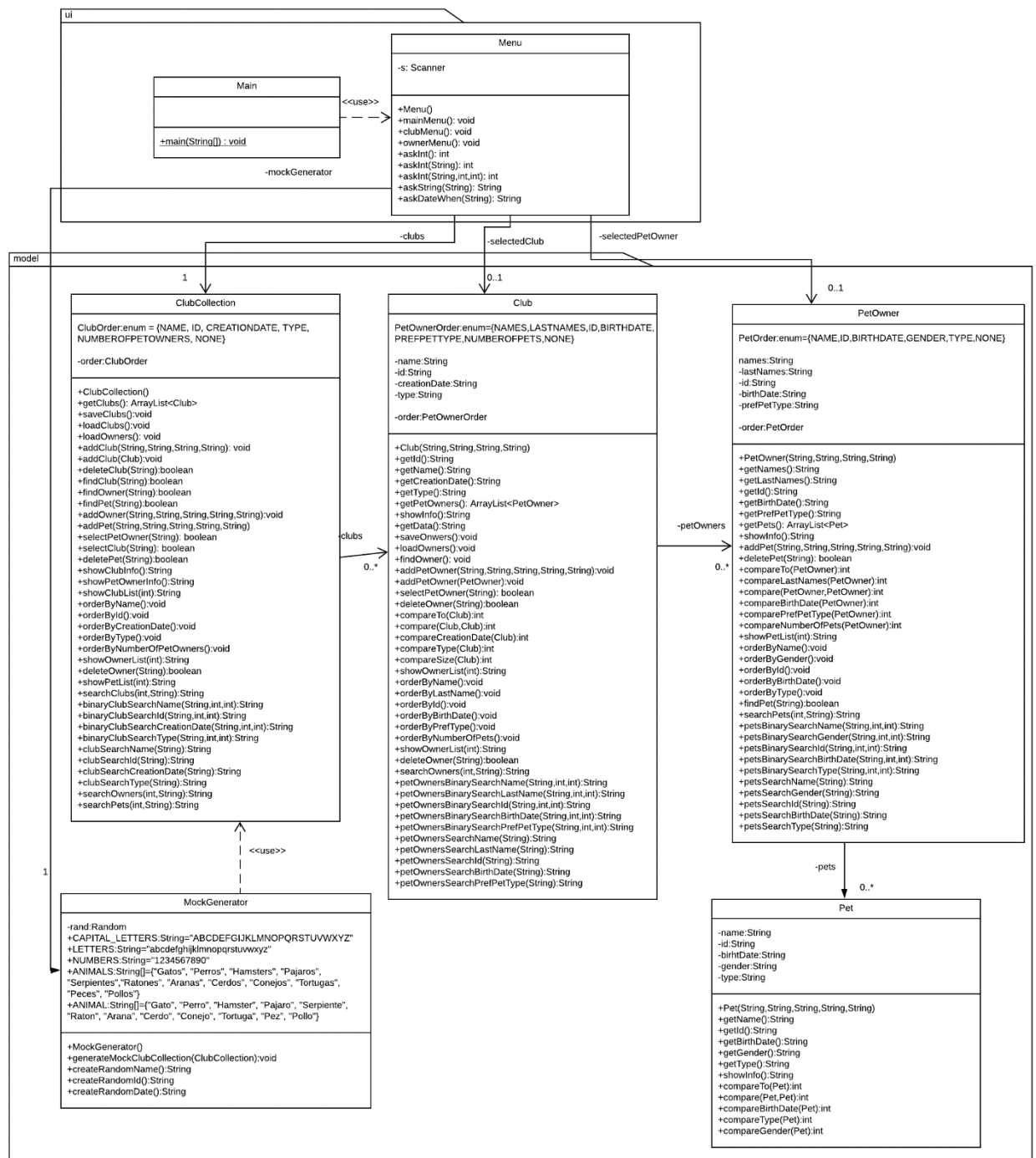
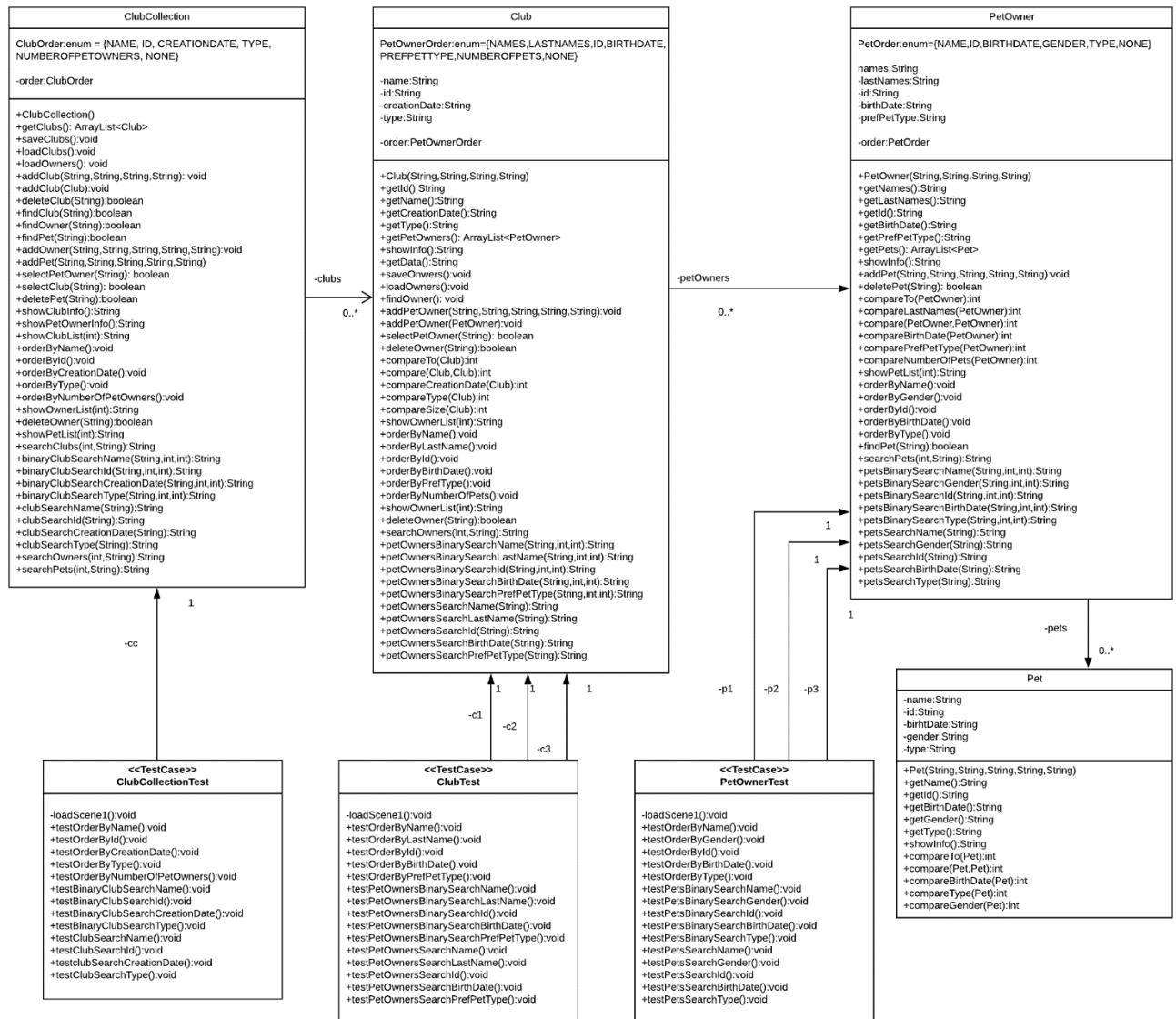


Diagrama de las pruebas:



3. Link al repositorio de github:

<https://github.com/mavaldot/Clubs>

NOTA: Por favor leer README.txt

4. Trazabilidad:

RF1: Registrar un club

Clase	Método
ClubCollection	addClub()

RF2: Registrar un dueño

Clase	Método
ClubCollection	addOwner()
Club	addPetOwner()

RF3: Registrar una mascota

Clase	Método
ClubCollection	addPet()
Owner	addPet()

RF4: Mostrar una lista ordenada de los clubes

Clase	Método
ClubCollection	showClubList()
	orderByName()
	orderById()
	orderByCreationDate()
	orderByType()
	orderByNumberOfPetOwners()
Clubs	showInfo()
	compareTo()
	compare()
	compareCreationDate()
	compareType()

RF5: Mostrar una lista ordenada de los dueños

Clase	Método
ClubCollection	showOwnerList()
Club	showOwnerList()
	orderByName()
	orderByLastName()
	orderById()
	orderByBirthDate()
	orderByPrefPetType()
	orderByNumberOfPets()
PetOwner	showInfo()
	compareTo()
	compareLastNames()
	compare()
	compareBirthDate()
	comparePrefPetType()

	compareNumberOfPets()
--	-----------------------

RF6: Mostrar una lista ordenada de las mascotas

Clase	Método
ClubCollection	showPetList()
PetOwner	showPetList()
	orderByNome()
	orderByld()
	orderByBirthDate()
	orderByGender()
	orderByType()
Pet	showInfo()
	compareTo()
	compare()
	compareBirthDate()
	compareGender()
	compareType()

RF7: Almacenar la información de los clubes, dueños y mascotas

Clase	Método
ClubCollection	saveClubs()
Club	saveOwners()
	getData()

RF8: Eliminar un club especificado

Clase	Método
ClubCollection	deleteClub()

RF9: Eliminar un dueño especificado

Clase	Método
ClubCollection	deleteOwner()
Club	deleteOwner()

RF10: Eliminar una mascota especificada

Clase	Método
ClubCollection	deletePet()
PetOwner	deletePet()

RF11: Buscar clubes

Clase	Método
ClubCollection	searchClubs()
	orderByName()
	orderById()
	orderByCreationDate()
	orderByType()
	binaryClubSearchName()
	clubSearchName()
	binaryClubSearchId()
	clubSearchId()
	binaryClubSearchCreationDate()
	clubSearchCreationDate()
	binaryClubSearchType()
	clubSearchType()
Club	showInfo()
	compareTo()
	compare()
	compareCreationDate()
	compareType()

RF12: Buscar dueños

Clase	Método
ClubCollection	searchOwners()
Club	searchOwners()
	orderByName()
	orderByLastName()
	orderById()
	orderByBirthDate()
	orderByPrefPetType()
	petOwnersBinarySearchName()
	petOwnersSearchName()
	petOwnersBinarySearchLastName()
	petOwnersSearchLastName()
	petOwnersBinarySearchId()
	petOwnersSearchId()
	petOwnersBinarySearchBirthDate()
	petOwnersSearchBirthDate()
	petOwnersBinarySearchPrefPetType()
	petOwnersSearchPetType()
PetOwner	showInfo()
	compareTo()
	compareLastNames()
	compare()
	compareBirthDate()
	comparePrefPetType()

	compareNumberOfPets()
--	-----------------------

RF13: Buscar mascotas

Clase	Método
ClubCollection	searchPets()
PetOwner	searchPets()
	orderByNome()
	orderByld()
	orderByBirthDate()
	orderByGender()
	orderByType()
	petsBinarySearchName()
	petSearchName()
	petsBinarySearchld()
	petSearchld()
	petsBinarySearchBirthDate()
	petSearchBirthDate()
	petsBinarySearchGender()
	petSearchGender()
	petsBinarySearchType()
	petSearchType()
Pet	showInfo()
	compareTo()
	compare()
	compareBirthDate()
	compareGender()
	compareType()

RF14: Cargar la información previamente almacenada de los clubes, dueños y mascotas

Clase	Método
ClubCollection	loadClubs()
	loadOwners()
Club	loadOwners()

5. Diseño de las pruebas:

Clase: ClubCollection

Escenarios:

loadScene1(): Se crea un escenario con un Club Collection. Al Club Collection se le agregan 3 clubes de la siguiente forma:

- Club #1: Nombre: Johan Club, ID: 12345678, Fecha de creación: 2000/05/05, Tipo de mascotas: Perros
- Club #2: Nombre: Ariza Club, ID: 12345679, Fecha de creación: 1993/09/05, Tipo de mascotas: Gatos
- Club #3: Nombre: Valdés Club, ID: 55554444, Fecha de creación: 1956/10/06, Tipo de mascotas: Pájaros

Al Club #1 se le agregan los siguientes miembros:

- Miembro #1: Nombre: Juan David, Apellido: Ossa, ID: 1234, Fecha de nacimiento: 2000/01/01, Tipo preferido de mascota: Perros
- Miembro #2: Nombre: Esteban, Apellido: Ariza, ID: 4445, Fecha de nacimiento: 2005/02/04, Tipo preferido de mascotas: Gatos
- Miembro #3: Nombre: Johan, Apellido: Giraldo, ID: 5656, Fecha de nacimiento: 2004/06/24, Tipo preferido de mascota: Serpiente

Al Club #2 se le agregan los siguientes miembros:

- Miembro #1: Nombre: David Jhun, Apellido: Kim, ID: 4321, Fecha de nacimiento: 1998/06/04, Tipo preferido de mascota: Perros
- Miembro #2: Nombre: Mateo, Apellido: Valdés, ID: 3333, Fecha de nacimiento: 2009/02/27, Tipo preferido de mascota: Hamsters

Al Club #3 se le agrega el siguiente miembro:

- Miembro #1: Nombre: Juan, Apellido: Sánchez, ID: 7767, Fecha de nacimiento: 2010/07/15, Tipo preferido de mascota: Peces

Método	Escenario	Valores de entrada	Resultado
orderByName()	loadScene1()	-	True si los clubes del Club Collection han sido ordenados por nombre y retorna los clubes con nombres en orden lexicográfico
orderById()	loadScene1()	-	True si los clubes del Club Collection han sido ordenados por nombre y retorna los clubes con nombres en orden lexicográfico
orderByCreationDate()	loadScene1()	-	True si los clubes del Club Collection han sido ordenados por fecha de creación y retorna los clubes con fechas de creación en orden lexicográfico

orderByType()	loadScene1()	-	True si los clubes del Club Collection han sido ordenados por tipo de mascota y retorna los clubes con tipo de mascota en orden lexicográfico
petOwnersBinarySearchName()	loadScene1()	-	True si al buscar un nombre me retorna un club con ese nombre
petOwnersBinarySearchId()	loadScene1()	-	True si al buscar un id me retorna un club con ese id
petOwnersBinarySearchCreationDate()	loadScene1()	-	True si al buscar una fecha de creación me retorna un club con ese nombre
petOwnersBinarySearchType()	loadScene1()	-	True si al buscar un tipo de mascota me retorna un club con ese tipo de mascota
petOwnersSearchName()	loadScene1()	-	True si al buscar un nombre me retorna un club con ese nombre
petOwnersSearchId()	loadScene1()	-	True si al buscar un id me retorna un club con ese id
petOwnersSearchCreationDate()	loadScene1()	-	True si al buscar una fecha de creación me retorna un club con ese nombre
petOwnersSearchType()	loadScene1()	-	True si al buscar un tipo de mascota me retorna un club con ese tipo de mascota

Clase: Club

Escenarios:

loadScene1(): Se crea un escenario con 3 clubes de la siguiente forma:

- Club #1: Nombre: Johan Club, ID: 12345678, Fecha de creación: 2000/05/05, Tipo de mascotas: Perros
- Club #2: Nombre: Ariza Club, ID: 12345679, Fecha de creación: 1993/09/05, Tipo de mascotas: Gatos
- Club #3: Nombre: Valdés Club, ID: 55554444, Fecha de creación: 1956/10/06, Tipo de mascotas: Pájaros

Al Club #1 se le agregan los siguientes miembros:

- Miembro #1: Nombre: Juan David, Apellido: Ossa, ID: 1234, Fecha de nacimiento: 2000/01/01, Tipo preferido de mascota: Perros
- Miembro #2: Nombre: Esteban, Apellido: Ariza, ID: 4445, Fecha de nacimiento: 2005/02/04, Tipo preferido de mascotas: Gatos
- Miembro #3: Nombre: Johan, Apellido: Giraldo, ID: 5656, Fecha de nacimiento: 2004/06/24, Tipo preferido de mascota: Serpiente

Al Club #2 se le agregan los siguientes miembros:

- Miembro #1: Nombre: David Jhun, Apellido: Kim, ID: 4321, Fecha de nacimiento: 1998/06/04, Tipo preferido de mascota: Perros
- Miembro #2: Nombre: Mateo, Apellido: Valdés, ID: 3333, Fecha de nacimiento: 2009/02/27, Tipo preferido de mascota: Hamsters

Al Club #3 se le agrega el siguiente miembro:

- Miembro #1: Nombre: Juan, Apellido: Sánchez, ID: 7767, Fecha de nacimiento: 2010/07/15, Tipo preferido de mascota: Peces

Método	Escenario	Valores de entrada	Resultado
orderByName()	loadScene1()	-	True si los dueños del Club #1 han sido ordenados por nombre y retorna los clubes con nombres en orden lexicográfico
orderByLastName()	loadScene1()	-	True si los dueños del Club #1 han sido ordenados por apellidos y retorna los dueños con nombres en orden lexicográfico
orderById()	loadScene1()	-	True si los dueños del Club #1 han sido ordenados por id y retorna los dueños con id en orden lexicográfico
orderByBirthDate()	loadScene1()	-	True si los dueños del Club #1 han sido ordenados por fecha de nacimiento y retorna los dueños con la fecha de nacimiento en orden lexicográfico
orderByPrefPetType()	loadScene1()	-	True si los dueños han sido ordenados por nombre y retorna los clubes con nombres en orden lexicográfico
petOwnersBinarySearchName()	loadScene1()	-	True si al buscar un nombre me retorna un dueño con ese nombre
petOwnersBinarySearchLastName()	loadScene1()	-	True si al buscar un apellido me retorna un dueño con ese apellido
petOwnersBinarySearchId()	loadScene1()	-	True si al buscar un id me retorna un dueño con ese id
petOwnersBinarySearchBirthDate()	loadScene1()	-	True si al buscar una fecha de nacimiento me retorna un dueño con esa fecha de nacimiento
petOwnersBinarySearchPrefPetType()	loadScene1()	-	True si al buscar un tipo de mascota de preferencia me retorna

			un dueño con ese tipo de mascota de preferencia
petOwnersSearchName()	loadScene1()	-	True si al buscar un nombre me retorna un dueño con ese nombre
petOwnersSearchLastName()	loadScene1()	-	True si al buscar un apellido me retorna un dueño con ese apellido
petOwnersSearchId()	loadScene1()	-	True si al buscar un id me retorna un dueño con ese id
petOwnersSearchBirthDate()	loadScene1()	-	True si al buscar una fecha de nacimiento me retorna un dueño con esa fecha de nacimiento
petOwnersSearchPrefPetType()	loadScene1()	-	True si al buscar un tipo de mascota de preferencia me retorna un dueño con ese tipo de mascota de preferencia

Clase: PetOwner

Escenarios:

loadScene1(): Se crea un escenario con tres dueños de mascota con los siguientes atributos:

- Miembro #1: Nombre: Juan David, Apellido: Ossa, ID: 1234, Fecha de nacimiento: 2000/01/01, Tipo preferido de mascota: Perros
- Miembro #2: Nombre: Esteban, Apellido: Ariza, ID: 4445, Fecha de nacimiento: 2005/02/04, Tipo preferido de mascotas: Gatos
- Miembro #3: Nombre: Johan, Apellido: Giraldo, ID: 5656, Fecha de nacimiento: 2004/06/24, Tipo preferido de mascota: Serpiente

Al miembro #1 se le agregan tres mascotas con los siguientes atributos:

- Mascota #1: Nombre: Max, ID: 4321, Fecha de nacimiento: 2000/01/05, Género: M, Tipo de mascota: Perro
- Mascota #2: Nombre: Draco, ID: 7564, Fecha de nacimiento: 2015/10/05, Género: M, Tipo de mascota: Serpiente
- Mascota #3: Nombre: Impa, ID: 4445, Fecha de nacimiento: 2017/12/18, Género: F, Tipo de mascota: Gato

Al miembro #2 se le agregan dos mascotas con los siguientes atributos:

- Mascota #4: Nombre: Milo, ID: 0123, Fecha de nacimiento: 2018/01/05, Género: M, Tipo de mascota: Perro
- Mascota #5: Nombre: Pocoya, ID: 4324, Fecha de nacimiento: 2012/02/05, Género: F, Tipo de mascota: Pollo

Al miembro #3 se le agrega una mascota con los siguientes atributos:

- Mascota #6: Nombre: Porkosinia, ID: 3232, Fecha de nacimiento: 2008/03/07, Género: F, Tipo de mascota: Cerdo

Método	Escenario	Valores de entrada	Resultado
orderByName()	loadScene1()	-	True si las mascotas del dueño #1 han sido ordenadas por nombre y retorna las mascotas con nombre en orden lexicográfico
orderByGender()	loadScene1()	-	True si las mascotas del dueño #1 han sido ordenadas por género y retorna las mascotas con género en orden lexicográfico
orderById()	loadScene1()	-	True si las mascotas del dueño #1 han sido ordenadas por ID y retorna las mascotas con ID en orden lexicográfico
orderByBirthDate()	loadScene1()	-	True si las mascotas del dueño #1 han sido ordenadas por fecha de nacimiento y retorna las mascotas con nacimiento en orden lexicográfico
orderByType()	loadScene1()	-	True si las mascotas del dueño #1 han sido ordenadas por tipo de mascota y retorna las mascotas con tipo de mascota en orden lexicográfico
petsBinarySearchName()	loadScene1()	-	True si al buscar un género me retorna una mascota con ese género
petsBinarySearchGender ()	loadScene1()	-	True si al buscar un nombre me retorna una mascota con ese nombre
petsBinarySearchId()	loadScene1()	-	True si al buscar un ID me retorna una mascota con ese ID
petsBinarySearchBirthDate()	loadScene1()	-	True si al buscar una fecha de nacimiento me retorna una mascota con esa fecha de nacimiento
petsBinarySearchType()	loadScene1()	-	True si al buscar un nombre me retorna una mascota con ese nombre
petsSearchName()	loadScene1()	-	True si al buscar un tipo de mascota me retorna una mascota con ese tipo de mascota

<code>petsSearchGender()</code>	<code>loadScene1()</code>	-	True si al buscar un género me retorna una mascota con ese género
<code>petsSearchId()</code>	<code>loadScene1()</code>	-	True si al buscar un ID me retorna una mascota con ese ID
<code>petsSearchBirthDate()</code>	<code>loadScene1()</code>	-	True si al buscar una fecha de nacimiento me retorna una mascota con esa fecha de nacimiento
<code>petsSearchType()</code>	<code>loadScene1()</code>	-	True si al buscar un tipo de mascota tipo de mascota me retorna una mascota con ese tipo de mascota