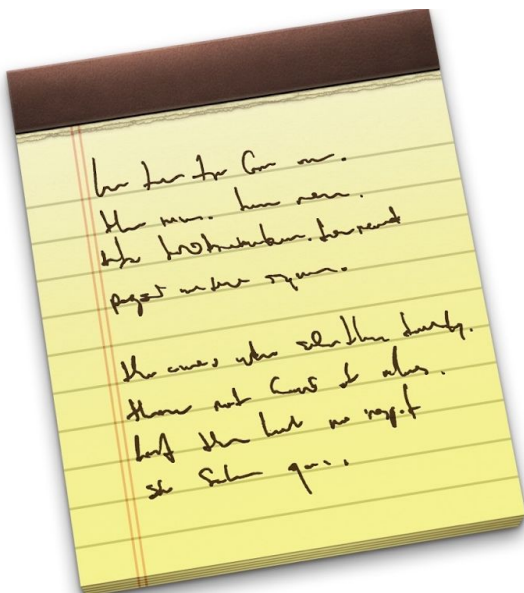


Ingeniería del Software

Trabajo de la asignatura

Aplicación Notas



Índice

Índice	2
Resumen	3
Introducción	3
Objetivos	4
Requisitos	5
Requisitos funcionales	5
Diagrama de casos de uso	6
Flujos de eventos de casos de uso	7
Análisis	13
Diagrama de clases	13
Diagramas de secuencia	14
Mapa de navegación	23
Diseño del sistema	24
Diagrama de paquetes	24
Diagrama de componentes	24
Diagrama de despliegue	25
Diseño de objetos	26
Decisiones de diseño tomadas	26
Diagrama de clases	27
Diagramas de secuencia	28
Modelo entidad-relación de la base de datos	39
Pruebas	40
Pruebas de caja negra	40
Prueba de volumen	45
Prueba de usabilidad	45
Prueba de sobrecarga	46
Pruebas de sistema	46
Resultados y conclusiones	48
Bibliografía	49

Resumen

Este trabajo ha consistido en la elaboración de una aplicación de notas en la que los usuarios pueden gestionar libremente notas, clasificadas o no con categorías. Para poder desarrollar esta aplicación, se ha tomado como punto de partida la aplicación Notepad, diseñada en las prácticas de la asignatura, junto con algunas mejoras adicionales con el objetivo de mejorar la calidad del diseño y proporcionarle al usuario una mejor experiencia a la hora de emplear la aplicación.

Introducción

En este documento se plantea la realización de un trabajo en equipo partiendo de la aplicación informática Notepad, con la que se ha trabajado a lo largo de las prácticas de laboratorio de la asignatura.

Al igual que en las últimas prácticas se tomará como punto de partida el código fuente de la aplicación Notepad en su última versión. Además de los requisitos funcionales básicos y las modificaciones exigidas durante las prácticas se han tenido en cuenta nuevos requisitos funcionales.

El trabajo de la asignatura se ha realizado en parejas de forma que la planificación y realización de las tareas se ha dividido entre ambos miembros, permitiendo así mayor agilidad y rapidez en la distribución del trabajo.

Objetivos

- Comprender los pasos en los que se divide el proceso de creación software.
- Distinguir y diferenciar la fase de análisis de la fase de diseño a la hora de diseñar una aplicación software.
- Familiarización con el lenguaje UML para realizar diagramas de clases, diagramas de componentes, diagramas de despliegue, diagramas de paquetes y diagramas de secuencia.
- Tomar contacto con programas dedicados al desarrollo de aplicaciones software, concretamente Android Studio.
- Conocer y saber emplear Modelio, una herramienta UML de código abierto basada en una amplia gama de funcionalidades basadas en estándares para desarrolladores de software.
- Realizar sin dificultad pruebas de caja blanca y de caja negra con el objetivo de poder probar que el software diseñado cumple los requisitos especificados durante la fase de requisitos.
- Tener claros los distintos patrones de diseño, así como la forma de implementarlos.

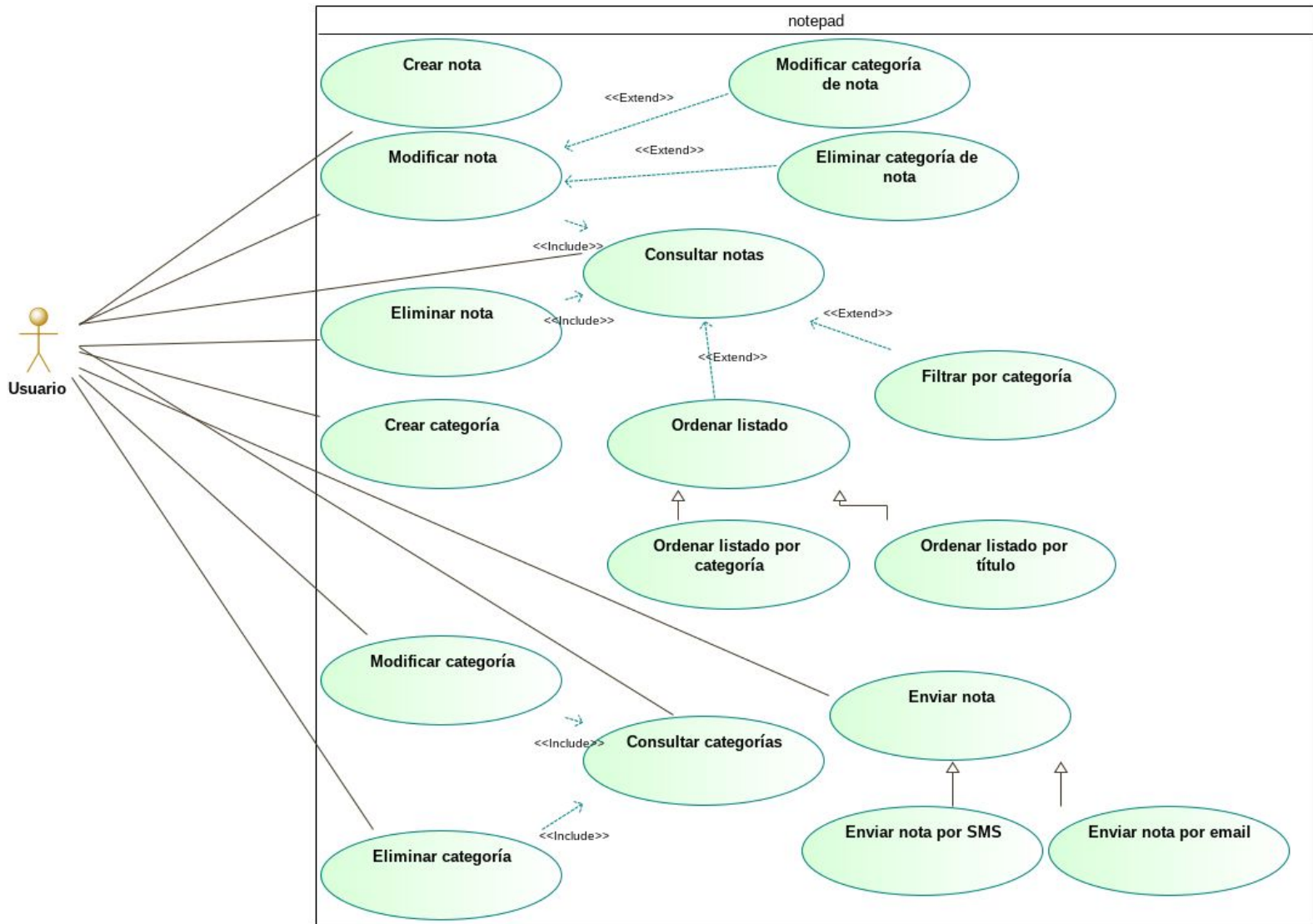
Requisitos

Requisitos funcionales

Código	Descripción
RF-1	El usuario puede crear una nota.
RF-2	Las notas constan de título y cuerpo.
RF-3	El usuario puede modificar una nota existente.
RF-4	El usuario puede eliminar una nota existente.
RF-5	El usuario puede crear una categoría.
RF-6	Una categoría está compuesta por letras y/o números, separadas entre sí por espacios en blanco.
RF-7	El usuario puede modificar una categoría existente.
RF-8	El usuario puede eliminar una categoría existente.
RF-9	El usuario puede asignar a una nota una categoría existente.
RF-10	El usuario puede modificar la categoría asignada previamente a una nota.
RF-11	El usuario puede eliminar la categoría asignada previamente a una nota.
RF-12	El usuario puede consultar el listado de notas ya creadas.
RF-13	El usuario puede ordenar el listado de notas por categoría o título.
RF-14	El usuario puede mostrar el listado de notas filtrado por una categoría determinada.
RF-15	El usuario puede enviar las notas por SMS.
RF-16	El usuario puede enviar las notas por email.

Restricción: Se debe utilizar Android para diseñar e implementar la aplicación.

Diagrama de casos de uso



Flujos de eventos de casos de uso

Caso de uso: Crear nota

Flujo de eventos principal:

1. El caso de uso comienza en el panel principal de notas.
2. El usuario selecciona la opción Crear nueva nota.
3. Se presenta al usuario con panel de creación de notas.
4. El usuario escribe el título y cuerpo de la nueva nota.
5. El usuario selecciona una categoría para la nota o la deja sin categoría.
6. El usuario confirma la creación.
7. La nota se añade al sistema y se presenta en el panel principal.

Caso de uso: Modificar nota

Flujo de eventos principal:

1. incluye Consultar notas.
2. El usuario selecciona una nota.
3. Se presenta al usuario con un panel de edición de la nota seleccionada.
4. El usuario modifica los campos título, cuerpo o ambos.
5. extend (modificar categoría de nota) o extend (eliminar categoría de nota).
6. El usuario confirma la modificación.
7. Se guardan los cambios y se presenta la nota modificada en el panel principal.

Caso de uso: Modificar categoría de nota

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla de edición de una nota.
2. El usuario elige la nueva categoría para la nota
3. La nota queda asignada a otra categoría.

Caso de uso: Eliminar categoría de nota

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla de edición de una nota.
2. El usuario elimina la categoría de la nota.
3. La nota queda sin categoría.

Caso de uso: Eliminar nota

Flujo de eventos principal:

1. incluye Consultar notas.
2. El usuario selecciona una nota.
3. Se presenta al usuario con un panel de edición de la nota seleccionada.
4. El usuario elige eliminar la nota seleccionada.
5. El usuario confirma la eliminación.
6. Se presenta un mensaje confirmando la eliminación.

Caso de uso: Consultar notas

Flujo de eventos principal:

1. El caso de uso comienza en el panel principal de notas.
2. extend (filtrar por categoria).
3. extend (ordenar listado).
4. Se presenta al usuario con un listado de notas.

Caso de uso: Filtrar por categoría

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario elige filtrar por categoría.
3. Se presenta al usuario con una lista de categorías sobre las que filtrar.
4. El usuario elige una categoría sobre la que filtrar las notas.
5. Se presenta una lista de notas filtradas con la categoría seleccionada.

Caso de uso: Ordenar listado por categoría

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona ordenar listado por categoría.
3. Se presenta una lista de notas ordenadas por categoría.

Caso de uso: Ordenar listado por título

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona ordenar listado por título.
3. Se presenta una lista de notas ordenadas por título.

Caso de uso: Crear categoría

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona la opción crear categoría.
3. Se presenta al usuario con una pantalla de creación de categoría.
4. El usuario escribe un nombre para la nueva categoría.
5. El usuario confirma la creación.
6. Se presenta al usuario una lista de categorías con la categoría creada.

Caso de uso: Consultar categorías

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona la opción consultar categorías.
3. Se presenta al usuario con un listado de categorías.

Caso de uso: Modificar categoría

Flujo de eventos principal:

1. incluye Consultar categorías.
2. El usuario selecciona una categoría.
3. Se presenta al usuario con un panel de edición de la categoría seleccionada.
4. El usuario modifica el campo nombre.
5. El usuario confirma la modificación.
6. Se guardan los cambios y se presenta la categoría modificada en un listado de categorías.

Caso de uso: Eliminar categoría

Flujo de eventos principal:

1. incluye Consultar categorías.
2. El usuario selecciona una categoría.
3. Se presenta al usuario con un panel de edición de la categoría seleccionada.
4. El usuario elige eliminar la categoría seleccionada.
5. El usuario confirma la eliminación.
6. Se presenta un mensaje confirmando la eliminación.

Caso de uso: Enviar nota por SMS

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona una nota a enviar.
3. El usuario selecciona la opción enviar nota por SMS.
4. Se presenta al usuario con la aplicación de mensajes, habiendo rellenado título y cuerpo.
5. El usuario rellena el destinatario.
6. El usuario envía el mensaje.
7. Se confirma el envío del mensaje.

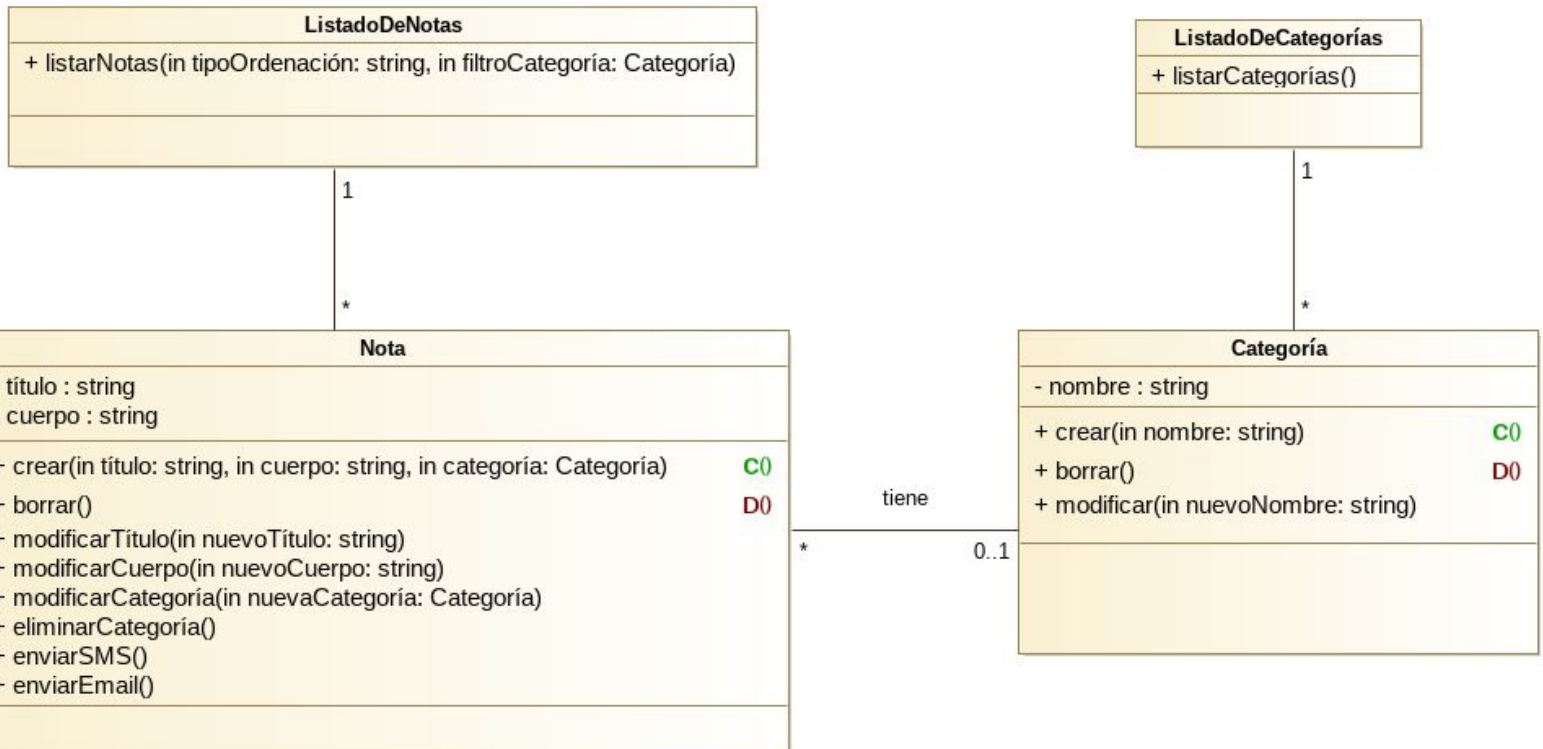
Caso de uso: Enviar nota por email

Flujo de eventos principal:

1. El caso de uso comienza en la pantalla principal de la aplicación.
2. El usuario selecciona una nota a enviar.
3. El usuario selecciona la opción enviar nota por email.
4. Se presenta al usuario con la aplicación de email, habiendo rellenado título y cuerpo.
5. El usuario rellena el destinatario.
6. El usuario envía el email.
7. Se confirma el envío del email.

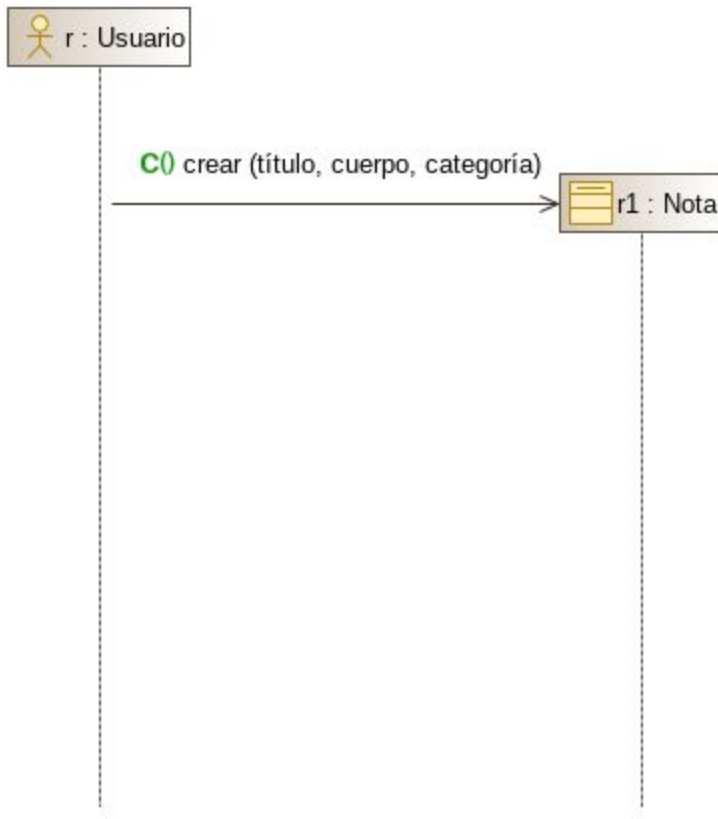
Análisis

Diagrama de clases

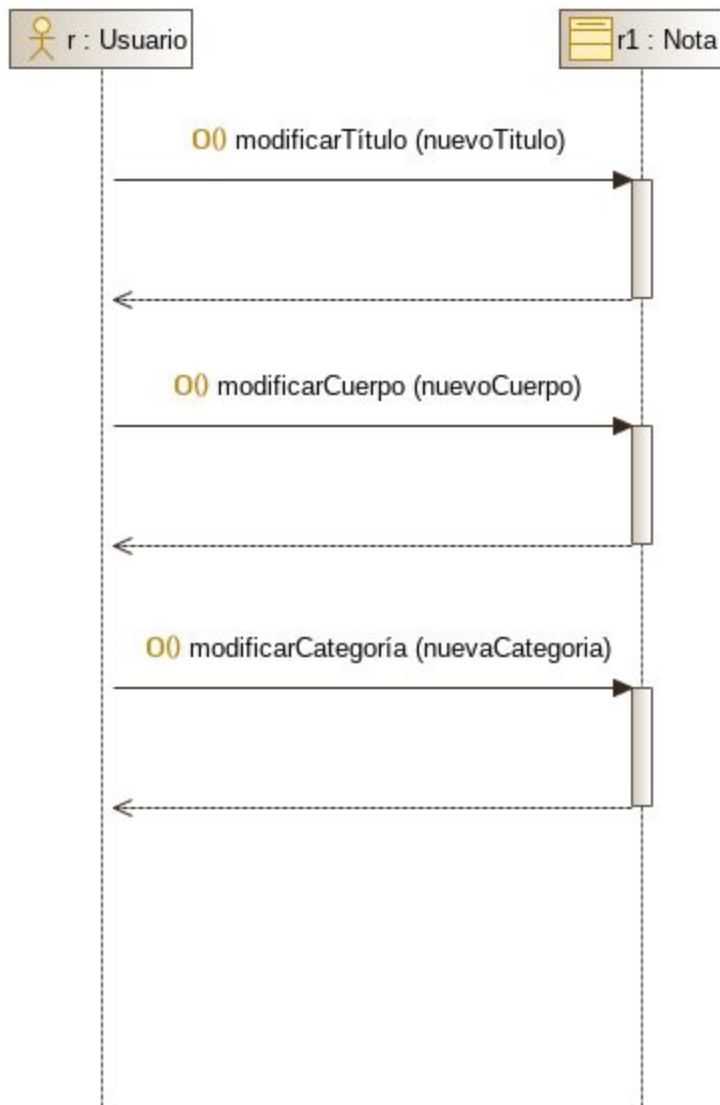


Diagramas de secuencia

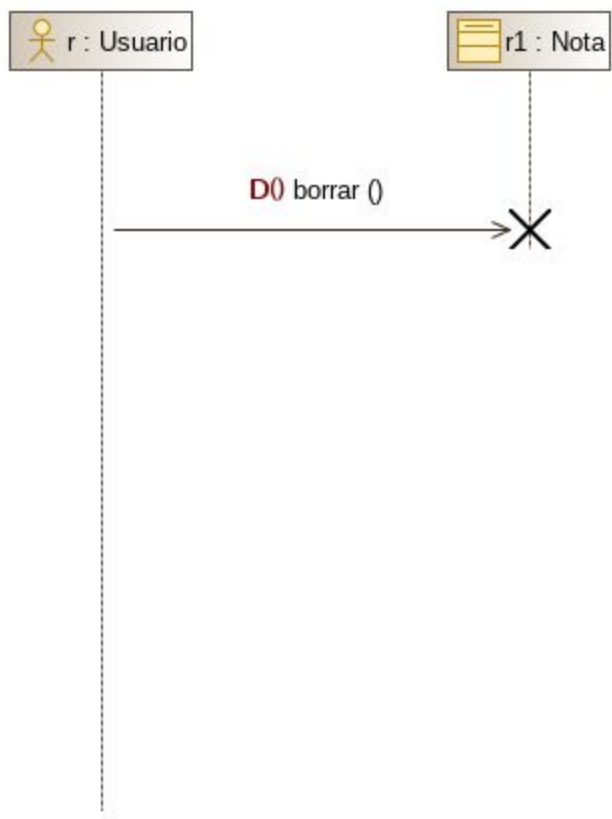
Crear nota



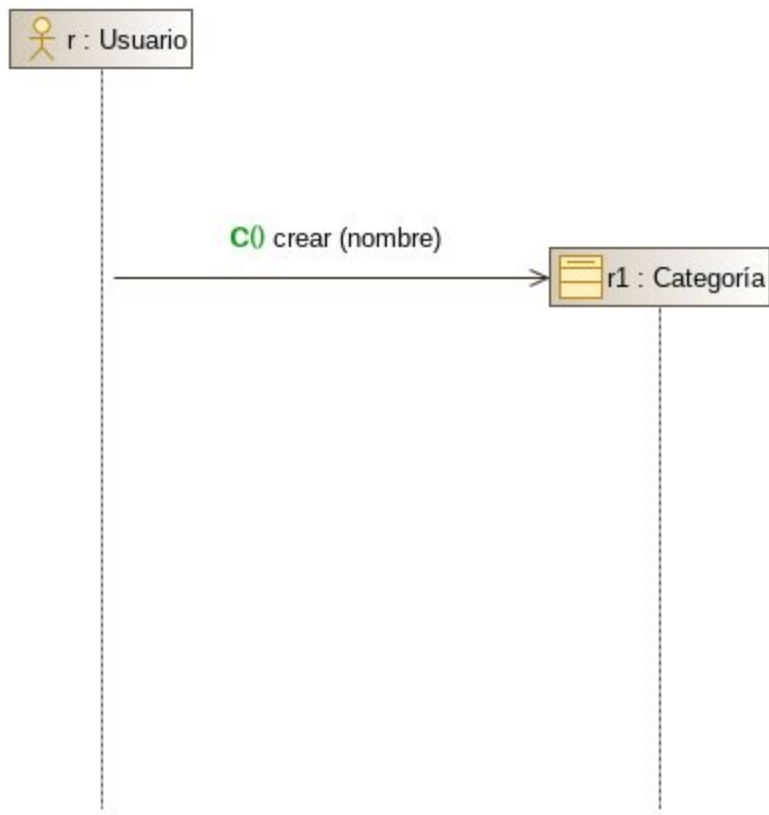
Modificar nota



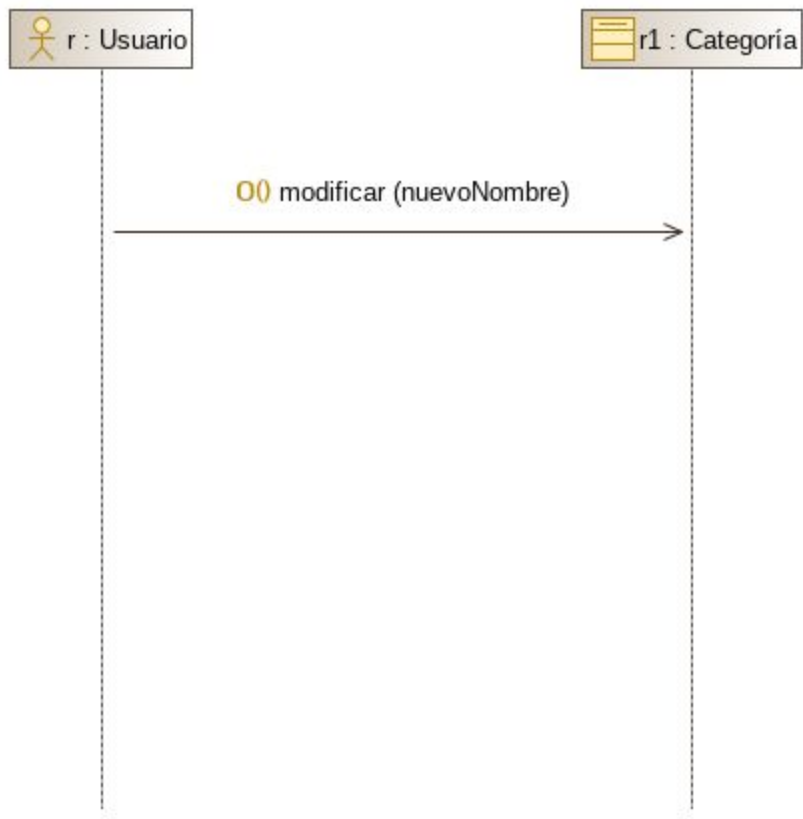
Eliminar nota



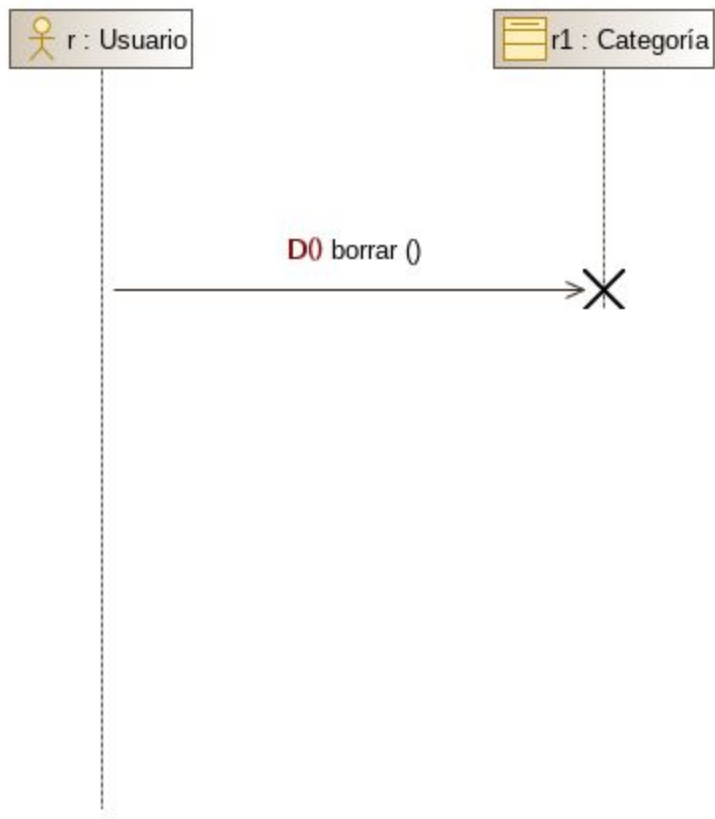
Crear categoría



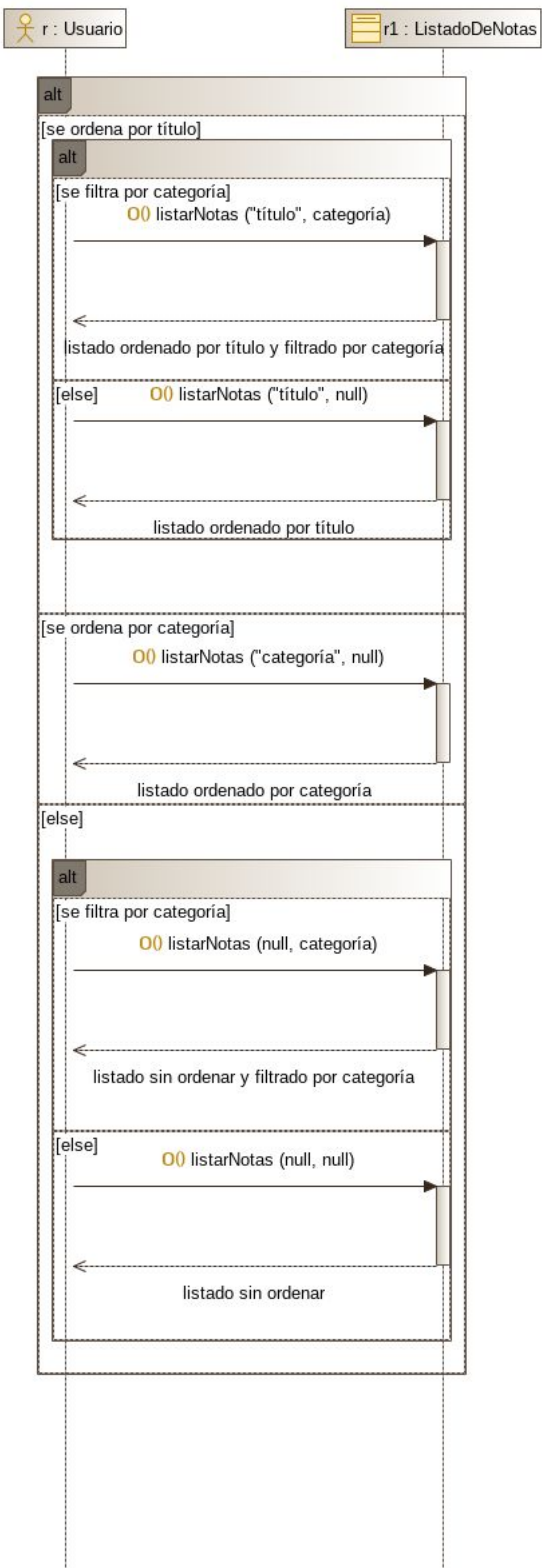
Modificar categoría



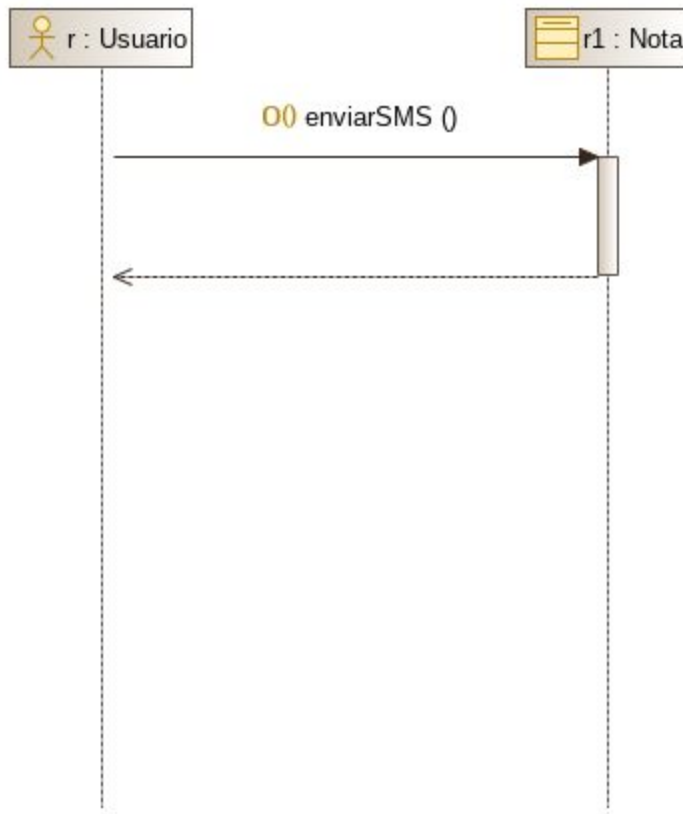
Eliminar categoría



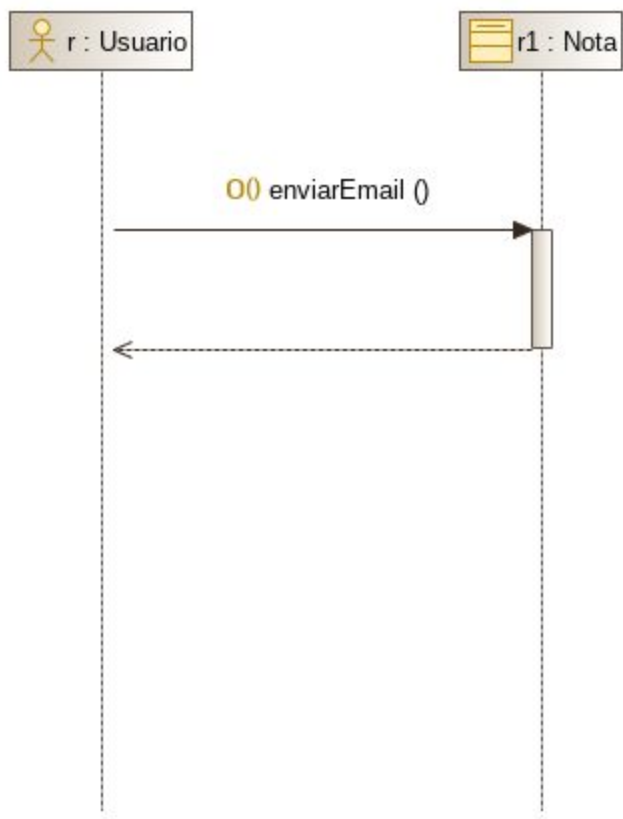
Consultar notas



Enviar SMS

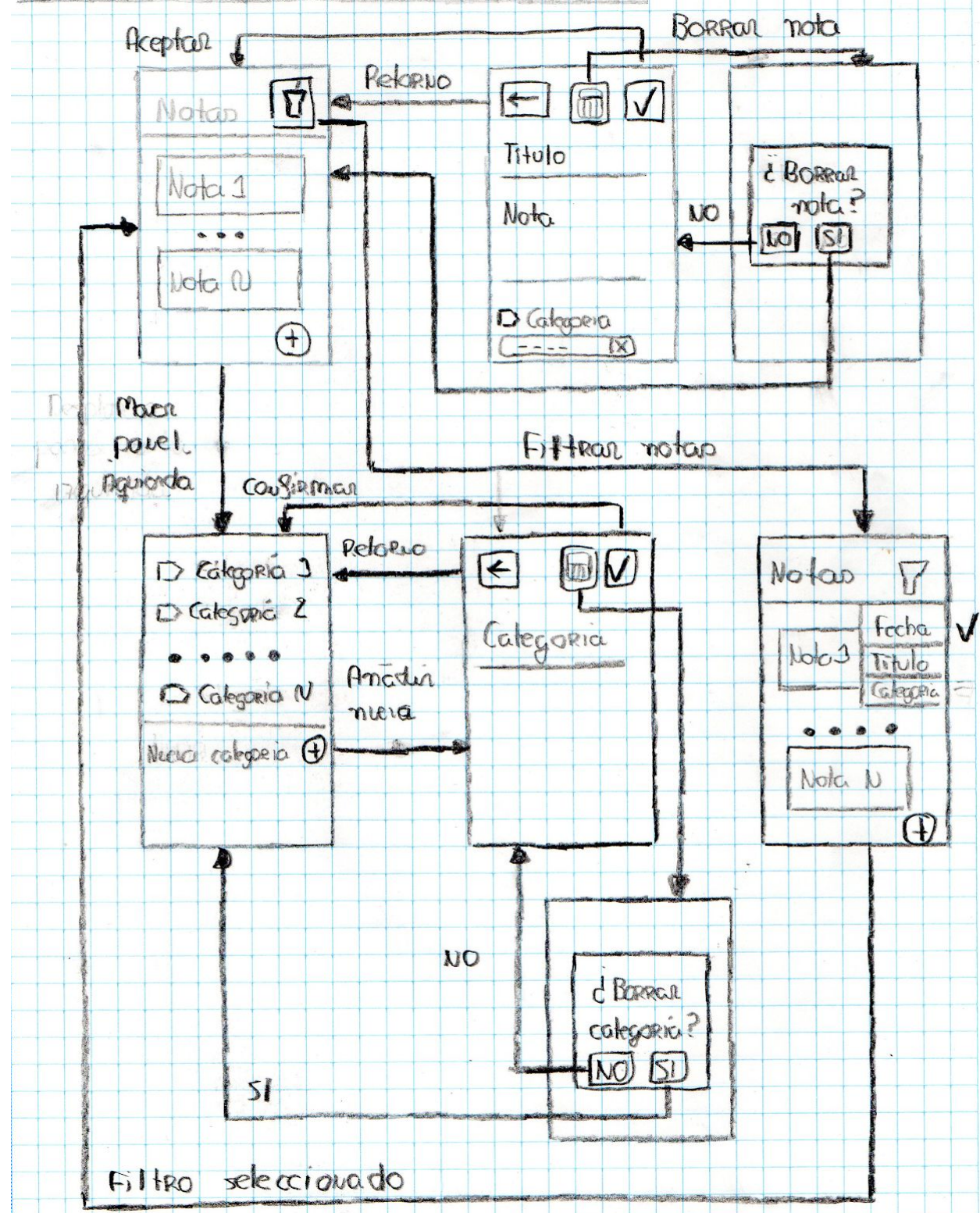


Enviar Email



Mapa de navegación

Mapa de navegación de la aplicación



Diseño del sistema

Diagrama de paquetes



Diagrama de componentes

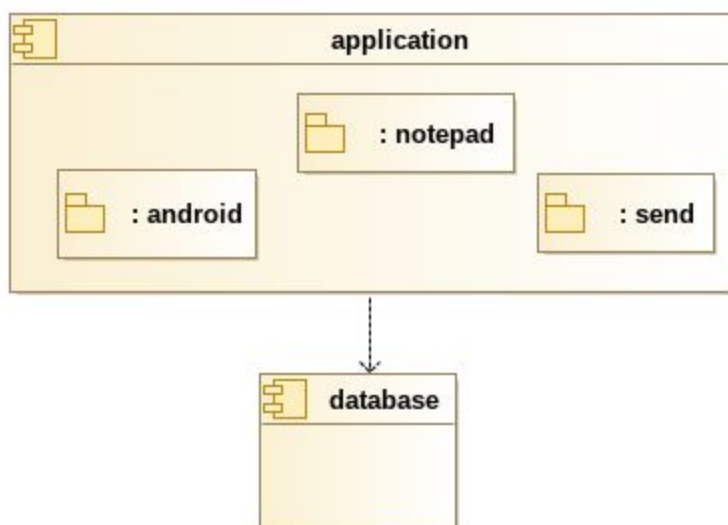
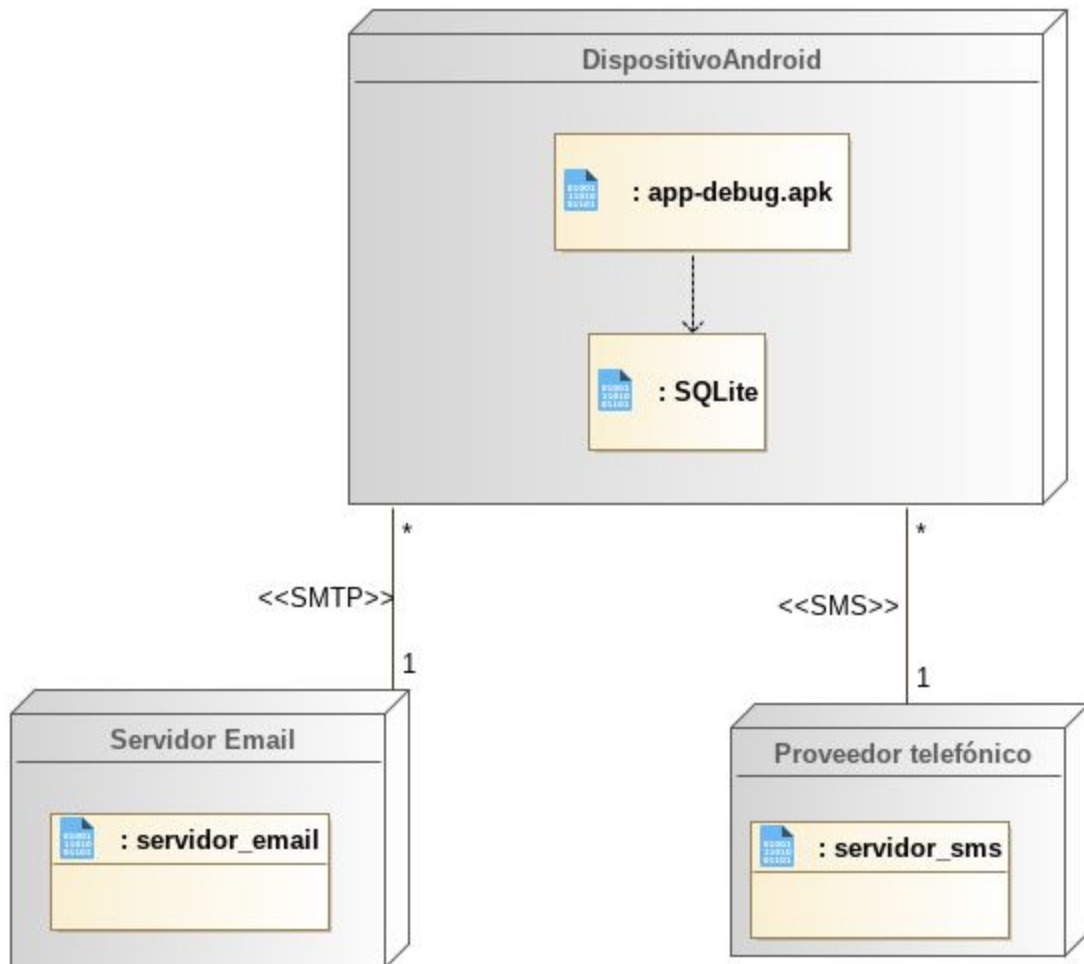


Diagrama de despliegue



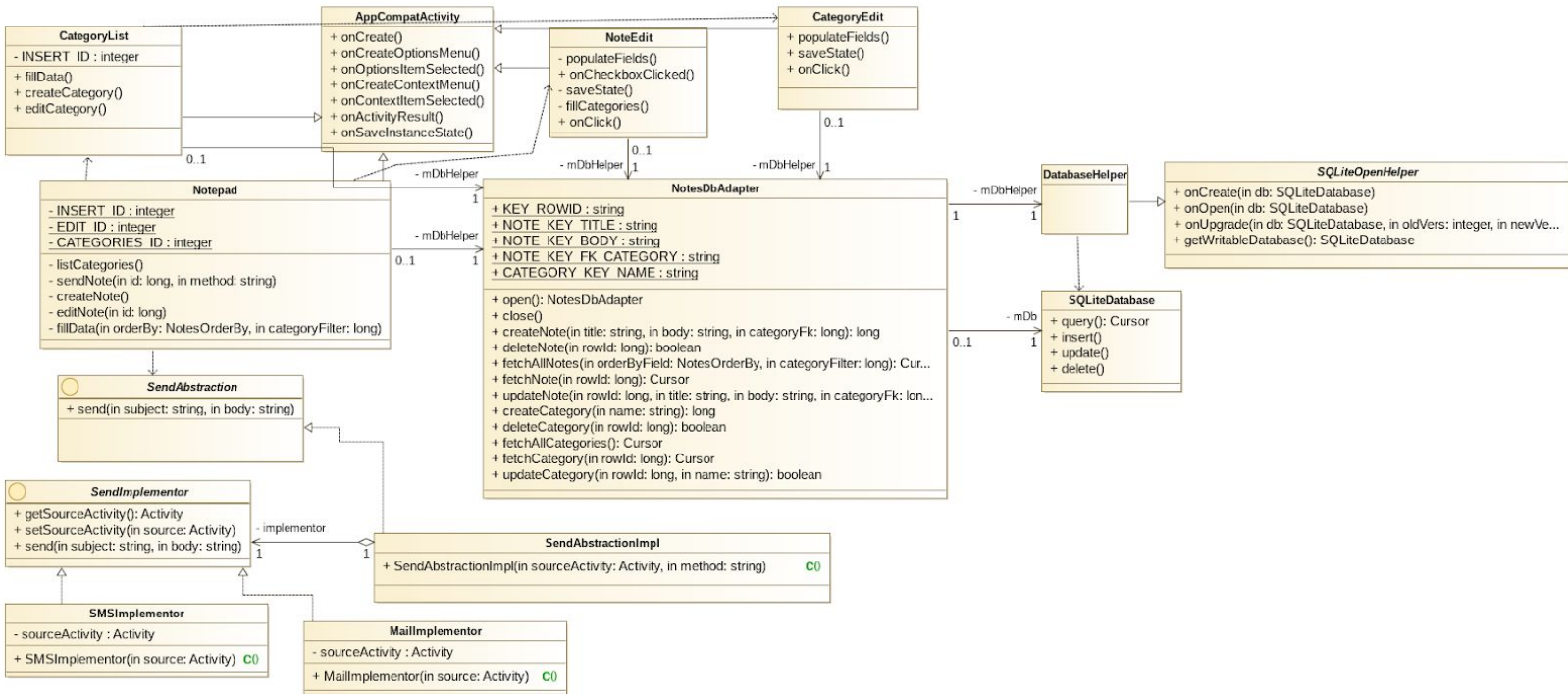
Diseño de objetos

Decisiones de diseño tomadas

A continuación se describen cuales han sido las decisiones de diseño más relevantes

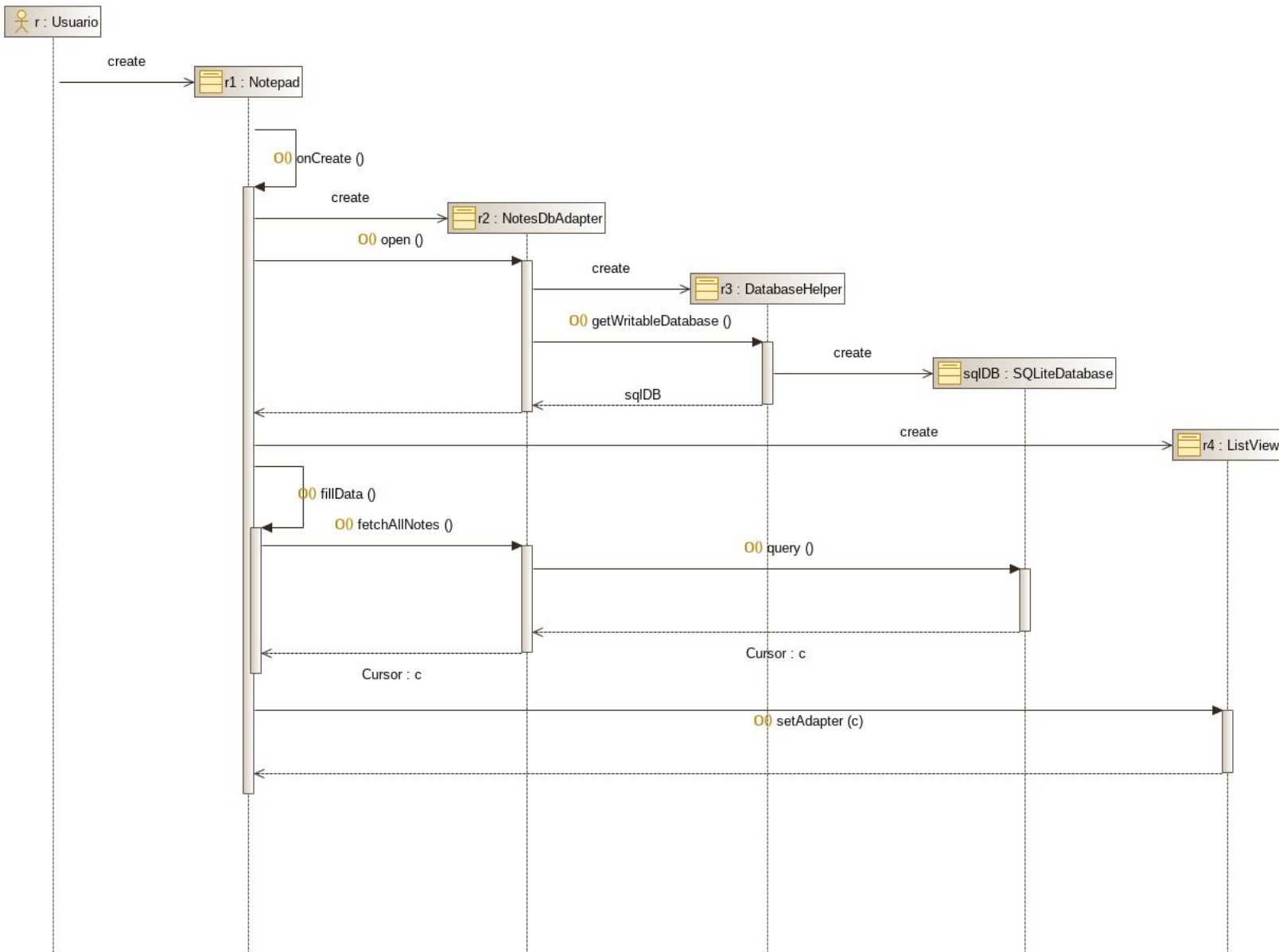
- Se ha empleado el patrón de diseño Bridge para poder seleccionar entre enviar las notificaciones vía SMS o email, proporcionando así poder incorporar con facilidad nuevos métodos de envío de notas.
- En el método `fetchAllNotes` se ha optado por emplear un enum para elegir el modo de ordenación de las notas, limitando de esta manera las posibilidades sobre las que ordenar frente a tener que usar una cadena de texto y evitando también posibles fallos a la hora de llamar al método.
- Se ha decidido activar la integridad referencial de las claves ajenas en la base de datos, desactivadas por defecto, para que al borrar una categoría ésta se elimine de todas las notas que le hagan referencia.
- Para representar una nota con categoría vacía se ha decidido utilizar el valor null.
- Para poder elegir la categoría de una nota se ha usado la clase `Spinner`, proporcionado por el paquete `android.widget`, para así poder mostrar un menú desplegable con la lista de posibles categorías a elegir.
- Para las categorías se ha usado una aproximación muy similar a las notas para poder crearlas y listarlas, teniendo dos clases para las listas (`CategoryList` y `Notepad`) y dos para la edición/creación (`NoteEdit` y `CategoryList`), todas ellas heredando de la clase `AppCompatActivity`.
- Todas las cadenas de texto fueron agrupadas en un único fichero, haciendo más sencilla su gestión al cambiarlas, eliminarlas, etc.
- Para poder mostrar mensajes de feedback al usuario, y para poder filtrar por categorías, se ha utilizado un `AlertDialog`, proporcionado por el paquete `android.app`, pudiendo así generar popups.

Diagrama de clases

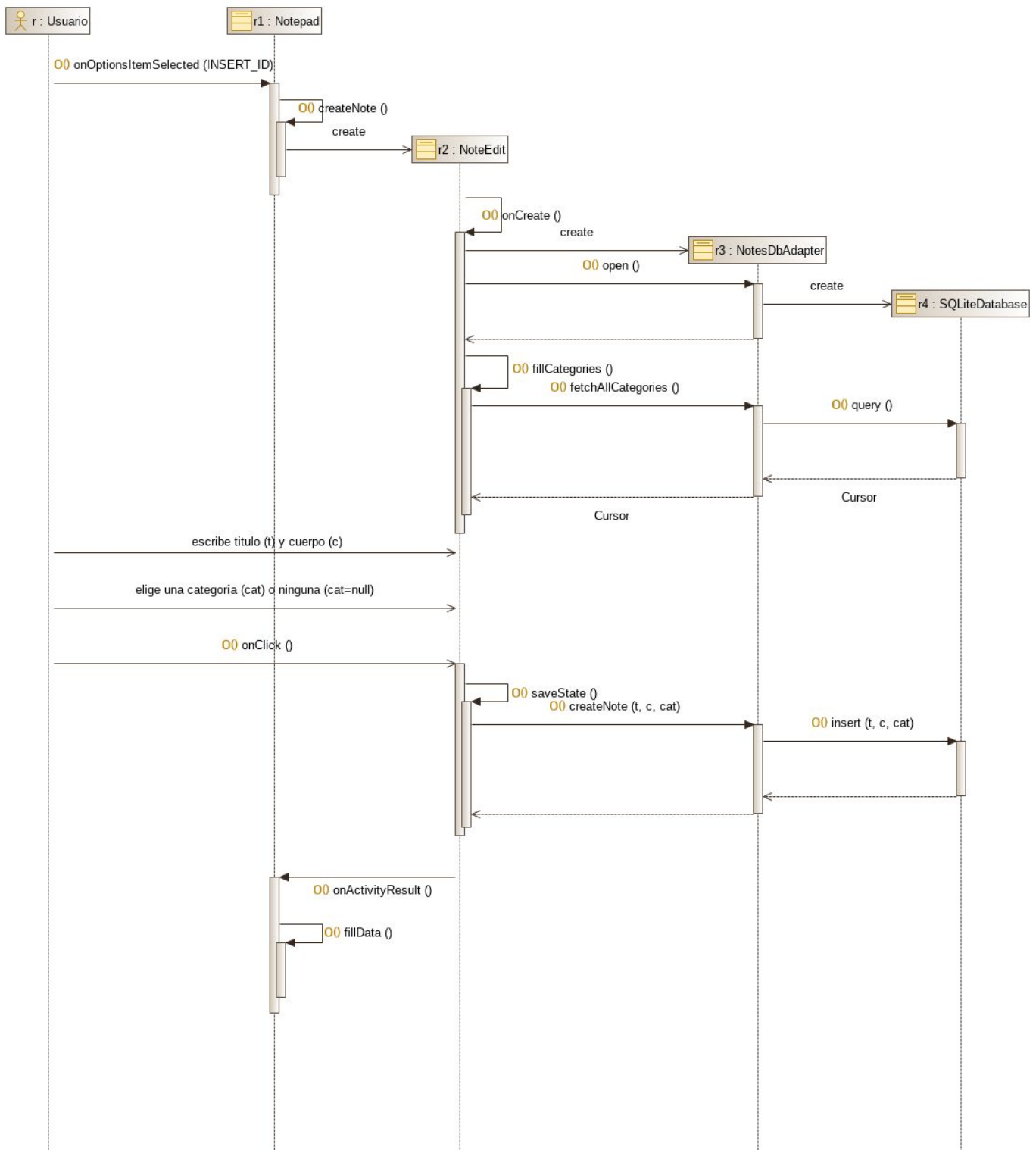


Diagramas de secuencia

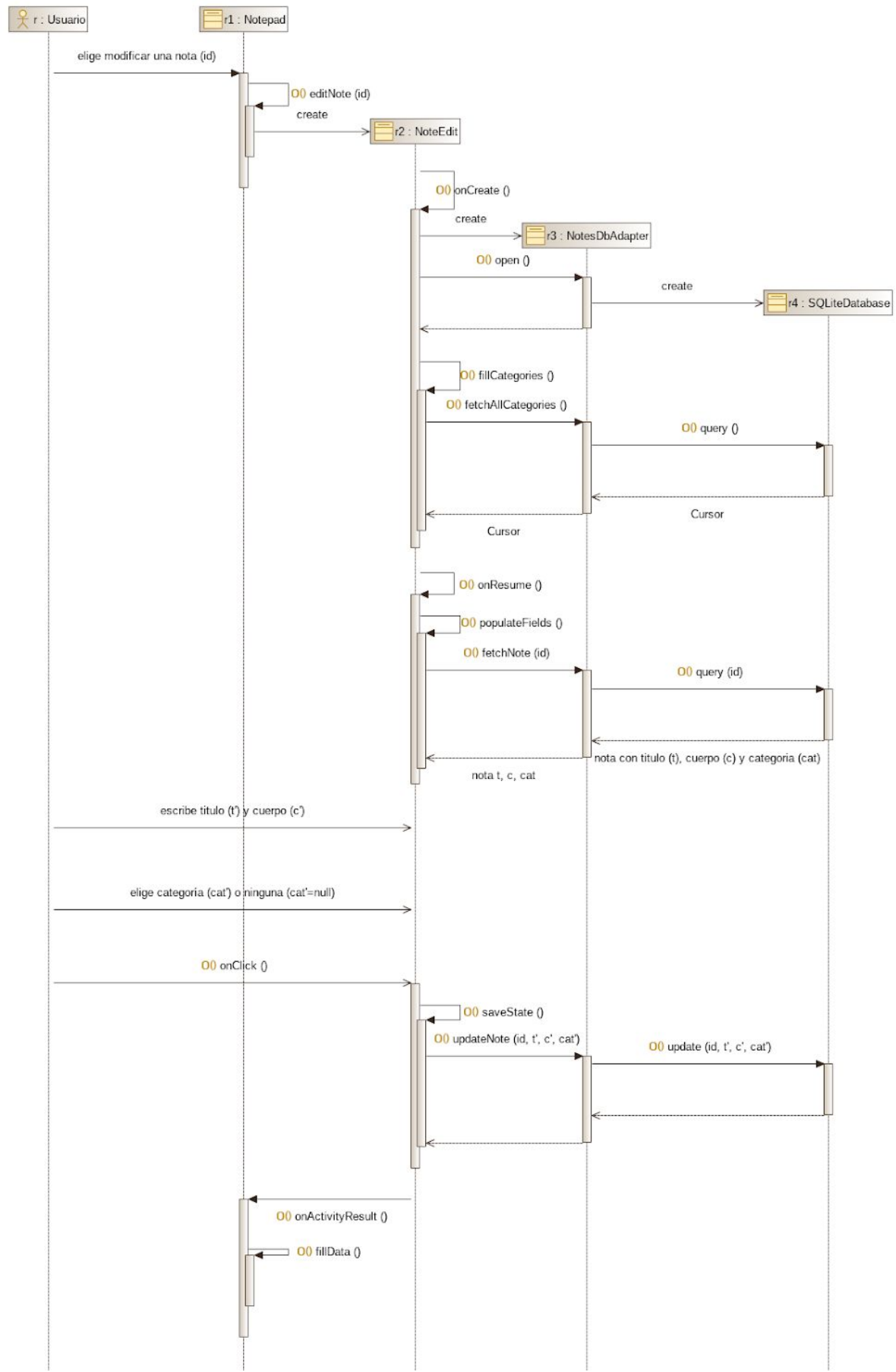
Inicialización de la aplicación



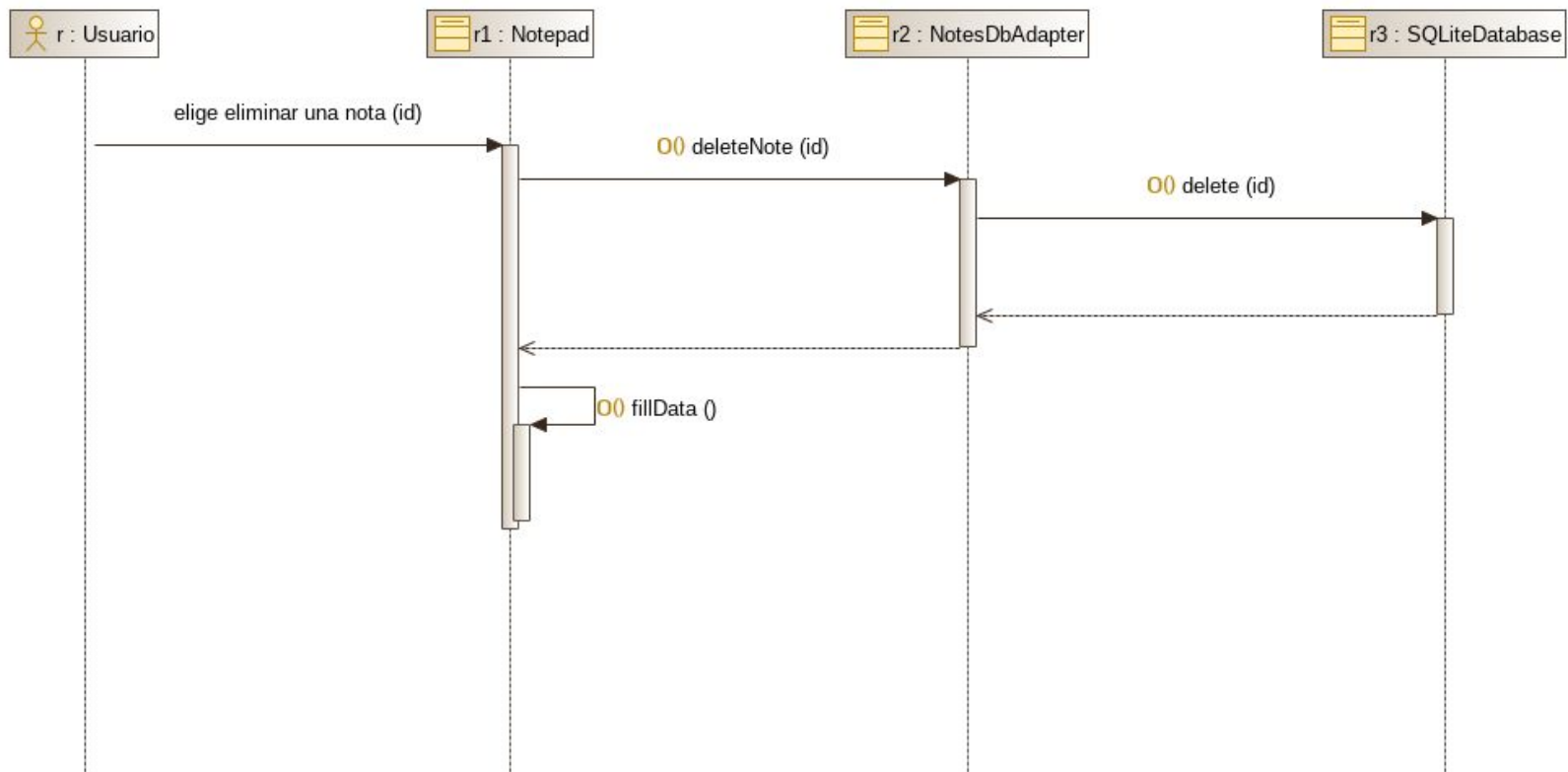
Crear nota



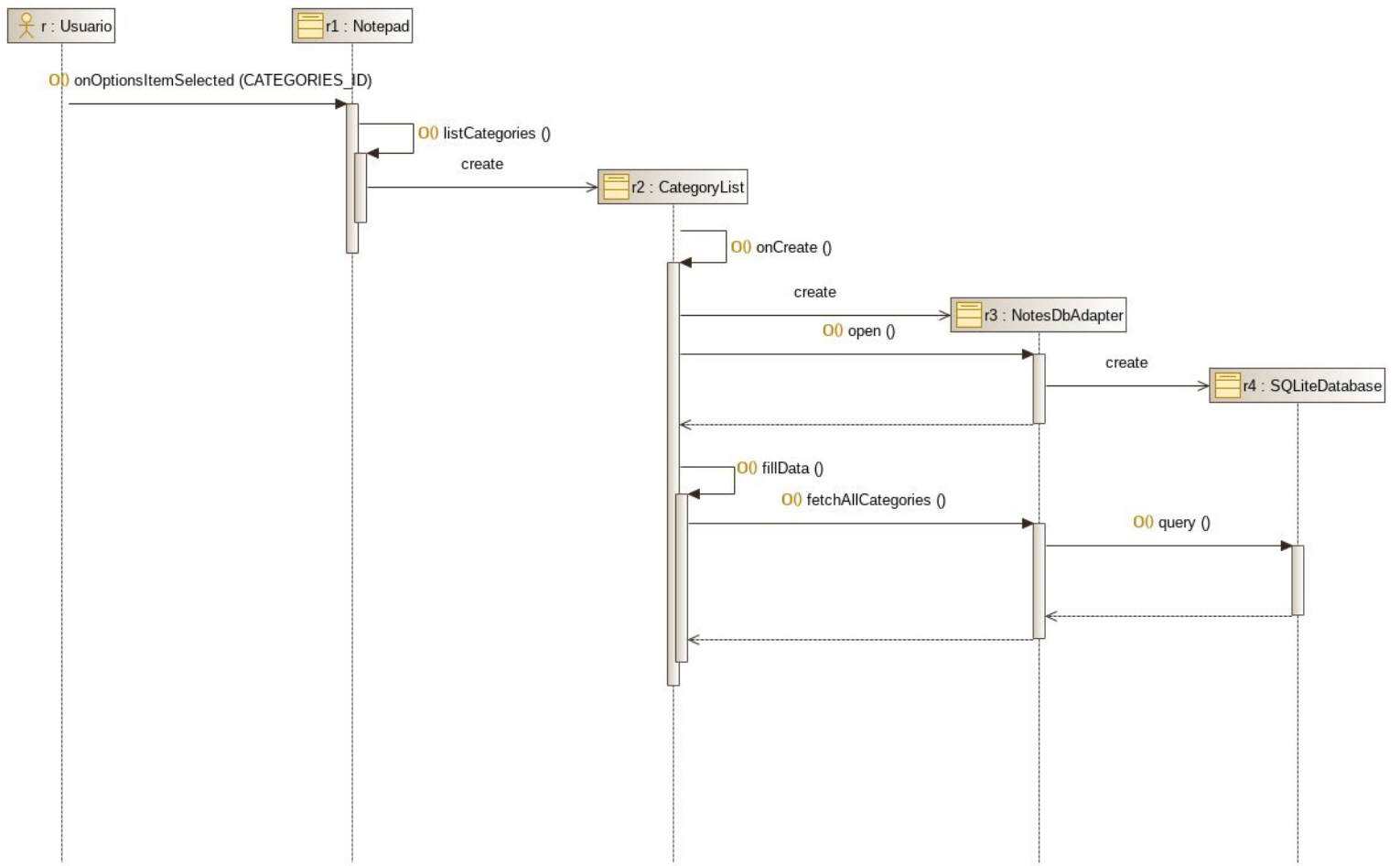
Modificar nota



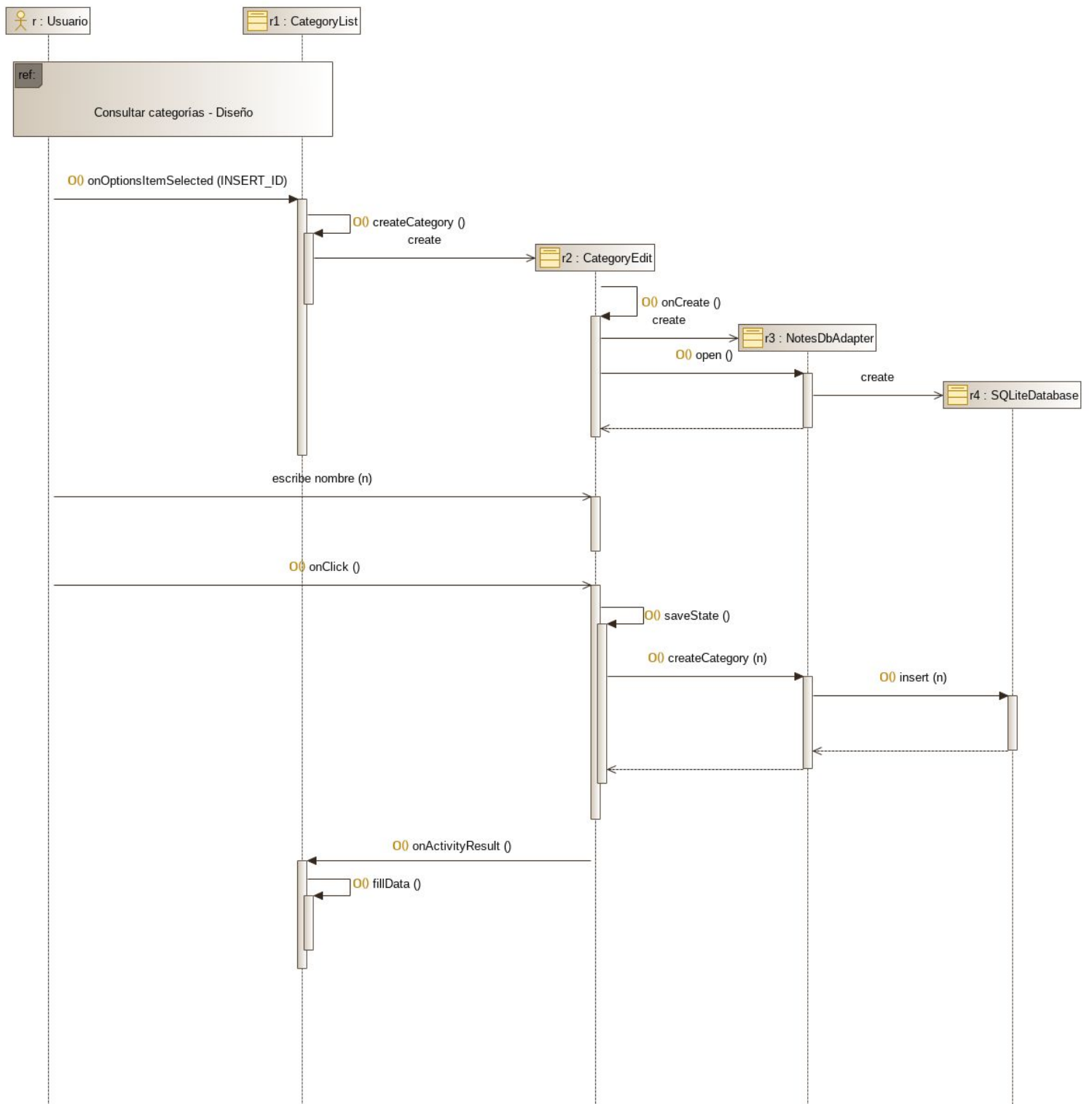
Eliminar nota



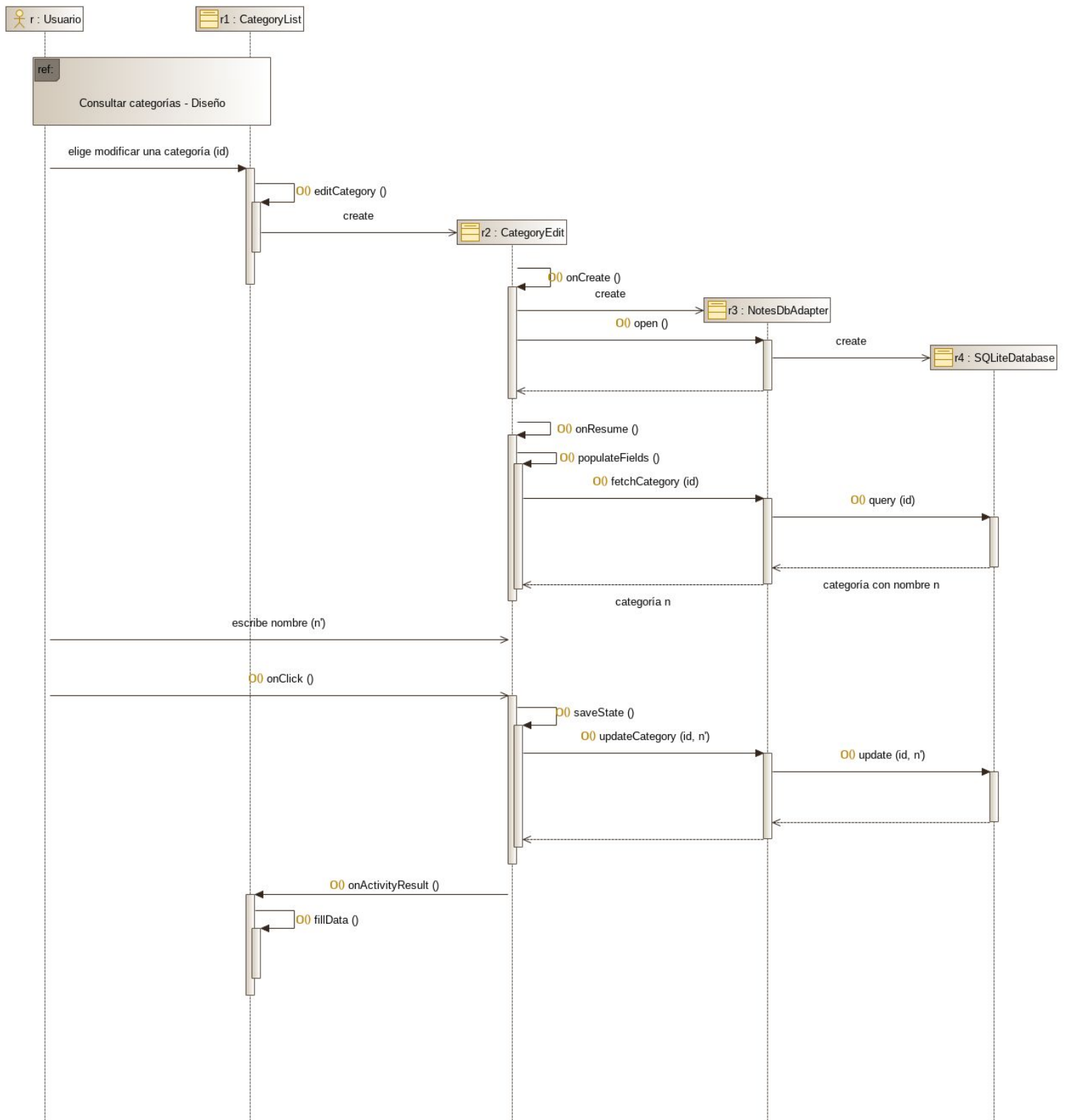
Consultar categorías



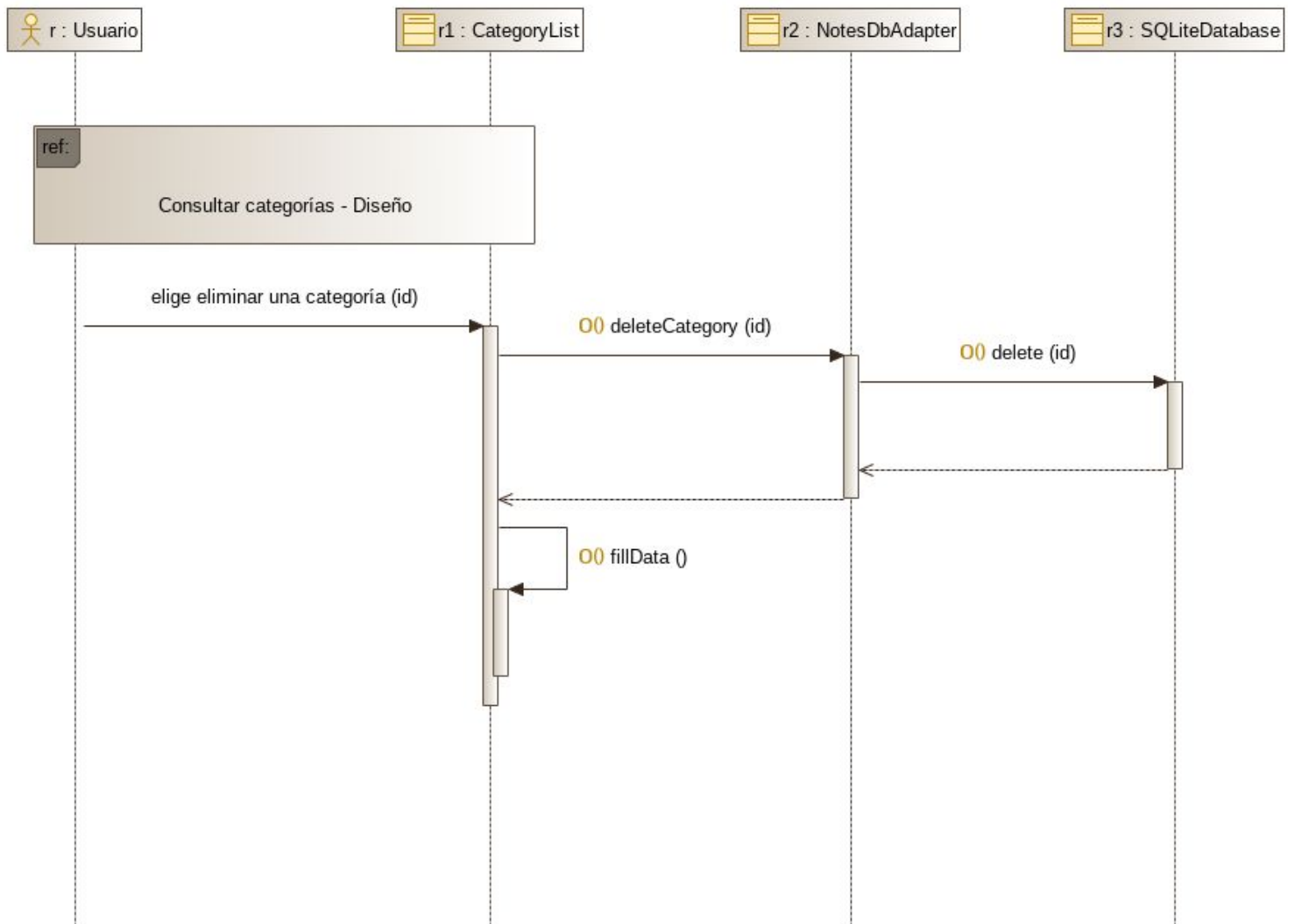
Crear categoría



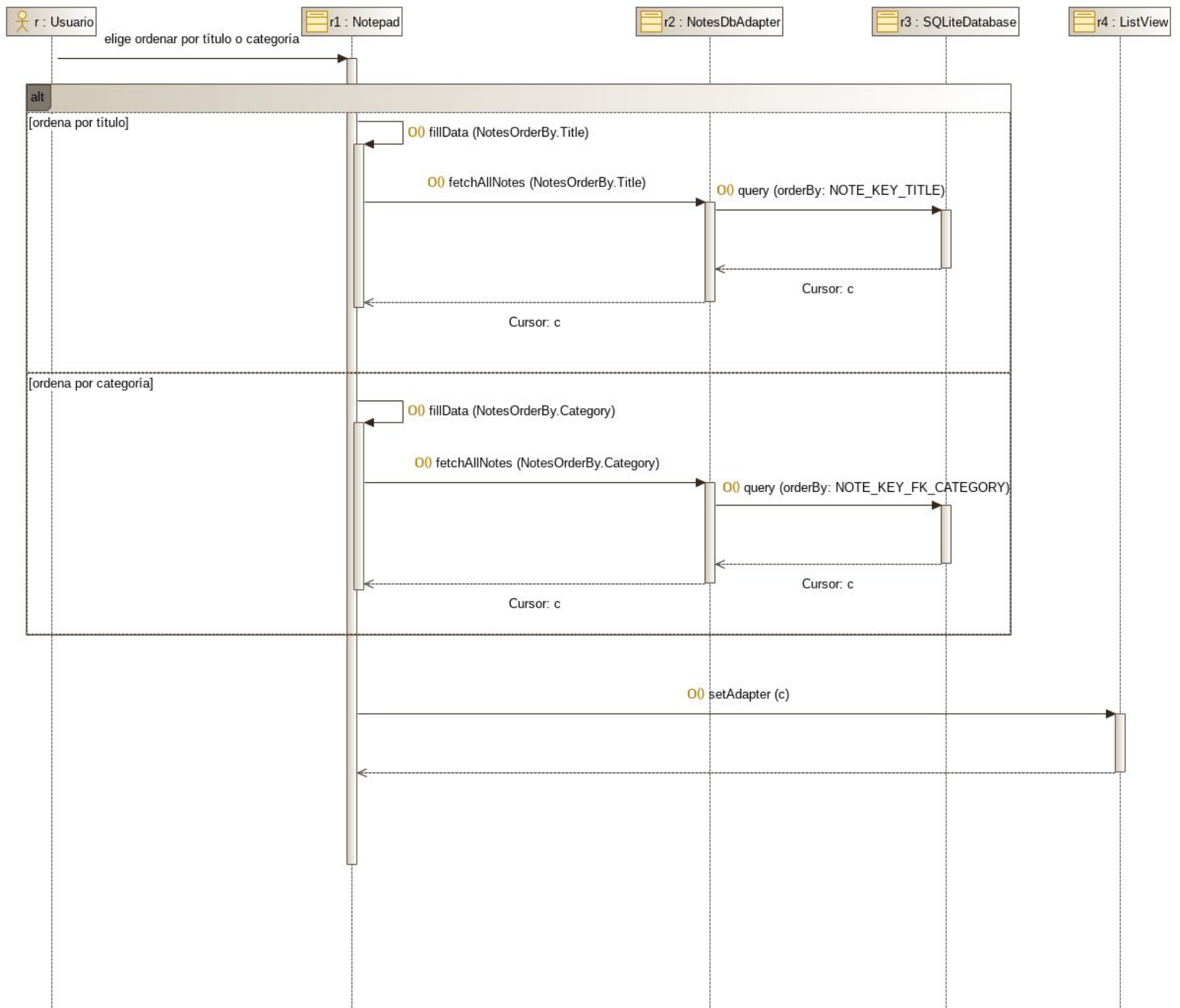
Modificar categoría



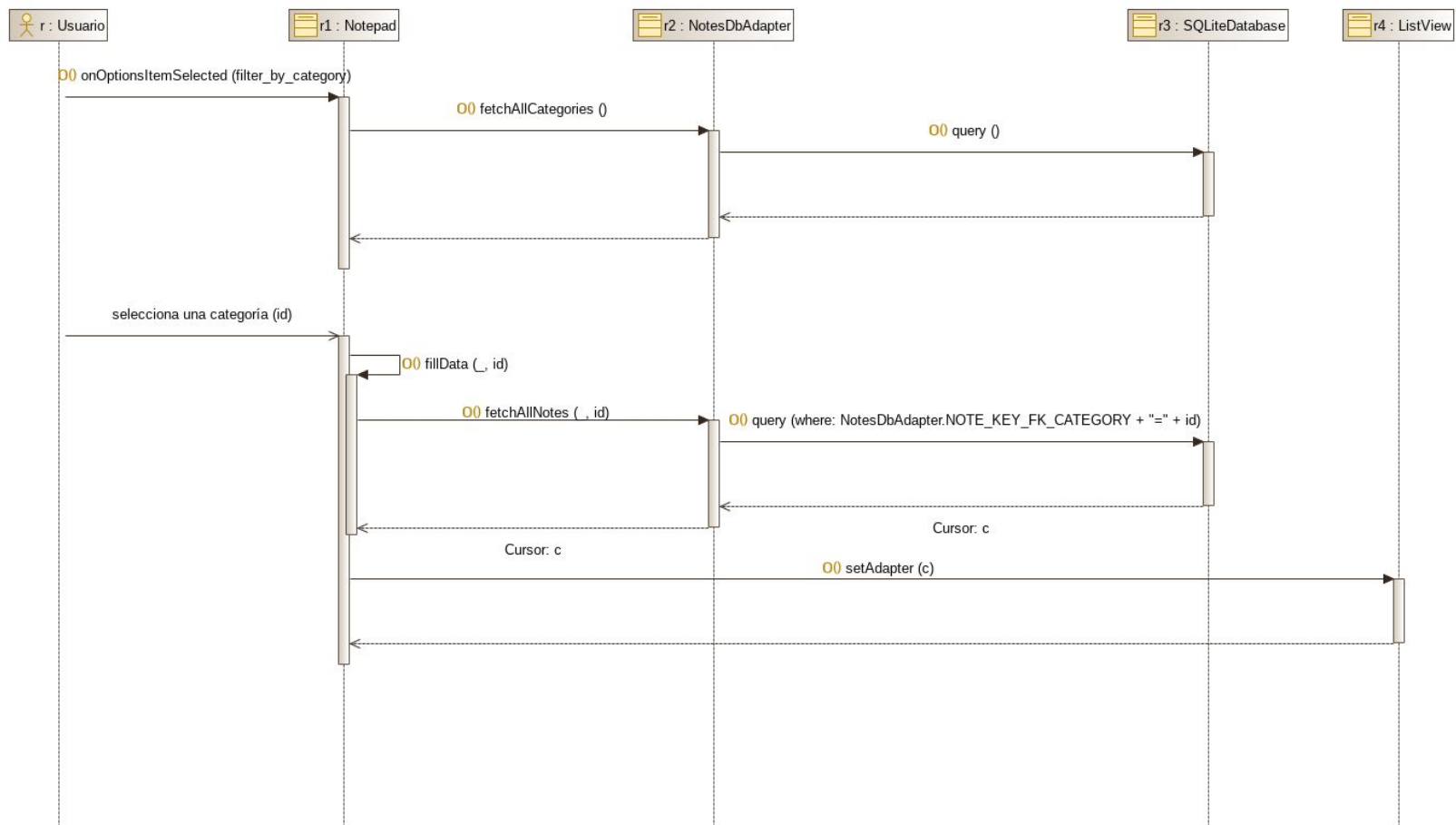
Eliminar categoría



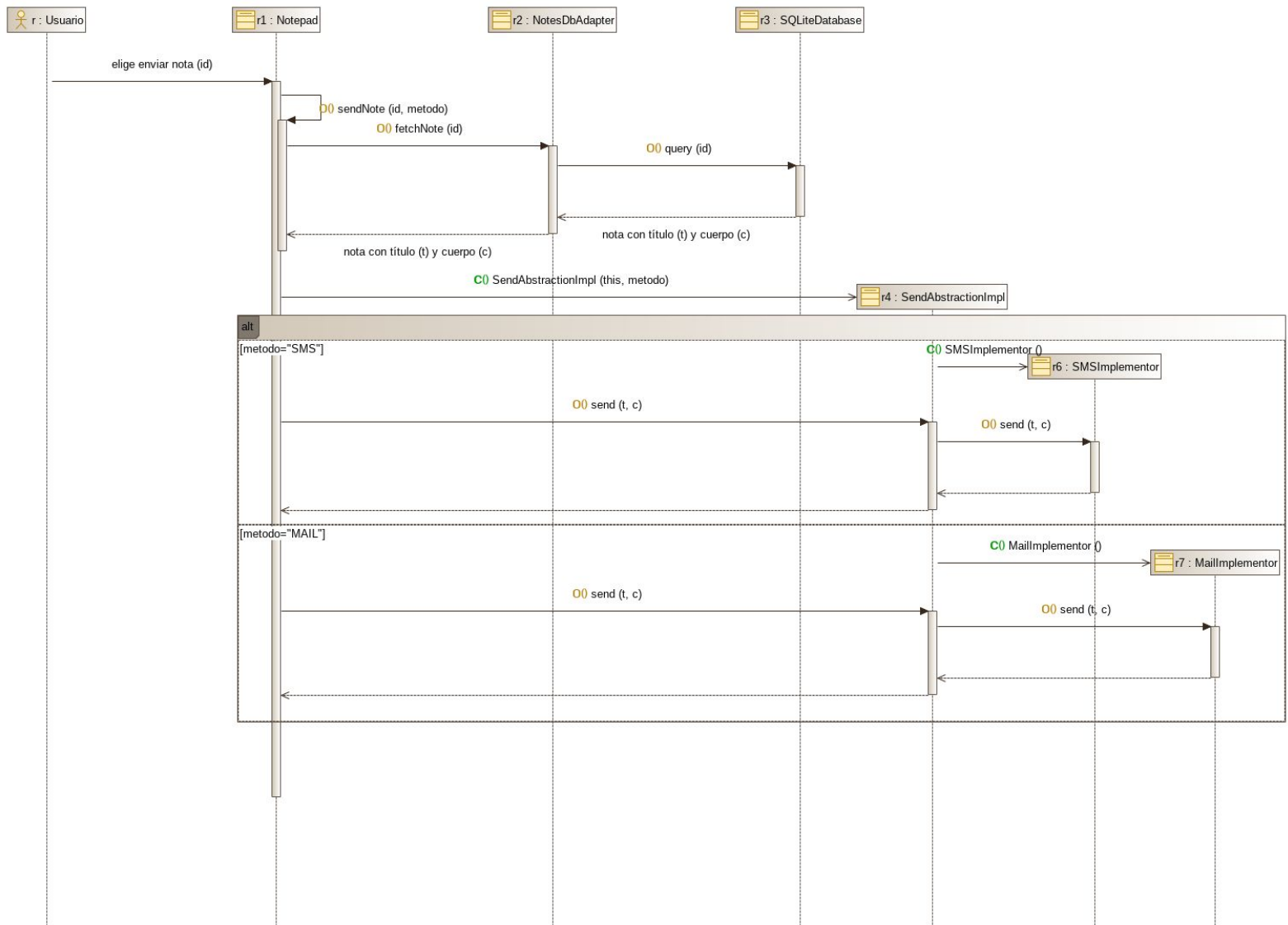
Ordenar por título o categoría



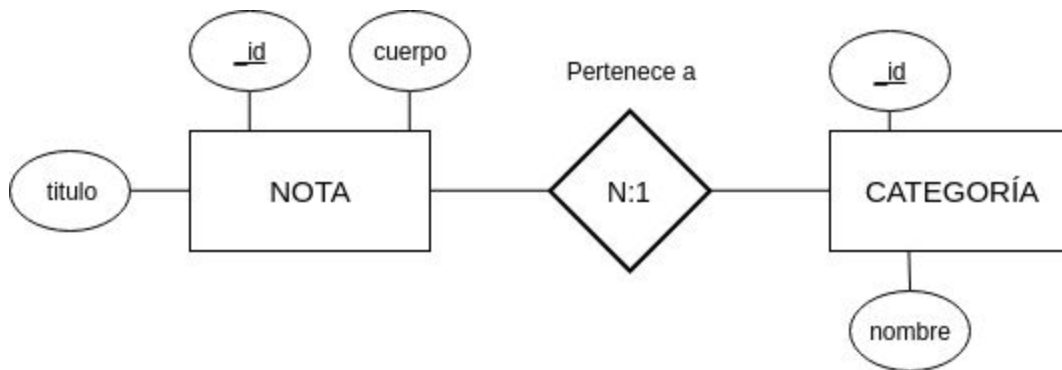
Filtrar por categoría



Enviar nota



Modelo entidad-relación de la base de datos



Pruebas

Pruebas de caja negra

Se han planteado pruebas para todos los métodos relativos a la creación, actualización y eliminación de notas y categorías de la base de datos.

createNote		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
title	1. title != null 2. title.length() > 0	3. title.length() == 0 4. title == null
body	5. body != null	6. body == null
categoryFk	7. categoryFk > 0 9. categoryFk == null	8. categoryFk <= 0

createNote				
Clases de equivalencia a probar	Entradas			Salida esperada
	title	body	categoryFk	
1, 2, 5 y 9	"TEST_NOTA_0"	"CuerpoTest"	null	id != -1
7	"TEST_NOTA_1"	"CuerpoTest"	idCategoria válido (> 0)	id != -1
3	""	"CuerpoTest"	null	-1
4	null	"CuerpoTest"	null	-1
6	"TEST_NOTA_"	null	null	-1
8	"TEST_NOTA_"	"CuerpoTest"	0	-1

updateNote		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
rowId	1. rowId > 0	2. rowId <= 0
title	3. title != null 4. title.length() > 0	5. title.length() == 0 6. title == null
body	7. body != null	8. body == null
categoryFk	9. categoryFk > 0 11. categoryFk == null	10. categoryFk <= 0

updateNote					
Clases de equivalencia a probar	Entradas				Salida esperada
	rowId	title	body	category Fk	
1, 3, 4, 7 y 11	idNota válido (> 0)	"TEST_NOTA_2"	"CuerpoTest2"	null	true
9	idNota válido (> 0)	"TEST_NOTA_2"	"CuerpoTest3"	idCategoría válido (> 0)	true
2	0	"TEST_NOTA_2"	"CuerpoTest2"	null	false
5	idNota válido (> 0)	""	"CuerpoTest2"	null	false
6	idNota válido (> 0)	null	"CuerpoTest2"	null	false
8	idNota válido (> 0)	"TEST_NOTA_2"	null	null	false
10	idNota válido (> 0)	"TEST_NOTA_2"	"CuerpoTest2"	0	false

deleteNote		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
rowId	1. rowId > 0	2. rowId <= 0

deleteNote		
Clases de equivalencia a probar	Entradas	Salida esperada
	rowId	
1	idNota válido (> 0)	true
2	0	false

createCategory		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
name	1. name != null 2. name.length() > 0	3. name.length() == 0 4. name == null

createCategory		
Clases de equivalencia a probar	Entradas	Salida esperada
	name	
1 y 2	"TEST_CATEGORIA_"	id != -1
3	""	-1
4	null	-1

updateCategory		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
rowId	1. rowId > 0	2. rowId <= 0
name	3. name != null 4. name.length() > 0	5. name.length() == 0 6. name == null

updateCategory			
Clases de equivalencia a probar	Entradas		Salida esperada
	rowId	name	
1, 3 y 4	idCategoria válido (> 0)	"TEST_CATEGORIA_"	true
2	0	"TEST_CATEGORIA_"	false
5	idCategoria válido (> 0)	""	false
6	idCategoria válido (> 0)	null	false

deleteCategory		
Parámetro de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
rowId	1. rowId > 0	2. rowId <= 0

deleteCategory		
Clases de equivalencia a probar	Entradas	Salida esperada
	rowId	
1	idCategoria válido (> 0)	true
2	0	false

Tras diseñar las pruebas de caja negra, se procedió a ejecutar por primera vez. Al hacerlo se detectó que podrían crearse notas y categorías con título vacío, lo que haría imposible su eliminación o modificación.

A su vez, se detectó también que si se creaba una nota o categoría con un título sin ningún carácter, resultaba imposible su eliminación o modificación.

Por último, al intentar crear desde la aplicación notas o categorías con títulos vacíos se detectó otro error, ya que al confirmar la creación/modificación se pasa a la pantalla principal pero sin haber creado nada, pudiendo confundir al usuario.

Al arreglar estos errores, todos los tests pasan correctamente.

Prueba de volumen

Para probar el nuevo requisito (No se espera que en la aplicación haya más de 1000 notas almacenadas), se implementó un test que crea 1000 notas, fallando si ha habido algún error en la creación de los mismos (`createNote` devuelve -1), o si salta alguna excepción.

También se desarrolló un método complementario que borra las 1000 notas del test anterior.

No hubo ningún problema en la ejecución de este test, pasa correctamente.

Prueba de usabilidad

Se comprobó que cuando hay muchas notas, al crear, editar o enviar una nota y volver a la lista de notas la nota sobre la que se realizó la última acción no estaba en pantalla, ya que la vista de la lista había sido actualizada, mostrando en pantalla el comienzo de la misma.

Para solucionarlo se forzó a la lista a mover la lista a la posición de la nota sobre la que se realizó la última acción de creación, edición o envío.

De igual modo pasaba con categorías, solucionándose de la misma manera.

Prueba de sobrecarga

Para probar el número de caracteres máximo en textos soportado por la aplicación, se desarrolló una prueba que fuese creando notas, incrementando la longitud de sus títulos de manera exponencial.

Al ejecutarse esta prueba desde la aplicación, el dispositivo virtual terminaba crasheando, no pudiendo acceder a LogCat ya que también crasheaba AndroidStudio. Para remediarlo, se decidió utilizar el framework de pruebas proporcionado por AndroidStudio, teniendo que eliminar los tests del menú de la aplicación. De esta manera se consiguió ejecutar la prueba correctamente, pudiendo recuperarse un estado óptimo de la aplicación tras su ejecución usando el método para eliminar notas de prueba desarrollado anteriormente desde el mismo IDE, sin tener que ejecutarlo desde la propia aplicación.

Se comprobó que la aplicación no fallaba por falta de espacio en disco para almacenar los títulos de las notas, sino que terminaba fallando por no poder asignar suficiente memoria a la String que representa el título. Por tanto, el número de caracteres máximo dependerá del dispositivo en el que se ejecute la aplicación. En nuestro caso, se llegó a 8388608 caracteres antes de quedarse sin memoria.

Pruebas de sistema

Requisito a validar	Validación
El usuario puede crear una nota.	Se pueden crear notas desde la pantalla principal de la aplicación.
Las notas constan de título y cuerpo.	Se puede introducir tanto título como cuerpo en cada nota.
El usuario puede modificar una nota existente.	Se puede modificar una nota existente desde la pantalla principal de la aplicación.
El usuario puede eliminar una nota existente.	Se puede modificar una nota existente desde la pantalla principal de la aplicación.

El usuario puede crear una categoría.	Se puede crear una categoría accediendo a la pantalla de categorías.
Una categoría está compuesta por letras y/o números, separadas entre sí por espacios en blanco.	Se puede introducir texto libre como nombre de la categoría.
El usuario puede modificar una categoría existente.	Se puede modificar una categoría desde la pantalla de categorías.
El usuario puede eliminar una categoría existente.	Se puede eliminar una categoría desde la pantalla de categorías.
El usuario puede asignar a una nota una categoría existente.	Se puede asignar una categoría existente a una nota desde la pantalla de edición/creación de notas, tras seleccionar una en concreto.
El usuario puede modificar la categoría asignada previamente a una nota.	Se puede modificar la categoría asignada previamente a una nota desde pantalla de edición/creación de notas, tras seleccionar una en concreto.
El usuario puede eliminar la categoría asignada previamente a una nota.	Se puede eliminar la categoría asignada previamente a una nota desde pantalla de edición/creación de notas, tras seleccionar una en concreto.
El usuario puede consultar el listado de notas ya creadas.	Se puede consultar el listado de notas desde la pantalla principal de la aplicación.
El usuario puede ordenar el listado de notas por categoría o título.	Se puede ordenar el listado de notas desde la pantalla principal de la aplicación.
El usuario puede mostrar el listado de notas filtrado por una categoría determinada.	Se puede filtrar el listado de notas por categorías desde la pantalla principal de la aplicación.
El usuario puede enviar las notas por SMS.	Se puede enviar una nota por SMS desde la pantalla principal de la aplicación.
El usuario puede enviar las notas por email.	Se puede enviar una nota por email desde la pantalla principal de la aplicación.

Resultados y conclusiones

Se ha aprendido a manejar con soltura herramientas CASE como Modelio e IDEs como AndroidStudio, las cuales han sido de extrema utilidad para el desarrollo y diseño de esta aplicación.

Se han valorado y comprendido las diferentes fases o etapas del desarrollo software, y por qué es importantes seguirlas y respetarlas si se desea desarrollar una aplicación de forma correcta.

La aplicación es bastante usable, entre otras cosas debido a que se ha intentado minimizar el número de acciones necesarias para realizar cualquier operación.

Un aspecto débil a mejorar sería la interfaz gráfica de la aplicación, ya que al no ser un requisito de la aplicación no se le ha dado demasiada atención.

Bibliografía

Documentación sobre Android: <https://developer.android.com/reference/>

Apuntes de la asignatura Ingeniería del Software