



---

## Modelo de casos de uso

Proyecto:

# HaramMusic

---

## Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
[Fecha]	[Rev]	[Descripción]	[Firma o sello]

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Fdo. D./Dña [Nombre]	Fdo. D./Dña [Nombre]



**haramMusic**

## Contenido

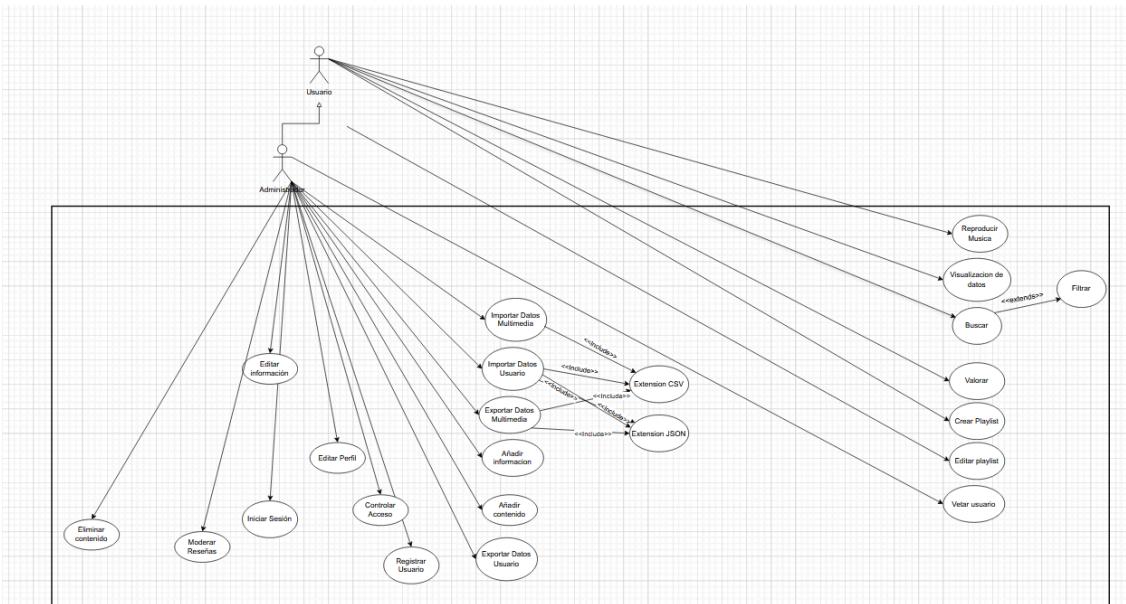
<b>HaramMusic.....</b>	<b>1</b>
1 Diagrama de casos de uso.....	3
2 Casos de uso.....	4
1. UC1. Iniciar sesión.....	5
2. UC2.Registrar usuario.....	6
3. UC3.Reproducir música.....	8
4. UC4.Reproducir música.....	10
5. UC5.Crear playlist.....	12
6. UC6.Editar playlist.....	14
7. UC7.Buscar y explorar canciones.....	16
8. UC8. Buscar y explorar playlist de otros usuarios.....	18
9. UC9. Buscar y explorar contenido con filtros especiales.....	20
10. UC10.Añadir canciones, artistas o álbumes.....	22
11. UC11.Eliminar canciones, artistas o álbumes.....	24
12. UC12.Editar información sobre las canciones, álbumes y artistas.....	26
13. UC13.Eliminar o editar reseñas inapropiadas.....	28
14. UC14.Controlar el acceso de usuarios.....	31
15. UC15.Editar perfil propio de usuario.....	34
16. UC16.Visualización de datos relacionados con las canciones.....	35
17. UC17.Importación de datos relativos a usuarios.....	37
18. UC18.Exportacion de datos relativos a usuarios.....	39
19. UC19.Importación de datos relativos al contenido multimedia.....	42
20. UC20. Exportación de datos relativos al contenido multimedia.....	44



## Modelo de casos de uso

Rev.  
Pág. 4

# 1 Diagrama de casos de uso





## 2 Casos de uso

### 1. UC1. Iniciar sesión

#### Caso de uso

- **Actor principal:** Usuario(Cliente/Administrador)
- **Personal involucrado / Intereses:**
  1. Usuario: acceder al sistema.
  2. Administrador: acceder al sistema.
- **Precondiciones**
  1. La aplicación debe estar iniciada.
  2. El usuario tiene cuenta registrada.
- **Garantías de éxito**
  1. El usuario queda autenticado y con sesión iniciada, pasa a la ventana principal de su área privada.
  2. Se registra el acceso en auditoría.
- **Escenario principal de éxito**
  1. El Usuario abre la ventana de Inicio de sesión.
  2. El Usuario introduce su usuario y contraseña.
  3. El Sistema valida credenciales.
  4. El Sistema redirige al usuario a la ventana principal de la aplicación
- **Extensiones**
  2. Campos vacíos / formato inválido:
    1. El Sistema muestra errores de validación y no procesa.
  3. Credenciales incorrectas:

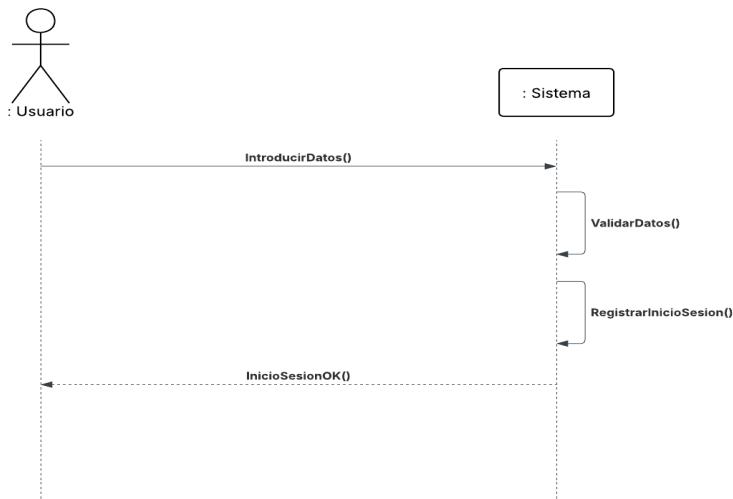
El Sistema rechaza el acceso, incrementa contador de fallos y muestra mensaje genérico. 2.

Tras N intentos fallidos, bloquea temporalmente la cuenta.
  3. Cuenta inactiva/bloqueada/no verificada:

El Sistema informa del estado.
- **Requisitos especiales**
- **Lista de tecnología y variaciones de datos**
- **Frecuencia**
  1. Alta.



## Diagrama de secuencia



## 2. UC2. Registrar usuario

### Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  1. Cliente: Crear un nuevo usuario.
  2. Administrador: Crear un nuevo administrador.
- **Precondiciones:**
  1. Aplicación en ejecución.
  2. Existencia de base de datos donde guardar el nuevo usuario.
- **Garantías de éxito(postcondiciones):**
  1. Si los campos se rellenan de forma válida el sistema redirige al usuario a la pantalla de inicio de sesión y registra al nuevo usuario en la base de datos.
  2. Si no son válidas, el sistema muestra un mensaje de error y vacía los campos para que el usuario vuelva a intentar el registro.
- **Escenario principal de éxito (flujo básico):**
  1. El usuario introduce los datos correspondientes a los campos y confirma.
  2. El sistema valida el registro



3. Si los campos se llenaron de forma válida el sistema redirige al usuario a la pantalla de inicio de sesión
4. El sistema registra el intento con logs introducidos en la base de datos.

- **Extensiones (Flujos alternativos):**

1. Nombre de usuario repetido:
  - i. El usuario introduce los datos correspondientes a los campos y confirma.
  - ii. El sistema valida el registro
  - iii. Si el nombre de usuario ya existe, el sistema muestra un mensaje de error y vacía los campos para que el usuario vuelva a intentar el registro.
2. Contraseña inválida:
  - i. El usuario introduce los datos correspondientes a los campos y confirma.
  - ii. El sistema valida el registro
  - iii. Si la contraseña no es válida, el sistema muestra un mensaje de error y vacía los campos para que el usuario vuelva a intentar el registro.
3. Campos en blanco:
  - i. El usuario introduce los datos correspondientes a los campos y confirma.
  - ii. El sistema valida el registro
  - iii. Si los campos están vacíos, el sistema muestra un mensaje de error y permite al usuario que vuelva a intentar el registro.

- **Requisitos especiales:**

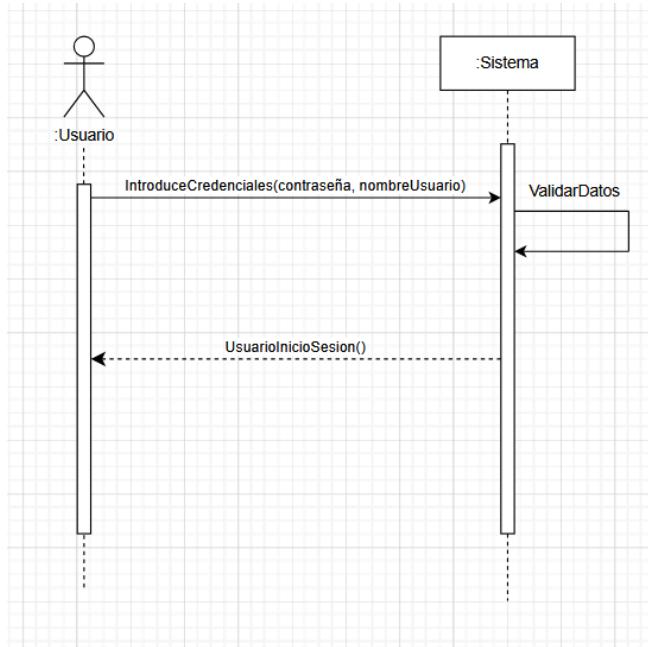
- **Lista de tecnología y variaciones de datos:**

- **Frecuencia:**

1. Media



## Diagrama de secuencia



### 3. UC3. Reproducir música

#### 2.3.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Cliente: escuchar canciones, pausar, continuar y cambiar de pista.
  - b. Sistema de audio / API Spotify: sincronizar y reproducir la música.
- **Precondiciones:**
  - a. Aplicación en ejecución
  - b. Usuario con sesión iniciada
  - c. Existen canciones en el catálogo
  - d. Conexión a internet
- **Garantías de éxito(postcondiciones):**
  - a. La canción seleccionada se reproduce o pausa según la acción del usuario.
- **Escenario principal de éxito (flujo básico):**
  1. El usuario selecciona una canción desde la interfaz.
  2. El sistema solicita el recurso a la API de Spotify.



**haramMusic**

3. El sistema reproduce la canción y muestra el reproductor con un botón para pausar o reanudar el track y datos sobre esta.
4. El usuario pulsa el botón de pausa para detener la reproducción.
5. El sistema pausa la canción manteniendo la posición actual.
6. El usuario pulsa play de nuevo.
7. El sistema reanuda la reproducción desde la posición guardada.
8. El sistema registra logs en los eventos de reproducción / pausa.

- **Extensiones (Flujos alternativos):**

- 2a. Canción no disponible:

- i. El sistema detecta que la canción ya no está disponible.
    - ii. El sistema muestra un mensaje de alerta y no inicia la reproducción.
    - iii. El sistema registra el evento en los logs.

- 2b. Error de conexión:

- i. El sistema no puede obtener el recurso.
    - ii. El sistema muestra un mensaje de error.

- 2c. El usuario selecciona otra canción mientras suena otra:

- i. El sistema detiene la reproducción de la canción actual.
    - ii. El sistema inicia la reproducción de la nueva canción seleccionada.
    - iii. El sistema registra el cambio en los logs.

- **Requisitos especiales:**

- a. Tiempo máximo de inicio de reproducción < 2 segundos desde la selección.
- b. Controles integrados con los controles del sistema Android (notificaciones, bloqueo de pantalla...)

- **Lista de tecnología y variaciones de datos:**

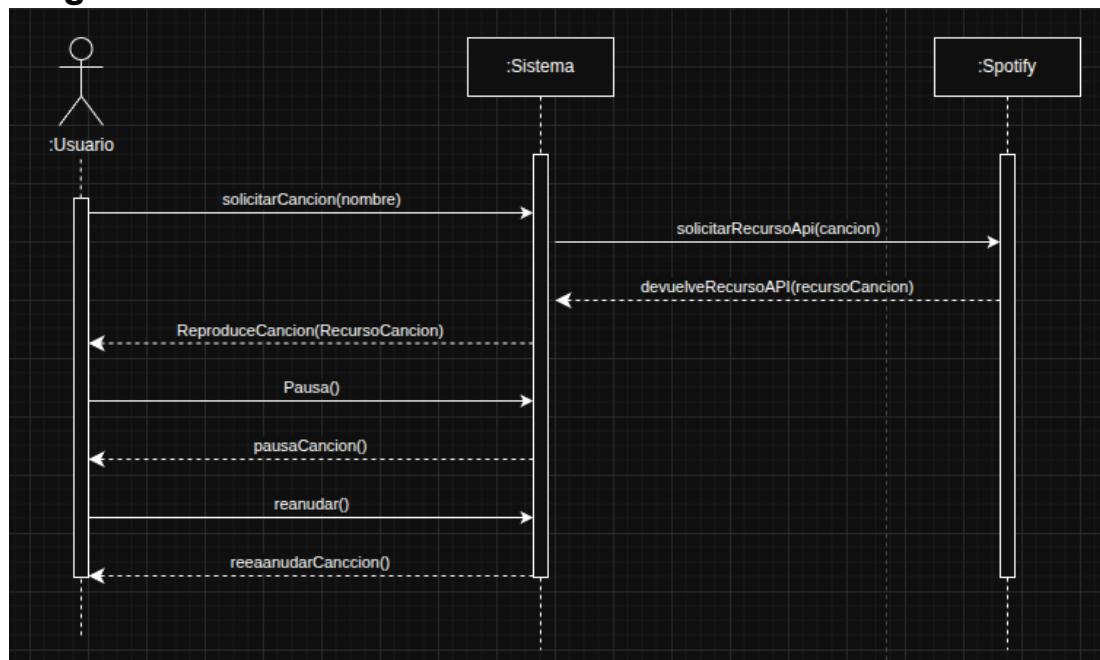
- a. Kotlin
- b. Spotify Web API

- **Frecuencia:**

- a. Alta



### 2.3.2 Diagrama de secuencia



## 4. UC4. Valorar contenidos

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: expresar su opinión sobre las canciones y álbumes.
  - b. Otros usuarios: consultar las valoraciones y reseñas.
  - c. Administrador: moderar reseñas.
- **Precondiciones:**
  - a. Usuario con sesión iniciada
  - b. Canción o álbum existente en la base de datos.
- **Garantías de éxito(postcondiciones):**
  - a. La valoración (1-5 estrellas) y la reseña de texto (si la hubiera) quedan registradas y asociadas al usuario y contenido.
  - b. Se actualiza la media de puntuación de la canción/álbum.
- **Escenario principal de éxito (flujo básico):**
  1. El usuario accede a la página de una canción o álbum



2. El usuario selecciona una puntuación entre 1 y 5 estrellas.
3. Opcionalmente, escribe una reseña de texto.
4. El usuario confirma el envío de la valoración.
5. El sistema valida la entrada (longitud máxima del texto, campos obligatorios)
6. El sistema registra la valoración y reseña en la base de datos, asociándola con el usuario y el contenido.
7. El sistema recalcula la puntuación media del contenido.
8. El sistema muestra la valoración registrada y un mensaje de confirmación.

- **Extensiones (Flujos alternativos):**

3a. Reseña demasiado larga:

- i. El sistema detecta que el texto supera el límite permitido.
- ii. El sistema muestra un mensaje de error indicando el límite.
- iii. El usuario puede volver a editar el texto y reenviar.

3b. (Cambiar numeración de la fukin lista) Usuario ya había valorado el contenido

- i. El sistema detecta una valoración previa de ese usuario para la canción/álbum.
- ii. El sistema muestra la valoración actual y permite modificarla.
- iii. Tras confirmar, el sistema actualiza la valoración anterior en lugar de crear una nueva.

3c. Error base de datos:

- i. El sistema no puede guardar la valoración
- ii. El sistema muestra un mensaje de error.
- iii. No se modifica la media ni se registra la valoración.

- **Requisitos especiales:**

- a. Cada usuario solo puede tener una valoración por contenido.
- b. Los administradores pueden modificar posteriormente las reseñas.

- **Lista de tecnología y variaciones de datos:**

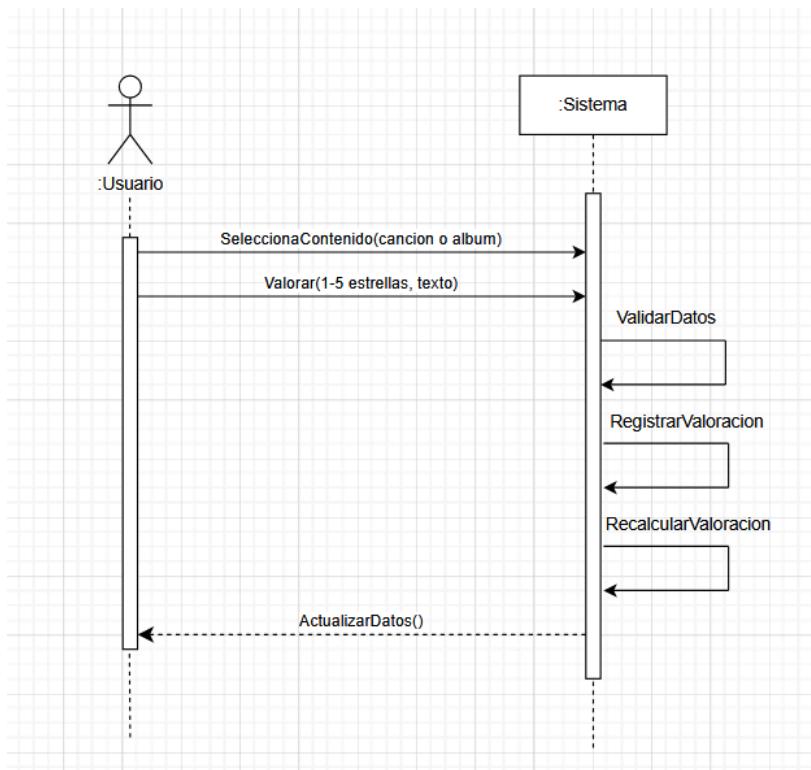
- a.



haramMusic

- **Frecuencia:**
  - a. Media

### 2.4.1 Diagrama de secuencia



## 5. UC5.Crear playlist

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: organizar canciones en listas personalizadas
- **Precondiciones:**
  - b. Usuario con sesión iniciada
- **Garantías de éxito(postcondiciones):**
  - a. La nueva playlist queda registrada, asociada a un usuario, con su nombre y contenido.
- **Escenario principal de éxito (flujo básico):**



**haramMusic**

1. El Usuario pulsa el botón “Nueva playlist” en la interfaz (pantalla principal o sección playlists).
2. El Sistema muestra un formulario para introducir el nombre de la playlist y seleccionar canciones iniciales (opcional).
3. El Usuario introduce el nombre de la playlist y selecciona una o varias canciones.
4. El Usuario confirma la creación.
5. El Sistema valida el nombre (no repetido para ese usuario, longitud, caracteres...).
6. El Sistema crea el registro de playlist en la base de datos asociada al usuario.
7. El Sistema asocia las canciones seleccionadas a la nueva playlist.
8. El Sistema muestra un mensaje de confirmación y la nueva playlist en la lista del usuario.

- **Extensiones (Flujos alternativos):**

- a. Playlist sin canciones
  - i. El usuario no selecciona canciones
  - ii. El sistema permite crear la playlist vacía solo con el nombre.
- b. Nombre de playlist duplicado
  - i. El sistema detecta que existe una playlist con el mismo nombre.
  - ii. El sistema muestra un mensaje de error y solicita otro nombre.
- c. Error al guardar en BD
  - i. El sistema no puede crear la playlist por un fallo técnico.
  - ii. Muestra un mensaje de error y no crea la playlist.

- **Requisitos especiales:**

- a. Nombre de playlist único por usuario
- b. Respuesta de creación en menos de 2 segundos.

- **Lista de tecnología y variaciones de datos:**

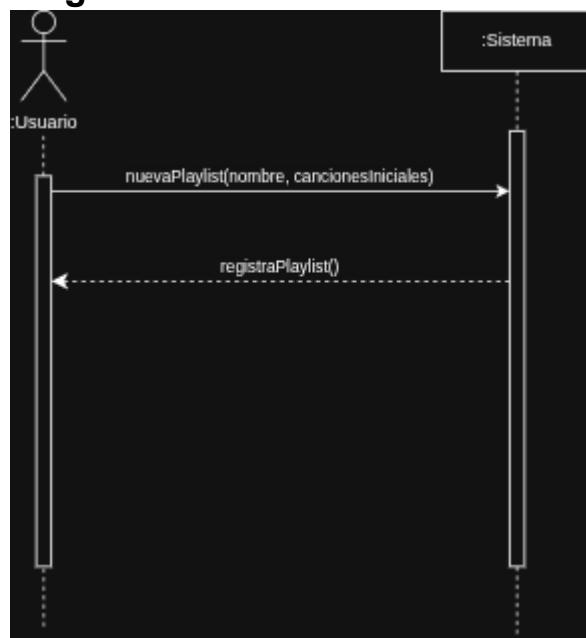
- a. Kotlin

- **Frecuencia:**

- a. Media



### 2.5.1 Diagrama de secuencia



## 6. UC6. Editar playlist

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: modificar, añadir o eliminar canciones de sus playlists.
- **Precondiciones:**
  - a. Usuario con sesión iniciada
  - b. Existencia de al menos una playlist propiedad del usuario.
- **Garantías de éxito(postcondiciones):**
  - a. Los cambios en la playlist (nombre, canciones) quedan guardados en la base de datos.
- **Escenario principal de éxito (flujo básico):**
  1. El Usuario accede a la lista de sus playlists.
  2. El Usuario selecciona una playlist y pulsa el botón “Editar”.
  3. El Sistema muestra la playlist con su nombre y las canciones actuales.



4. El Usuario modifica el nombre (opcional), añade nuevas canciones y/o elimina canciones existentes.
5. El Usuario confirma los cambios.
6. El Sistema valida el nuevo nombre (si ha cambiado) y las modificaciones.
7. El Sistema actualiza la playlist y sus canciones asociadas en la base de datos.
8. El Sistema muestra un mensaje de confirmación y la playlist actualizada.

- **Extensiones (Flujos alternativos):**

- a. El usuario elimina todas las canciones de la playlist:
  - i. El Sistema permite una playlist vacía.
  - ii. La playlist queda sin canciones pero sigue existiendo.
- b. Nuevo nombre duplicado para el mismo usuario:
  - i. El Sistema detecta duplicidad.
  - ii. El Sistema informa y solicita otro nombre o permite mantener el anterior.

- **Requisitos especiales:**

- a. Solo el propietario de la playlist puede editarla.

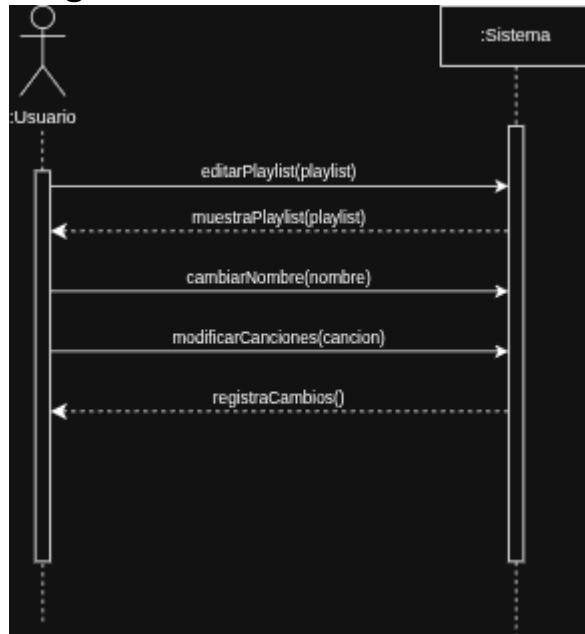
- **Lista de tecnología y variaciones de datos:**

- a.

- **Frecuencia:** Media



## 2.6.1 Diagrama de secuencia



## 7. UC7. Buscar y explorar canciones

### 2.7.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: encontrar canciones, artistas o álbumes de interés.
- **Precondiciones:**
  - a. Aplicación en ejecución.
  - b. Existencia de canciones, álbumes y artistas en la base de datos.
- **Garantías de éxito(postcondiciones):**
  - a. El usuario visualiza una lista de resultados que coinciden con los términos de búsqueda y filtros.
- **Escenario principal de éxito (flujo básico):**
  1. El Usuario accede a la pantalla de búsqueda.
  2. El Usuario introduce texto en la barra de búsqueda.
  3. El Usuario selecciona en un menú desplegable el tipo de resultado (canciones, álbumes, artistas). (Opcional)
  4. El Usuario confirma la búsqueda.



**haramMusic**

5. El Sistema consulta la base de datos según el texto y filtro seleccionado.

6. El Sistema muestra los resultados en una lista, con la información relevante (título, artista, álbum, etc.).

7. El Usuario puede seleccionar un elemento para ver más información o reproducirlo.

- **Extensiones (Flujos alternativos):**

- a. Sin resultados para la búsqueda

- i. El Sistema muestra un mensaje del tipo “No se han encontrado resultados”.

- b. Campos de búsqueda vacíos

- i. El Sistema puede mostrar resultados destacados o sugerencias por defecto, o no iniciar la búsqueda hasta que haya texto.

- **Requisitos especiales:**

- a. Búsquedas insensibles a mayúsculas y acentos.

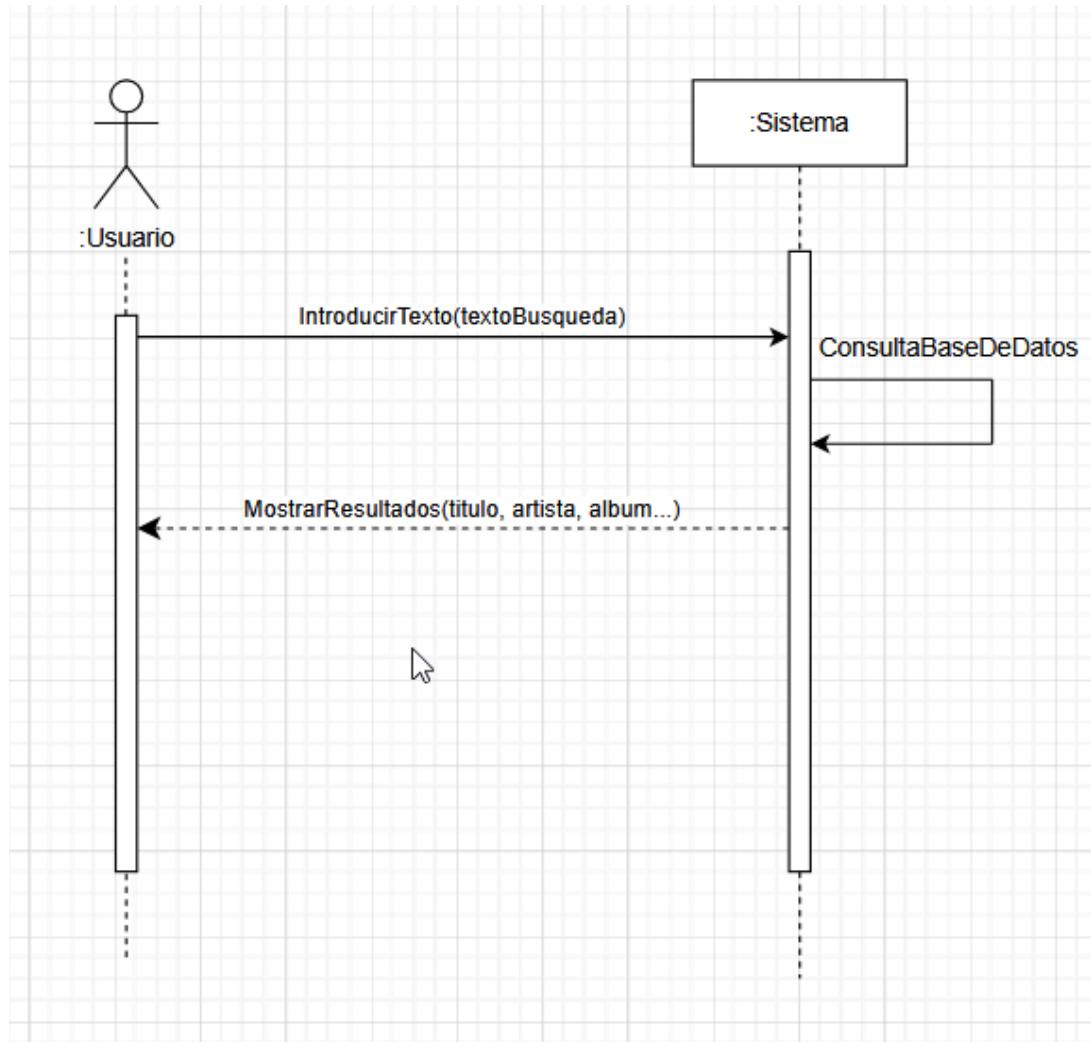
- b. Tiempo de respuesta < 1–2 segundos (según RNF de rendimiento).

- **Frecuencia:** Alta



haramMusic

## 2.7.2 Diagrama de secuencia



## 8. UC8. Buscar y explorar playlist de otros usuarios

### 2.8.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: descubrir playlists públicas de otros usuarios
- **Precondiciones:**
  - a. Usuario con sesión iniciada.
  - b. Existencia de playlists públicas en la base de datos.
  - c. Filtro “playlist” activado.



**haramMusic**

- **Garantías de éxito(postcondiciones):**
  - a. El usuario visualiza una lista de playlists públicas coincidentes con su búsqueda.
  - b. Puede acceder al detalle de una playlist concreta.
- **Escenario principal de éxito (flujo básico):**
  1. El Usuario accede a la pantalla de búsqueda y selecciona el filtro “Playlists”.
  2. El Usuario introduce texto en la barra de búsqueda (nombre de playlist, usuario, etc.).
  3. El Usuario confirma la búsqueda.
  4. El Sistema consulta la base de datos de playlists públicas que coincidan con el texto.
  5. El Sistema muestra la lista de resultados (nombre, creador, número de canciones, vista previa).
  6. El Usuario selecciona una playlist.
  7. El Sistema muestra el detalle de la playlist (canciones, creador, valoraciones...).
- **Extensiones (Flujos alternativos):**
  - a. Busqueda sin resultados:
    - i. El Sistema muestra “No se han encontrado playlists públicas con esos criterios”.
- **Requisitos especiales:**
  - a. Solo playlists marcadas como públicas son visibles.
- **Frecuencia:** Media

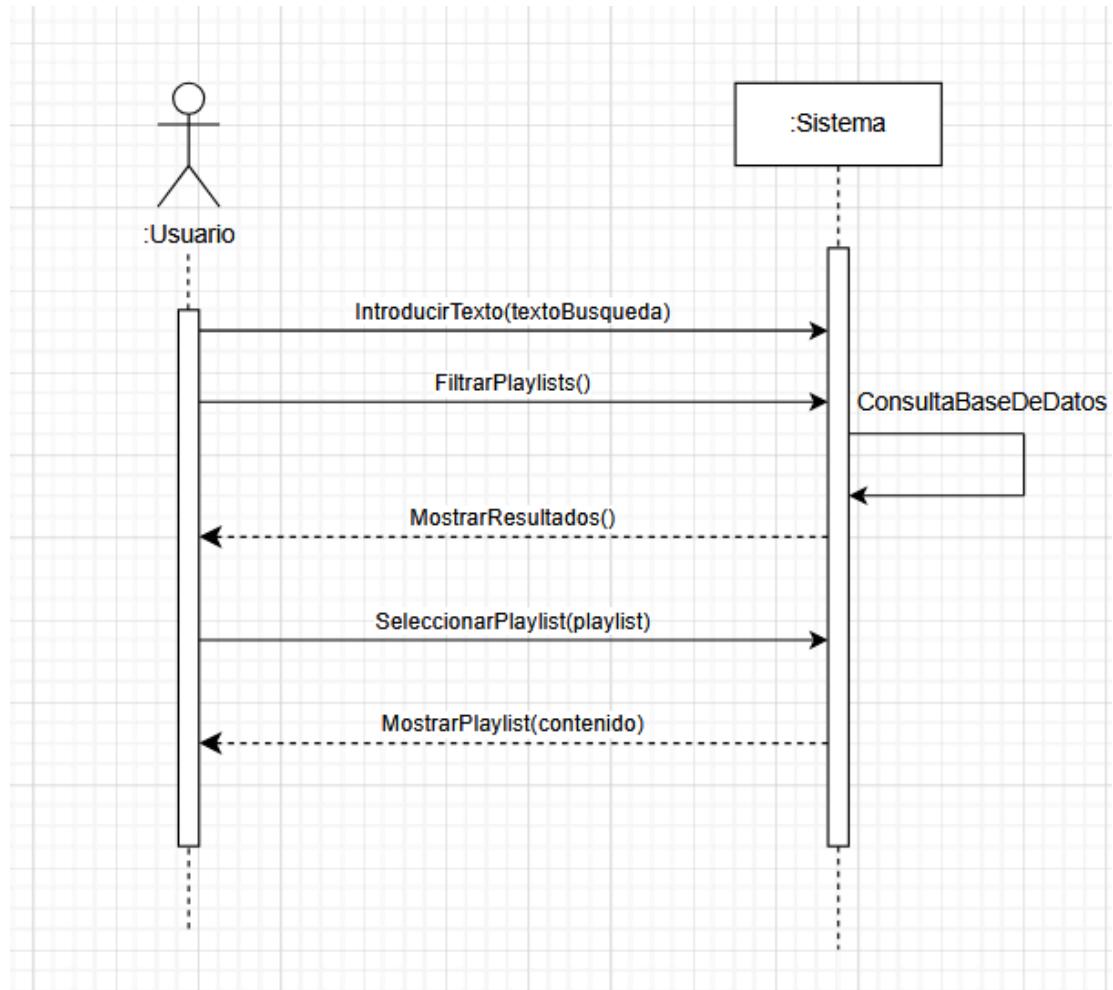


**haramMusic**

Modelo de casos de uso

Rev.  
Pág. 20

## 2.8.2 Diagrama de secuencia



## 9. UC9. Buscar y explorar contenido con filtros especiales

### 2.9.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuarios: refinar resultados de búsqueda según tipo de contenido.
- **Precondiciones:**



- a. Aplicación en ejecución.
- b. Existencia de canciones, álbumes y artistas.

- **Garantías de éxito(postcondiciones):**

- a. Los resultados de búsqueda se muestran filtrados según los criterios seleccionados.
- b. Los filtros utilizados quedan registrados (opcionalmente) en los logs.

- **Escenario principal de éxito (flujo básico):**

1. El Usuario accede a la pantalla de búsqueda.
2. El Usuario introduce texto en el campo de búsqueda.
3. El Usuario abre el menú desplegable de filtros y marca uno o varios criterios (canción, artista, álbum).
4. El Usuario confirma la búsqueda o la búsqueda se actualiza automáticamente.
5. El Sistema procesa la consulta aplicando los filtros seleccionados.
6. El Sistema muestra los resultados filtrados.

- **Extensiones (Flujos alternativos):**

- a. Sin ningún filtro activado
  - i. El Sistema interpreta que se deben mostrar todos los tipos de resultados.
- b. Búsqueda completa con muchos filtros
  - i. El Sistema puede paginar resultados o limitar su número para mantener el rendimiento.

- **Requisitos especiales:**

- a. Actualización de resultados en < 1 segundo tras cambiar filtros.
- b. Visibilidad clara de filtros activos.

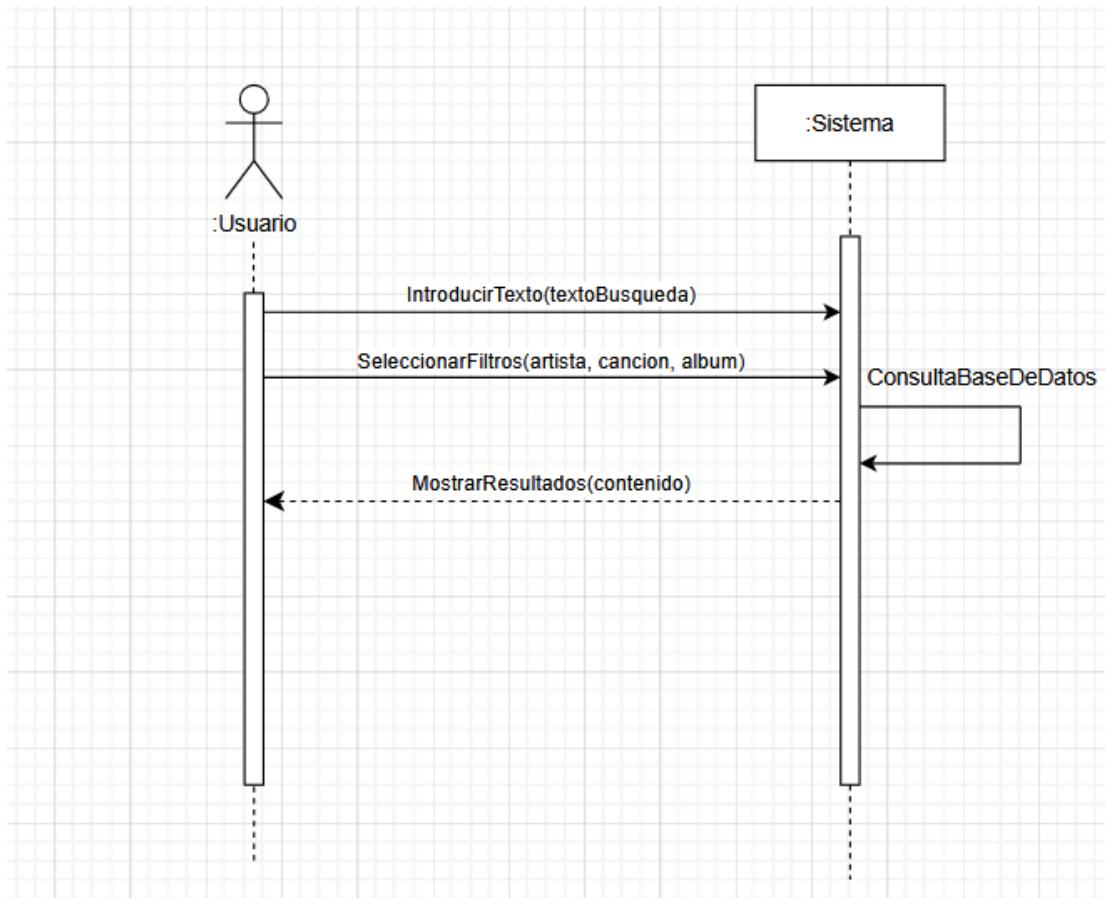
- **Lista de tecnología y variaciones de datos:**

- a.

- **Frecuencia:** Media-Alta



## 2.9.2 Diagrama de secuencia



## 10. UC10. Añadir canciones, artistas o álbumes

### 2.10.1 Casos de uso

- **Actor principal:** Administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: añadir canciones adicionales al catálogo.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
  - b. Conexión a la base de datos operativa.
- **Garantías de éxito(postcondiciones):**

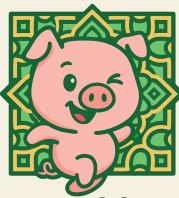


**haramMusic**

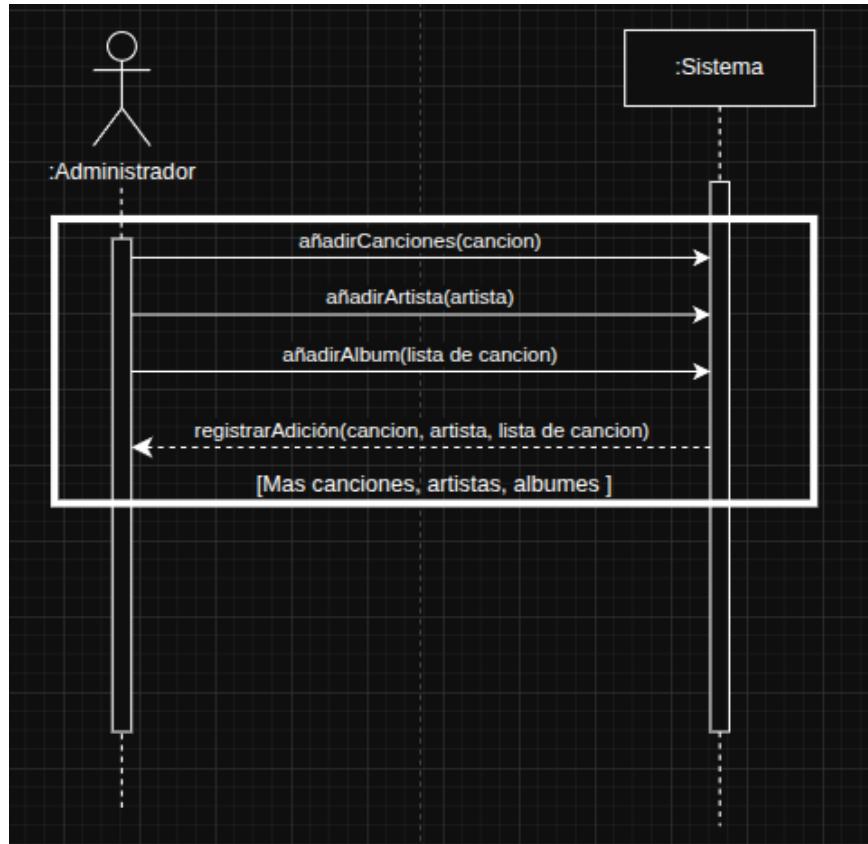
- a. El nuevo contenido (canción/álbum/artista) queda registrado y disponible en la aplicación.

- **Escenario principal de éxito (flujo básico):**
  1. El Administrador accede al panel de administración.
  2. El Administrador selecciona “Añadir contenido” (canción, artista o álbum).
  3. El Sistema muestra un formulario con los campos requeridos.
  4. El Administrador completa los campos (título, artista, álbum, género, fecha, archivo de audio, etc.) y confirma.
  5. El Sistema valida los campos.
  6. El Sistema guarda el nuevo registro en la base de datos.
  7. El Sistema muestra un mensaje de confirmación.
  8. El Sistema registra la operación en los logs.
- **Extensiones (Flujos alternativos):**
  - a. Datos incompletos o inválidos:
  - b. Duplicados (mismo título, artista y álbum):
- **Frecuencia:** Media

## 2.10.2      Diagrama de secuencia



haramMusic



## 11. UC11. Eliminar canciones, artistas o álbumes

### 2.11.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: eliminar canciones adicionales al catálogo.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
  - b. Conexión a la base de datos operativa.
- **Garantías de éxito(postcondiciones):**
  - a. El contenido (canción/álbum/artista) queda eliminado y no disponible en la aplicación.
- **Escenario principal de éxito (flujo básico):**
  1. El Administrador accede al panel de administración.



**haramMusic**

2. El Administrador selecciona “Eliminar contenido” (canción, artista o álbum).
3. El Sistema muestra un mensaje de confirmación.
4. El Administrador confirma.
5. El Sistema elimina el recurso en la base de datos.
6. El Sistema muestra un mensaje de confirmación de la eliminación.
7. El Sistema registra la operación en los logs.

- **Extensiones (Flujos alternativos):**

- a. El Administrador no confirma la eliminación en el mensaje de confirmación:
  - i. El administrador es redirigido a la pantalla anterior
  - ii. El sistema sigue como estaba

- **Requisitos especiales:**

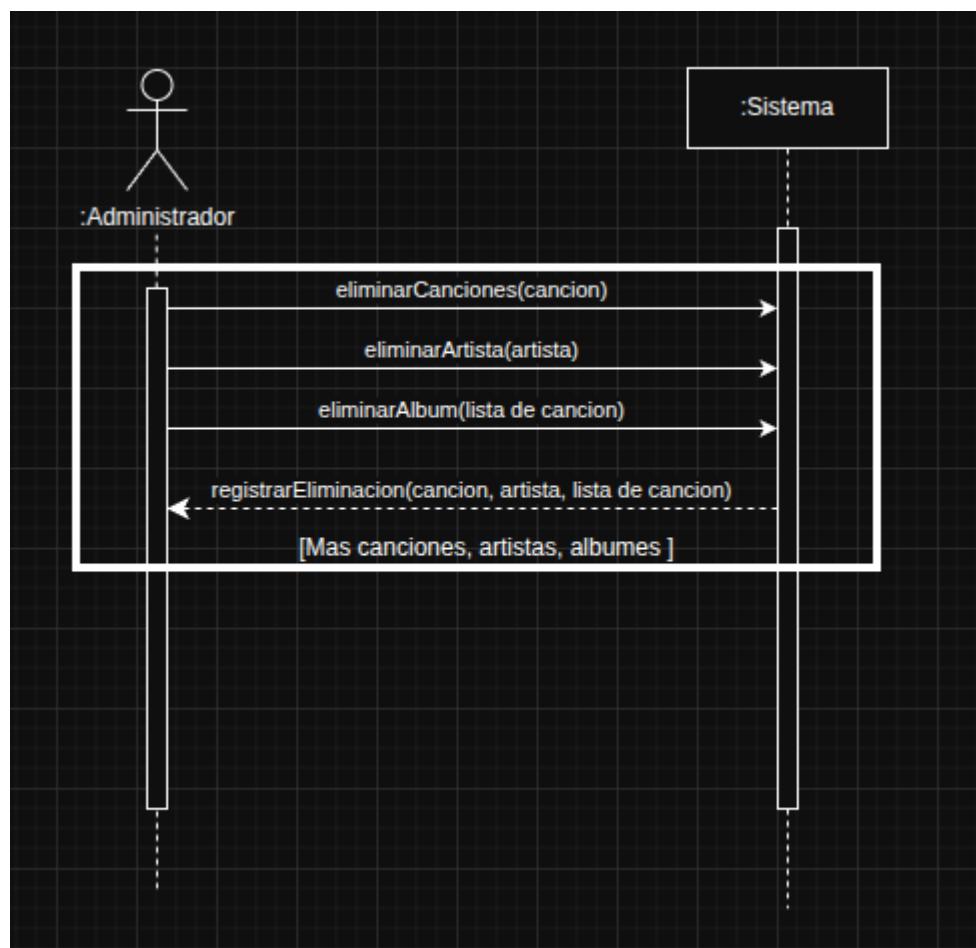
- a. Existencia del contenido a eliminar

- **Frecuencia:** Media



**haramMusic**

### 2.11.2 Diagrama de secuencia



## 12. UC12. Editar información sobre las canciones, álbumes y artistas

### 2.12.1 Casos de uso

- **Actor principal:** Administrador
- **Personal involucrado/intereses:**
  - a. Administrador: ampliar y mejorar la información descriptiva.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.



**haramMusic**

- b. Canción/álbum/artista existente.

- **Garantías de éxito(postcondiciones):**

- a. Los campos de descripción e información adicional se actualizan y son visibles en la app.

- **Escenario principal de éxito (flujo básico):**

1. El Administrador accede a la ficha de una canción, álbum o artista.
2. El Administrador pulsa el botón “Editar información”.
3. El Sistema muestra los campos de texto de descripciones e información asociada.
4. El Administrador modifica o añade información (biografía, notas, créditos, etc.).
5. El Administrador confirma los cambios.
6. El Sistema valida longitud/formato de los textos.
7. El Sistema guarda la nueva información en la base de datos.
8. El Sistema muestra un mensaje de confirmación.

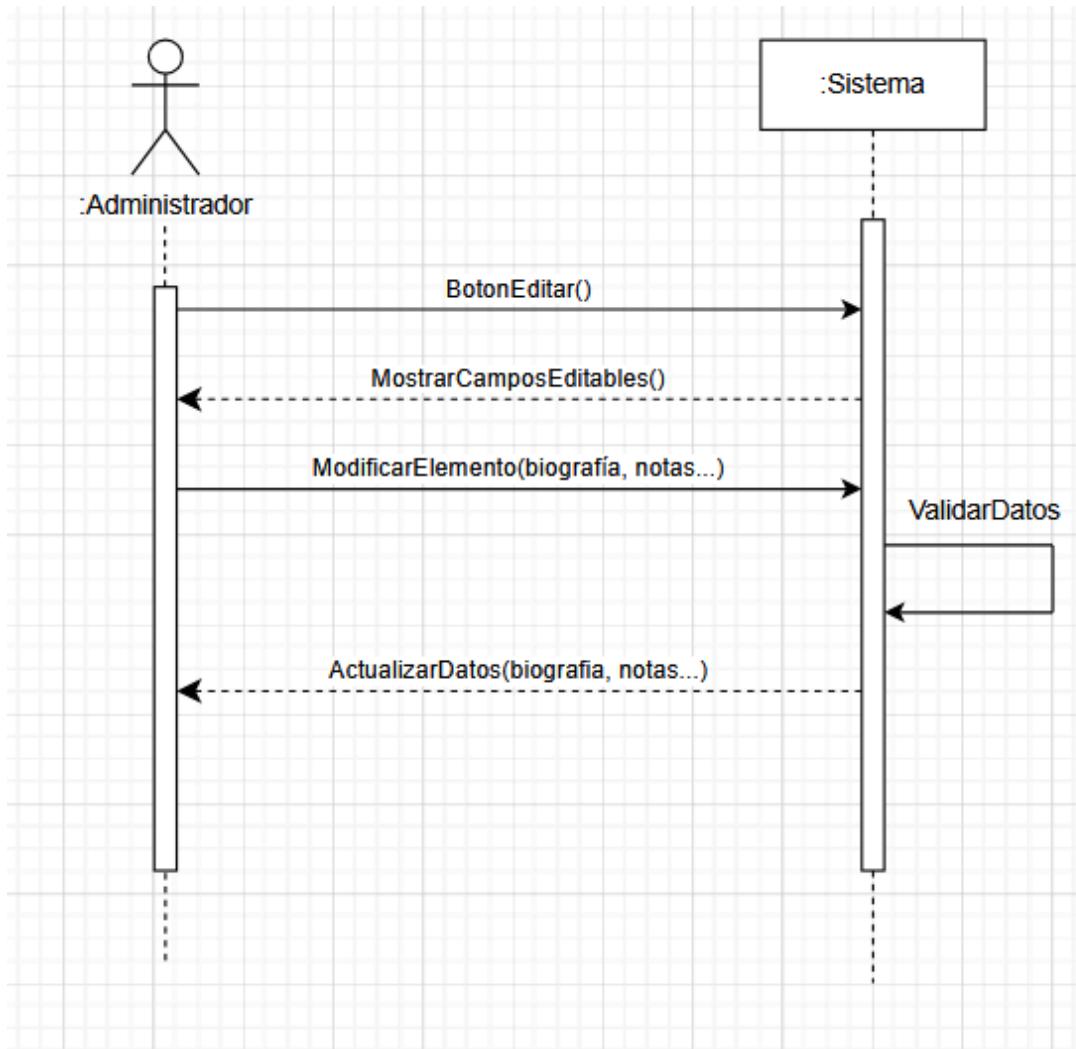
- **Extensiones (Flujos alternativos):**

- a. Texto demasiado largo / formato invalido
    - i. El Sistema informa del error y no guarda los cambios.

- **Frecuencia:** baja



## 2.12.2 Diagrama de secuencia



## 13. UC13.Eliminar o editar reseñas inapropiadas.

### 2.13.1 Casos de uso

- Actor principal: Administrador
- Personal involucrado/intereses:
  - a. Administrador: mantener el contenido de reseñas dentro de normas éticas.
- Precondiciones:



- a. Sesión iniciada como administrador.
- b. Existencia de reseñas publicadas en la base de datos.

- **Garantías de éxito(postcondiciones):**

- a. La reseña inapropiada queda editada o eliminada.
  - b. Se registra la acción en logs de moderación, guardando el texto original si procede.

- **Escenario principal de éxito (flujo básico):**

1. El Administrador accede al panel de moderación de reseñas.
2. El Sistema muestra una lista de reseñas recientes o reportadas.
3. El Administrador selecciona una reseña inapropiada.
4. El Administrador elige “Editar” o “Eliminar”.
5. Si selecciona “Editar”, modifica el texto para adaptarlo a las normas.
6. El Administrador confirma la acción.
7. El Sistema actualiza o elimina la reseña en la base de datos.
8. El Sistema registra la acción (tipo, autor admin, fecha, reseña afectada).
9. El Sistema deja de mostrar reseñas eliminadas a los usuarios.

- **Extensiones (Flujos alternativos):**

- a. Confirmacion de eliminacion
  - i. El Sistema pide confirmación antes de borrar definitivamente.

- **Requisitos especiales:**

- a. Conservación de copia del texto original para auditoría interna.

- **Lista de tecnología y variaciones de datos:**

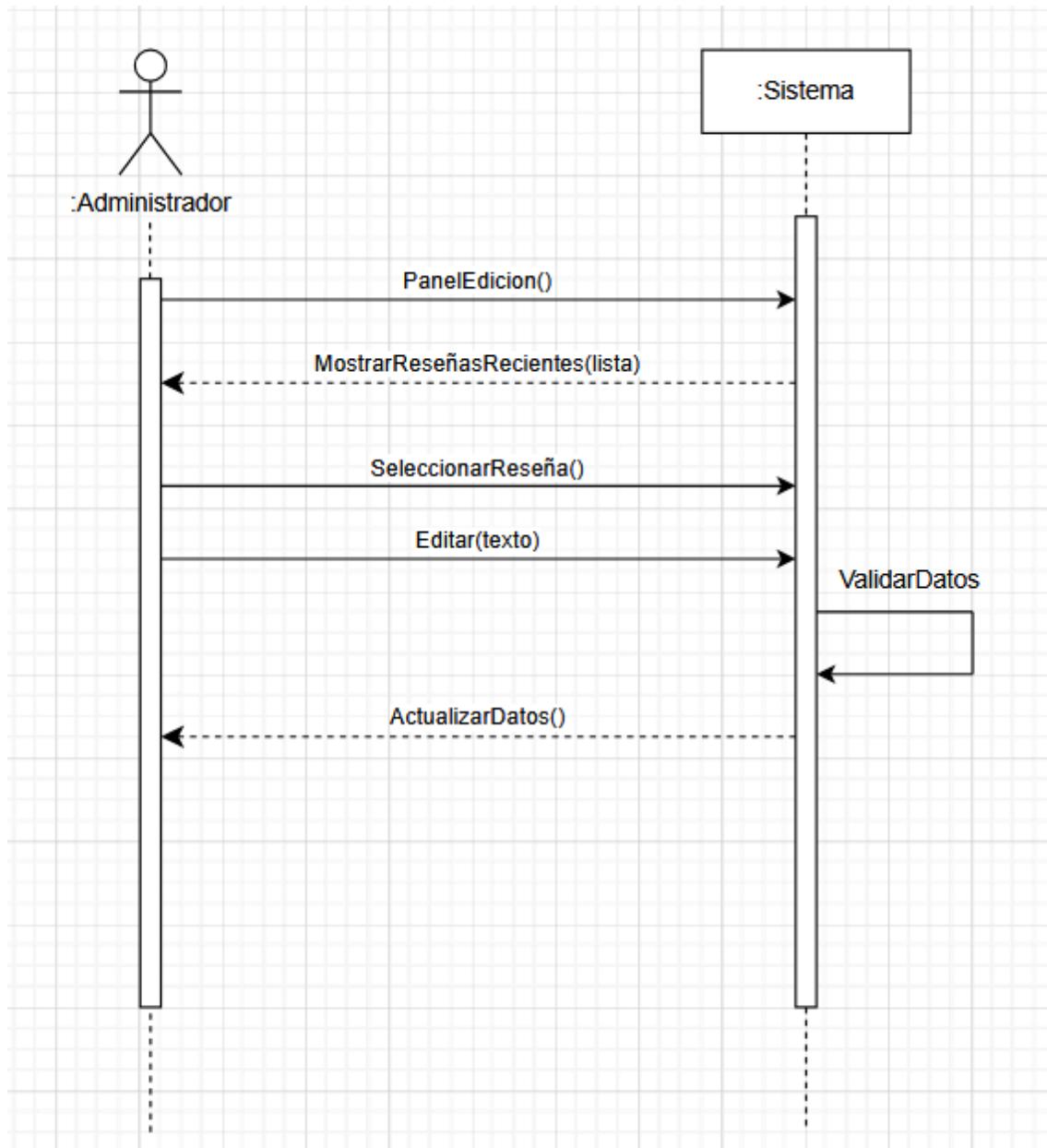
- a. Tabla de reseñas con estados (visible, moderada, eliminada).

- **Frecuencia:** baja



haramMusic

### 2.13.2 Diagrama de secuencia



### 14. UC14. Controlar el acceso de usuarios



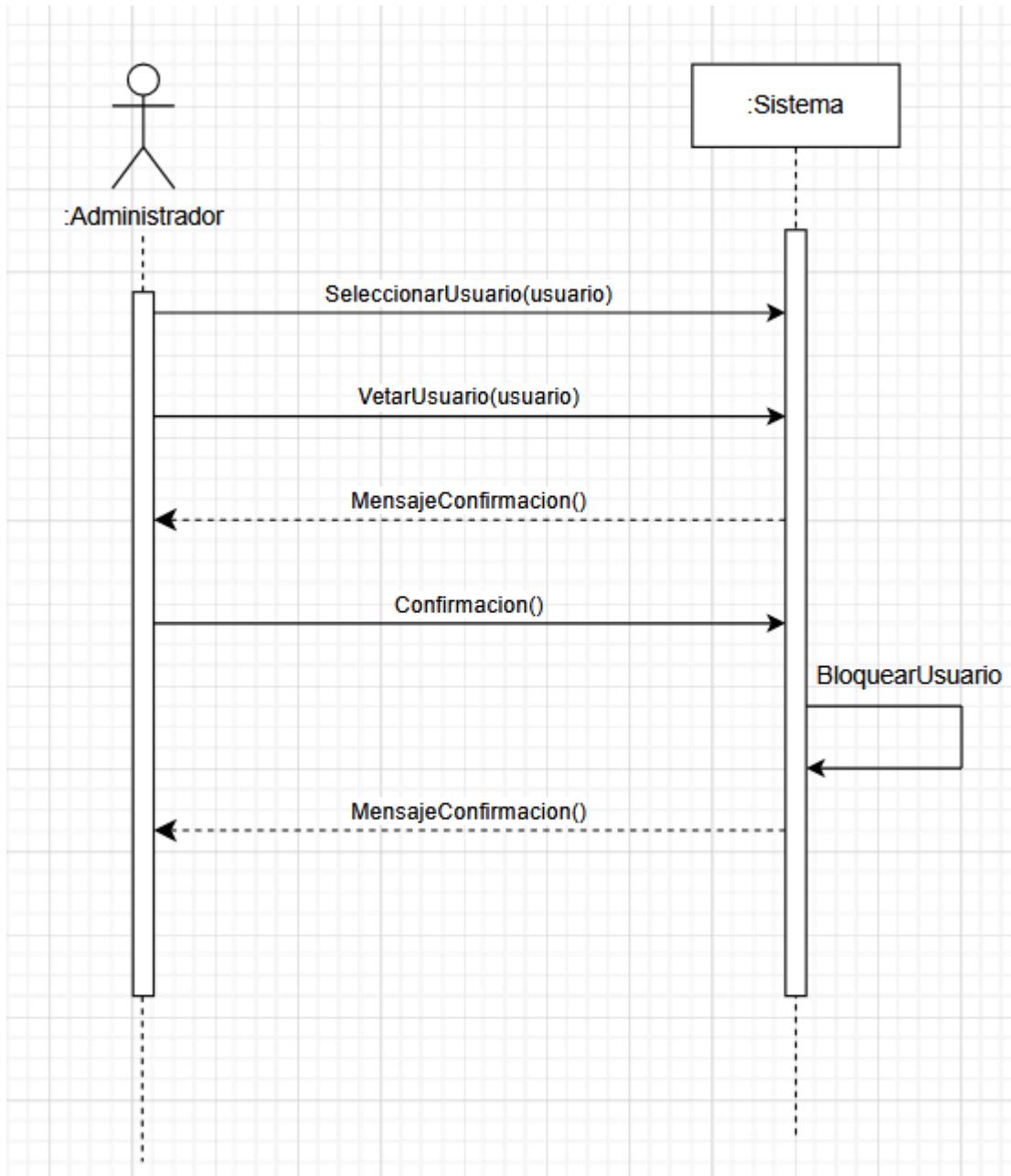
### **2.14.1 Casos de uso**

- **Actor principal:** Administrador
- **Personal involucrado/intereses:**
  - a. Administrador: bloquear usuarios problemáticos.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
  - b. Usuario objetivo existente en la base de datos.
- **Garantías de éxito(postcondiciones):**
  - a. El usuario queda marcado como vetado/bloqueado.
  - b. El usuario no puede iniciar sesión ni utilizar la aplicación.
- **Escenario principal de éxito (flujo básico):**
  1. El Administrador accede al listado o buscador de usuarios.
  2. El Administrador selecciona un usuario problemático.
  3. El Administrador pulsa el botón “Vetar/Bloquear usuario”.
  4. El Sistema muestra una ventana de confirmación.
  5. El Administrador confirma la acción.
  6. El Sistema marca al usuario como bloqueado en la base de datos.
  7. El Sistema impide futuros inicios de sesión de ese usuario.
- **Extensiones (Flujos alternativos):**
  - a. El administrador cancela en la ventana de confirmación:
    - i. El sistema no realiza cambios
  - b. Usuario con sesión actualmente iniciada:
    - i. El sistema puede forzar cierre de sesión en el siguiente acceso a servidor o en la siguiente acción crítica.
- **Requisitos especiales:**
  - a. Existencia del usuario al que se pretende sancionar.
- **Lista de tecnología y variaciones de datos:**
  - a. Sistema de vetos activo.
- **Frecuencia:** Baja



haramMusic

## 2.14.2 Diagrama de secuencia





## 15. UC15.Editar perfil propio de usuario

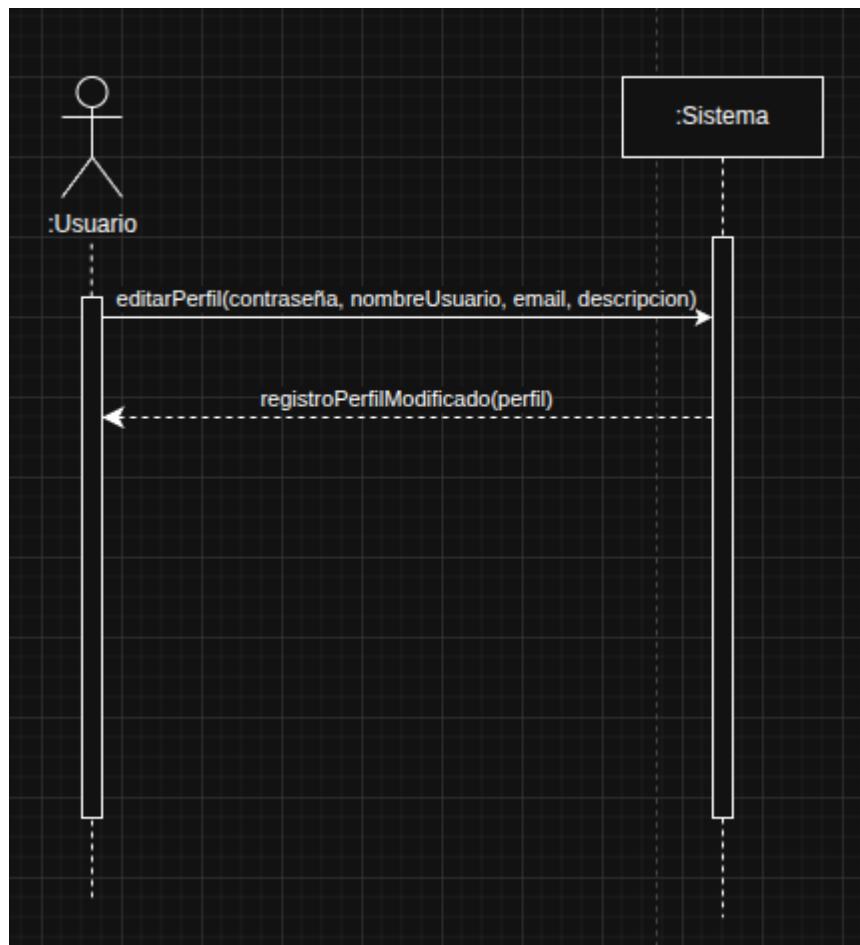
### 2.15.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuarios: mantener su perfil actualizado (nombre visible, email, foto, etc.).
- **Precondiciones:**
  - a. Sesión iniciada.
- **Garantías de éxito(postcondiciones):**
  - a. Los datos de perfil del usuario quedan actualizados en la base de datos.
- **Escenario principal de éxito (flujo básico):**
  1. El Usuario accede a la pantalla “Mi perfil”.
  2. El Sistema muestra la información actual del perfil.
  3. El Usuario pulsa el botón “Editar”.
  4. El Sistema permite modificar campos editables (nombre visible, email, foto, descripción, etc.).
  5. El Usuario guarda los cambios.
  6. El Sistema valida los campos (formatos, longitud...).
  7. El Sistema guarda los cambios en la base de datos.
  8. El Sistema muestra un mensaje de confirmación.
- **Extensiones (Flujos alternativos):**
  - a. Campos inválidos:
    - i. El Sistema muestra mensajes de error y no guarda los cambios.
- **Frecuencia:** Media



**haramMusic**

## 2.15.2 Diagrama de secuencia



## 16. UC16. Visualización de datos relacionados con las canciones

### 2.16.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Usuario: conocer información detallada sobre la canción.
- **Precondiciones:**
  - a. Canción existente en la base de datos.
- **Garantías de éxito(postcondiciones):**



**haramMusic**

- a. El usuario ve información completa (artista, álbum, fecha, género, etc.) en la interfaz.

- **Escenario principal de éxito (flujo básico):**

1. El Usuario selecciona una canción desde cualquier listado o reproductor.
2. El Usuario abre la ficha de detalle de la canción.
3. El Sistema recupera de la base de datos la información asociada (artista, álbum, fecha, géneros, etc.).
4. El Sistema muestra la información en pantalla.

- **Extensiones (Flujos alternativos):**

- a. Datos incompletos en la base de datos:
  - i. El Sistema muestra solo la información disponible y, opcionalmente, un texto indicándolo.

- **Requisitos especiales:**

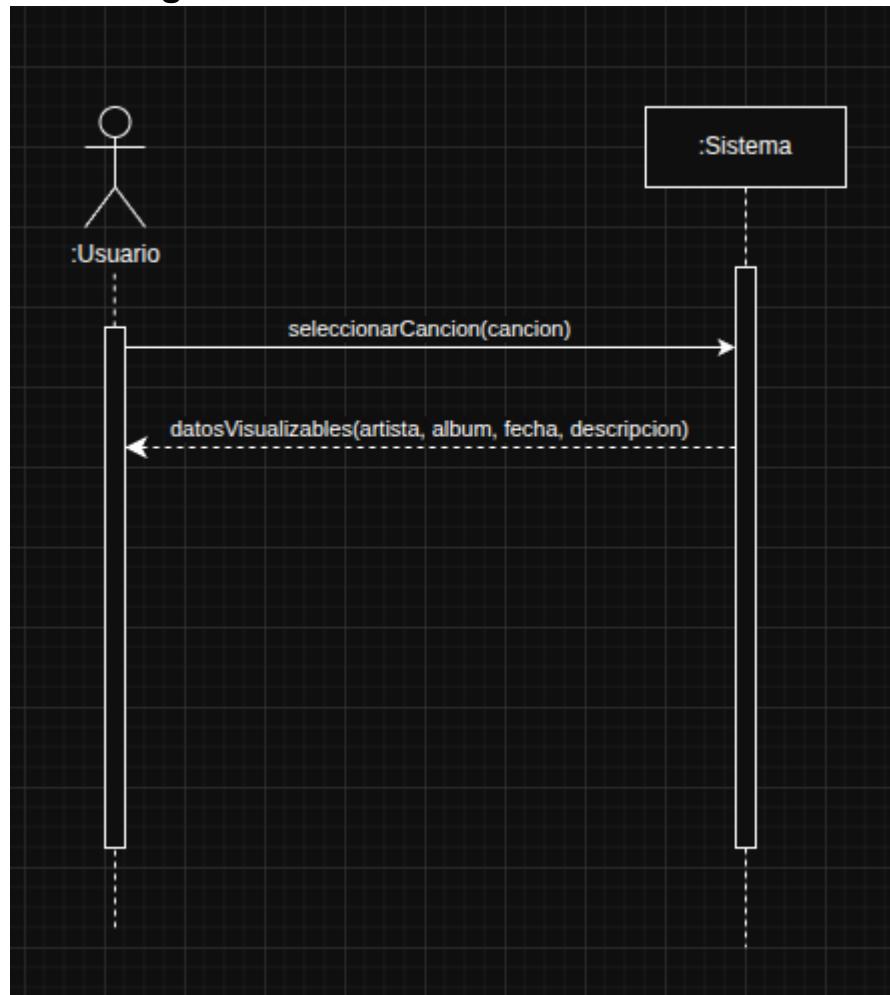
- a. Información legible y ordenada; posible enlace a fichas de artista/álbum.

- **Frecuencia:** Alta



haramMusic

## 2.16.2 Diagrama de secuencia



## 17. UC17.Importación de datos relativos a usuarios

### 2.17.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: cargar en bloque usuarios, playlists, reseñas, etc.
  - b. Usuarios: migrar datos
- **Precondiciones:**
  - a. Sesión iniciada como administrador.



- b. Archivo .CSV o .JSON con formato correcto accesible

- **Garantías de éxito(postcondiciones):**

- a. Los datos válidos del archivo se importan correctamente y se integran en la base de datos.
- b. Se genera un informe o mensaje de resumen (importados, errores).

- **Escenario principal de éxito (flujo básico):**

1. El Administrador accede al menú de importación de usuarios.
2. El Administrador selecciona un archivo .CSV/.JSON desde el dispositivo.
3. El Sistema valida el formato del archivo y su estructura.
4. El Sistema procesa el archivo e importa los usuarios y sus datos (perfil, canciones valoradas, reseñas, playlists).
5. El Sistema muestra un resumen de la importación (nº registros importados, posibles advertencias).

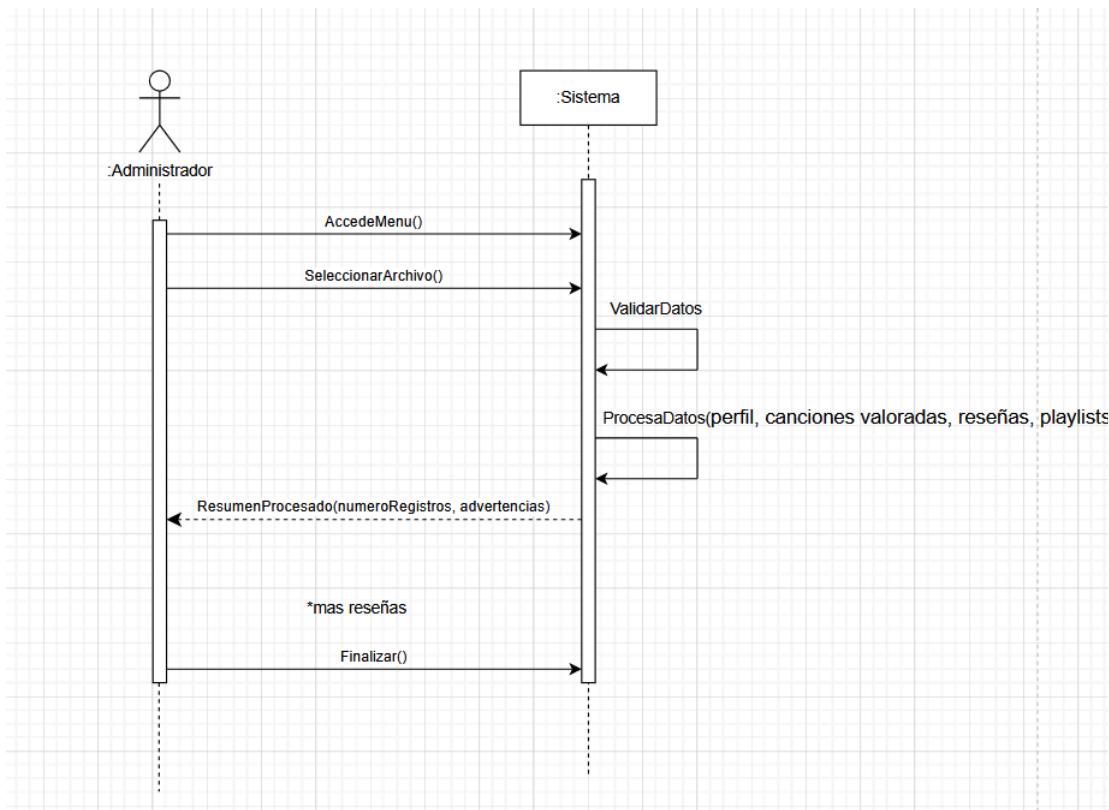
- **Extensiones (Flujos alternativos):**

- a. Formato de archivos incorrecto:
  - i. El Sistema muestra un mensaje de error y no importa nada.
- b. Algunos registros con errores:
  - i. El Sistema importa los registros correctos y marca los erróneos en un informe.

- **Frecuencia:** baja



## 2.17.2 Diagrama de secuencia



## 18. UC18.Exportacion de datos relativos a usuarios

### 2.18.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: obtener copias o informes de datos de usuarios.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
- **Garantías de éxito(postcondiciones):**
  - a. Se genera un archivo .CSV o .JSON con los datos de usuarios deseados y se descarga/guarda correctamente.
- **Escenario principal de éxito (flujo básico):**



**haramMusic**

1. El Administrador accede al menú de exportación de datos de usuarios.
2. El Administrador selecciona el formato de salida JSON y el rango de datos (todos o filtrados).
3. El Administrador confirma la exportación.
4. El Sistema consulta la base de datos y genera el archivo.
5. El Sistema ofrece el archivo para descargar o guardar en el dispositivo.

- **Extensiones (Flujos alternativos):**

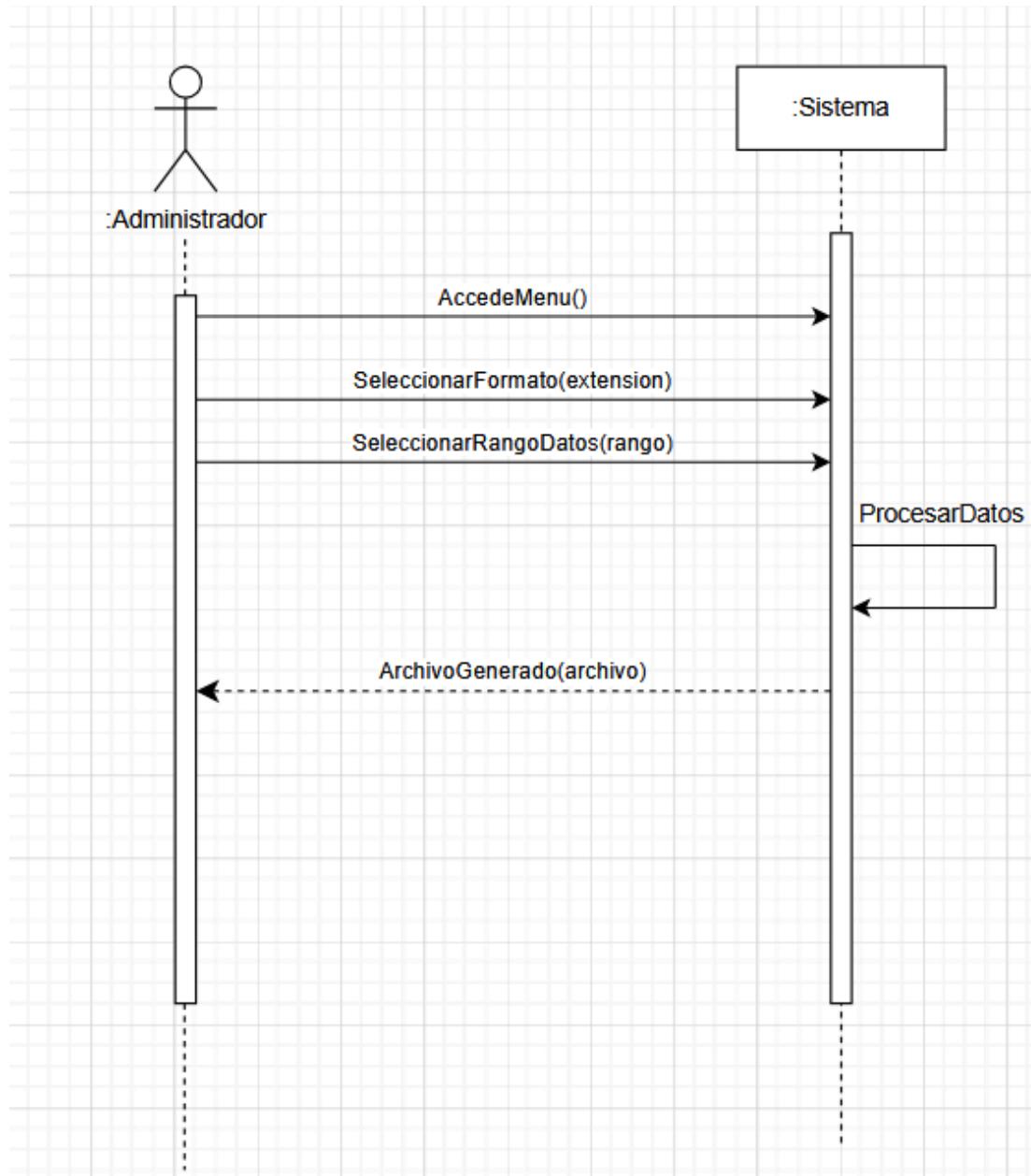
- a. Exportar a CSV:
  - i. El Administrador accede al menú de exportación de datos de usuarios.
  - ii. El Administrador selecciona el formato de salida CSV y el rango de datos (todos o filtrados).
  - iii. El Administrador confirma la exportación.
  - iv. El Sistema consulta la base de datos y genera el archivo.
  - v. El Sistema ofrece el archivo para descargar o guardar en el dispositivo.

- **Frecuencia:** baja



haramMusic

## 2.18.2 Diagrama de secuencia





## 19. UC19.Importación de datos relativos al contenido multimedia

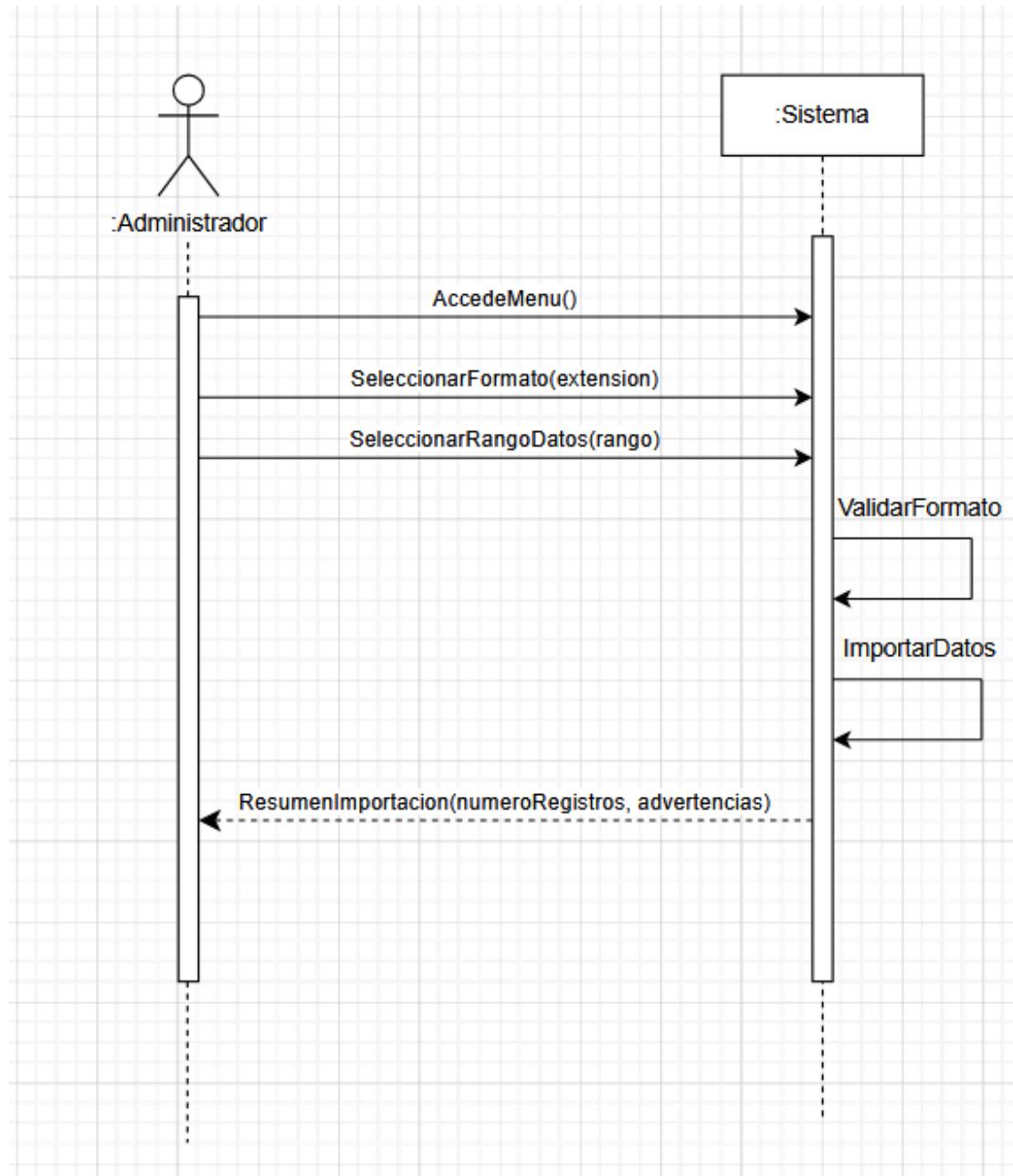
### 2.19.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: cargar grandes catálogos de contenido multimedia (canciones, álbumes, artistas).
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
  - b. Archivo .CSV/.JSON válido con datos de contenido multimedia.
- **Garantías de éxito(postcondiciones):**
  - a. El contenido multimedia del archivo se integra en la base de datos y pasa a estar disponible en la app.
- **Escenario principal de éxito (flujo básico):**
  1. El Administrador accede al menú de “Importar contenido multimedia”.
  2. El Administrador selecciona el archivo CSV/JSON de contenidos.
  3. El Sistema valida formato y campos obligatorios.
  4. El Sistema importa los registros válidos (canciones, álbumes, artistas).
  5. El Sistema muestra un resumen de la importación.
- **Extensiones (Flujos alternativos):**
  - a. Error al importar
- **Requisitos especiales:**
  - a.
- **Frecuencia:** baja



haramMusic

## 2.19.2 Diagrama de secuencia





## 20. UC20. Exportación de datos relativos al contenido multimedia

### 2.20.1 Casos de uso

- **Actor principal:** Usuario(cliente/administrador)
- **Personal involucrado/intereses:**
  - a. Administrador: obtener copias o migrar el catálogo a otros sistemas.
- **Precondiciones:**
  - a. Sesión iniciada como administrador.
- **Garantías de éxito(postcondiciones):**
  - a. Se genera un archivo CSV/JSON con información del contenido multimedia (canciones, álbumes, artistas).
- **Escenario principal de éxito (flujo básico):**
  1. El Administrador accede al menú de “Exportar contenido multimedia”.
  2. El Administrador selecciona el formato JSON y el alcance (todo catálogo o filtrado).
  3. El Administrador confirma.
  4. El Sistema genera el archivo con la información requerida.
  5. El Sistema ofrece el archivo para descarga/almacenamiento.
- **Extensiones (Flujos alternativos):**
  - a. Exportar a CSV:
    - i. El Administrador accede al menú de “Exportar contenido multimedia”.
    - ii. El Administrador selecciona el formato CSV y el alcance (todo catálogo o filtrado).
    - iii. El Administrador confirma.
    - iv. El Sistema genera el archivo con la información requerida.
    - v. El Sistema ofrece el archivo para descarga/almacenamiento.
- **Frecuencia:** baja



**haramMusic**

Modelo de casos de uso

Rev.  
Pág. 44

## 2.20.2 Diagrama de secuencia

