

|  |
| --- |
| Documento de Análisis y Diseño |
| Nivel 13: Soundbox  Sebastian Florez |
| Felipe Otálora |
| Estructuras de Datos  Universidad de Los Andes  2014-1 |
|  |

Documento de Análisis y Diseño

[Nombre del ejercicio]

|  |  |  |
| --- | --- | --- |
| **Versión del documento** | **Fecha de modificación** | **Responsables** |
| v1 | Febrero 1/2014 | Felipe Otálora  Sebastian Florez |
| v2 | Febrero 4/2014 | Felipe Otálora  Sebastian Florez |
|  |  |  |

Contenido

[1 Requerimientos Funcionales 3](#_Toc378579315)

[2 Análisis del modelo del mundo 4](#_Toc378579316)

[3 Diseño de Interfaces 5](#_Toc378579317)

[3.1 Interfaces del mundo de la aplicación 5](#_Toc378579318)

[3.2 Interfaces de las estructuras de datos 5](#_Toc378579319)

[4 Diseño del plan de pruebas de la estructura de datos 6](#_Toc378579320)

[5 Diseño de estructuras de datos 7](#_Toc378579321)

[6 Justificación de decisiones de diseño de las estructuras de datos 8](#_Toc378579322)

[7 Diseño final de la aplicación 9](#_Toc378579323)

[8 Justificación de decisiones de diseño del mundo 10](#_Toc378579324)

[9 Mapa de navegación de la aplicación 11](#_Toc378579325)

[10 Análisis de complejidad de operaciones del mundo 12](#_Toc378579326)

1. *Requerimientos Funcionales*

A continuación se describen los requerimientos funcionales de la aplicación. La definición de estos es independiente de la tecnología que será usada para implementarlos.

|  |  |
| --- | --- |
| **ID** | <identificador único, generalmente un consecutivo> |
| **Nombre** | <nombre, comienza con verbo en infinitivo> |
| **Resumen** | <breve descripción del requerimiento> |
| **Entradas** | <nombre entrada i> <tipo> <descripción> |
| **Resultados** | <nombre resultado i> <tipo> <descripción> |

Requerimientos:

|  |  |
| --- | --- |
| **ID** | R1 |
| **Nombre** | Importar Sonidos desde una ubicación determinada |
| **Resumen** | La aplicación debe poder importar sonidos a los canales desde la biblioteca interna de sonidos. |
| **Entradas** | sonido: El sonido que se va a añadir al canal  canal: El canal al cual se le van a añadir el sonido |
| **Resultados** | El sonido es añadidos al canal. |

|  |  |
| --- | --- |
| **ID** | R2 |
| **Nombre** | Importar sonidos desde una ubicación determinada |
| **Resumen** | La aplicación debe poder importar sonidos a la Biblioteca de Sonidos cargando un archivo o carpeta con sonidos. |
| **Entradas** | -El archivo de sonido que se quiere añadir al canal  -La carpeta que contiene los sonidos que se quieren añadir. |
| **Resultados** | Los sonidos son añadidos a la Biblioteca de sonidos. |

|  |  |
| --- | --- |
| **ID** | R3 |
| **Nombre** | Crear un nuevo proyecto |
| **Resumen** | La aplicación debe poder crear un nuevo proyecto especificando el autor, el número inicial de canales y su autor. |
| **Entradas** | * El nombre del nuevo proyecto * El numero inicial de canales para el proyecto. * El autor del proyecto |
| **Resultados** | Se crea un nuevo proyecto con el nombre, autor y número de canales especificados. |

|  |  |
| --- | --- |
| **ID** | R4 |
| **Nombre** | Nombrar un canal |
| **Resumen** | La aplicación debe poder nombrar y editar el nombre del canal |
| **Entradas** | El nuevo nombre del canal |
| **Resultados** | Se le ha cambiado el nombre al canal |

|  |  |
| --- | --- |
| **ID** | R5 |
| **Nombre** | Editar los datos de un proyecto |
| **Resumen** | Se deben poder editar los datos de un proyecto, su nombre, bpm |
| **Entradas** | * El nuevo nombre del proyecto * El nuevo bpm del proyecto |
| **Resultados** | Se han cambiado los datos del proyecto, tanto su nombre como su bpm |

|  |  |
| --- | --- |
| **ID** | R6 |
| **Nombre** | Salvar un proyecto |
| **Resumen** | La aplicación debe poder salvar el proyecto en el que se está trabajando en el estado en el que esté, esto incluye número de canales, bpm, sonidos y la unidad de tiempo en el que inicia cada uno. |
| **Resultados** | Se ha salvado el proyecto en un archivo de texto con los datos de cada uno de los canales y sonidos. |

|  |  |
| --- | --- |
| **ID** | R7 |
| **Nombre** | Cargar un proyecto |
| **Resumen** | La aplicación debe poder cargar un proyecto ya existente |
| **Entradas** | El proyecto que se quiere cargar |
| **Resultados** | Se ha cargado el proyecto existente |

|  |  |
| --- | --- |
| **ID** | R8 |
| **Nombre** | Buscar un proyecto por criterio |
| **Resumen** | Se debe poder buscar entre los proyectos existentes mediante algún criterio |
| **Entradas** | El criterio de búsqueda, ya sea el nombre o el autor del proyecto. |
| **Resultados** | El proyecto que se quería buscar |

|  |  |
| --- | --- |
| **ID** | R9 |
| **Nombre** | Agregar sonidos (samples) a los canales de reproducción |
| **Resumen** | El usuario debe poder seleccionar qué sonido quiere agregar a un canal de reproducción en un instante determinado sin sobrelaparse con los otros samples. |
| **Entradas** | El sonido que se quiere añadir al canal  El canal que recibe el sonido |
| **Resultados** | El sonido se ha agregado al canal |

|  |  |
| --- | --- |
| **ID** | R10 |
| **Nombre** | Filtrar sonidos por categoría o nombre |
| **Resumen** | Para poder facilitar la navegación, el usuario debe poder filtrar los sonidos por un criterio de búsqueda, ya sea el autor o la categoría de los sonidos importados. |
| **Entradas** | El criterio de búsqueda, ya sea la categoría o nombre |
| **Resultados** | Los sonidos se han filtrado por el criterio |

|  |  |
| --- | --- |
| **ID** | R11 |
| **Nombre** | Cambiar el tempo de reproducción |
| **Resumen** | El usuario debe poder alter el tiempo de reproducción del proyecto |
| **Entradas** | El nuevo tempo de reproducción del proyecto |
| **Resultados** | Se ha cambiado el tiempo de reproducción |

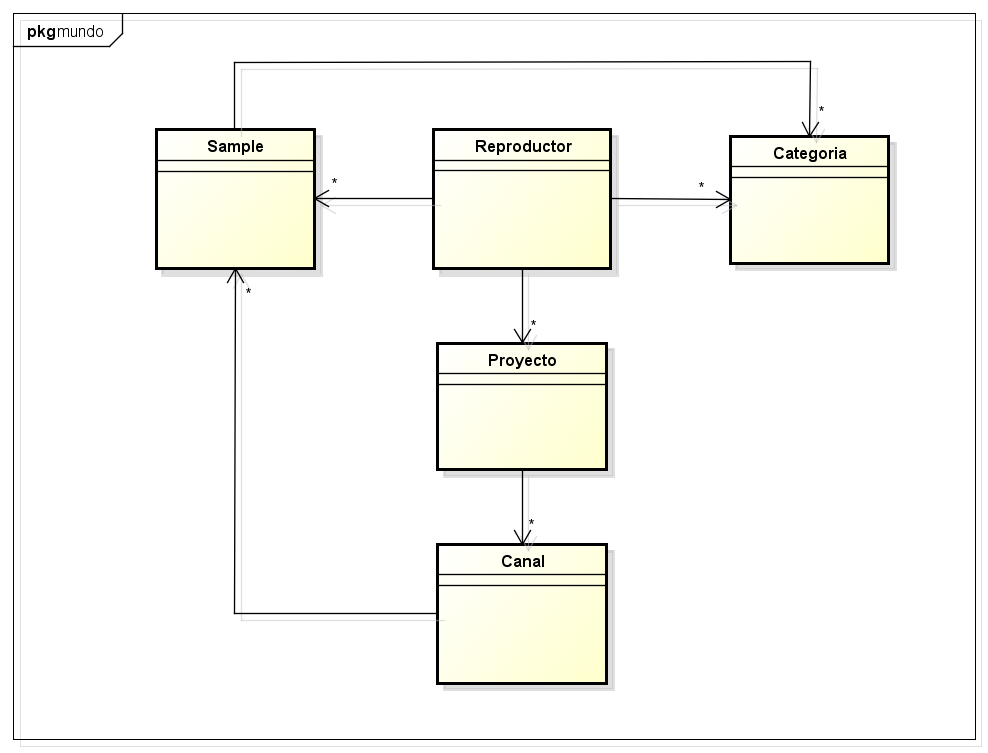
|  |  |
| --- | --- |
| **ID** | R12 |
| **Nombre** | Adicionar/Eliminar un canal |
| **Resumen** | El usuario debe poder adicionar o eliminar un canal existente |
| **Entradas** | El canal que se quiere eliminar o el nombre del nuevo canal |
| **Resultados** | Se ha agregado/eliminado un canal |

|  |  |
| --- | --- |
| **ID** | R13 |
| **Nombre** | Adicionar/eliminar una categoría |
| **Resumen** | Se debe poder crear una nueva categoría de sonidos o eliminarla |
| **Entradas** | La nueva categoría, la categoría para eliminar |
| **Resultados** | Se ha adicionado/eliminado la categoría |

|  |  |
| --- | --- |
| **ID** | R14 |
| **Nombre** | Adicionar/eliminar un sonido |
| **Resumen** | El usuario debe poder agregar o eliminar un sonido de una categoría |
| **Resultados** | Se ha agregado/eliminado un sonido de la categoría |

|  |  |
| --- | --- |
| **ID** | R15 |
| **Nombre** | Reproducir un proyecto |
| **Resumen** | Se debe poder reproducir el proyecto creado |
| **Entradas** | El proyecto que se quiere reproducir |
| **Resultados** | El proyecto se ha reproducido |

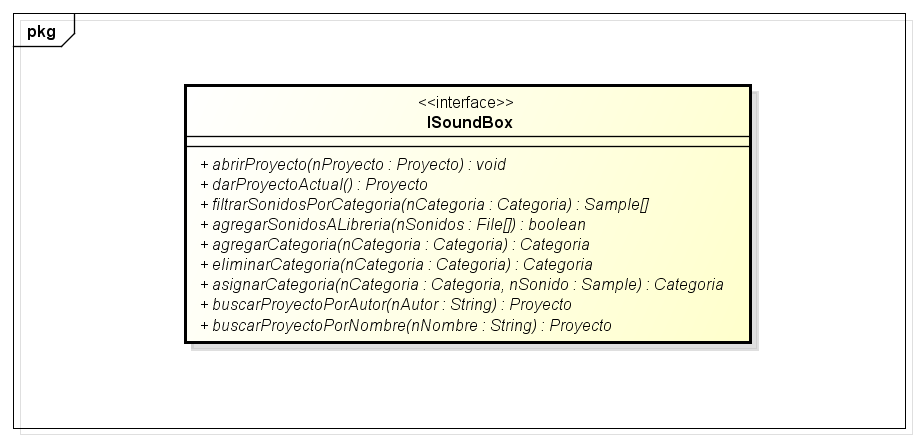
1. *Análisis del modelo del mundo*

Diagrama de clases (UML) de la abstracción del problema que se pretende solucionar. Solo se tienen en cuenta entidades, características y relaciones relevantes para el mismo. Aún no se toman decisiones de diseño. 

1. *Diseño de Interfaces*
   1. Interfaces del mundo de la aplicación

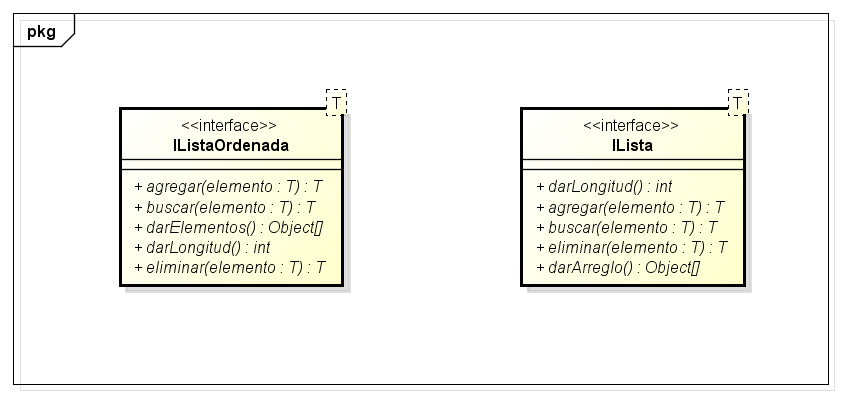
A partir de los requerimientos funcionales, defina las operaciones (a través de métodos) que debe proveer el mundo del problema. Incluya parámetros, tipos de retorno y documentación completa de cada operación (incluyendo precondiciones).

La documentacion se encuentra en la clase ISounbox en el paquete de la entrega.



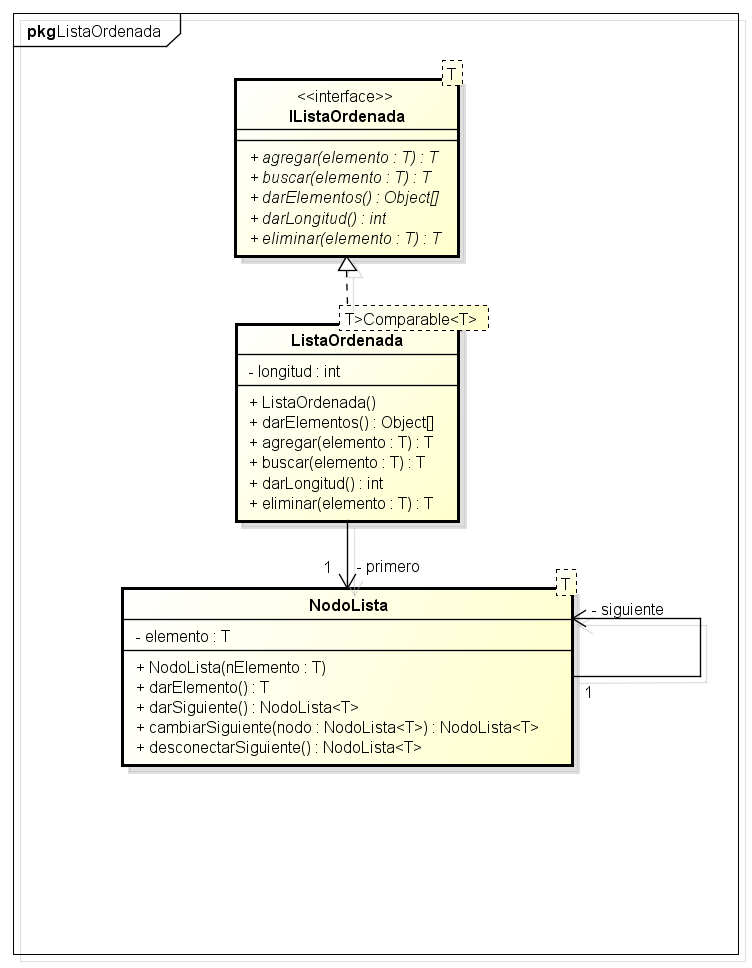
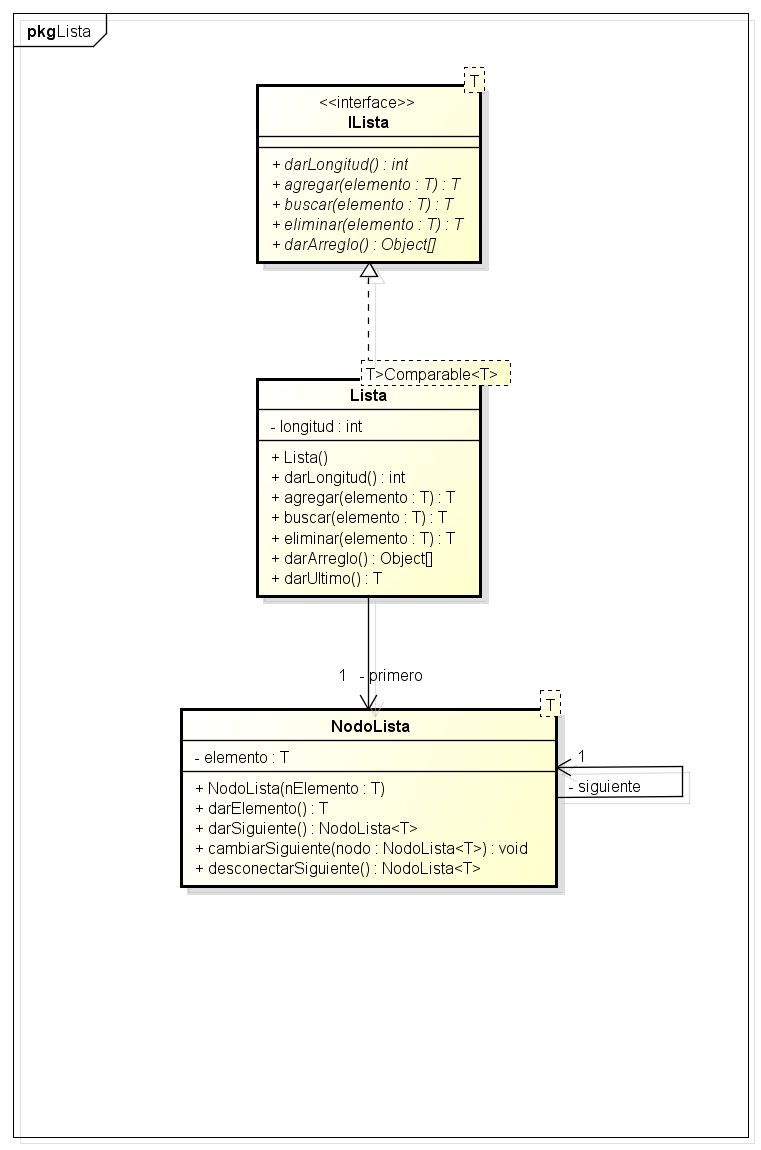
* 1. Interfaces de las estructuras de datos

Seleccione cuáles son las estructuras de datos genéricas que debe utilizar y las operaciones que estas deben proveer para minimizar la complejidad temporal de las operaciones definidas en el numeral 3.1.



1. *Diseño de estructuras de datos*

Describa el detalle de las Estructuras de Datos que va a utilizar en su solución, a través de un diagrama UML de las mismas. No olvide incluir invariantes si las requiere.



1. *Justificación de decisiones de diseño de las estructuras de datos*

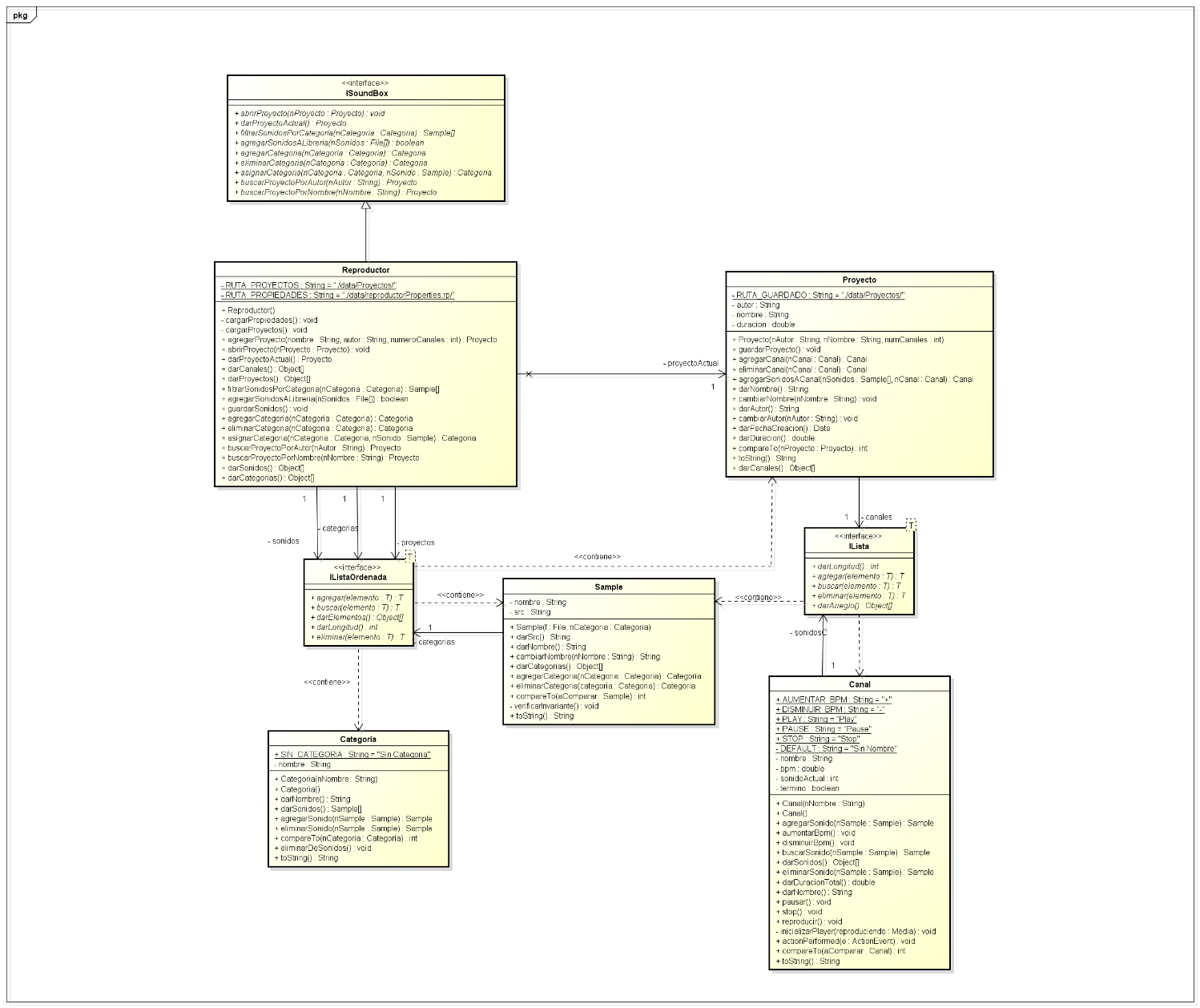
Describa en detalle los aspectos que tuvo en cuenta para seleccionar y diseñar las estructuras de datos descritas en el punto anterior.

Lista: Escogimos utilizar una lista encadenada simple para el manejo de los canales ya que la referencia de éstos con respecto al proyecto no requiere que se encuentren organizados en algún orden específico. Por consiguiente, la reproducción de los canales no necesita que éstos sean recorridos en un orden sino que todos sean recorridos al mismo tiempo para ajustarse a la posición y tiempo de reproducción. Por otra parte, la cantidad de canales que manejará la aplicación es considerablemente menor a la cantidad de sonidos y proyectos guardados.

Lista Ordenada: Escogimos utilizar una lista ordenada encadenada para poder manejar las referencias de los sonidos y de los proyectos. La justificación corresponde al manejo de búsquedas que se realicen entre los proyectos y los sonidos. Es decir, debido a que la aplicación cuenta con una biblioteca de más de 500 sonidos y el usuario debe poderlos filtrar por nombre o categoría, es necesario que el recorrido de los sonidos sea el más eficiente para minimizar el tiempo de ejecución de las tareas y al establecer un criterio de orden en tanto los sonidos como los proyectos se garantiza el recorrido más eficiente.

1. *Diseño final de la aplicación*

Describa en detalle el diseño final de su aplicación a través de un diagrama UML, incluya atributos y métodos. Cuando se refiera a una estructura de datos no es necesario que incluya toda su definición dentro del diagrama. Basta con referenciar su interfaz.



1. *Justificación de decisiones de diseño del mundo*

Describa en detalle los aspectos que guiaron el diseño final de su aplicación.

Se quería lograr la mayor eficiencia posible a partir del manejo de las distintas listas. Dado que tanto la lista ordenada, como la lista encadenada tienen la misma complejidad la implementación de ambas permite una libertad desde un aspecto práctico y de interacción con el programa. Por un lado la lista ordenada filtrara los sonidos y las categorías a partir de un criterio de ordenamiento dado. Por otro lado, la lista enlazada filtrara los sonidos y canales según el orden en el que se vayan agregando.

Los sonidos (Samples) estarán presentes en dos listas, una principal, que será la ordenada, manejada por la clase principal de reproductor a partir del manejo de una librería local de sonidos. Por otro lado, la clase categoría listara los distintos sonidos de su dominio, estos serán referencias de la librería anteriormente mencionada y permitirá que al momento de filtrar se logre un procesos eficiente.

Se utilizará una interfaz de sonido aplicable para los samples y canales dado que ellos tienen las mismas funcionalidades elementales de reproducción, nombre y duración.

1. *Análisis de complejidad de operaciones del mundo*

Basado en los diseños de las estructuras de datos y en el mundo de la aplicación, defina la complejidad de cada una de las operaciones. Utilice la notación O(f(n)). Justifique por qué considera que esta es la mínima posible para el problema que está resolviendo.

Análisis de complejidad Estructuras de Datos:

1. Lista
   1. Agregar Elemento: O(k) Justificación: Se mantiene una referencia al último elemento de la lista, razón por la cual, el agregar elemento consiste en simplemente cambiarle el siguiente elemento al último y reajustar el último elemento.
   2. Buscar Elemento: O(n) Justificación: Para poder buscar el elemento, es necesario realizar un solo recorrido para avanzar por toda la lista y comparar si el elemento actual de la lista corresponde al buscado. En el peor de los casos se recorre toda.
   3. Eliminar Elemento: O(n) Justificación: Para poder eliminar el elemento, es necesario realizar un solo recorrido para avanzar por toda la lista y comparar si el elemento actual de la lista corresponde al que se quiere eliminar. En el peor de los casos se recorre toda.
   4. Dar Longitud: O(k) Justificación: Debido a que la longitud de la lista está contenida como un atributo de la lista y los métodos de tanto eliminar como agregar se encargan de actualizar el valor, basta con devolver el valor del atributo.
   5. Dar Arreglo: O(n) Justificación: Para construir el arreglo es necesario realizar un solo recorrido para en la lista para que, a medida de que se avanza, se puedan agregar los nodos al arreglo.
2. Lista Ordenada:
   1. Agregar Elemento: O(n) Justificación: Debido a que existe un criterio de orden para poder agregar los elementos, es necesario recorrer al menos una vez la lista para agregar correctamente el elemento, respetando el criterio de orden. En el peor de los casos se recorre toda la lista.
   2. Buscar Elemento: O(n) Justificación: Para poder buscar el elemento, es necesario realizar un solo recorrido para avanzar por toda la lista y comparar si el elemento actual de la lista corresponde al buscado. En el peor de los casos se recorre toda.
   3. Eliminar Elemento: O(n) Justificación: Para poder eliminar el elemento, es necesario realizar un solo recorrido para avanzar por toda la lista y comparar si el elemento actual de la lista corresponde al que se quiere eliminar. En el peor de los casos se recorre toda.
   4. Dar Longitud: O(k) Justificación: Debido a que la longitud de la lista está contenida como un atributo de la lista y los métodos de tanto eliminar como agregar se encargan de actualizar el valor, basta con devolver el valor del atributo.
   5. Dar Arreglo: O(n) Justificación: Para construir el arreglo es necesario realizar un solo recorrido para en la lista para que, a medida de que se avanza, se puedan agregar los nodos al arreglo.
3. Modelo del Mundo:
   1. Abrir Proyecto: O(n) Justificación: El reproductor, en el peor de los casos, debe recorrer al menos toda la carpeta que contiene los proyectos serializados para poder abrir el proyecto buscado.
   2. Agregar Canal: O(k) Justificación: Corresponde al agregar del método de la lista simple, por lo tanto tiene la misma complejidad.
   3. Agregar Categoría: O(n) Justificación: Corresponde al agregar del método de la lista ordenada, por lo tanto tiene la misma complejidad.
   4. Buscar Proyecto por Autor: O(n) Justificación: El reproductor debe iterar, en el peor de los casos, hasta el último elemento de los proyectos para comparar si corresponde al proyecto por autor deseado.
   5. Agregar Sonido: O(n) { Es O(k) cuando se agrega en un canal } Justificación: Corresponde al agregar del método de la lista ordenada, por lo tanto tiene la misma complejidad.
   6. Asignar Categoría: O(k) Justificación: El proyecto solo debe realizar una asignación para la categoría del sonido.
   7. Buscar Proyecto por Nombre: O(n) Justificación: El reproductor debe iterar, en el peor de los casos, hasta el último elemento de los proyectos para comparar si corresponde al proyecto por el nombre deseado.
   8. Crear Proyecto: O(k) Justificación: La aplicación solo debe realizar una asignación para la crear el proyecto.
   9. Editar proyecto: O(k) Justificación: La aplicación solo debe realizar una asignación para editar la información del proyecto y no es necesario recorrer la lista de proyectos debido a que se edita el ya cargado.
   10. Eliminar canal: O(n) Justificación: Corresponde al método de eliminar de la lista simple, por lo tanto tiene la misma complejidad.
   11. Eliminar categoría: O(n) Justificación: Corresponde al método de eliminar de la lista ordenada, por lo tanto tiene la misma complejidad.
   12. Filtrar Sonidos: O(n) Justificación: Solo debe recorrer la lista de categorías para buscar la categoría deseada, después la categoría entrega los sonidos que le pertenecen.
   13. Guardar Proyecto: O(k) Justificación: El proyecto sabe guardarse con sus objetos debido a la serialización.
   14. Reproducir: O(n^2) Justificación: Para poder reproducir el proyecto, es necesario que se recorran primero los canales y luego de recorrer los canales se deben recorrer los sonidos para poderlos reproducir uno a uno.
   15. Pausar: O(n^2) Justificación: Justificación: Para poder reproducir el proyecto, es necesario que se recorran primero los canales y luego de recorrer los canales se deben recorrer los sonidos para poderlos pausar uno a uno.
   16. Stop: O(n^2) Justificación: Justificación: Para poder reproducir el proyecto, es necesario que se recorran primero los canales y luego de recorrer los canales se deben recorrer los sonidos para poderlos parar uno a uno.