

Middleware: Herramientas de Desarrollo

# **Práctica final**

## **Manual de uso de la herramienta**



Autor: Johnny Guayanay Troya.

Profesor: Angel Rodriguez Ballesteros.

## índice

Introducción .....	3
Diseño de la interfaz.....	4
Diseño del código .....	5
Ampliaciones para el futuro .....	8

# Introducción

En esta práctica se ha realizado una herramienta de traducción de canciones.

Las funcionalidades que se han implementado en esta herramienta han sido, la de poder **guardar las secuencias de tiempo asociadas a un texto en un archivo .txt** y la de **convertir dichos datos en un archivo serializado .bin**, también se ha implementado la operación inversa, la de **deserialización**.

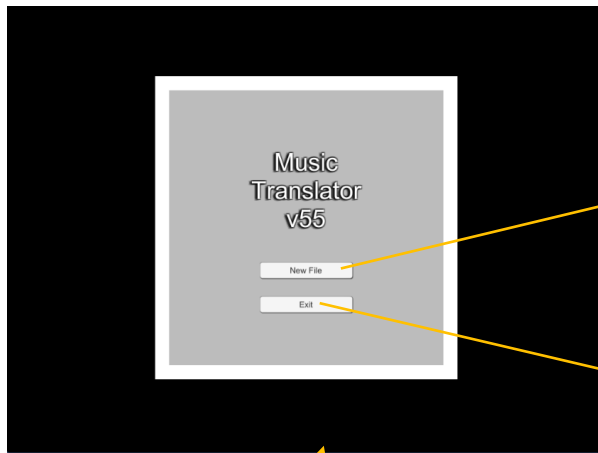
A parte de esto se creó **una .dll escrita en C++** donde **establecemos el backend** de la herramienta y como **frontend hemos usado el motor Unity 3D**.

Para llegar a la creación de la herramienta final, se tuvo que seguir una fase previa de diseño. Tanto de interfaz como de código.

# Diseño de la interfaz

Para esta fase nos decantamos primero por crear la interfaz con la que el usuario va a trabajar, de este modo definimos el flujo de pantallas de la siguiente manera:

## Menú Inicial

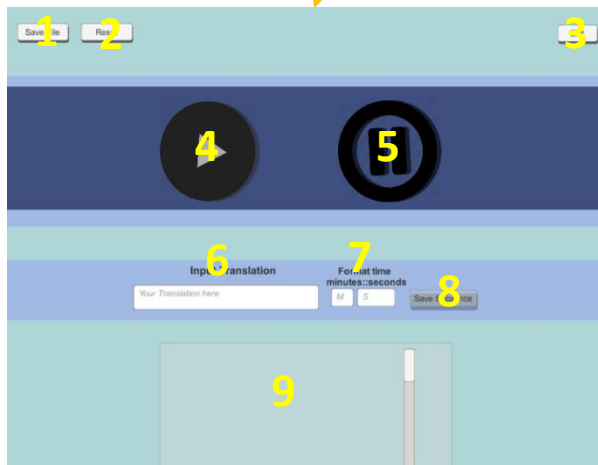


¿Qué va a poder hacer/ ver el usuario en esta pantalla?

Va a poder seleccionar una canción desde el explorador de archivos. Esta se guarda en los archivos temporales de audio.

Podrá salir de la herramienta.

## Menú del editor



¿Qué va a poder hacer/ ver el usuario en esta pantalla?

1. Guardar su progreso.
2. Volver a empezar.
3. Salir y guardar sus datos serializados.
- 4 y 5 Reproducir y para el audio.
6. Introducir la traducción.
7. Ver el tiempo de reproducción.
8. Guardar los datos de traducción y de tiempo.
9. Ver como está quedando el texto traducido.

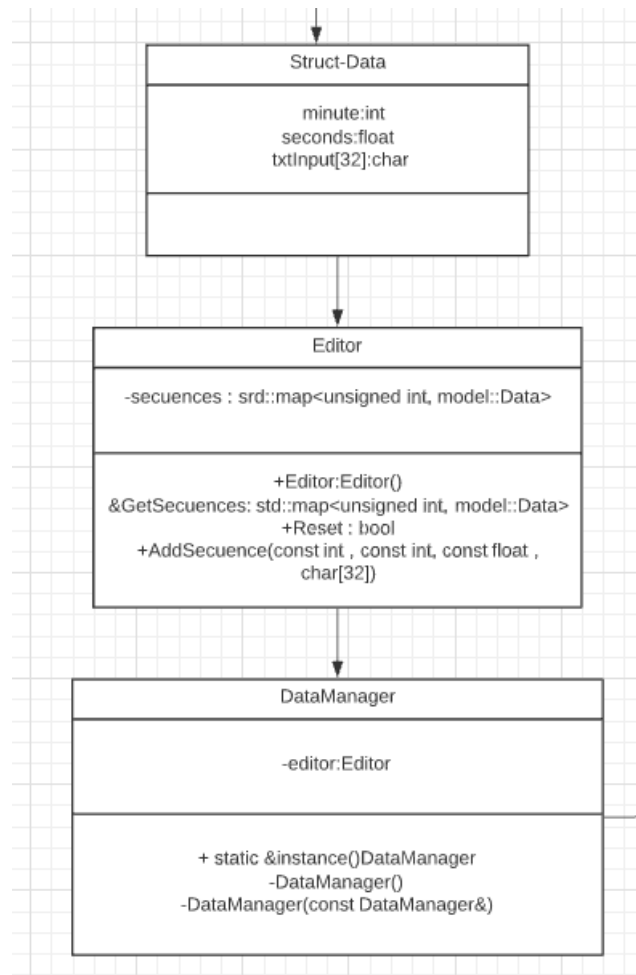
Con un visual de la interfaz la tarea de crear el modelo de datos y la parte backend de herramienta se hizo de manera ágil.

## Diseño del código

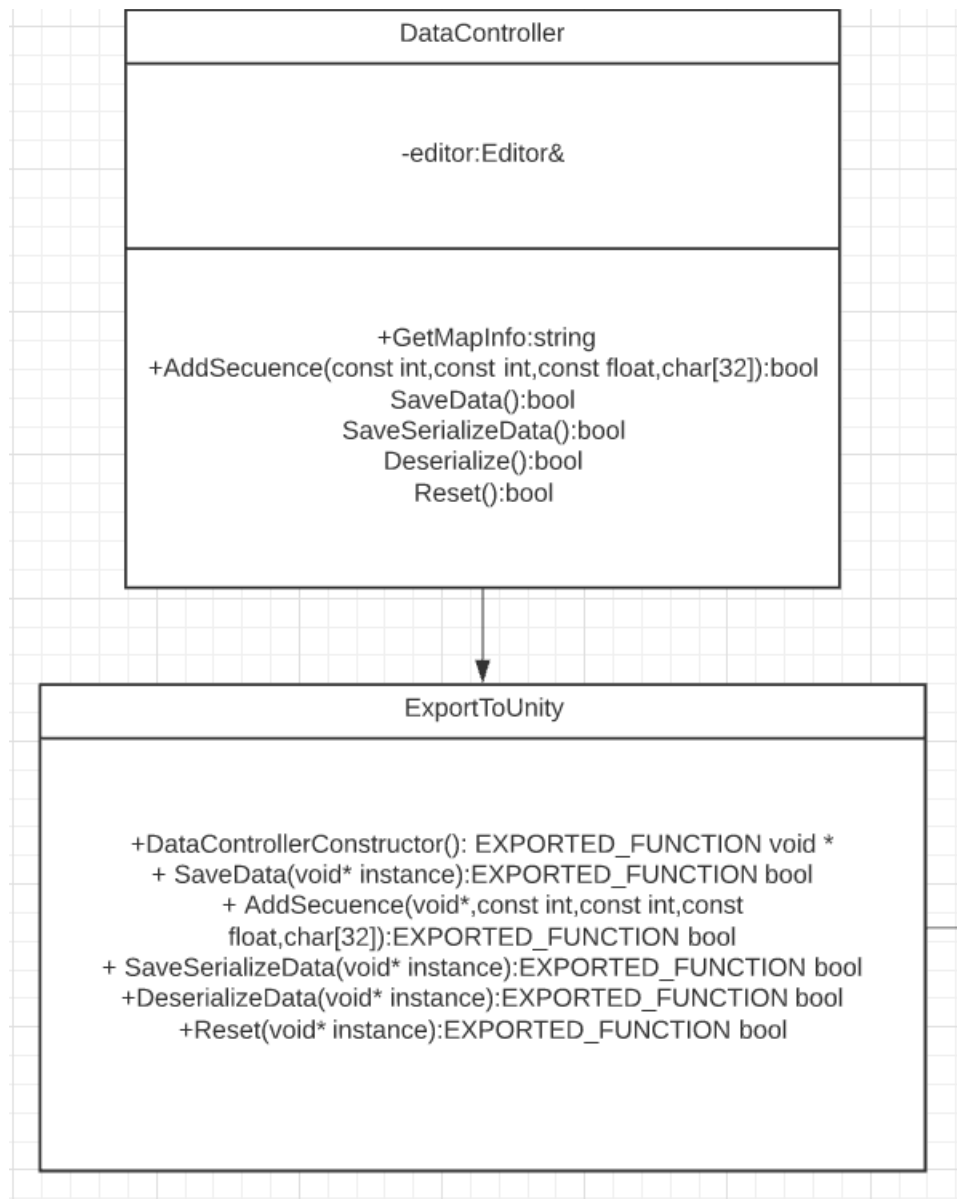
A la hora de guardar los archivos de texto se decidió seguir el esquema que siguen los archivos **SRT**, secuencia, tiempo y texto. Este patrón se tomó en cuenta a la hora de formatear los datos que tenemos en **nuestro mapa**. Dicho mapa toma como clave la secuencia y como valor se ha utilizado la **clase Data** que almacena los datos de tiempo (minutos y segundos) y texto(char[32]). Se tuvo que optar por un array de chars, ya que al hacer el importe usando strings daba errores en la aplicación.

Continuando con la explicación cabe mencionar que este proyecto se ha desarrollado siguiendo el esquema: **Modelo, controlador y vista**.

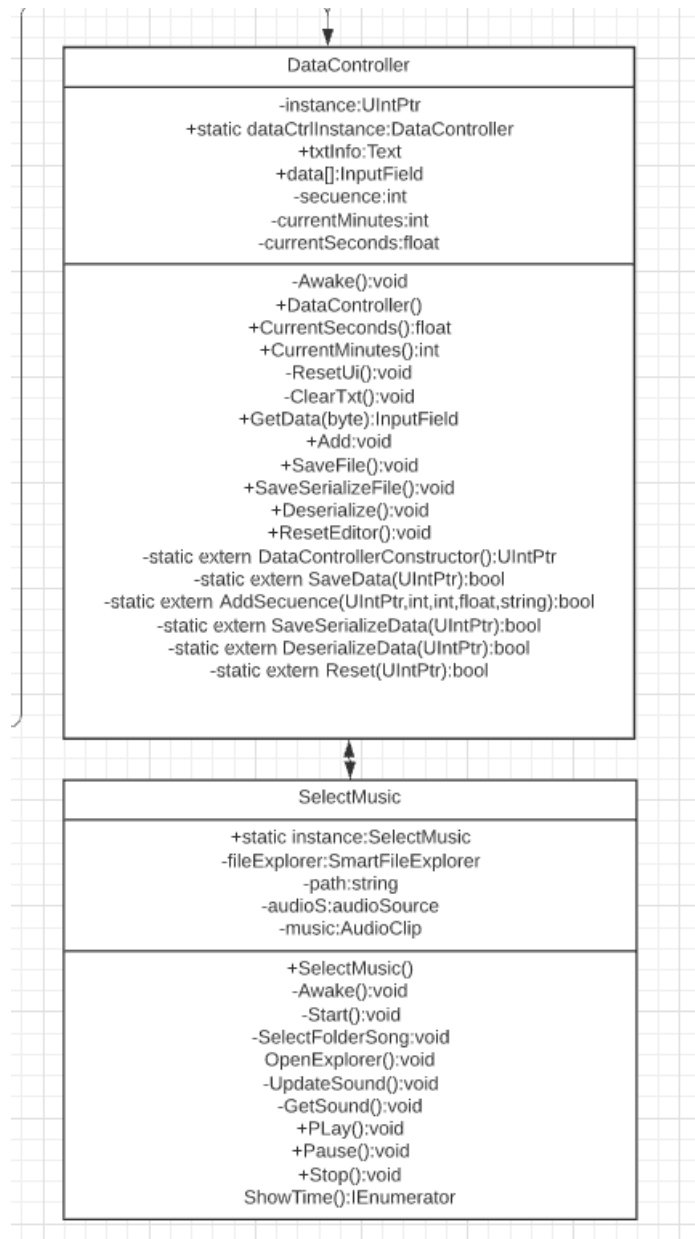
La parte de **Modelo** ha quedado así definida:



## El controlador



## Vista



## Ampliaciones para el futuro

La herramienta ha quedado completa sus principales funcionalidades se han probado y no dejan nada que desear, pero aún le hace falta algunas mejoras como la de poder eliminar las secuencias que el usuario quiera, y la de poder reproducir el audio hacía atrás.

A parte de un pulido estético donde podamos ver la línea de tiempo de la pista de audio.