

Informe Inicial

Marc Martín Martínez

~~February 28, 2020~~

Índice

1	Objetivos	3
2	Planificación	3
2.1	Introducción al tema: 2 semanas	3
2.2	Análisis de Requisitos: 1 semana	3
2.3	Diseño: 3 semanas	4
2.3.1	Funcionalidad	4
2.3.2	Interfaz de usuario	4
2.4	Implementación y Testing: 6 semanas	4
3	Metodología	4
4	Bibliografía	4

1 Objetivos

Creación de un entorno web en el que se puedan subir archivos MusicXML, modificarlos y descargarlos.

Visualización de la información contenida en el MusicXML en un pentagrama virtual, simulando uno real.

Desarrollar una herramienta que permita hacer cambios en el pentagrama virtual de manera que estos se vean reflejados en el archivo MusicXML original. La herramienta contará con una interfaz gráfica con elementos fácilmente reconocibles para un usuario con conocimientos musicales, evitando que el usuario necesite conocer el funcionamiento de MusicXML.

Seleccionar el elemento que se quiere modificar con un click de ratón.

Los elementos que podrán ser modificados serán las notas y sus tempos, la armadura, los accidentes, la clave, el compás, el carácter y las ligaduras entre notas.

Opción para cambiar entre el sistema de notación musical latina o anglosajona para modificar las notas.

Una vez finalizada la modificación de la partitura, descargar el archivo MusicXML resultante de dicha modificación.

2 Planificación

En esta sección se representa en diferentes etapas como se desarrollará el proyecto y la duración aproximada de las mismas.

2.1 Introducción al tema: 2 semanas

Aprendizaje del funcionamiento de MusicXML. [2]

Creación de un glosario para el entendimiento de tecnicismos musicales.

Comprobar el estado del arte. [1]

2.2 Análisis de Requisitos: 1 semana

Identificación de fuentes de requisitos y por ende listar una serie de requisitos.

Clasificación de los requisitos en funcionales, de calidad y, en caso de haberlas, restricciones. Posteriormente, reclasificar estos requisitos siguiendo el modelo de Kano.

2.3 Diseño: 3 semanas

La fase de diseño será dividida en dos subsecciones para distinguir entre el diseño de la funcionalidad de la aplicación y el diseño de la interfaz de usuario.

2.3.1 Funcionalidad

Identificación de patrones ya existentes para su posterior aplicación a la aplicación.

Revisar que se asignan correctamente las responsabilidades correspondientes a cada clase, siguiendo los patrones GRASP.

Creación de diagramas de secuencia y clase.

2.3.2 Interfaz de usuario

Esta fase constará de varias iteraciones en las que el diseño de la UI irá mejorando con cada iteración.

Uso de técnicas y reglas para crear una UI fácil de usar y aprender a la vez que es eficiente y segura.

Crear *personas* para tener una perspectiva mas fiel a la del usuario final de la aplicación.

Crear prototipos de baja fidelidad al inicio para realizar pruebas con posibles usuarios para identificar errores y corregirlos. A medida que hayan mas iteraciones se harán prototipos mas sofisticados y detallados.

2.4 Implementación y Testing: 6 semanas

Estas dos fases normalmente separadas se harán de forma simultanea siguiendo el paradigma *Test-Driven Development*.

Una vez finalizada la implementación se llevaran a cabo pruebas adicionales para asegurar que el programa actúa de forma correcta mediante *exploratory testing*.

Creación de *test cases* y documentarlos de forma adecuada.

3 Metodología

4 Bibliografía

- [1] Arnau Baró et al. "From Optical Music Recognition to Handwritten Music Recognition: A baseline". In: *Patter Recognition Letters* 123 (2019), pp. 1–8. DOI: <https://doi.org/10.1016/j.patrec.2019.02.029>.
- [2] Make Music. *User manuals for MusicXML*. URL: <http://usermanuals.musicxml.com/MusicXML/MusicXML.htm>.