



Universitat Autònoma de Barcelona

Partituras Musicales. Creación de una aplicación web para validar sistemas de reconocimiento de partituras musicales.(OMR)

Marc Martín Martínez

Directora: Dr. Alicia Fornés Bisquerra

Escola d'Enginyeria
Universitat Autònoma de Barcelona

Índice

1	Planificación Prevista	3
1.1	Objetivos	3
1.2	Testing	3
2	Metodología seguida	3
3	Resultados obtenidos	4
3.1	Backend	4
3.2	Frontend	5
3.2.1	Subida de archivos	5
3.2.2	Visualización de la Partitura	5
4	Conclusiones Provisionales	6
5	Bibliografía	7

1 Planificación Prevista

1.1 Objetivos

En esta etapa habían dos grandes objetivos a realizar: visualizar la partitura en el navegador a partir del archivo MusicXML e implementar el editor de esta misma partitura. A la hora de planificar estos objetivos, se fue muy optimista con el tiempo que se tardaría en realizar el primer objetivo y la realidad es que se ha necesitado mucho mas tiempo para desarrollarlo. Por ello, la implementación del editor se ha de retrasar a la siguiente etapa y, por consecuencia, alguna iteración planificada de mejora del software y de user testing puede no llevarse a cabo por falta de tiempo.

1.2 Testing

En cuestión del testing, la meta era seguir el paradigma *Test-Driven Development* utilizando Jest, pero no se ha llevado a cabo por el aumento de carga de trabajo que causaba el aprendizaje del uso de la herramienta y realizar en si mismo los tests.

No obstante si que se han realizado *test scenarios* para asegurar el funcionamiento de la aplicación en los casos más comunes.

2 Metodología seguida

La metodología seguida, como hasta ahora, ha sido incremental. En esta etapa varios elementos han sufrido grandes cambios respecto al planteamiento inicial. El prototipo creado para el editor ha sido totalmente desechado. Era demasiado grande teniendo en cuenta las nuevas adiciones a la aplicación(imágenes de la partitura y la partitura) y se ha creído oportuno que se acabará haciendo uno más compacto.

El diagrama de clases también ha sido modificado y se han añadido nuevas clases que hasta ahora no se habían tenido en cuenta y que han surgido durante la implementación. Este diagrama no es la versión final ya que con la implementación del editor surgirán nuevos elementos. El nuevo diagrama puede verse en la figura 1.

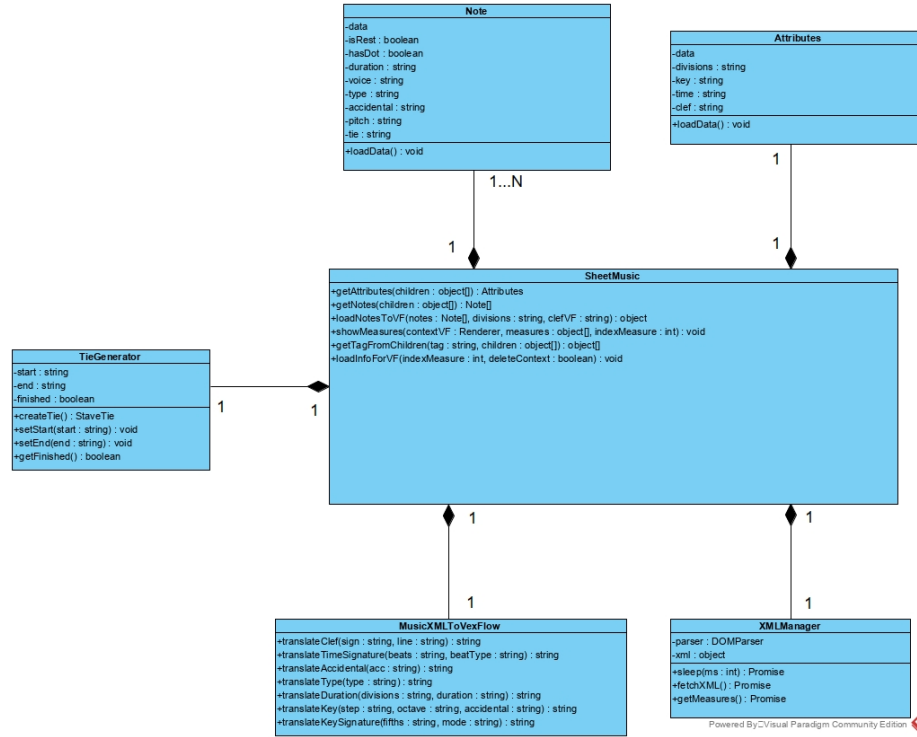


Fig. 1: Diagrama de Clases

3 Resultados obtenidos

En esta sección se repasarán las funcionalidades desarrolladas hasta la fecha.

3.1 Backend

El servidor usa el framework *Express* [3] para el *routing*. Hasta ahora sus únicas funciones son almacenar los archivos subidos por el usuario y servirlos cuando sea necesario. Para la subida de archivos al servidor se añade un modulo a express, express-fileupload.[4]

3.2 Frontend

3.2.1 Subida de archivos

El formulario acepta un único archivo XML y una o más imágenes de la partitura relacionada con dicho XML, como se ve en la figura 2.

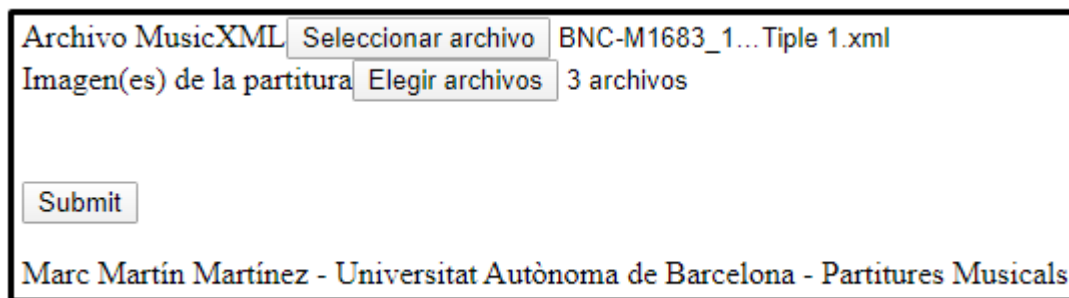


Fig. 2: Formulario de subida de archivos

A este formulario se le dará un formato más apropiado en el futuro.

3.2.2 Visualización de la Partitura

A partir del archivo XML provisto por el usuario, se visualizan hasta 5 compases, 6 en el caso que haya una ligadura entre una nota del último compás de la iteración y una nota en el primer compás de la siguiente iteración, en el navegador y una de las imágenes correspondientes, como se puede observar en la figura 3.

INDEX-HTML

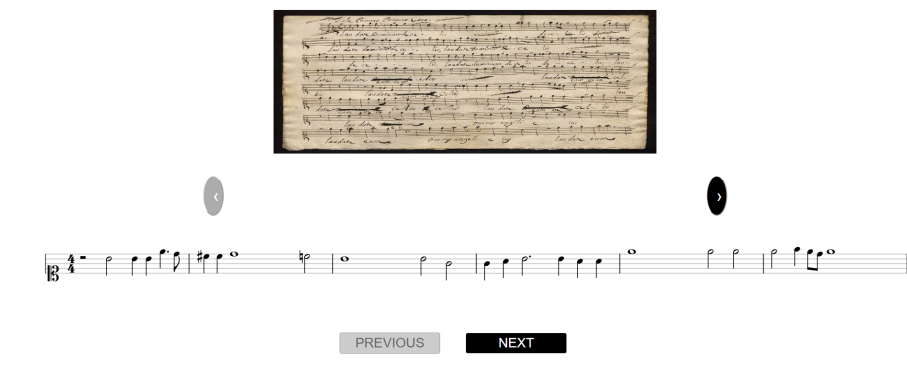


Fig. 3: Página Principal

El usuario puede cambiar entre las diferentes imágenes que haya subido anteriormente. También puede hacer click en la imagen para hacerle *zoom*. Finalmente puede usar los botones *previous* y *next* para navegar en los diferentes compases que se hayan creado.

La aplicación parsea el archivo XML y extrae los tags necesarios para crear una partitura, para lo que se usó la API *Document* [1]. A partir de esos tags se interpreta la información contenida en ellos y se formatea para su uso en la API *VexFlow* [2], que representa la partitura en un SVG(Scalable Vector Graphics).

La aplicación, de momento, es capaz de representar los siguientes elementos de una partitura:

- Notas
- Legatos y ligaduras
- Silencios
- Armadura
- Accidentes
- Clave¹
- Puntillos
- Signatura de compás

4 Conclusiones Provisionales

Se debería haber sido mas exhaustivo en la fase de diseño de la aplicación. Ir creando nuevas funciones o clases sobre la marcha lleva a una situación caótica en el código y se hacen cosas poco eficientes o que al poco tiempo son obsoletas y se tiene que reprogramar funcionalidades que se consideraban ya completas.

La planificación en su gran mayoría se considera que ha sido acertada, pero su ejecución no ha sido la más óptima y eso lleva a perder funcionalidades u oportunidades debido a la proximidad de la fecha limite del proyecto.

¹ Las claves *Bass Clef (Down Octave)*, *Vocal Tenor Clef* y la de tablatura (usada en guitarras, bajos, etc) no estarán disponibles en la aplicación.

5 Bibliografía

- [1] *API Document*. URL: <https://developer.mozilla.org/es/docs/Web/API/Document>.
- [2] Mohit Muthanna Cheppudira. *VexFlow*. URL: www.vexflow.com.
- [3] OpenJS Foundation. *Express*. URL: expressjs.com.
- [4] Richard Girges and Roman Burunkov. *express-fileupload*. URL: <https://github.com/richardgirges/express-fileupload#readme>.