10 de Junio | 2018

Desarrollo de un sitio web para trabajar con partituras de forma colaborativa.

Maria Villanueva Almirantearena

APP Multimedia para e-learning

Resumen

El presente trabajo consiste en la memoria del diseño e implementación de un sitio web para la edición de partituras musicales de forma colaborativa, dirigida a instituciones de enseñanza musical como escuelas de música, conservatorios o el aula de música de escuelas de educación primaria y secundaria. Se trata de una aplicación e-learning en la que se podrán crear partituras generales a partir de particellas creadas por distintos usuarios pudiendo realizar pequeñas modificaciones, reproducirlas y guardarlas. Este proyecto consta de tres fases: análisis de las aplicaciones relacionadas y el diseño y desarrollo del sitio web. Por último, se evaluará el resultado final con el objetivo de que se adecúe a las exigencias del usuario final.

Palabras clave
Partituras, colaboración y música.

Keywords
Score, collaboration and music.

ÍNDICE

1. Introducción 6

1.1. Justificación 6

1.2. Descripción del proyecto: sitio web....7

1.2.1. Posibles aplicaciones. 9

1.2.1.1. Educación musical....9

1.2.1.2. Bases de datos o archivos musicales 10

1.2.2. Descripción del usuario. 10

1.3. Estructura del Trabajo Fin de Grado. 11

2. Objetivos del proyecto 13

2.1. Objetivos generales del proyecto....13

2.2. Objetivos específicos 13

3. Estado del arte 15

3.1. Tecnología software 15

3.1.1. Entorno virtual 15

3.1.2. Software de edición de partituras 17

3.2. Tecnología de programación 18

3.3. Software ABC 19

3.4. Éxitos en el ámbito de colaboración musical. 21

3.5. Modelos de colaboración y problemas en los que se trabaja actualmente. ..22

3.5.1.1. Modelo de documentos colaborativos con la Web 2.0....22

3.5.1.2. Matriz WIKI::SCORE....23

3.5.1.3. LilyPond y proyectos de colaboración 24

4. Diseño del sitio web....26

4.1. Arquitectura del sitio web 26

4.2. Componentes del sitio web....27

4.2.1. Entorno virtual de aprendizaje 27

4.2.1.1. Requisitos del entorno virtual 27

4.2.1.1. Diseño del entorno virtual 28

4.2.2. Actividades de partituras colaborativas. 28

4.2.2.1. Primera propuesta: actividades con Musescore 28

4.2.2.2. Nueva propuesta: aplicación para unir particellas. 29

4.2.2.2.1. Requisitos de las páginas web. 30

4.2.2.2.2. Tecnologías utilizadas. 31

4.2.2.3. Base de datos 31

4.3. Estructura de navegación. 32

4.4. Diseño gráfico o apariencia del sitio web 33

4.4.1. Diseño adaptativo. 34

5. Desarrollo de la web 35

5.1. Entorno virtual 35

5.1.1. Administración del sitio: apariencia e idioma. 35

5.1.2. Administración de cursos 35

5.1.3. Administración de usuarios 37

5.2. Páginas web de actividades 38

5.3. Guardar particellas en una base de datos. (guardarXML_2.html) 38

5.3.1. Base de datos 39

5.3.2. Acceso a la página web para guardar particellas la base de datos. 40

5.4. Conexión con la base de datos. (conexion.php) 45

5.5. Página web para unir particellas (unirparticellas.html) 47

5.5.1. Cargar contenidos 49

5.5.2. Visualizar archivos ABC 51

5.5.3. Crear la partitura completa a partir de las partes 52

5.5.3.1. Notación ABC 52

5.5.3.2. Juntar dos particellas. 55

5.5.3.3. Particella de piano. 56

5.5.4. Excepciones. 57

5.5.5. Limpiar o borrar los archivos cargados. 60

5.5.6. Reproducir online la partitura completa....60

5.5.6.1. Tempo de reproducción. 62

5.5.6.2. Transposición.....63

5.5.6.2.1. Instrumentos transpositores 64

5.5.7. Guardar la partitura completa 67

5.5.8. Editor ABC.....67

5.6. Página web para recuperar antiguas versiones de las particellas. (recuperar.html)....68

5.7. Apariencia....70

6. Enlaces web....71

7. Instalación y testeo.....72

7.1. Funcionalidad de las páginas....72

7.1.1. Página para guardar archivos XML.....72

7.1.2. Página para unir particellas. 72

7.2. Navegación y diseño. 74

7.3. Pruebas de compatibilidad. 75

8. Conclusiones y líneas futuras 76

9. Bibliografía 78

10. Librerías:....79

11. Anexo I: Función para unir particellas....80

12. Anexo II: Glosario musical....84

1. Introducción

El documento que se presenta a continuación consiste en la memoria del Trabajo Fin de Grado correspondiente al Grado en Tecnologías de las Telecomunicaciones de la Universitat Oberta de Catalunya del curso 2018. Dicho TFG se compone del desarrollo de un sitio web relacionado con el e-learning para la creación de partituras musicales completas a partir de partituras individuales, y de la memoria resultante.

Este proyecto pretende facilitar la edición colaborativa de partituras entre los integrantes de un grupo musical, o el profesorado o alumnado de un centro de enseñanza musical. De forma que puedan generar una partitura general a partir de particellas correspondientes a un instrumento y reproducirlas para poder escuchar el arreglo completo.

1.1. Justificación

El mundo se encuentra en constante cambio y al mismo tiempo en que la tecnología se va desarrollando se va convirtiendo en una necesidad para la sociedad. Tiene una gran utilidad en nuestra vida diaria y hoy en día resulta una parte indispensable de la educación por las posibilidades que ofrece.

En los colegios, escuelas o institutos se está comprando material tecnológico y se realiza una gran apuesta por la utilización de estos nuevos recursos. No obstante, en las escuelas de música o conservatorios este cambio no es tan patente a pesar de su potencial. Internet ha revolucionado el mundo de la música, transformando las bases de la creación, interpretación, producción, difusión o consumo digital (Giráldez, 2005). Este nuevo escenario ha abierto nuevas posibilidades al proceso de enseñanza – aprendizaje musical proporcionando nuevos recursos como el acceso a la reproducción de un amplio repertorio, al trabajo de intérpretes, acceso a partituras, etc. Todavía las partituras son una de las principales herramientas de la educación musical y en los últimos años se han ido creando nuevas herramientas para crear, editar y reproducir partituras online.

En cuanto a la vinculación personal con este proyecto, soy Titulada Superior de Música (en la especialidad de Piano) y trabajo desde hace cinco años como profesora en distintas escuelas de música impartiendo las siguientes asignaturas: Lenguaje Musical, Coro, Música y Movimiento, Agrupaciones Instrumentales y Piano. En este periodo de tiempo he podido comprobar que el profesorado realiza arreglos musicales frecuentemente para las clases y para los conciertos.

Muchas veces se realizan agrupaciones poco comunes y es difícil encontrar un repertorio que pueda encajar con estas clases y otras veces el nivel de cada

instrumentista no corresponde con el repertorio disponible. Por tanto, se hacen necesarias estas adaptaciones. Además, en los conciertos de fin de trimestre o fin de curso las escuelas de música y el aula de música de educación primaria y secundaria optan por realizar arreglos de canciones para mayores agrupaciones.

Cada docente crea el arreglo de su propio instrumento con un software de edición de partituras diferente o distinta versión del mismo software. Después, se deben compartir estos archivos mediante el correo electrónico personal y un miembro del claustro debe juntar todas las partes. No obstante, normalmente no se suelen juntar las partes y los problemas surgen en los ensayos. Por ello, se considera la posibilidad de crear partituras de forma colaborativa en la web. Esto también puede ser útil para las clases de agrupaciones instrumentales, colectivas o la clase de combo ya que el docente no tiene porqué conocer la escritura de todos los instrumentos y puede trabajar de forma colegiada con otros docentes en la creación de arreglos.

Por otro lado, en las escuelas de música el alumnado aprende, por un lado, a utilizar programas de edición de partituras y por otro, en los últimos cursos de Lenguaje Musical realizan pequeñas composiciones entre grupos de estudiantes. Cada vez se le da más importancia al trabajo colaborativo y al uso de las TIC en el proceso de enseñanza – aprendizaje y por tanto, sería interesante que el alumnado pudiera crear sus composiciones de forma colaborativa y realizar otras actividades similares en una plataforma gestionada por el centro.

Por su utilidad, decidí desarrollar este proyecto y utilizarlo en la escuela de música en la que imparto clase y así, poder atender a las exigencias del usuario final.

1.2. Descripción del proyecto: sitio web

Este Trabajo Fin de Grado tiene como objetivo principal trabajar con partituras musicales de forma colaborativa. Con este propósito se va a adaptar un entorno virtual de aprendizaje para la creación y administración de cursos en el que se incluyan actividades de almacenamiento y reproducción de partituras online. Para poder trabajar de forma colaborativa cada usuario tendrá acceso a uno de los instrumentos de la partitura y podrá crear una partitura general uniendo las partes creadas por todos los usuarios de un grupo. Esta partitura se podrá visualizar y reproducir en la web y podrá descargarse en formato MIDI.

Se trata de crear un sitio web privado para realizar actividades musicales dirigido al alumnado y profesorado de escuelas de música, al aula de música de escuelas de educación primaria y secundaria y a los conservatorios. La novedad de estas actividades reside en la posibilidad de trabajar en grupo en la edición, modificación y

escucha de partituras musicales. Este servicio requerirá el almacenamiento de antiguas versiones de una misma partitura garantizando la imposibilidad de perder un archivo de forma irrevocable.

Se podrá trabajar desde cualquier ordenador con acceso a internet y el trabajo quedará almacenado en la nube. No obstante, los usuarios deberán utilizar algún editor de partituras de escritorio como Sibelius, Finale o MuseScore (este último con Licencia GNU) y trabajar off-line para más tarde subir los archivos a la plataforma.

Se estudiará la posibilidad de crear un entorno virtual de aprendizaje con Licencia Pública GNU (GPL), utilizando Moodle, que permita la creación de diversos usuarios (administradores, docentes y estudiantes) con diferentes roles y permisos. El docente podrá crear actividades y dirigirlas a grupos concretos de estudiantes. Al mismo tiempo podrá dar permisos para visualizar o editar partituras a cada uno de los estudiantes. Este espacio virtual permite agregar tareas provenientes de distintos sitios web.

Uno de los requisitos del proyecto es la creación de un espacio privado en el que el administrador gestione los usuarios para garantizar la seguridad de los estudiantes menores de edad. De esta forma el centro se hace responsable de la creación de cuentas, de su control y tiene la posibilidad de eliminar mensajes, entradas o comentarios que contradigan las normas. La decisión de crear cuentas personales o impersonales dependerá del centro.

Por último, el profesorado también deberá tener la posibilidad de crear composiciones musicales de forma colaborativa. Para ello, se creará un espacio de colaboración en el que podrán generar partituras completas y así, compartir la edición con el resto del claustro. Se considerará la posibilidad de incluir un espacio de almacenamiento de arreglos del centro para futuros cursos académicos.

Con las necesidades expuestas se va a programar un sitio web en el que se incorpore un espacio virtual de aprendizaje. Este espacio contendrá un curso de Lenguaje Musical, otro de Agrupaciones Instrumentales y un Espacio Para El Profesorado con actividades de colaboración musical. Para ello, se desarrollará una aplicación web que permita almacenar particellas y juntarlas consiguiendo una partitura general. Esta partitura final se visualizará y reproducirá online. En definitiva, se diseñan tres páginas web: una para almacenar particellas en grupos, otra para juntar las partes, visualizar y reproducir la partitura resultante y la última, para recuperar antiguas versiones.

1.2.1. Posibles aplicaciones.

1.2.1.1. Educación musical

Este proyecto está dirigido a la educación musical con la utilización de las nuevas tecnologías. A su vez, las clases de música se pueden dividir en tres categorías: las clases instrumentales individuales, las clases de interpretación grupales (coros o combos) y las clases de teoría musical (Lenguaje Musical, Armonía, Análisis, etc.). La utilización de la herramienta de colaboración para la edición de partituras que se creará con este proyecto tiene cabida en los siguientes escenarios:

a) Partituras colaborativas en el aula de teoría musical

La interdisciplinariedad entre las diversas áreas de conocimiento supone una innovación educativa de los últimos tiempos, especialmente por la incorporación de la cultura audiovisual. La LOMCE, en la Orden EDC/65/2015, de 21 de enero, exige un aprendizaje funcional que favorezca la enseñanza orientada a la acción a través de la integración de diversas áreas. Lo cual conlleva cambios en la organización, en los métodos de enseñanza y prácticas de trabajo tal y como afirman Giráldez (2005) y Galera (2013).

Por ello, las nuevas generaciones están trabajando con nuevas metodologías en el aula: se trabaja por proyectos y de forma colaborativa. Con estos objetivos y con la necesidad de trabajar la creatividad se han creado actividades de composición musical o edición de partituras en grupo. Hoy en día, esta creación se puede realizar con ayuda de las nuevas tecnologías y podría realizarse mediante documentos colaborativos. Como ya se realiza con otro tipo de documentos, como los documentos de texto. Estas actividades recomendadas por Lorenzo y Extremiana (2009) promueven la colaboración, motivan al alumnado, desarrollan la creatividad, favorecen el desarrollo personal y multidisciplinar y permiten la preparación a experiencias de la vida real.

b) Partituras colaborativas entre el profesorado

En las escuelas de música se trabajan partituras a todos los niveles y además, se realizan agrupaciones en función de las matrículas. Por ello, es difícil encontrar un repertorio adecuado para estas clases y se tienen que realizar arreglos musicales. Estas partituras se utilizarán tanto en clases de agrupaciones instrumentales así como en los conciertos. No obstante, no todos los docentes tienen nociones de composición y por ello, no conocen la escritura para todo tipo de instrumentos. En este sentido, cada docente realiza los arreglos correspondientes a su instrumento y posteriormente

se unen todas las partes. La posibilidad de crear partituras de forma colaborativa facilitaría este trabajo entre docentes.
Calderón, Martín y Gustems (2017) proponen incorporar al aula de música actividades de colaboración utilizando el software libre. Proponen distintas actividades con el uso de MuseScore y ampliar este trabajo colaborativo a la utilización de otras herramientas como Audiotool y Audacity.

1.2.1.2. Bases de datos o archivos musicales

Las escuelas de música, conservatorios, bandas u otro tipo de agrupaciones musicales disponen de numerosas partituras. Durante muchos años estas partituras se han ordenado en archivos físicos y desde hace años se han empezado a digitalizar estos materiales. Hasta ahora las bases de datos se creaban con documentos en PDF. Pero una aplicación de este proyecto sería crear bases de datos (transcritas a notación musical) de forma colaborativa para el uso privado por distintas agrupaciones o instituciones musicales.

1.2.2. Descripción del usuario.

Este sitio web está relacionado con la enseñanza a través de las nuevas tecnologías (e-learning), en este caso focalizado a la enseñanza musical y más concretamente a las escuelas de música y conservatorios. No obstante, también podría ser utilizada en las escuelas de educación primaria y secundaria en la asignatura de Música pero, para ello, el nombre de los cursos sería modificado.
El nivel musical y/o la edad del alumnado de las escuelas de música es muy variado. Se propone trabajar en un entorno virtual de aprendizaje que permita realizar la gestión de los usuarios desde el centro, de forma particular. Los usuarios con rol de alumno tendrán conocimientos informáticos básicos ya que habrán trabajado previamente con editores de partituras. No obstante, se incluirá en los cursos un apartado para que el alumnado pueda realizar y responder preguntas a cerca del funcionamiento de la aplicación.
Por otro lado, teniendo en cuenta que las escuelas de música cuentan con pocas ayudas económicas se prioriza la tecnología de bajo coste. Para ello, se propone trabajar en software de Licencia GNU y gratuito.
Por otro lado, se tendrá en cuenta que los equipos de trabajo de los centros serán principalmente PCs y los navegadores más utilizados Mozilla Firefox y Google Chrome. Por último, esta web debe ser accesible desde tabletas y dispositivos móviles para poder trabajar en clases en las que el número de ordenadores sea limitado.

Usuario:	Estudiantes	Docentes
Sexo	ambos	ambos
Edad	8 - 75 años	25 - 65 años
Conocimientos informáticos	Nivel bajo-medio	Nivel medio
Plataformas más utilizadas	1 Móvil 2 Tablet 3 Ordenador	1 Ordenador 2 Móvil 3 Tablet
Navegadores	Firefox y Chrome	Firefox y Chrome
Utilidad del proyecto en escuelas de música	Alta	Alta
Utilidad en escuelas de educación obligatoria	Alta	Baja

Tabla 1: Características del usuario.

Se han realizado una serie de entrevistas previas al inicio del desarrollo del proyecto a trabajadores de centros de enseñanza obligatoria y escuelas de música de Navarra para conocer sus características y su opinión sobre el proyecto. Se han considerado los apartados mostrados en la tabla 1, es decir, se han centrado en el usuario de la futura web. Por otro lado, se ha recabado información sobre los recursos informáticos de los centros y el presupuesto en tecnología. Las escuelas de educación obligatoria cuentan con aulas de informática pero, en las escuelas de música el número de equipos informáticos es más reducido y hoy en día en varios centros utilizan las tablets del alumnado en el aula.

1.3. Estructura del Trabajo Fin de Grado.

Este Trabajo Fin de Grado consta de dos partes principales: el desarrollo del producto o el sitio web y la memoria del proyecto. El sitio web está formado por un entorno virtual de aprendizaje, una página web para guardar partituras, una base de datos para almacenar las partituras correspondientes a cada usuario (particellas), y otra página para poder crear partituras completas o generales. También se añade una última página web para recuperar antiguas versiones de las particellas guardadas. La siguiente figura muestra los entregables en forma de WBS:

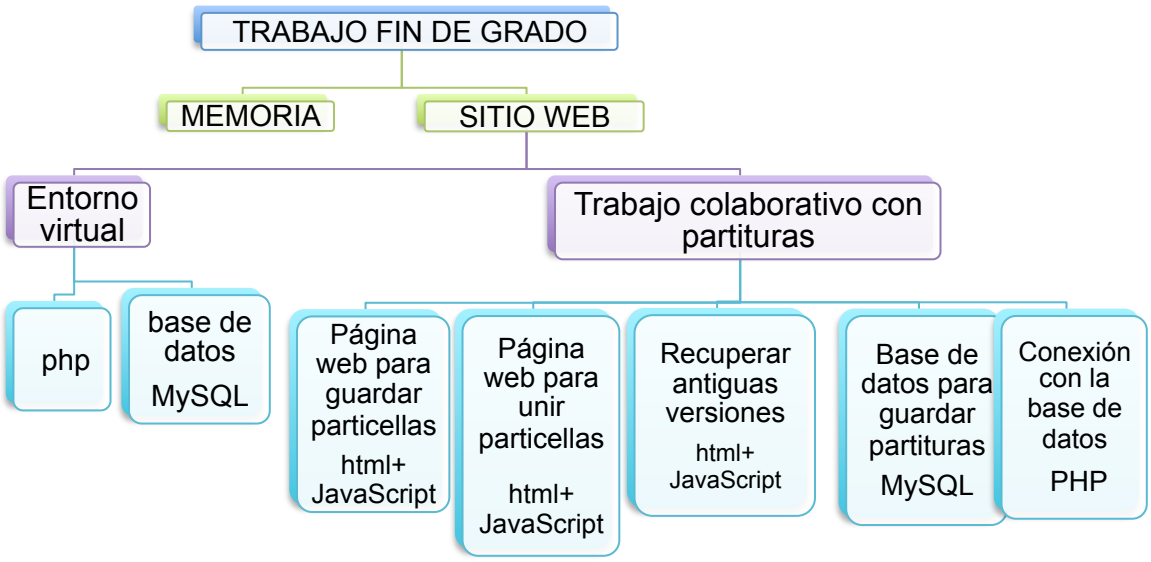


FIGURA 1: Estructura del TFG.

2. Objetivos del proyecto

2.1. Objetivos generales del proyecto

- Trabajar con partituras de forma colaborativa. Crear partituras generales uniendo partituras de una voz creadas por distintos usuarios.
- Establecer un espacio virtual de aprendizaje para almacenar y crear actividades relacionadas con la edición musical compartida.
- Guardar antiguas versiones del proceso de creación o edición de partituras.

2.2. Objetivos específicos

- Adaptar un espacio virtual utilizando la plataforma Moodle.
- Crear usuarios con distintos roles: administrador, docente y estudiante.
- Crear tres cursos e introducir usuarios.
    - Lenguaje Musical.
    - Agrupaciones Instrumentales.
    - Espacio de almacenamiento para el profesorado.
- Crear actividades para trabajar y crear partituras de forma colaborativa.
    - Crear una página web para que cada usuario pueda almacenar una particella o partitura correspondiente a su instrumento. Las partituras serán almacenadas en una base de datos junto a las partituras creadas por otros miembros de su grupo.
        - El usuario podrá almacenar nuevas versiones de su particella y las versiones anteriores serán almacenadas directamente en otra tabla de la base de datos.
    - Crear una página web para visualizar las partituras creadas por cada grupo y crear una partitura general a partir de las partes. La partitura final se podrá visualizar en pantalla, reproducir y realizar pequeñas modificaciones de tempo y de tonalidad. Por último, se podrá descargar la partitura resultante en formato MIDI.

- Modificar los derechos de edición o visualización de dichas actividades:
    - Creación colaborativa de partituras por el alumnado. El resto de usuarios del curso tendrán acceso a la visualización del trabajo de todos los grupos. Los miembros del grupo podrán editarla. (Curso de Lenguaje Musical).

- El administrador del aula virtual podrá crear una partitura y dar acceso a otros docentes para modificar la parte de su instrumento. El alumnado y el propio profesorado tendrán acceso la visualización de las mismas. (Curso de Agrupaciones Instrumentales).
- El profesorado podrá almacenar arreglos creados a lo largo de cada curso. (Base de datos para el centro).

3. Estado del arte

Las TIC han adquirido un importante papel dentro de la educación en los últimos años y ofrecen distintas posibilidades didácticas para la educación musical. Entre ellas la edición de partituras, el empleo de medios audiovisuales o la realización de grabaciones. También facilitan el almacenamiento de materiales, el acceso a información y establecen un nuevo canal de comunicación.

3.1. Tecnología software

Este Trabajo Fin de Grado tiene como objetivo principal trabajar en el aula de música de forma colaborativa, tanto entre estudiantes como entre el profesorado. Con este propósito se va a adaptar un entorno virtual de aprendizaje para la administración de cursos en el que se incluyan actividades de edición de partituras de forma colaborativa. Para ello, se requieren dos software: uno para ayudar a los docentes a crear cursos online y entornos de aprendizaje virtuales; y otro para la edición de partituras. Los dos apartados siguientes describen la tecnología software disponible.

3.1.1. Entorno virtual

A partir de los resultados de las entrevistas realizadas en centros de enseñanza musical y teniendo en cuenta la falta de presupuesto de los mismos, se van a considerar únicamente las plataformas de Licencia GNU para crear entornos virtuales de aprendizaje:

- Moodle es un software diseñado para crear cursos en línea de alta calidad en entornos de aprendizaje virtuales. Estos entornos virtuales de aprendizaje facilitan al docente la creación de cursos, actividades y la evaluación online del alumnado. Una de las principales características de Moodle es que está hecho en base a la pedagogía social constructivista, por ello, se le da gran importancia a las herramientas de comunicación como los foros, correo electrónico, chat, etc. Moodle se publicó utilizando la Licencia Pública GNU (GPL) y puede ser utilizado gratuitamente. Además, la plataforma Moodle conforma un sistema permanentemente activo y en constante evolución (Dougiamas, 2002).

  A continuación, se comentan algunas de las ventajas y facilidades que puede ofrecer Moodle a este proyecto:

  - Sistema en constante evolución y actualización.
  - Licencia GNU.

  - Posibilidad de personalizar la plataforma.
  - Creación de diversos perfiles de usuarios (administrador, tutor, alumno). Permite la gestión interna de usuarios apropiada para los estudiantes menores de edad.
  - Importación y exportación de datos en formato SCORM.
  - Sistema escalable en cuanto al número de alumnos y alumnas.
  - Creación de cursos virtuales y entornos de aprendizaje virtuales.
  - Posibilidad de diversos métodos de evaluación y calificación.
  - Accesibilidad y compatibilidad desde cualquier navegador web, independiente del sistema operativo utilizado. (Equipo editorial Linux, 2016).

- Sakai es un entorno de colaboración y aprendizaje para la educación superior. Para gestionar el Proyecto Sakai se creó una fundación a la que pertenecen más de 100 universidades. Al igual que Moodle se trata de un software educativo de código abierto y, a diferencia del anterior, está programado en Java.

  Sakai está diseñado para que el aprendizaje pueda ser colaborativo y tiene alta flexibilidad para crear y diseñar cursos, lo cual puede facilitar la creación de espacios para el trabajo colaborativo entre docentes. Por otro lado, incluye muchas de las características comunes a los entornos virtuales de aprendizaje, incluyendo distribución de documentos, calificaciones, foros de discusión, chat en vivo, tareas y pruebas en línea. Además, Sakai incluye la posibilidad de cambiar las configuraciones de todas las herramientas basadas en roles, permitiendo modificar los permisos de cada usuario (Apereo Foundation, 2014).

Al igual que Moodle, Sakai contiene todas las herramientas básicas para crear un entorno de enseñanza virtual. Dispone de herramientas de enseñanza – aprendizaje, de colaboración y portafolios. Está diseñado para que el aprendizaje pueda ser colaborativo por lo que podría incorporarse en el proyecto en lugar de Moodle. Por otro lado, tiene alta flexibilidad para crear y diseñar cursos, lo cual puede facilitar la creación de espacios para el trabajo colaborativo entre docentes. Se trata de una arquitectura abierta que permite integrar herramientas de otras arquitecturas. Al igual que Moodle, facilita la gestión de usuarios y permite cambiar las configuraciones de las herramientas o permisos de los usuarios (docentes o estudiantes).

Ambos permiten crear un entorno virtual de aprendizaje privado en el que la gestión de usuarios la realiza el administrador para evitar problemas con menores de edad. No obstante, se decide utilizar Moodle en el proyecto por su fácil manejo y porque Sakai

está orientado a la enseñanza superior. Por otro lado, Sakai está programado en Java y Moodle en PHP con bases de datos gestionadas con MySQL al igual que la base de datos de partituras que se creará en este proyecto y su correspondiente lectura.

3.1.2. Software de edición de partituras

Entre los software de edición musical más conocidos y utilizados destacan:

- Sibelius: una aplicación de escritorio comercial la cual proporciona una interfaz clara e intuitiva para escribir música. Se utiliza para transcribir partituras musicales y, con el uso de otros complementos, permite exportar las partituras a otros formatos como MusicXML, PDF, MIDI o audio.
- Finale: se trata de otra aplicación de escritorio comercial utilizada para escribir o transcribir música, y que también permite importar e exportar partituras a los mismos formatos.
- Denemo: es un editor de partituras (GPL), disponible para distintas plataformas.
- MuseScore: es un programa de notación musical que está **disponible para Mac, Windows y Linux**. Lanzado bajo licencia GNU (GPL), es libre, gratuito y de código abierto. Ha sido desarrollado en C++ y ha ido evolucionando en los últimos años gracias a la colaboración de la comunidad. El software consiste en un editor WYSIWYG con soporte completo para partituras e importación y exportación de MusicXML y otros ficheros MIDI estándar. MuseScore tiene una interfaz de usuario limpia, con una entrada de notas en edición similar al ingreso rápido que tienen otros programas comerciales de notación musical como Finale o Sibelius. Entre las capacidades más destacables de MuseScore se encuentran: la importación y exportación a distintos formatos musicales, la importación de ficheros nativos de otros programas musicales como Band-in-a-Box, la generación de partituras en PDF, SVG o PNG y la importación de ficheros GuitarPro.

Sibelius, Finale o MuseScore son aplicaciones útiles para escribir música, pero carecen de soporte directo para el trabajo distribuido y colaborativo. MuseScore dispone de una herramienta web que permite el almacenamiento, descarga en distintos formatos y reproducción de partituras online. También está disponible para el sistema operativo *Android* desde 2015. En la página web de MuseScore es posible crear grupos privados para la visualización de partituras entre docentes y el alumnado. Con estos grupos privados, el alumnado podrá entregar de forma privada el trabajo y

recibir comentarios por parte del docente. No obstante, no permite el trabajo colaborativo en forma de wiki.

El objetivo de este proyecto es trabajar de forma colaborativa con partituras musicales y la aplicación web de Musescore todavía no lo permite. Sin embargo, dadas sus características este software puede utilizarse para la edición off-line de particellas:

- Software profesional para la notación musical.
- Gratuito.
- Código fuente abierto.
- Transfiere desde y hacia otros programas musicales mediante MusicXML, MIDI y otros formatos.

MuseScore es una herramienta útil para utilizar con el alumnado en el proceso de enseñanza – aprendizaje en la escritura y reproducción de obras musicales. Estimulando la capacidad creativa mediante actividades de composición y escritura (Carrera, 2017). Sin embargo, las alternativas de pago son igual de válidas y la elección del editor de partituras dependerá del usuario.

3.2. Tecnología de programación

En este apartado se realizará un análisis de las tecnologías que mejor se adaptan a las necesidades de este proyecto y que cumplan los requisitos establecidos (Aydeé, 2015). La interfaz de usuario del sitio web de este proyecto se podrá desarrollar utilizando HTML, CSS, y JavaScript. Y para la parte de administración del sitio y la base de datos web se utilizarán PHP y MySQL. Moodle esta programado en PHP, sin embargo, permite realizar modificaciones desde la cuenta de administrador.

- HTML 5: se trata de la quinta versión del lenguaje HTML (HyperText Markup Language), un lenguaje de programación web. Esta versión permite mayor compatibilidad con los distintos navegadores que las versiones anteriores y su código es más sencillo.
- CSS (hojas de estilo en cascada): es un lenguaje dirigido al diseño para definir la presentación de un documento HTML.
- JavaScript: es otro lenguaje de programación interpretado, orientado a objetos, que se ejecuta en un navegador al igual que HTML. JavaScript permite realizar acciones e interacciones en un sitio web como validar formularios o realizar comunicaciones con el servidor mediante Ajax. Su interacción con la librería DOM proporcionara un conjunto de objetos para representar documentos HTML.

- Librería jQuery: es una librería JavaScript que permite simplificar la forma de interactuar con los documentos HTML, manipular el árbol DOM y agregar interacción con la técnica Ajax.

- PHP: es un lenguaje de programación web Hipertext Prepocessor que permite interactuar con el usuario y utilizar bases de datos. Será necesario utilizar este lenguaje para interactuar con la base de datos del proyecto y por otro lado, Moodle está programado en PHP.
- MySQL: es uno de los sistemas de gestión de bases de datos más utilizados por ser software libre. Existen otros sistemas libres como PostgreSQL o SQLite, y otros de pago. Sin embargo, Moodle utiliza el sistema MySQL y por tanto, se utilizará la misma tecnología para la base de datos de este proyecto. Por otro lado, PostgreSQL consume más recursos y es más lento que MySQL.

3.3. Software ABC

La notación ABC es un lenguaje para escribir música creado por Chris Walsahaw. Es una notación parecida a un lenguaje de programación que fue creado inicialmente para transcribir composiciones de música folk y melodías tradicionales inglesas, escocesas e irlandesas de una sola voz. Este sistema está basado en caracteres ASCII y por tanto, puede crearse mediante cualquier editor de texto.

Por otro lado, existen varios paquetes de software que permiten leer y procesar música escrita en sistema ABC. La mayoría de este software es de distribución libre o shareware y está disponible para distintos sistemas operativos (Microsoft Windows, Unix/Linux, Macintosh) y también hay un conjunto de software para la web.

Las características de esta notación, es decir, su simplicidad y la disponibilidad de software de libre distribución, la hacen adecuada para este proyecto. Las necesidades de este proyecto serán: la posibilidad de visualizar partituras online, reproducirlas y guardarlas en formatos compatibles con los editores de partituras más utilizados.

No obstante, no todos los editores de partituras permiten guardar los archivos en notación ABC. Por ello, se busca otro formato de entrada libre y compatible con cualquier editor de partituras: MusicXML. Utilizando las librerías de conversión del formato de entrada MusicXML a la notación ABC podrán utilizarse los programas de ABC disponibles para la web.

1) Conversión MusicXML a notación ABC.
    - Mxml2abc: es una herramienta que convierte archivos MusicXML de una o varias voces en archivos ABC. Mxml2abc se utilizó para convertir

más de 1700 villancicos de Navidad de XML a ABC, pero es probable que haya posibles mejoras y errores.

- xml2abc-js es una librería JavaScript que traduce MusicXML a notación ABC. Se trata de una traducción línea por línea de la versión de Python xml2abc.py. xml2abc.js es una parte del programa abcweb que procesa un archivo (MusicXML o ABC), lo muestra en pantalla y reproduce un archivo multimedia (audio o video) sincronizado con la partitura. Al tratarse de software JavaScript se ejecutará en el navegador.

2) Visualizar y reproducir una partitura a partir de la notación ABC.
    - ABC notation plugin (para wordpress) incluye el sistema abc.js en un sitio WordPress. Esta aplicación permite visualizar en pantalla y reproducir una partitura a partir de su correspondiente notación en ABC. Por otro lado, también permite la conversión de este formato a MIDI.
    - ScoreRender es un complemento de Abel Cheung, también para el sistema de blogs WordPress, que convierte fragmentos de partituras en imágenes. Admite música escrita en notación ABC y otros sistemas de notación.
    - ABC Viewer lee las canciones en formato ABC y muestra la notación musical correspondiente. Actualmente está en desarrollo, le faltan varias características y tiene algunos errores.
    - Abcjs es una librería JavaScript para representar notación musical ABC. Esta biblioteca facilita la incorporación de partituras en sitios web. Convierte el texto ABC en partituras, genera archivos MIDI y puede reproducirlos directamente en el navegador. No obstante, la parte de audio de esta biblioteca tiene limitaciones ya que no funciona en Internet Explorer y solo funciona con Safari 9 y 10. Se necesitan recursos del ordenador o dispositivo para reproducir el sonido, por lo que se necesita que sea lo suficientemente rápida.
    - abc2svg es una librería JavaScript que traduce la notación ABC a SVG que permite visualizar la partitura correspondiente en el navegador. Es posible utilizarlo combinado con abc2midi para reproducir la partitura resultante.
    - abc2midi es un programa para crear archivos MIDI a partir de archivos ABC. Contiene funciones especiales, como el manejo de archivos de

múltiples voces, la transposición de voces y la incorporación de acompañamiento de percusión.

3) Guardar la partitura resultante.
   A partir de la transformación del archivo ABC a MIDI se podrá descargar este último desde el navegador para poder leerlo con cualquier editor de partituras.

4) Editor de ABC.
   - Abcjs una librería JavaScript de Gregory Dyke y Paul Rosen para renderizar archivos ABC en páginas web, analizada en el apartado 2. Este programa permite transformar un área de texto en un editor de ABC, visualizar su correspondiente partitura y reproducirla como MIDI. Aunque en este proyecto no es completamente necesario se puede utilizar como una función añadida.

3.4. Éxitos en el ámbito de colaboración musical.

Hoy en día, las partituras musicales son el principal material de trabajo tanto en escuelas de música como en agrupaciones instrumentales de cualquier tipo. Para ordenar y almacenar el material musical se han empezado a digitalizar estos materiales.

- IMSLP: entre los archivos de dominio público, uno de los proyectos más conocidos es Petrucci Music Library. Se trata de un proyecto basado en tecnología wiki que tiene como objetivo crear una biblioteca virtual de partituras de dominio público. Todas las partituras se almacenan escaneadas en PDF.
- Mutopía: el Proyecto Mutopía pretende crear un catálogo virtual de partituras de música clásica (de dominio público) o licencia Creative Commons similar al proyecto anterior. Hoy en día, el Proyecto Mutopía se compone principalmente de un repertorio pianístico. La diferencia principal con ISMLP es que las partituras se almacenan transcritas en formato LilyPond, un software libre de edición musical que permite exportar los archivos a formatos como PDF o MIDI de alta calidad. En este caso, no se aceptan partituras escaneadas.
- MuseScore: su página web contiene una base de datos de partituras creadas por los usuarios de la propia web, MuseScore.com. Permite escoger entre guardar la partitura de forma privada para el acceso personal desde cualquier ordenador, o compartirla públicamente. MuseScore.com también facilita la

visualización y reproducción de partituras en el navegador web e incluso permite sincronizar las partitura con un vídeo en YouTube. Estas partituras pueden ser descargadas en distintos formatos (PDF, MIDI, MP3, XML o el archivo original de MuseScore).

- En la música electrónica se ha creado una aplicación Android colaborativa mediando entre herramientas de composición musical de un solo usuario y la improvisación de música colaborativa. En esta aplicación se intentan combinar las ventajas de la improvisación musical colaborativa y las aplicaciones orientadas a un único usuario para una composición de música sistemática e iterativa que de forma sincronizada, colaborativa y compartida que permita la composición y reproducción musical. (Klügel, FrieB, y Groh, 2011).
- En el Trinity College se han hecho estudios sobre la adaptación de aplicaciones de composición para percusión como DrumSteps para su uso en plataformas de colaboración. Este software utiliza un sistema de notación más sencilla exclusiva para percusión que le da al usuario la oportunidad de construir y explorar conceptos musicales de una manera intuitiva (Bligh, Jennings, y Tangney, 2006).

Ha habido otros intentos para crear bases de datos de partituras digitales como el proyecto WIMA o Wikifonia que han sido fusionados con otros proyectos como ISMLP y la base de datos del sitio web de MuseScore, respectivamente. No obstante, estos proyectos se alejan del objetivo de este proyecto el cual está dirigido al uso privado y la enseñanza para trabajar en la edición de partituras en grupos.

3.5. Modelos de colaboración y problemas en los que se trabaja actualmente.

3.5.1.1. Modelo de documentos colaborativos con la Web 2.0

La Web 2.0 hizo posible administrar textos, plantillas de cálculo o presentaciones en línea sin la necesidad de instalar software específico para ello. A través de la conexión a internet y mediante algunos recursos gratuitos se pueden crear estos archivos y editarlos de forma colaborativa. Esto es posible debido a que la información se almacena en la nube y no en el ordenador propio. Un ejemplo destacable es Google Docs. Es posible crear un documento desde cero online o subir un documento empezado para que otras personas puedan interactuar con él. De esta forma se evitan problemas de compatibilidad. Una vez terminados, los documentos se pueden exportar en distintos formatos. Por otro lado, se tiene acceso a todas las modificaciones que sufra un documento pudiendo recuperar antiguas versiones.

Siguiendo este modelo se pretenden dar los primeros pasos para trabajar de la misma manera con partituras online, para facilitar el intercambio de información, la cooperación y evitar trabajos duplicados y problemas de compatibilidad.

3.5.1.2. Matriz WIKI::SCORE

Los recursos musicales se comparten más comúnmente mediante partituras escaneadas o en hojas de papel. Estos documentos contienen mucha información, pero debido a que son imágenes, es difícil reutilizar y adecuar su contenido. Para ello, muchas veces se hace necesario transcribir el contenido de cada partitura. Esta tarea consume tiempo, debido a la complejidad del proceso y al tamaño de las partituras. Por ello, se pretende facilitar la transcripción de partituras mediante archivos con partituras transcritas.

WIKI::SCORE es un entorno colaborativo donde varias personas trabajan juntas para transcribir una serie de partituras propuestas por los creadores en un medio compartido utilizando la notación ABC. Esto facilita el proceso de transcribir documentos y su almacenamiento. Esta notación permite publicar todo el contenido en varios formatos como PDF, imágenes o archivos de audio. La notación ABC para la transcripción de música se basa en texto sin formato. Es muy compacto y con muchas herramientas asociadas ya disponibles. Se basa esencialmente en una representación directa de los elementos musicales como caracteres ASCII.

```
X:1
M:C
L:1/4
K:C
c2de | A>Bc2 | f>de2 | g2de | A>Bc2 | d>ec2 | d3e | c4 |
```

FIGURA 2: Ejemplo de notación ABC. La altura se representa con letras y la duración con números.

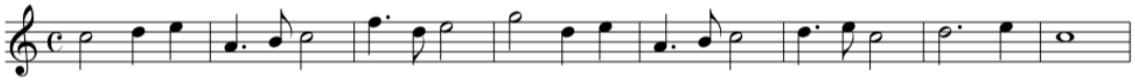


FIGURA 3: Partitura generada a partir del ejemplo de notación de la Figura 2.

ABC es la notación adoptada en este entorno colaborativo ya que permite transcribir la mayoría de partituras musicales y existen varias herramientas para transformar los archivos de texto. Además, se trata de un código abierto. En el proyecto Wiki::Score se decidieron por esta notación por su sencillez frente a los formatos LilyPond y

MusicXML que son menos compactos y su sintaxis es más complicada. Por otro lado, la notación ABC resulta más adecuada para la edición en línea en un entorno wiki.
En este proyecto se ha utilizado un entorno wiki porque permite la colaboración multiusuario, es decir, diferentes personas pueden agregar contenido al mismo tiempo y además, almacena un historial de archivos subidos por distintas personas (Almeida, Carvalho y Oliveira, 2012).
La wiki::score de Almeida, Carvalho y Oliveira (2012) se basa en una matriz de información para permitir la colaboración entre usuarios. Es decir, divide cada partitura en secciones e instrumentos, en columnas y filas respectivamente, para que cada usuario pueda trascribir la parte correspondiente a una celda. Se trata de construir de forma colaborativa una base de datos digital a partir de un manuscrito original. No obstante, sólo están disponibles las partituras propuestas por los creadores de la web y sólo se admiten transcripciones de la partitura original.
Wiki::Score es una plataforma para trabajar de forma colaborativa con partituras de música. Sin embargo, el usuario sólo podrá participar en la edición de partituras propuestas por los desarrolladores. El objetivo de los creadores es distinto al marcado en este proyecto, en su caso pretenden crear ediciones modernas de partituras "enterradas" o antiguas, es decir, realizar ediciones originales de música clásica.
El principio básico de funcionamiento de este proyecto puede ser aplicado al sitio web que se va a crear dividiendo una partitura en instrumentos. Es decir, puede aplicarse como método de colaboración. No obstante, este TFG esta dirigido a la realización compartida de arreglos o adaptaciones musicales para utilizar en actividades de enseñanza musical. Por tanto, a diferencia de Wiki::Score el usuario podrá elegir la partitura que quiera transcribir o los instrumentos que la compondrán. Por otro lado, se pretende incorporar posibilidades de reproducción de la partitura final para comprobar que el arreglo funcione.

3.5.1.3. LilyPond y proyectos de colaboración

LilyPond es un software de edición de partituras de Licencia GNU que utiliza una notación de texto como entrada y produce una partitura que puede ser exportada a distintos formatos como PDF o MIDI. Este programa no incluye una interfaz gráfica para la realización de partituras, funciona por líneas de comandos invocando a un archivo fuente. Este programa se caracteriza por la calidad de la salida porque incorpora reglas que evitan colisiones entre objetos, correcciones ópticas en función del espacio disponible y por el tamaño de la tipografía.

Se han creado varios catálogos de música transcrita en LilyPond, principalmente de música clásica de copyright vencido. Entre estos proyectos está el Proyecto Mutopía que incluye más de 1500 partituras de música clásica para su libre descarga, 'Etude' una aplicación para iPhone que presenta en pantalla música para piano editada con LilyPond mostrando las teclas del piano que hay que ir tocando y 'Adoro Music Publishing' de música sacra. Estos proyectos ponen a disposición del usuario todas las partituras, todas ellas de alta calidad, sin embargo, crecen lentamente por la dificultad de su escritura.

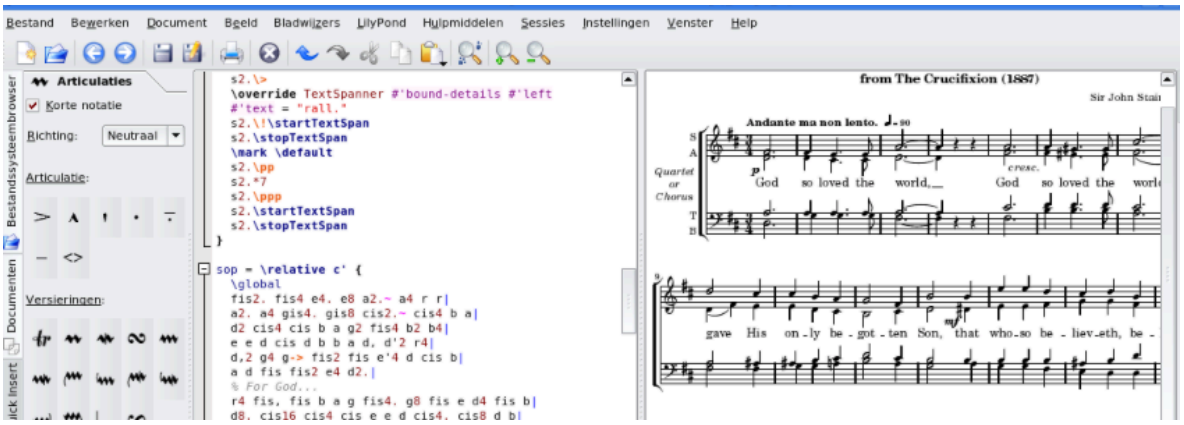


FIGURA 4: Ejemplo de partitura transcrita con LilyPond. Líneas de comandos y partitura.

LilyPond permite la inclusión de archivos externos para poder crear documentos que dependan de otros archivos. De esta manera, se puede trabajar de forma colaborativa. Es decir, se trabaja de forma colaborativa incorporando particellas de distintos instrumentos en una partitura general. Con esta opción de incorporar archivos externos se están intentando ampliar los archivos de partituras citados anteriormente.

No obstante, el software Lilypond no dispone de una interfaz gráfica para la creación o edición de partituras. LilyPond es un sistema que se ejecuta sobre un archivo de texto que describe la música. Es más parecido a un lenguaje de programación que a un programa de edición gráfica de partituras.

Lilypond se utiliza a nivel profesional, para crear partituras para conservatorios por su tipografía y porque permite representar métricas distintas en una partitura de piano o partituras de canto gregoriano, por ejemplo. Los programas de edición de partituras como MuseScore, Sibelius o Finale, con editores WYSIWYG son más utilizados por los músicos que no quieren invertir su tiempo en aprender la sintaxis de Lilypond. Por la dificultad del lenguaje de Lylipond se descarta para este proyecto.

4. Diseño del sitio web

En este apartado se describirán las propuestas de diseño lógico y los detalles técnicos del sitio web a partir del análisis de los requisitos del proyecto. Es decir, se van a detallar los componentes necesarios para cumplir esos requisitos y la forma en la que se comunicarán unos componentes con otros. Estos componentes deberán satisfacer los requisitos de usabilidad y el presupuesto establecido.

4.1. Arquitectura del sitio web

Primero, se define la arquitectura conceptual, la cual consiste en una arquitectura de tres capas (presentación, aplicación y datos). Con esta arquitectura los distintos niveles serán independientes unos de otros permitiendo cambios en algunas capas sin que influya en las demás.

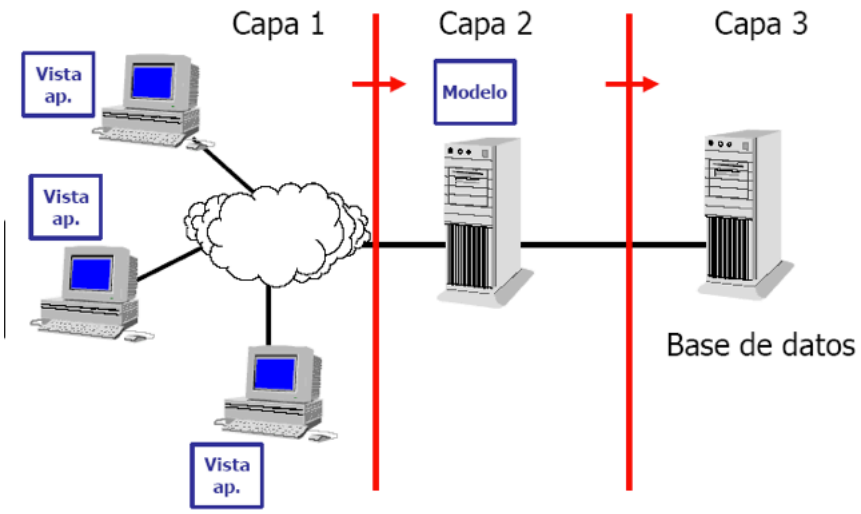


FIGURA 5: Arquitectura

- El primer nivel consiste en la capa de presentación. Esta capa proporciona los contenidos de la página web a la que el usuario accede.
- La segunda capa es la encargada de realizar las funciones de los procesos en la aplicación. Estas funciones toman decisiones para que el usuario pueda visualizar los datos que necesite. Sirve como puente entre las capas de presentación y de datos.
- El tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Las actividades de este sitio web recogerán datos del usuario, los guardará en la base de datos del servidor, estos datos serán tratados o procesados y se volverán a presentar al usuario.
Es en la segunda capa de la arquitectura descrita anteriormente donde se implementa la funcionalidad del sitio web el cual va a estar formado por un entorno virtual de aprendizaje conectado a un sitio web de actividades colaborativas con partituras.

4.2. Componentes del sitio web

4.2.1. Entorno virtual de aprendizaje

El objetivo de este proyecto es facilitar la colaboración en la edición de partituras en la enseñanza musical. Para ello, se decide utilizar un Sistema de Gestión de Aprendizaje (LMS) de código abierto que potencia las posibilidades de colaboración e interactividad de un espacio virtual dedicado a la enseñanza. Un LMS es un software que planifica y gestiona eventos de aprendizaje dentro de una organización como puede ser una escuela de música o un conservatorio al que va dirigido este proyecto.
Entre los LMS analizados en el apartado 3, se escoge Moodle por ser una plataforma de código abierto, porque permite crear usuarios con distintos perfiles y cursos virtuales e integrar actividades de otras páginas web. Además, tiene un sistema muy intuitivo y de fácil manejo para la administración de cursos y de usuarios, lo cual permite que los docentes sean quienes gestionen la página.

4.2.1.1. Requisitos del entorno virtual

- Crear cuentas de usuarios para distintos usos. Definición de roles:
    - Administrados o gestor del entorno virtual.
    - Docentes que puedan gestionar sus cursos.
    - Estudiantes.

- Crear cursos para trabajar con partituras de forma colaborativa.
    - Curso para que los estudiantes trabajen en grupos en la edición de partituras musicales.
    - Curso para que los docentes trabajen en la edición de partituras de forma colaborativa y los estudiantes puedan acceder a los contenidos creados por los docentes.

- Crear un espacio para almacenar partituras para el centro.
- Integrar en la plataforma las actividades de edición de partituras de forma colaborativa.
- Personalización de la plataforma: apariencia e idioma.

4.2.1.1. Diseño del entorno virtual

Se va a adaptar Moodle como entorno virtual de aprendizaje en el cual se crearán dos cursos dirigidos a la enseñanza musical. En el primero de ellos, el alumnado podrá trabajar en grupos creando partituras de forma colaborativa. En el segundo, los docentes crearán partituras de forma colaborativa para poder utilizar dichos arreglos en las clases. En este segundo curso el alumnado podrá tener acceso a la visualización de los arreglos creados por el profesorado, pero, no tendrá permiso para almacenar sus partituras.

Por otro lado, se quiere crear un espacio o archivo de partituras para el centro. Para ello, se establecerá un nuevo curso en el que sólo se matricularán los docentes. En él se incluirá una base de datos para almacenar partituras.

Para poder diferenciar los permisos entre docentes y estudiantes, los usuarios van a ser matriculados con distintos roles: profesor o estudiante. Por último, el usuario con rol de administrador será quien gestione y personalice toda la página.

4.2.2. Actividades de partituras colaborativas.

En las escuelas de música se utilizan las partituras como principal herramienta de trabajo y debido a las características de cada agrupación se realizan arreglos de las partituras originales adecuándolas a los intérpretes. Estos arreglos se realizan de forma conjunta entre el profesorado y cada profesor o profesora es responsable de la parte de su instrumento. Para facilitar la edición colaborativa de partituras surge se propone esta actividad.

4.2.2.1. Primera propuesta: actividades con Musescore

MuseScore junto con Sibelius y Finale es uno de los editores de partituras más utilizado entre los músicos. Es una de las primeras opciones en las escuelas de música, conservatorios y universidades por ser de código abierto y gratuito. Por otro lado, dispone de un archivo de partituras online (musescore.com) que permite el almacenamiento de un máximo de cinco partituras donde cada usuario podrá visualizar, reproducir y descargar cualquier partitura compartida por otros miembros.

A partir de las ventajas que ofrece MuseScore, la primera propuesta del proyecto ha sido incorporar algunas de las funcionalidades que ofrece este programa y crear un entorno de colaboración para editar partituras online.

Con este planteamiento, el primer paso fue acudir a la web de desarrolladores de MuseScore. La cual contiene manuales y foros sobre el funcionamiento del software de escritorio y el modo de colaborar con los desarrolladores del mismo. Para facilitar las contribuciones de distintos colaboradores/as, los fundadores de MuseScore facilitan información sobre el lenguaje de programación utilizado en el desarrollo del programa (C++ y Qt) y los pasos a seguir para compilar este código en distintos sistemas operativos.

Partiendo de esta información se adaptó el entorno de trabajo y desde GitHub se realizó una clonación del código del programa. Una vez descargado, se comprobó que su utilización online requería reprogramarlo en un lenguaje de programación web. Se analizaron algunos trabajos anteriores relacionados y los resultados no habían sido satisfactorios. Los creadores de MuseScore consiguieron adaptar las funciones para visualizar, reproducir y descargar partituras a un lenguaje de programación web y así, utilizarlo en musescore.com. Sin embargo, este código no tiene Licencia GNU y su uso no está permitido.

Consultando en los foros de musescore.com se encontraron varios hilos que hablan sobre la posibilidad de utilizar MuseScore online. Muchos usuarios demandan esta posibilidad debido a que musescore.com está limitado. Cada usuario puede subir y compartir un máximo de 5 partituras, una cantidad muy reducida para utilizarlo en escuelas de música, conservatorios o por agrupaciones musicales. No obstante, las cuentas Premium de la web las cuales no tienen límite de partituras son su principal fuente de ingreso siendo el software de escritorio gratuito. Pero, las escuelas de música no tienen presupuesto para utilizar estas cuentas.

4.2.2.2. Nueva propuesta: aplicación para unir particellas.

A partir del análisis anterior se descarta la posibilidad de utilizar MuseScore o algunas de sus funciones para trabajar en su lugar con posibilidades gratuitas, más accesibles para las instituciones musicales de hoy en día.

Se propone una aplicación que permita crear una partitura general a partir de las particellas de varios instrumentos creados por distintos usuarios. Utilizando una combinación de visores y reproductores de partituras online los cuales admiten formatos abiertos de partituras (MusicXML o ABC). En concreto se utilizarán librerías de JavaScript añadiendo la funcionalidad del trabajo colaborativo. En definitiva, un

entorno de colaboración para crear partituras utilizando librerías JavaScript de Licencia GNU. Cada músico podrá encargarse de la edición de la partitura correspondiente a su instrumento (particella) y posteriormente unir todas las partes para comprobar el arreglo online.

Esta aplicación web se divide en tres páginas principales: la primera será para guardar particellas de los usuarios en la nube, la segunda para juntar particellas de distintos usuarios y crear partituras completas, y la última, se utilizará para recuperar antiguas versiones de las particellas subidas por los usuarios.

4.2.2.2.1. Requisitos de las páginas web.

Como se ha descrito en el apartado anterior esta aplicación permitirá crear una partitura general a partir de las particellas de varios instrumentos creados por distintos usuarios. Por tanto, los usuarios trabajarán en grupo en la creación de una partitura general siendo cada miembro del grupo el responsable de un instrumento.

- Página web para guardar particellas: un usuario podrá guardar y/o actualizar una particella para posteriormente juntarla con otras particellas de los miembros de su grupo en la siguiente página web.
    - Control de usuarios: al acceder a esta página deberá realizarse una identificación del usuario y del grupo al que pertenece.
    - Formulario para cargar una partitura correspondiente a un instrumento.
    - Envío del archivo con la información del usuario y el grupo a la base de datos.

- Página web para unir particellas: los usuarios podrán juntar las particellas que seleccionen de cada uno de los grupos. Cada grupo se dedicará a la realización de una partitura o canción.
    - Selección del grupo del grupo o de la canción.
    - Visualización de las particellas que el usuario seleccione.
    - Unir las particellas para crear una partitura completa. Esta partitura completa se podrá visualizar y reproducir en la web.
    - Descargar la partitura completa de cualquier grupo.

- Página web para recuperar antiguas versiones de particellas: el usuario podrá recuperar las antiguas versiones de sus particellas.
    - Control de usuarios: al acceder a esta página deberá realizarse una identificación del usuario.
    - Se mostrarán en pantalla el listado de las particellas subidas por este usuario.
    - El usuario podrá seleccionar alguna de sus partituras para visualizarla en pantalla y descargarla.

4.2.2.2.2. Tecnologías utilizadas.

Las páginas web serán escritas en HTML para describir la estructura y el contenido de las mismas. Estas páginas irán asociadas a una hoja de estilos común CSS que definirá la presentación de los documentos.

Para poder interactuar con los datos en estas páginas se va a utilizar el lenguaje JavaScript conjuntamente con jQuery y Ajax. JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML y ofrece una serie de funciones propias que ayudarán a simplificar el código.

Por otro lado, se va a crear una archivo PHP para realizar la conexión con la base de datos que será llamada desde las páginas HTML creadas. Por último, para cargar, manipular, reproducir o guardar las particellas y partituras se van a utilizar distintas librerías JavaScript creadas bajo licencia GNU que se han descrito en el apartado 3:

1. xml2abc.js: para poder trabajar con partituras en notación ABC.
2. abc2svg.js: para visualizar las partituras en pantalla.
3. abc2midi.js: para reproducir las partituras en la web y descargarlas en un formato compatible con los editores de partituras.

4.2.2.3. Base de datos

Estas páginas de actividades colaborativas con partituras requieren almacenar partituras con información sobre el usuario en una base de datos. Moodle proporciona la posibilidad de crear bases de datos para almacenar cualquier tipo de archivo. No obstante, se ha decidido crear otra base de datos independiente para poder reutilizar este sitio web cuando no se quiera emplear Moodle.

Se va utilizar el sistema de gestión de datos MySQL tanto en esta base de datos como en Moodle. Para poder almacenar y leer la información de la base de datos se va a crear otra página en PHP. Esta página contendrá las funciones necesarias para que el

usuario pueda almacenar y actualizar sus partituras, recuperar antiguas versiones y leer las particellas correspondientes a un grupo.

La base de datos estará formada por dos tablas principales: una para guardar las particellas de cada usuario y otra para guardar antiguas versiones de las mismas. En la siguiente tabla se muestra la información que se recogerá en las tablas de la base de datos:

Campos de las bases de datos:
Identificador usuario (número)
Identificador grupo (número)
Archivo de la particella (archivo MusicXML)
Nombre del archivo (cadena de caracteres)
Nombre del instrumento (cadena de caracteres)
Fecha de almacenamiento
Versión de la partitura (número)

Tabla 2: Estructura de los datos.

Por último, se creará una tabla para realizar la gestión de grupos. En esta tabla se relacionarán dos valores: el número de identificación del usuario con el número del grupo en el que trabaje.

4.3. Estructura de navegación.

Este sitio web se compone de un entorno virtual a través del cual se podrá acceder a las actividades de edición colaborativa de partituras. Estas actividades estarán alojadas en otra página web.

Para acceder al sitio web tanto el alumnado como el profesorado se identificará en el entorno virtual. Una vez en el entorno virtual podrán acceder a los distintos cursos que contendrán los enlaces a estas actividades. La actividad está dividida en tres páginas web cada una de ellas encargada de una función: la primera se utiliza para guardar particellas, la segunda para unir particellas de distintos usuarios y en la última se podrán recuperar antiguas versiones de las partituras guardadas por los usuarios.

En la siguiente imagen se puede observar la estructura de navegación general de todo el sitio web:

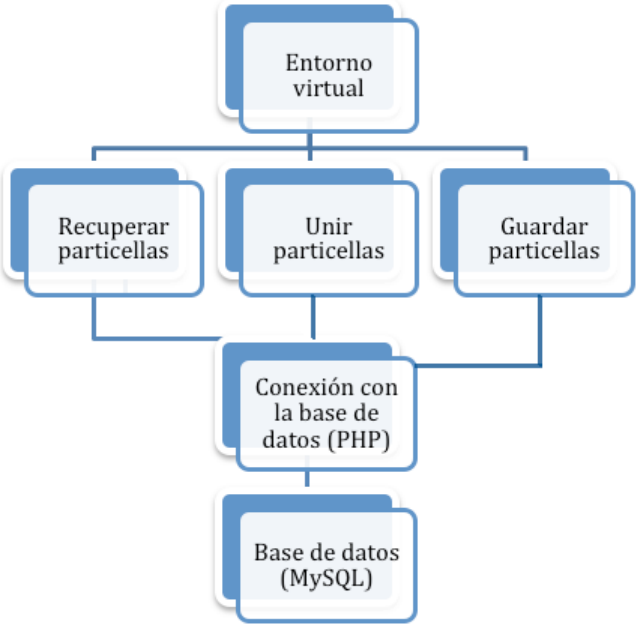


FIGURA 6: Esquema de navegación del sitio web.

Las páginas web de las actividades requieren comunicación con la base de datos para almacenar o extraer información. Para ello, utilizarán las funciones definidas en la página de conexión PHP.

En el entorno virtual habrá dos cursos para realizar las actividades de unir particellas. En el primer curso, el alumnado será quien trabaje en grupos para crear partituras generales. Por tanto, tendrá acceso a las páginas para guardar, recuperar y unir particellas. No obstante, en el segundo curso, únicamente el profesorado tendrá acceso a guardar y recuperar particellas y tanto docentes como estudiantes tendrán acceso a su visualización en la página web para juntar particellas.

Las páginas de las actividades cuentan con un menú principal para volver al entorno virtual y navegar entre las páginas de actividades.

4.4. Diseño gráfico o apariencia del sitio web.

Los elementos de diseño gráfico del sitio web se consideran la tipografía, el color y las imágenes utilizadas. Se va utilizar un mismo color para crear las cabeceras y pies de página, un color oscuro y letras blancas. Los contenidos se escribirán sobre fondo blanco con letras de colores negros y azules.

Por otro lado, se va a utilizar una imagen principal relacionada con la música y las partituras en todas las páginas web de actividades. Esta misma imagen se utilizará también en el entorno virtual para que la transición entre páginas no sea tan perceptible para el usuario.

Las páginas de actividades siguen la misma estructura general. En la parte superior habrá un menú para poder volver al entorno virtual o navegar entre las páginas de esta aplicación. En la parte inferior un footer o pie de página en el que se podrá incluir información sobre el centro de enseñanza. Por último, la parte central contendrá una misma imagen para todas las páginas y la sección de contenidos que variará en función de la página.

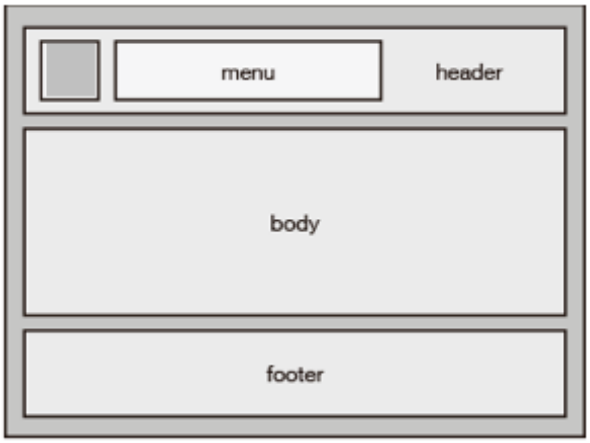


FIGURA 7: Estructura de las páginas de actividades.

4.4.1. Diseño adaptativo.

Teniendo en cuenta que uno de los requisitos de los usuarios es utilizar la aplicación en sus teléfonos móviles o tablets, se va a adaptar el diseño del sitio web a este tipo de pantallas. Moodle se adapta a cualquier pantalla y para las páginas web de actividades se realizarán una serie de cambios.

Primero, se adecuará el menú dispositivos móviles. Además, se modificará el tamaño de algunos contenidos para que ocupen el 100% del ancho de la pantalla cuando se detecten pantallas menores a 1000 píxeles. También se podrán eliminar algunos contenidos menos relevantes.

5. Desarrollo de la web

En este apartado se describirá el desarrollo del sitio web formado por un entorno virtual de aprendizaje, tres páginas web para realizar actividades de colaboración con partituras y una base de datos.

5.1. Entorno virtual

Como entorno virtual de aprendizaje se ha utilizado Moodle, una herramienta de gestión de aprendizaje LMS de distribución libre desarrollada en PHP. Esta plataforma implementa mecanismos de seguridad a lo largo de toda su interface y, además, es fácil de administrar.

Primero, se instala la plataforma en un servidor local para empezar a adaptarlo a las necesidades de este proyecto. Después, se utilizará Moodle Cloud, un entorno gratuito de Moodle que permite hacer uso de esta plataforma y modificarla a través de una cuenta de administrador. Este entorno se utiliza para realizar trabajos fin de grado o fin de máster. No obstante, presenta algunas limitaciones respecto a la instalación local que se detallarán a continuación.

5.1.1. Administración del sitio: apariencia e idioma.

Una vez instalada la plataforma y creado el usuario administrador se comenzará a configurar Moodle adecuándolo a las necesidades del proyecto. Desde la cuenta de administrador se crean los cursos necesarios y una serie de usuarios con distintos roles como se explica en los siguientes apartados.

Por otro lado, a través de la propia cuenta de administrador se realiza la configuración general de la plataforma: personalización del diseño, idioma, etc. Primero, se ha establecido como idioma por defecto el castellano. A continuación, en la instalación local se han probado varios temas disponibles en la página oficial de Moodle entre ellos el tema denominado eGuru. No obstante, Moodle Cloud sólo permite utilizar el tema Boost el cual ha sido ajustado a las exigencias de diseño del apartado anterior. Se han modificado los colores de algunas clases mediante la incorporación de código CSS y se ha establecido una imagen de fondo que será utilizada en las páginas web de actividades.

5.1.2. Administración de cursos

Uno de los módulos más importantes de Moodle son los cursos, los cuales permiten a los usuarios profesores/as crear actividades o añadir recursos que serán accesibles a

los estudiantes matriculados al curso. Este proyecto va dirigido a instituciones de educación musical y pretende facilitar el trabajo colaborativo de edición de partituras entre estudiantes o entre docentes. Con estos dos posibles usos se crearán dos cursos:

1. Lenguaje Musical. Se crea un curso de Lenguaje Musical en el que el alumnado trabajará en la edición de una partitura por grupos. Cada estudiante editará offline la partitura correspondiente a un instrumento, es decir, una particella, y la guardará en una base de datos a través de una actividad de Moodle. Después, podrá acceder a las partituras de todo el grupo y crear la partitura completa a través de otra actividad.

FIGURA 8: Curso Lenguaje Musical.

Este primer curso esta formado por tres temas dedicados a una actividad cada uno. Primero, se incluye información sobre el modo de trabajo en un foro en el cual los estudiantes podrán realizar y responder preguntas relacionadas con la metodología. El primer tema o actividad consiste en guardar una particella en formato XML a través de un enlace. El segundo consiste en un enlace a una web en la que se pueden juntar las particellas de cada grupo para crear una partitura completa y la tercera, una actividad para recuperar antiguas versiones de particellas.

2. Por otro lado, se ha creado un curso de Agrupaciones Instrumentales que sigue la misma estructura que el curso anterior sin la primera actividad del foro. No obstante, en este curso la primera y tercera actividad sólo serán visibles para los docentes, es decir, los docentes serán los únicos usuarios que tendrán permiso para guardar y recuperar partituras en la base de datos. La segunda actividad será accesible a todos los usuarios matriculados en el curso, de forma que docentes y estudiantes puedan escuchar y descargar las partituras completas creadas colaborativamente por los docentes.

3. Por último, se crea un curso para docentes formado por dos actividades ambas forman una base de datos de partituras para el centro de música. La primera base de datos dedicada a las clases del centro y la segunda a los arreglos para conciertos. Para ello, se utilizará una tabla en la base de datos de Moodle.

FIGURA 9: Base de datos para el centro.

En estos cursos los estudiantes serán matriculados con rol de estudiantes y los docentes con el rol de profesor. Este último tendrá el control sobre el curso y podrá añadir otro tipo de actividades al curso como cuestionarios, materiales, consultas o tareas. El docente también podrá calificar cada una de las tareas propuestas a los estudiantes.

5.1.3. Administración de usuarios

Moodle permite la creación y gestión de usuarios por el administrador de la página, lo cual facilita la gestión de usuarios menores de edad. Se han creado una serie de usuarios con distintos roles distinguiendo entre el administrador, profesor/a y estudiante. El administrador gestionará toda la página, incluyendo cursos y usuarios, y podrá dar distintos permisos a profesores/as y estudiantes. Los docentes podrán gestionar sus propios cursos.
A cada usuario se le asignará un número id que luego facilite su identificación en las páginas de actividades para guardar y recuperar partituras.

Nombre usuario	Contraseña	ID	Nombre usuario	Contraseña	ID
admin	-------	ID: 2	usuario3grupo1	Usuario3Grupo!	103
test			usuario4grupo1	Usuario4Grupo!	104
profesora	Profesora1%	5	usu1delgrupo2	Alumno11””	201
usuario1grupo1	Alumna1!	ID:101	usu2delgrupo2	Alumna11%%	202
usuario2grupo1	Alumno1!	ID: 102			

Tabla 3: Relación de usuarios Moodle.

5.2. Páginas web de actividades.

Como se ha propuesto en el apartado de diseño se van a crear tres páginas web en HTML para realizar las actividades de creación de partituras de forma colaborativa. Se pretende que los usuarios divididos en grupos puedan trabajar conjuntamente en la creación de una partitura. Cada usuario creará la partitura correspondiente a un instrumento y la almacenará a través de la primera página web (apartado 5.3). A continuación, el usuario podrá seleccionar las particellas creadas por su grupo y unirlas creando una partitura general (apartado 5.5). Por último, el usuario podrá recuperar antiguas versiones de sus particellas (apartado 5.6). Estas páginas tendrán la misma estructura (archivo HTML) y utilizarán una hoja de estilos CSS compartida. Todas ellas incorporarán funciones JavaScript para realizar las acciones de la página que serán analizadas en los apartados siguientes.
Las partituras serán almacenadas en una base de datos independiente de las bases de datos de Moodle. La lectura y el almacenamiento de los datos se hará a través de una serie de funciones definidas en la página de conexión PHP (apartado 5.4).

5.3. Guardar particellas en una base de datos. (guardarXML_2.html)

Esta primera página web sirve para que cada usuario guarde sus particellas en formato MusicXML, correspondientes a un instrumento concreto de una obra musical. El objetivo es que cada miembro de un grupo, ya sea de estudiantes o docentes, pueda guardar su parte instrumental correspondiente para poder juntarla después con las partes creadas por sus compañeros o compañeras de grupo en la página del apartado 5.5.

MusicXML file 1: Seleccionar archivo TenorJoyful.xml

Guardar BD: Guardar

Alumno	Grupo	Nombre archivo	Fecha última edición	Versión
2	3	TenorJoyful.xml	2018-05-18 16:40:10	3

FIGURA 10: Página para guardar particellas.

El usuario al entrar en la web comprobará si tiene alguna partitura almacenada en la base de datos. En caso de tener una partitura almacenada se mostrará una tabla con la información correspondiente a la partitura cargada anteriormente. En caso contrario no se mostrará nada.
El usuario podrá seleccionar un archivo MusicXML de su ordenador y guardarlo. Además, podrá almacenar (actualizar) la partitura creada tantas veces como sea necesario. Al pulsar el botón de guardar se actualizará la información de la tabla.

5.3.1. Base de datos

Para almacenar las partituras se crea una base de datos sencilla en la que cada miembro de un grupo pueda guardar su particella e incluso pueda guardar nuevas versiones de la misma particella sin perder las versiones anteriores. Esta base de datos está formada por tres tablas denominadas partes_xml, historico_xml y grupos.

a) grupos:

Cada usuario tendrá un número id y a su vez este id estará relacionado a un número de identificación de grupo. El número id del alumnado coincidirá con el utilizado por Moodle. Es decir, se asocia el número id de Moodle a un número identificador de grupo. Los usuarios se dividen en grupos y cada grupo es responsable de una partitura completa.

	grp_idalumno	grp_idgrupo
ditar Copiar Borrar	2	3
ditar Copiar Borrar	5	3
ditar Copiar Borrar	7	1

Tabla 4: Base de datos grupos

b) partes_xml: se utiliza para almacenar las particellas.

Partes_xml está formada por un campo de identificación del número de la partitura (prt_id), un número de identificación del alumno/a (prt_alumno), un identificador del grupo al que pertenece el usuario (prt_grupo), prt_xml que contiene el archivo subido por el usuario, el nombre del archivo en prt_nombre_xml, la fecha de subida del archivo prt_fecha y un número de la versión de la partitura (prt_version).

prt_id	prt_alumno	prt_grupo	prt_xml	prt_nombre_xml	prt_fecha	prt_instrumento	prt_version
21	2	3	[BLOB - 167.2 KB]	ARANAtecladoa.xml	2018-06-06 10:13:23	teclado a	10
22	202	2	[BLOB - 196.7 KB]	Izeiaren_acom.xml	2018-05-17 12:14:57	piano	6
32	5	3	[BLOB - 96.2 KB]	ARANAtecladob.xml	2018-06-06 10:13:40	teclado b	5
33	104	1	[BLOB - 225.2 KB]	ABBAtrompa.xml	2018-06-06 08:46:36	trompa	12
34	101	1	[BLOB - 225 KB]	ABBA_trombon.xml	2018-06-06 08:28:35	trombón	11
36	102	1	[BLOB - 178 KB]	ABBA_trompeta.xml	2018-06-06 08:33:54	trompeta 2	8
37	103	1	[BLOB - 189.6 KB]	ABBA_trompeta1.xml	2018-06-06 08:23:10	trompeta 1	6
38	201	2	[BLOB - 252.8 KB]	izeia_melodia.xml	2018-05-23 08:06:58	instrumento viento	3

Tabla 5: Base de datos de la última versión de las particellas.

En partes_xml serán almacenadas las particellas subidas por cada usuario con la información sobre el grupo y el id del usuario.

c) historico_xml.

hst_id	hst_alumno	hst_grupo	hst_xml	hst_nombre_xml	hst_fecha	hst_instrumento	hst_version
47	54	5	[BLOB - 112.1 KB]	O SON DO ARpartetxistu.xml	2018-06-01 09:42:07		7
48	56	5	[BLOB - 240.5 KB]	O SON DO AR.xml	2018-05-31 11:07:16		1
49	56	5	[BLOB - 201.3 KB]	O SON DO ARpartebateria.xml	2018-06-01 09:42:46		2
50	54	5	[BLOB - 112.1 KB]	O SON DO ARpartetxistu.xml	2018-06-01 09:42:18		8

Tabla 6: Base de datos de versiones anteriores de particellas.

En esta tabla se guardarán automáticamente las versiones anteriores de las particellas subidas por los usuarios. Sigue la misma estructura de datos que la tabla anterior.

5.3.2. Acceso a la página web para guardar particellas la base de datos.

Para acceder a esta base de datos se ha creado una página web a la cual se accederá a través de Moodle. Para ello, se crea una actividad en Moodle incluyendo una URL que corresponderá con la página web dedicada a guardar las particellas en la base de datos. A esta URL se le ha añadido el número id del usuario del estudiante o profesor que acceda a la página.

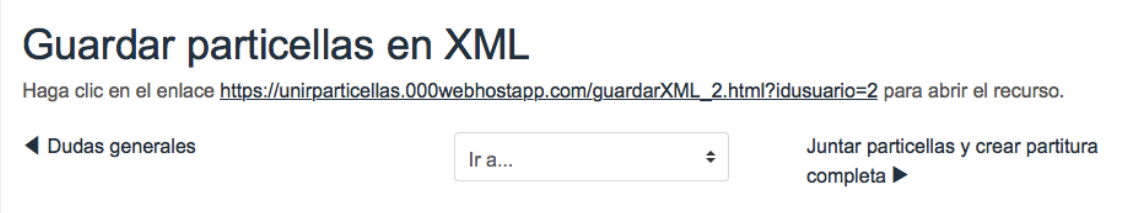


FIGURA 11: URL para guardar particellas desde Moodle.

De esta forma, el usuario será identificado automáticamente al acceder al link. Al entrar en la página se leerá el número id del usuario y se relacionará con el grupo al que pertenece a través de la tabla 'grupos' de la base de datos. Siguiendo el ejemplo de la figura, el usuario con id=2 será del grupo 3 (ver tabla 3).

Para realizar esta comprobación, al cargarse la página de guardar particellas y cuando JQuery detecta que el documento está listo se realiza una llamada a la función obtenerIdUsuarioURL() y desde aquí otra llamada a obtenerParametroURL() mediante la cual se devolverá el valor del id del usuario de la URL. Esta última función realiza una búsqueda de la posición de 'idusuario' en la URL y almacena el dato en *results*.

```
function obtenerParametroURL(name) {
    name = name.replace(/[\[]/, "\\[").replace(/[\]]/, "\\]");
    var regex = new RegExp("[\\?&]" + name + "=([^&#]*)"),
    results = regex.exec(location.search);
    return       results       ===       null       ?       ""       :
decodeURIComponent(results[1].replace(/\+/g, " "));
}
function obtenerIdUsuarioURL() {
      id_usuario = obtenerParametroURL('idusuario');
}
```

Una vez en la página, el usuario podrá subir un archivo MusicXML a la base de datos y será incluido con su número de usuario y grupo. Se especifica la necesidad de que sean archivos MusicXML y una vez cargados se realiza una comprobación del tipo de archivo cargado. En caso de que el usuario cargue un archivo distinto aparecerá un mensaje de error. Para completar la acción el usuario deberá pulsar el botón guardar mediante la cual se llamará a la función JavaScript guardar():

```
function guardar() {
      if (archivosXML.length > 0) {
```

```
            if (archivosXML[0].prt_id != null) {
                  actualizar();
            } else {
                  insertar();
            }
            $( "#guardarBD" ).prop( "disabled", true );
      } else {
            alert('Alumno no encontrado');
      }}
```

Esta función comprueba que se haya identificado al usuario comprobando que archivosXML haya sido definida. En caso contrario se mostrará un mensaje de error: 'Alumno no encontrado'. A continuación, se comprobará si previamente el mismo usuario ha cargado una particella. En caso de ser la primera particella se llamará a la función insertar() y en caso contrario a actualizar().
Ambas funciones introducen en la variable *file* el archivo subido por el usuario. Después, se comprueba que el archivo haya sido cargado y en caso contrario se muestra un mensaje de error: 'No hay parte para guardar'. Cuando el archivo ha sido cargado se llamará a la función insertarParte() en el caso de la función insertar() y a la función actualizarParte() en el caso de actualizar(). Ambas similares, pero únicamente se muestra la primera.

```
function insertar() {
      var file;
    file = $('#fileinput').prop ('files')[0];
      if (!file || alumno == '' || grupo == ''){
     alert('No hay parte para guardar');
     } else {
            var r = new FileReader();
      r.onload = function(e) {
            var contents = e.target.result;
            var                        contenido                        =
contents.replace(/(\r\n\t|\n|\r\t)/gm,"").replace(/\"/g,'\\"');
            insertarParte(alumno, grupo, contenido, file.name);}
      r.readAsText(file);
       }}

function insertarParte(alumno, grupo, parte_xml, nombre_xml) {
    $.ajax({
        url: getUrl("partes"),
```

```
        type: 'post',
        data: "{\"prt_alumno\": \"" + alumno + "\",\"prt_grupo\": \""
+ grupo + "\",\"prt_xml\": \"" + parte_xml + "\",\"prt_nombre_xml\":
\"" + nombre_xml + "\"}",
        dataType: "json",
        success: function (r) {
            alert(r.mensaje);
                  ComprobarAlumno();
                  $( "#guardarBD" ).prop( "disabled", false );
        },
        error: function (r) {
            alert(r.mensaje);
        }
    });
}
```

La función insertarParte() llama a la función getUrl mediante la cual se enviarán los datos incluidos en *data* a una función definida en el archivo de conexión con la base de datos (conexion.php) que almacenará dichos datos en la base de datos. En concreto en la tabla partes_xml.

```
var BASE_URL = ‘conexion.php/';
function getUrl(url) {
    return BASE_URL.concat(url);}
```

Por otro lado, insertarParte() llama a la función ComprobarAlumno() mostrada a continuación. Esta función primero comprueba que se haya definido id_usuario a través de la información obtenida mediante la URL. A continuación, llamará a la función definida en conexion.php para comprobar que el id del usuario se corresponde con algún id definido en la tabla ‘grupos’ de la base de datos. Además, relacionará este id con el número id de la tabla de partes_xml para obtener los datos correspondientes a este usuario. En caso de no haber registro anterior no se visualizará la tabla de los datos de usuario por pantalla. Y en caso de haber un registro anterior se mostrará en pantalla una tabla con los siguientes datos: el número id del usuario, el grupo, el nombre del archivo cargado, su fecha y versión.

```
function ComprobarAlumno() {
      if (id_usuario != '') {
            $.ajax({
                  url: getUrl("alumnos/") + id_usuario,
```

```
                type: 'get',
                async: false,
                dataType: "json",
                success: function (data) {
                      if (data.length > 0) {
                            $('#tb_partes').html("");
                            archivosXML = [];
                            alumno = null;
                            grupo = null;
                            for (var i = 0; i < data.length; i++) {
                                  archivosXML[i] = data[i];
                                  row = '<tr>';
                      row += '<td>' + data[i].grp_idalumno + '</td>';
                      row += '<td>' + data[i].grp_idgrupo + '</td>';
                row += '<td>' + data[i].prt_nombre_xml + '</td>';
                      row += '<td>' + data[i].prt_fecha + '</td>';
                      row += '<td>' + data[i].prt_version + '</td>';
                            row += '</tr>';
                                  $('#tb_partes').append(row);
                            }
                            alumno = data[0].grp_idalumno;
                            grupo = data[0].grp_idgrupo;
document.getElementById('table_partes').style.display= 'inline-block';
      document.getElementById('noExiste').style.display= 'none';
      $( "#guardarBD" ).prop( "disabled", false );
      }else {
      document.getElementById('table_partes').style.display= 'none';
   document.getElementById('noExiste').style.display= 'inline-block';
                            alert("No existe registro");
                      }},
                error: function (r) {
                      alert("error: comprobar");
                }});
      } else {
            alert('ERROR: No se ha podido identificar al alumno');
      }}
```

El proceso de guardado de particella podrá repetirse las veces que se quiera y las versiones anteriores o los archivos subidos con anterioridad pasarán a la base de datos denominada historico_xml. Sin embargo, al actualizar las particellas se sobrescribirán los datos correspondientes a cada usuario.

5.4. Conexión con la base de datos. (conexion.php)

Para realizar la conexión con la base de datos se utilizará un archivo PHP que será llamado desde la página para guardar particellas y por las páginas web que se explicarán a continuación para crear partituras completas y recuperar versiones anteriores (apartados 5.5 y 5.6).

En este archivo PHP primero se inicia sesión (mediante @session_start();) para almacenar información del usuario que se mantendrá disponible hasta cerrarla. Esta página utilizará Slim, un framework de PHP. Slim se utiliza para enviar una solicitud http y devolver una respuesta. Al construir una aplicación con Slim, se trabaja directamente con los objetos de solicitud y respuesta. En este caso Slim será utilizado para trabajar con el formato de intercambio de datos JSON. JSON permite el intercambio de datos entre dos sistemas desarrollados en diferentes lenguajes de programación. Se utiliza para comunicar el servidor en PHP con objetos JavaScript.

A continuación, se establece la conexión con el servidor y para ello, primero se definen el nombre del servidor, de la base de datos, del usuario de la base de datos y la contraseña. Se utiliza el siguiente código:

```
define('BD_SERVIDOR', 'localhost');
define('BD_NOMBRE', 'id5745243_partituras');
define('BD_USUARIO', 'id5745243_maria');
define('BD_PASSWORD', '-------');

$db = new PDO ('mysql:host=' . BD_SERVIDOR . ';dbname=' . BD_NOMBRE .
';charset=utf8', BD_USUARIO, BD_PASSWORD);
```

Después, se definen una serie de funciones para enviar a la base de datos el archivo MusicXML que el usuario haya guardado junto con la información del propio usuario (id, grupo, versión, etc.) o para leer y mostrar los datos recogidos en la base de datos. Para ello, se utilizarán los métodos GET y POST. Se analizarán dos ejemplos concretos: un ejemplo para guardar datos y otro para leerlos.

El primer ejemplo es una función para guardar el archivo MusicXML que subirá el usuario. Para ello, primero se accede a los datos recibidos del formulario utilizado para cargar el archivo (ver siguiente código). A continuación, comienza el envío de este archivo a la base de datos. Mediante bindParam() se vincula un parámetro o un valor a cada nombre de variable. Estas variables forman las columnas de las tablas de la base de datos, es decir, cada una de estas variables tiene un nombre y tipo de datos. Las variables serán vinculadas en el momento que se llame a execute(). Por último, se

comprueba que no haya ocurrido ningún error, y se mostrará un mensaje en pantalla sobre el estado de la transacción.

```
$app->post('/partes', function () use ($db, $app) {
    $body = $app->request()->getBody();
    $parte = json_decode($body);
    try {
        $db->beginTransaction();
        $consulta = $db->prepare("INSERT INTO partes_xml (prt_alumno,
prt_grupo, prt_xml, prt_nombre_xml) values (?,?,?,?)");
        $consulta->bindParam(1, $parte->prt_alumno);
        $consulta->bindParam(2, $parte->prt_grupo);
        $consulta->bindParam(3, $parte->prt_xml);
        $consulta->bindParam(4, $parte->prt_nombre_xml);
        $estado = $consulta->execute();
        $db->commit();
    } catch (Exception $ex) {
        $db->rollBack();
        echo $ex->getMessage();
    }
    if ($estado) {
        echo json_encode(array(
            'estado' => true,
            'mensaje' => 'Parte insertada correctamente.'
        ));
    } else {
        echo json_encode(array(
            'estado' => false,
            'mensaje' => 'ERROR: No se ha podido insertar la parte.'
        ));}});
```

Para que el usuario pueda guardar una nueva versión de la partitura se utiliza una función similar en la que se envían los nuevos datos a la base de datos sin modificar el id de grupo y alumno. Y la versión anterior se almacenará en historico_xml.
A continuación, se analiza una función que se utilizará para escoger los datos asociados a la base de datos en la que se guardan los archivos MusicXML. Esta función se encarga de capturar los datos correspondientes a un grupo de la base de datos partes_xml. Es decir, primero comprueba el grupo que se ha escogido o con el que se va a trabajar y a continuación, devuelve en $*resultados* los datos correspondientes a ese grupo.

```
$app->get('/partes/:grupo', function ($grupo) use ($db) {
    $sql = "select prt_id, prt_alumno, prt_grupo, prt_xml, prt_nombre_xml, prt_fecha, prt_version from partes_xml where prt_grupo = :grupo";
    $consulta = $db->prepare($sql);
    $consulta->execute(array(
            ':grupo' => $grupo
    ));
    $resultados = $consulta->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($resultados);
});
```

Por último, se incluye una función para ayudar a realizar la identificación del usuario en la página de guardar y recuperar partituras. Al acceder a estas páginas Moodle añade en la URL información sobre el número id del usuario. Este número id estará relacionado con un grupo en la tabla de 'grupos' de la base de datos. Por ello, al cargar la página se hace una consulta del id del usuario y se relaciona con el grupo y con el número id de la base de datos de 'partes_xml'.

```
$app->get('/alumnos/:id', function ($id) use ($db) {
    $sql = "select grp_idalumno, grp_idgrupo, prt_id, prt_alumno, prt_grupo, prt_xml, prt_nombre_xml, prt_fecha, prt_version from grupos left join partes_xml on grp_idalumno = prt_alumno where grp_idalumno = :id";
    $consulta = $db->prepare($sql);
    $consulta->execute(array(
            ':id' => $id
    ));
    $resultados = $consulta->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($resultados);
```

5.5. Página web para unir particellas (unirparticellas.html)

Esta es la página web más importante del proyecto. Se trata de una página que convierte una serie de particellas a notación ABC, las muestra en pantalla y las une para crear una partitura general al final. Esta partitura final podrá visualizarse, reproducirse y descargase.

Uno de los requisitos del proyecto es la utilización de programas de Licencia GNU y gratuita, por tanto, se decide trabajar con archivos de formato abierto (MusicXML) y

con librerías JavaScript de Licencia GNU para ayudar en la visualización, reproducción y descarga de partituras.

Esta página muestra las partituras cargadas por distintos grupos. En la siguiente imagen se observan las particellas almacenadas por un grupo (el grupo 1) formado por cuatro miembros.

FIGURA 12: Parte superior de la página web unirparticellas.html.

Después, se podrán seleccionar una serie de particellas para mostrar su correspondiente partitura y crear una partitura general a partir de las mismas pulsando 'Unir particellas'. Por último, se creará una partitura completa a partir de las particellas seleccionadas como la que se muestra en la siguiente imagen.

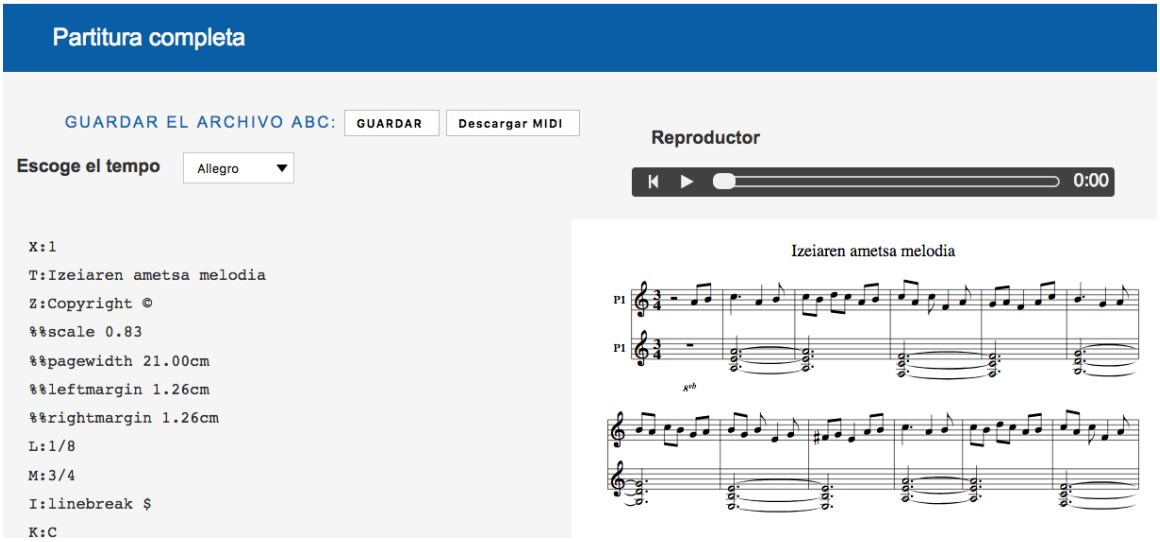


FIGURA 13: Visualización de la partitura general en unirparticellas.html.

En los siguientes apartados se describe el código o la programación de la página. Principalmente las funciones JavaScript.

5.5.1. Cargar contenidos

El primer paso en esta página web será cargar los contenidos de la base de datos. Para ello, se incluye un formulario en HTML en el que el usuario podrá escoger el número de su grupo y así cargar en una tabla los archivos correspondientes a este grupo. Todos los usuarios podrán visualizar los arreglos realizados por otros grupos ya que en esta web no será necesaria la autenticación del usuario.

FIGURA 14: Vista de la página para unir particellas.

En la figura anterior el usuario ha seleccionado el grupo 2 y a continuación, se han cargado los archivos cargados en la base de datos por usuarios del grupo 2. En esta tabla se cargarán los archivos correspondientes a la base de datos partes_xml.

Al pulsar el botón seleccionar grupo se llamará a la función JavaScript de leerPartesGrupo(). Esta función llamará a la función de conexion.php que se encarga de leer los datos de la tabla partes_xml y para ello envía el número del grupo que ha sido seleccionado por el usuario en el formulario. Por tanto, esta función devolverá un objeto JSON con los datos correspondientes al grupo seleccionado. A continuación, se mostrarán en una tabla por pantalla dichos datos y en caso de que no haya ningún registro correspondiente a ese grupo se mostrará un mensaje indicando que no existen particellas almacenadas para ese grupo.

```
function leerPartesGrupo() {
    $.ajax({
     url: getUrl("partes/") + $("#sel_grupos option:selected").text(),
        type: 'get',
       async: false,
        dataType: "json",
```

```
        success: function (data) {
            if (data.length > 0) {
                $('#tb_partes').html("");
            archivosXML = [];
               for (var i = 0; i < data.length; i++) {
            archivosXML[i] = data[i];
                    row = '<tr>';
              row += '<td> <input type="checkbox" /> </td>'
                    row += '<td>' + data[i].prt_nombre_xml + '</td>';
                    row += '<td>' + data[i].prt_fecha + '</td>';
                    row += '<td>' + data[i].prt_version + '</td>';
              <!-- row += '<td style="width:0%; display: none">' +
data[i] + '</td>'; -->
                    row += '</tr>';
                    $('#tb_partes').append(row);
                }
            } else {
                alert("No existen particellas para este grupo");
            } },
        error: function (r) {
            alert("error: leerparte");
        }});}
```

Una vez mostrados los archivos de un grupo en pantalla, el usuario podrá seleccionar los que quiera y mostrarlos por pantalla. Estos archivos podrán visualizarse a través del botón ‘cargar’. Al pulsar este botón se llamará a la función cargarPartes() la cual comprueba los archivos seleccionados de la tabla mostrada en pantalla para llamar a la función mostrarParte().

```
function cargarPartes() {
      'use strict';
 function(){
      var num = 0;
$('#tb_partes tr').each(function(){
var isChecked =$(this).find('input[type="checkbox"]').prop("checked");
      if (isChecked) {
            num +=1;
            var ix = $(this).index();
            mostrarParte(num, ix);
      };}}
```

Al llamar a la función mostrarParte() se visualizarán por pantalla las cajas HTML donde se carga la notación ABC y la transcripción de la partitura de cada archivo seleccionado. Se ha establecido un máximo de diez instrumentos para cargar sus contenidos y unir particellas. Este máximo se ha escogido teniendo en cuenta el número de instrumentos de las agrupaciones instrumentales de escuelas de música, bandas o fanfarres. Las agrupaciones orquestales superarían este máximo, no obstante, estas agrupaciones trabajan repertorio original y no arreglos realizados por las escuelas o sus miembros. Esta aplicación no está dirigida a este formato.

```
function mostrarParte(num, ix) {
drop_files = null;
document.getElementById(titulos[num-2]).style.display= 'inline-block';
document.getElementById(reses[num-2]).style.display= 'inline-block';
        instrumentos = num;
        xmlFileName2 = archivosXML[ix].prt_nombre_xml;
        loadXml2(archivosXML[ix].prt_xml);
        <!-- leesFile2 (); }); -->
      }
```

Por otro lado, la función anterior llamará a otra función, loadXML2, para cargar los archivos en variables JavaScript y desde la última función se llamará a vertaal() encargada de convertir el archivo MusicXML a notación ABC.

En esta primera parte del programa, se añade una librería desarrollada en Javascript de Licencia GNU la cual realiza la conversión de archivos MusicXML a ABC. Se escogen los archivos MusicXML como archivos de entrada debido a que cualquier partitura creada con los editores de partituras más utilizados pueden exportar a este formato. No obstante, se realiza la conversión a notación musical ABC la cual utiliza un lenguaje de caracteres ASCII ya que permite realizar modificaciones de forma sencilla mediante JavaScript. El formato ABC es también abierto y recoge la información necesaria para poder visualizar y reproducir su contenido. Además, existen varios paquetes de software que permiten leer y procesar música en sistema ABC. El enlace a este archivo JavaScript se incluye en el apartado 10: xml2abc.

5.5.2. Visualizar archivos ABC

Cada uno de los archivos cargados, se transforman primero en ABC y a continuación, a través de la aplicación o JavaScript abc2svg.js se visualizará su contenido en pantalla. En concreto se va a visualizar la notación musical ABC del archivo en la parte izquierda (similar a un lenguaje de programación) y la partitura correspondiente en la

parte derecha una vez realizada la transformación a SVG. Esta transformación se realiza mediante la función dolayout() definida por el mismo desarrollador de abc2svjg.js.

Se crean diez cajas en HTML (igual al máximo de instrumentos) para poder visualizar este contenido. También se añade una caja al final para poder visualizar el contenido del archivo generado al unir las particellas o las partes de cada instrumento en un solo archivo.

Esta librería JavaScript incluye dos botones: uno de opciones de visualización de las partituras (Opciones) y otro de ayuda (Ayuda). Al pulsar el primero se abre un desplegable para modificar la maquetación de la partitura y el segundo muestra una ventana de ayuda sobre el programa. No obstante, se ha modificado el botón de ayuda para explicar el funcionamiento de esta nueva página web.

5.5.3. Crear la partitura completa a partir de las partes

El objetivo de este proyecto es permitir la creación de partituras de forma colaborativa, es decir, crear una partitura general a partir de distintas particellas creadas por distintos usuarios. De esta forma, los usuarios pueden comprobar su arreglo musical online. Las particellas serán partituras de una sola voz.

5.5.3.1. Notación ABC

Para poder crear un único archivo a partir de cada una de las partes se ha realizado un análisis de la estructura de los archivos MusicXML y ABC. Se ha analizado una partitura de dos voces en los formatos MusicXML y ABC. A partir de la misma partitura, se han creado dos particellas suprimiendo la información musical de una de las voces. Con la información de los tres archivos en cada formato, se ha analizado la estructura de las partituras. Esta estructura se detalla a continuación:

1. Información general sobre la partitura constituida por distintos campos que informan o regulan la partitura (Rodríguez, 2012):

X: Es el campo que dice que se trata de una melodía ABC. Después de los dos puntos debe aparecer un número o índice diferenciador de las melodías.

T: Título de la melodía.

C: Nombre del compositor.

O: Origen de la melodía.

Z: Copyright.
M: Medida o ritmo. El compás de la partitura.
L: Duración de la unidad relativa de tiempo. La corchea se denota con L: 1/8.
Q: Tempo o velocidad de ejecución.
K: Clave.
R: Instrumento (si este campo no se explicita, el instrumento por defecto es el piano).

Estos son los campos de descripción de la partitura más generales de la notación ABC. No obstante, el estándar ABC 2.0 presenta un conjunto de campos de información, conforme a una nueva sintaxis. Comienzan con los caracteres %%, seguidos con el nombre del campo y su contenido. Estos campos se definen en el encabezado del archivo. Entre otras opciones estos campos permiten definir aspectos de maquetación como la escala o márgenes de la partitura.

2. Información sobre el número de instrumentos de la partitura (voces). Después de la información general de la partitura se especifica cada una de las voces o instrumentos de la partitura.

Instrumento 1

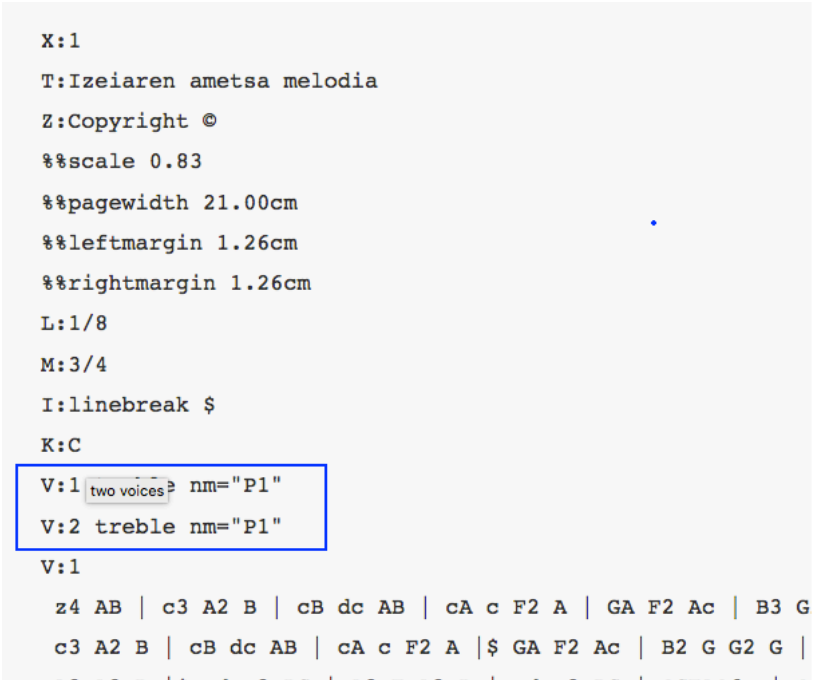
```
X:1
T:Izeiaren ametsa melodia
Z:Copyright ©
%%scale 0.83
%%pagewidth 21.00cm
%%leftmargin 1.26cm
%%rightmargin 1.26cm
L:1/8
M:3/4
I:linebreak $
K:C
V:1 treble nm="P1"
V:1
 z4 AB | c3 A2 B | cB dc AB | cA c F2
```

Instrumento 2

```
X:1
T:Izeiaren ametsa melodia
Z:Copyright ©
%%scale 0.83
%%pagewidth 21.00cm
%%leftmargin 1.26cm
%%rightmargin 1.26cm
L:1/8
M:3/4
I:linebreak $
K:C
V:1 treble nm="P1"
V:1
!8vb(! z6 | [A,,E,A,]6- | [A,,E,A,]6
```

FIGURA 15: Información general y voces de las particellas (notación ABC).

Estas partituras tienen una sola voz o instrumento. En ambas se indica que se trata de un instrumento o voz ‘treble’ un instrumento agudo o ‘tiple’ que será representado en clave de Sol. En partituras de instrumentos graves se indicará ‘bass’ en su lugar y se representarán en clave de Fa. En ocasiones esta información es acompañada por el nombre del instrumento en cuestión.

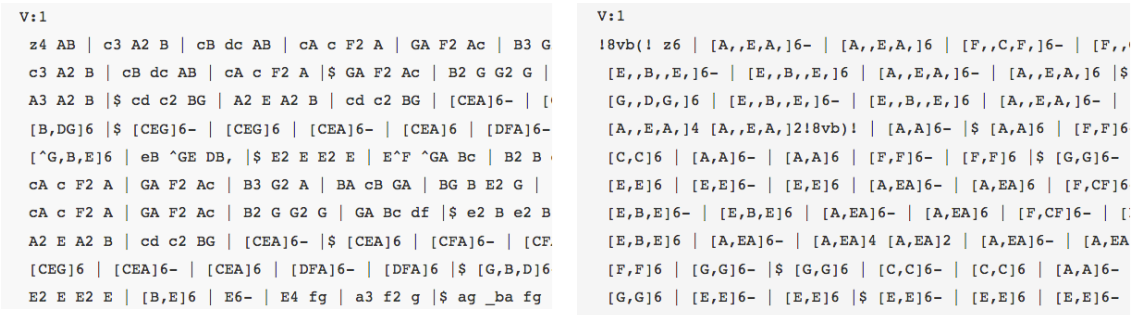


```
X:1
T:Izeiaren ametsa melodia
Z:Copyright ©
%%scale 0.83
%%pagewidth 21.00cm
%%leftmargin 1.26cm
%%rightmargin 1.26cm
L:1/8
M:3/4
I:linebreak $
K:C
V:1 two voices nm="P1"
V:2 treble nm="P1"
V:1
 z4 AB | c3 A2 B | cB dc AB | cA c F2 A | GA F2 Ac | B3 G
 c3 A2 B | cB dc AB | cA c F2 A |$ GA F2 Ac | B2 G G2 G |
```

FIGURA 16: Información general y sobre las voces de la partitura general (notación ABC).

El segundo ejemplo es de una partitura de dos voces (el resultado de unir las dos anteriores).

3. Información musical correspondiente a cada voz. En la tercera parte del archivo se encuentra información sobre la duración y altura de las notas correspondientes a cada voz.



```
V:1
 z4 AB | c3 A2 B | cB dc AB | cA c F2 A | GA F2 Ac | B3 G
 c3 A2 B | cB dc AB | cA c F2 A |$ GA F2 Ac | B2 G G2 G |
 A3 A2 B |$ cd c2 BG | A2 E A2 B | cd c2 BG | [CEA]6- | [
 [B,DG]6 |$ [CEG]6- | [CEG]6 | [CEA]6- | [CEA]6 | [DFA]6-
 [^G,B,E]6 | eB ^GE DB, |$ E2 E E2 E | E^F ^GA Bc | B2 B
 cA c F2 A | GA F2 Ac | B3 G2 A | BA cB GA | BG B E2 G |
 cA c F2 A | GA F2 Ac | B2 G G2 G | GA Bc df |$ e2 B e2 B
 A2 E A2 B | cd c2 BG | [CEA]6- |$ [CEA]6 | [CFA]6- | [CF
 [CEG]6 | [CEA]6- | [CEA]6 | [DFA]6- | [DFA]6 |$ [G,B,D]6
 E2 E E2 E | [B,E]6 | E6- | E4 fg | a3 f2 g |$ ag _ba fg
```

```
V:1
!8vb(! z6 | [A,,E,A,]6- | [A,,E,A,]6 | [F,,C,F,]6- | [F,,
 [E,,B,,E,]6- | [E,,B,,E,]6 | [A,,E,A,]6- | [A,,E,A,]6 |$
 [G,,D,G,]6 | [E,,B,,E,]6- | [E,,B,,E,]6 | [A,,E,A,]6- |
 [A,,E,A,]4 [A,,E,A,]2!8vb)! | [A,A]6- |$ [A,A]6 | [F,F]6
 [C,C]6 | [A,A]6- | [A,A]6 | [F,F]6- | [F,F]6 |$ [G,G]6-
 [E,E]6 | [E,E]6- | [E,E]6 | [A,EA]6- | [A,EA]6 | [F,CF]6
 [E,B,E]6- | [E,B,E]6 | [A,EA]6- | [A,EA]6 | [F,CF]6- | [
 [E,B,E]6 | [A,EA]6- | [A,EA]4 [A,EA]2 | [A,EA]6- | [A,EA
 [F,F]6 | [G,G]6- |$ [G,G]6 | [C,C]6- | [C,C]6 | [A,A]6-
 [G,G]6 | [E,E]6- | [E,E]6 |$ [E,E]6- | [E,E]6 | [E,E]6-
```

FIGURA 17: Información musical de las particellas (notación ABC).

La partitura completa formada a partir de las dos particellas anteriores incluiría la información musical de ambas. La diferencia en esta segunda sería que la información musical del segundo instrumento comenzaría con V: 2.

5.5.3.2. Juntar dos particellas.

Una vez cargadas las particellas podrán unirse en una partitura completa pulsando el botón denominado 'Unir particellas'. Al pulsarlo, se creará una sola partitura a partir de la información musical de cada uno de los instrumentos. En estos archivos de las voces o instrumentos, la primera parte correspondiente a la información general de la partitura se repite. No obstante, varía la información relacionada con cada voz: el campo 'V: '.

Por tanto, para poder unir las partes se va a tomar la información en ABC correspondiente a cada una de las particellas y se almacena en un string en JavaScript. A continuación, se toma la información general del primer archivo y se realizarán modificaciones en el mismo añadiendo la información musical correspondiente al resto de archivos. Se van a seguir los siguientes pasos:

1. Construir un string con la información general de la partitura a partir de la información de la primera particella.

2. Construir un string con la información correspondiente al instrumento o voz de cada particella. Cambiar el número de instrumento de los instrumentos en función del orden en el que hayan sido incluidos o cargados en la web:

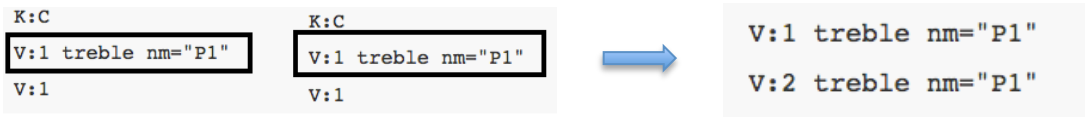


FIGURA 18: String con la información sobre el número y tipo de instrumentos.

3. Añadimos la información musical correspondiente a las siguientes particellas y volvemos a modificar el número de voz en función del número de instrumento que ocupa en la web. En la partitura del ejemplo anterior (figura 18), se añade la información musical de la segunda particella, definiéndola como la segunda voz, V:2.

En este proceso se utiliza una función propia JavaScript denominada myFunction() incluida en el Anexo I. Se trata de la función que crea un string con la información de la partitura completa. Esta función primero muestra todas las secciones del HTML que se incluirán al visualizar la partitura completa, como la propia partitura, reproductor y distintas opciones de reproducción y guardado. A continuación, se comprueba que

haya al menos dos partituras cargadas para poder unirlas. Después, se almacena la primera particella en un string y en otra el resto de partituras (*recabcText*). De esta última, se extraerá la información correspondiente al tipo de instrumento en un string (punto 2) y la información musical en otro. A continuación, se insertarán estos nuevos strings en el primero para crear la partitura completa. Por último, desde la función se llamará a la función dolayout() para convertir el archivo a SVG y a la función cargar() para la conversión a MIDI y asociarlo al reproductor.

5.5.3.3. Particella de piano.

La música para piano es por defecto música para dos voces, una para la mano derecha y otra para la izquierda. Se codifica en dos pentagramas uno debajo del otro, el superior para la mano derecha, en clave de Sol con sus modificadores o armadura y el inferior para la izquierda en clave de Fa que tiene la misma armadura que el superior. Puede, sin embargo, haber variaciones con notas accidentales.
En el programa se pide que el piano se incluya siempre en último lugar. Por ello, el programa realiza una comprobación automática para verificar si la última partitura añadida corresponde a una partitura de piano. Simplemente comprueba el número de voces de esta partitura. En caso de encontrar una partitura de piano, se añadirá la información correspondiente a dos nuevas voces o instrumentos, tanto después de la cabecera como en la parte de la información musical.

A continuación, se observa la información de los instrumentos de cada una de las particellas correspondientes a una flauta y a un piano de una canción (izquierda y derecha). La particella del piano incluye dos voces.

```
L:1/8
M:4/4
I:linebreak $
K:Bb
V:1 treble nm="Flauta" snm="Fl."
V:1
|: g2 ef g>b ag | a2 fg a>c' ba | b2 ga b
 b2 fg f>e de | d2 df d>c"^4 aldiz" df ::
```

```
V:1 treble nm="Piano" snm="Pno."
V:2 bass
V:1
|: z8 | z8 | z8 | z8 | z8 | z8 | z8 |$
```

FIGURA 19: Particellas de la canción Otsagi.

```
V:1 treble nm="Flauta" snm="Fl."
V:2 treble nm="Piano" snm="Pno."
V:3 bass
V:1
|: g2 ef g>b ag | a2 fg a>c' ba | b2 ga b>d' c'b
```

FIGURA 20: Partitura completa de Otsagi.

En la partitura completa se definen tres voces, la primera correspondiente a la flauta, y las dos últimas correspondientes a cada pentagrama del piano. La información musical también se ordena en tres voces. El resultado es una partitura con dos instrumentos, y tres voces:

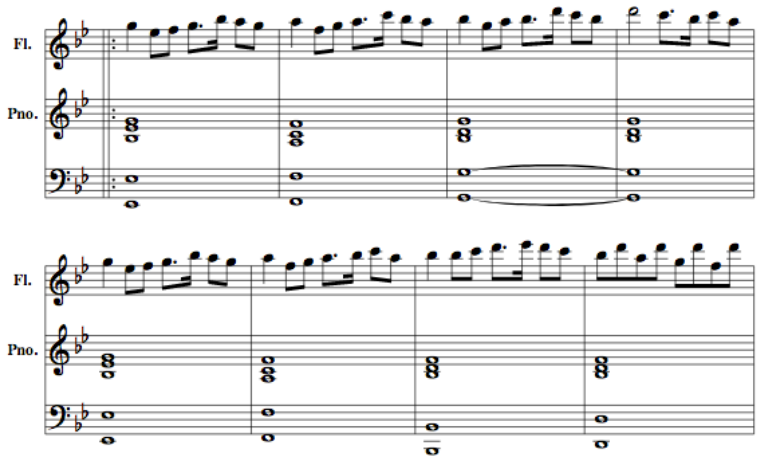


FIGURA 21: Visualización de la partitura completa de Otsagi.

Para visualizar la partitura completa, se realiza la transformación del nuevo string creado con la información en notación ABC de la partitura completa a SVG. Es decir, se vuelve a llamar a la función vertaal() definida en abc2svg.js.

5.5.4. Excepciones.

Tras realizar pruebas con distintos archivos o partituras se obtiene un problema al unir particellas cuando tienen un valor distinto en el campo L. Este campo define la duración de la unidad relativa de tiempo. Este problema ocurre cuando el ritmo general de una partitura no coincide con la de otro instrumento. La canción del ejemplo anterior tiene estas características.

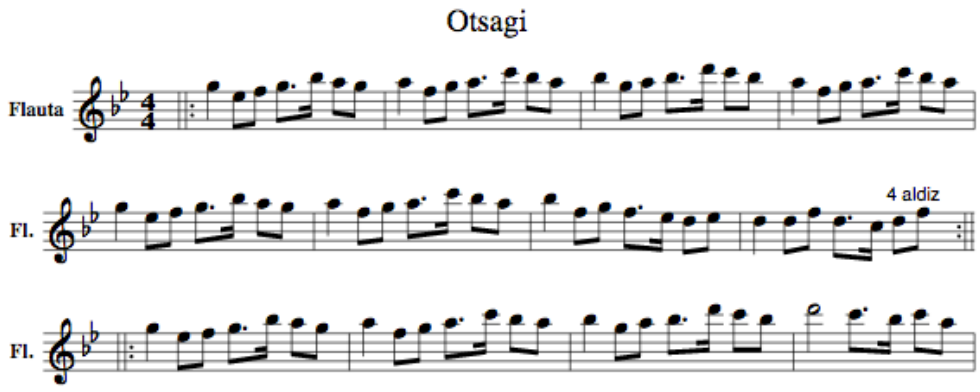


FIGURA 22: Particella de flauta de Otsagi.

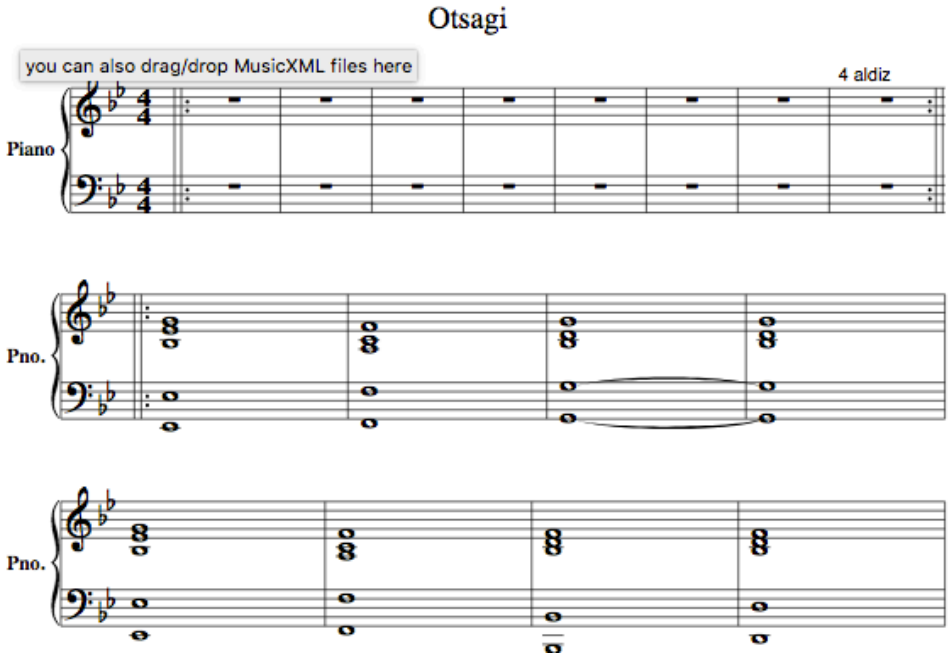


FIGURA 23: Particella de piano de Otsagi.

En la partitura de flauta dominan las figuras rítmicas como las negras, corcheas y semicorcheas, sin embargo, en la del piano el ritmo es más lento y predominan las redondas. Por tanto, a la hora de generarse el archivo ABC, el valor del campo L correspondiente a la duración relativa de tiempo es distinto. En el primer caso es de 1/8 y el piano tiene distinta L para cada mano (L:1/8 la derecha y L:1/4 la izquierda). Al unir dos partituras con distinto valor de L, los compases de una partitura no coinciden con la siguiente, tal y como se puede ver en la siguiente imagen. Concretamente, no coincide el pentagrama inferior del piano ya que es el que tiene el valor de L distinto. En cada compás de flauta se introducen dos compases de la segunda voz del piano.

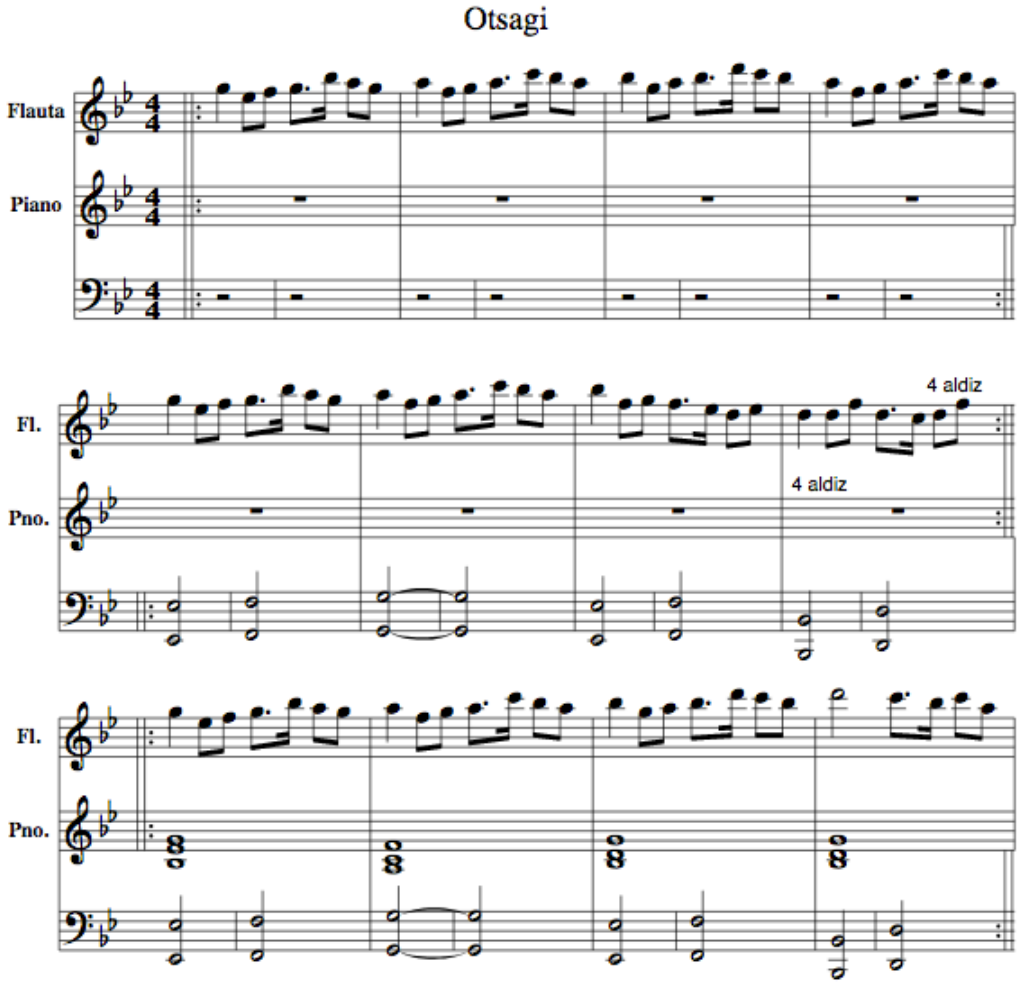


FIGURA 24: Partitura completa de Otsagi a partir de dos particellas con distinta L.

La primera opción para solucionar este problema fue añadir el valor de cada L a la información de las voces de la siguiente manera:

```
X:1
T:Otsagi
Z:Otsagi
%%scale 0.83
%%pagewidth 21.00cm
%%leftmargin 1.24cm
%%rightmargin 1.24cm
L:1/8
M:4/4
I:linebreak $
K:Bb
V:1 treble nm="Flauta" snm="Fl."
V:2 treble nm="Piano" snm="Pno."
L:1/8
V:3 bass
L:1/4
V:1
```

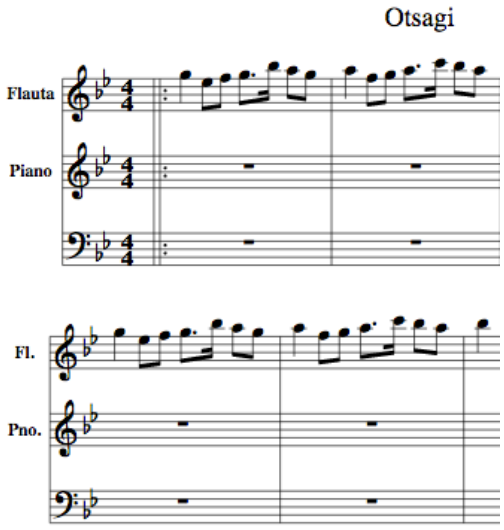


FIGURA 25: Cambios en la notación ABC y partitura.

De esta forma se soluciona la partitura creada tanto su visualización en pantalla como el MIDI disponible para la descarga. Sin embargo, el reproductor no comprende esta información y sigue reproduciendo la partitura con el mismo error.

Por tanto, la solución a este problema es forzar a cargar las partituras con un mismo valor de L. Se escoge un valor pequeño, de L:8 (corchea), para todas las partituras. En el desplegable del botón opciones se puede fijar el valor de L para que todas las partituras se carguen con una duración de tiempo relativa determinada. Sin embargo, para que no ocurra el problema anterior se establece un valor de 8 por defecto. De esta forma todas las conversiones de las particellas a notación ABC se realizan con un valor de L igual a 8. Es decir, en caso de que el usuario deje en blanco el formulario correspondiente a la duración L, el programa tomará un valor de 8.
En la función de opciones se lee el valor de L que escoge el usuario, y en caso de no rellenarlo se completa con un 8. Para ello, también se fuerza una llamada a la función de lectura de opciones antes de cargar el contenido.

```
xml_options.d = parseInt ($('#den').val () || 0);
      if(xml_options.d==0){xml_options.d = 8;}
```

5.5.5. Limpiar o borrar los archivos cargados.

Esta página web ha sido diseñada para poder seleccionar las partituras que el usuario quiera juntar y, por otro lado, permite la visualización y reproducción de partituras de otros grupos. Por ello, para que el usuario pueda realizar las operaciones que desee, se incluye la opción de ‘limpiar’. Este botón permite borrar el contenido de los strings correspondientes a las operaciones de juntar particellas anteriores y oculta las partituras cargadas anteriormente. De esta forma el usuario podrá realizar varias acciones de juntar particellas y repetir procesos.

5.5.6. Reproducir online la partitura completa

La partitura completa construida por la web es una guía o medio para interpretar la pieza de música en cuestión. Los músicos saben interpretarla y a simple vista comprobar si el arreglo musical es correcto. Viendo la partitura puede comprobarse si la estructura de cada particella coincide con el resto de las voces. No obstante, para comprobar cada sonido o detalle de la partitura se hace necesaria la reproducción de la misma. Por ello, se ha incluido un reproductor online de la partitura completa.

Para la reproducción se ha vuelto a incluir una librería, en concreto un JavaScript que contiene una función para convertir la notación ABC a MIDI (abcjs_midi_4.0.1-min.js) y así posibilitar la reproducción de la partitura. Una vez realizada la conversión a MIDI se podrá reproducir la partitura a partir del navegador.
Desde la función cargar() de la cual se muestra una parte a continuación, se comprueba si ha sido modificado el MIDI o la partitura final mediante las opciones que se describen en los apartados siguientes. Los últimos dos métodos crean un objeto ABC y otro MIDI, y este último será utilizado por el reproductor.

```
if(definitivomidi!=-1){
        tune=definitivo;
}else{
        tune= definitivomidi;
}
    $('.source').text(tune);
window.ABCJS.renderAbc('engraving', tune, {responsive: "resize" });
window.ABCJS.renderMidi("main-midi", tune, { generateInline: true });
```

Por tanto, se incluye un reproductor en el archivo HTML que se mostrará una vez unidas las partituras instrumentales o una vez creada la partitura completa.
La licencia de la librería JavaScript incluida permite su uso de forma gratuita, a cualquier persona que obtenga una copia de este software para tratar el software sin restricciones, incluidos, entre otros, los derechos de uso, copia, modificación y fusión. También permite publicar, distribuir y vender copias del software, siempre incluyendo el aviso de copyright anterior. Por tanto, en este JavaScript se incluye información sobre el copyright como comentario. En el siguiente enlace puede encontrarse información sobre la licencia de esta librería: https://github.com/paulrosen/abcjs/blob/master/LICENSE.md. También se incluye un enlace a la propia librería en el apartado 10.

No obstante, abc2midi tiene algunas limitaciones como por ejemplo que no reconoce signos D.C. (Da Capo) o similares. También existen limitaciones a la hora de escribir letras de las canciones, ya que no reconoce todos los caracteres y da problemas cuando hay dos letras en la misma sección. Si hay signos de repetición mal emparejados en el ABC el programa intenta corregirlos pero no cuando haya varias voces. Si se especifica un cambio de tempo en una de las voces pero no en otra el reproductor reproducirá primero una voz y después la siguiente.

5.5.6.1. Tempo de reproducción.

La interpretación de una pieza musical puede realizarse a distinta velocidad o tempo. Estas variaciones son fundamentales en la música y todos los editores de partituras musicales permiten modificar el tempo de las mismas para escuchar cada nota de la partitura o reproducirlo a tempo real. El código ABC controla la velocidad de reproducción mediante el campo Q:. Por ejemplo, al definir Q: 1/4 = 180, se interpreta la figura musical correspondiente a '1/4' a 180, es decir, se interpretarán 180 figuras por minuto (la figura rítmica dependerá de la L definida). De acuerdo con el estándar ABC 2.0, también se puede indicar el tempo usando las siguientes indicaciones: "lento", "allegro", "vivace", etc.

El problema es que muchas veces, en las partituras creadas por los editores de partituras no se define la velocidad de reproducción o tempo ya que podemos modificarlo desde el panel de reproducción del software. En caso de no definirlo, al transformar el archivo MusicXML a notación ABC, el campo Q: estará ausente. En este caso, los reproductores o los programas de conversión a MIDI suponen un tempo de 100 para la longitud de la nota L:. Este tempo no siempre es adecuado para todas las partituras, por tanto, se da la posibilidad de incluir este campo a la partitura final. Se incluye un formulario en HTML para escoger entre distintas posibilidades (allegro, presto, largo, moderato, etc.) y se define la Q en función de la siguiente tabla:

Larghissimo	1/4=40	Moderato	1/4=104
Adagissimo	1/4=44	Allegretto	1/4=112
Lentissimo	1/4=48	Allegro	1/4=120
Largo	1/4=56	Vivace	1/4=168
Adagio	1/4=59	Vivo	1/4=180
Lento	1/4=62	Presto	1/4=192
Larghetto	1/4=66	Allegrissimo	1/4=208
Adagietto	1/4=76	Vivacissimo	1/4=220
Andante	1/4=88	Prestissimo	1/4=240
Andantino	1/4=96		

FIGURA 26: Tempo (Shlien, 2018).

Por tanto, al escoger un tempo concreto a través del desplegable, se incluye el campo Q: en la información ABC y se visualizará este campo en pantalla. A continuación, se realizará de nuevo la conversión a SVG y se visualizará en la partitura el tempo definido mediante el desplegable. Posteriormente, se volverá a realizar la conversión a MIDI y el tempo de reproducción variará en función de este cambio. Se podrán realizar tantas transformaciones como sea necesario.

Al escoger un tempo distinto a través del formulario HTML correspondiente se llamará a la función establecervelocidad() y se le pasa la opción escogida en el formulario. A continuación, se muestra el código correspondiente a la elección de la primera opción,

es decir, la elección de allegro. Esta función define el tempo correspondiente al allegro ('Q:¼=120') y llama a la función de transporte la cual modificará el tono (ver apartado siguiente) y velocidad de la partitura general al mismo tiempo. La función transporte() vuelve a llamar a las funciones dolayout() y cargar() para mostrar por pantalla la nueva partitura y modificar el archivo MIDI del reproductor con el cambio de tempo.

```
function establecervelocidad(velocidad){
var
velo=(document.getElementById(velocidad)[document.getElementById(veloc
idad).options.selectedIndex].value);
if(velo==1){
      tempo="Q:1/4=120 ";
      transporte(tono11);
            }…
```

La variable tono11 será definida con un valor igual a 12 correspondiente al tono original. En caso de que el usuario modifique el tono a través de otro formulario, el nuevo tono se almacenará en esta misma variable.

5.5.6.2. Transposición.

Otra funcionalidad añadida a la reproducción de la partitura completa creada es la posibilidad de transportarla a cualquier tonalidad entre las 12 posibilidades existentes. Esta opción transportará la partitura hasta un máximo de 3 tonos hacia arriba y 2 tonos y medio hacia abajo. Una vez variado el tono podrá recuperarse el tono original. Esta opción añade en el código ABC un campo el cual indica que el MIDI generado debe transportarse la distancia escogida por el usuario. Por tanto, esta modificación afectará a la reproducción de la partitura y al MIDI disponible para la descarga.

El cambio incluirá en el código ABC una línea con la información de transposición. Ejemplo: *%%MIDI transpose 1*. La indicación anterior subirá medio tono la reproducción de la partitura original. Una vez modificada la tonalidad se podrá volver a modificar la velocidad de reproducción de la pieza.

A continuación, se muestra parte de la función transpose() que será llamada al cambiar de tono a través del formulario. Esta función modificará la variable tono11 en función del tono escogido por el usuario y llamará a la función transporte().

```
function transpose(nuevotono){
      tono11=(document.getElementById(nuevotono)[document.getElementBy
Id(nuevotono).options.selectedIndex].value);
```

```
    transporte(tono11);
}
```

La función transporte tiene como variable de entrada la tonalidad escogida por el usuario en caso de que haya sido modificada y en caso de no haber sido modificada tendrá un valor igual a 12. La función transporte() busca la posición del campo L: en el string con la información ABC de la partitura general e introducirá a continuación la nueva información sobre el tempo o la tonalidad escogidos. Por último, vuelve a llamar a las funciones dolayout() y cargar() para mostrar por pantalla la nueva partitura y modificar el archivo MIDI con el cambio de tono y de tempo. A continuación, se muestra parte de la función transporte:

```
function transporte(tono1){
if(tono1==12){
    var posT= definitivo.indexOf("L:")+5;
   var princ= definitivo.substring(0,posT);
    var def = definitivo.substring(posT+1,definitivo.length-1);
            cambiotono="%%MIDI transpose 0";
            var todo='';
      if (tempo.length<1){
      todo=cambiotono;
      }else{
      todo=tempo+ "\n"+ "%%MIDI transpose 0";
      }
    definitivomidi= princ+ "\n" + todo + "\n" +def;
      var div2 = document.getElementById("abcfinal");
        div2.textContent = definitivomidi;
        dolayout3 (definitivomidi);
        cargar(definitivomidi);
        return definitivomidi;
            }
```

5.5.6.2.1. Instrumentos transpositores

Muchos de los instrumentos más comunes en las escuelas de música y conservatorios son transpositores. Estos instrumentos tocan una nota escrita y suena otra distinta, es decir, la altura de la nota que suena no corresponde con la altura de la nota escrita. Por tanto, para que estos instrumentos toquen una misma melodía junto a otros deberán tocar en distinta tonalidad.

No obstante, los editores de partituras realizan el cambio a la tonalidad en la que tienen que tocar estos instrumentos de forma automática únicamente indicando el

instrumento para el que está compuesta la partitura. Los editores podrán mostrar dicha partitura de dos maneras, en la tonalidad en la que sonará y en la tonalidad en la que deberá tocar el instrumento transpositor, es decir, existen dos opciones.

Si escogemos la opción de guardar la particella en la tonalidad en la que sonará no habrá problemas en la página de unir particellas y se unirá como un instrumento más. Pero, en caso de guardar la particella en la tonalidad en la que este instrumento tocará, es decir, mostrando la partitura con transposición en el editor de partituras, se deberá realizar una transposición de la misma para poder reproducirla junto a partituras de otros instrumentos.

La partitura general del siguiente ejemplo se compone por los siguientes instrumentos: piano, guitarra, saxofón, txistu y batería. El piano no es un instrumento transpositor pero el txistu y el saxofón sí. Estos instrumentos deberán tocar en otras tonalidades para que la tonalidad resultante coincida con la del piano y guitarra.

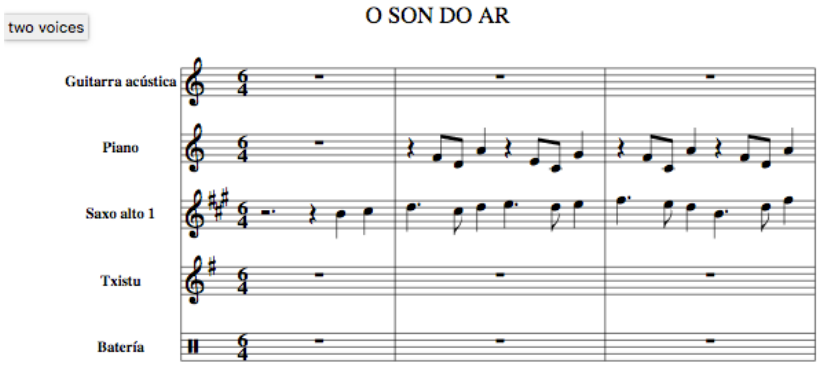


FIGURA 27: Partitura con transposición.

Por ello, a la hora de realizar la transformación del archivo MusicXML a notación ABC en cada uno de los instrumentos transpositores se indica la transposición que debe realizarse para reproducir el archivo.

```
V:1 treble transpose=-12 nm="Guitarra acústica" snm="Guitarra acústi
two voices reble nm="Piano" snm="Piano"
V:3 treble transpose=-9 nm="Saxo alto 1" snm="Saxo alto 1"
V:4 treble transpose=-7 nm="Txistu" snm="Txistu"
V:5 perc nm="Batería" snm="Batería"
```

FIGURA 28: Notación ABC, descripción de los instrumentos transpositores.

Como se puede observar en la imagen anterior, la reproducción de la voz del saxofón requiere transportar esa voz nueve semitonos hacia abajo y la del txistu siete. El piano y la batería se mantienen. La guitarra requiere un transporte de 12 semitonos hacia abajo lo cual indica que se reproducirá una octava más abajo pero coincidiendo con las notas escritas.

El problema al reproducir estos instrumentos viene cuando se quiere realizar un cambio de tono a la partitura general. El cambio de tono se indica en la información general de la partitura, pero los instrumentos transpositores tendrán su propia indicación de transporte y el reproductor mantiene el tono de estos instrumentos. Por tanto, se realizan una serie de cambios en la función de transporte para modificar el transporte indicado en cada uno de estos instrumentos.

```
definitivomidi = definitivomidi.replace("transpose=-12", "guitarra1");
definitivomidi = definitivomidi.replace("transpose=-9", "saxo1");
definitivomidi = definitivomidi.replace("transpose=-2", "metal1");
definitivomidi = definitivomidi.replace("transpose=-7", "txistu1");
definitivomidi = definitivomidi.replace("transpose=-5", "Flautasol");
definitivomidi =definitivomidi.replace("transpose=-3", "clarinetela");
definitivomidi = definitivomidi.replace("transpose=2", "clarinetere");
definitivomidi = definitivomidi.replace("Flautasol","transpose=-4");
definitivomidi =definitivomidi.replace("clarinetela", "transpose=-2");
definitivomidi =definitivomidi.replace( "clarinetere", "transpose=3");
definitivomidi = definitivomidi.replace("guitarra1", "transpose=-11");
definitivomidi = definitivomidi.replace("saxo1", "transpose=-8");
definitivomidi = definitivomidi.replace("metal1", "transpose=-1");
definitivomidi = definitivomidi.replace("txistu1", "transpose=-6");
```

Mediante el código anterior se realizan las transformaciones necesarias en los instrumentos transpositores para modular un semitono hacia arriba la partitura. Es decir, si se quiere modular la canción un semitono hacia arriba, el transporte de la guitarra será de -11 semitonos en lugar de -12. Con estos cambios en la partitura general y en la información de los instrumentos transpositores la reproducción será correcta y las tonalidades coincidirán con los cambios realizados en otros instrumentos.

Como se puede ver en el código anterior se realizan cambios para instrumentos en Mib, Sib, Fa y la guitarra que requiere un cambio de octava, ya que los instrumentos transpositores comunes se encuentran en esos tonos. Clarinetes, trompetas, saxofón

tenor o bombardino en Sib, saxofón contralto en Mib y trompa o txistu en Fa. También se han incluido instrumentos poco comunes en la, re y sol.

5.5.7. Guardar la partitura completa

Una vez creada la partitura completa, es decir, una vez creada una partitura con las particellas correspondientes, se podrá descargar y guardar el archivo ABC resultante. Se trata del archivo que incluye la notación ABC que se visualiza en pantalla en la parte izquierda. Esta opción solo estará disponible una vez creada la partitura completa.

No obstante, no todos los editores de partituras permiten leer archivos en notación ABC. Sin embargo, estos software son capaces de abrir y reconocer archivos MusicXML o MIDI. En el apartado anterior se había realizado la conversión de notación ABC a MIDI necesaria para reproducir la partitura. Por tanto, una vez creada la partitura completa se permite la opción de descargar el archivo resultante como MIDI. Para ello, se ha creado la siguiente función que toma el archivo ABC creado y el nombre del archivo XML subido y se llama a la función downloadMidiFromAbc() que convertirá el archivo en formato ABC a MIDI y se iniciará la descarga.

```
function abctomidi(){
      var archivo2 = document.getElementById("abcfinal").textContent;
    var fnmidi = xmlFileName.replace (/\.xml/i, '.mid');
      downloadMidiFromAbc(archivo2, fnmidi);
}
```

La partitura completa creada puede transportarse a cualquier tonalidad y esta opción modificará la tonalidad del MIDI creado. Por tanto, al descargar la partitura como MIDI la tonalidad podrá variar en función de los cambios realizados. La armadura reflejará la nueva tonalidad. Por ello, en la función abctomidi() se coge la notación ABC correspondiente al contenedor de HTML en la que se muestra la partitura final.

5.5.8. Editor ABC.

Como último cometido de esta página se ha incluido al final un *iframe* el cual incluye un editor de ABC. De forma que el usuario pueda copiar el código ABC de la partitura completa creada en el editor y reproducir el resultado en la web. Este editor incluye también un visor de partituras, aunque no es tan preciso como el utilizado en la propia web. No obstante, este editor puede comprobar el resultado de realizar pequeñas

modificaciones en la partitura ABC para aquellos músicos que comiencen a familiarizarse con este código.

5.6. Página web para recuperar antiguas versiones de las particellas. (recuperar.html)

En el apartado 5.3 se ha descrito la base de datos creada para almacenar las particellas de los usuarios. Ésta incluye una tabla en la que se almacenan las antiguas versiones de las partituras creadas por cada usuario (historico_xml). En un principio, se pensaban dejar disponibles todas las versiones de cada particella en la página para crear una partitura general. Sin embargo, por la cantidad de archivos que pueden acumularse se ha decidido crear una página para que cada usuario pueda revisar y recuperar antiguas versiones de su partitura (recuperar.html).

Esta página mostrará a cada usuario una tabla con la información de las particellas que haya ido guardando. El usuario podrá realizar una lectura de los archivos en pantalla de forma parecida a la lectura de partituras de la página de unir particellas. Seleccionará una de las partituras o antigua versión y podrá visualizarla en pantalla a través del botón 'carga una partitura'. Al pulsar este botón, quedará disponible un botón para descargar dicha particella, en formato MIDI, siguiendo el mismo procedimiento de conversión que en la página de unir particellas.

FIGURA 29: Página recuperar.html.

Esta página primero realiza un control de usuarios similar al de la página de guardar particellas. Se accede a través de una actividad con URL de Moodle y al igual que en el caso anterior se le sumará el número id del usuario a dicha URL. Una vez reconocido el usuario la página mostrará las particellas del usuario guardadas en historico_xml. Para ello, se utiliza una función similar a la anterior ComprobarAlumno():

```
function ComprobarAlumno() {
      if (id_usuario != '') {
    $.ajax({
        url: getUrl("alumnoshst/") + id_usuario,
        type: 'get',
            async: false,
        dataType: "json",
        success: function (data) {
            if (data.length > 0) {
                $('#tb_partes').html("");
            archivosXML = [];
            for (var i = 0; i < data.length; i++) {
                   archivosXML[i] = data[i];
                    row = '<tr>';
                    row += '<td> <input type="checkbox" /> </td>'
                    row += '<td>' + data[i].hst_nombre_xml + '</td>';
                    row += '<td>' + data[i].hst_fecha + '</td>';
                    row += '<td>' + data[i].hst_instrumento + '</td>';
                    row += '<td>' + data[i].hst_version + '</td>';
                    row += '</tr>';
                     $('#tb_partes').append(row);
                }}else {
                alert("No existen particellas para este usuario");
            }}, error: function (r) {
            alert("error: leerparte");
        }});}
return archivosXML;}
```

Esta función, en caso de que reconozca al usuario llamará a una nueva función definida en conexion.php para extraer los datos correspondientes al usuario desde la tabla historico_xml. Estos datos se mostrarán en una tabla especificando el nombre de las particellas, la fecha, el nombre del instrumento y la versión. En conexión.php se añade la siguiente función:

```
$app->get('/alumnoshst/:id', function ($id) use ($db) {
    $sql = "select hst_id, hst_alumno, hst_grupo, hst_xml,
hst_nombre_xml, hst_fecha, hst_instrumento, hst_version from
historico_xml where hst_alumno = :id";
    $consulta = $db->prepare($sql);
    $consulta->execute(array(
        ':id' => $id,
    ));
    $resultados = $consulta->fetchAll(PDO::FETCH_ASSOC);
      echo json_encode($resultados);
});
```

En la llamada a esta función se añade el número id del usuario que ha accedido a la página para recuperar antiguas versiones. Esta última función accederá a la tabla historico_xml leyendo únicamente las filas correspondientes al id de usuario que haya accedido a la página. Para ello, iguala hst_alumno con el número id que se introduce en la función.

5.7. Apariencia

Por último, para acabar con la página web correspondiente a la actividad de partituras colaborativas se ha modificado la apariencia general a través del archivo CSS. Se ha escogido un color principal para el menú superior y el *footer* o pie de página. Además, el título principal de la página también combina con el color principal. En la parte superior se ha incluido una imagen relacionada con la música y las partituras que será utilizada también en la página de guardar los archivos XML.

Por otro lado, se han definido clases para aplicar el mismo formato a los botones y desplegables utilizados en la web, definiendo un estilo genérico para todos. Además, se ha respetado el mismo formato también para la visualización de cada partitura. A cada una de ellas se le ha incluido un título en una banda azul y se ha modificado el color de fondo de la partitura y de la notación ABC de la misma.

Se ha intentado respetar la apariencia del tema escogido para el entorno virtual de Moodle. De forma que la transición entre las distintas páginas no sea tan notable visualmente.

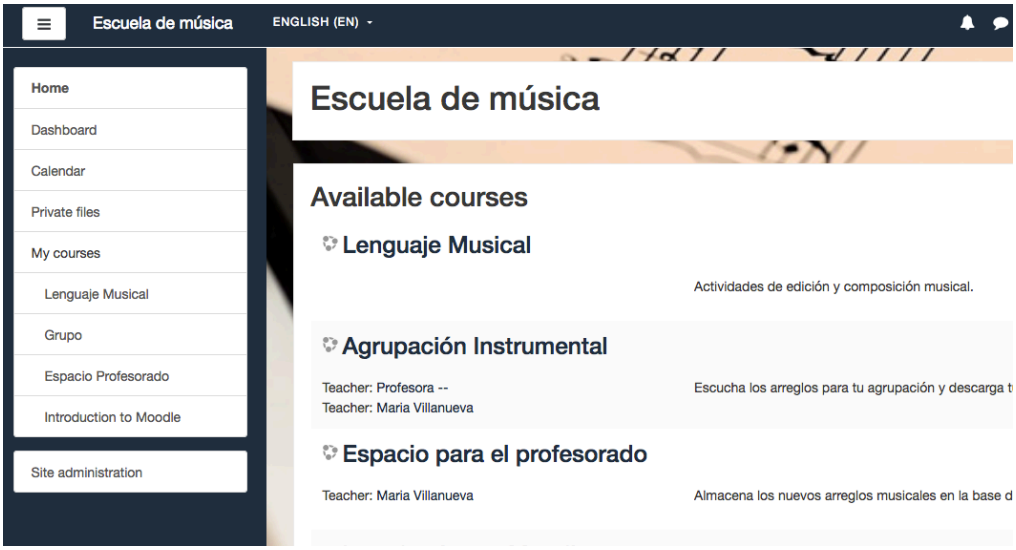


FIGURA 30: Moodle.

6. Enlaces web

Las páginas web creadas se han subido a un servidor gratuito el cual muestra las páginas durante el día.

1. Entorno virtual de aprendizaje:

https://escuelasdemusica.moodlecloud.com

2. Página para guardar particellas:

https://unirparticellas.000webhostapp.com/guardarXML_2.html?idusuario=7

3. Página para recuperar antiguas versiones de particellas:

https://unirparticellas.000webhostapp.com/recuperar.html?idusuario=7

4. Página para unir particellas:

https://unirparticellas.000webhostapp.com/unirparticellas.html

7. Instalación y testeo.

Para comprobar el funcionamiento de la aplicación se ha realizado un testeo en dos fases. El primero de ellos, consiste en probar el funcionamiento de las páginas web para guardar o recuperar partituras en XML y la de unir particellas. El segundo consiste en probar el funcionamiento de Moodle y la relación entre estas páginas.
Para comprobar el funcionamiento de las páginas web creadas han sido alojadas en un servidor gratuito. A continuación, se han probado con diversos usuarios con distintas características: músicos, estudiantes de música y personas ajenas a la música. La página de unir particellas ha sido testeada sin recibir indicaciones previas sobre el funcionamiento o la finalidad de la página.

7.1. Funcionalidad de las páginas

7.1.1. Página para guardar archivos XML.

El resultado de la utilización de esta página web ha sido correcta. El usuario comprende su funcionamiento ya que consiste simplemente en guardar una particella en una base de datos. No obstante, al acceder por primera vez como usuario a la página, mostraba una tabla con los datos de la base de datos partes_xml con los campos vacíos como null.
Al resultar confusa esta información se decide ocultar la tabla para usuarios que todavía no han guardado una versión de su particella. Para posibles actualizaciones de la misma se dejará visible.

7.1.2. Página para unir particellas.

Los usuarios prueban esta página con una base de datos con tres grupos y distintas composiciones. Cada uno de los grupos dispondrá de más de una particella para poder cargar en pantalla y unir en una partitura general. Los usuarios no tendrán indicaciones sobre el funcionamiento de la página aunque se les indicará que pueden seguir las instrucciones indicadas al pulsar el botón de ayuda.
En general, el resultado es que tras varias pruebas el usuario comprende el funcionamiento de la página, sin embargo, considera que hay muchos botones.
Una de las medidas que se toma para simplificar el funcionamiento y disminuir el número de botones es eliminar el botón de limpiar e incluir esta opción cada vez que se presione el botón cargar. No obstante, este botón no se suprime ya que da la posibilidad de cargar únicamente las partituras que seleccione el usuario. Tanto el

proceso de cargar como el de unir particellas es lento y por tanto, se decide realizarlo en dos fases por si el usuario quisiera ver las partituras antes de unirlas.

Por otro lado, los editores de partituras permiten crear partituras que no especifican el instrumento para las que están escritas. En este caso, la partitura en ABC tampoco contendrá esta indicación. Además, es decisión del usuario incluir el nombre del instrumento en el archivo que guarde en la base de datos. En consecuencia el usuario de la página de unir particellas no conoce el instrumento para el que se crea cada particella y resulta confuso.

Para solucionar este problema, se decide incluir en la página para guardar particellas un campo en el que el usuario escriba el nombre del instrumento para el que ha editado la partitura que vaya a subir. Esta información del instrumento será almacenada en la base de datos creando para ello una nueva columna tipo VCHAR. Esta información será incluida en las tablas de la página de unir particellas para que el usuario pueda conocer el nombre del instrumento al que corresponden.

En la siguiente imagen se puede observar el cambio realizado en la tabla, la cual incluye el nombre del instrumento de cada particella. Además, se ha prescindido del botón limpiar.

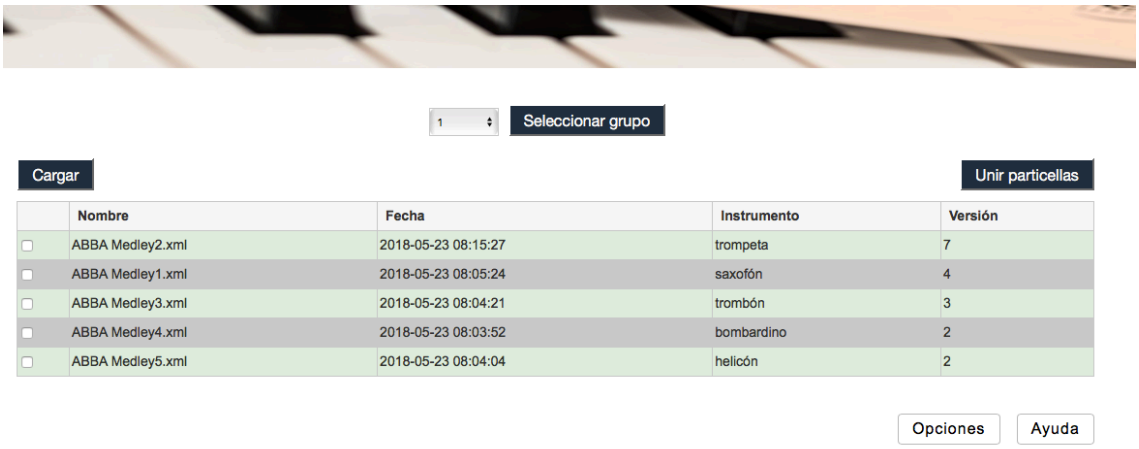


Figura 31: Cambios introducidos en la página para unir particellas.

Para poder incluir la información sobre el instrumento de cada particella se ha incluido un input en la página web de guardar particellas. Este cambio se puede visualizar en la siguiente imagen:

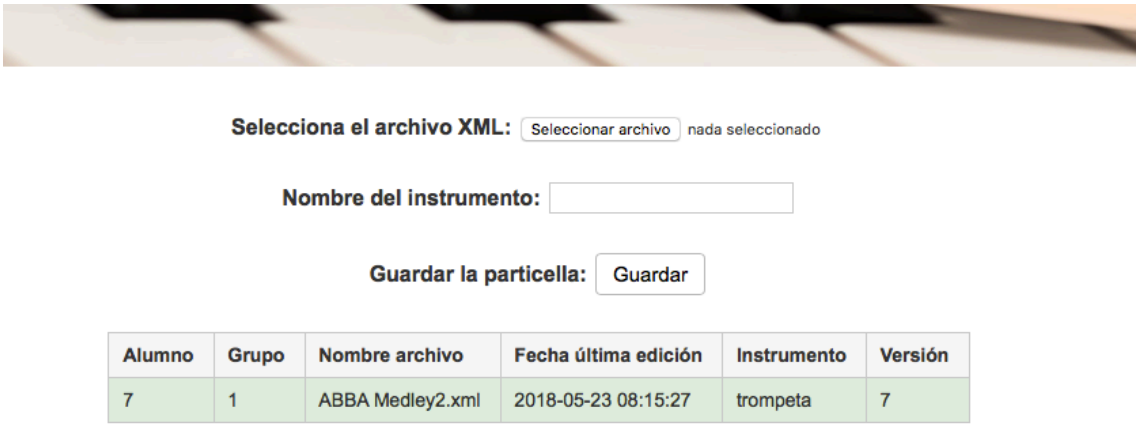


Figura 32: Cambios introducidos en la página para guardar particellas.

7.2. Navegación y diseño.

Para verificar la interacción del usuario con el sistema se realizan pruebas de diseño y navegación. Estas pruebas consisten en verificar los enlaces del menú de las páginas creadas así como los enlaces que conducen desde Moodle a dichas páginas. Por otro lado, se verifica que el diseño de las dos páginas web creadas se adapten a cualquier dispositivo o tamaño de pantalla. Para ello, se probará en móviles, tablets y ordenadores.

En los dispositivos de pequeñas pantallas la información de cada partitura en ABC así como la partitura se muestran en un tamaño muy pequeño. Por tanto, se decide prescindir de la visualización en pantalla de la notación ABC, para mostrar únicamente la partitura de cada instrumento. En dispositivos de tamaño de pantalla medio se realiza el mismo cambio, sin embargo, se mantiene la notación en ABC de la partitura resultante para poder realizar cambios con el editor.

A partir de esta modificación, el usuario demanda poder ver la partitura completa en todo el ancho de la pantalla en el ordenador o pantallas más grandes también. Por ello, se incorpora una última función en la que al clicar sobre la partitura general creada se amplía utilizando el 100% del ancho de la pantalla. Al volver a clicar se reduce el tamaño volviendo al tamaño original y así sucesivamente.

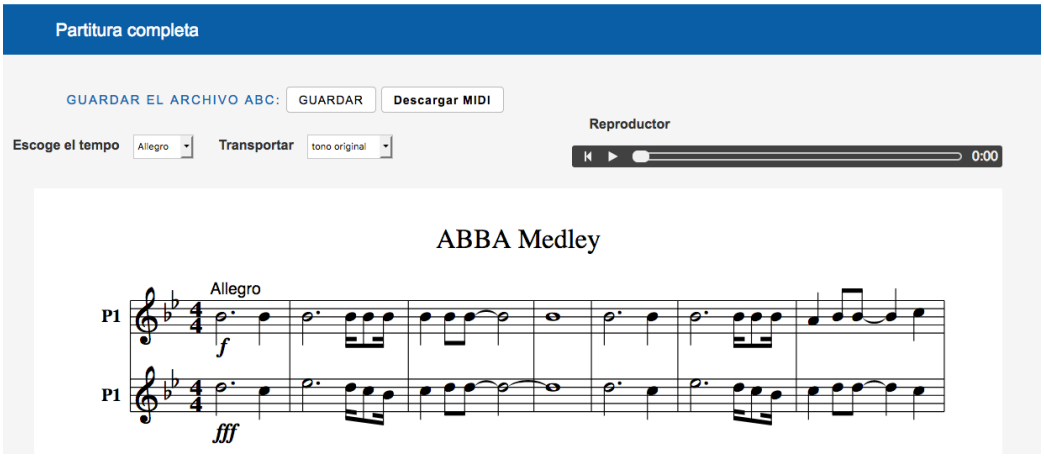


Figura 33: Mostrar la partitura ampliada.

7.3. Pruebas de compatibilidad.

Además de realizar pruebas en distintos dispositivos electrónicos también se realizarán pruebas para verificar la compatibilidad con distintos navegadores web. Para ello, se ha probado la web con Firefox, Chrome, Safari y otros navegadores de dispositivos móviles.

En estas pruebas hemos comprobado que el funcionamiento del entorno virtual y de la página de guardar particellas no varía de un navegador a otro, excepto pequeñas modificaciones como el tipo de letra. Sin embargo, si hay diferencias en la página de unir particellas. El funcionamiento general de esta página es correcto, no obstante, hay problemas con la reproducción en Safari, ya que realiza variaciones en el tempo de reproducción. En Google Chrome y Mozzilla funciona correctamente teniendo en cuenta que la web ha sido probada con navegadores actualizados. Por otro lado, en las últimas versiones de navegadores móviles también funciona.

8. Conclusiones y líneas futuras

El objetivo principal de este proyecto ha sido crear actividades en la web para trabajar con partituras de manera colaborativa. La primera propuesta consistía en utilizar algunas de las funcionalidades de MuseScore, sin embargo, por problemas de licencias del software se descartó esta opción. En su lugar, tras un estudio de la tecnología de licencia GNU disponible, se ha desarrollado un sitio web para guardar particellas de distintos usuarios y crear una partitura general a partir de dichas partituras dirigido a la enseñanza musical.

- Se ha adaptado un entorno virtual a través de Moodle para la gestión de usuarios y las actividades de colaboración musical de las páginas web creadas.
- Se ha creado una base de datos para almacenar las particellas de los usuarios con la información sobre el grupo al que pertenecen.
- Se ha desarrollado una página web para almacenar particellas en la base de datos y una segunda para leer dichas particellas y unir las correspondientes al mismo grupo. Esta última página web permite crear una partitura general a partir de partituras instrumentales creadas por distintos usuarios y reproducir el resultado.
- Se ha creado una última página web para recuperar antiguas versiones de las partituras almacenadas por cada usuario.

La apariencia del sitio web se adapta a los distintos dispositivos y se ha intentado mantener un diseño similar al de Moodle en las páginas web creadas. El sitio web no ha sido puesto en marcha en una escuela específica pero si se han realizado pruebas con usuarios de distintas características a través de distintos navegadores. A partir de las sugerencias de los usuarios se han realizado cambios para facilitar el funcionamiento de la web.

No obstante, han surgido problemas de compatibilidad con el reproductor MIDI de la web en algunos navegadores. En Safari la reproducción no se realiza a tiempo real y para garantizar el funcionamiento correcto en navegadores como Mozilla o Chrome hay que instalar las últimas versiones. A pesar de estos problemas, el sitio web podría ponerse en marcha en escuelas de música o conservatorios, ya que en estos centros se utilizan PCs y no ordenadores Apple. Este proyecto está dirigido a centros de bajo presupuesto que en general no pueden permitirse estos equipos, aunque siempre habrá usuarios que trabajen con estos equipos desde casa.

Por otro lado, a la hora de realizar la conversión de particellas de notación ABC a MIDI surgen avisos o pequeños errores. Estos errores dividen la reproducción en partes, separando la reproducción de cada voz.

A partir de las pruebas realizadas con los usuarios de la web se proponen las siguientes líneas de trabajo futuras:

- Cambiar o adaptar el reproductor MIDI disponible en la web para que sea compatible con todos los navegadores incluyendo Safari.
- Diseñar pruebas para analizar los errores de conversión de notación ABC a MIDI y corregir los fallos en la reproducción de las partituras. Modificar el conversor de abc2midi cuando se corrijan sus limitaciones.
- Ampliar el número máximo de instrumentos o particellas para poder utilizarlo por orquestas o bandas.
- Incluir la posibilidad de introducir particellas de más de una voz por cada usuario. Como por ejemplo, incluir varias partituras de piano que no hayan sido separadas por voces o partituras instrumentales con varias voces de una misma cuerda.

9. Bibliografía

1. Almeida, J.J., Carvalho, N.R anaperd Oliveira J.N. (2012). *WIKI::SCORE A collaborative Enviroment For Music Transcription And Publishing*. Social Shaping of Digital Publishing: Exploring the Interplay Between Culture and Technology, 82-93. Recuperado de: https://repositorium.sdum.uminho.pt/bitstream/1822/37235/3/9781614990659.pdf#page=96
2. Apereo Foundation (2014). Sakai Project. Recuperado de: https://www.sakaiproject.org
3. Aydeé, M. (2015). Desarrollo de una aplicación Web empresarial de ayuda a la producción de campañas publicitarias. Trabajo Fin de Grado de la Universidad Politécnica de Madrid. España, Madrid. Consultado el 27/05/2018. Recuperado de: http://oa.upm.es/34773/1/PFC_MARJORIE_GANCHALA_MORA.pdf
4. Bligh, J., Jennings, K., y Tangney, B. (2006). Designing Interfaces for Collaborative Composition. Trinity College. Dublín, Irlanda. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=18175B2D5C58F1FA12395141934010A3?doi=10.1.1.157.5121&rep=rep1&type=pdf
5. Bonte, T., Froment, N. Y Schweer, W. (2008). MuseScore. Recuperado de: https://musescore.org/es
6. Calderón, D., Martín, C. Y Gustems, J. (2017). Las tecnologías: un recurso interdisciplinar en la educación artística en Secundaria. Experiencias Educativas pp.201-211. Universidad Internacional de La Rioja y Universitat de Barcelona.
7. Carrera, E. (2017). Trabajo fin de Grado: Música y TIC en el aula del siglo XXI. Universidad de Cádiz, España.
8. Dougiamas, M. (2002). Moodle - Open-source learning platform. Recuperado de: https://moodle.org
9. Equipo Editorial Linux (2016). Guía sobre MuseScore: un gran centro de partituras para tu distro Linux. *AB Internet Networks.* Recuperado de: https://www.linuxadictos.com/guia-musescore.html
10. Galera, M.M. (2013). Recursos Para La Enseñanza De La Lectura Musical: El Uso De Los Editores De Partituras En Actividades Creativas. ESPACIO Y TIEMPO, Revista de Ciencias Humanas, No 27-2013, pp. 95-105.
11. Giráldez, A. (2005). Internet y Educación Musical. Barcelona, España. Grao.
12. Klügel, N., FrieB, M. Y Groh, G. (2011). An Approach to Collaborative Music Composition. Munich University of Applied Sciences. Múnich, Alemania.

Disponible en: https://www.researchgate.net/profile/Florian_Echtler/publication/228963416_An_Approach_to_Collaborative_Music_Composition/links/09e41505b718053f2c000000.pdf

13. Lorenzo, A. y Extremiana, P. (2009). Proyectos Colaborativos: una metodología de desarrollo de la creatividad en educación musical. Revista RecreArte 11, Diciembre del 2009. Santiago de Compostela, España.
14. Rodríguez, J.C. (2012). ABC musical. Consultado el 07/05/2018. Recuperado de: http://www.evoljava.com/dl/CursoDeMusica.pdf
15. Sébastien, V., Sébastien, D. y Conruyt, N. (2014). Dynamic music lessons on a collaborative score annotation platform. IREMIA - Laboratoire d'Informatique et de Mathématiques. Saint-Denis, Francia.
16. Shlien, S (2018). Guide to Advanced Features o Abc2midi. Consultado el 3/04/2018. Recuperado de: https://ifdo.ca/~seymour/runabc/abcguide/abc2midi_guide.html

10. Librerías:

4. xml2abc.js. Enlace: https://wim.vree.org/svgParse/xml2abc.html
5. abc2svg.js. Enlace: https://github.com/moinejf/abc2svg
6. abc2midi.js. Disponible en: https://github.com/skooter500/abc2midi-js/blob/master/abc2midi.js.mem

11. Anexo I: Función para unir particellas

```
function myFunction(obj) {
$('#savebtn').attr ('disabled', false);
    $('#downloadMidi').attr ('disabled', false);
document.getElementById('num_instrumentos1').style.display=    'inline-
block';
document.getElementById('transpose').style.display= 'inline-block';
document.getElementById('h3velocidad').style.display= 'inline-block';
document.getElementById('h3transpose').style.display= 'inline-block';
document.getElementById('reproductor').style.display= 'inline-block';
document.getElementById('guardar').style.display= 'inline-block';
document.getElementById('resfinal').style.display= 'inline-block';
document.getElementById('particompleta').style.display=        'inline-
block';
document.getElementById('example').style.display= 'inline-block';
   obj.disabled=false;
var  file,  file2,  file3,  file4,  file5,  file6,  file7,  file8,  file9,
file10;
   var elemento = document.getElementById('particompleta');
      var posicion = elemento.getBoundingClientRect();
      var posici = posicion.top ;
    $("html, body").animate({ scrollTop: posici }, 2000);
    <!-- file = $('#fileinput').prop ('files')[0]; -->
    <!-- file2 = $('#fileinput2').prop ('files')[0]; -->
    <!-- file3 = $('#fileinput3').prop ('files')[0]; -->
    <!-- file4 = $('#fileinput4').prop ('files')[0]; -->
    <!-- file5 = $('#fileinput5').prop ('files')[0]; -->
    <!-- file6 = $('#fileinput6').prop ('files')[0]; -->
    <!-- file7 = $('#fileinput7').prop ('files')[0]; -->
    <!-- file8 = $('#fileinput8').prop ('files')[0]; -->
    <!-- file9 = $('#fileinput9').prop ('files')[0]; -->
    <!-- file10 = $('#fileinput10').prop ('files')[0]; -->
      var save='';
    <!-- if (!file || !file2){ -->
    if (1 == 2){
     alert('No hay dos partituras cargadas');
     } else {
        recabcText=abcText+recabcText;
        var V2= recabcText.indexOf("V:2");
        var num_ins_piano2='';
        if (V2!=-1){
```

```
            num_ins_piano2 = instrumentos +1;
                  var construyendo2='';
            var principio2 = abcText.indexOf('V:1');
            var introduccion2 = abcText.substring(0, principio2-1);
            var insertar_ins2 = '';
            var                                          contadorinst2=
['V:1','V:2','V:3','V:4','V:5','V:6','V:7','V:8','V:9','V:10'];
            for(j2=0;j2<instrumentos; j2++){
                  var posV12 = recabcText.indexOf('V:1');
                  save = recabcText.substring(0,recabcText.length-1);
recabcText = recabcText.substring(posV12-1, recabcText.length-1);
      var buffer22= recabcText.substring(2, recabcText.length-1);
                  var posV22= buffer22.indexOf('V:1');
                  var ins2= recabcText.substring(0, posV22+1);
                  var posV1verdadera2 = recabcText.indexOf('V:1');
                  ins2= ins2.replace('V:1',contadorinst2[j2]);
                  insertar_ins2 = insertar_ins2 +ins2;
                  recabcText  =  recabcText.substring(posV1verdadera2+1,
recabcText.length-1);
                  var posV2verdadera2 = recabcText.indexOf('V:1');
recabcText  =  recabcText.substring(posV2verdadera2,  recabcText.length-
1);         if(recabcText.indexOf('X:1')!=-1){
                  var posX12 = recabcText.indexOf('X:1');
                  var infoinst4 = recabcText.substring(0, posX12);
      recabcText=recabcText.substring(posX12-1, recabcText.length-1);

      infoinst4=infoinst4.replace('V:1', contadorinst2[j2]);
      construyendo2 = construyendo2 + infoinst4;
                  }else{
      var infoinst22 = recabcText.replace('V:1', contadorinst2[j2]);
                  construyendo2 = construyendo2 + infoinst22;    }}
insertar_ins2=insertar_ins2.replace(contadorinst2[1],
'cambio');insertar_ins2=insertar_ins2.replace(contadorinst2[1],
contadorinst2[instrumentos]);
      insertar_ins2=insertar_ins2.replace('cambio', contadorinst2[1]);
            if(insertar_ins2.indexOf('L:')!=-1){
                  var posL = introduccion2.indexOf('L:');
var buffer1 = introduccion2.substring(posL, introduccion2.length-1);
                  var posSigV = buffer1.indexOf('\n');
                  var buffer11 = buffer1.substring(0, posSigV);
                  var posLotra = insertar_ins2.indexOf('L:');
```

```
var bufferL2= insertar_ins2.substring(posLotra, insertar_ins2.length-
1);         var saltos= bufferL2.indexOf('\n');
                  var bufferL1 = bufferL2.substring(0, saltos);
                  var posLultimo=save.indexOf('L:');
                  save=save.substring(posLultimo, save.length-1);
                  var posFinL= save.indexOf('\n');
                  save=save.substring(0,posFinL);
                 //var mute= '%%MIDI voice mute';
                  if(buffer11==save){
                        insertar_ins2 = insertar_ins2 +"\n"+  buffer11;
                  }else{
                      insertar_ins2=insertar_ins2 +"\n"+ save;}}
      construyendo2=construyendo2.replace(contadorinst2[1], 'cambio');
construyendo2=construyendo2.replace(contadorinst2[1],
contadorinst2[instrumentos]);
      construyendo2=construyendo2.replace('cambio', contadorinst2[1]);
      definitivo = introduccion2 + insertar_ins2 +"\n" +construyendo2;
        }else{
            num_ins_piano=instrumentos;

            var construyendo='';
            var principio = abcText.indexOf('V:1');
            var introduccion = abcText.substring(0, principio-1);
            var insertar_ins = '';
            var                                          contadorinst=
['V:1','V:2','V:3','V:4','V:5','V:6','V:7','V:8','V:9','V:10'];

            for(j=0;j<num_ins_piano; j++){
                  var posV1 = recabcText.indexOf('V:1');
                  recabcText      =      recabcText.substring(posV1-1,
recabcText.length-1);
                  var         buffer2=         recabcText.substring(2,
recabcText.length-1);
                  var posV2= buffer2.indexOf('V:1');
                  var ins= recabcText.substring(0, posV2+1);

                  var posV1verdadera = recabcText.indexOf('V:1');
                  ins= ins.replace('V:1',contadorinst[j]);
                  insertar_ins = insertar_ins +ins;
recabcText = recabcText.substring(posV1verdadera+1, recabcText.length-
1);
```

```
                var posV2verdadera = recabcText.indexOf('V:1');
recabcText = recabcText.substring(posV2verdadera, recabcText.length-
1);
                if(recabcText.indexOf('X:1')!=-1){
                var posX1 = recabcText.indexOf('X:1');
                var infoinst = recabcText.substring(0, posX1);
      recabcText=recabcText.substring(posX1-1, recabcText.length-1);
                infoinst=infoinst.replace('V:1', contadorinst[j]);
                construyendo = construyendo + infoinst;}
                else{
            var infoinst2 = recabcText.replace('V:1', contadorinst[j]);
                construyendo = construyendo + infoinst2;}}
            definitivo = introduccion + insertar_ins +"\n"
+construyendo;
            }
        var div = document.getElementById("abcfinal");
        div.textContent = definitivo;
            document.getElementById("iframe").style.display= 'inline-
block';
        dolayout3 (definitivo);
        cargar(definitivo);
        return definitivo;
   }}
```

12. Anexo II: Glosario musical

Instrumentos transpositores: Son instrumentos que al tocar sonidos producidos suenan más agudo o grave que los escritos. Algunos de los ejemplos más comunes son el clarinete y la trompeta en si b y la trompa en fa.

Transporte: Es el cambio de tonalidad en una pieza musical.

Notación ABC: La notación musical ABC es un lenguaje para escribir música el cual se basa en los caracteres ASCII. Es un lenguaje para almacenar y leer música en el ordenador pero al mismo tiempo es comprensible por el humano. Hay disponible software de licencia GNU y gratuita para trabajar con archivos de este formato.

Particellas: es una partitura en la que está escrito únicamente lo que debe interpretar un intérprete o conjunto de intérpretes de un instrumento. Al unísono.

Partitura general o completa: en estas partituras se escriben simultáneamente todos los sonidos que han de tocar todos los músicos, es decir, las partes de todos los instrumentos.

Tonalidad: la tonalidad es la clave de una obra musical. Gira en torno a una nota específica denominada tónica, la cual tiene asociados una serie de acordes o progresiones armónicas y escalas.