
This is the **published version** of the bachelor thesis:

Esteban Ferrer, Abel; Fornes Bisquerra, Alicia, tut. Creació d'una aplicació per validar partitures musicals. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/308762>

under the terms of the  license

Creació d'una aplicació per validar partitures musicals

Abel Esteban Ferrer

Resum— Aquest projecte té per objectiu el desenvolupament d'una aplicació innovadora per validar i editar partitures musicals llegides per un sistema de reconeixement òptic musical (OMR). Els sistemes de OMR són àmpliament utilitzats, especialment en les partitures manuscrites o amb notació complexa, però sovint introdueixen errors en el procés de conversió, especialment en les partitures amb notació complexa o antiga, com és el cas d'aquest treball. L'aplicació desenvolupada permet als usuaris veure les partitures en diferents formats, com ara l'imatge original o l'interpretat pel programa MuseScore4 i editar-les per corregir els errors generats durant la transcripció. El treball inclou funcionalitats com visualització per compassos, ajustament de brillantor per a l'anàlisi i navegació intuïtiva. A més, el projecte posa un fort èmfasi en la usabilitat i l'accessibilitat, buscant que l'aplicació sigui útil per musicòlegs sense necessitat de coneixements tècnics.

Paraules clau— Reconeixement òptic musical (OMR), Correcció, Partitures manuscrites, MusicXML, MuseScore4, Digitalització de partitures

Abstract— This project aims to develop an innovative application for validating and editing musical scores read by an Optical Music Recognition (OMR) system. OMR systems are widely used, especially to transcribe handwritten scores or complex notation, but they often introduce errors during the conversion process, especially with complex or old notation, as is the case in this work. The developed application allows users to view the scores in different formats, such as the original image or the version interpreted by the MuseScore4 program, and edit them to correct errors generated during transcription. The work includes features such as measure-by-measure visualization, brightness adjustment for analysis, and intuitive navigation. Additionally, the project places a strong emphasis on usability and accessibility, aiming to make the application useful for musicologists without requiring technical knowledge.

Index Terms— Optical Music Recognition (OMR), Correction, Handwritten Musical Scores, MusicXML, MuseScore4, Score digitization



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

ACTUALMENT, la digitalització de partitures musicals ha experimentat un progrés significatiu gràcies als sistemes de Reconeixement Òptic de Música (OMR). Aquestes eines permeten convertir partitures físiques en formats digitals, com MusicXML, que són molt més fàcils de processar i compartir. No obstant això, aquests sistemes no són perfectes i acostumen a generar errors, especialment quan s'apliquen a partitures antigues, manuscrites o amb notacions complexes. Aquestes limitacions poden donar lloc a transcripcions incorrectes que requereixen una intervenció manual per corregir els errors.

A més, una part important de les partitures encara es crea manualment, ja sigui amb paper i llapis o amb eines digitals bàsiques. Aquest enfocament tradicional sovint

limita l'eficiència dels processos de creació i edició, així com la precisió dels resultats finals. Per tant, la necessitat d'una solució que combini el millor de la transcripció automàtica amb eines de correcció manual accessibles per a tot tipus d'usuaris es fa més evident que mai.

Aquest projecte té com a objectiu desenvolupar una aplicació que permeti editar i validar la sortida dels OMR amb una plataforma intuïtiva i visual. Això inclou proporcionar eines eficients de gestió i correcció per tal de garantir que el resultat final sigui de la màxima qualitat possible, independentment de la complexitat o l'estat inicial de la partitura. Amb aquesta proposta, es pretén no només facilitar la correcció dels sistemes OMR, sinó també facilitar-ne l'ús per a músics, investigadors i altres professionals del món de la música.

- E-mail de contacte: abelestebanferrer4@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Alicia Fornés Bisquerra
- Curs 2024/25

2 OBJECTIUS

L'objectiu principal del projecte és desenvolupar una aplicació per a ordinador que permeti visualitzar una partitura musical manuscrita, que ha estat reconeguda amb un sistema de OMR i validar-la, corregint els errors de reconeixement que hagi pogut realitzar el sistema. Aquesta eina ha de ser accessible i eficient per a usuaris. Els subobjectius són:

- Estudi d'eines existents actuals per a editar partitures i estudi d'OMRs existents.
- Dissenyar i implementar l'aplicació d'edició i validació de partitures.
- Integració d'una base de dades per gestionar la informació associada a les partitures editades.
- Dissenyar i executar proves per validar la funcionalitat i usabilitat de l'aplicació. Es faran proves internes per verificar el correcte funcionament del sistema i proves amb usuaris per obtenir feedback i assegurar que l'aplicació respon a les necessitats reals dels usuaris.
- Elaborar un conjunt de mètriques per a mesurar l'eficiència del OMR, comparant el resultat de l'OMR amb la partitura corregida per l'usuari.

3 ESTAT D'ART

En aquest apartat es presenta una revisió de les eines i tecnologies disponibles per al reconeixement de partitures musicals, especialment enfocades en la lectura i edició de les generades a través de sistemes de Reconeixement Òptic Musical (OMR). Aquesta anàlisi permet identificar les solucions més adequades per a la implementació del sistema proposat.

3.1 Reconeixement òptic musical (OMR)

El Reconeixement Òptic Musical (OMR) és una tecnologia destinada a la digitalització de partitures musicals, convertint-les en formats digitals llegibles, com ara MusicXML. Entre les eines més conegudes i utilitzades dins d'aquest àmbit hi trobem:

- **Audiveris:** Plataforma de codi obert capaç de processar partitures escanejades i convertir-les a formats digitals. Tot i la seva capacitat per analitzar imatges complexes, presenta limitacions en elements decoratius de les partitures, com ara ornamentacions.
- **Photoscore i SmartScore:** Solucions comercials que ofereixen un bon rendiment, tot i que presenten restriccions d'ús associades a llicències de pagament.

Altrament, podem trobar OMRs d'àmbit acadèmic dissenyats per a sistemes on es requereix d'una complexitat i tècnica diferent degut als casos evaluats [18]. Per al present projecte, s'utilitzarà un sistema OMR existent en desenvolupament al CVC. Per tant, no s'aprofundirà en els aspectes tècnics interns d'aquestes tecnologies. [19], [20] i [21]

3.2 Formats de representació musical

La digitalització de partitures implica la conversió a formats que permetin la seva manipulació i processament de manera eficient [1]. A continuació, es descriuen alguns dels formats més rellevants:

- **MusicXML:** Considerat el format estàndard en la representació digital de partitures i utilitzat per a les proves del meu sistema. És àmpliament acceptat i compatible amb diverses aplicacions. [1]
- **MIDI:** Encara que no està directament orientat a la representació visual de partitures [15], és un format àmpliament utilitzat en la música per a la transmissió de dades musicals.
- **MEI:** Format especialment orientat a la recerca musicològica [16]. Permet una edició i anotació precisa de les partitures, tot i que no té la mateixa popularitat que MusicXML.

3.1 Eines d'edició de partitures

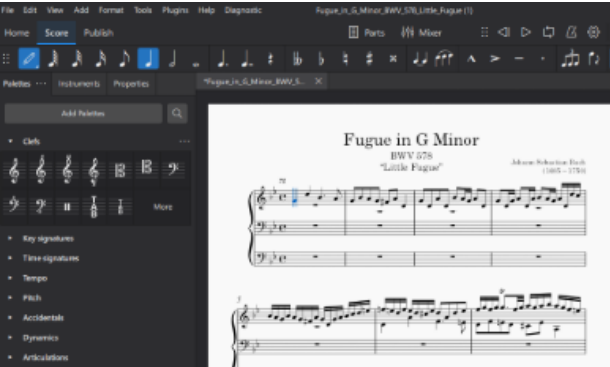
En aquest apartat s'avaluen diferents eines i APIs disponibles per a la visualització, correcció i edició de partitures musicals, preferiblement en format MusicXML. S'ha establert una comparativa basada en criteris clau que han permès seleccionar la millor opció per a la implementació del sistema final. Els criteris principals són:

- La facilitat d'integració, que referencia al grau de complexitat associat a la integració de l'eina en una plataforma web o mòbil.
- La Capacitat d'edició, per determinar el nivell de funcionalitat que ofereix l'eina per editar partitures.
- El Suport amb MusicXML per avaluar la compatibilitat de l'eina amb el format MusicXML, incloent la seva capacitat per importar, editar i exportar partitures.
- La viabilitat com a API, per a indicar la possibilitat d'utilitzar l'eina com a API en un projecte web o aplicació mòbil.

Finalment, saber si és o no de codi obert i, per tant, editable i adaptable segons les necessitats del projecte. Les opcions són:

- **Music21:** És una de les eines més potents per a edicions complexes de partitures [14]. Tanmateix, la seva integració en aplicacions web o mòbils requereix un backend Python, fet que pot complicar una implementació ràpida.
- **Verovio:** És una bona opció per a la visualització de partitures en plataformes web. No obstant això, la funcionalitat d'edició és limitada, i no permet fàcilment la captura d'imatges de partitures originals amb dispositius mòbils. [17]
- **OpenSheetMusicDisplay(OSMD):** OSMD [2] és una eina adequada per a la visualització de partitures, però la manca de funcionalitat d'edició i el desconeixement previ de la llibreria VexFlow fan que sigui menys viable en comparació amb altres alternatives.
- **MuseScore:** Mostrada en la il·lustració 1, tot i ser una eina ideal per a l'edició de partitures, la seva integració

com a API és actualment limitada. Malgrat això, és la opció més interessant per a aplicacions com la meua i es per això que es combinarà amb el sistema. [5] i [6]



Il·lustració 1: Pestanya MuseScore 4

4 METODOLOGIA I PLANIFICACIÓ

El desenvolupament de l'aplicació segueix una metodologia Agile [4], dividida en curtes iteracions que permeten fer ajustaments constants basats en els resultats obtinguts en les reunions i proves realitzades. Aquesta metodologia Agile facilita una forma de treball dinàmica i flexible, amb un enfocament iteratiu que permet revisar i millorar les funcionalitats del projecte de manera contínua. Cada iteració es centra en assolir petits objectius concrets, fet que permet una avaluació constant del progrés i la detecció ràpida d'errors o possibles millores. [7]

Aquesta metodologia ha permès adaptar-se de manera flexible als canvis i nous requeriments del projecte, millorant contínuament les funcionalitats i optimitzant el rendiment de l'aplicació. A la il·lustració 2 es troba el diagrama de Gantt, que mostra les diferents fases i cronograma del desenvolupament del projecte. Concretament, durant el desenvolupament s'ha utilitzat Python, MuseScore [6] i SQLite [11] per a la base de dades, així com diverses llibreries externes de Python [13], les quals han estat fonamentals per implementar les funcionalitats del projecte i garantir la seva eficàcia.

5 REQUISITS

Els següents requisits defineixen les característiques de l'aplicació. Aquests requisits s'han establert tenint en compte les necessitats dels musicòlegs a l'hora de transcriure.

5.1 Requeriments funcionals

La taula 1 mostra els requisits funcionals per a aquest projecte.

ID	Descripció
RF-1	L'usuari ha de poder carregar i visualitzar les partitures de manera que tingui la imatge original per una banda i el MusicXML per l'altre.
RF-2	L'usuari ha de poder veure amb algun tipus d' il·luminació en quin compàs estroba editant respecte a la imatge original.

ID	Descripció
RF-3	L'usuari ha de poder veure el pentagrama i navegar en ell dividit en parts.
RF-4	L'usuari ha de poder fer modificacions sobre la partitura, corregint els errors del OMR.
RF-5	L'usuari ha de poder emmagatzemar partitures i versions editades.

Taula 1: Funcionals

5.2

Requeriments no funcionals

La taula 2 mostra els requisits no funcionals per a aquest projecte.

ID	Descripció
RNF-1	Usable: L'UX ha de ser específic, sent una app intuïtiva i fàcil d'utilitzar sense necessitat de guies.
RNF-2	Segur: La connexió entre BD i plugins ha de ser segura, garantint la protecció de les dades.
RNF-3	El feedback de l'aplicació ha de ser tot lo ràpida possible, executant-se en el menor temps que es pugui cada una de les parts.
RNF-4	Compatible: L'aplicació ha de ser compatible completament almenys amb alguna de les versions de MuseScore. Amés, ha de ser accessible des de diversos dispositius diferents, ja sigui més o menys modern.

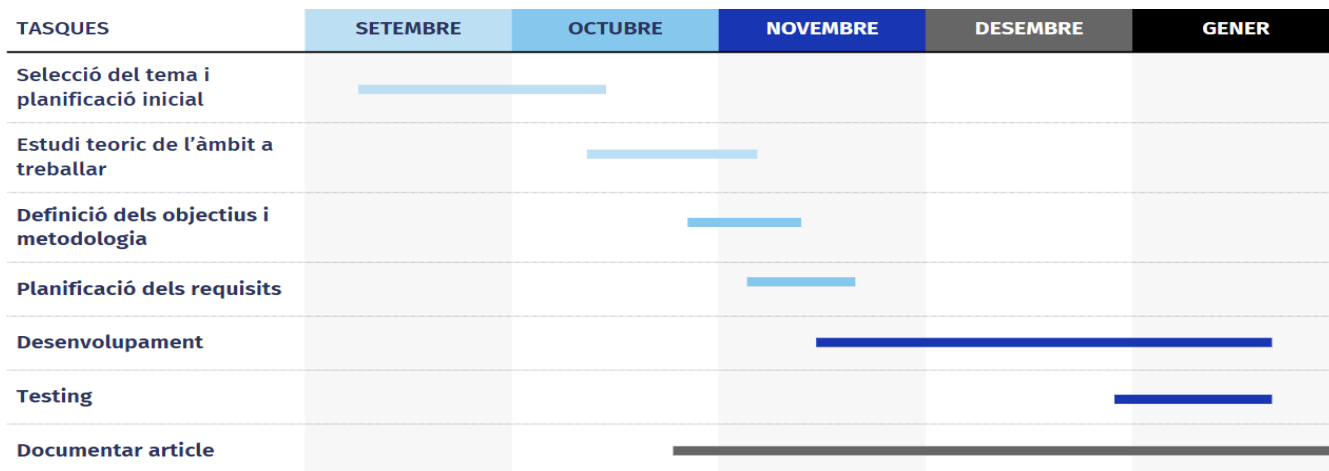
Taula 2: No funcionals

6 DISSENY I DESENVOLUPAMENT

Aquest apartat descriu els processos, les tecnologies i les metodologies emprades per a la implementació del projecte, detallant les arquitectures, els diagrames i les funcionalitats clau que integren l'aplicació.

La fase inicial del projecte va consistir en el disseny de l'arquitectura global de l'aplicació. Aquesta arquitectura modular, que es mostra en el diagrama corresponent, permet una distribució eficient de les funcionalitats i facilita la seva escalabilitat.

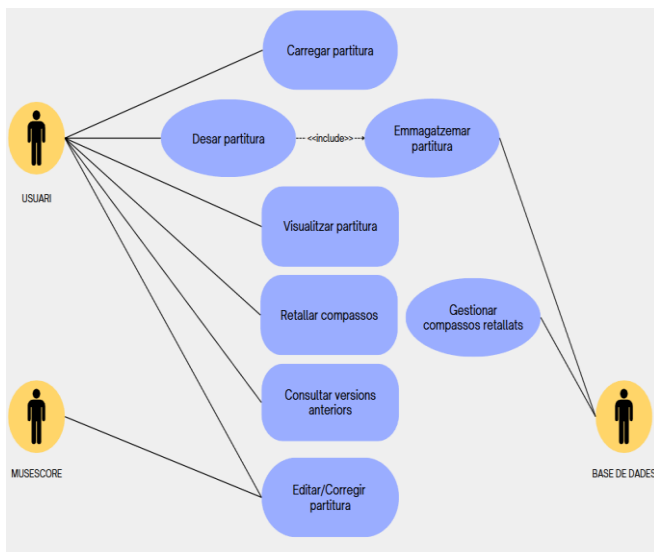
Cada secció de l'aplicació ha estat dissenyada per ser intuïtiva i eficient, permetent als usuaris interactuar amb el sistema de manera fàcil i ràpida. Les funcionalitats estan dividides en diferents mòduls independents, cadascun amb el seu conjunt de característiques específiques. Per exemple, un mòdul pot estar dedicat a la gestió de usuaris, mentre que un altre pot manejar les operacions relacionades amb la seguretat i protecció de dades. Tots aquests mòduls estan interconnectats, com es mostra en els diagrames, de manera que cada acció de l'usuari genera una resposta immediata i adequada des de l'aplicació.



Il·lustració 2: Diagrama de Gantt

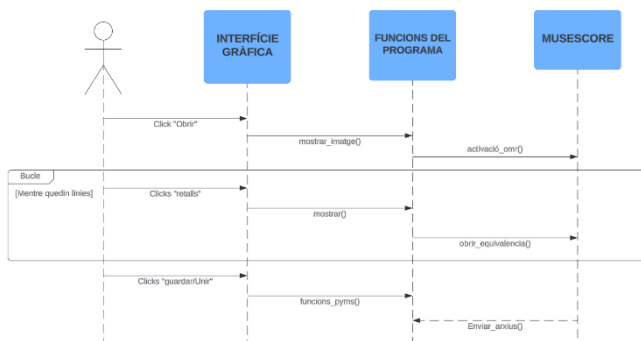
6.1 Casos d'ús

El diagrama de casos d'ús de la tercera imatge il·lustra les funcionalitats principals de l'aplicació, així com les interaccions entre els diferents actors (Usuari, MuseScore i Base de Dades). [10]



Il·lustració 3: Diagrama de casos d'ús

6.2 Diagrama de seqüència



Il·lustració 4: Diagrama de seqüència

El diagrama de la quarta il·lustració permet visualitzar de manera clara i ordenada com es produeixen les comunicacions i quins missatges es transmeten al llarg de l'ús de l'app. En aquest cas, el diagrama destaca les accions específiques que es duen a terme en cada etapa del procés més els fluxos de dades o informació entre els elements del sistema (interfície gràfica - musescore). [12]

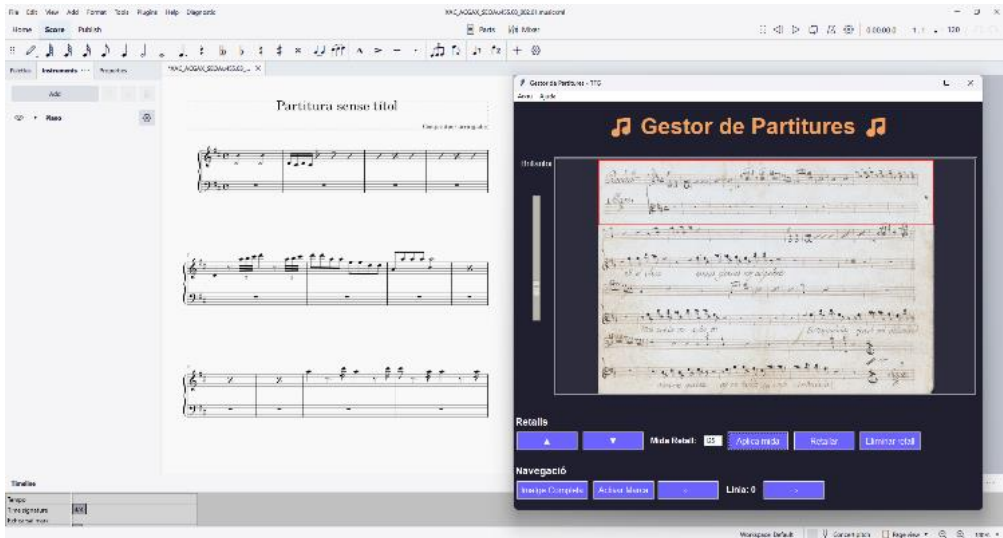
6.3 Arquitectura del sistema

L'arquitectura del projecte és simple i es basa en l'ús de Python per al processament de dades i SQLite com a sistema de base de dades. Els component principal és l'aplicació. Implementada íntegrament en Python, on incorpora llibreries com 'music21' o 'os' per al processament de MusicXML. Gestiona totes les funcionalitats, des de la càrrega i edició de partitures fins a l'emmagatzematge i recuperació de dades. Inclou eines per corregir errors de transcripció detectats en els arxius MusicXML generats per l'OMR. L'altre component principal és la Base de dades (SQLite). Aquesta emmagatzema les partitures editades i les seves versions. Proporciona un historial de modificacions per a cada fitxer.

Aquest enfocament elimina la necessitat d'una separació client-servidor, simplificant el desplegament i manteniment del sistema. Cosa que ens segueix sent viable al únicament voler treballar en local al ordinador del transcriptor.

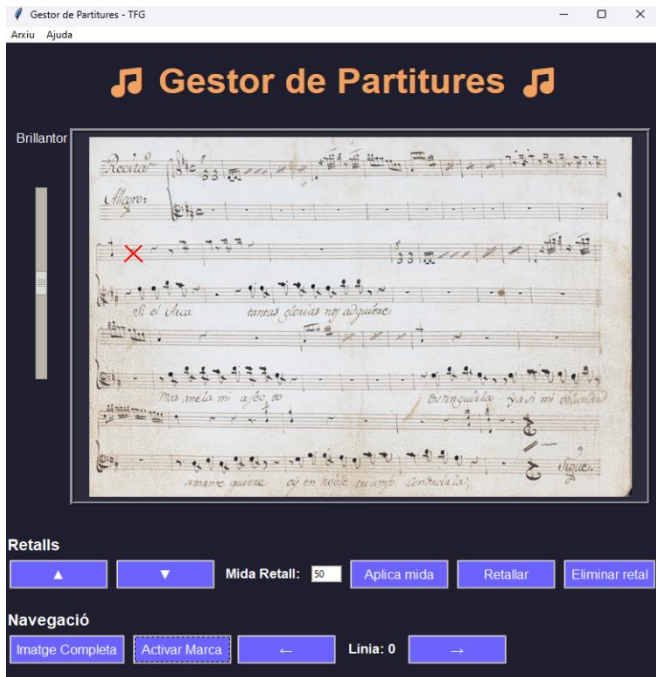
6.4 Funcionalitats

A continuació es presenten els resultats a través de la cinquena il·lustració, on es mostra el prototip de l'aplicació. Aquest prototip representa l'estat més recent del desenvolupament i és el resultat de la implementació de totes les funcionalitats proposades en els objectius inicials. L'aplicació ha estat dissenyada amb un enfocament molt detallat per tal de complir amb les expectatives establertes en la fase inicial del projecte. En la quarta il·lustració es poden observar les diferents seccions de la interfície d'usuari, les quals han estat curosament dissenyades per garantir una experiència d'usuari òptima.



Il·lustració 5: Pantalla de l'aplicació desenvolupada integrada al MuseScore

Es procedirà ara a detallar cada un dels mètodes utilitzats en el desenvolupament de l'aplicació, oferint una explicació i les seves respectives aportacions al projecte. Cada mètode ha tingut un impacte directe en la creació d'un prototip robust i adaptable, amb un alt grau de personalització i flexibilitat per al futur.



Il·lustració 6: Menú principal

Obrir partitura: El botó "Obrir" és una de les funcionalitats més essencials de l'aplicació, ja que permet als usuaris carregar i visualitzar imatges de partitures originals en el seu dispositiu. En accionar aquest botó, l'usuari té la capacitat de seleccionar una imatge de partitura des de la seva galeria o arxius personals, que després serà carregada a l'aplicació per ser visualitzada. Aquesta acció no només obre la partitura original en format imatge, sinó que també detecta i obra automàticament una visualització simultània

de les seves equivalències línia per línia en el programa MuseScore 4, de manera que es visualitza com en la quarta il·lustració.

Guardar treball: Aquest botó permet desar la partitura actual, juntament amb la data d'emmagatzematge, a la base de dades del sistema. En prémer-lo, l'aplicació assegura que la informació es guardi de manera segura i eficient, evitant la pèrdua de dades i garantint-ne l'accés posterior.

Botó BD: Aquest botó ofereix una funcionalitat completa per interactuar amb la base de dades de les partitures. En prémer-lo, es mostra una llista detallada amb les següents dades:

- ID: Identificador únic de cada partitura.
- Títol: Nom o títol assignat a la partitura, facilitant la seva identificació.
- Dates: Específicament, la data d'inici i la data de finalització del treball associat a cada partitura.

A més de la visualització, aquesta funcionalitat és interactiva: permet seleccionar qualsevol entrada de la llista fent clic sobre ella. En fer-ho, l'aplicació carrega automàticament la partitura corresponent per poder visualitzar-la i, si cal, continuar treballant-hi.

Primer pla: Aquest botó ofereix una funcionalitat fonamental per millorar la concentració i adaptar l'espai de treball a les necessitats de l'usuari. Permet alternar entre dos modes de visualització segons el moment. Quan l'usuari selecciona la vista única del MuseScore 4, l'aplicació només mostra la interfície d'aquest programa, amagat qualsevol altre element o finestra, cosa que permet treballar exclusivament amb el programa de partitures i sense distraccions. Això optimitza l'atenció en l'edició musical. D'altra banda,

quan es desactiva aquesta opció, es torna a la visualització completa de l'aplicació, amb tots els elements visibles,

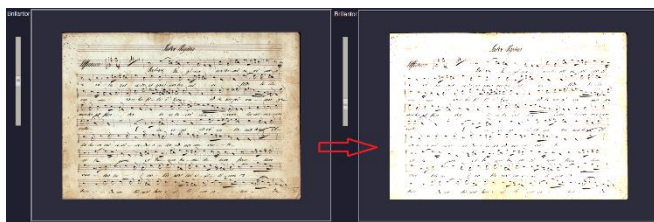
incloent-hi els botons i les funcionalitats addicionals. En aquest sentit, el botó funciona com una eina d'adaptabilitat que dona a l'usuari el control total sobre la configuració visual de l'entorn de treball, permetent-li personalitzar-lo segons les seves preferències i les necessitats en cada moment.

Veure Base de dades: El botó proporciona a l'usuari l'accés a una nova finestra on pot consultar l'històric complet de les partitures sobre les quals s'ha treballat. Aquesta secció inclou, com a funcionalitat addicional, la ruta d'emmagatzematge de la partitura original, el seu nom complet, un identificador únic (ID), així com les dates d'edició per permetre un control detallat de les versions de les transcripcions realitzades. A més, es disposa d'una eina de cerca per nom, situada a la part superior, que facilita la localització i obertura de les partitures directament des de l'aplicació, evitant la necessitat d'accedir a l'explorador d'arxius del sistema operatiu.

Barra de brillantor: Aquesta barra ajustable ofereix una eina indispensable per a la visualització òptima de les partitures carregades, especialment útil en aquells casos en què es treballa amb documents antics o deteriorats. La seva funcionalitat permet modificar el nivell de brillantor de les imatges de les partitures, oferint els avantatges com:

- Millora de la visibilitat: En augmentar la brillantor ja que les parts clares de la imatge es destaquen més, facilitant la identificació de notes i altres elements de la partitura.
- Reducció d'interferències visuals: Quan les partitures presenten taques, soroll o zones fosques a causa del pas del temps o l'escaneig, l'ajust del brillantor ajuda a minimitzar aquests defectes visuals.
- Millora de les precisions en l'anàlisi garantint que fins i tot els detalls més subtils de la partitura es puguin identificar correctament, millorant l'eficiència i l'efectivitat del treball de l'usuari.

Amb aquesta barra, l'aplicació s'assegura de ser una eina versàtil i adaptada a diversos contextos, on cada partitura pot necessitar un ajust diferent per a la seva correcta visualització i edició, com podria ser en el cas de la sisena il·lustració.



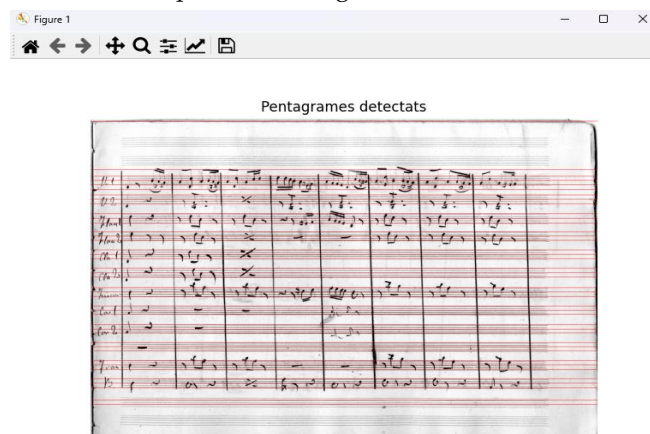
Il·lustració 7: Exemple d'eliminació de taques dubtoses amb ús de la barra brillantor

Retalls: El botó "Retalls" és una funcionalitat avançada dissenyada per adaptar-se a les necessitats específiques dels usuaris que treballen amb partitures complexes, com ara

partitures d'orquestra o altres formats. Aquesta eina permet:

- Definició de mides de línies: L'usuari pot indicar la mida de les línies o grups de línies de la partitura que es volen separar. Això resulta especialment útil per adaptar-se a les particularitats dels diferents tipus de partitures, com ara les d'orquestra, que solen tenir múltiples línies, o les de piano, que solen ser més compactes.
- Emmagatzematge automàtic dels retalls: Cada secció definida es desa automàticament com un retall independent. Això facilita el treball segmentat, ja que cada part de la partitura pot ser analitzada de manera individual.
- Visualització individual: Els retalls emmagatzemats es poden visualitzar de manera independent, amb funcionalitats addicionals com el zoom i el drag per explorar detalls específics de cada secció. Aquesta interactivitat permet una anàlisi més precisa i còmoda. També pot modificar-se la brillantor de cada retall de manera individual, sense afectar a la partitura general, cosa que pot ser de gran ajuda en petites taques focalitzades per l'usuari.
- Sincronització amb MuseScore: Mentre l'usuari es desplaça entre els diferents retalls, MuseScore actualitza automàticament la seva visualització per mostrar l'equivalent exacte de la secció seleccionada dins de la partitura completa.

Abans que l'eina de retalls arribés a la seva versió actual, es va intentar desenvolupar de manera automatitzada mitjançant una detecció amb intel·ligència artificial [3]. Aquesta tecnologia pretenia identificar les diferents parts de les antigues partitures, però, a causa de la complexitat de les formes i els detalls d'aquestes partitures, el sistema no va aconseguir el rendiment esperat. Com es pot observar en la il·lustració 7, la detecció automàtica no ha estat completament precisa, mostrant limitacions en la capacitat de processar adequadament les variacions i els elements intricats de les partitures antigues.



Il·lustració 8: Exemple de la detecció de pentagrames automatitzat

Marca: El botó "Marca" és una funcionalitat clau que permet col·locar una marca visual en qualsevol punt de la partitura per tal d'ajudar l'usuari a mantenir el seguiment durant el procés de transcripció o revisió. Aquesta eina està pensada per evitar que el transcriptor es perdi mentre revisa les correccions o treballa en seccions específiques de la partitura. En prémer aquest botó, l'usuari pot situar una marca al punt desitjat de la partitura, que serveix com a referència immediata per identificar la secció en què està treballant. A més, aquesta marca és totalment interactiva, ja que pot ser desplaçada amb precisió utilitzant les fletxes del teclat, cosa que facilita ajustar la seva posició sense necessitat d'interrompre el flux de treball. Aquesta funcionalitat resulta especialment útil en projectes llargs o complexos, on és fàcil desorientar-se entre diverses seccions, garantint així una experiència més organitzada durant l'edició i la revisió.

Anàlisi estadístic: L'anàlisi estadístic es realitza mitjançant un notebook de Python, amb l'objectiu de comparar els resultats obtinguts pel sistema OMR amb les partitures ja corregides manualment. Aquesta comparació permet extreure informació rellevant sobre la precisió i els errors del sistema OMR, identificant patrons en les equivocacions i proposant possibles millores.

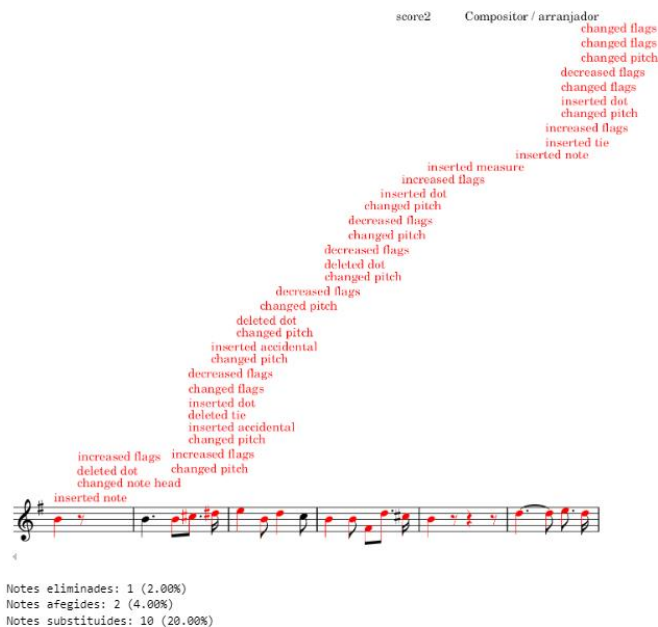
En concret, l'anàlisi permet calcular el percentatge de notes eliminades, afegides i substituïdes, oferint una visió detallada sobre l'eficàcia del sistema OMR i ajudant a identificar les possibles fonts d'error. Aquests resultats es poden visualitzar en forma de taules i gràfics, facilitant la interpretació i l'extracció de conclusions. Per exemple, si es detecta que un OMR té un percentatge especialment alt de notes afegides en comparació amb altres sistemes, això pot indicar un problema específic en la segmentació de la partitura o en la detecció de símbols musicals.

Cal destacar que, degut a que aquesta implementació s'ha portat a terme al final del projecte, no es troba del tot integrada dins l'aplicació per evitar possibles errors o conflictes de compatibilitat amb altres funcionalitats. Per aquest motiu, l'anàlisi es manté com una eina separada.

A continuació, veiem com partint de dues partitures del mateix instrument o tipus (il·lustració 8), podem obtenir tots els canvis ressaltats i els tres percentatges que indiquen les diferències (il·lustració 9). Això és de gran utilitat, ja que el sistema està dissenyat per a implementar-se amb diferents possibles OMRs. D'aquesta manera, es pot dur a terme un estudi comparatiu de quin OMR pot ser el més adequat en funció del tipus de partitures analitzades i el context d'ús. Aquesta comparació ajuda a escollir el sistema que millor s'adapti a les necessitats dels usuaris, assegurant un millor rendiment en la transcripció i validació de partitures digitals.



Il·lustració 9: Partitures d'exemple



Il·lustració 10: Output de l'eina estadística

7 TESTS

Per tal de garantir la usabilitat i eficàcia de l'aplicació desenvolupada, s'han realitzat tests unitaris, tests d'integració i tests d'usuaris amb la col·laboració d'un musicòleg. L'objectiu principal d'aquests tests és analitzar com es desenvolupa l'usuari en l'ús de l'eina, identificant tant els punts forts com aquells aspectes que podrien ser millorats per a una experiència més fluida i satisfactòria. Durant les proves, el musicòleg ha treballat amb l'aplicació realitzant tasques específiques com la càrrega i visualització de partitures, la simulació d'identificació i correcció d'errors, i la utilització de les diferents funcionalitats disponibles, com ara el botó de retalls, la barra de brillantor i les marques. Aquest enfocament pràctic no només ha permès obtenir un feedback directe sobre el funcionament de les eines, sinó que també ha ajudat a avaluar si l'aplicació compleix amb les necessitats reals dels usuaris finals en el context de l'edició i la transcripció de partitures musicals. La principal retroalimentació que s'ha obtingut gràcies a aquesta prova junt amb els tests desenvolupats per part pròpia ha sigut la següent:

- **Obertura general:** S'ha detectat una falla en el procés d'obertura automàtica de MuseScore en fer clic sobre les partitures, on l'aplicació no localitzava correctament el fitxer executables de MuseScore. Aquesta incidència impedia que les partitures s'obrissin de manera automàtica en l'entorn de l'aplicació, afectant la fluïdesa de l'usuari en la seva experiència.

Per solucionar aquesta situació, s'ha afegit un pas extra al procés de configuració. En lloc d'intentar detectar automàticament la ubicació de MuseScore, s'ha implementat una solució personalitzada que permet als usuaris indicar manualment la ubicació exacta del seu arxiu executables de MuseScore només una vegada. Aquest procés es fa a través d'un arxiu de configuració personalitzat per a cada usuari, on s'haurà de proporcionar el camí complet fins al fitxer executables de MuseScore al sistema. Un cop configurat, l'aplicació farà ús d'aquesta informació per obrir les partitures correctament cada vegada que l'usuari faci clic sobre elles, sense necessitat d'introduir aquesta configuració novament.

Aquesta solució ha estat testada amb èxit i ha permès restablir la funcionalitat original de manera estable, millorant l'experiència de l'usuari en obrir partitures de manera més àgil i sense interrupcions.

- **Marca errònea:** Un dels errors trobats es referia a la funció de marca per no perdre el punt de treball en la partitura. Aquesta marca no es desava a la base de dades, fet que causava una pèrdua de la informació sobre el punt on es va interrompre el treball. Com a conseqüència, l'usuari es veia obligat a tornar a posar la marca quan tornava a obrir la partitura en una sessió posterior, cosa que feia que la feina feta en sessions anteriors fos menys útil i ràpida. Aquesta manca de persistència de la marca dificultava l'ús continuat de l'aplicació, especialment quan els usuaris treballaven en diferents hores o dies. Gràcies al musicòleg s'ha corregit i es queda fixada on toca per a cada partitura treballada.
- **Emmagatzematge dels retalls:** Relacionat amb el punt anterior, no es va suposar que una pàgina es pogués deixar a mitja transcripció, per tant, l'error identificat va ser el fet que els retalls realitzats a la partitura no es guardessin a la base de dades. Això suposava una gran limitació per a l'usuari, ja que els retalls, que eren elements fonamentals per a l'edició de la partitura, no quedaven emmagatzemats ni es podien recuperar en sessions posteriors. Aquesta pèrdua de dades feia que la funcionalitat d'edició fos poc fiable i que els usuaris no poguessin continuar treballant sobre els retalls realitzats en sessions anteriors, afectant greument l'eficàcia de l'eina. Finalment s'ha implementat de nou i ja tenim els retalls creats de zero o sobreescrits tots a base de dades en cas que el transcriptor ho desitgi. Fent que el musicòleg pugui incorporar-se de nou on ho havia deixat.
- **Retall seqüencial:** Es va identificar un petit problema de seqüencialitat en el procés de crear i desfer retalls. Això feia que l'eina fos difícil d'utilitzar, ja que els usuaris no podien realitzar accions d'edició de forma flexible i ordenada, fent així un procés massa rupestre. El fet que no es pogués garantir l'ordre correcte en la creació i anul·lació dels retalls feia que la funcionalitat fos

poc intuïtiva, amb un comportament erràtic que complicava la tasca d'edició i que resultava més feina per als usuaris de la que els hi treia.

Al veure-ho, s'ha redissenyat l'eina per a que cada creació o eliminació de retalls permeti indicar manualment sobre quin pas del diccionari s'està fent, eliminant així la pèrdua de temps d'haver de tornar enrere fins a on hi havia el error.

8 IMPACTE DE L'APLICACIÓ EN LA INVESTIGACIÓ MUSICOLÒGICA

La implementació d'aquesta app pot tenir un impacte bastant important en el camp de la investigació musicològica, ja que facilita l'accés a partitures digitalitzades i millora la qualitat de les transcripcions. Gràcies a la seva capacitat de correcció d'errors de sistemes OMR, l'eina permet una representació més fidel de les partitures originals, evitant distorsions que poden afectar l'anàlisi musicològic.

Un altre avantatge clau és la capacitat de l'aplicació per gestionar grans volums de dades, emmagatzemant diferents versions d'una mateixa partitura i permetent-ne la consulta i modificació posterior. Això possibilita la creació d'arxius digitals col·laboratius on musicòlegs de diferents institucions poden contribuir a l'anàlisi i correcció de transcripcions.

Finalment, aquesta eina també contribueix a la preservació del patrimoni musical, ja que permet digitalitzar i corregir partitures que, en molts casos, es troben en risc de deteriorament. D'aquesta manera, es facilita la seva conservació i estudi a llarg termini, assegurant que les futures generacions puguin accedir a aquests documents en formats digitals de qualitat.

9 CONCLUSIÓ

Aquest projecte ha desenvolupat una aplicació capaç de validar i corregir partitures musicals digitalitzades mitjançant OMR, proporcionant una eina intuïtiva i eficient per als usuaris. El sistema final ha estat provat per un musicòleg, el qual ha validat la funcionalitat i usabilitat amb casos de partitures reals. Els resultats després dels tests han demostrat que l'aplicació és útil en la correcció d'errors de transcripció i en la visualització detallada de partitures. A més, les funcionalitats de comparació (estadística) han facilitat la detecció d'errors de l'OMR, permetent ajustar el sistema per millorar-ne la precisió o elecció d'aquest.

L'aplicació final desenvolupada inclou diverses funcionalitats com la càrrega i visualització de partitures, comparació de versions, edició manual de notes, ajust de brillantor per a una millor llegibilitat, identificació i marcatge de pentagrames, anàlisi estadístic i integració amb MuseScore. Aquest conjunt de funcionalitats permet als usuaris treballar de manera eficient amb les partitures digitalitzades, millorant la precisió i facilitant la seva validació i correcció.

Tot i això, encara hi ha aspectes pendents de millora, com

la integració completa de l'anàlisi estadístic dins l'aplicació, així com la incorporació de nous mètodes d'intel·ligència artificial per a l'optimització del reconeixement de partitures. En futures versions, es preveu expandir les funcionalitats del sistema per oferir una experiència encara més precisa i adaptada a les necessitats dels usuaris. Amés, es pot plantejar possibles evolucions amb plugins del propi MuseScore, els quals s'implementen amb QML, llenguatge del qual es van fer proves inicials però no va ser viable plantejar-ho per temps. [9]

En definitiva, aquest projecte ha establert una base sòlida per a la correcció i validació de partitures digitals, obrint noves possibilitats en la recerca i aplicació d'OMR en l'àmbit de la digitalització musical.

AGRAÏMENTS

Vull expressar el meu agraïment a la tutora Alicia Fornés Bisquerra, pel seu suport, la seva orientació i els seus consells valuosos durant tot el procés de desenvolupament del treball. La seva ajuda i consells han estat fonamentals per superar els problemes que han sorgit al llarg del projecte. També vull agrair al Pau Torras per haver assistit en múltiples reunions i seguiments, aportant els seus coneixements i suggerències que han estat molt valuoses per millorar diversos aspectes del projecte. Finalment, agraeixo al musicòleg Carles Badal per la seva participació en els tests i la seva ajuda proporcionant feedbacks constructius. Les seves aportacions han estat clau per detectar millores i per assegurar que l'aplicació compleixi amb els requisits necessaris.

BIBLIOGRAFIA

- [1] Documentació MusicXML:
<https://www.musicxml.com>
- [2] Documentació OpenSheetMusicDisplay:
<https://opensheetmusicdisplay.github.io/>
- Documentació OMRs:
- [3] Mendeley. (n.d.). A Comprehensive Study on the Applications of Machine Learning in Data Science. Mendeley.
<https://www.mendeley.com/catalogue/2f6f6185-ed95-3cb0-946e-d35f201cf30d/>
- [4] IEEE. (2023). A Study on the Challenges of Implementing Agile Practices in Large-Scale Software Development Projects. IEEE Xplore.
<https://ieeexplore.ieee.org/document/10113018>
- Documentació MuseScore:
- [5] <https://musescore.org/es>
- [6] <https://www.youtube.com/watch?v=s7TrqV2Vfs&list=PLSfpTs1MnKvsc3zEpOef5pc7ji-2CIV4n>
- [7] Documentació Metodologies:
- Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
- [8] Guia HTML:
<https://www.uv.es/jac/guia/index.html>
- [9] Documentació llenguatge propi MS4:
https://wiki.qt.io/Getting_Started_Programming_with_QML/es-AR
- [10] Documentació diagrames de secuencia:
https://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso
- [11] Documentació sqlite:
<https://docs.python.org/3/library/sqlite3.html>
- [12] Documentació diagrames de secuencia:

https://es.wikipedia.org/wiki/Diagrama_de_secuencia

[13] Documentació llibreria tkinter:

<https://docs.python.org/3/library/tk.html>

[14] Documentació per treballar dins el codi python amb musicxmls i mscz:

<https://www.music21.org/music21docs/>

[15] Documentació MIDI:

<https://es.wikipedia.org/wiki/MIDI>

[16] Documentació MEI:

<https://music-encoding.org/>

[17] Documentació Verovio:

<https://www.verovio.org/index.xhtml>

[18] Torras, P. (2024). A Unified Representation Framework for the Evaluation of Optical Music Recognition Systems.

<https://link.springer.com/article/10.1007/s10032-024-00485-8>

[19] Calvo-Zaragoza, J., Hajič, J., Jr., Pacha, A.: Understanding optical music recognition. ACM Comput. Surv. 53(4), 1–35 (2021).

<https://doi.org/10.1145/3397499>

[20] Rebelo, A., Fujinaga, I., Paszkiewicz, F., et al.: Optical music recognition: state-of-the-art and open issues. Int. J. Multimed. Inf. Retr. 1(3), 173–190 (2012). <https://doi.org/10.1007/s13735-012-0004-6>.

<http://link.springer.com/10.1007/s13735-012-0004-6>

[21] Pacha, A.: Advancing OMR as a community: best practices for reproducible research. In: Calvo-Zaragoza, J., Hajič, Jr. J., Pacha, A. (eds) 1st International Workshop on Reading Music Systems, Paris, France, pp. 19–20 (2018).

<https://sites.google.com/view/worms2018/proceedings>

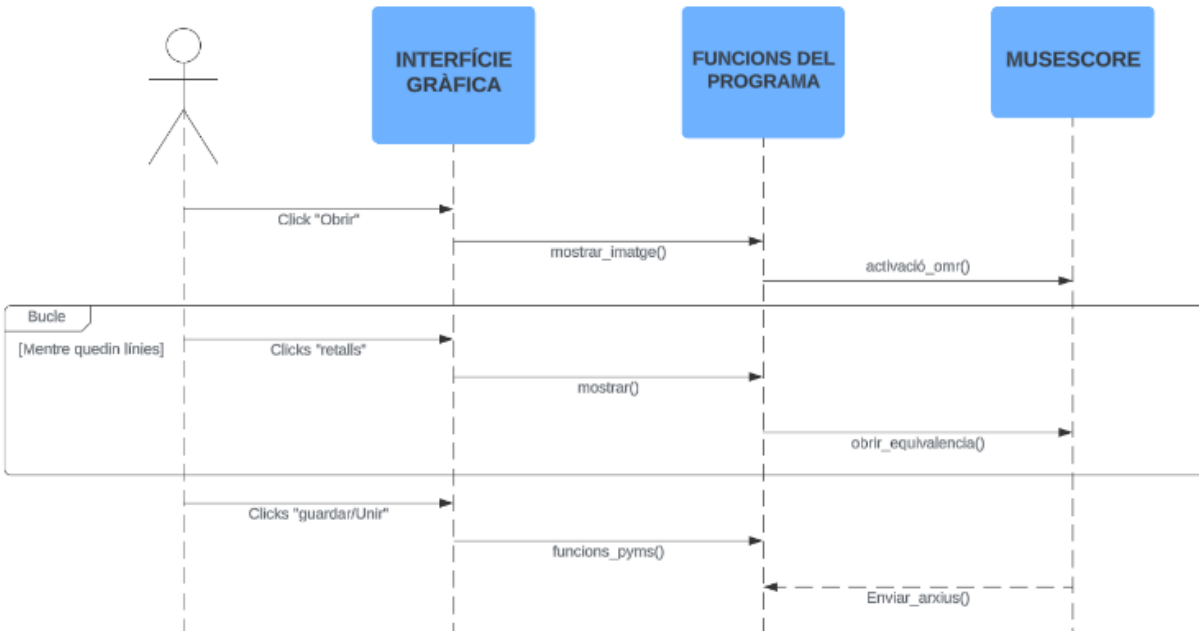
APÈNDIX

A1. Diagrama de classes



Il·lustració 10: Diagrama de classes

A2. Diagrama de seqüència ampliati



Il·lustració 11: Diagrama de seqüència visible amb zoom