



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MUSAPP:
PLATAFORMA DE EVALUACIÓN VIRTUAL ARTÍSTICA A NIVEL CREATIVO Y
APRECIATIVO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

VICENTE GABRIEL DÍAZ GALLEGUILLOS

PROFESOR GUÍA:
JÉRÉMY BARBAY

MIEMBROS DE LA COMISIÓN:
ADOLFO CARRASCO ACOSTA
CLAUDIO GUTIÉRREZ GALLARDO

SANTIAGO DE CHILE
2023

Resumen

Durante la pandemia, la docencia en línea se volvió primordial y evidenció la falta de evolución tecnológica de las evaluaciones artísticas [16] [17] . Si bien las evaluaciones estándar que permiten selección de alternativas y preguntas de desarrollo se pueden realizar virtualmente por medio de plataformas como Google Forms, Kahoot o Edulastic, asignaturas como música y artes visuales donde los ejes centrales educativos son la apreciación y creación, necesitan más herramientas de evaluación para poder corroborar los conocimientos de los estudiantes en dichas áreas. Aunque el confinamiento terminó y los estudiantes volvieron a las aulas, esta falta de instrumentos evaluativos y el estancamiento tecnológico de las mediciones de aptitud artísticas se mantiene. ¿Sería posible generar evaluaciones virtuales que permitan a los docentes medir las capacidades del estudiante en los ejes de apreciar y crear? Para corroborar esto, se implementó un prototipo de ecosistema educativo llamado MusApp [10] en donde los profesores tienen la capacidad de crear evaluaciones con preguntas enfocadas a los ejes de creación y apreciación, para luego corregirlas logrando ver y oír las respuestas de sus alumnos. Los estudiantes pueden responder dichas evaluaciones con distintas herramientas, tales como pianos virtuales, sistemas de escritura y reproducción de partituras, sumado a sistemas de respuesta que manejan otras aplicaciones estándar: verdadero y falso, selección de alternativas y desarrollo escrito. Adicionalmente, se implementó un sistema administrativo con Niveles, Cursos y Asignaturas para facilitar tanto a docentes como al alumnado el desarrollo de esta actividad académica virtual.

Se evaluó con profesores, alumnos de pedagogía en música y alumnos de los niveles de séptimo y octavo básico para analizar a partir de pruebas de usabilidad moderadas, el comportamiento de los usuarios al momento de utilizar la plataforma, consiguiendo resultados positivos, pues en términos de usabilidad y funcionalidad el prototipo de ecosistema logra generar instrumentos de evaluación que encapsulan completamente el eje creativo y de manera parcial el eje apreciativo.

Para mi familia docente

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia por su enorme apoyo y grandes enseñanzas, que me han brindado y siguen entregando hasta el día de hoy. A Claudio mi padre, por presentarme la música desde mi niñez, enseñándome a tocar guitarra, y luego apoyándome a cada taller musical al que me integraba, siempre preocupado de que fuera responsable y haciéndose partícipe en mi educación. A mi madre Mónica, pues en ella conocí a una mujer muy amable, cuidadosa del medio ambiente, comprometida con su rol de educadora, que siempre me enseñó una manera distinta de ver las cosas, buenos valores, y un cariño incondicional que me ha acogido en mis buenos y malos momentos. También a mi hermano Nicanor, que, con su ejemplo de vivir una vida simple, rodeada de naturaleza, enfocada a conseguir sus metas y a hacer lo que le gusta, me mantiene presente que la vida no es tan grave, que las cosas siempre se pueden afrontar con una sonrisa y esfuerzo. A mi tía Angélica y mis primas Xaviera, Catalina Tamara, por darme un hogar, mucho cariño, fines de semanas de tranquilidad y entretenidas conversaciones en el patio.

Agradecer a mi pareja y mejor amiga Faina, por todo su apoyo, risas y amor que me ha entregado, por nuestros viajes y conciertos, nuestras noches de anime y videojuegos, por estar presente en mis mejores y peores momentos. Creo que la vida no me va a alcanzar para darle todo el cariño y apapachos que se merece.

A mi profesor guía Jérémy Barbay, con el que he formado una simpática amistad a lo largo de mis años universitarios, una persona que siempre ha tenido plena confianza en mi trabajo y mis ideas, y me alienta a avanzar y a aprender, agradezco mucho su enseñanza y me alegra ver como siempre intenta innovar e ingeniárselas para mejorar la forma de educar.

A Nelson Rosas, quien tomó un rol paternal en mi educación musical, un profesor a quien le tengo mucho cariño y me acompañó por cinco años en la banda instrumental del liceo Narciso Tondreau, taller donde conocí a grandes amigos como Mauricio, Tomás, Félix, Esteban y Benjamín, junto a quienes pasé mis momentos más entretenidos de mi época liceana.

Y finalmente agradecer a mis amigos del Hogar Paulina Starr, cómo dejarlos de lado si fueron mi familia durante toda mi época universitaria. Gracias Diego, Manjulo, Matías, Patito y Jorge por ser tan buenos amigos, por las tardes de Smash Bros y Brawl Stars, las juntas en el sótano, las noches arriba del techo o en las salas de estudio hablando de la vida, por ser todos a su manera tan trombas y garkas.

Tabla de Contenido

1. Introducción	1
1.1. Objetivos	3
1.1.1. Objetivos Generales	3
1.1.2. Objetivos Específicos	3
2. Estado del Arte	5
2.1. Plataformas de evaluación virtual artística	5
2.1.1. Soluciones existentes para la evaluación digital	6
2.1.2. Características de interés para la evaluación digital artística	6
2.1.3. Visión Global	7
3. Diseño	8
3.1. Ambiente y plataforma	8
3.2. Sistema de Sonido	9
3.2.1. Sistema de almacenamiento y reproducción de audios	9
3.2.2. Almacenamiento de audios en la base de datos	11
3.2.3. Diseño de la clase nota	11
3.3. Herramientas evaluativas de creación musical	12
3.3.1. Piano Virtual	12
3.3.2. Sistema de partitura	13
3.4. Modelo administrativo y de usuarios	16
3.5. Sistema de Evaluaciones	17

3.6.	Interfaz	18
3.6.1.	Pantallas de Estudiante	19
3.6.2.	Pantallas de Docente	20
4.	Implementación	21
4.1.	Ambiente y plataforma	21
4.2.	Sistema de Sonido	22
4.2.1.	Sistema de Almacenamiento de Audios	22
4.2.2.	Implementación de la clase Nota	23
4.3.	Implementación de herramientas evaluativas de creación musical	24
4.3.1.	Piano virtual	24
4.3.2.	Implementación Sistema de partituras	27
4.4.	Modelo administrativo y de usuarios	31
4.5.	Sistema de evaluaciones	32
4.6.	Interfaz	32
4.6.1.	Inicio de sesión	32
4.6.2.	Estudiante	33
4.6.3.	Profesor	35
4.6.4.	Administrador	38
5.	Validación	40
5.1.	Validación del piano virtual	40
5.1.1.	Validación de Piano Virtual: Alumnos	40
5.1.2.	Validación de Piano Virtual: Estudiantes de Pedagogía y Profesores	43
5.2.	Validación de evaluaciones para estudiantes	46
5.3.	Validación para creación y revisión de evaluaciones por profesores	49
5.4.	Validaciones de rendimiento, uso de recursos computacionales y envío de datos	51
5.4.1.	Recursos del computacionales	51

5.4.2. Envío de Datos	51
5.5. Comparación con los objetivos iniciales	52
5.5.1. Diseños	52
5.5.2. Implementación	53
6. Conclusión	55
6.1. Logros	55
6.2. Aprendizajes	56
6.3. Trabajo Futuro	56
6.3.1. Almacenamiento de piezas musicales	57
6.3.2. Portabilidad	57
6.3.3. Inserción de recursos audiovisuales	57
6.3.4. Inserción de un metrónomo al piano virtual	57
6.3.5. Agregar el piano virtual y el sistema de partituras dentro de una pregunta	58
6.3.6. Entregar una retroalimentación dentro de la plataforma	58
6.3.7. Sistema pixel-Art	58
6.3.8. Sistema de ecuaciones y gráficos	58
6.3.9. Hardware de Piano con Arduino	58
6.3.10. Enseñar a trabajar con una estación de trabajo de audio digital	59
6.3.11. Desarrollo de minijuegos educativos	59
Bibliografía	61
Anexo A. Sistema generador de notas económico	63
Anexo B. Modelo de Base de Datos	65
B.1. Modelo administrativo y de usuarios	65
B.1.1. Nivel (Level)	65
B.1.2. Curso (Grade)	65
B.1.3. Usuarios (User)	66

B.1.4. Asignatura (Subject)	67
B.2. Sistema de Evaluaciones	67
B.2.1. Evaluación (Quiz)	67
B.2.2. Preguntas (Question)	68
B.2.3. Respuesta de una Evaluación (AnsweredQuiz)	69
B.2.4. Respuestas a preguntas (AnsweredQuestion)	70
B.2.5. Diagrama de Relación de Entidades	71
Anexo C. Sistema de partituras	72
C.1. Configuraciones de Armadura	72
Anexo D. Implementación	74
D.1. Frecuencia media de instrumentos	74
D.2. Diagrama de la configuración del piano virtual	75
Anexo E. Validaciones	76
E.1. Validación de evaluaciones para estudiantes	76
E.1.1. Usuario 1	76
E.1.2. Usuario 2	77
E.1.3. Usuario 3	77

Índice de Ilustraciones

3.1. Sistema latino y anglosajón de notas musicales	10
3.2. Diseño de la clase Nota	11
3.3. Diseño del piano virtual	12
3.4. Concepto de uso del piano virtual	13
3.5. Elementos de la configuración inicial	14
3.6. Notas de una partitura, cada nota ocupa intercaladamente una línea o un espacio	14
3.7. Figuras con distancias proporcionales al tiempo de sonido correspondiente . .	15
3.8. Silencios con distancias proporcionales al tiempo de silencio correspondiente	15
3.9. Silencios ordenados de mayor a menor, por ende la Q genera el silencio con mas tiempo (4 tiempos) y la T genera el menor (0.25 tiempos)	16
3.10. Sonidos ordenados de mayor a menor, por ende la A genera el sonido con mas tiempo (4 tiempos) y la T genera el menor (0.25 tiempos)	16
3.11. Mockup Pantalla Inicio del Estudiante	19
3.12. Mockup Pantalla de Evaluación del Estudiante, con una pregunta de Piano Virtual	19
3.13. Mockup Pantalla Inicio del Profesor	20
3.14. Mockup Pantalla de Crear Evaluaciones	20
4.1. Instrumentos importados a Godot	22
4.2. Diccionarios de instrumentos musicales	22
4.3. Implementación de la ecuación (4.1) en Godot	23
4.4. Clase Nota	23
4.5. Visualización del Piano Virtual	24

4.6. Distintas configuraciones de nota inicial	24
4.7. Código para comenzar a grabar	25
4.8. Presionar y liberar una tecla	25
4.9. Código para detener una grabación	26
4.10. Código para reproducir el piano	26
4.11. Feedback al usuario de las teclas presionadas	27
4.12. Sistema de partituras	27
4.13. Indicador de reproducción	27
4.14. Configuración del sistema de partituras	28
4.15. Cambio de llave	29
4.16. Cambio de Armadura	29
4.17. Dos filas de partitura, en la primera se ve un precompás y tres compases, en la segunda se ve un precompás y solo dos compases	30
4.18. Código para insertar una nota	30
4.19. Código para eliminar una nota	30
4.20. Código para Reproducir una partitura	31
4.21. Inicio de sesión	33
4.22. Alumno: Pantalla principal	33
4.23. Alumno: Pantalla principal, ver notas	34
4.24. Alumno: Responder evaluación, Piano	34
4.25. Alumno: Responder evaluación, Partituras	35
4.26. Docente: Pantalla principal	35
4.27. Docente: Crear evaluación	36
4.28. Docente: Creación de preguntas	36
4.29. Docente: Lista de alumnos que respondieron una evaluación	37
4.30. Docente: Revisión de una pregunta de piano virtual	37
4.31. Docente: Revisión de una pregunta de partitura	38
4.32. Interfaz Administrador: Pantalla Principal	38

4.33. Interfaz Administrador: Editar un Alumno	39
4.34. Interfaz Administrador: Crear Asignatura	39
5.1. Primera versión visual del piano	40
5.2. Encuesta a Alumnos: uso de teclado en el piano virtual	41
5.3. Encuesta a Alumnos: uso de mouse en el piano virtual	41
5.4. Encuesta a Alumnos: Interfaz de nota	42
5.5. Encuesta a Alumnos: Interfaz de octava	42
5.6. Encuesta a Alumnos: Interfaz de instrumento	42
5.7. Encuesta a Alumnos: Facilidad para grabar	43
5.8. Encuesta a Alumnos: Sugerencias	43
5.9. Encuesta a Docentes: uso de teclado en el piano virtual	44
5.10. Encuesta a Docentes: uso de mouse en el piano virtual	44
5.11. Encuesta a Docentes: Interfaz de nota	45
5.12. Encuesta a Docentes: Interfaz de octava	45
5.13. Encuesta a Docentes: Interfaz de instrumento	45
5.14. Encuesta a Docentes: Visualización de las notas en el piano	46
5.15. Versión final del piano	46
5.16. Primera versión visual del sistema de partituras	47
5.17. Versión final del sistema de partituras	49
5.18. Uso en disco de MusApp	51
5.19. Uso de Ram	51
5.20. Pregunta vacía, siendo que algunas pesan un poco más, se deja en promedio 80B	52
5.21. Respuesta vacía, tiene 87 caracteres de 1B cada uno	52
B.1. Relaciones de Entidades del Modelo de Datos	71
D.1. Frecuencia media de instrumentos	74
D.2. Configuración del piano virtual	75

Índice de Tablas

2.1. Tabla de comparación entre plataformas de evaluación online y MUSSAP ✓: Cumple la característica ✗: No cumple la característica *: Se diseñó para MU- SAPP pero no se implementó	7
---	---

Capítulo 1

Introducción

Debido a la pandemia de Covid 2020 - 2021, la docencia en línea se convirtió en la principal forma de enseñanza en Chile y en muchas partes del mundo, tanto en enseñanza básica, media y universitaria. La parte expositiva de una clase resultó posible gracias a plataformas como Zoom, Discord y Meet, en donde el profesor puede dictar una clase y los alumnos pueden hacer preguntas y tener un diálogo. Por otro lado, para la evaluación del aprendizaje del estudiante, los docentes recurrieron a variadas páginas web y aplicaciones que permitieron crear evaluaciones (como un conjunto de preguntas) y enviárselas a los alumnos para que las contestasen virtualmente, pero tratar de examinar algunas materias, sobre todo del ámbito artístico, expuso las limitaciones que presentaban los instrumentos de evaluación virtuales en dicho campo.

Actualmente, con la vuelta a las aulas por parte de estudiantes y docentes, se podría considerar innecesario tener herramientas de evaluación virtual para el ámbito artístico-musical, pero al poner en discusión la desigualdad social existente en el país [14], y la falta de implementaciones requeridas por parte de establecimientos educacionales para la facilitación de elementos sonoros [15], se evidencia la necesidad de tener una amplia paleta de instrumentos educativos que contribuyan al desarrollo artístico de los estudiantes a nivel nacional, puesto que no siempre los alumnos tendrán disponibles materiales tangibles para la realización de las actividades planteadas, como tener un instrumento musical, un metrónomo o materiales para una actividad de dibujo o pintura. Sumado a este planteamiento, políticas públicas como las becas TIC [11], que contribuyen tecnológicamente a superar la brecha económico-social que nuestro país posee ayudando a aproximadamente a 150.000 estudiantes cada año, permite considerar la integración de herramientas computacionales para el apoyo a la enseñanza.

A nivel de educación artística, tanto visual como musical, existen dos ejes educativos centrales [13] presentes en todos los niveles: apreciar y crear. Del eje apreciativo se desprende el análisis crítico de obras, movimientos, géneros y prácticas artísticas, por lo que es requerido que el alumno posea conocimientos teóricos, históricos y culturales, y que sea capaz de emitir juicios críticos, expresivos y estéticos a partir de sus conocimientos. Por otro lado, el eje de creación implica la capacidad expresiva del estudiante, que también requiere de conocimientos teóricos, históricos y culturales, pero adicionalmente, el individuo debe ser capaz de poder plasmar sus emociones, ideas y sentimientos dentro de una obra, crear a partir de movimientos

o género específicos, experimentar paletas de colores, espectros de sonidos, matices, etc. Ambos ejes son cruciales y, por ende, poder medir los conocimientos de los estudiantes en estas dos aristas es importante y necesario.

Para poder medir los conocimientos históricos, teóricos y culturales de los estudiantes, al ser conocimientos objetivos, evaluaciones con preguntas de alternativa y desarrollo son perfectamente válidas y funcionales. Incluso, para el eje apreciativo, las preguntas de desarrollo encapsulan perfectamente la medición de la apreciación y análisis que pueda tener un alumno frente a una obra, debido a que la percepción de un elemento artístico puede ser expresada a través de palabras. Pero para medir el eje de creación, preguntas de desarrollo y selección de alternativas no bastan y es por ello que es necesario entregarles herramientas a los estudiantes que les permitan crear y expresarse.

En vista de los puntos expuestos anteriormente, se considera que un ecosistema educativo virtual con herramientas que faciliten la creación artística, sumado a evaluaciones online tradicionales, permitirá a los docentes medir los conocimientos en los ámbitos apreciativo y creativo de los estudiantes.

En este trabajo, se desarrolló una aplicación de escritorio de un ecosistema para educación musical, en donde alumnos pueden responder preguntas del área creativa. Para ello, se implementaron respuestas con un piano virtual, que los alumnos pueden tocar con el teclado y/o mouse de su computador, y un sistema de escritura y reproducción de partituras. Adicionalmente, pueden responder preguntas de verdadero y falso, selección de alternativas y desarrollo escrito para la evaluación de conocimientos y apreciación. Por otro lado, los docentes pueden crear evaluaciones para sus asignaturas con todas las preguntas descritas anteriormente, y luego revisar las respuestas de sus estudiantes, logrando reproducir lo que el alumno haya tocado en el piano virtual y también lo que haya escrito en la partitura. Junto a esto, el administrador del establecimiento puede crear Niveles, Cursos y Asignaturas; asignar Estudiantes y Docentes a las asignaturas correspondientes y editar sus datos de ser requerido. Todo enmarcado en la concepción de una aplicación que no tenga altos requisitos computacionales, ni consuma grandes recursos de red, para así poder ser usada en todo el territorio nacional y en un gran espectro socio-económico. A nivel de tecnologías utilizadas, este ecosistema fue desarrollado como una aplicación de escritorio en Windows con el motor de videojuegos Godot para el frontend, el conjunto de herramientas Django Rest framework para backend y SQLite para la base de datos.

Para evaluar este ecosistema, fueron consideradas dos cualidades, que son usabilidad y funcionamiento. Se realizaron tres procesos de validación, utilizando una herramienta llamada prueba de usabilidad moderada. Esta consiste en presentarle al usuario el elemento a probar e ir registrando los comportamientos y acciones que realice dicho usuario. Dicha herramienta fue utilizada en una primera instancia para validar el piano virtual con profesores y alumnos (9 participantes), luego se utilizó nuevamente la herramienta para la resolución de una evaluación (3 participantes), y finalmente una última prueba de usabilidad moderada para la creación de evaluación (1 participante). Dichos procesos fueron documentados con encuestas y/o anotaciones durante las validaciones.

Los resultados de usabilidad y funcionamiento se consideran exitosos. En el ámbito estudiantil, los usuarios lograron ingresar al ecosistema y responder una evaluación sin ninguna

explicación previa de uso, aunque sí tuvieron ligeros problemas al escribir partituras, pues ninguno había escrito una partitura antes ni conocían los nombres de algunas teclas de teclado necesarias para escribir, tales como SHIFT, ALT o CTRL. En cuanto a los docentes, no hubo problemas para crear una prueba y luego revisarla, logrando oír lo que usuarios tocaron en el piano virtual y lo que habían escrito en partitura. Al ser una plataforma que valida ser sencilla al momento de su uso, se estima viable para utilizarse como instrumento evaluativo de conocimientos y para el eje educativo de creación, puesto que las herramientas creativas que posee fueron correctamente utilizadas y fácilmente integradas por todos los usuarios. En el eje apreciativo, si bien los alumnos disponían de distintas formas de responder, el profesor solo podía enviar preguntas con texto, sin imágenes, audios, interpretaciones del piano (los alumnos pueden tocar el piano y los profesores pueden reproducir sus grabaciones, pero no al revés) o partituras (mismo caso anterior). Por ende, queda como trabajo futuro lograr implementar un sistema de preguntas más versátil y con mayores recursos audiovisuales.

1.1. Objetivos

1.1.1. Objetivos Generales

El objetivo de este trabajo es diseñar, implementar y validar un prototipo de aplicación con un ecosistema virtual con bajos requerimientos de conectividad y de recursos computacionales, que permita medir el conocimiento de los alumnos de asignaturas musicales, a partir de preguntas creadas por el docente, enfocadas a evaluar los ejes de creación y apreciación.

1.1.2. Objetivos Específicos

Para lograr el objetivo general es necesario conseguir los siguientes hitos.

1. Diseñar las funcionalidades principales para una aplicación de escritorio que:
 - (a) Genere instrumentos de evaluación que encapsulen los ejes de Crear y Apreciar,
 - (b) Permita a los docentes de música: crear, aplicar y corregir dichos instrumentos evaluativos,
 - (c) Permita a estudiantes: responder y recibir retroalimentación de las evaluaciones aplicadas,
 - (d) Permita a los administradores: crear y editar Niveles, Cursos y Asignatura,
 - (e) Sea óptima y no consuma grandes recursos computacionales,
 - (f) Sea simple de utilizar, cómoda y estéticamente agradable para los distintos usuarios.
2. Implementar:
 - (a) Visualizaciones de los instrumentos de evaluación y su interacción con el usuario,

- (b) Un ambiente exclusivo del ecosistema educativo para el uso de la aplicación por parte de los docentes,
 - (c) Un ambiente exclusivo para el uso de la aplicación por parte de los estudiantes,
 - (d) Un ambiente exclusivo para el uso de la aplicación enfocado en los administradores,
 - (e) Un sistema de datos común para todos los usuarios,
 - (f) Optimizaciones al sistema para tener una aplicación que no requiera recursos computacionales elevados.
3. Validar esta aplicación con profesores y estudiantes de artes musicales.

Capítulo 2

Estado del Arte

Las formas de evaluar virtualmente a alumnos de manera artística pueden ser bastantes y variadas entre sí, debido a las posibilidades que presenta internet. Dentro de las opciones tenemos plataformas de evaluación virtual, como Kahoot o Google Forms, plataformas para subir videos y así evaluar los avances de un alumno, presentaciones virtuales en alguna plataforma de conferencias, envío de imágenes, audios o videos por correo electrónico, etc. Todo esto depende del eje que el profesor quiera evaluar y de las herramientas con las que cuenten los alumnos para desarrollar la evaluación o actividad solicitada.

Un punto importante también a evaluar son las condiciones país para integrar una nueva solución evaluativa en los colegios de Chile. Si se toman en cuenta los 2.689.832 estudiantes matriculados a Diciembre de 2021 a lo largo de todo el territorio nacional (enseñanza básica y media), según los datos entregados por el Ministerio de Educación, el Índice de Vulnerabilidad Escolar (IVE [6]) alcanza el 81 %. Con esta realidad presente, es indispensable que la solución propuesta para el problema de las evaluaciones virtuales no presente un costo adicional para los estudiantes, con lo que es importante destacar que no se puede asumir que un alumno tendrá instrumentos musicales en su casa o grabadoras. Solo se considerará como un recurso mínimo para una evaluación virtual que los alumnos cuenten con un dispositivo electrónico (computador, celulares, tablet o laptop) con conexión a internet y, considerando las becas TIC dadas por el gobierno (entrega de laptops a alumnos de séptimo básico), es más factible asegurar que existe una población de alumnos con un dispositivo compatible. Por ende, se descartarán para la solución aplicaciones y plataformas que requieran de un elemento externo.

2.1. Plataformas de evaluación virtual artística

Si bien no existe una plataforma que esté enfocada específicamente a la evaluación digital artística, en donde un profesor crea la evaluación y los alumnos la responden, actualmente existen variadas plataformas enfocadas a la evaluación digital general, algunas de las cuales se presentan en esta sección. Posterior a ello, se entrega un resumen con las características relevantes de cada una de estas aplicaciones y con ello, se pasa a la discusión de por qué es necesaria una nueva solución, comparando las características de las plataformas mostradas

con características y formas de evaluación existentes dentro del contexto aula presencial.

2.1.1. Soluciones existentes para la evaluación digital

Google Forms [5]: Sistema enfocado a la creación de formularios con distintos tipos de preguntas como checklist, selección múltiple, preguntas de desarrollo, entre otras. Tiene la funcionalidad de asignar respuestas correctas y puntajes a cada pregunta, permitiéndole al docente tener un resumen tipo dashboard en el cual puede estudiar visualmente el comportamiento del curso frente a la evaluación respondida.

Kahoot [7]: Plataforma de aprendizaje basada en juegos educativos estilo trivia, en donde docentes crean sus evaluaciones y los alumnos las contestan consiguiendo puntaje. Con ella, es posible hacer preguntas tipo selección de alternativa o preguntas de desarrollo, y el docente tendrá acceso a las respuestas de sus alumnos.

Quizizz [9]: Similar a la plataforma anterior, Quizizz permite desarrollar cuestionarios lúdicos, tanto para público general como para un curso en específico, creado dentro de la misma plataforma. Permite ver las evaluaciones creadas y las respuestas de los alumnos.

Edulastic [2]: Sistema enfocado a la creación de evaluaciones digitales con un gran espectro de tipos de preguntas (alrededor de 30), enfocadas principalmente a las preguntas matemáticas. Permite crear evaluaciones para un curso en particular y corregirlas. Además, posee un banco de evaluaciones creadas por otros docentes al cual se puede acceder.

Easy LMS [1]: Plataforma para crear evaluaciones digitales limitadas a los integrantes ingresados a la evaluación. Permite ingresar audios, videos e imágenes a las preguntas y es posible revisar, como también comentar las evaluaciones de cada alumno.

Quizalize [8]: Otro sistema lúdico de evaluación online que contiene un banco de preguntas creadas por otros usuarios. Permite la creación de grupos a los cuales tomarle las evaluaciones y está enfocado a preguntas de selección de alternativas.

2.1.2. Características de interés para la evaluación digital artística

A continuación se presentan que características deseadas para una plataforma de evaluación online enfocada a la educación artística en el contexto de un establecimiento educativo.

- **Organización:** La plataforma permite que el profesor pueda asociar la evaluación a una Asignatura y un Curso en específico existente en la aplicación, para poder enviar la evaluación y que los alumnos la respondan dentro del mismo sistema.
- **Preguntas multimedia:** Permite que los alumnos puedan subir sus propios archivos con la imagen, video o audio como forma de respuesta a alguna pregunta dentro de una evaluación.

- Preguntas con alternativas: Se pueden crear evaluaciones con preguntas de verdadero y falso, selección de una alternativa y selección múltiple.
- Preguntas de desarrollo: Es posible crear evaluaciones con preguntas en las que el alumno deba contestar escribiendo un texto.
- Preguntas de creación y/o apreciación artística: Permiten que el alumno se exprese y/o cree a partir de elementos visuales o musicales.

2.1.3. Visión Global

A continuación se presenta una tabla que por un lado tiene las soluciones a la educación online, y por otro las características que debe tener un sistema de evaluación digital artístico.

	Organización	Preguntas multimedia	Preguntas de alternativas	Preguntas de desarrollo	Preguntas de creación y apreciación artística
Google Form	✗	✓	✓	✓	✗
Kahoot	✗	✗	✓	✓	✗
Quizizz	✓	✗	✓	✓	✗
Edulastic	✓	✓	✓	✓	✗
Easy LMS	✗	✗	✓	✓	✗
Quizalize	✓	✓	✓	✗	✗
MUSSAP	✓	*	✓	✓	✓

Tabla 2.1: Tabla de comparación entre plataformas de evaluación online y MUSSAP

✓: Cumple la característica

✗: No cumple la característica

*: Se diseñó para MUSAPP pero no se implementó

Como se puede apreciar, existen plataformas como Edulastic que cumplen casi todas las características deseadas, pero esta aplicación junto a todas las otras no tiene la opción de preguntas de creación y expresión visual y/o musical. Si bien se podría argumentar que los ejes de creación y expresión podrían ser evaluados a partir de audios, imágenes o videos que el alumno pueda subir con la característica de Preguntas multimedia, es importante recordar que no todos los alumnos poseen un instrumento musical o los materiales necesarios para hacer una creación para artes visuales en sus casas. Es por ello que resulta importante tener una plataforma que posea las características presentadas en la sección anterior, en particular “Preguntas de creación y expresión visual y/o musical”, en donde el alumno pueda crear y expresarse con herramientas entregadas dentro de la plataforma, debido a que esta área es la que actualmente no está siendo atacada por ninguna aplicación de evaluación virtual enfocada a colegios.

Capítulo 3

Diseño

Al pensar un prototipo de plataforma que usará un público objetivo amplio y diverso, salen a la luz ciertos elementos indispensables que debe poseer su diseño. En primer lugar, debe ser un sistema ordenado, sencillo de entender, y altamente escalable, pues el prototipo no deja de ser un primer molde al cual se le debe ir constantemente haciendo cambios (en este caso enfocados a la usabilidad y funcionalidad), y agregando diversas funcionalidades. En segundo lugar, debe ser sencillo de entender para los usuarios, logrando que puedan desenvolverse por la plataforma sin necesidad de explicaciones y lo sientan cómodo y natural. A continuación se detallarán las distintas características que se idearon para la plataforma, en orden cronológico.

3.1. Ambiente y plataforma

En primer lugar, fue necesario concretar la idea del ecosistema educativo virtual diseñado: qué tipo de aplicación iba a ser, en qué dispositivo se iba a trabajar y con ello, ver qué ventajas y desventajas tendrá al momento de implementar. A continuación se detallan las opciones elegidas y el por qué:

- Dispositivos: Computadores y laptop's. Permiten una amplia visualización debido a sus pantallas de entre 13 y 24 pulgadas en promedio, cuenta con un teclado de membrana o mecánico que facilitará el uso de un piano virtual, pues golpear una tecla es mucho más cercano sensitivamente a tocar un piano real (debido a que se tiene una respuesta táctil de movilidad) que en dispositivos de pantalla táctil, donde dimensionalmente la pantalla es más pequeña y la sensación táctil es tocar un vidrio o lámina. Como desventaja, limita al estudiante a dónde y cuándo podrá responder evaluaciones pues un dispositivo móvil ofrece mayores comodidades de transporte.
- Tipo de plataforma: Aplicación de escritorio. considerando que no en todo Chile la conexión a internet es estable ni ilimitada, una aplicación de escritorio conviene al usuario para evitar un consumo elevado de datos, pues los recursos audiovisuales (que son los elementos más pesados de una plataforma) se cargan de manera única en el

dispositivo al momento de su descarga, y por ende, no es necesario volver a pedir dichos recursos a través de la red cada vez que la aplicación se abre. Además, entrega una mejor estabilidad de uso, debido a que, al ser una aplicación de escritorio, no depende exclusivamente de la conexión a internet, y aprovechando esa ventaja, se puede diseñar una aplicación que necesita conectarse a la red solo en momentos específicos, y capaz de poder utilizarse en periodos offline.

- Sistema operativo: Windows (al menos en una primera etapa). Esto principalmente porque es el sistema operativo más popular en Chile y todos los computadores entregados por el gobierno lo utilizan.

3.2. Sistema de Sonido

El diseño del sistema de sonido es indispensable al momento de pensar en el sistema musical, y teniendo presente una plataforma de escritorio, se muestra a continuación como se ideó el almacenamiento de audios, su almacenamiento en la base de datos y el diseño de la clase Nota.

3.2.1. Sistema de almacenamiento y reproducción de audios

Para poder tener un banco de sonidos de distintos instrumentos y por cada uno de ellos una variedad de notas, se ideó utilizar un archivo en formato .wav (Waveform Audio File) por instrumento, todos en la frecuencia de 440Hz, que equivale a tocar la nota A4 (Notación anglosajona para La4), y a partir de ella junto con la modificación de su frecuencia, poder obtener el resto de notas de un mismo instrumento. Esto supuso un tamaño de 1.34MB aproximadamente por instrumento, en comparación a los aproximadamente 118MB que equivaldría tener todas las notas de un instrumento guardadas (Ver Anexo A). La siguiente ecuación permite calcular la frecuencia de una nota, sabiendo la distancia de semitonos que hay entre la nota deseada y A4.

$$F_{nota} = 440r^d \quad (3.1)$$

Donde F_{nota} es la frecuencia de la nota que queremos tocar, r es una constante igual a $12\sqrt[12]{2}$ y d es la distancia en semitonos de la nota La4

Pero para saber la distancia d es necesario construir una segunda ecuación.

Convertor de nombre de nota a distancia en semitonos

La ecuación anterior, se utilizó para diseñar un sistema de conversión de nombre de notas a frecuencias, para así, al querer tocar un Mi3, la frecuencia de la nota se calcule

automáticamente, sin la necesidad de contar distancias en semitonos a mano.

En primer lugar, es importante saber que una nota se compone de 3 elementos: un Nombre N, una Alteración (opcional) A y un índice de octava I. Se utilizarán estos elementos para construir un conversor de estos tres parámetros en una frecuencia.

Para el nombre de las notas se utilizó el sistema anglosajón pues, a diferencia del sistema latino, solo utiliza una letra por nota, y van consecutivas, por lo tanto, es iterable y facilitará muchos diseños e implementaciones futuras (tener en cuenta que parte en C y que después de G, se vuelve a A). En la Figura 3.1 se muestra la equivalencia del sistema latino y anglosajón.

Do Re Mi Fa Sol La Si Do
C D E F G A B C

Figura 3.1: Sistema latino y anglosajón de notas musicales

Al saber que entre cada nota hay dos semitonos, exceptuando E-F y B-C donde solo hay uno, se puede construir el siguiente diccionario, que represente la distancia en semitonos desde C al resto de notas:

$$Notas = \{C : 0, D : 2, E : 4, F : 5, G : 7, A : 9, B : 11\}$$

Adicionalmente, una nota puede contar con una Alteración A, que equivale a desplazar una cierta cantidad de semitonos a una nota. Las más comunes son el sostenido # y el bemol b, también existe el becuadro que se ocupa después de un sostenido o bemol y vuelve la nota a su frecuencia base. Con estos datos se crea el siguiente diccionario:

$$Alteraciones = \{\# : 1, - : 0, b : -1\}$$

Finalmente, todas las notas son acompañadas por un índice de octava I, el cual equivale a cuántos $I \cdot 12$ semitonos se encuentra una nota NI por sobre la nota N0 (con N cualquier nombre de nota). Uniendo los tres puntos anteriores se consigue la fórmula para ver la distancia de semitonos entre NAI y C0:

$$Distancia_{C0} = Notas[N] + 12 * I + A \tag{3.2}$$

Donde N representa el nombre de la nota A la alteración e I su octava.

Si restamos la distancia entre NAI y C0 y A4 y C0 se obtiene:

$$\begin{aligned} Distancia_{NI-A4} &= Notas[N] + (12 * I) + Alteraciones[A] + Notas[A](12 * 4) \\ Distancia_{NI-A4} &= Notas[N] + (12 * I) + Alteraciones[A] - 57 \end{aligned} \tag{3.3}$$

Con esto tenemos la distancia entre una nota NI y A4, sustituyendo entonces en la ecuación (3.1):

$$\begin{aligned}
 F_{nota} &= 440(12\sqrt{2})^d \\
 F_{nota} &= 440(12\sqrt{2})^{Notas[N]+(12*I)+Alteraciones[A]-57}
 \end{aligned}
 \tag{3.4}$$

Con esta fórmula se pudo tomar cualquier nota en escritura anglosajona del estilo NAI y al multiplicarla por 440hz, obtener la frecuencia de la nota solicitada. Pero se cambió el orden dentro de la aplicación a NIA, debido a que A es opcional. Entonces, se guardó a nivel local solo notas La440 para todos los instrumentos que se utilizaron en la plataforma y se podían tocar las notas deseadas alterando la frecuencia original del archivo.

3.2.2. Almacenamiento de audios en la base de datos

Con la fórmula anterior, se obtiene virtualmente cualquier nota a partir de su nombre y de un único archivo de audio, por esto, si un usuario quiere guardar en una base de datos que tocó una nota, solo tendrá que guardar 3 caracteres: la nota anglosajónica, la escala y la alteración, obteniendo, por ejemplo, *G3#* (Sol 3 Sostenido), minimizando el uso de datos de 1,34MB a 3B.

3.2.3. Diseño de la clase nota

Nota fue una clase ideada para ser utilizada por todas las herramientas de la plataforma que requieran emitir sonidos. Se construye a partir del nombre de una nota anglosajónica, un índice de octava y la alteración musical (si posee). A continuación, se presenta una representación UML del diseño de la clase:

Al haber construido la clase con los tres elementos requeridos para calcular la frecuencia (Nombre Nota, Octava, Alteración), se idea que una nota guarde una copia del audio (copia que se genere en tiempo de ejecución) y le modifique la frecuencia con la Ecuación 3.4. Con esto, distintas instancias de la clase entregan distintas Notas para reproducir. Figura 3.2.



Figura 3.2: Diseño de la clase Nota

3.3. Herramientas evaluativas de creación musical

3.3.1. Piano Virtual

El piano virtual se concibió como una herramienta creativa que posee el alumno de responder preguntas entregadas por el profesor, por ende es necesario, que el alumno identifique un piano, sus notas visualmente, y la posibilidad de guardar lo que toque para que el profesor después lo pueda reproducir. Visualmente se pensó como se muestra en la Figura 3.3.

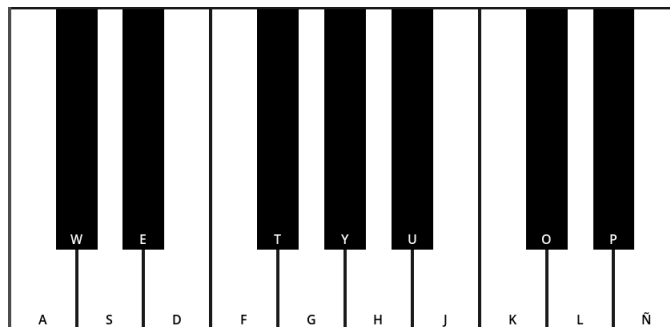


Figura 3.3: Diseño del piano virtual

En donde, en la parte inferior se puede ver qué letra del teclado es necesaria apretar para hacer sonar la nota deseada. El piano se pensó de este tamaño pues era la máxima separación entre dos letras en un teclado tipo QWERTY. Luego, basándose en el sistema de audio y la clase de Notas ya diseñadas, que permite tener múltiples frecuencias, lo único que se requiere para poder guardar una nota presionada, es saber su nombre NIA, en qué momento se comenzó a tocar y cuánto tiempo duró presionada.

Para ello, se piensa el piano virtual como dos arreglos de nombres de notas (una para las teclas blancas y otra para las teclas negras) y añadirle un botón de grabación, que al momento de presionar dicho botón se inicie un contador, para así tener una guía capaz de entregar los tiempos de inicio de la presión de cada nota y cuánto tiempo se mantuvo presionada. Así una melodía tocada con el piano se transformaría en un arreglo del estilo:

$$[(nota_1, inicio_1, duracion_1), (nota_2, inicio_2, duracion_2)...(nota_n, inicio_n, duracion_n)]$$

Esto tiene la facilidad de ser liviano para ser subido a una base de datos y ser obtenido por el profesor para reproducir lo que el estudiante tocó, para esto, en la aplicación del docente, al momento de iniciar la reproducción deberá comenzar a funcionar un contador de tiempo, y en el momento que el tiempo de inicio de una nota sea la misma que la del contador, se reproducirá por el periodo que el mismo arreglo especifica. Suponiendo que el alumno tocara durante 20 segundos, haber grabado en formato .wav generaría un archivo de 5MB (ver ecuación A.1), mientras que con este sistema de arreglos, si bien es cierto que el peso varía dependiendo de la cantidad de notas, tiene una cota inferior de 0B y una cota superior de 54KB (equivalente a presionar las 17 teclas del teclado al mismo tiempo, cada 0.1 segundos, por 30 segundos).

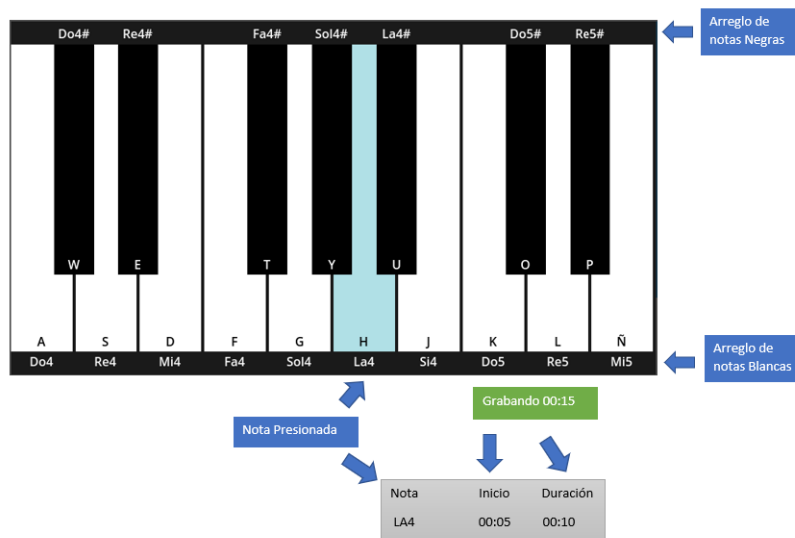


Figura 3.4: Concepto de uso del piano virtual

3.3.2. Sistema de partitura

Al igual que el piano virtual, el sistema de partituras también se ideó como una herramienta para evaluar las capacidades creativas del alumno, pero además, permite evaluar el conocimiento teórico que se requiere para poder escribir una partitura. Para poder facilitarle un poco el trabajo a estudiantes y profesores, se decidió que este sistema iba a tener un componente sonoro y lo que fuera escrito dentro de la partitura, se pudiera oír, con esto el alumno podrá saber a ciencia cierta, qué fue lo que estipuló en su partitura. Para esto, se vuelve a utilizar el sistema de audio diseñado.

Para idear todas las aristas que un sistema de partituras contiene, se separará la explicación en dos ejes: las configuraciones iniciales de una partitura, y la zona de edición de notas.

Configuraciones iniciales de una partitura

Una partitura contiene varios elementos que estipulan características de la pieza a tocar, las que se señalan en la Figura 3.5.

- La llave: Se encarga de definir que notas representa cada línea y espacio dentro de un pentagrama.
- Armadura: Define las alteraciones que presentarán un cierto conjunto de notas a lo largo de la pieza musical.
- Compás: Fracción que estipula cuántas figuras musicales son capaces de entrar dentro de un compás. Para calcularlo se utiliza la siguiente expresión:

$$Capacidad_{Compás} = (4,0/Divisor) * Dividendo \quad (3.5)$$

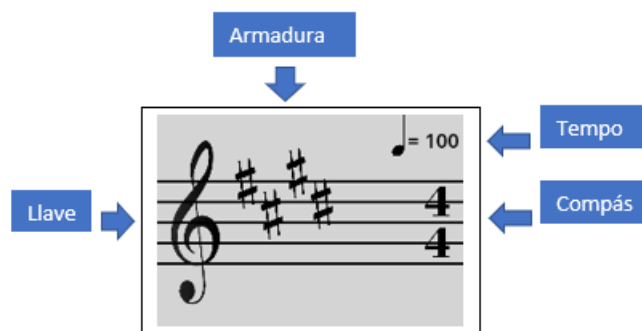


Figura 3.5: Elementos de la configuración inicial

- Tempo: Estipula la cantidad de pulsaciones por minuto que tendrá una partitura, donde cada pulsación es un tiempo de compás.

Cada una de estas configuraciones será guardada de la siguiente manera:

- La llave: Un valor número que representa la llave de Sol o la llave de Fa.
- Armadura: El índice de una posible configuración de Armadura (Ver Anexo).
- Compás: Dos números que representan la cantidad de figuras que caben dentro de un compás en el pentagrama.
- Tempo: Guarda un valor que configura el tiempo en el cual se lee, interpreta o reproduce una grabación.

Al guardar todo esto en un diccionario, se logra almacenar todas las configuraciones que contiene la partitura y ahora es posible enfocarse en el área de la inserción de elementos para una partitura.

Zona de edición de notas

Se ideó una partitura con la capacidad de 21 notas, como se puede ver en la Figura 3.6.



Figura 3.6: Notas de una partitura, cada nota ocupa intercaladamente una línea o un espacio

Para guardar una melodía escrita en la partitura se diseñaron dos estructuras. La primera estructura corresponde a un arreglo de diccionarios ordenado cronológicamente, en donde cada diccionario equivale a la configuración de una figura musical, este contiene la posición donde fue insertada una figura (índice), el tipo de figura (redonda, negra, etc), el tiempo que representa, si es un sonido o un silencio, y finalmente, si se le añadió alguna alteración.

La segunda estructura es un arreglo de tamaño 21 donde cada elemento corresponde a una nota principal más sus 5 posibles alteraciones: {doble sostenido, sostenido, nota base, bemol y doble bemol}, que modifican la nota principal añadiéndole {2, 1, 0, -1, -2} semitonos respectivamente.

Al sumar ambos modelos se puede saber qué nota tocar, por cuánto tiempo y desde qué momento. Del diccionario de la primera estructura se puede obtener el índice de la nota y la alteración sufrida, ambas al ser insertadas en la segunda estructura permite saber qué nota se ha de tocar. La primera estructura también define por cuánto tiempo se ha de tocar (por el tipo de figura), esto sumado a lo anterior permite saber qué nota y por cuánto tiempo tocarla. Para saber cuándo se debe reproducir la nota, basta con hacerla sonar justo después de terminada la nota anterior, pues este sistema de partituras fue pensado solo para tocar una nota a la vez.

A nivel visual también se ideó que la separación entre figuras musicales sea acorde a la duración de las mismas, para tener una representación gráfica de su duración, tal como se muestra en la Figuras 3.7 y 3.8.

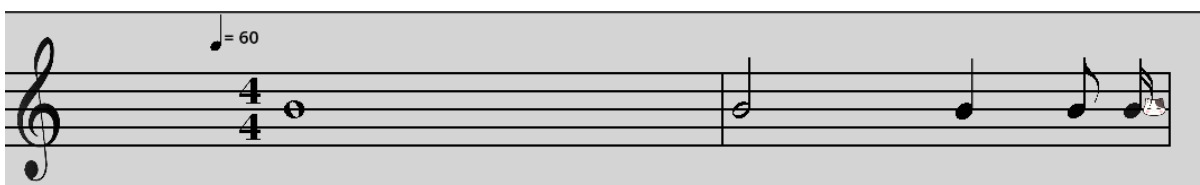


Figura 3.7: Figuras con distancias proporcionales al tiempo de sonido correspondiente

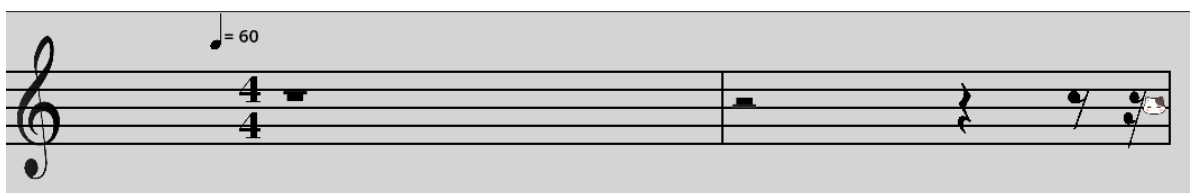


Figura 3.8: Silencios con distancias proporcionales al tiempo de silencio correspondiente

De igual manera se puede observar que la redonda y su silencio abarcan el compás completo (4 tiempos, pues el compás es 4/4), mientras que la blanca y su silencio usa la mitad (2 tiempos), la negra y su silencio un cuarto (1 tiempo), la corchea y su silencio un octavo (0.5 tiempos) y finalmente la semicorchea y su silencio un dieciseisavo (0.25 tiempos).

En términos de inputs necesarios para escribir dentro del lenguaje de partituras, se decantó por utilizar un cursor que se pudiera mover con las flechas Arriba/Abajo del teclado. Para insertar notas, se definió tener un mapeo de letras que tengan su equivalencia con figuras musicales, de forma tal que para agregar una figura musical al pentagrama se debe mover el

cursor hacia la altura deseada (altura que representaría la nota) y, al momento de presionar la letra acorde a la figura, esta aparezca en el pentagrama. Para eliminar, el botón Delete eliminaría el último elemento escrito, sin alterar la altura actual del cursor. Finalmente, para agregar alteraciones, se deben combinar entre las letras de inserción de figuras y los botones SHIFT (para sostenidos), CTRL (para bemoles) y ALT (para becuadros).

Los mapeos de figuras musicales con letra, fueron diseñados de tal forma que los sonidos están todos organizados en la fila de al medio del teclado, y los silencios en la fila superior, de tal forma que cada tecla de la fila de arriba tiene un silencio que dura lo mismo que la duración del sonido que produce la tecla que esta inmediatamente abajo. Esto se puede ver en las Figuras 3.9 y 3.10.

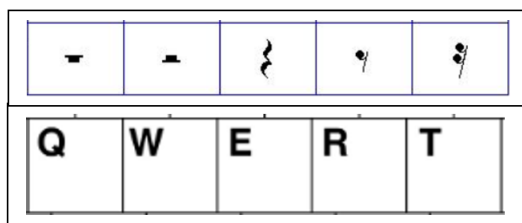


Figura 3.9: Silencios ordenados de mayor a menor, por ende la Q genera el silencio con mas tiempo (4 tiempos) y la T genera el menor (0.25 tiempos)

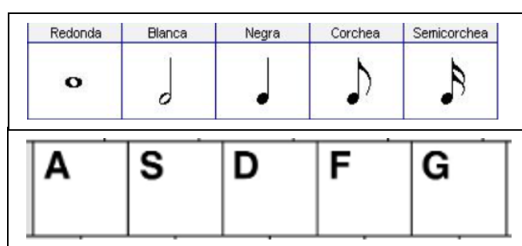


Figura 3.10: Sonidos ordenados de mayor a menor, por ende la A genera el sonido con mas tiempo (4 tiempos) y la T genera el menor (0.25 tiempos)

3.4. Modelo administrativo y de usuarios

Una vez diseñadas las herramientas musicales, se decantó por comenzar a estipular cómo se iba a organizar la aplicación y qué conceptos de un establecimiento se iban a plasmar en este ecosistema educativo. (Para ver el modelo a detalle consultar Anexo B.1).

- *Level(id, number, type)*: nivel o año académico del estudiante (Ej: Tercero Medio, Quinto Básico).
- *Grade(id, id_level, letter)*: curso con letra del estudiante (Ej: Octavo Básico C, Segundo Medio A).
- *User(id, email, first_name, middle_name, last_name, mothers_maiden_name, type, grade_id)*: modelo de usuario simple, que contiene la información suficiente para ser utilizado dentro del ecosistema educativo.

- *Student(id, email, first_name, middle_name, last_name, mothers_maiden_name, type, grade_id)*: Proxy del modelo User, que define automáticamente el type como STUDENT, modelo encargado de guardar todos los estudiantes de la aplicación.
- *Teacher(id, email, first_name, middle_name, last_name, mothers_maiden_name, type, grade_id)*: Proxy del modelo User, que define automáticamente el type como TEACHER, y grade_id como nulo. Modelo encargado de guardar todos los docentes de la aplicación.
- *Admin(id, email, first_name, middle_name, last_name, mothers_maiden_name, type, grade_id)*: Proxy del modelo User, que define automáticamente el type como SCHOOL_ADMIN y grade_id como nulo. Modelo encargado de guardar a los administradores de la aplicación.
- *Subject(id, name, description, teacher_id, grade_id, students)*: Modelo donde se relacionan los cursos, alumnos y profesores. Las asignaturas son todas las materias que se enseñan en un curso y donde serán aplicados los instrumentos evaluativos.

3.5. Sistema de Evaluaciones

El flujo del sistema de preguntas se definió de la siguiente manera: Un profesor crea una pregunta en el contexto de una evaluación, para una Asignatura en específico, el alumno al momento de abrir una evaluación puede ver dicha pregunta y asignarle una respuesta. Por ende, en primer lugar se creó el modelo evaluación (para más detalles, consultar AnexoB.2).

- *Quiz(id, name, subject_id, ended_date, is_active, is_ended, max_score, min_quiz_grade, aprobal_quiz_grade, max_quiz_grade, scale, ended_time, numbers_of_questions)*: Ente representativo de una evaluación. Entrega todos los datos necesarios para calcular la nota de los estudiantes a partir de una nota mínima, máxima y de aprobación, sumado a la escala para calcular la nota. Además, entrega información del tiempo que tienen los estudiantes, fecha de término de la evaluación, y a qué asignatura pertenece.
- *Question(id, quiz_id, number, score)*: Modelo abstracto de preguntas, contiene la evaluación a la que pertenece, su número dentro de la prueba, y el puntaje que obtendrá el alumno si contesta bien.
- *TOFQuestion(id, quiz_id, number, score, correct_answer)*: Modelo que hereda de Question, además de los datos ya expuestos, contiene la respuesta correcta que debiese entregar el alumno.
- *SelectionQuestion(id, quiz_id, number, score, correct_answer)*: Modelo que hereda de Question, además de los datos ya expuestos, contiene la respuesta correcta que debiese entregar el alumno.
- *WritingQuestion(id, quiz_id, number, score, rubric)*: Modelo que hereda de Question, además de los datos ya entregados, añade una rúbrica evaluativa.

- *PianoQuestion*(**id**, *quiz_id*, *number*, *score*, *is_visible*, *rubric*): Modelo que hereda de *Question*, además de los datos ya entregados, añade una rúbrica evaluativa y la posibilidad que los alumnos puedan ver o no, los nombres de las notas del Piano.
- *MusicSheetQuestion*(**id**, *quiz_id*, *number*, *score*, *rubric*): Modelo que hereda de *Question*, además de los datos ya entregados, añade una rúbrica evaluativa.

Para responder una evaluación fueron creados los siguientes modelos:

- *AnswerwedQuiz*(**id**, *quiz_id*, *student_id*, *subject_id*, *score_obtained*, *comments*, *time_left*, *is_checked*, *create_by_teacher*): Ente que representa la respuesta de un Estudiante, identifica al alumno, la asignatura y el tiempo que tardó en responder. El profesor revisará la evaluación y añadirá la nota y comentarios en este mismo modelo, editándolo. Si el alumno no contesta una prueba, un *AnsweredQuiz* será creado manualmente por el profesor y le podrá asignar nota mínima.
- *AnsweredQuestion*(**id**, *answered_quiz_id*, *comments*, *score*): Modelo abstracto de respuesta, tiene el id de la pregunta, los comentarios que pudo dejar el docente, y el puntaje que obtuvo el alumno.
- *AnsweredTOFQuestion*(**id**, *answered_quiz_id*, *comments*, *score*, *answer*): Respuesta que eligió el estudiante respecto a la pregunta de verdadero y falso, guardado en un string, puede ser Verdadero, Falso o No respondió.
- *AnsweredSelectionQuestion*(**id**, *answered_quiz_id*, *comments*, *score*, *answer*): Alternativa que respondió el estudiante respecto a la pregunta de selección de alternativa.
- *AnsweredWritingQuestion*(**id**, *answered_quiz_id*, *comments*, *score*, *answer*): Respuesta a una pregunta de desarrollo, que se responde en formato texto.
- *AnsweredPianoQuestion*(**id**, *answered_quiz_id*, *comments*, *score*, *answer*): Respuesta a una pregunta de Piano Virtual, que se responde tocando el instrumento y se guarda un archivo JSON, con las notas y las configuraciones iniciales.
- *AnsweredMusicSheetQuestion*(**id**, *answered_quiz_id*, *comments*, *score*, *answer*): Respuesta a una pregunta de Partitura, que se responde configurando e insertando figuras musicales a dicha partitura. Se guarda un archivo JSON, con las notas y las configuraciones iniciales.

Para ver cómo se relacionaron dichos modelos consultar el Anexo B.2.5.

3.6. Interfaz

Al estar dentro de los objetivos tener una plataforma sencilla de usar, se consideró de suma importancia darse el tiempo de modelar las distintas pantallas para los usuarios. A continuación, serán mostrados los diseños más relevantes, desarrollados en LucidChart, y que sirvieron de base para los modelos finales (ver Sección 4.6).

3.6.1. Pantallas de Estudiante

Pantalla Principal de Estudiantes

En primer lugar, se considera que un alumno está en distintas asignaturas, por ende, necesita un selector de asignaturas. Además, el alumno tiene pruebas por rendir y pruebas rendidas en dicha asignatura. Finalmente, un alumno puede cerrar sesión cuando quiera. Estos datos se organizan como muestra la Figura 3.11.

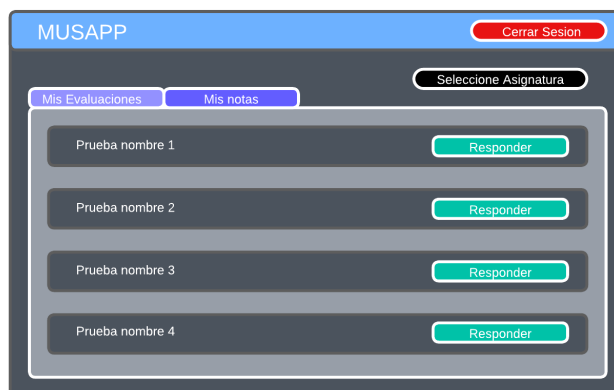


Figura 3.11: Mockup Pantalla Inicio del Estudiante

Evaluación

Luego, dentro de una evaluación, el alumno necesita pocos distractores visuales, no debería salir de la sesión y para responder necesitará ver tanto la pregunta como el instrumento evaluativo asignado de una manera clara e intuitiva de responder, tal como se muestra en la Figura 3.12.

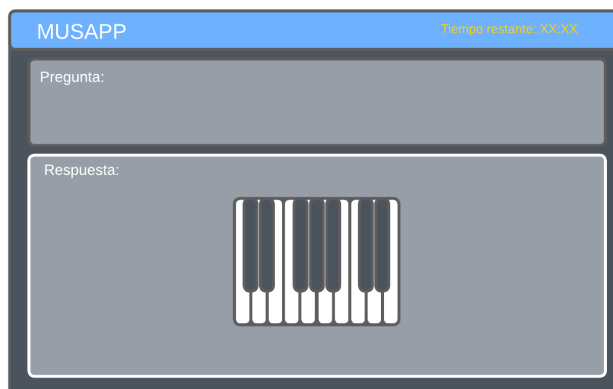


Figura 3.12: Mockup Pantalla de Evaluación del Estudiante, con una pregunta de Piano Virtual

3.6.2. Pantallas de Docente

Inicio Profesores

El docente tiene distintas asignaturas separadas por curso y estos, a su vez, por nivel, por lo que hay que agregar botones para poder seleccionar estos tres parámetros correctamente. Por otro lado, un docente puede crear evaluaciones y una vez de creadas, tomar una y publicarla, luego cerrarla y finalmente revisarla. En la Figura 3.13 se muestra cómo se diseñó visualmente estas funcionalidades.



Figura 3.13: Mockup Pantalla Inicio del Profesor

Crear Evaluación

Para crear una evaluación el profesor puede seleccionar un tipo de pregunta y crearla, de esta forma, puede crear toda las preguntas que tendrá la evaluación. En la Figura 3.14 se muestra la pantalla de creación de una evaluación.

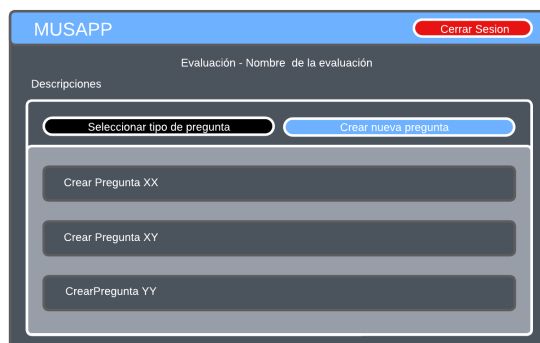


Figura 3.14: Mockup Pantalla de Crear Evaluaciones

Capítulo 4

Implementación

El proceso de implementación de cada característica de la aplicación fue desarrollado justo después de su diseño, es por eso que en esta sección se mantuvo el orden de la Sección 3. Aquí se describirán los códigos, clases y sistemas ideados para materializar los conceptos creados en la sección anterior.

4.1. Ambiente y plataforma

Luego de haberse pensado el ecosistema educativo como una aplicación de escritorio con conexión a internet, se estudiaron distintas tecnologías para poder trabajar eficientemente, que cumplieran con todos los requerimientos solicitados y se decantó por las siguientes:

- Django REST Framework: Conjunto de herramientas del framework Django que permite crear una API (Interfaz de programación de aplicaciones) del formato REST (Transferencia de estado representacional) por el lado del servidor. Entrega una interfaz que separa claramente el servidor del cliente, y logra una comunicación efectiva entre ambos utilizando los métodos básicos de HTTP, como lo son: POST, GET, PUT, PATCH y DELETE.
- MySQL: Sistema de gestión de Base de Datos Open Source, de pequeño tamaño y gran velocidad.
- Godot 4.0: Motor de videojuegos 2D y 3D Open Source, que permite desarrollar todo tipo de proyectos, tanto juegos como aplicaciones de escritorio o páginas web. Utiliza como lenguaje GDScript, lenguaje de alto nivel que sostiene una sintaxis similar a Python, por ende, permite un desarrollo rápido y fluido. Adicionalmente, entrega un editor visual para las distintas pantallas de la aplicación, con un banco de botones, contenedores y organizadores de elementos, que permiten un desarrollo cómodo, rápido y flexible. Al ser un motor de videojuegos, los sistemas de inputs y outputs veloces añadieron una agradable sensación en los modelos interactivos de sonido, como el piano y la partitura. Adicionalmente, permite la portabilidad a otros sistemas operativos como Linux o MacOS.

4.2. Sistema de Sonido

4.2.1. Sistema de Almacenamiento de Audios

Ya obtenida una Ecuación 4.1, que permite alterar al frecuencia de una nota, se buscó un banco de instrumentos musicales, donde se encontró un dataset de TinySol [12] con licencia Creative Commons Attribution 4.0, del cual solo se extrajo las notas con frecuencia 440Hz y se eligió: Acordeón, Clarinete, Contrabajo, Fagot, Flauta, Piano, Saxofón Alto, Trombón, Trompeta, Violín y Violoncello. Pero al hacer pruebas de sonido, muchos audios presentaban sonidos poco naturales y sin similitud al instrumento real que representaban. ¿Por qué? Pues porque no todos los instrumentos llegan con facilidad a la frecuencia de 440Hz, a algunos les queda aguda, y a otros grave. Por ende, se buscaron los rangos de cada instrumento y se buscó la nota A en la octava que les quedara aproximadamente a una frecuencia intermedia de su rango (ver Anexo D.1). Consiguiendo los siguientes valores:

- Nota A2: Contrabajo.
- Nota A3: Fagot, Violoncello, Trombón.
- Nota A4: Piano, Trompeta, Acordeón, Clarinete, Saxofón Alto.
- Nota A5: Flauta, Violín.

Una vez establecidos se importan dentro de Godot, en la carpeta instrument, la que alcanzó un tamaño de solo 7,10MB. Luego, se genera un conjunto de diccionarios con cada instrumento importado y el nombre de la nota base, para así poder calcular las notas requeridas. Esto se muestra en las Figuras 4.1 y 4.2.

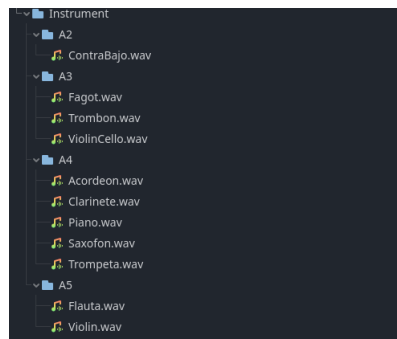


Figura 4.1: Instrumentos importados a Godot

```
var _instruments = [{"instrument": preload("res://Audios/Instrument/A4/Acordeon.wav"), "note": "A4"},
{"instrument": preload("res://Audios/Instrument/A4/Clarinete.wav"), "note": "A4"},
{"instrument": preload("res://Audios/Instrument/A2/ContraBajo.wav"), "note": "A2"},
{"instrument": preload("res://Audios/Instrument/A3/Fagot.wav"), "note": "A3"},
{"instrument": preload("res://Audios/Instrument/A5/Flauta.wav"), "note": "A5"},
{"instrument": preload("res://Audios/Instrument/A4/Piano.wav"), "note": "A4"},
{"instrument": preload("res://Audios/Instrument/A4/Saxofon.wav"), "note": "A4"},
{"instrument": preload("res://Audios/Instrument/A3/Trombon.wav"), "note": "A3"},
{"instrument": preload("res://Audios/Instrument/A4/Trompeta.wav"), "note": "A4"},
{"instrument": preload("res://Audios/Instrument/A5/Violin.wav"), "note": "A5"},
{"instrument": preload("res://Audios/Instrument/A3/ViolinCello.wav"), "note": "A3"}]
```

Figura 4.2: Diccionarios de instrumentos musicales

Adicionalmente, fue necesario modificar la Ecuación 3.4 para poder conseguir notas a partir de cualquier frecuencia base, se implementó así:

$$E = \text{Notas}[N] + \text{Alteraciones}[A] + (12 * I) - (\text{Nota}[N_{base}] + (12 * I_{base})) \quad (4.1)$$

$$F_{nota} = \text{Frecuencia}_{base} ({}^{12}\sqrt{2})^E$$

Con F_{nota} , la frecuencia de la nota base a utilizar, $\text{Nota}[N_{base}]$ e I_{base} , el Nombre e Índice de Octava de la nota base (no se colocaron alteraciones pues no se utilizó ninguna nota base con alguna alteración).

4.2.2. Implementación de la clase Nota

Godot trae una clase enfocada en el manejo de audios llamado `AudioStreamPlayer`, permite poder cambiar la frecuencia de una nota base con el atributo `pitch_scale`, de la misma forma que la Ecuación (4.1), por lo tanto, fue implementada de la forma que muestra la Figura 4.3.

```
var accidental = note[2] if note.length() > 2 else "-"
var exponent = ((notes[note[0]] + note[1].to_int() * 12 + _accidentals[accidental]) - (notes[initial_note[0]] + initial_note[1].to_int() * 12)) / 12.0
_audio = AudioStreamPlayer.new()
_audio.stream = instrument
_audio.pitch_scale = pow(2, exponent)
_audio.set_volume_db(12.0)
```

Figura 4.3: Implementación de la ecuación (4.1) en Godot

Adicionalmente, `AudioStreamPlayer`, permite reproducir y detener el audio que se le haya asignado en el atributo `stream`, por lo tanto, los métodos más importantes de la clase fueron implementados tal como muestra la Figura 4.4.

```
#Reproducir el audio
func play():
    _timer.stop()
    _audio.play()

#Detener el audio con delay, utilizando un timer
func stop():
    _timer.start()

#Detencion del timer de delay
func on_timer_timeout():
    _timer.stop()
    _audio.stop()

#Detener el audio de manera forzosa
func force_stop():
    _audio.stop()

#Reproducir un audio, por una cantidad de tiempo time, con un delay final de delta
func play_x_time(time, delta = 0):
    _audio.play()
    await get_tree().create_timer(time).timeout
    if delta and _audio.is_playing():
        wait_delta(delta)
    else:
        _audio.stop()

#genera el delay del delta
func wait_delta(delta):
    await get_tree().create_timer(delta).timeout
    _audio.stop()
```

(a)

(b)

Figura 4.4: Clase Nota

Para grabar, primero que nada el usuario no debe estar reproduciendo ninguna grabación anterior, luego, se detiene cualquier tecla que esté presionando el usuario previo a comenzar a grabar, se guarda en un diccionario la configuración que el piano presente en ese momento y se inicia un cronómetro, el cual servirá para guardar el momento que se comience a presionar una nota. Además se crea un diccionario en donde se irán guardando los tiempos de inicio de presión para cada tecla. El código fuente de esto se ve en la Figura 4.7.

```
#Iniciar una grabación
func record_song():
  if not _is_playing:
    stop_all_notes()
    _song = {"init_note" : _init_note_idx, "init_pitch" : _init_pitch, "instrument" : _index_instrument, "notes" : []}
    _dict = {}
    _is_recording = true
    _chronometer.start()
```

Figura 4.7: Código para comenzar a grabar

Al momento de presionar una tecla, se genera un arreglo llamado “new_note” que contiene el índice de la tecla, el tiempo que fue presionada (este tiempo se sobrescribirá al momento de liberarse la tecla, pero por defecto es 0.01 que es el tiempo mínimo de sonido que guarda el cronómetro), y el tiempo que tenía el cronómetro al momento de ser presionada la tecla, considerado el tiempo inicial de la nota. Este sub arreglo se añade al final del arreglo “_notes” dentro del diccionario “_song”. Además, en el diccionario “_dict” se guarda el conjunto llave-valor, correspondiente a el índice de la tecla y una referencia a “new_note”. Internamente, la tecla al momento de ser presionada inicia un nuevo cronómetro en paralelo, que finaliza al ser liberada y envía este tiempo en el atributo “time” en la función key_released(index, time). Con esto, al liberarse la tecla, se revisa si previamente fue presionada, y de ser así, se consigue la referencia guardada en el diccionario de “new_note” con el índice de la nota tocada y se actualiza el tiempo que fue presionada por “time”. El código asociado a lo anterior se muestra en la Figura 4.8.

```
#Una tecla del piano fue presionada
func key_pressed(index):
  if _is_recording:
    var new_note = [index, 0.01, _chronometer.get_time()]
    _song["notes"] += [new_note]
    _dict[index] = new_note

#Una tecla de piano dejo de ser presionada
func key_released(index, time):
  if _is_recording:
    if _dict.has(index):
      _dict[index][1] = time
```

Figura 4.8: Presionar y liberar una tecla

Destacar que se usa un diccionario para guardar la referencia, pues así, al ser presionada nuevamente una nota ya tocada, se sobrescribirá la referencia a “new_note”. Al finalizar de grabar, se detiene el cronómetro y cualquier nota que se haya mantenido presionada. El código para la detención de la grabación se muestra en la Figura 4.9.

Notar que la grabación entrega un diccionario “_song” que contiene tanto las notas como la configuración que tenía el piano al momento de grabar.

```

#Detener una grabación
func stop_record_song():
>| if not _is_playing:
>| | stop_all_notes()
>| | _is_recording = false
>| | _chronometer.stop()
>| | emit_signal("record_stopped")

```

Figura 4.9: Código para detener una grabación

Como última función importante, se hablará del sistema de reproducción. Este envía una señal al piano para que se adapte a las configuraciones guardadas en la grabación. Luego, se calcula el tiempo total de la grabación sumando para cada nota su tiempo de inicio + su duración, pues la mayor de estas sumas equivale al momento que la última nota tocada se libera, marcando el fin de canción.

Para reproducir la canción, no debe haber una grabación en curso, luego para cada nota del arreglo, se hace sonar la nota por el tiempo estipulado por la grabación, de forma paralela al hilo principal, desde el momento que indique su tiempo inicial. Para poder saber en qué momento se cumple el tiempo inicial sin un cronómetro global al cual consultarle constantemente, se harán pausas que duren la diferencia entre el inicio de nota anterior y el actual (partiendo con el inicio de nota anterior, *last_time* = 0). Así el código continuará justo para el tiempo de inicio de cada nota, y luego se pausará hasta el tiempo de inicio de la siguiente nota. Para detener la canción de manera anticipada, si así el usuario lo desea, se cancela la pausa del hilo principal y se tiene un condicional que revisa si la canción debe seguir sonando. El código de esta implementación se muestra en la Figura 4.10

```

func play_song():
  _initial_wait.stop()
  var max_time = 0
  for arr in _song["notes"]:
    >| if arr[1]+arr[2] > max_time:
    >| | max_time = arr[1]+arr[2]
  enable_input(false)
  emit_signal("song_playing")
  stop_song_after_x_time(max_time)
  if not _is_recording:
    >| _is_playing = true
    >| var last_time = 0
    >| var time = 0
    >| for arr in _song["notes"]:
    >| | time = arr[2] - last_time if arr[2] > last_time else 0
    >| | last_time = arr[2]
    >| | if time > _chronometer.get_delta():
    >| | | _initial_wait.set_wait_time(time)
    >| | | _initial_wait.start()
    >| | | await _initial_wait.timeout
    >| | | if not _is_playing:
    >| | | | break
    >| | | _keys[arr[0]].activate_x_time(arr[1])

```

Figura 4.10: Código para reproducir el piano

Finalmente destacar, que tanto al tocar como al reproducir una grabación, las notas que el usuario toque o haya tocado, se muestran visualmente como en la Figura 4.11

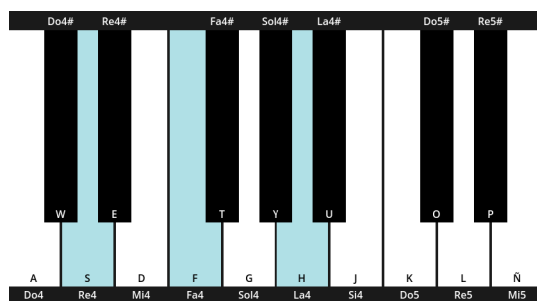


Figura 4.11: Feedback al usuario de las teclas presionadas

4.3.2. Implementación Sistema de partituras

El sistema de partituras a nivel visual resultó como se puede apreciar en la Figura 4.12.



Figura 4.12: Sistema de partituras

Se le añadió el componente sonoro a la partitura y, al momento de reproducirla muestra una guía visual, indica qué parte de la partitura está sonando en el momento, como en la Figura 4.13.



Figura 4.13: Indicador de reproducción

Configuraciones iniciales de una partitura

A nivel de visualización para el usuario, para configurar la partitura se tiene el siguiente cuadro disponible, presionando el botón de configuración.



Figura 4.14: Configuración del sistema de partituras

Donde se puede apreciar que todos los elementos mencionados en el diseño están representados (llave, armadura, compás y tempo). Se le añadió una ventana de previsualizaciones en donde el estudiante podrá ver cómo cambiar una configuración afecta la forma de la partitura.

A nivel de código, las configuraciones más relevantes son el cambio de llave, y el cambio de armaduras, pues el compás y el tempo son solo variables que serán almacenadas para más adelante.

Iniciando con el cambio de llave, siguiendo la idea del modelo que consistía en, teniendo el nombre de una nota, iterar por el arreglo de 21 elementos representante de las líneas y espacios del pentagrama para configurar que nota le corresponde a cada uno de los elementos. Para ello, al momento de configurar se borran todas las notas anteriores, luego se itera el arreglo de manera descendente y se van creando cinco notas por cada elemento de arreglo, representando todas las alteraciones posibles que puede sufrir la nota (doble bemol, bemol, normal o estado fundamental, sostenido, y doble sostenido). Esto es almacenado como un sub arreglo del arreglo “notes”, y finalmente se genera un arreglo en paralelo que tiene el nombre de las notas con la que comparte índice en el arreglo “notas”, pero el nombre es sin la octava (esto será necesario para insertar notas). El código del cambio de llave está en la Figura 4.15.

Por otro lado, para la configuración de armaduras, es necesario saber que existen 15 configuraciones distintas: 1 sin alteraciones, 7 con alteraciones de sostenidos, y 7 con alteraciones de bemoles, cada uno con su nombre (Ver Anexo C.1). A modo resumen, la configuración sin alteraciones es C, las alteraciones con sostenidos, son incrementales, es decir, cada alteración tendrá una nota sostenida más que la anterior, en el orden $[F, C, G, D.A, E, B]$, (la primera tendrá F, la segunda F y C, etc). Por otro lado, las alteraciones con sostenidos también son incrementales pero se incrementan en el orden inverso $[B, E, A, D, G, C, F]$ (la primera tendrá B, la segunda B y E, etc).

A partir de esto, la configuración de una armadura se basa en un índice que equivale a alguno de estas configuraciones: 0 sin alteraciones, del 1 al 7 con las alteraciones de sostenidos en orden y de 8 al 15 las alteraciones de bemoles en orden. Con esto en mente, si el índice es menor o igual a 7, se irán agregando sostenidos en el orden $GSingatureIds = [F, C, G, D.A, E, B]$, hasta alcanzar el índice. De ser mayores a 7, se agregaran bemoles en el orden inverso hasta

```

#Cambio de llave
func changes_key_notes(letter, pitch):
    if len(notes)>0:
        for current_notes in notes:
            for current_note in current_notes:
                remove_child(current_note)
                current_note.queue_free()
        notes = []
        notes_letter = []
        for i in range(21):
            if letter == "B":
                pitch -= 1
            var new_note_2flat = note.new(letter+str(pitch)+str("f"))
            var new_note_flat = note.new(letter+str(pitch)+str("b"))
            var new_note = note.new(letter+str(pitch))
            var new_note_sharp = note.new(letter+str(pitch)+str("#"))
            var new_note_2sharp = note.new(letter+str(pitch)+str("x"))
            notes+=[new_note_2flat, new_note_flat, new_note, new_note_sharp, new_note_2sharp]
            add_child(new_note_2flat)
            add_child(new_note_flat)
            add_child(new_note)
            add_child(new_note_sharp)
            add_child(new_note_2sharp)
            notes_letter += [new_note.get_letter()]
            if letter == "A":
                letter = String.chr(letter.unicode_at(0) - 1)
            else:
                letter = "G"

```

Figura 4.15: Cambio de llave

alcanzar el valor (7 - índice). El código se puede ver en la Figura 4.16.

```

func changes_accidentals(index):
    var sharp = true
    if index > len(GSignatureIds):
        index = index % 7
        sharp = false
    for i in accidentals.keys():
        accidentals[i] = 2
    for i in range(index):
        var correct_id
        var correct_accidental
        if sharp:
            correct_id = i
            correct_accidental = 3
        else:
            correct_id = len(GSignatureIds) - 1 - i
            correct_accidental = 1
        accidentals[GSignatureIds[correct_id]] = correct_accidental

```

Figura 4.16: Cambio de Armadura

Zona de edición de notas

Al momento de insertar una nota, ocurren dos procesos, se almacena en un arreglo el conjunto de elementos que permitirán reproducir la nota en un futuro, y se inserta de manera visual la figura correspondiente al compás en donde se encuentre el cursor.

En primer lugar, se verá la implementación de la inserción visual. Para contextualizar, la zona de edición de notas corresponde a un conjunto de un precompás, en donde se estipula la configuración inicial de la partitura (Desde la llave hasta la fracción), y luego un máximo de tres compases uno tras otro (el espacio entre la fracción y una línea vertical, o entre dos líneas verticales).

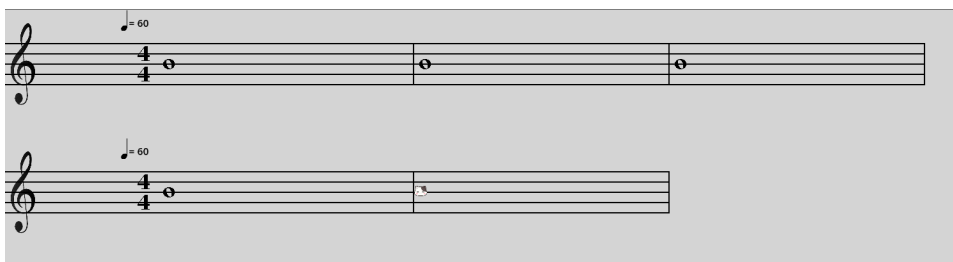


Figura 4.17: Dos filas de partitura, en la primera se ve un precompás y tres compases, en la segunda se ve un precompás y solo dos compases

Al presionar una tecla tipo letra para insertar una nota, se llama a la función `save_note`(figure, accidental, silence = false, sound = true), los parámetros son: el nombre de la figura a insertar, la alteración, si es un silencio, y si al momento de insertar la figura, se debe emitir el sonido de la nota insertada. Primero se ve si la figura a insertar es un silencio o no, luego se intentará insertar la figura de manera visual, pues la inserción visual se encarga de ver si la nota cabe dentro de un compás o no. De poder insertarse, se emite el sonido de la nota (si `sound` es verdadero), y se guarda en el arreglo `current_song`, la configuración de la nota: la altura actual del cursor, la figura, la alteración (tanto escrita como numérica) y si emite sonido o es silenciosa. Finalmente se revisa si la cantidad de figuras dentro del compás es equivalente al espacio disponible que tiene el compás para insertar notas, de cumplirse, se añade un nuevo compás.

```
func save_note(figure, accidental, silence = false, sound = true):
  if silence:
    if current_measure.insert_silence(figure_time[figure], figure):
      current_song += [{"figure": figure, "is_silence": silence}]
  else:
    if current_measure.insert_note(figure_time[figure], figure, accidental):
      if sound:
        notes[current_cursor][current_measure.get_last_accidental_time(notes_letter[current_cursor])].play_x_time(0.5)
      current_song += [{"cursor": current_cursor, "figure": figure, "accidental": accidental, "note_accidental": current_measure.get_last_accidental_time(notes_letter[current_cursor]), "is_silence": silence}]
    if current_measure.next_measure():
      create_measure()
```

Figura 4.18: Código para insertar una nota

El caso del silencio es análogo pero se omite todo lo relacionado con alteraciones.

Para borrar notas, en primer lugar se ve si existe un compás, de ser así, revisa si el compás tiene notas. Si el compás está vacío se borra el compás y se vuelve a llamar la función `delete_notes()`. De no estar vacío el compás, se borra visualmente la figura, luego se elimina la última nota del arreglo de notas “`current_song`”.

```
func delete_note():
  if current_measure:
    var has_notes = current_measure.delete_note()
    if not has_notes and len(measures) > 1:
      delete_measure()
      current_measure.set_cursor(current_cursor)
      delete_note()
    elif has_notes:
      current_song.pop_back()
```

Figura 4.19: Código para eliminar una nota

Finalmente para reproducir una canción guardada, las configuraciones iniciales se hacen previas a iniciar la canción. Dentro de `play song` ocurren dos cosas, en primer lugar se inicia

un play song que recorre cada compás. Esto lo hace calculando la cantidad de tiempos del compás y lo multiplica por el tempo de la configuración inicial, eso es el tiempo que debe tomar el indicador en cruzar el compás completo. Mientras que las notas, se tocan de manera similar a como se reproducían en el piano virtual. Se deja una nota sonando en paralelo el tiempo que le corresponde a la figura por el tempo de las configuraciones iniciales, luego se pausa el hilo principal el mismo tiempo que debe sonar la nota (esto debido a que este sistema de partitura solo va tocando notas de a una), cuando deja de sonar una nota, continua el hilo para dejar sonando la siguiente nota. Cuando la cantidad de figuras reproducidas igual a la cantidad de figuras que admite el compás. Se ejecuta play song en el siguiente compás para que inicie el indicador en dicho compás. Así sucesivamente hasta terminar de reproducir todas las figuras existentes en la partitura.

```
func play_song():
  if not is_playing_song:
    var current_scroll = 0
    $ VBoxContainer/Sheet/ScrollContainer.set_v_scroll(current_scroll)
  var i = 0
  var wait_time
  var current_time = 0
  is_playing_song = true
  measures[i].play_song()
  for current_note in current_song:
    if not current_note["is_silence"]:
      notes[current_note["cursor"]][current_note["note_accidental"]].play_x_time(get_current_figure_time(current_note["figure"]))
      wait_time = get_current_figure_time(current_note["figure"])
      current_time += get_current_figure_time(current_note["figure"])
      if wait_time > 0:
        await wait_time(wait_time)
        if not is_playing_song:
          measures[i].stop_song()
          break
      if abs(current_time - (time_max * tempo_normalized)) < 0.001:
        current_time = 0
        measures[i].stop_song()
        i += 1
        if i < len(measures):
          measures[i].play_song()
          if i % 3 == 0:
            current_scroll += vertical_scroll
            $ VBoxContainer/Sheet/ScrollContainer.set_v_scroll(current_scroll)
  stop_song()
```

Figura 4.20: Código para Reproducir una partitura

4.4. Modelo administrativo y de usuarios

Este modelo no sufrió modificaciones con respecto su diseño. Pero sí fueron implementadas las formas de consultar y modificar estos modelos, utilizando una API REST. A continuación mencionaremos los más relevantes:

- Login: Permite el inicio de sesión con un email y contraseña, creando y entregando un Token.
- Logout: Elimina el Token del usuario, cerrándole sesión.
- CreateUser: Crea un usuario (Alumno o Docente).
- ProfileStudent: Entrega un diccionario con el estudiante y sus asignaturas. También permite editar y borrar el usuario (Solo disponible para el ADMIN).
- ProfileTeacher: Entrega un diccionario con el docente y sus asignaturas. También permite editar y borrar el usuario (Solo disponible para el ADMIN).

- ProfileSchoolManager: Entrega un diccionario con el administrador, los niveles, cursos, asignaturas, profesores y estudiantes, de la aplicación.
- LevelAndGrade: Entrega todos los niveles y cursos de la aplicación, también permite borrar y crear cursos para un nivel, y borrar el nivel con todos sus cursos.

4.5. Sistema de evaluaciones

Al igual que el modelo anterior, tampoco se realizaron cambios con respecto al diseño, y nuevamente se mencionarán las funcionalidades más importantes implementadas en la API:

- TeacherQuiz: Permite crear una evaluación con todas sus pregunta. Permite también su edición y obtención.
- StudentQuiz: Le entrega a un alumno una evaluación con todas sus preguntas (eliminando el campo correct_answer de los modelos TOFQuestion y SelectionQuestion). También permite crear una respuesta a dicha evaluación y actualizar el tiempo que el alumno tiene disponible para terminar la evaluación.
- StudentAnsweredQuestions: Permite crear todas las respuestas a las preguntas de una evaluación.
- TeacherQuizzesBySubject: Le entrega al profesor todas las evaluaciones que haya creado dentro de una asignatura, separadas por las evaluaciones inactivas, activas y terminadas.
- StudentQuizzesBySubject: Le entrega al alumno todas las evaluaciones que se encuentran activas para ser respondidas, y las respuestas a evaluaciones que haya contestado, con ello podrá saber que nota obtuvo al momento que el profesor la revise.
- StudentAndAnswersByQuiz: Le entrega al docente las respuestas de todos los alumnos para una evaluación.
- CheckAnsweredQuizbyTeacher: Le entrega a un docente todas las respuestas a preguntas de una evaluación. También permite que modifique tanto las respuestas (su puntaje) como la nota final del estudiante.

4.6. Interfaz

4.6.1. Inicio de sesión

En primer lugar, se implementó una interfaz de Inicio de Sesión para la cual muestra una descripción de la aplicación, las opciones de iniciar sesión y crear un usuario, además de la posibilidad de probar tanto el piano como la partitura (sin conexión a Internet).

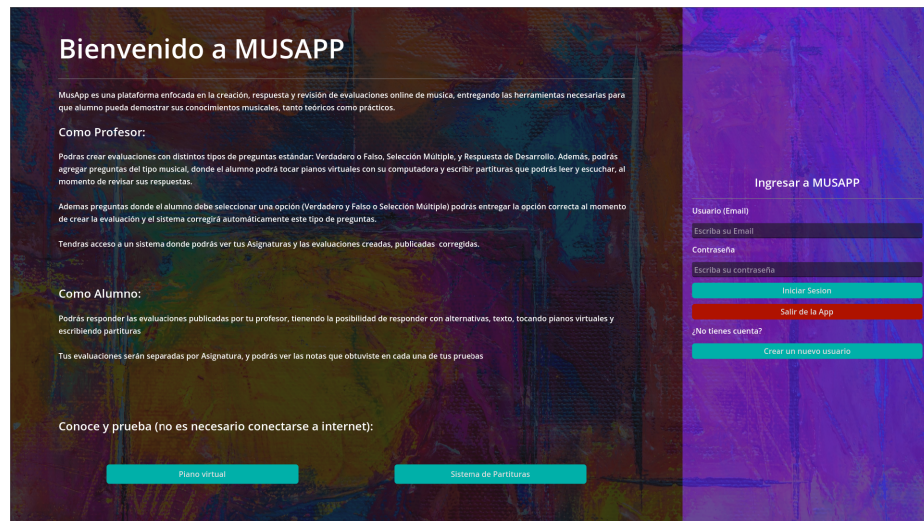


Figura 4.21: Inicio de sesión

4.6.2. Estudiante

Pantalla principal

La implementación de la interfaz de la pantalla principal no tuvo grandes cambios con respecto al diseño (Figura 3.11). En donde se puede apreciar la opción de ver sus evaluaciones o sus notas (Figuras 4.22 y 4.23)

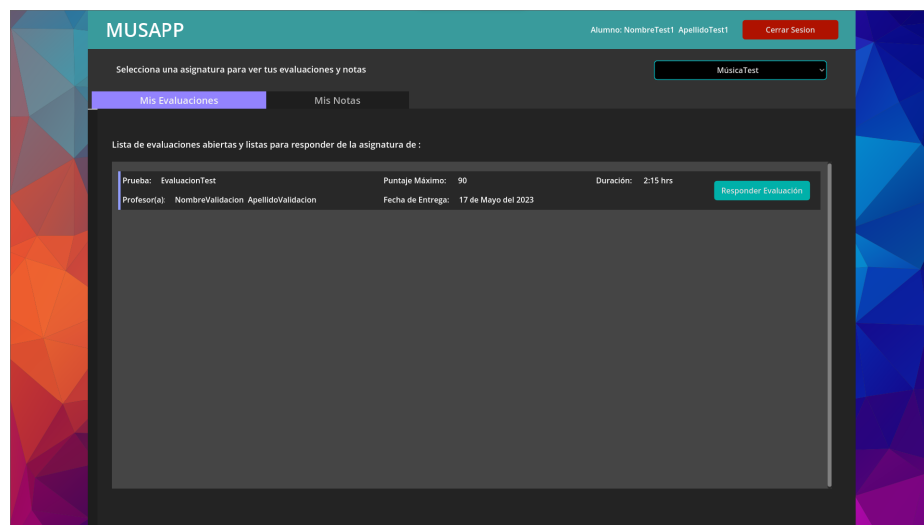


Figura 4.22: Alumno: Pantalla principal

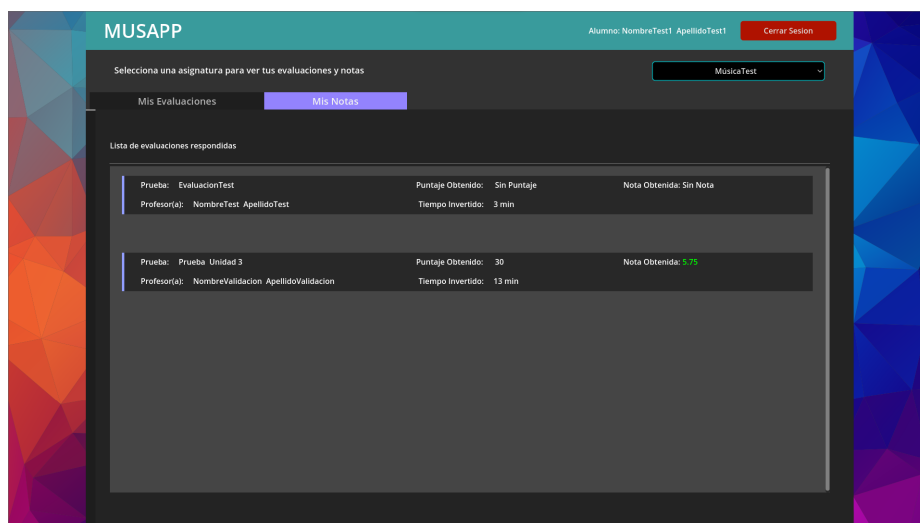


Figura 4.23: Alumno: Pantalla principal, ver notas

Responder una evaluación

Al momento de comenzar una evaluación, el alumno no podrá cerrar sesión y se mostrará un contador del tiempo restante, notar que se mantuvo la idea original presentada en la Figura 3.12.

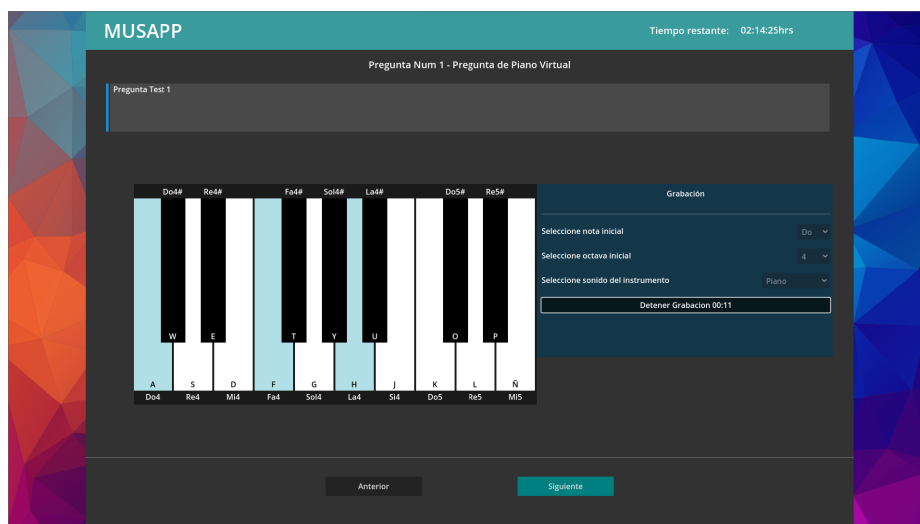


Figura 4.24: Alumno: Responder evaluación, Piano

Destacar que el el piano presenta un contador de cuanto tiempo lleva grabando, y la partitura muestra todas las instrucciones de uso.

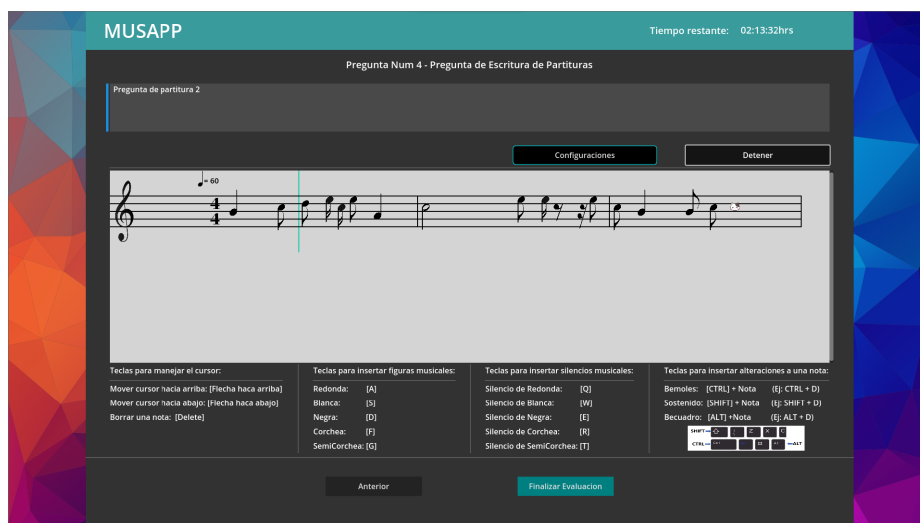


Figura 4.25: Alumno: Responder evaluación, Partituras

4.6.3. Profesor

Se decidió cambiar los colores de fondo del docente, para que al ingresar alumno y profesores, vean una aplicación un tanto diferente, se considera un cambio sutil pero necesario, para entender que son ambientes distintos. Se puede apreciar también, que la pantalla del docente permite crear evaluaciones, publicarlas y revisarlas (Figura 4.26).

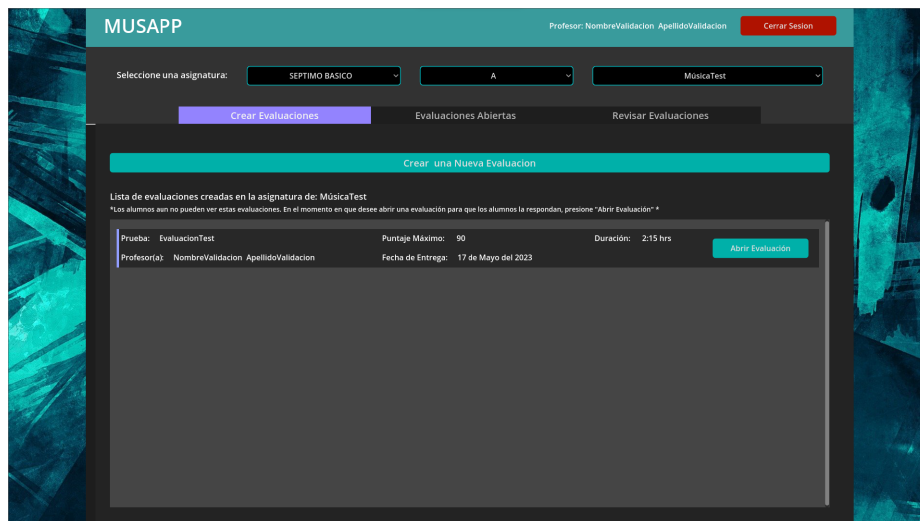


Figura 4.26: Docente: Pantalla principal

Al momento de presionar, crear una nueva asignatura, el docente verá una pantalla con todas configuraciones asociadas al nombre de la evaluación, el sistema de notas y los tiempos de respuesta del alumno (Figura 4.27)

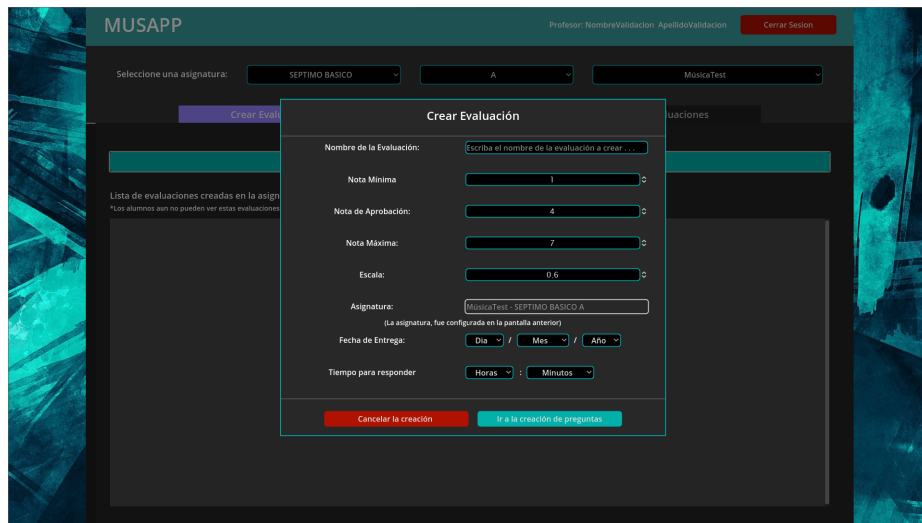


Figura 4.27: Docente: Crear evaluación

Al momento de crear una evaluación el docente verá una pantalla muy similar a la diseñada para este propósito (Figura 3.14), resultando en la Figura 4.28.

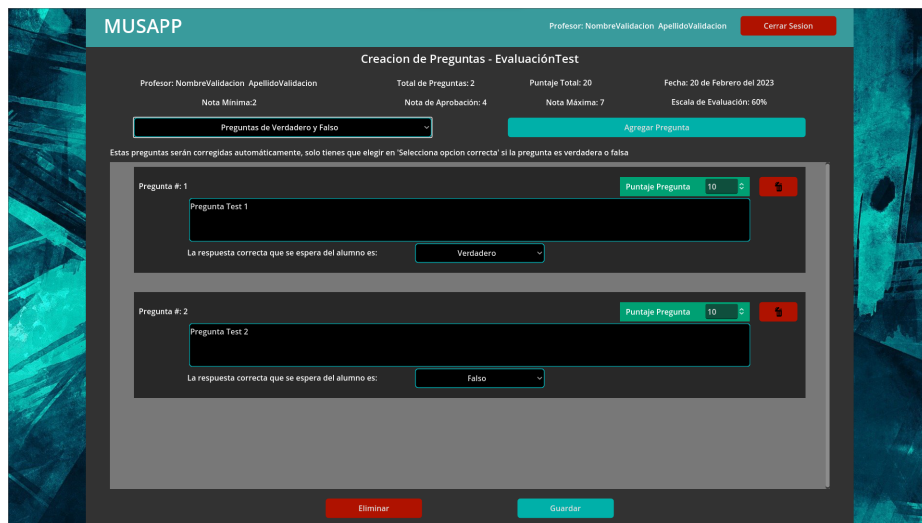


Figura 4.28: Docente: Creación de preguntas

Al revisar una evaluación el profesor verá el listado de alumnos que respondieron (y que no, a los que le podrá asignar la nota mínima como muestra la Figura 4.29).

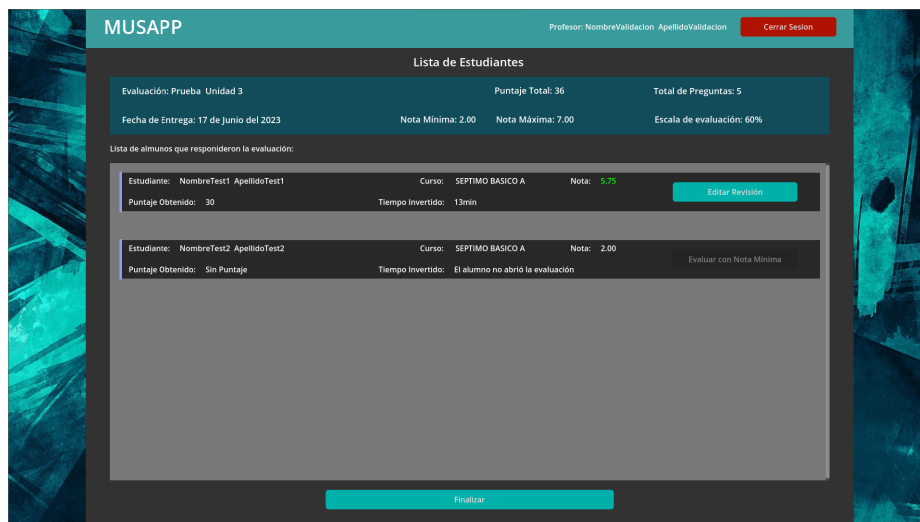


Figura 4.29: Docente: Lista de alumnos que respondieron una evaluación

Finalmente, al momento de revisar la evaluación de un alumno, la interfaz es la siguiente (Figuras 4.30 y 4.31).

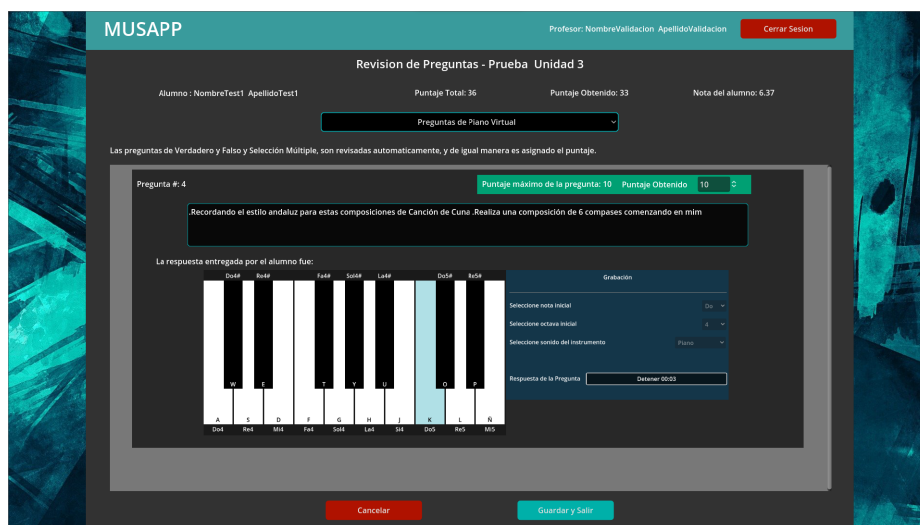


Figura 4.30: Docente: Revisión de una pregunta de piano virtual

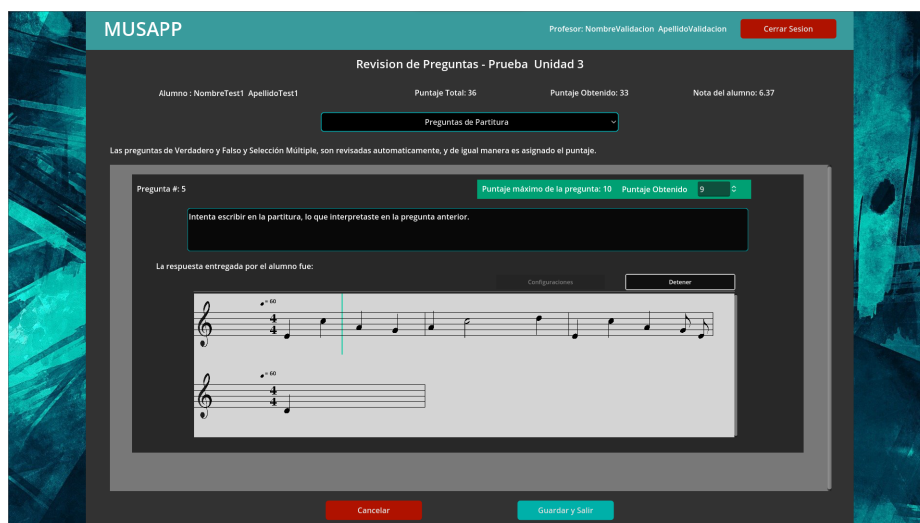


Figura 4.31: Docente: Revisión de una pregunta de partitura

4.6.4. Administrador

El administrador tiene la siguiente pantalla de inicio (también con un color distinto) en donde puede ver que puede entrar a la secciones: Alumnos cursos y asignaturas, teniendo seleccionada por defecto la de profesores.

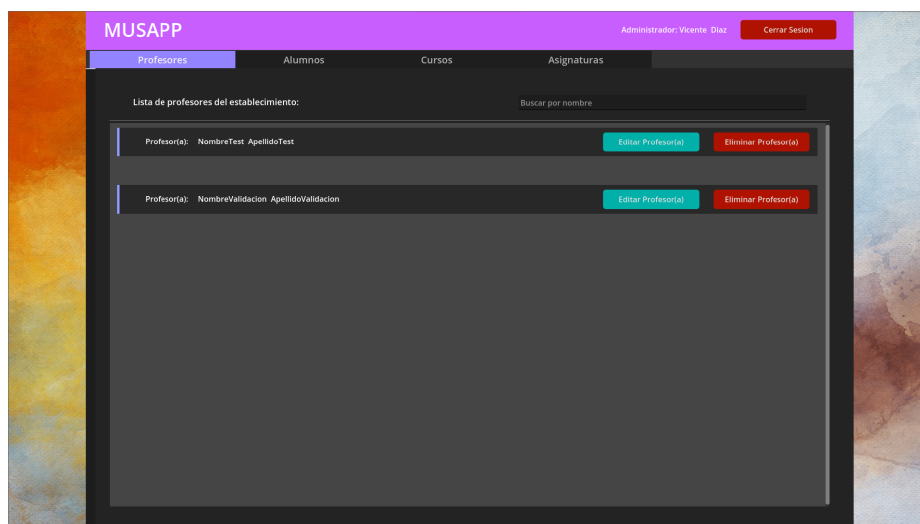


Figura 4.32: Interfaz Administrador: Pantalla Principal

El administrador dentro de sus funciones puede editar un usuario o crear una asignatura.

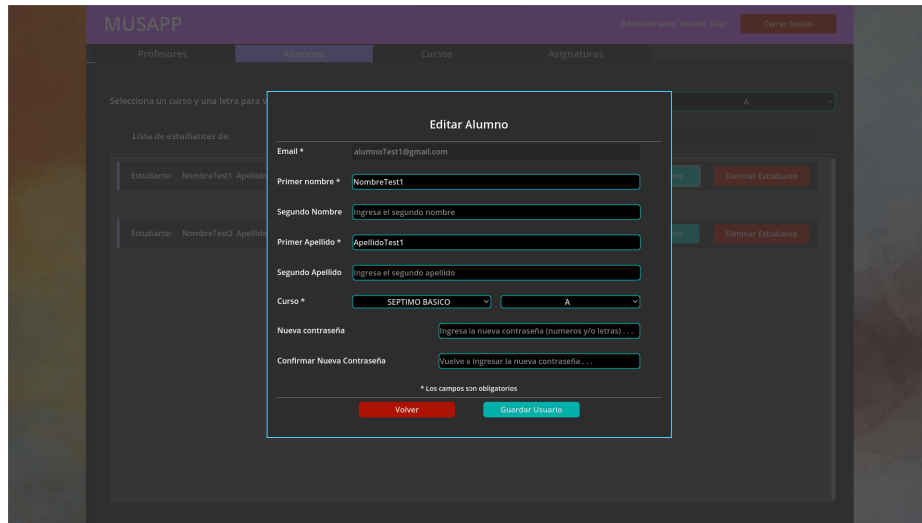


Figura 4.33: Interfaz Administrador: Editar un Alumno

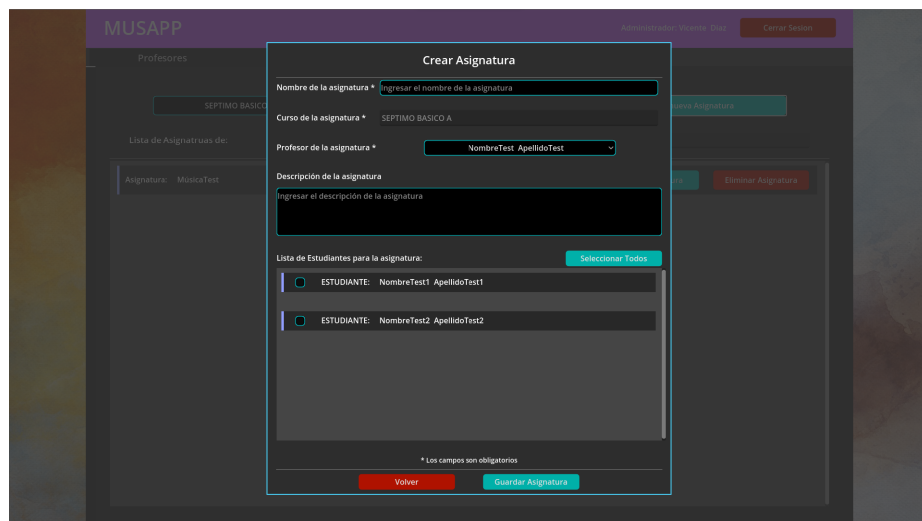


Figura 4.34: Interfaz Administrador: Crear Asignatura

Capítulo 5

Validación

5.1. Validación del piano virtual

Para validar el funcionamiento del piano virtual, se realizaron 9 pruebas de usabilidad moderada a 5 alumnos de séptimo y octavo básico, 3 alumnos de pedagogía en música y un profesor de educación básica. A medida que los usuarios utilizaban el piano, se fue recopilando información de cómo interactuaban con la plataforma, si algo les costó entender o si necesitaron ayuda para realizar alguna acción con el piano virtual. Adicionalmente al finalizar, se les hizo algunas preguntas para registrar sus impresiones y comentarios de la herramienta.

5.1.1. Validación de Piano Virtual: Alumnos

La versión presentada a los estudiantes fue la siguiente:

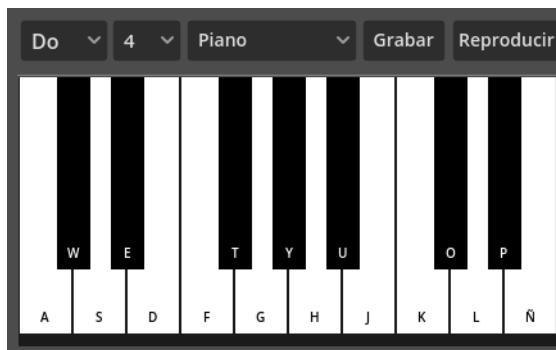


Figura 5.1: Primera versión visual del piano

En primer lugar, se les mostró el piano sin ninguna explicación de cómo se utilizaba. Solo se les mencionó que era interactivo y que lo intentarían hacer funcionar. Esto se realizó para ver si comprendían que las letras representaban las teclas del teclado o si se les hacía más intuitivo usar el touchpad/mouse.

¿Al alumno le resultó sencillo identificar como tocar el piano con el teclado del computador?
5 respuestas



Figura 5.2: Encuesta a Alumnos: uso de teclado en el piano virtual

Se puede notar que a ninguno le tomo más de un par de segundos entender cómo hacer sonar el piano con el teclado.

¿Al alumno le resultó sencillo identificar como tocar el piano con el mouse/touchpad del computador?
5 respuestas

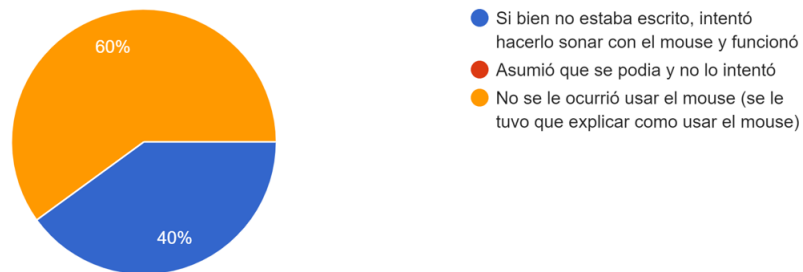


Figura 5.3: Encuesta a Alumnos: uso de mouse en el piano virtual

Por otro lado, no a todos se les hizo intuitivo, o se les ocurrió usar el mouse, incluso a muchos se les tuvo que explicar primero que se podía, como se puede apreciar en la Figura 5.3

Luego se les pidió que utilizaran la interfaz en la parte superior del teclado, y se les consultó si entendían qué hacían cada uno de los botones o si les gustaría ver un texto descriptivo que los ayude a entender. Si bien los alumnos en su mayoría mencionaron que no era necesario, al tener algunos alumnos que les costó un poco entender para qué servía cada botón, se prefirió modificar la interfaz del piano para que cada configuración tenga una pequeña descripción.

¿Cree que es necesario que la interfaz para cambiar la nota inicial deba ser más descriptiva?
5 respuestas

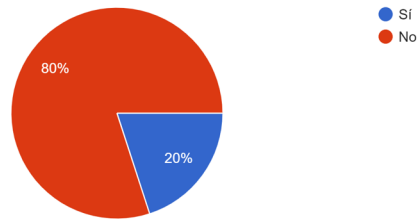


Figura 5.4: Encuesta a Alumnos: Interfaz de nota

¿Cree que es necesario que la interfaz para cambiar el índice de la octava de la escala necesite ser más descriptiva?
5 respuestas

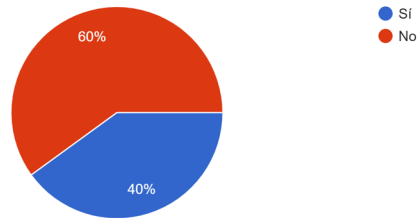


Figura 5.5: Encuesta a Alumnos: Interfaz de octava

¿Cree que es necesario que la interfaz para cambiar el instrumento necesite un título descriptivo?
5 respuestas

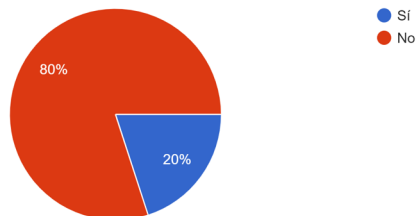


Figura 5.6: Encuesta a Alumnos: Interfaz de instrumento

A continuación, se les mencionó que el piano tenía la capacidad de grabar y reproducir lo grabado, se les pide que investiguen la plataforma hasta conseguirlo.

¿El alumno pudo grabar y reproducir una melodía que tocó en piano?

5 respuestas

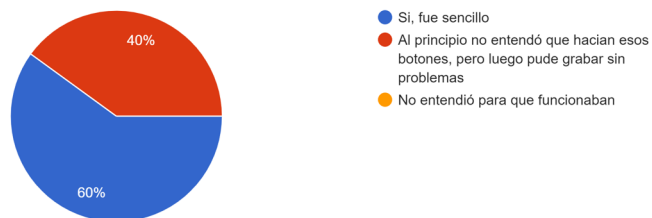


Figura 5.7: Encuesta a Alumnos: Facilidad para grabar

Finalmente se les pidió que, de tener alguna sugerencia, la mencionarán y se registraron la siguientes.

Espacio para dar sugerencias o preguntas acerca del piano

5 respuestas

Que aparecieran las notas facilitaría aprender

Ninguna

Me gustó lo de las letras y no se me hubiera ocurrido crearlo.

Nada, está bien

Si este piano es un formato pequeño, podría visualizarse mediano o grande en la pantalla.

Figura 5.8: Encuesta a Alumnos: Sugerencias

Donde se menciona un tamaño de piano más grande y que aparecieran las notas del piano.

5.1.2. Validación de Piano Virtual: Estudiantes de Pedagogía y Profesores

Se decidió probar con profesores y estudiantes de pedagogía en música, pues se consideró importante saber la opinión de futuros docentes, y adicionalmente al ser jóvenes, su acercamiento hacia tecnologías y dispositivos tecnológicos tiende a ser mayor.

Se siguió el mismo orden y metodología que con los alumnos. En primer lugar se les pidió que interactuaban con el piano sin explicarles cómo. Los resultados que se muestran en las Figuras 5.9 y 5.10 muestran mucha similitud con respecto a las obtenidas con los estudiantes:

¿Te resultó sencillo identificar como tocar el piano con el teclado del computador?

4 respuestas

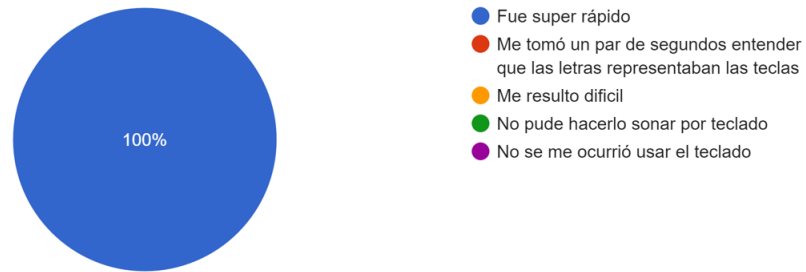


Figura 5.9: Encuesta a Docentes: uso de teclado en el piano virtual

¿Te resultó sencillo identificar como tocar el piano con el mouse/touchpad del computador?

4 respuestas

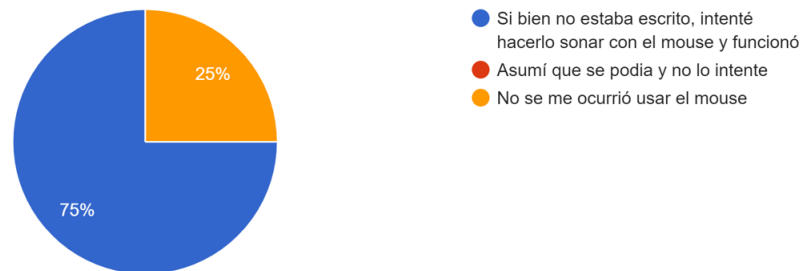


Figura 5.10: Encuesta a Docentes: uso de mouse en el piano virtual

Luego, se les pidió que utilizaran la interfaz de configuraciones. Las apreciaciones de los pedagogos se muestran a continuación.

¿Crees que es necesario que la interfaz para cambiar la nota inicial necesite ser más descriptiva?
4 respuestas

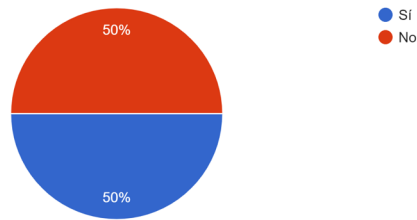


Figura 5.11: Encuesta a Docentes: Interfaz de nota

¿Crees que es necesario que la interfaz para cambiar el índice de la octava de la escala necesite ser más descriptiva?
4 respuestas

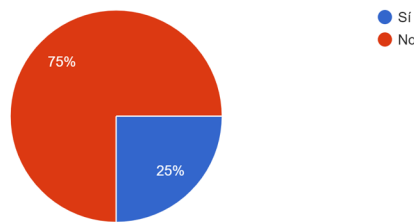


Figura 5.12: Encuesta a Docentes: Interfaz de octava

¿Crees que es necesario que la interfaz para cambiar el instrumento ser más descriptiva?
4 respuestas

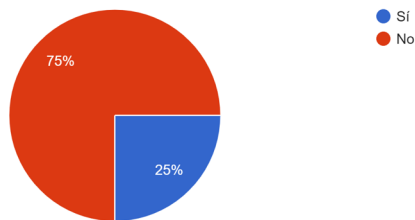


Figura 5.13: Encuesta a Docentes: Interfaz de instrumento

De igual manera que con los estudiantes, se ve una tendencia por entender sin mayores problemas para que servía cada botón.

Finalmente se les consultó si consideraban necesario que los alumnos pudieran ver los nombres de cada nota en las teclas del piano.

Como se puede ver en la Figura 5.14 resultados difirieron bastante, por lo tanto se decidió que los docentes al momento de crear una pregunta que se deba responder con el piano virtual, seleccionen una opción que configure si el piano va a tener las notas o no para dicha pregunta.

Tomando todos estos puntos en cuenta, se modificó el diseño resultando en la Figura 5.15

¿Te gustaría darle la opción a tus alumnos de que puedan ver las notas del piano?

4 respuestas

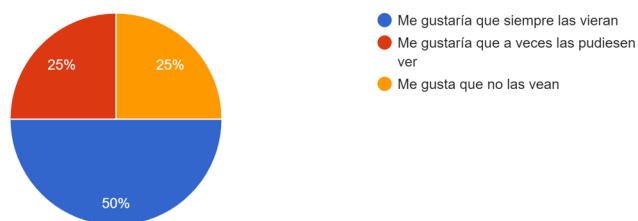


Figura 5.14: Encuesta a Docentes: Visualización de las notas en el piano

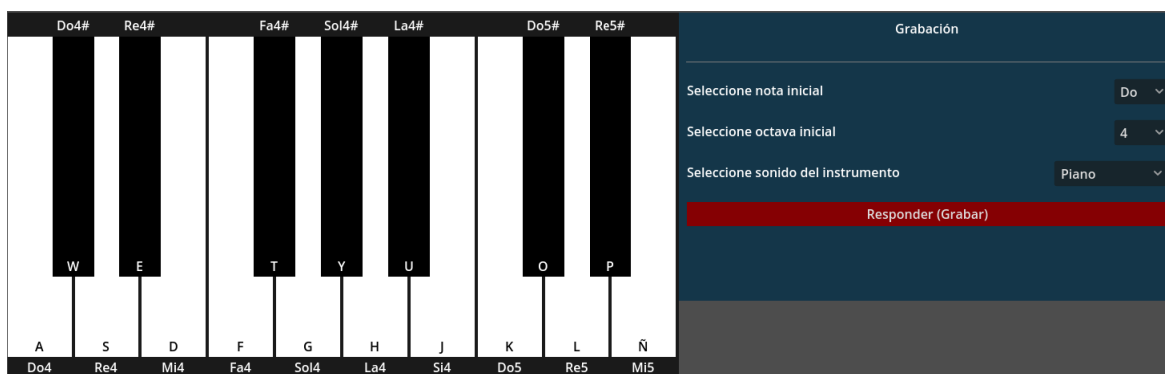


Figura 5.15: Versión final del piano

5.2. Validación de evaluaciones para estudiantes

Para validar las evaluaciones con los estudiantes, nuevamente se hizo una prueba de usabilidad moderada, pero esta vez a 3 usuarios. Al ser una validación mas larga (el tiempo que duró la prueba con cada uno fue de aproximadamente 15 minutos), se fueron anotando comentarios y acciones importantes del usuario a medida que la evaluación iba avanzando.

En un principio se evaluó con dos usuarios, uno de séptimo y otro de octavo. Al tener ambos nulo conocimientos de partituras, se decidió hacer una tercera validación con un usuario de primero medio, pero desafortunadamente, tampoco había tenido acercamiento alguno a una partitura.

Es importante mencionar que para esta validación el piano virtual presentado es el que se puede apreciar en la Figura 5.1 mientras que el sistema de partituras es el presentado a continuación en la Figura 5.16.

Para realizar esta validación se generó una evaluación con dos preguntas de cada tipo, para ver como los alumnos la respondían, si podían utilizar los diferentes instrumentos evaluativos de manera natural, o si tenían que recibir explicaciones para poder responder la evaluación. Es importante destacar que en la primera pregunta de piano virtual se les pedía tocar y en la segunda configurar el piano antes de tocar. De igual manera, en las preguntas de partituras se les pidió solo escribir en la primera y en la segunda configurar la partitura e intentar tocar bemoles, sostenidos y becuadros. Adicionalmente se quiso evaluar si los alumnos podían

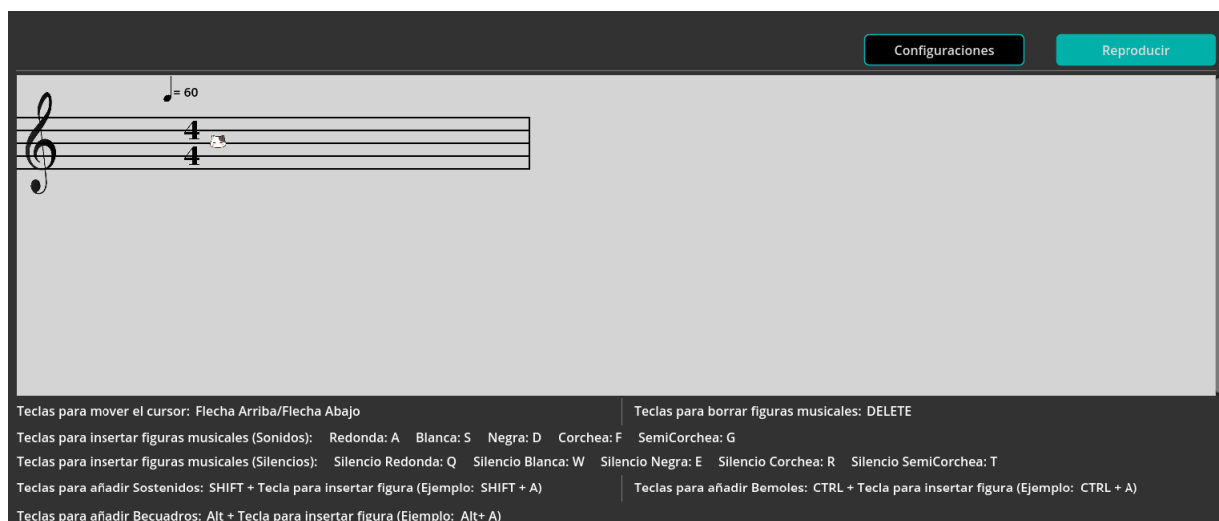


Figura 5.16: Primera versión visual del sistema de partituras

moverse por la plataforma. Para esto, se les presentó la aplicación desde el inicio de sesión. Se les creó una cuenta con un alumnoTestX@gmail.com y se les entregó el computador para que iniciaran desde la pantalla de inicio de sesión. Luego, se les pidió ir a una asignatura llamada “MusicaTest” y rendir la evaluación de nombre “EvaluacionTest”. A continuación se detallan los puntos más importantes rescatados durante las pruebas de usabilidad a los tres usuarios (Para ver la transcripción completa de las anotaciones por usuario ver el Anexo E.1)

- Todos tres usuarios lograron iniciar sesión, buscar la asignatura e iniciar la Evaluación, de manera inmediata.
- Todos los usuarios pudieron responder las preguntas de Verdadero y Falso, Selección de alternativas y Desarrollo, de forma intuitiva, no fue necesaria ninguna explicación.
- Todos los alumnos utilizaron el teclado para hacer sonar el piano. Solo a uno no se le ocurrió utilizar el botón grabar para registrar la respuesta. Todos en la pregunta dos de piano virtual pudieron configurar el piano, tal como se detallaba
- Todos pudieron insertar y eliminar notas después de unos segundos leyendo las instrucciones, aún cuando ninguno había escrito o leído una partitura anteriormente
- Les costó leer las instrucciones, las encontraban desordenadas (Ver Figura 5.16) y consideraban que era demasiada información en un corto periodo de tiempo
- Ningún usuario conocía las teclas SHIFT, Ctrl y Alt. Solo uno presionó Ctrl y Alt, luego de haberlas buscado un rato en el teclado. Todos solicitan una representación visual de las teclas, sobre todo de SHIFT
- Ningún usuario encontró por su cuenta el botón configurar de las partituras, se les tuvo que indicar.
- Todos los usuarios comentaron que el piano lo pudieron usar de forma natural, pero para la partitura necesitarían al menos una clase práctica o ver ejemplos de uso antes de manejarse fluidamente.

Al notar que a los usuarios mencionaron que las instrucciones estaban desordenadas, se procede a reordenar las filas, se añade una imagen explicando donde se encuentran las teclas SHIFT, Ctrl y Alt, y se agrandaron los botones de reproducir y configurar, lo que resultó en la Figura 5.17

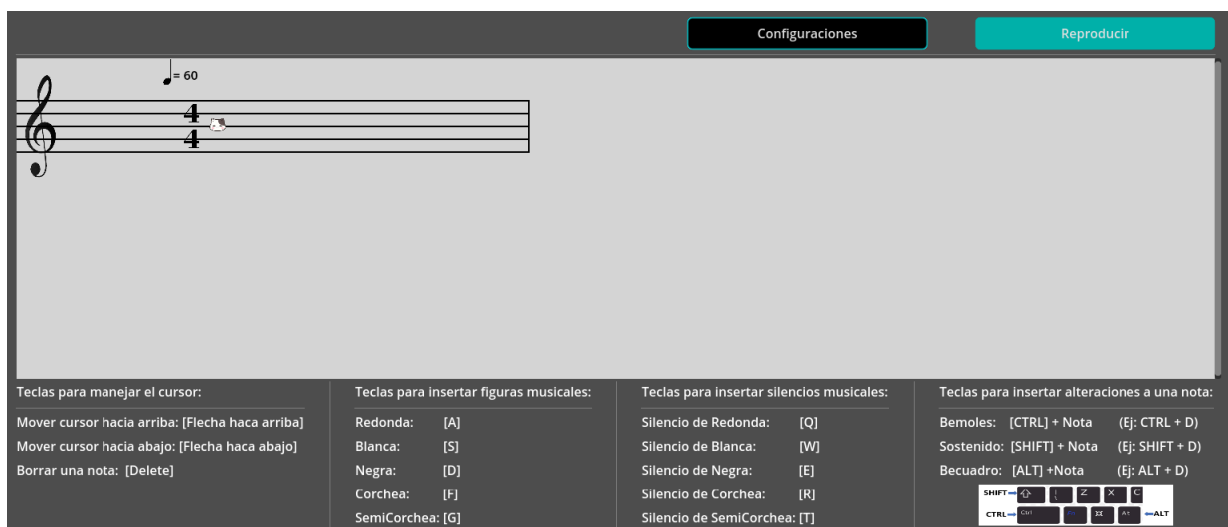


Figura 5.17: Versión final del sistema de partituras

5.3. Validación para creación y revisión de evaluaciones por profesores

Se hizo una evaluación en conjunto con un profesor de artes de educación básica de primer y segundo ciclo, en donde se revisó la aplicación en conjunto desde su creación de usuario, hasta la revisión de evaluaciones.

En la pantalla de creación de usuarios logró identificar que habían campos opcionales (segundo nombre, segundo apellido), no le resulta intuitivo notar que al seleccionar ocupación como profesor, no debe seleccionar un curso, luego de intentarlo y notar que los botones estaban deshabilitados, leyó la nota que mencionaba que los profesores no debían seleccionar un curso.

Cuando inicia sesión menciona que preferiría leer “Ingresar email”, en vez de “Ingresar usuario”, pues tuvo un momento de duda al momento al rellenar este campo.

Una vez dentro de su ambiente como docente, identifica las pantallas de crear evaluación, revisar evaluaciones y finalizar evaluación. Pero no tenía asignaturas asociadas, por ende, en modo administrador, se le asignó una asignatura con dos alumnos para testear.

A continuación se le solicitó crear una evaluación, pudo cambiar la nota mínima correctamente y crear preguntas, el botón para finalizar la creación decía “guardar”, y el docente mencionó que se prestaba para confusión respecto a guardar preguntas, por ende se cambia a “Finalizar evaluación”. Además menciona que le gustaría poder añadir los objetivos de la evaluación, pero esa característica escapaba de los alcances definidos para la evaluación

Al momento de crear la evaluación, el usuario pregunta si es posible insertar un audio, pues el eje apreciativo lo quería evaluar a partir de la experiencia auditiva de una pieza musical, pero, no se diseñó la aplicación para admitir archivos multimedia, por ende se le

menciona la imposibilidad de esto. Decide entonces ingresar la letra de una canción para que la analicen. El resto de la creación de preguntas resultó sin problemas, creando dos preguntas de verdadero y falso, una de desarrollo, una de piano virtual y una de partituras.

Luego se toma el rol de estudiante y se responde la prueba frente al docente, se le muestra cómo el estudiante ve las preguntas y los instrumentos evaluativos, y responde muy positivamente al notar que la partitura reproduce una nota al momento de insertarse. Se responde cada una de las preguntas para su posterior revisión.

Luego el profesor evalúa las respuestas entregadas, notando que la sección de verdadero y falso es corregido automáticamente, lo que agradece, pues reduce el tiempo de revisión. Luego evalúa la de respuesta de desarrollo sin problemas, y continua con el piano, donde logra reproducir lo tocado anteriormente, de igual manera reproduce lo escrito en la partitura. Asigna los puntajes a cada respuesta, y descubre que la nota se calcula automáticamente con los parámetros ingresados al momento de crear la evaluación. Así, finaliza la revisión y el test de usabilidad

Se le solicita una retro alimentación oral de la aplicación y menciona los siguientes puntos

1. Comentarios negativos

- (a) Falta un poco de espacio para escribir y mostrar las preguntas,
- (b) Cambiar “Ingresar usuario” por “Ingresar email”, al iniciar sesión,
- (c) Cambiar “Guardar” por “Finalizar evaluación”, en la pantalla de creación de evaluaciones,
- (d) Faltó la posibilidad de agregarle contenido multimedia a la aplicación,

2. Comentarios positivos

- (a) Menciona que: “La partitura permite la creación musical, entrega mucha estimulación hacia los sentidos del estudiante y el hecho que sea auditiva facilita la escritura (de partituras) para quien no tenga tanto dominio teórico de música, permitiéndole entrenar su oído y le entrega motivación para querer aprender cómo escribir apropiadamente”,
- (b) Permite recabar evidencias del aprendizaje del estudiante de múltiples formas,
- (c) Al estudiante le amplía la experiencia musical expresándose por distintos medios,
- (d) Revisar una evaluación le resulta más sencillo que revisarla en papel. Asegura que ahorraría tiempo, el cual podría dedicar a entregar una mejor retroalimentación a los estudiantes, para poder mostrarle en que se equivocó y en qué mejorar,
- (e) Considera que se pueden evaluar ambos ejes con la aplicación, pero desearía complementar el eje apreciativo con la carga de archivos audiovisuales,
- (f) Permite mayor transparencia por parte de los estudiantes y docentes, pues no permite a ningún usuario modificar las respuestas, por ende deja un registro verídico del desarrollo de la evaluación.

5.4. Validaciones de rendimiento, uso de recursos computacionales y envío de datos

5.4.1. Recursos del computacionales

De haberse guardado las 88 notas del piano este hubiera generado un peso de 118MB en disco (Ver Anexo A). Por otro lado en MusApp se pueden reproducir 11 instrumentos distintos con solo 8,22MB, incluso, el peso total de la aplicación de escritorio es de solo 250MB en disco, y un uso de memoria RAM de 150MB en promedio. Se considera entonces, que el diseño e implementación del Sistema de almacenamiento y reproducción de audios fue un éxito, pues el uso de recursos en un dispositivo es muy bajo.

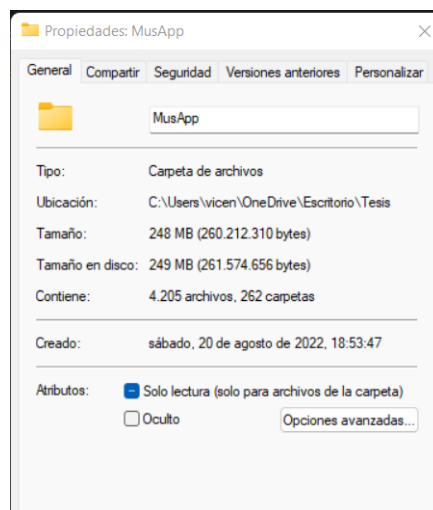


Figura 5.18: Uso en disco de MusApp

Nombre	Estado	50% CPU	83% Memoria
> Administrador de tareas		0,7%	34,9 MB
> Bloc de notas (3)		0%	16,1 MB
> Explorador de Windows		3,8%	153,9 MB
Godot Engine		6,2%	111,0 MB
MusApp (DEBUG)			

Figura 5.19: Uso de Ram

5.4.2. Envío de Datos

A nivel de envío de datos cada Pregunta tiene un peso de 80B + el tamaño del texto de la pregunta,

cuyo máximo es 500B, pues se admiten preguntas de hasta 500 caracteres. Por ende cada pregunta tendrá una cota superior de 500B. Asumiendo 30 preguntas por evaluación, se tendría un peso de 14,6KB.

```
{"id":5, "quiz_id":4, "number":2, "score":0, "correct_asnwer":"","option":""}
```

Figura 5.20: Pregunta vacía, siendo que algunas pesan un poco más, se deja en promedio 80B

Por otro lado las respuestas tienen un tamaño de 87B + el tamaño de la respuesta.

```
{"id":5, "question_id":3, "comments":"","score":0, "answer":"","answered_quiz_id":3}
```

Figura 5.21: Respuesta vacía, tiene 87 caracteres de 1B cada uno

El tamaño de la respuesta para las preguntas de piano y partitura es variable y son las que realmente aportan peso, debido a que depende de la cantidad de notas. Para estimar un promedio, se tocó 15 veces el piano durante un minuto, a veces tocando pocas notas, y a veces tocando acordes (3 o más notas en paralelo), aumentando y disminuyendo la frecuencia de pulsación en las distintas muestras. alcanzando un promedio de: 5,83KB.

Mientras tanto, el sistema de partituras al no tener notas en paralelo presenta menos peso, pero al ser un diccionario el que se crea por nota, aumenta considerablemente su tamaño, para generar una cota superior promedio, se colocaron 16 semicorcheas por compás (en un compás de 4/4 que es el más común), y por compás se alcanzó un tamaño de 1.46KB, que en una partitura con un tempo de 120ppm, equivale a solo 2 segundos de partitura (en este caso una semicorchea equivale a 0.125s de nota). Por ende, un minuto de grabación alcanzaría los 46KB (equivalente a 30 compases). Se dejará entonces una cota de 50KB por respuesta.

Uniendo todos estos datos, la cota superior de una respuesta, se estima en los 1.4MB (aproximadamente 30 respuestas de partitura de 50KB, que son las más pesadas). Para tener un punto de referencia del uso de datos en una página web, solo abrir Google (sin hacer ninguna búsqueda), carga 2,9MB de datos.

5.5. Comparación con los objetivos iniciales

Para poder corroborar que el objetivo general se cumplió, se hará un recorrido por cada uno de los objetivos específicos y se compararán con el trabajo realizado en este informe.

5.5.1. Diseños

Uno de los ejes de los objetivos específicos fue diseñar funcionalidades para una aplicación de escritorio que:

1. “Genere instrumentos de evaluación que encapsulen los ejes de Crear y Aprender”: Se logró diseñar los instrumentos evaluativos, para crear (el piano y la partitura) y apreciar (respuestas de desarrollo, alternativa y verdadero y falso). Pero, el diseño de

apreciaciones contempló solo la respuesta, y no se contempló que es de gran importancia presentarle al estudiante un elemento para apreciar, tal como se pudo ver en la validación de la Sección 5.3 donde solo fue posible medir su capacidad apreciativa con texto y conocimientos previos.

2. “Permita a los docentes de música: crear, aplicar y corregir dichos instrumentos evaluativos”: Se logró diseñar un modelo de datos y pantallas, que permiten al docente navegar por estas tres funcionalidades.
3. “Permita a estudiantes: responder y recibir retroalimentación de las evaluaciones aplicadas”: Se logró diseñar un modelo de datos y pantallas, que permiten al estudiante navegar por estas dos funcionalidades .
4. “Permita a los administradores: crear y editar Niveles, Cursos y Asignatura”: Se logró diseñar un modelo de datos y pantallas que permiten a los administradores poder crear y editar los tres modelos mencionados .
5. “Sea óptima y no consuma grandes recursos computacionales”: Se ideó un sistema de almacenamiento y reproducción de notas que reducía los costos de almacenamiento de audios tanto local como remoto.
6. “Sea simple de utilizar, cómoda y estéticamente agradable para los distintos usuarios”: Con las validaciones de usabilidad (Secciones 5.1.1, 5.2 y 5.3) se logró comprobar que este objetivo se cumplió a cabalidad, tanto para estudiantes como docentes.

5.5.2. Implementación

El siguiente objetivo a cumplir fue implementar:

1. “Visualizaciones de los instrumentos de evaluación y su interacción con el usuario”: Se implementaron pantallas que permitían crear, responder y corregir una evaluación, validado en las Secciones 5.2 y 5.3.
2. “Un ambiente exclusivo del ecosistema educativo para el uso de la aplicación por parte de los docentes”: Logrado, un docente iniciando sesión tenía un sistema distinto al de estudiantes y administradores (4.26). Y se validó su funcionamiento (Sección 5.3).
3. “Un ambiente exclusivo para el uso de la aplicación por parte de los estudiantes”: Logrado, un estudiante iniciando sesión tenía un sistema distinto al de docentes y administradores (4.25). Y se validó su funcionamiento (Sección 5.2).
4. “Un ambiente exclusivo para el uso de la aplicación enfocado en los administradores”: Logrado, un estudiante iniciando sesión tenía un sistema distinto al de estudiantes y docentes (Sección 4.32). Pero no fue validado más allá de su uso para generar el resto de las validaciones (crear cursos y asignaturas para que los docentes y estudiantes trabajaran).

5. “Un sistema de datos común para todos los usuarios”: Logrado, la plataforma se conectó correctamente con una base de datos, utilizando todos los modelos diseñados (Sección 3.4), que permitieron tener interacciones entre docentes, estudiantes, niveles, cursos y asignaturas. Adicionalmente sus interacciones se basaban en la publicación, respuesta y corrección del Sistema de evaluaciones, tal como se muestra en (Ver Anexo B.2.5), y se valida en (Sección 5.2 y 5.3).
6. “Optimizaciones al sistema para tener una aplicación que no requiera recursos computacionales elevados”: Logrado, tal y como se describió en la sección (Sección 5.4), los recursos computacionales que requería la plataforma eran de muy bajo consumo tanto en el dispositivo, como en la transferencia de datos con el servidor.

Con esta información, es posible asegurar que el objetivo de “diseñar, implementar y validar un prototipo de aplicación de un ecosistema virtual con bajos requerimientos de conectividad y de recursos computacionales, que permita medir el conocimiento de los alumnos de asignaturas musicales, a partir de preguntas creadas por el docente, enfocadas a evaluar los ejes de creación y apreciación” (Sección 1.1.1), fue logrado de manera satisfactoria, solo faltando haber diseñado e implementado la carga y visualización de archivos audiovisuales en las preguntas, tema que puede quedar expuesto ser desarrollado y validado en un trabajo futuro.

Capítulo 6

Conclusión

El trabajo realizado fue motivado por las grandes posibilidades que tienen las ciencias de la computación como un apoyo a la enseñanza. Adicionalmente el interés artístico y musical permitió tener las herramientas necesarias para sacar adelante un proyecto que requería tener un amplio conocimiento tanto computacional, musical y estético. Fue de sumo interés desarrollar una plataforma de fácil acceso y uso, cuya meta fue que la única dificultad que presente para los usuarios sea la propia complejidad de crear y responder una evaluación. A continuación, se detalla los logros conseguidos durante el proyecto (Sección 6.1) y lo aprendido a medida que se iba desarrollando (Sección 6.2). Finalmente, se espera que MusApp siga creciendo en el futuro, pues los comentarios que recibió de los docentes y estudiantes fueron altamente positivos, y las posibilidades que entrega un ecosistema virtual es amplio e interesante (Sección 6.3).

6.1. Logros

En primer lugar tal y como se detalló (Sección 5.5), los objetivos fueron cumplidos, pero con algunos detalles. Faltó diseñar características en la aplicación que acompañaran de manera más completa el eje apreciativo, cumple con tener la posibilidad entregar conocimiento apreciativo, pero se puede seguir iterando para una herramienta más versátil. Pero, aún sin obviar esto, la plataforma sí cumple con entregar herramientas que permitan medir los ejes de creación y apreciación.

Si bien se considera que los logros del proyectos son lo suficientemente completos a nivel de prototipo, si bien aún queda un largo camino para ser integrado a la sala de clases, se considera útil para experimentar y medir el conocimiento de los alumnos. Un ítem importante y que no fue considerado dentro de los objetivos de este informe, fue a la configuración de un servidor que alojara el backend de MusApp. Así, todas las validaciones que se hicieron con usuarios fueron desarrolladas en ambientes locales.

6.2. Aprendizajes

En cuanto a lo que se aprendió durante el proceso, en primer lugar se destaca de vital importancia no subestimar el tiempo de diseño. Se le entregó mucho tiempo al diseño del sistema de audios, lo que permitió tener una implementación sencilla y súper potente, que facilitó en gran medida el desarrollo del resto de elementos auditivos de la implementación. Además también fue dedicado gran parte del inicio del proyecto a generar un modelo de datos y relaciones lo suficientemente robusto, para poder implementar el ecosistema educativo sin tener que realizarle algún tipo de modificación.

En segundo lugar, que un desarrollo ordenado, en todos los diversos ámbitos que la palabra lo permite, ayuda muy positivamente a la eficiencia, mutabilidad y escalamiento de un proyecto. En MusApp fueron separadas cada una de las vistas y las funcionalidades que se pensaban desarrollar y con ello, siempre se pudo ver un panorama general de como iba evolucionando la aplicación. Además, se crearon horarios de trabajo, se estipularon días de descanso y se organizó un orden de trabajo diseño-implementación, para así siempre generar variaciones del trabajo y mantener un desarrollo dinámico.

En tercer lugar, definir los alcances de manera anticipada y clara. En este caso, MusApp trató de abarcar diversos ámbitos, lo que implicó muchas horas de estudio y de desarrollo, pues si bien se tenían conocimientos musicales, la traducción de dichos conocimientos a las áreas computacionales fue una tarea laboriosa. Adicionalmente, a medida que se fueron terminando objetivos, siempre se buscaron nuevas metas de desarrollo por conseguir, lo que resultó en un ritmo acelerado dentro de los tiempos designados al proyecto y que utilizó mucho tiempo en la escritura de código y poco tiempo para validaciones, escritura del informe y la falta de tiempo para hacer un deploy del servidor. Por ende, se considera importante para proyectos futuros delimitar de mejor manera los alcances.

Finalmente comentar que, trabajar en áreas educacionales fue una experiencia muy enriquecedora. Los docentes que ayudaron a validar, dieron retroalimentación muy útil y comentarios positivos que permitieron siempre tener una línea de desarrollo cercana a lo que los usuarios querían, que permitió dilucidar objetivos claros, mezclando sus necesidades y requerimientos, con las ideas que tenía el mismo desarrollador, permitiendo complementar conocimientos. Por ende, la validación con usuarios es un proceso por el cual hay que pasar, si lo que se pretende es desarrollar software de calidad

6.3. Trabajo Futuro

A partir de este momento MusApp quedará a disposición de la comunidad, pues desde la concepción de este proyecto, siempre estuvo la idea de intentar ser un aporte a nivel social y educativo. Es por ello que quedará en un repositorio público con licencia MIT, para que cualquier persona con conocimientos en programación y/o con ideas para seguir haciendo crecer este proyecto, sea libre de poder hacerlo. A continuación se presentarán diversas ideas para ampliar y mejorar dicho ecosistema a nivel musical, artístico y educativo

6.3.1. Almacenamiento de piezas musicales

Actualmente, una grabación de piano virtual o una partitura se guardan en formatos JSON. Este archivo se guarda en la base de datos, pero el alumno no tiene acceso al dicho archivo una vez envía su evaluación. Para corregir esto, una buena idea sería poder guardar de manera local un archivo .txt que contenga el JSON generado, y luego poder cargar este archivo a MusApp para poder reproducirlo. A nivel de interfaz, para guardar el .txt con el JSON de la grabación, es necesario integrar en el piano y el sistema de partituras la opción de guardar (un botón) y utilizar el navegador de archivos del propio motor Godot para indicar el nombre de la grabación y la carpeta donde se guarda. Por otro lado, el lugar idóneo para agregar la carga de archivos es en el piano virtual y el sistema de partituras presentes en el inicio de sesión (ver Figura 4.21), pues esa pantalla no necesita conexión a internet, por ende alumnos y profesores podrán usar este sistema de manera offline.

6.3.2. Portabilidad

Si bien el sistema fue desarrollado para Windows, se eligió el motor de Godot, pues permite sistemas multiplataforma, se puede generar un ejecutable para Linux [3] y MacOS [4], siguiendo las instrucciones en los enlaces de la bibliografía. Se hicieron algunas pruebas en Linux, pero no de manera metódica como para asegurar su correcto funcionamiento, por ende queda como un trabajo para hacer más adelante.

6.3.3. Inserción de recursos audiovisuales

Una importante mejora que debe ser agregada, es que tanto las preguntas como respuestas puedan añadir contenido audiovisual. Esto tiene gran versatilidad pues puede servir de apoyo para hacer preguntas de carácter apreciativo de una obra, canción, o algún otro elemento artístico. También permitiría al alumno poder hacer contenido audiovisual, enviar audios interpretando algún instrumento musical, o fotografías de algún dibujo, etc. Se considera una característica que puede favorecer enormemente a MusApp.

6.3.4. Inserción de un metrónomo al piano virtual

Un elemento interesante a integrar para facilitar la creación dentro del piano virtual es un metrónomo. Se idea como una opción dentro de la configuración del piano, en donde se pueda elegir tanto el tempo como el compás, y que comience a sonar al momento de grabar, así el alumno podrá tener una pauta rítmica que seguir para poder componer o reproducir una pieza musical con mayor facilidad.

6.3.5. Agregar el piano virtual y el sistema de partituras dentro de una pregunta

Si bien el estudiante puede responder con estas herramientas, es una buena idea que el docente también pueda utilizar tanto el piano como la partitura y guardar su interpretación o creación dentro de una pregunta. Esto podría apoyar a medir el eje apreciativo de una mejor forma, o al menos, con más recursos.

6.3.6. Entregar una retroalimentación dentro de la plataforma

Añadirle la opción a los estudiantes de poder ver la corrección hecha por el docente y comentarios que haya dejado. Así el alumno tendrá transparencia frente a lo que el docente haya evaluado y lo que él como alumno haya respondido.

6.3.7. Sistema pixel-Art

Extender el sistema de MusApp a las artes visuales es tal vez el siguiente paso. El pixel-Art no requiere de tabletas digitalizadoras o una pantalla táctil gigante, por lo que puede ser una buena opción de bajo presupuesto para que los alumnos puedan crear dibujos y pinturas.

6.3.8. Sistema de ecuaciones y gráficos

Tomando como referencia el sistema educativo de MusApp, se puede considerar crear aplicaciones similares (o extender la misma aplicación), para integrar otras áreas del conocimiento es una idea latente. Así, los alumnos y docentes podrían escribir y leer fórmulas matemáticas en la plataforma y desarrollar gráficos interactivos, lo que podría apoyar el conocimiento a partir de la experimentación visual de las matemáticas.

6.3.9. Hardware de Piano con Arduino

Debido a que los teclados de membrana que trae un computador no siempre permiten presionar mas de una nota a la vez, se ve la necesidad de desarrollar un hardware que facilite el uso del piano virtual. Además, generando una estética similar al de un piano real, podrá facilitar el aprendizaje de este instrumento, con un costo mucho menor a los teclados tradicionales.

6.3.10. Enseñar a trabajar con una estación de trabajo de audio digital

El manejo de audios en el motor de videojuegos Godot es bastante completo, entregando un filtros, ecualizadores, reverberadores, analizadores de espectro, entre muchos otros. El potencial de poder presentar todo este tipo de herramientas a niños y jóvenes de una manera sencilla y con una interfaz agradable, no debe ser pasado por alto. Es interesante pensar qué podrá llegar a crear un estudiante teniendo las herramientas necesarias y presentadas de manera didáctica. Es una idea ambiciosa pero no por eso hay que dejarla de lado. Además, la creación musical con TICs está presente en los planes y programas de 3 Medio.

6.3.11. Desarrollo de minijuegos educativos

Al estar MusApp basada en un motor de videojuegos, extenderla a áreas más lúdicas es una opción viable de implementar: juegos para estudiar el azar y las probabilidades, trivias educativas, juegos de ritmo estilo Guitar-Hero, juegos para entrenar la memoria o la motricidad fina. La gamificación de la enseñanza es un área de estudio actualmente y MusApp puede ser un aporte en el estudio de nuevas formas de la enseñanza.

Bibliografía

- [1] Easy lms. <https://www.easy-lms.com/es/funciones/creador-de-examenes-online>. Accessed: 2021-10-05.
- [2] Edulastic. <https://edulastic.com>. Accessed: 2021-10-05.
- [3] Exportar en motor de videojuegos godot para sistemas linux. https://docs.godotengine.org/es/stable/tutorials/export/exporting_for_linux.html. Accessed: 2022-01-06.
- [4] Exportar en motor de videojuegos godot para sistemas macos. https://docs.godotengine.org/es/stable/tutorials/export/exporting_for_macos.html. Accessed: 2022-01-06.
- [5] Google forms. <https://docs.google.com/forms/u/0/>. Accessed: 2021-10-05.
- [6] Índice de vulnerabilidad escolar, ministerio de educación. <https://www.junaeb.cl/ive>. Accessed: 2021-12-10.
- [7] Kahoot. <https://kahoot.com/>. Accessed: 2021-10-05.
- [8] Quizalize. <https://www.quizalize.com/>. Accessed: 2021-10-05.
- [9] Quizizz. <https://quizizz.com/>. Accessed: 2021-10-05.
- [10] Repositorio de musapp. <https://github.com/mago-diaz/musapp>. Accessed: 2022-03-27.
- [11] Respecto a la cobertura del programa becas tic 2022. <https://www.mineduc.cl/disminucion-en-la-cobertura-del-programa-becas-tic-2022/>. Accessed: 2022-12-20.
- [12] Tinsol: un dataset de audio de notas musicales aisladas. <https://zenodo.org/record/3685367.ZCFb7nbMJD9>. Accessed: 2022-08-19.
- [13] Unidad de currículum y evaluación ministerio de educación. <https://www.curriculumnacional.cl/portal/Educacion-General>. Accessed: 2021-10-05.
- [14] Sylvia Eyzaguirre, Carmen Le Foulon, Valentina Salvatierra, et al. Educación en tiempos de pandemia: antecedentes y recomendaciones para la discusión en Chile. *Estudios Públicos*, (159):111–180, 2020.

- [15] Nahuelán Jerez, Oscar Daniel, et al. La música en el contexto escolar chileno: estudio exploratorio desde la perspectiva del profesorado. 2019.
- [16] Nuria Díez Latorre and Xavier Carrera Farran. Integración de las tic en los procesos de enseñanza-aprendizaje de la especialidad de pedagogía en los conservatorios superiores de música. *RIITE Revista Interuniversitaria de Investigación en Tecnología Educativa*, pages 40–55, 2018.
- [17] Marcello Gonzalo Chiuminatto Orrego. La investigación en educación musical en Chile. *ArtsEduca*, (33):55–70, 2022.

Anexos

Anexo A

Sistema generador de notas económico

En primer lugar se eligió utilizar archivos .wav (Waveform Audio File), pues tienen una gran calidad de sonido (los archivos .wav no tienen compresión), lo que permite una gran versatilidad en su uso y edición, y con esto, es un formato muy utilizado en la producción musical. El único problema de trabajar con él es su peso, pues si calculamos el tamaño de un segundo de grabación en un formato de 16 bits y una frecuencia de muestreo de 44100Hz con dos canales (para sonido estéreo) nos da

$$\begin{aligned} \text{Peso} &= 44100\text{Hz} * 16\text{bits} * 2 \\ \text{Peso} &= 1,411,200\text{bits} \end{aligned} \tag{A.1}$$

Que en términos de KiloBytes equivale a 172.26KB.

Al tener claro desde las primeras etapas de diseño tener un piano virtual dentro del sistema de evaluaciones, y que un piano tiene 88 teclas en promedio, guardar un archivo de sonido por cada tecla del piano, por al menos 8 segundos (para poder sostener la nota un tiempo suficiente), nos daría un costo de de 1.34MB por nota, y un peso total de aproximadamente 118MB, que si bien por si solo no es un conjunto de archivos muy pesado, al considerar en tener sonidos de otros instrumentos adicionales al piano, eleva su costo computacional. Entonces, se prefirió idear una estrategia mucho más económica, que consiste en guardar solo un archivo de audio sin compresión (.wav) y editar su frecuencia, con esto solo se guardarían 1.34MB por instrumento, permitiendo tener una gran variedad de sonidos, sin que esto constituya un riesgo en el tamaño de la aplicación. Para ello, se utiliza la frecuencia de la nota La4 (equivalente a 440hz) y la siguiente formula:

$$F_{\text{nota}} = 440r^d \tag{A.2}$$

Donde F_{nota} es la frecuencia de la nota que queremos tocar
r es una constante igual a $^{12}\sqrt{2}$
d es la distancia en semitonos de la nota La4

Así por ejemplo, si se quiere calcular la frecuencia de la nota Do5, la cual está a 3 semitonos de La4, ingresando los datos a la formula se obtiene

$$\begin{aligned} F_{nota} &= 440({}^{12}\sqrt{2})^3 \\ F_{nota} &= 523,25 \end{aligned} \tag{A.3}$$

Con ello, fue posible ocupar un archivo .wav con un piano tocando la nota La4, modificar su frecuencia a 523.25 y así obtener un Do5.

Anexo B

Modelo de Base de Datos

B.1. Modelo administrativo y de usuarios

Se colocaron los nombres del modelo en español en esta sección para una mejor comprensión, aunque su implementación está en inglés.

B.1.1. Nivel (Level)

Sin perder de vista el objetivo, que es medir son los conocimientos de los estudiantes, lo primero es organizar a estos estudiantes a partir de su nivel o año académico, por ende el primer modelo será Nivel, el que se detalla a continuación:

- Id: Identificador del modelo.
- Número: Año de escolaridad que el estudiante esta cursando actualmente, Corresponde a una cadena de texto limitada a las palabras: Primero, Segundo, Tercero, ... , Octavo.
- Tipo: Tipo de escolaridad que el alumno esta cursando: Básica o Media.

Es importante tener en cuenta al momento de implementar, que si un alumno está en enseñanza Media, solo puede tener un número entre Primero y Cuarto.

B.1.2. Curso (Grade)

A nivel nacional, un curso es un conjunto de estudiantes que están en el mismo nivel académico que comparten una misma sala de clases, y se diferencian entre cursos por letras, por lo tanto, el modelo fue el siguiente

- Id: Identificador del modelo.

- IdNivel: Identificador del nivel asociado.
- Tipo: Tipo de escolaridad que el alumno esta cursando: Básica o Media.

B.1.3. Usuarios (User)

Para la plataforma educativa se idearon 3 tipos de usuarios diferentes, profesores, estudiantes, y administradores, por ende, se idea una clase usuario como un modelo padre, de las cuales puedan heredar los modelos para cada uno de los usuarios mencionados

Modelo Padre: Usuarios (User)

- Id: Identificador del modelo
- Email: Único, un usuario no podrá tener dos cuentas asociadas al mismo email
- Primer Nombre: Nombre del usuario (obligatorio)
- Segundo Nombre: Segundo nombre del usuario (opcional)
- Primer Apellido: Primer apellido del usuario (obligatorio)
- Segundo Apellido: Segundo apellido del usuario (opcional)
- Tipo: Profesor, Alumno o Administrador

Profesores (Teacher)

- No agrega campos adicionales, sirve solo para poder diferenciarlo del resto de modelos de usuarios.

Estudiantes (Student)

- Id_Curso: Curso al que pertenece el estudiante

Administrador (Admin)

- No agrega campos adicionales, sirve solo para poder diferenciarlo del resto de modelos de usuarios.

B.1.4. Asignatura (Subject)

Ahora teniendo un modelo de Alumnos, Profesores y Cursos, fue necesario estipular un espacio donde estos tres entes se relacionaran, y este lugar corresponde a una Asignatura. Las Asignaturas son todas las materias que se enseñan en un curso y donde son aplicados los instrumentos evaluativos:

- Id: Identificador del modelo.
- Nombre: Nombre de la asignatura
- Descripción: Texto que estipula los detalles de la asignatura
- IdProfesor: Identificador del profesor a cargo de la asignatura (Su email)
- IdEstudiantes: Conjunto de Identificadores de los estudiantes que tomaron la asignatura (Sus emails)

B.2. Sistema de Evaluaciones

El flujo del sistema de preguntas se definió de la siguiente manera: Un profesor crea una pregunta en el contexto de una evaluación, para una Asignatura en específico, y el alumno al momento de abrir una evaluación puede ver dicha pregunta y asignarle una respuesta (Ver sistema de respuestas). Por ende, en primer lugar se creó la modelo evaluación, que contuvo:

B.2.1. Evaluación (Quiz)

- Id: Identificador del modelo.
- Nombre: Nombre de la evaluación
- IdAsignatura: Asignatura donde se toma lugar la evaluación
- FechaDeTermino: Último día donde la evaluación esté disponible para ser contestada
- Activa: Booleano que denota si la evaluación esta disponible para ser contestada
- Terminada: Booleano que denota si la evaluación ya terminó. Es necesario tener este dato adicional pues una evaluación puede estar: no activa y no terminada (sin publicarse), activa y no terminada (publicada), y no activa y terminada (finalizada)
- PuntajeMaximo: La cantidad máxima de puntaje que puede conseguir un estudiante respondiendo correctamente la evaluación.
- NotaMinima: Nota mínima que puede obtener un alumno si no contesta correctamente la evaluación. (por defecto 1.0)

- NotaAprobacion: Nota mínima que requiere un estudiante para aprobar la evaluación (por defecto 4.0)
- NotaMaxima: Nota máxima que puede obtener un alumno, al obtener el Puntaje máximo (por defecto 7.0)
- Escala: Porcentaje que define cuanto puntaje requiere conseguir el estudiante para considerarse aprobado.
- TiempoDisponible: Cantidad de minutos que tiene el alumno para responder.
- NumeroDePreguntas: Cantidad de preguntas que contiene la evaluación

Es importante mencionar que el código 67 del Ministerio de Educación menciona que la nota final de un alumno en alguna asignatura debe estar entre 1.0 y 7.0 con nota de aprobación 4.0, pero eso no necesariamente aplica a una evaluación, por lo tanto, es necesario tener variables para la Nota mínima, máxima y de aprobación.

B.2.2. Preguntas (Question)

Una vez definido el marco de una evaluación, se definió el modelo de una pregunta. Es necesario que una pregunta enmarque el cuestionamiento, el instrumento evaluativo que el alumno tendrá que utilizar para resolver, y el puntaje máximo que puede llegar a tener un alumno en dicha pregunta. Se ideó crear una modelo abstracto de Preguntas y que de ella hereden preguntas, de la siguiente forma:

Modelo Padre: Pregunta (Question)

- Id: Identificador del modelo
- IdEvaluacion: Evaluación a la que pertenece la pregunta
- Numero: Número de pregunta dentro de la evaluación
- Puntaje: Puntaje que puede conseguir un alumno al contestar bien esta pregunta

Pregunta de Verdadero y Falso (TOFQuestion)

- Respuesta Correcta: El docente puede definir cual es la respuesta correcta y así, conseguir que las respuestas de este tipo de preguntas se revisen automáticamente

Pregunta de Alternativas (SelectionQuestion)

- Respuesta Correcta: El docente puede definir cual es la alternativa correcta y así, conseguir que las respuestas de este tipo de preguntas se revisen automáticamente

Pregunta de Desarrollo (WrittingQuestion)

- Rubrica: Descripción y marco evaluativo para definir si la respuesta del alumno es correcta

Pregunta de Piano Virtual (PianoQuestion)

- Visible: Configura si el alumno puede ver en las teclas del piano el nombre de las notas
- Rubrica: Descripción y marco evaluativo para definir si la respuesta del alumno es correcta

Pregunta de Partitura (MusicSheetQuestion)

- Rubrica: Descripción y marco evaluativo para definir si la respuesta del alumno es correcta

B.2.3. Respuesta de una Evaluación (AnsweredQuiz)

Al igual que el modelo de tener una evaluación con un conjunto de preguntas, para responder se hará un modelo de “respuesta de una evaluación”, con un conjunto de respuestas a cada una de las preguntas de la evaluación correspondiente.

- Id: Identificador del modelo.
- IdEvaluación: Identificador de la evaluación que se respondió
- IdEstudiante: Identificador del alumno que respondió
- IdAsignatura: Asignatura donde tomó lugar la evaluación
- PuntajeObtenido: Puntaje que obtuvo el alumno que respondió
- Comentarios: Comentarios dejados por el docente
- TiempoRestante: Tiempo que le sobró al alumno al responder
- Revisada: Booleano que especifica si la respuesta ya la revisó el docente
- CreadaPorElProfesor: En caso que un alumno no responda una evaluación, el profesor le creará una respuesta con nota mínima

Notar que este modelo es creado por alumno, y luego puede ser editado por el profesor al momento de revisar la evaluación.

B.2.4. Respuestas a preguntas (AnsweredQuestion)

Se genera un tipo de respuesta para cada instrumento evaluativo acorde al tipo de pregunta. De igual forma, se genera un modelo padre de respuestas, y luego modelos que heredan, añadiéndole parámetros acordes:

Modelo Padre: Respuesta (AnsweredQuestion)

- Id: Identificador del modelo
- IdRespuestaEvaluacion: “Respuesta de una Evaluación” a la que pertenece la respuesta
- Comentarios: Comentarios a la respuesta del alumno por parte del docente
- Puntaje: Puntaje que consiguió el alumno en esta pregunta

Respuesta de Verdadero y Falso (AnsweredTOFQuestion)

- IdPregunta: Id de una pregunta de Verdader y Falso
- Respuesta: Verdadero, Falso o Nulo (si no respondió)

Respuesta de Alternativas (AnsweredSelectionQuestion)

- IdPregunta: Id de una pregunta de Alternativas
- Respuesta: Alternativa que eligió el estudiante o Nulo (si no respondió)

Respuesta de Desarrollo (AnsweredWrittingQuestion)

- IdPregunta: Id de una pregunta de Desarrollo
- Respuesta: Texto que entregó el alumno como respuesta (puede ser ”)

Respuesta de Piano Virtual (AnsweredPianoQuestion)

- IdPregunta: Id de una pregunta de Piano Virtual
- Respuesta: JSON que almacena el arreglo de notas del estudiante

Respuesta de Partitura (AnsweredMusicSheetQuestion)

- IdPregunta: Id de una pregunta de Partitura
- Respuesta: JSON que almacena las configuraciones iniciales y el conjunto de diccionario por nota insertada

B.2.5. Diagrama de Relación de Entidades

A modo de síntesis, se presenta un diagrama de relaciones entre entidades presentadas en las secciones anteriores

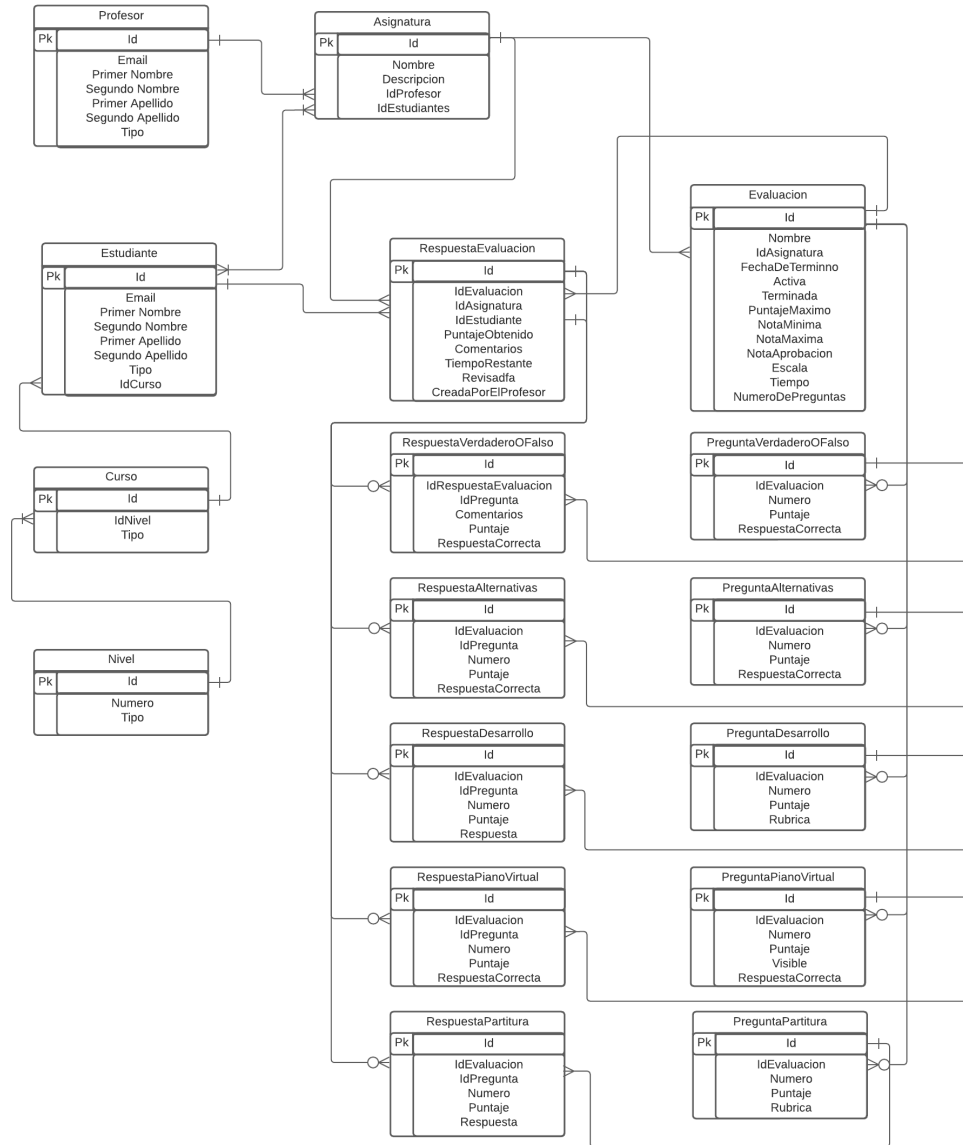


Figura B.1: Relaciones de Entidades del Modelo de Datos

Anexo C

Sistema de partituras

C.1. Configuraciones de Armadura

A continuación se presentan las 15 configuraciones de Armadura existente, en primer lugar muestra el nombre de dicha configuración y a continuación que notas altera.

- C: Ninguna nota es bemol o Sostenido

Configuración de Armaduras por sostenidos

- G: Altera la nota F y le agrega un sostenido
- D: Altera la nota F y C y les agrega un sostenido
- A: Altera las notas F, C y G y les agrega un sostenido
- E: Altera las notas F, C, G Y D y les agrega un sostenido
- B: Altera las notas F, C, G, D y A y les agrega un sostenido
- F#: Altera las notas F, C, G, D, A y E y les agrega un sostenido
- C#: Altera las notas F, C, G, D, A, E y B y les agrega un sostenido

Configuración de Armaduras por bemoles

- F: Altera la nota B y le agrega un bemol
- Bb: Altera la nota B y E, y les agrega un bemol
- Eb: Altera las notas B, E y A, y les agrega un bemol
- Ab: Altera las notas B, E, A Y D, y les agrega un bemol

- Db: Altera las notas B, E, A, D y G, y les agrega un bemol
- Gb: Altera las notas B, E, A, D , G y C, y les agrega un bemol
- Cb: Altera las notas B, E, A, D , G , C y F, y les agrega un bemol

Anexo D

Implementación

D.1. Frecuencia media de instrumentos

En el gráfico de la Figura D.1 se muestran los rangos de frecuencias alcanzados por cada uno de los instrumentos y su promedio. Además, se encuentran trazadas las notas A, con el fin de buscar la octava más cercana al promedio de frecuencia de cada instrumento.

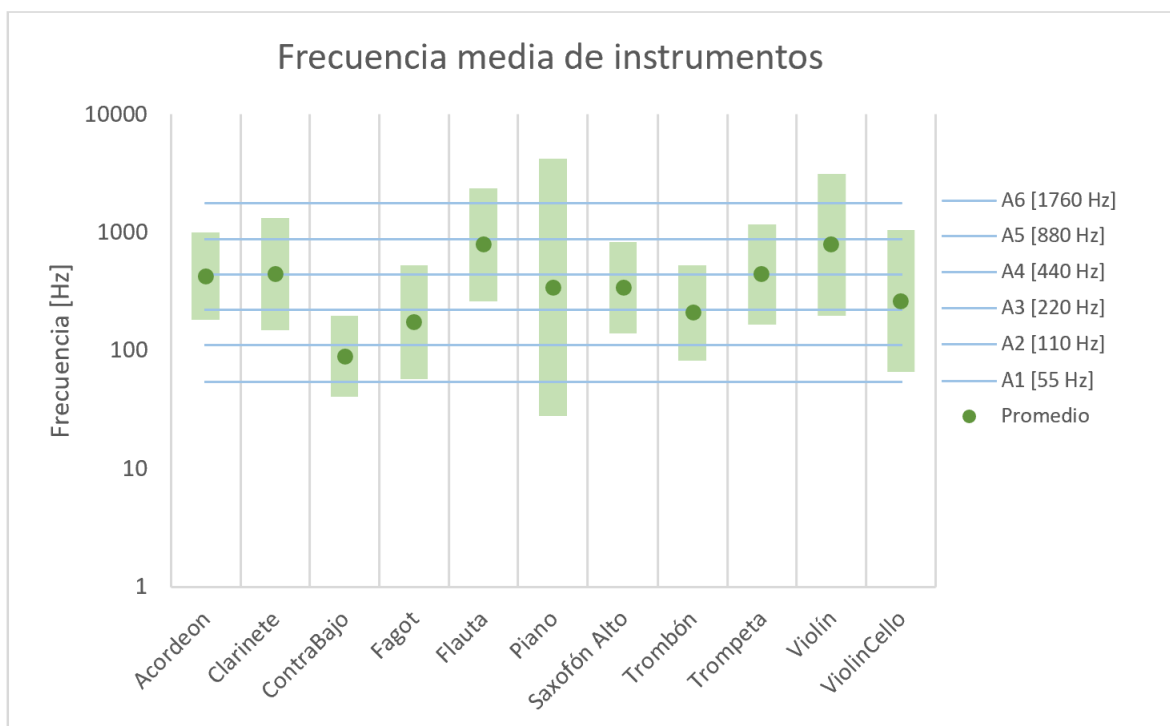


Figura D.1: Frecuencia media de instrumentos

D.2. Diagrama de la configuración del piano virtual

A continuación se presenta un diagrama de flujo para la configuración del piano (Figura D.2), teniendo la nota inicial, octava inicial e instrumento a tocar. Notar que no siempre van intercaladas teclas negras pues la distancia entre teclas blancas no siempre es homogénea, pues en la mayoría la distancia entre blancas es de un tono (es decir, va una negra al medio), mientras que entre E y F, y B y C, la diferencia es de solo medio tono, por ende, no van notas negras entre medio.

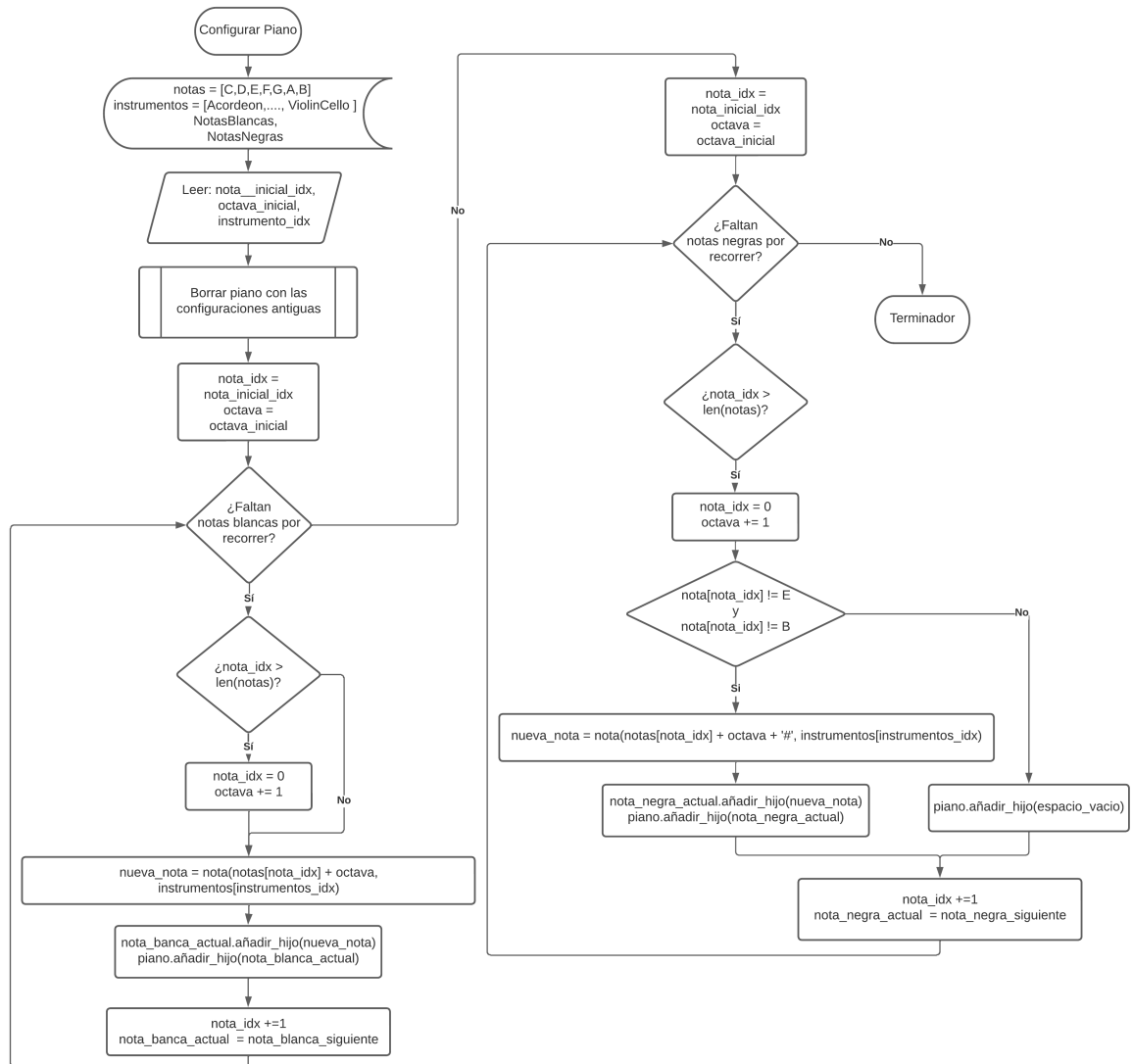


Figura D.2: Configuración del piano virtual

Anexo E

Validaciones

E.1. Validación de evaluaciones para estudiantes

A continuación se transcribe por usuario todas las anotaciones realizadas durante las pruebas de usabilidad moderadas

E.1.1. Usuario 1

El usuario 1 supo cómo iniciar sesión, navegó por las asignaturas hasta encontrar “MusicaTest” e inició “EvaluacionTest”, de manera rápida y sin vacilación. De igual manera, las preguntas de Verdadero y Falso, Alternativas y Desarrollo las pudo responder muy fácilmente, y mencionó que le recordaba a la forma de responder evaluaciones en Google Forms durante la pandemia.

No tuvo problemas para tocar el piano, y tocó las teclas con el teclado, se le preguntó si se le ocurría otra forma de hacer sonar el piano y rápidamente se le ocurrió usar el mouse. Presionó la tecla siguiente sin haber detenido la grabación, lo que causó un bug en el sistema, por lo que se dejó anotado como reproducirlo para su posterior corrección, y se tuvo que reiniciar la prueba, lo que no tomó más de 3 minutos. No tuvo problemas configurar el piano y responder la segunda pregunta del piano, ahora si apretando el botón detener grabación antes de pasar a la siguiente pregunta.

Cuando vio la primera pregunta de Partituras se asustó, pues nunca le había tocado escribir una partitura anteriormente. Pero se le menciona que no era necesario tener experiencia previa. Se le explica un poco en que consiste una partitura, y se le pide que lea las instrucciones de la parte inferior. Menciona que es mucha información de golpe, y que preferiría un dibujo. Al momento de leer la segunda pregunta, no encuentra el botón de configuración en la partitura, se le tuvo que indicar su posición, y pudo configurar lo que se le solicitaba. Luego, menciona que no conoce la tecla SHIFT, por lo que se le indica cual es, y tiene problemas para entender como añadir un sostenido (como lo pedía la pregunta dos), pues era necesario apretar SHIFT + Letra, pero una vez ejemplificándolo lo aprende, e inmediatamente puede

realizar un becuadro y un bemol (Alt + letra y Ctrl + letra, respectivamente).

Como comentarios finales luego de haber contestado todas las preguntas, menciona que el sistema de Partituras lo sintió complicado al leerlo, pero luego de ver ejemplos de uso por parte del validador, no tuvo problemas para utilizarlo y lo terminó dominando en un par de minutos. Propone la opción de tener una clase previa a el primer uso de la partitura en una evaluación, pero que con unos pocos minutos era posible escribir, aunque no sabía que notas estaba escribiendo. También menciona que le gustó poder ir experimentando con las figuras musicales y las alturas, pues aprendió de manera autodidacta que entre más altura tenga una figura musical, más aguda suena.

E.1.2. Usuario 2

Al igual que el primer usuario, pudo iniciar sesión, encontrar la asignatura e iniciar la evaluación, sin problemas. También pudo responder las preguntas de Verdadero y Falso, Alternativas y Desarrollo

Al momento de tener que responder la primera pregunta de piano virtual, no grabó, por lo que se le explicó que el profesor lo que evaluará será solo lo grabado. Reconoce rápidamente el botón para grabar y toca unas cuantas notas con el teclado del computador, se le pregunta de que otra manera habría hecho sonar el piano y probó tocar las teclas con el mouse, aunque le pareció mas cómodo el teclado. En la segunda pregunta, cuando se le pidió configurar, lo hizo sin ningún problema y en ese caso si grabó su respuesta.

Para la partitura, pudo insertar y eliminar notas sin problemas, y si bien no había escrito una partitura antes, notó el mapeo de notas, en donde las figuras de mayor duración están a la derecha, y las de menor duración a la izquierda. Tampoco encontró el botón de configurar, pero pudo configurar bien, finalmente pudo generar notas con bemoles y becuadros, pero tampoco conocía por nombre la tecla SHIFT. Menciona que no le resultó difícil usar la partitura, pero que las teclas SHIFT, Ctrl y Alt deberían estar dibujadas en vez de nombradas.

E.1.3. Usuario 3

Al igual que los usuarios anteriores, logro iniciar sesión e iniciar la evaluación rapidamente. Tampoco tuvo dificultades para las preguntas de verdadero y falso, alternativas y desarrollo.

El piano virtual lo pudo utilizar sin ningún problema, pues este usuario había participado en la primera evaluación.

Para el sistema de partituras fue necesario ejemplificar todos los comandos una vez, luego pudo utilizarlo sin ningún problemas, consideró que sin un ejemplo práctico, es mucha información de golpe. Sugiere que se realice una clase práctica para que el resto de estudiantes pueda aprender sin necesidad de tener que leer instrucciones. Menciona que no sabia combinar teclas, y que no conocía las teclas SHIFT, Ctrl y Alt.