



Universidad de Valladolid

E.T.S Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**Juego para el aprendizaje musical
del ritmo en educación infantil**

Autor:

D. Alfredo Castrillo Rodríguez

Tutor:

D. Carlos Enrique Vivaracho Pascual

Tabla de Contenidos

1. Introducción.....	11
1.1. Objetivos.....	12
1.1.2. Objetivo General.....	12
1.1.3. Objetivos específicos	12
1.2. Organización del documento.....	13
2. Parte teórica	17
2.1 La música en edades tempranas	17
2.2 Genero de los juegos por ordenador educativos	18
2.3 Juegos serios: Tipos	19
2.4 Ideas de otros juegos.....	23
3. Herramientas utilizadas.....	29
3.1 Adobe Flash y Adobe Flash Player	29
3.2 ActionScript 3.0	32
3.3 Audacity	33
3.4 YAKiToMe	33
4. Análisis	37
4.1 Análisis de requisitos	37
4.1.1 Requisitos funcionales.....	37
4.1.2 Requisitos no funcionales.....	42
4.1.3 Requisitos de información	43
4.2 Casos de uso	44
4.3 Diagrama de casos de uso	54
5. Diseño	57
5.1 Arquitectura.....	57
5.2 Diagramas de secuencia	58
5.3 Diagrama de clases.....	68

5.4 Diagrama de paquetes.....	69
6. Implementación y pruebas	73
6.1. Implementación.....	73
6.2. Pruebas	81
7. Conclusiones y posibles ampliaciones	91
8. Bibliografía y Glosario	95
8.1 Bibliografía	95
8.2 Glosario de términos y siglas	96
Anexo A.....	97
Anexo A.1: Manual de usuario	97
Anexo A.2: Manual de instalación	108
Anexo A.3: Contenido del CD	109

Lista de figuras y tablas

Figura 1: Pantalla inicio Adobe Flash CS3	11
Figura 2: Art Academy para NDS	21
Figura 3: E.I.S. Simulator	23
Figura 4: Multipistas infantil	24
Figura 5: Juego Bombillas	25
Figura 6: Juego musical de jCyL	26
Figura 7: Evolución histórica de Flash.....	31
Figura 8: Diagrama de casos de uso	54
Figura 9: DS Introducción	58
Figura 10: DS seleccionar nivel	59
Figura 11: DS seleccionar instrumentos	60
Figura 12: DS presentación academia	61
Figura 13: DS academia	62
Figura 14: DS tocar.....	63
Figura 15: DS escuchar.....	64
Figura 16: DS seleccionar canción	65
Figura 17: DS juego	66
Figura 18: DS modificar velocidad	67
Figura 19: Diagrama de clases	68
Figura 20: Diagrama de paquetes.....	69
Figura 21: Subclases de DisplayObject.....	75
Figura 22: Entrada al juego	97
Figura 23: Menú principal.....	98
Figura 24: Presentación academia 1.....	99
Figura 25: Presentación academia 2.....	100
Figura 26: Menú de niveles	101
Figura 27: Menú de selección de instrumentos	102
Figura 28: Escena de interpretación.....	103
Figura 29: Trofeo	104
Figura 30: Menú de canciones.....	105
Figura 31: Interpretación de canción	106
Figura 32: Escena puntuaciones	107
Figura 33: Trofeo canción	108

Tabla FRQ 1: Muñeco	37
Tabla FRQ 2: Introducción	37
Tabla FRQ 3: Selección de instrumentos	37
Tabla FRQ 4: Realizar demostración	38
Tabla FRQ 5: Puntuación	38
Tabla FRQ 6: Academia.....	38
Tabla FRQ 7: Presentación Instrumentos	39
Tabla FRQ 8: Niveles	39
Tabla FRQ 9: Velocidad de interpretación	40
Tabla FRQ 10: Juego.....	40
Tabla FRQ 11: Elegir canción	40
Tabla FRQ 12: Niveles	40
Tabla FRQ 13: Relacionar líneas rítmicas e instrumentos	41
Tabla FRQ 14: Longitud de las canciones	41
Tabla FRQ 15: Tiempo de compas	41
Tabla FRQ 16: Claqueta de interpretación	41
Tabla FRQ 17: Canciones	41
Tabla NRF 1: Flash.....	42
Tabla NFR 2: Navegador web.....	42
Tabla NFR 3: Pantalla táctil	42
Tabla NFR 4: Orientado a niños	42
Tabla NFR 5: Voz sintetizada	42
Tabla NFR 6: Muñeco robot.....	43
Tabla IRQ 1: Conjunto de instrumentos	43
Tabla IRQ 2: Conjunto de figuras musicales y silencios	44
Tabla IRQ 3: Canciones	44
Tabla UC 1: Introducción	45
Tabla UC 2: Seleccionar nivel.....	45
Tabla UC 3: Seleccionar los instrumentos	46

Tabla UC 4: Seleccionar canción	47
Tabla UC 5: Presentación academia	49
Tabla UC 6: Academia	49
Tabla UC 7: Escuchar	50
Tabla UC 8: Tocar	51
Tabla UC 9: Modificar velocidad	52
Tabla UC 10: Juego	53
Tabla CP 1: Introducción.....	81
Tabla CP 2: Introducción.....	81
Tabla CP 3: Selección nivel	82
Tabla CP 4: Selección nivel	82
Tabla CP 5: Selección Instrumentos	82
Tabla CP 6: Selección Instrumentos	83
Tabla CP 7: Selección Instrumentos	83
Tabla CP 8: Selección de canción.....	83
Tabla CP 9: Selección de canción.....	84
Tabla CP 10: Presentación academia.....	84
Tabla CP 11: Presentación academia.....	84
Tabla CP 12: Presentación academia.....	85
Tabla CP 13: Academia	85
Tabla CP 14: Escuchar	85
Tabla CP 15: Escuchar	86
Tabla CP 16: Tocar	86
Tabla CP 17: Tocar	87
Tabla CP 18: Tocar	87
Tabla CP 19: Modificar velocidad	88
Tabla CP 20: Modificar velocidad	88
Tabla CP 21: Juego	88

Capítulo 1

Introducción

1. Introducción

Este proyecto ha sido motivado por la metodología de Charo Castañón, profesora en educación, basada en la introducción del aprendizaje musical en edades tempranas y del uso de herramientas informáticas multimedia que puedan apoyar este aprendizaje, siendo además muy atractivas para los niños.

El fin de este juego educativo musical y para lo que está destinado es ser una herramienta de ayuda pedagógica recomendada a niños pequeños para que aprendan a leer rítmicamente a nivel básico y al mismo tiempo desarrollen la psicomotricidad, a la vez que les permiten conocer algunos instrumentos con sus sonidos, ayudar al desarrollo de la capacidad lectora y como introducción al uso de aplicaciones informáticas.

Para la realización de la aplicación de este proyecto se utilizó la tecnología Flash. Desde la aparición de ActionScript en la tecnología Flash en el año 2000, esta se ha utilizado cada vez más para crear juegos de forma sencilla pudiendo utilizarlos en cualquier ordenador vía web.

En todos estos años, Flash y ActionScript han ido evolucionando a la vez que la complejidad de los juegos creados con este. La posibilidad de utilizar el paradigma de la orientación a objetos y la creación de archivos de código ActionScript externos fue clave para poder avanzar en la creación de juegos para ordenador.



Figura 1: Pantalla inicio Adobe Flash CS3

A día de hoy Flash ha servido para dar vida a infinidad de juegos, desde los más sencillos hasta los más complejos, pudiendo compartirlos inmediatamente a través de la World Wide Web.

Muchos son los videojuegos y compañías que han surgido y se han hecho un hueco en el mercado de los videojuegos, algunos continuando su programación en Flash con juegos en Facebook y otras páginas web, otros adaptándose al mercado de los videojuegos en ordenador y consolas.

1.1. Objetivos

1.1.2. Objetivo General

Crear un juego educativo para la ayuda al aprendizaje musical del ritmo en edades tempranas (3 a 5 años). Esta herramienta servirá de apoyo a la nueva metodología para la introducción del aprendizaje del lenguaje musical en educación infantil desarrollada por la profesora Dña. María del Rosario Castañón.

1.1.3. Objetivos específicos

- Aplicar los conocimientos adquiridos durante la carrera a un proyecto informático basado en los requisitos expuestos por un cliente.
- La aplicación deberá tener un interfaz atractivo y adaptado a niños de corta edad.
- La aplicación deberá ser muy sencilla de utilizar.
- El programa final deberá poder ser ejecutado en pizarras electrónicas
- La aplicación deberá estar documentada y realizada para permitir posteriores modificaciones de manera sencilla.
- Estudiar la tecnología de desarrollo de aplicaciones en Flash y adaptarlas a la realización del juego.

1.2. Organización del documento

Esta memoria se divide en 8 capítulos:

- El capítulo 1. Introducción sintetiza los objetivos de este proyecto y de que manera esta organizado el documento.
- El capítulo 2. Parte teórica que contiene documentación relacionada con los beneficios de la música en edades tempranas, juegos serios y un pequeño análisis de otros juegos.
- El capítulo 3. Herramientas utilizadas. Se da información sobre los programas que se han utilizado para el desarrollo Adobe Flash, ActionScript, Audacity y YAKiToMe.
- El capítulo 4. Análisis. Comprende lo relacionado con el análisis del desarrollo software. Análisis de requisitos y casos de uso.
- El capítulo 5. Diseño. Comprende lo relacionado con el diseño del desarrollo software. Arquitectura, diagramas de secuencia, clases y paquetes.
- El capítulo 6. Implementación y pruebas. Está compuesto por una descripción de la implementación y el documento de casos de prueba utilizado.
- El capítulo 7. Conclusiones y posibles ampliaciones. Contiene las conclusiones obtenidas de la realización del presente proyecto y posibles ampliaciones futuras del mismo que se contemplan.
- El capítulo 8. Bibliografía y Glosario contiene las referencias bibliográficas utilizadas en este trabajo y el glosario con los términos empleados en el mismo.

Capítulo 2

Parte teórica

2. Parte teórica

2.1 La música en edades tempranas

La música está siendo introducida en la educación de los niños en edades prescolares debido a la importancia que representa en su desarrollo intelectual, auditivo, sensorial, del habla y motriz.

La música es un elemento fundamental en esta primera etapa del sistema educativo. El niño empieza a expresarse de otra manera y es capaz de integrarse activamente en la sociedad, porque la música le ayuda a lograr autonomía en sus actividades habituales, asumir el cuidado de sí mismo y del entorno, y ampliar su mundo de relaciones.

La música tiene el don de acercar a las personas. El niño que vive en contacto con la música aprende a convivir de mejor manera con otros niños, estableciendo una comunicación más armoniosa. A esta edad la música les encanta. Les da seguridad emocional, confianza, porque se sienten comprendidos al compartir canciones, e inmersos en un clima de ayuda, colaboración y respeto mutuo.

La etapa de la alfabetización del niño se ve más estimulada con la música. A través de las canciones infantiles, en las que las sílabas son rimadas y repetitivas, y acompañadas de gestos que se hacen al cantar, el niño mejora su forma de hablar y de entender el significado de cada palabra. Y así, se alfabetizará de una forma más rápida.

La música también es beneficiosa para el niño cuanto al poder de concentración, además de mejorar su capacidad de aprendizaje en matemáticas. La música es pura matemática. Además, facilita a los niños el aprendizaje de otros idiomas, potenciando su memoria.

Con la música, la expresión corporal del niño se ve más estimulada. Utilizan nuevos recursos al adaptar su movimiento corporal a los ritmos de diferentes obras, contribuyendo de esta forma a la potenciación del control rítmico de su cuerpo. A través de la música, el niño puede mejorar su coordinación y combinar una serie de conductas.

2.2 Genero de los juegos por ordenador educativos

La facilidad con la que los niños entran en la era digital tiene mucho que ver con la maleabilidad de sus jóvenes mentes. Pero, naturalmente, necesitan tener acceso a los ordenadores para poder comprenderlos, y conseguir así los conocimientos básicos para desenvolverse con las TIC. Esos conocimientos son conseguidos a través de la educación informal que proporciona el uso habitual de ordenadores.

En concreto, esta alfabetización instrumental se produce a partir del uso de los juegos, lo que, además, despierta el interés por la tecnología a una temprana edad. Así, al igual que los juegos motores permitían a los bebés familiarizarse con su entorno, los juegos se convierten en una excusa perfecta para que pierdan el miedo a los ordenadores y se acostumbren a ellos.

Esto es también cierto para los adultos. El sencillo juego "Buscaminas", tradicional en los sistemas Windows, no persiguen más que la familiaridad con el uso del ratón en los primeros momentos. Gracias a la práctica y la repetición constante, el cambio desde los interfaces basados en consola hacia los interfaces gráficos pudo realizarse de una manera más distendida.

En España es ADESE (Asociación Española de Distribuidores y Editores de Software de Entretenimiento) la más interesada en convencer a la sociedad de las bondades de este tipo de entretenimiento. Así, a parte de la habitual mejora de coordinación ojo-mano proporcionada por los videojuegos, también tienen otras ventajas educativas:

- El 77% de los jugadores cree que su uso potencia la competitividad. Esto hace además que aumente la capacidad de superación.
- El 76% opina que los videojuegos proporcionan agilidad mental.
- El 71% prefiere jugar en grupo. Esto permite interacciones sociales valiosas, que hace que los niños formen parte de un grupo, encuentren su lugar en él y pongan en práctica habilidades que les serán importantes en el futuro como el trabajo en grupo, la colaboración y la cooperación.
- El 49% considera que mejora la creatividad.

Los expertos han determinado que cuando los niños se sientan a estudiar se desmotivan fácilmente, ya que no saben cuál es el objetivo del estudio. Sin embargo, esto no ocurre a la hora de utilizar el videojuego como fórmula de aprendizaje, ya que en el

juego siempre hay un objetivo que conseguir, lo que le aporta un gran nivel de motivación. El niño se esfuerza por lograr la meta en un entorno donde el error lleva a un nuevo intento, y cuando finalmente consigue su objetivo, la recompensa es una gran satisfacción consigo mismo y un aumento de la autoestima.

Pero no sólo el videojuego está en el hogar. También los centros educativos han comenzado a utilizarlo como método de aprendizaje, ya que el lenguaje audiovisual lo convierte en una actividad formativa muy atractiva, al tiempo que potencia la motivación e integración del alumnado, según han comprobado diferentes estudios elaborados por universidades como la de Alcalá y la UNED.

Dada la capacidad educativa informal de los videojuegos y su increíble motivación, resulta tentador tratar de utilizarlos para "enseñanza intencionada", es decir, para cuestiones más allá de la competitividad.

Surge así el ámbito de los conocidos como juegos serios.

2.3 Juegos serios: Tipos

Todos sabemos que un juego es ante todo entretenimiento, un desafío para lograr un objetivo, aunque un "serious game" o "juego serio" aporta valor más allá del entretenimiento.

Este tipo de juegos normalmente se asocian con la educación y el aprendizaje de nuevos conceptos y habilidades, pero también se utilizan con fines comerciales, de concienciación o de denuncia social o política.

Los serious games pueden ayudar a la comprensión de conceptos o materias y como medio de comunicación o publicitario, pero también pueden funcionar como entrenamiento y simulación de diversas actividades, como la de aprender la gestión de empresas a gran escala o a pilotar un vehículo.

El valor de un juego como serious game depende de la relación directa que tenga con el mundo real y en no basarse en una pura fantasía. En otras palabras, un juego serio debe tener un vínculo evidente entre el mundo real y el virtual, y un objetivo más allá del puro entretenimiento.

Los juegos serios se pueden clasificar como:

- Advergaming. Los advergames son juegos interactivos que permiten una exposición continuada del usuario ante la marca o producto publicitado. Como ejemplo, el juego de motos online que fue desarrollado para Repsol YPF para promocionar su portal web.
- Juegos educativos. Ya se ha comentado un poco anteriormente, es donde nos centraremos en este proyecto. Son juegos que tratan de formar con aprendizaje asistido. Los juegos educativos más comunes son los orientados a niños, normalmente clasificados por materias y edades.

Los juegos educativos suelen ser juegos muy concretos destinados únicamente para educar de una manera interactiva, normalmente son mini juegos muy simples como podrían ser de ordenar elementos, hacer un puzle, unir puntos, responder preguntas, hacer parejas de elementos, etc.

Una evolución de este tipo de juegos son los surgieron a partir de año 2005 con Brain Training que consisten en un conjunto de mini juegos que ayudan a ejercitar la mente con diferentes dinámicas. A su vez estos juegos han ido evolucionando y pueden servir tanto para ejercitar la mente, aprender idiomas, o incluso aprender a dibujar.

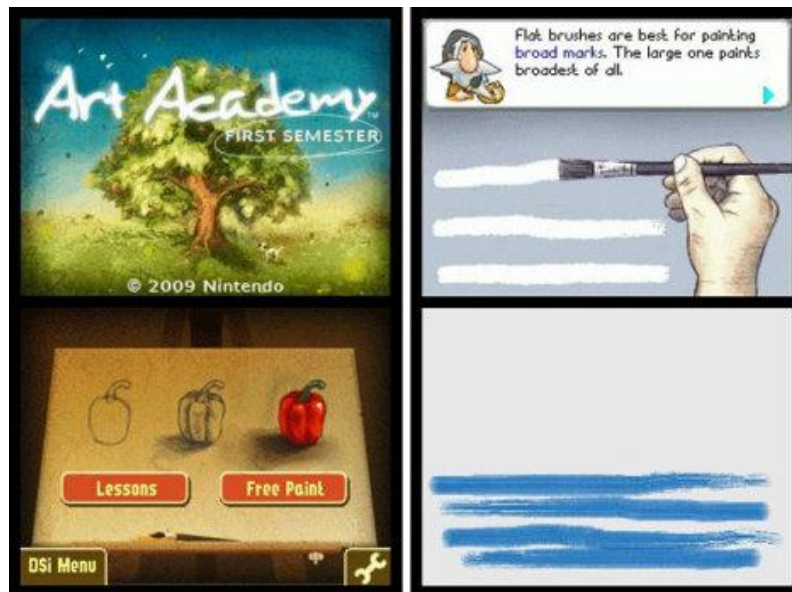


Figura 2: Art Academy para NDS

El principal problema en los juegos educativos está en la necesidad de un equilibrio entre el contenido y el componente lúdico. Es necesario que ambas partes aparezcan completamente integradas en el diseño del juego. Por ejemplo, si se quiere desarrollar un juego educativo para practicar las operaciones matemáticas básicas, resulta tentador pedir al jugador la respuesta de una suma antes de, por ejemplo, permitirle disparar. Sin embargo, en ese caso la parte educativa está totalmente separada del propio juego: éste tendría sentido sin esas preguntas. La sensación es de "fuego y hielo": la parte divertida mantiene la atención, pero la parte educativa hace que ésta desaparezca.

- Juegos de entrenamiento. En los juegos de entrenamiento las personas son capaces de moverse e interactuar libremente, experimentando la vida como simulaciones, como si estuvieran realmente allí. La experiencia interactiva y realista ayuda a las personas a aprender más rápidamente, recordar mejor los procedimientos y desarrollar las competencias esenciales.

Un ejemplo de este tipo de juego son los juegos de entrenamiento militar, juegos que tratan varios aspectos del entrenamiento militar (desde combates de guerrilla a pilotaje de aviones). Son juegos por encargo y “mods” pagados directamente por las agencias militares.

- Salud. Los juegos sobre la salud (también llamados “juegos de bienestar” o “health games”), tratan temas relacionados con la prevención de enfermedades, la vida sana, el ejercicio, etc. Es el tipo de juego más investigado por claros beneficios sociales que aportan.
- Políticos y Sociales. Estos son juegos diseñados para educar al público sobre sus derechos y obligaciones como ciudadanos, para fomentar los comportamientos cívicos y las normas de convivencia. Las ONG los utilizan para denunciar injusticias o para promover la conciencia social.

Algunos grupos políticos también los utilizan para comunicar algunos aspectos de sus campañas o de sus gobiernos.

- Religiosos. Los más abundantes son los de temática cristiana.
- Negocios. Juegos en los que se simulan estrategias relacionadas con situaciones económicas y de control de empresas. Sirven para reflejar un posible escenario y ver los resultados que nos daría una simulación. Con los resultados obtenidos podremos cambiar nuestras estrategias y decisiones.

En la escuela hay instalados varios de simulación de situaciones en empresas. Por poner un ejemplo, E.I.S. simulator, en el que la idea es la de introducir una situación innovadora en la filial “Teleswitches Ltd” de la Corporación EuroComm (ver fig.3).



Figura 3: E.I.S. Simulator

2.4 Ideas de otros juegos

Antes de empezar con el desarrollo del juego hubo una primera etapa de investigación en la que se observaron otros juegos serios de similares características para obtener ideas que pudiesen servir en nuestro juego. A continuación se muestra un resumen de lo obtenido.

Análisis de otros juegos educativos:

- Guías: En los juegos educativos que se han revisado, en algunos aparece una especie de muñeco que guía al niño por la aplicación. Por ejemplo, según pasas por encima de un botón el muñeco se pone en movimiento y nos dice para que sirve. Su objetivo suele ser mas que nada guiarnos a través de los menús. Este muñeco siempre es el mismo a lo largo del juego. Aparte de esto, el muñeco sirve

para cuando se están realizando las actividades nos diga cuando lo hemos hecho bien, si lo tenemos que intentar de nuevo etc...

En algunos juegos educativos de tipo musical puede existir un apartado tipo “enciclopedia” que nos habla un poco de los instrumentos que aparezcan en el juego y nos muestra como suenan.

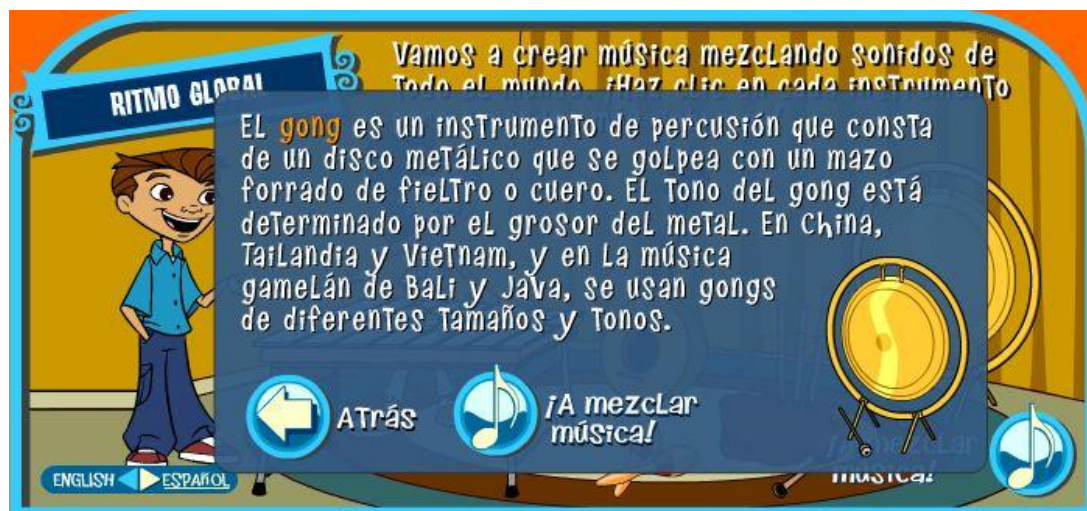


Figura 4: Multipistas infantil

- Puntuaciones/recompensas: En este tipo de juegos educativos no suele haber ningún sistema de recompensas o puntuaciones mas haya del número de aciertos/fallos o el número de intentos cuando se están realizando las actividades. Quizás sea por que tener una voz que te diga que lo has hecho bien sea suficiente. Otros juegos te van dando trofeos según vas completando cierto número de actividades, por ejemplo, cada vez que lo hagas bien 10 veces te aparecerá un trofeo.

En otro tipo de juegos no educativos, por ejemplo los que nos encontramos en <http://www.kongregate.com>, existen los llamados achievements (logros) que son como pequeñas misiones como, por ejemplo, alcanzar determinada cantidad de puntos.

Estos están ordenados por nivel de dificultad. La idea de este sistema es alargar la vida del juego repitiendo las actividades e intentando completar todos los logros que se puedan.

- Dificultad: Hay juegos en los que al principio te piden que elijas el nivel de dificultad dependiendo de cierto criterio. Por ejemplo, el juego que aparece en la imagen es un juego de encender un número determinado de bombillas. Dependiendo del nivel que se elija este número estará en un intervalo o en otro.



Figura 5: Juego Bombillas

Otras veces la dificultad aparece implícita en el propio juego. Es el caso de un juego musical de memoria donde tienes que recordar cada vez mas notas según vas avanzando.

- Estructura/imágenes: Todas las imágenes que aparecen en los juegos educativos para niños son en su mayoría dibujos. No se suelen utilizar instrumentos reales sino dibujos. La mayoría de los juegos educativos no tienen ninguna introducción

y se pasa directamente al juego. En algunos simplemente tenemos un fondo con un botón que pone “jugar” que nos pasa al juego junto con un botón de “instrucciones”.

En el caso de que el juego tenga varios apartados nos aparece un menú al comienzo con las distintas actividades que se pueden realizar. Es el caso de la imagen que aparece a continuación donde el juego nos ofrece: un karaoke, tocar el piano o ver una pequeña “enciclopedia” de instrumentos de orquesta.



Figura 6: Juego musical de jCyl

Capítulo 3

Herramientas utilizadas

3. Herramientas utilizadas

3.1 Adobe Flash y Adobe Flash Player

Adobe Flash Professional es una aplicación de creación y manipulación de gráficos vectoriales con posibilidad de manejo de código mediante el lenguaje ActionScript en forma de estudio de animación que trabaja sobre "fotogramas" y está destinado a la producción y entrega de contenido interactivo para las diferentes audiencias alrededor del mundo sin importar la plataforma. Es actualmente desarrollado y comercializado por Adobe Systems Incorporated y forma parte de la familia Adobe Creative Suite.

Adobe Flash utiliza gráficos vectoriales y gráficos rasterizados, sonido, código de programa, flujo de vídeo y audio bidireccional (el flujo de subida sólo está disponible si se usa conjuntamente con Macromedia Flash Communication Server). En sentido estricto, Flash es el entorno de desarrollo y Flash Player es el reproductor utilizado para visualizar los archivos generados con Flash. En otras palabras, Adobe Flash crea y edita las animaciones o archivos multimedia y Adobe Flash Player las reproduce.

Los archivos de Adobe Flash, que tienen generalmente la extensión de archivo SWF, pueden aparecer en una página web para ser vistos en un navegador web, o pueden ser reproducidos independientemente por un reproductor Flash. Los archivos de Flash aparecen muy a menudo como animaciones en sitios web multimedia, y más recientemente en Aplicaciones de Internet Ricas.

En versiones anteriores, Macromedia amplió a Flash más allá de las animaciones simples, convirtiéndolo en una herramienta de desarrollo completa, para crear principalmente elementos multimedia e interactivos para Internet.

Originalmente Flash no fue un desarrollo por el propio de Adobe, sino de una pequeña empresa de desarrollo de nombre FutureWave Software y su nombre original fue FutureSplash Animator. En diciembre de 1996 Macromedia adquiere FutureWave Software, y con ello su programa de animación vectorial que pasa a ser conocido como Flash 1.0.

Desde 1996 hasta 2005 Macromedia lanzó 8 versiones del producto Flash, añadiendo nuevas funcionalidades en cada versión. En Flash 2 se incluyó la librería de objetos, en Flash 3 el elemento MovieClip que es la base de Flash e integración con JavaScript, en

Flash 4 la utilización de código escrito y compatibilidad con MP3s, en Flash 5 la primera versión de ActionScript y soporte para XML, en Flash MX codecs para videos y la API de dibujo, en MX 2004 ActionScript 2.0 que permitía la programación orientada a objetos, el formato FLV que permitió la creación de webs como Youtube y en Flash 8 más control para las animaciones y mayor facilidad de uso.

En 2005 Adobe compra Macromedia y junto con ella sus productos, entre ellos Flash, que pasa a llamarse Adobe Flash.

Desde el año 2007 hasta el 2010 Adobe ha lanzado 3 versiones de Flash bajo el sello Creative Suite, una colección de programas de edición de Adobe y siguieron añadiendo nuevas funcionalidades en cada versión. En Flash CS3 ActionScript 3.0 e integración con otros productos de Adobe como Adobe Photoshop. En Flash CS4 la utilización de cinemática inversa (huesos), manipulación de objetos 3D y la no retro compatibilidad con muchas de las funciones y clases anteriores a ActionScript 3.0. En Flash CS5 un nuevo gestor de textos conocido como TLF.

Para el desarrollo de este juego se ha utilizado la versión Adobe Flash Professional CS3 pues es la que empieza a trabajar con ActionScript 3.0.

En la figura 7 se puede ver una representación gráfica de la evolución expuesta.

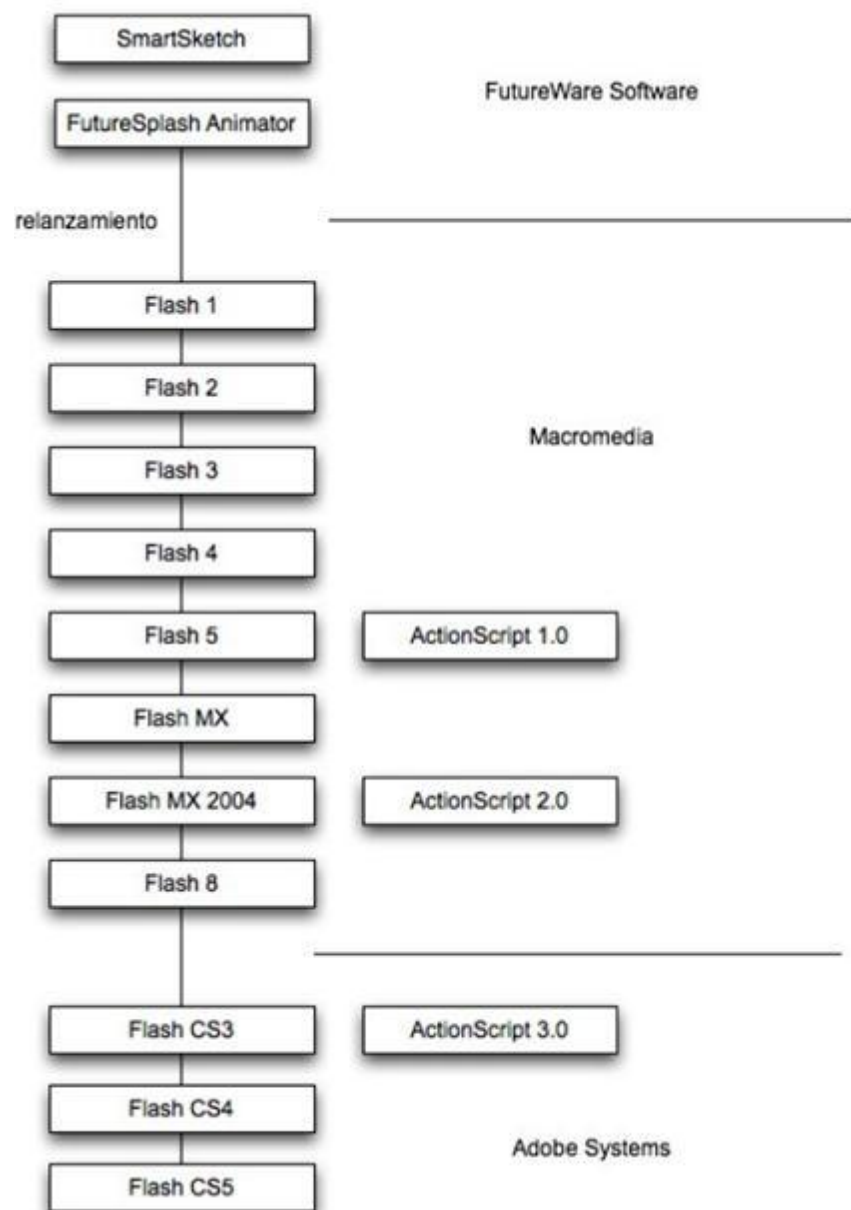


Figura 7: Evolución histórica de Flash

3.2 ActionScript 3.0

Adobe ActionScript es el lenguaje de programación de la Plataforma Adobe Flash. Originalmente desarrollado como una forma para que los desarrolladores programen en forma más interactiva. La programación con ActionScript permite mucha más eficiencia en las aplicaciones de la plataforma Flash para construir animaciones de todo tipo, desde simples a complejas, ricas en datos e interfaces interactivas.

ActionScript 3.0 ofrece un modelo de programación robusto que resultará familiar a los desarrolladores con conocimientos básicos sobre programación orientada a objetos, en especial java y C++.

Como ya se dijo en la creación del juego se utiliza Adobe Flash CS3 como plataforma básica, utilizando el propio lenguaje de programación ActionScript 3.0 para gestionar el funcionamiento del juego. En Flash se crean los elementos visuales del juego, los cuales van encapsulados en MovieClips que posteriormente ActionScript puede utilizar para modificar su comportamiento.

Además de gestionar el comportamiento de los elementos visuales, ActionScript controla toda la ejecución del juego, con una programación basada en orientación a objetos.

Desde ActionScript pueden utilizarse datos dinámicamente como sonidos o archivos de texto, sin necesidad de añadirlos al archivo de flash. La ejecución del bucle principal del juego y la entrada de datos de parte del usuario se hacen a través de eventos que se pueden escuchar desde ActionScript.

ActionScript 3.0 aumenta las posibilidades de creación de scripts de las versiones anteriores de ActionScript. Se ha diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player 9, ActionScript 3.0 permite introducir unas mejoras de rendimiento que sólo están disponibles con AVM2, la nueva máquina virtual. El código ActionScript 3.0 puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado.

La versión anterior de la máquina virtual ActionScript (AVM1) ejecuta código ActionScript 1.0 y ActionScript 2.0. Flash Player 9 admite AVM1 por compatibilidad con contenido existente y heredado de versiones anteriores.

3.3 Audacity

Audacity es una aplicación informática multiplataforma libre, que se puede usar para grabación y edición de audio, fácil de usar, distribuido bajo la licencia GPL.

En nuestro caso, Audacity se utilizó para toda la parte de edición de las canciones además de para retocar el sonido de los instrumentos que fueron obtenidos de un banco de sonidos.

Permite trabajar, entre otros formatos, con el formato mp3 que es el que se utiliza en esta aplicación aunque audacity también permite la conversión entre distintos formatos de audio.

3.4 YAKiToMe

De entre todos los sintetizadores que se barajaron utilizar se optó por este por la calidad de la voz (no es tan robotizada) además de tener voz en español. YAKiToMe es un sintetizador de voz, conocidos también por las siglas en inglés TTS (Text to speech).

Es gratuito y se utiliza on-line, no hace falta descargarse o instalarse. Hace falta registrar una cuenta para usarlo, guarda todas las síntesis que hayas realizado lo cual es muy útil.

Tiene algunas desventajas comparado con otros programas de pago. Algunas veces las síntesis que realiza tienen algunos problemas con algunos textos, además, te obliga a esperar una cola con cada síntesis que hagas aunque sea muy pequeña.

La idea de los TTS es la de producir una voz sintética que resulte natural y sea inteligible de manera automática sin tener que introducir ningún reajuste manual en ninguna parte del proceso. Se debe de poder utilizar para cualquier texto arbitrario que se introduzca.

La conversión de texto a voz tiene dos fases:

- En la primera fase se realiza una representación lingüística simbólica, para ello se siguen tres procesos consecutivos:
 1. Normalización del texto. Se convierte la totalidad del texto a una forma textual convencional. Esto afecta principalmente a las cifras,

- abreviaturas, etc. A la normalización del texto también se la denomina pre procesado o tokenización.
2. Conversión fonética. Una vez normalizado el texto se asignan transcripciones fonéticas a cada palabra. El proceso de convertir las transcripciones fonéticas en palabras se denomina “conversión texto-fonema” (TTP en sus siglas en inglés de text-to-phoneme) o “conversión grafema-fonema” (GTP en sus siglas en inglés de grapheme-to-phoneme).
 3. División prosódica. Se divide el texto en unidades prosódicas, tales como unidades sintagmáticas, proposiciones y frases.
- En la segunda fase, la que forma el sintetizador propiamente dicho, toma como entrada la representación lingüística simbólica y la transforma en voz sintética.

Capítulo 4

Análisis

4. Análisis

En este apartado se muestra el resultado de la primera etapa del proceso de desarrollo.

4.1 Análisis de requisitos

4.1.1 Requisitos funcionales

Se dividen en 4 partes: requisitos generales de la aplicación, requisitos sobre la academia, requisitos sobre el juego y requisitos sobre los ostenatos/canciones.

Requisitos generales:

FRQ-1:	Muñeco
Descripción:	El sistema deberá hacer uso de un muñeco que guíe y ayude al usuario a lo largo de la aplicación.

Tabla FRQ 1: Muñeco

FRQ-2:	Introducción
Descripción:	El sistema deberá tener una escena de “introducción” donde se presente el muñeco y la aplicación. Tiene que tener la opción de saltarse

Tabla FRQ 2: Introducción

FRQ-3:	Selección de instrumentos
Descripción:	El sistema deberá mostrar los instrumentos que dispone para que el usuario seleccione los que crea correspondientes para la prueba a realizar

Tabla FRQ 3: Selección de instrumentos

FRQ-4:	Realizar demostración
Descripción:	El sistema deberá proporcionar al usuario la capacidad de escuchar la canción que se encuentre en pantalla a modo de demostración, tanto en la academia como en el juego.

Tabla FRQ 4: Realizar demostración

FRQ-5:	Puntuación
Descripción:	El sistema deberá proporcionar un mecanismo para puntuar la interpretación del usuario y que sirva de retroalimentación. El mecanismo tendrá en cuenta la distancia entre la pulsación y la figura musical correspondiente. La puntuación se mostrará en forma de estrellas, siendo un máximo de 5 estrellas la puntuación máxima. Se utilizarán trofeos.

Tabla FRQ 5: Puntuación

Requisitos Academia:

FRQ-6:	Academia
Descripción:	El sistema deberá proporcionar un apartado denominado “Academia” en el que el usuario pueda aprender a utilizar la aplicación. La academia se dispondrá de la siguiente forma, tendrá dos partes: presentación de instrumentos e interpretación.

Tabla FRQ 6: Academia

FRQ-7:	Presentación Instrumentos
Descripción:	El sistema presentará, dentro de la academia, los instrumentos que se pueden utilizar en la aplicación. Se hará uso del muñeco para proponer al usuario un pequeño juego de relacionar el nombre del instrumento al instrumento. Este juego se dividirá en dos partes: primero se presentarán y se jugará con los instrumentos básicos y luego se hará lo mismo con los de la batería.

Tabla FRQ 7: Presentación Instrumentos

FRQ-8:	Niveles
Descripción:	<p>El sistema deberá proporcionar, dentro de la academia, un sistema de niveles donde a mayor nivel mayor dificultad tenga la interpretación del ostinato rítmico a realizar. Se contemplan un total de 4 niveles cuyo contenido será el siguiente:</p> <p>Nivel 1. Se permitirá el uso de solo 1 instrumento. Contendrá 2 ostinatos utilizando solo blancas y negras. El tiempo de compas será de 2/4.</p> <p>Nivel 2. Se permitirá el uso de solo 1 instrumento. Contendrá 1 ostinato donde se añadirá el uso de corcheas. El tiempo de compas será 2/4.</p> <p>Nivel 3. Se añadirá el uso de 2 instrumentos a la vez. El tiempo de compas será 2/4.</p> <p>Nivel 4. Se añadirá el uso de 2 instrumentos a la vez. El tiempo de compas será 3/4.</p>

Tabla FRQ 8: Niveles

FRQ-9:	Velocidad de interpretación
Descripción:	El sistema deberá proporcionar al usuario un mecanismo para controlar la velocidad del cursor de interpretación.

Tabla FRQ 9: Velocidad de interpretación

Requisitos juego:

FRQ-10:	Juego
Descripción:	El sistema deberá proporcionar un apartado denominado “Juego” en el que el usuario pueda realizar un acompañamiento rítmico a una canción.

Tabla FRQ 10: Juego

FRQ-11:	Elegir canción
Descripción:	El sistema deberá mostrar las canciones que dispone para que el usuario seleccione una de ellas para el juego.

Tabla FRQ 11: Elegir canción

FRQ-12:	Niveles
Descripción:	De la misma forma que en la academia, el sistema deberá utilizar un sistema basado en niveles para controlar la dificultad.

Tabla FRQ 12: Niveles

Requisitos ostinatos/canciones:

FRQ-13:	Relacionar líneas rítmicas e instrumentos
Descripción:	El sistema deberá utilizar iconos para indicar al usuario que partitura de la canción corresponde tocarla a que instrumento.

Tabla FRQ 13: Relacionar líneas rítmicas e instrumentos

FRQ-14:	Longitud de las canciones
Descripción:	El sistema deberá de poder mostrar ostinatos de hasta 4 compases.

Tabla FRQ 14: Longitud de las canciones

FRQ-15:	Tiempo de compas
Descripción:	El sistema deberá poder ser utilizado con los siguientes tiempos de compas: 2/4, 3/4 y 4/4.

Tabla FRQ 15: Tiempo de compas

FRQ-16:	Claqueta de interpretación
Descripción:	El sistema deberá contener un cursor que vaya pasando por encima de las canciones para marcar el ritmo de la canción.

Tabla FRQ 16: Claqueta de interpretación

FRQ-17:	Canciones
Descripción:	El sistema deberá proporcionar varias canciones infantiles para la parte del juego.

Tabla FRQ 17: Canciones

4.1.2 Requisitos no funcionales

NFR-1:	Flash
Descripción:	El sistema deberá estar implementado utilizando la tecnología flash.

Tabla NRF 1: Flash

NFR-2:	Navegador web
Descripción:	El sistema deberá estar diseñado de forma que pueda ser utilizado desde navegador web.

Tabla NRF 2: Navegador web

NFR-3:	Pantalla táctil
Descripción:	El sistema deberá poder ser utilizado en dispositivos táctiles.

Tabla NRF 3: Pantalla táctil

NFR-4:	Orientado a niños
Descripción:	El sistema deberá estar diseñado de forma adecuada para niños de 3-5 años (textos, voces, dibujos...).

Tabla NRF 4: Orientado a niños

NFR-5:	Voz sintetizada
Descripción:	El sistema deberá utilizar voz sintetizada cuando el muñeco hable.

Tabla NRF 5: Voz sintetizada

NFR-6:	Muñeco robot
Descripción:	El sistema deberá utilizar un robot como muñeco para que “encaje” con la voz sintética.

Tabla NFR 6: Muñeco robot

4.1.3 Requisitos de información

IRQ-1:	Conjunto de instrumentos
Descripción:	El sistema deberá tener el conjunto de instrumentos que se detalla a continuación:
Crótalos Pandero Triangulo Claves Bombo Plato Charles Caja	

Tabla IRQ 1: Conjunto de instrumentos

IRQ-2:	Conjunto de figuras musicales y silencios
Descripción:	El sistema deberá tener el conjunto de figuras musicales y silencios que se detalla a continuación:
Corchea Doble corchea Negra Blanca Blanca con puntillo Redonda	

Y los correspondientes silencios.

Tabla IRQ 2: Conjunto de figuras musicales y silencios

IRQ-3:	Canciones
Descripción:	El sistema deberá tener el conjunto de canciones infantiles que se detalla a continuación:
Tengo una muñeca vestida de azul La vaca lechera, La gallina turuleca Mi burro enfermo	

Tabla IRQ 3: Canciones

4.2 Casos de uso

UC-1	<u>Introducción</u>
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso al arrancar la aplicación.
Precondiciones	No hay.
Flujo de eventos	
Activación	El usuario ingresa en la aplicación.
Flujo principal	
1. El sistema muestra al usuario la escena de introducción presentando la aplicación. 2. El usuario espera a que acabe la introducción. 3. El sistema pasa al menú principal y el caso de uso finaliza.	
Postcondición	No hay.
Flujos alternativos	

2. Si el usuario decide saltarse la introducción, el sistema no espera a haber acabado la introducción. Se continúa en el paso 3.

Tabla UC 1: Introducción

UC-2	Seleccionar nivel
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de selección de nivel.
Precondiciones	No hay.
Flujo de eventos	
Activación	El usuario ingresa en la escena de selección de niveles.
Flujo principal	
1. El sistema muestra al usuario los cuatro niveles para que este elija. 2. El usuario selecciona de entre uno de los cuatro. 3. El sistema guarda la elección del usuario y avanza a la siguiente pantalla y el caso de uso termina.	
Postcondición	El sistema almacena el nivel con el que se va a jugar.
Flujos alternativos	
2. Si el usuario elije volver a la pantalla anterior, el sistema retrocede al menú principal, el caso de uso finaliza.	

Tabla UC 2: Seleccionar nivel

UC-3	<u>Seleccionar los instrumentos</u>
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de selección de instrumentos.
Precondiciones	El usuario debe de haber elegido un nivel anteriormente en el caso de estar en la academia o debe haber elegido una canción en el caso de estar en el juego.
Flujo de eventos	
Activación	El usuario ingresa en la escena de selección de instrumentos.
Flujo principal	
<ol style="list-style-type: none"> 1. El sistema muestra al usuario el conjunto de instrumentos y el número de estos que puede elegir. 2. El usuario selecciona uno de ellos. 3. El sistema guarda la elección del usuario y comprueba el número de instrumentos que lleva elegidos. Se vuelve al paso 2. 4. El usuario indica que está listo para continuar. 5. El sistema guarda los instrumentos seleccionados y pasa a la pantalla siguiente. El caso de uso termina. 	
Postcondición	El sistema almacena los instrumentos que se van a utilizar en el juego.
Flujos alternativos	
<ol style="list-style-type: none"> 2. Si el usuario elije volver a la pantalla anterior, el sistema retrocede al menú de canciones (juego) o al menú de niveles (academia). El caso de uso finaliza. 3. Si el número de instrumentos elegidos está dentro del rango permitido, el sistema da la opción al usuario de empezar el juego, a continuación este caso de uso continúa. 3. Si el usuario selecciona alguno de los instrumentos que ya estaba seleccionado previamente, el sistema lo deselectiona. El caso de uso vuelve al paso 3. 4. Si el usuario aun no quiere comenzar el juego, el caso de uso vuelve al paso 2. 	

Tabla UC 3: Seleccionar los instrumentos

UC-4	<u>Seleccionar canción</u>
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de selección de canción.
Precondiciones	El usuario debe de haber elegido la opción de entrar al juego.
Flujo de eventos	
Activación	El usuario ingresa en la escena de selección de canción.
Flujo principal	
<ol style="list-style-type: none"> 1. El sistema muestra al usuario el conjunto de canciones disponibles. 2. El usuario selecciona una de ellas. 3. El sistema muestra la canción elegida como seleccionada. 4. El usuario decide continuar. 5. El sistema guarda la canción elegida. El caso de uso finaliza. 	
Postcondición	El sistema almacena la canción seleccionada.
Flujos alternativos	
<ol style="list-style-type: none"> 1. Si el usuario elije retroceder, el sistema vuelve al menú principal y el caso de uso finaliza. 4. Si el usuario decide escoger otra canción, el sistema deselecciona la anterior y marca como seleccionada la nueva. El caso de uso continúa. 	

Tabla UC 4: Seleccionar canción

UC-5	<u>Presentación academia</u>
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de presentación de la academia.
Precondiciones	El usuario debe de haber elegido la opción de entrar a la academia.
Flujo de eventos	
Activación	El usuario ingresa en la academia.
Flujo principal	
<ol style="list-style-type: none"> 1. El sistema muestra al usuario el conjunto de instrumentos básicos y le propone el primer juego. 2. El sistema le pide al usuario que toque determinado instrumento básico. 3. El usuario toca el instrumento correcto. 4. Al terminar, el sistema felicita al usuario y vuelve a ejecutar el juego con los instrumentos de la batería. 5. El sistema le pide al usuario que toque determinado instrumento de la batería. 6. El usuario toca el instrumento correcto. 7. Al terminar, el sistema felicita al usuario y le explica el funcionamiento de la academia, se pasa al menú de niveles y el caso de uso finaliza. 	
Postcondición	No hay.
Flujos alternativos	
<ol style="list-style-type: none"> 1. Si el usuario elije retroceder, el sistema vuelve al menú principal y el caso de uso finaliza. 1. Si el usuario decide avanzar, el sistema pasa al menú de niveles y el caso de uso finaliza. 3. El usuario no toca el instrumento correcto, el sistema le avisa y le da otro intento, se vuelve al paso 3. 4. Si el sistema no ha repetido la prueba determinado número de veces se vuelve al paso 2. 6. El usuario no toca el instrumento correcto, el sistema le avisa y le da otro intento, se vuelve al paso 6. 7. Si el sistema no ha repetido la prueba determinado número de veces se vuelve al paso 5. 	

--

Tabla UC 5: Presentación academia

UC-6	<u>Academia</u>
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de presentación de la academia.
Precondiciones	El usuario debe de haber seleccionado nivel e instrumentos.
Flujo de eventos	
Activación	El usuario ingresa en la escena de interpretación dentro de la academia.
Flujo principal	
1. El sistema muestra al usuario la pantalla de juego compuesta por la/s partitura/s del ostinato, instrumentos elegidos, el cursor de interpretación y botones para las opciones de escuchar y tocar. 2. El sistema le explica al usuario el funcionamiento. 3. El usuario selecciona una de las opciones. 4. El sistema da por finalizada la academia y pasa a la escena de niveles, el caso de uso finaliza.	
Postcondición	No hay.
Flujos alternativos	
3. Si el usuario selecciona la opción de escuchar, se realiza el caso de uso escuchar, a continuación el caso de uso vuelve al paso 3. 3. Si el usuario selecciona la opción de tocar, se realizar el caso de uso tocar, a continuación el caso de uso continua. 3. Si el usuario selecciona la opción de cambiar velocidad, se realizar el caso de uso cambiar velocidad, a continuación el caso de uso vuelve al paso 3.	

Tabla UC 6: Academia

UC-7	Escuchar
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los casos de uso: [UC-6: Academia] y [UC-10: Juego].
Precondiciones	No hay.
Flujo de eventos	
Activación	El usuario ha seleccionado la opción de escuchar.
Flujo principal	
1. El sistema realiza una demostración al usuario del ostinato que tiene en pantalla con los instrumentos correspondientes. El caso de uso termina.	
Postcondición	No hay.
Flujos alternativos	
1. Si el sistema esta realizando la demostración de una canción, se deberá acompañar de la canción correspondiente, el caso de uso finaliza.	

Tabla UC 7: Escuchar

UC-8	Tocar
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los casos de uso: [UC-6: Academia] y [UC-10: Juego].
Precondiciones	No hay.
Flujo de eventos	
Activación	El usuario ha seleccionado la opción de tocar.
Flujo principal	
<ol style="list-style-type: none"> 1. El sistema hace avanzar el cursor de interpretación por encima de las partituras. Se elimina la opción de escuchar. 2. El usuario interpreta la canción utilizando los instrumentos que aparecen en pantalla. 3. El sistema va calculando las puntuaciones dependiendo de los errores que vaya cometiendo el usuario. 4. Al terminar, el sistema felicita al usuario, le muestra la puntuación final y el caso de uso termina. 	
Postcondición	Se almacena el nivel actual como superado.
Flujos alternativos	
<ol style="list-style-type: none"> 1. Si el sistema está en el juego, se deberá acompañar de la canción correspondiente, el caso de uso continúa. 2. Si el usuario elige de nuevo la opción de tocar, el sistema reinicia el juego, a continuación el caso de uso vuelve al paso 1. 4. Si el usuario no consigue determinada puntuación, se avisa de que no se ha superado la prueba, se reinicia el juego. Se vuelve al paso 1. 	

Tabla UC 8: Tocar

UC-9	Modificar velocidad
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: [UC-6: Academia].
Precondiciones	No hay.
Flujo de eventos	
Activación	El usuario ha seleccionado la opción de modificar velocidad.
Flujo principal	
1. El usuario modifica la velocidad. 2. El sistema ajusta la velocidad del cursor y muestra al usuario la nueva velocidad. El caso de uso termina.	
Postcondición	No hay.
Flujos alternativos	
1. Si la velocidad sale fuera del intervalo de velocidades permitido, el sistema no modifica la velocidad, a continuación el caso de uso continua.	

Tabla UC 9: Modificar velocidad

UC-10	Juego
Descripción del caso de uso	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario entre en la pantalla de juego.
Precondiciones	Debe de haber seleccionado previamente canción e instrumentos.
Flujo de eventos	
Activación	El usuario entra en la escena del juego.
Flujo principal	
1. El sistema muestra al usuario la pantalla de juego compuesta por la/s partitura/s del ostinato, instrumentos elegidos, el cursor de interpretación y botones para las opciones de escuchar y tocar. 2. El sistema le explica al usuario el funcionamiento. 3. El usuario selecciona una de las opciones. 4. El sistema da por finalizado el juego y pasa a la escena de selección de canciones, el caso de uso finaliza.	
Postcondición	No hay.
Flujos alternativos	
3. Si el usuario selecciona la opción de escuchar, se realiza el caso de uso escuchar, a continuación el caso de uso vuelve al paso 3. 3. Si el usuario selecciona la opción de tocar, se realizar el caso de uso tocar, a continuación el caso de uso continua.	

Tabla UC 10: Juego

4.3 Diagrama de casos de uso

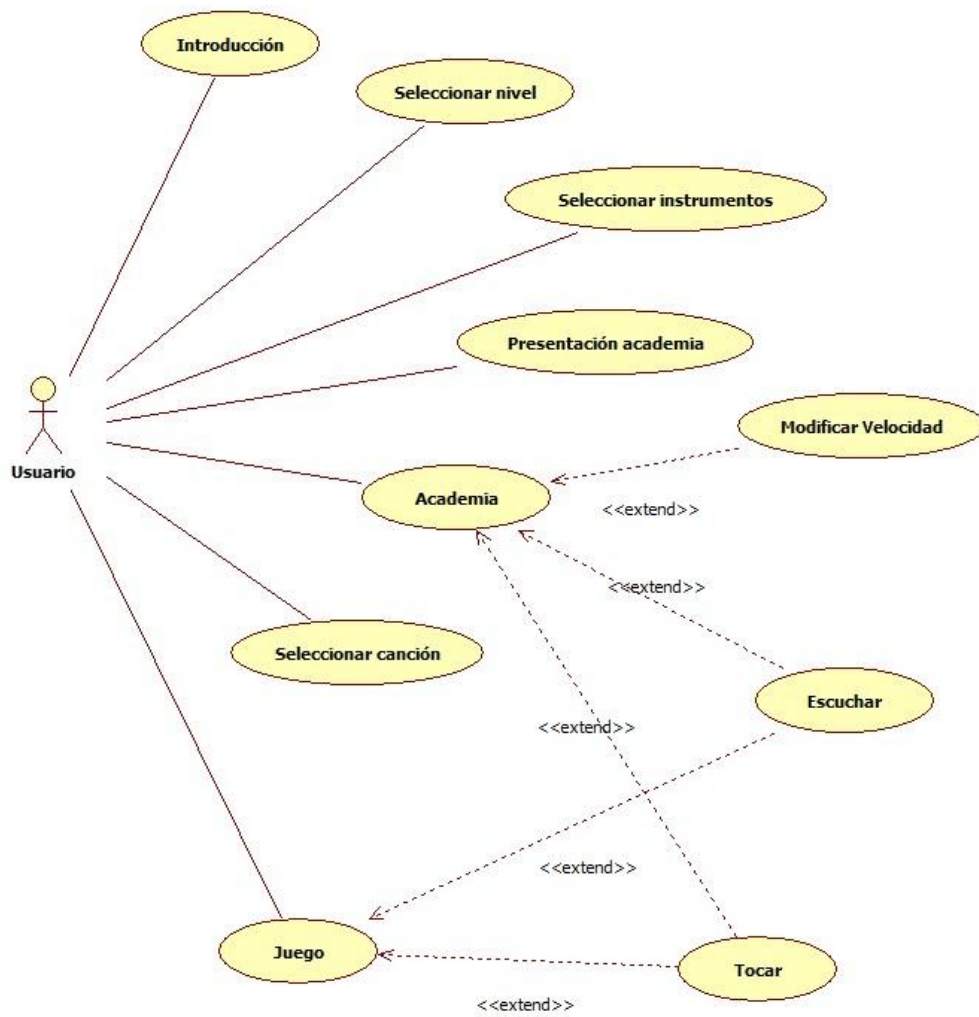


Figura 8: Diagrama de casos de uso

Capítulo 5

Diseño

5. Diseño

5.1 Arquitectura

Se utiliza la arquitectura de dos niveles: capa de interfaz y capa de lógica. No se utiliza capa de persistencia pues no es necesaria, todo lo persistente que utiliza la aplicación (imágenes, animaciones, sonidos, canciones...) está soportado por la plataforma Flash.

En resumen: en la capa de lógica se llevará el funcionamiento de la aplicación y en la capa de interfaz se controlaran todos los elementos que aparecen por pantalla.

En el apartado 5.4 Diagrama de paquetes se puede ver con más detalle.

5.2 Diagramas de secuencia

Introducción:

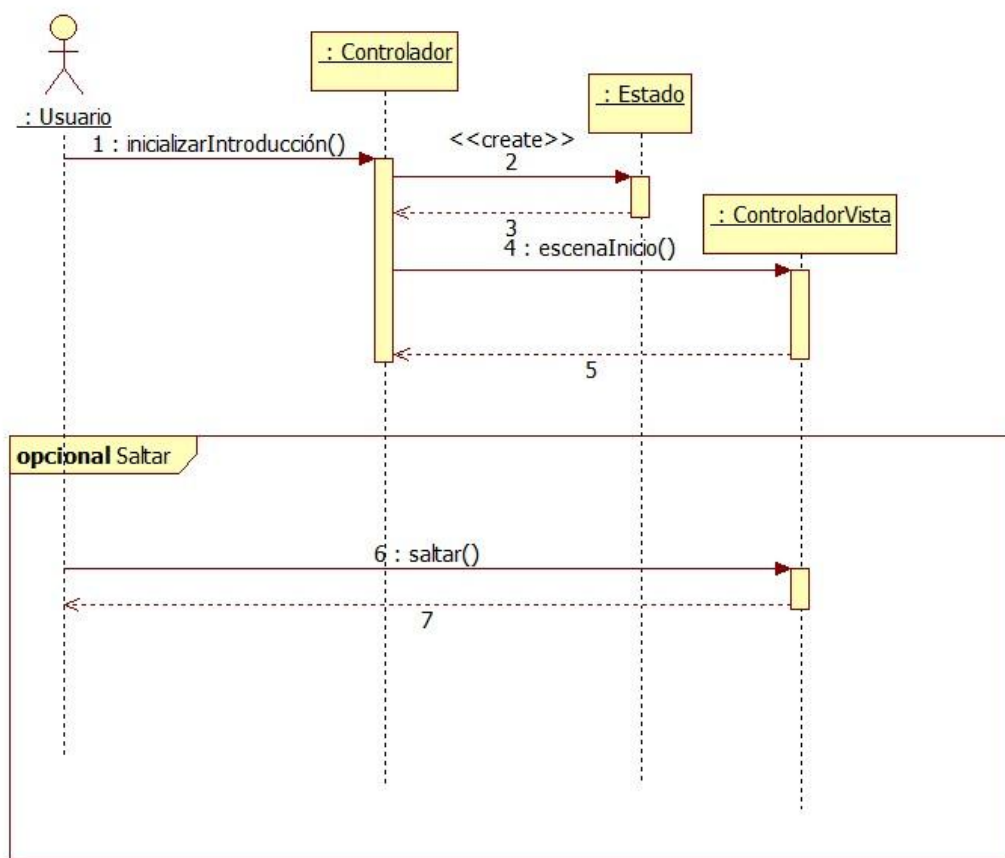


Figura 9: DS Introducción

Seleccionar nivel:

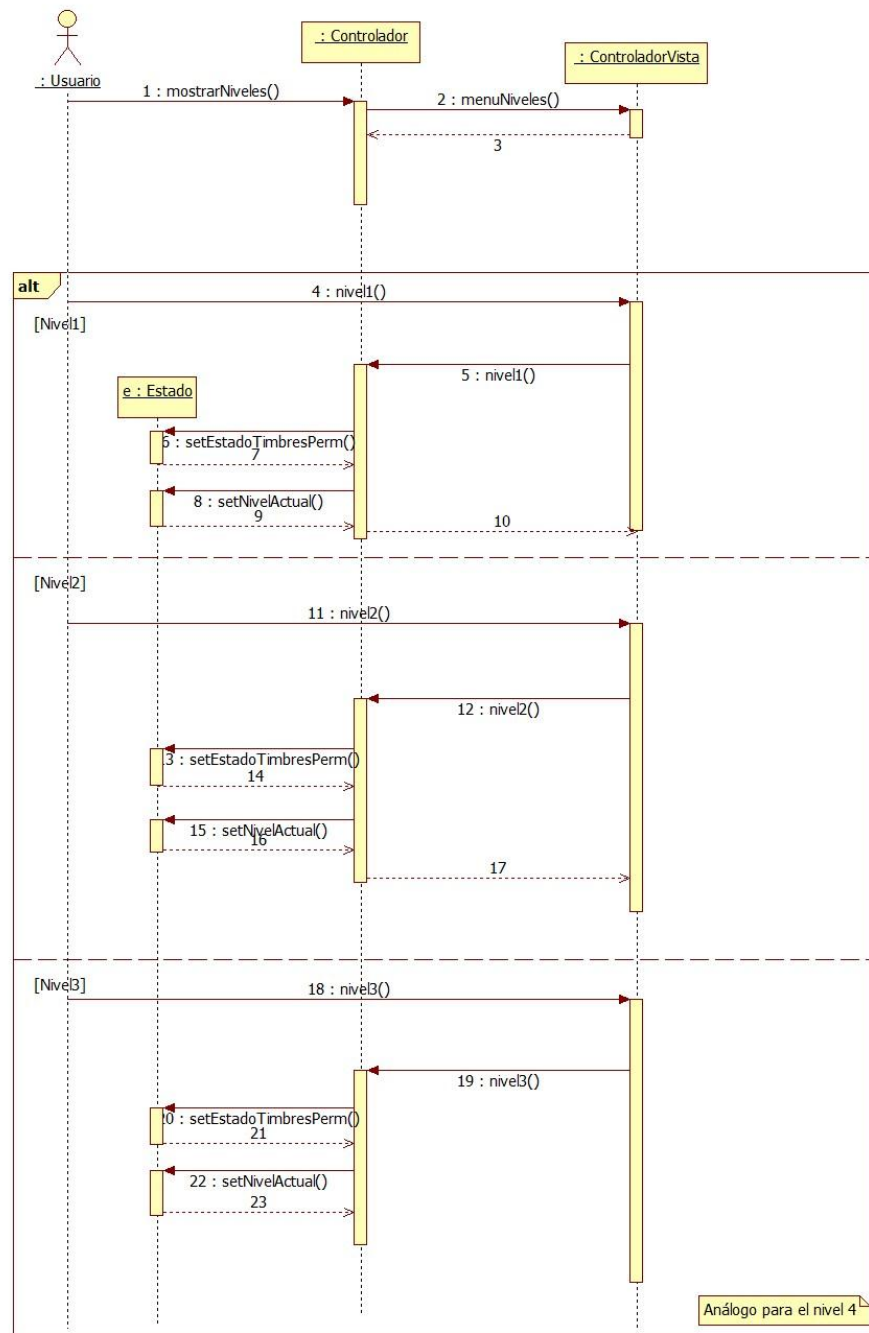


Figura 10: DS seleccionar nivel

Seleccionar Instrumentos:

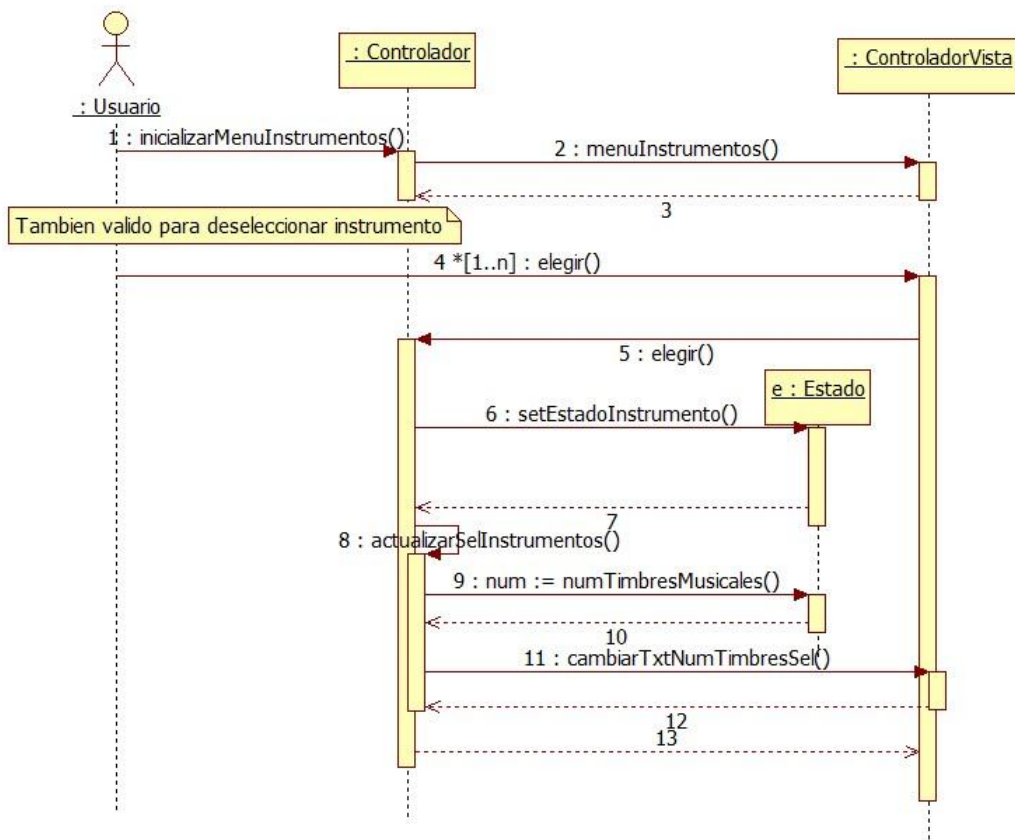


Figura 11: DS seleccionar instrumentos

Presentación academia:

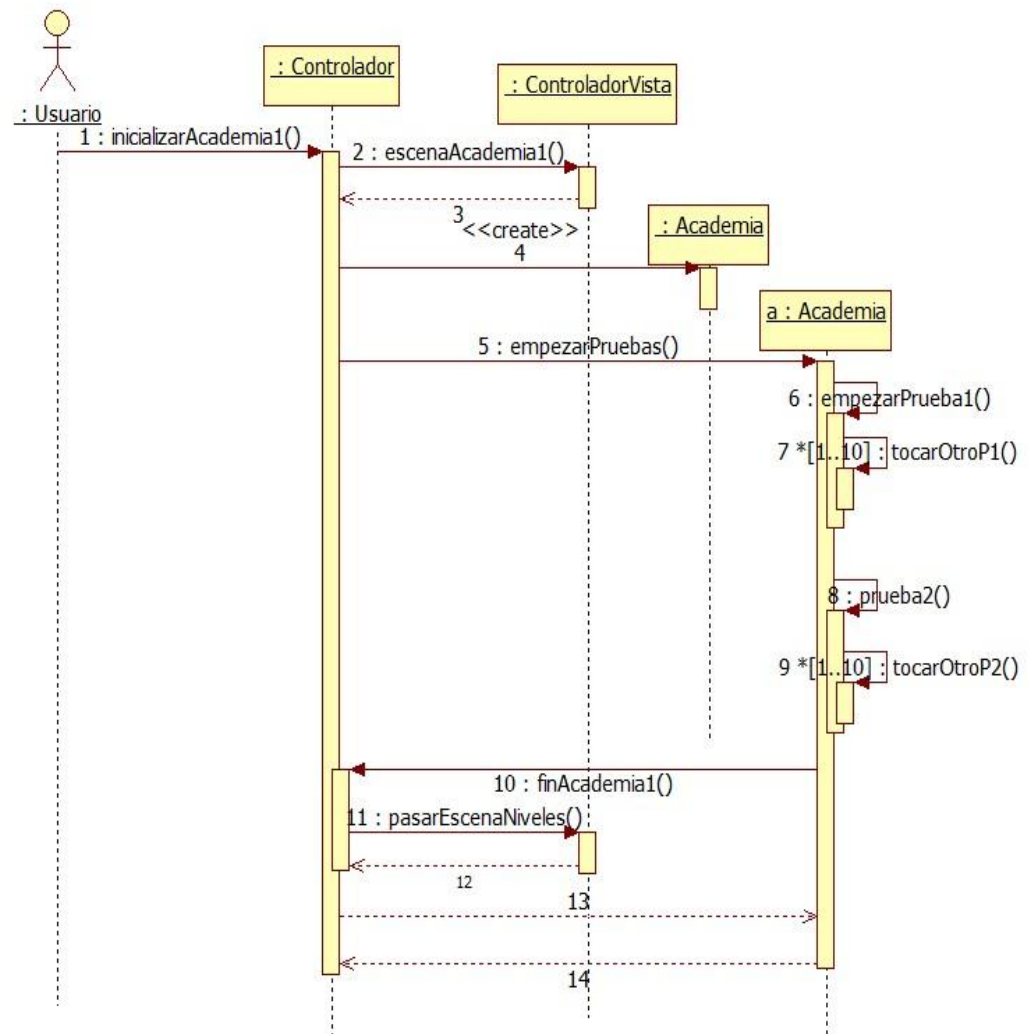


Figura 12: DS presentación academia

Academia:

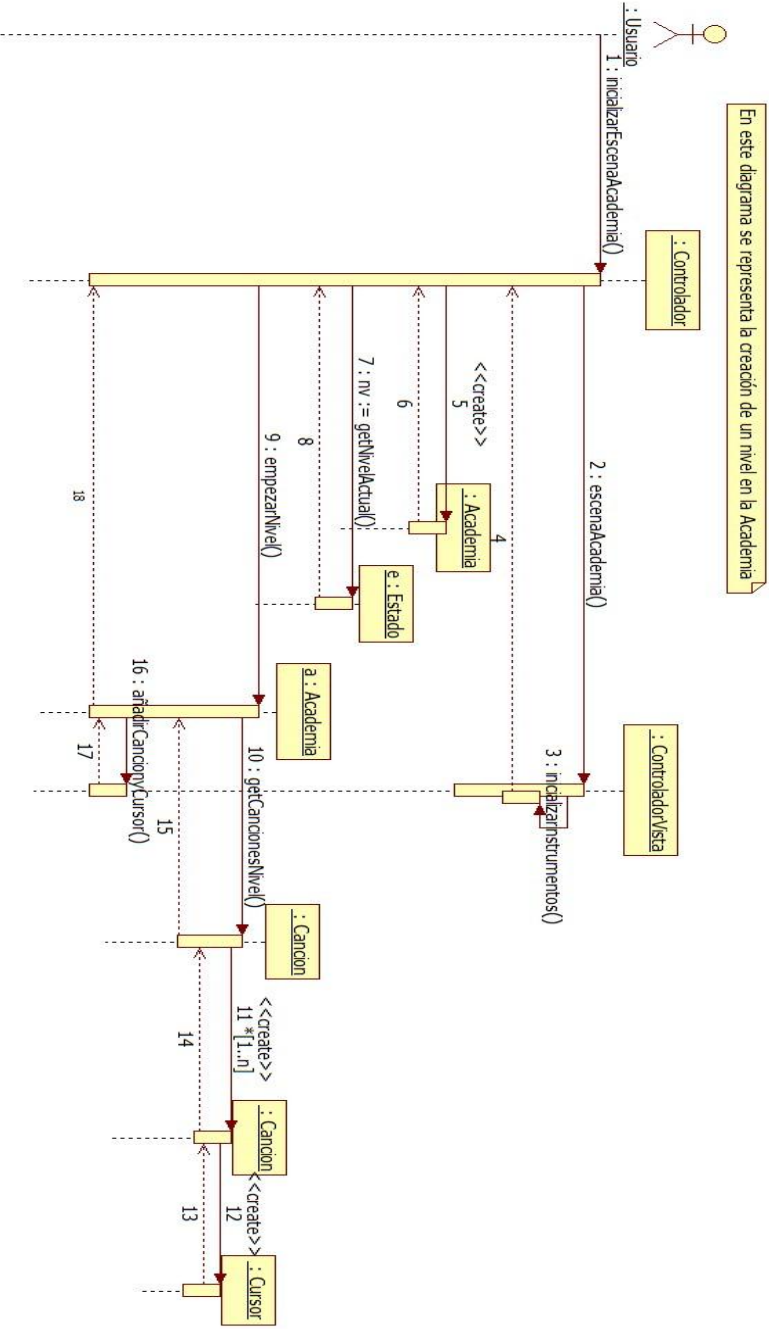


Figura 13: DS academia

Tocar:

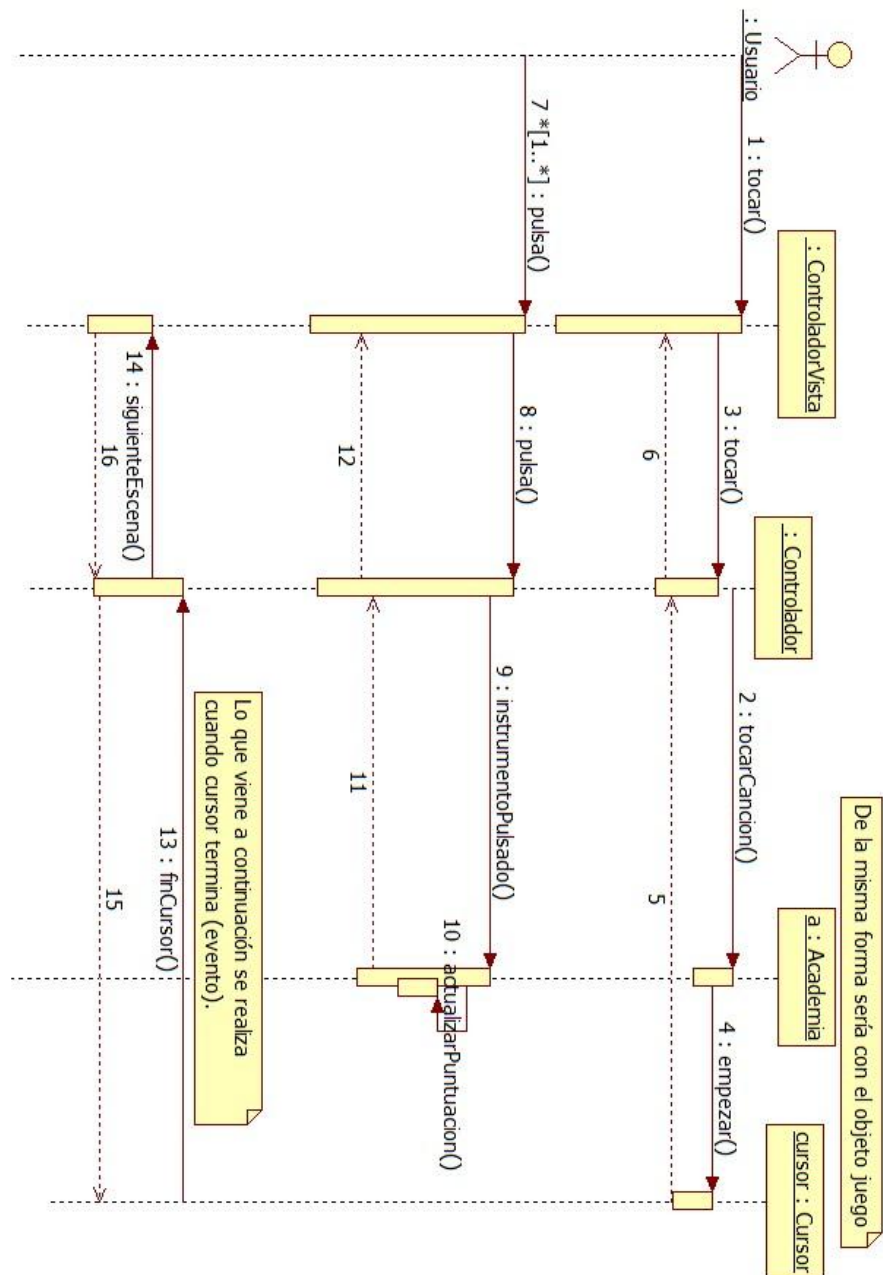


Figura 14: DS tocar

Escuchar:

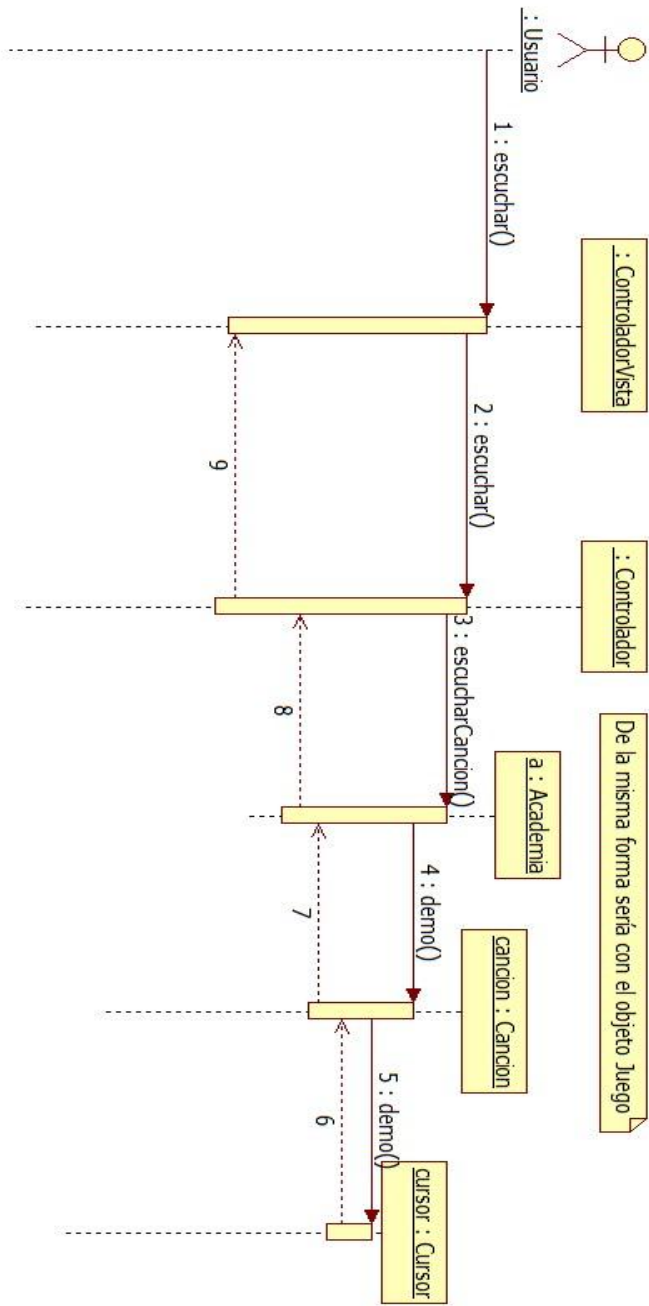


Figura 15: DS escuchar

Seleccionar canción:

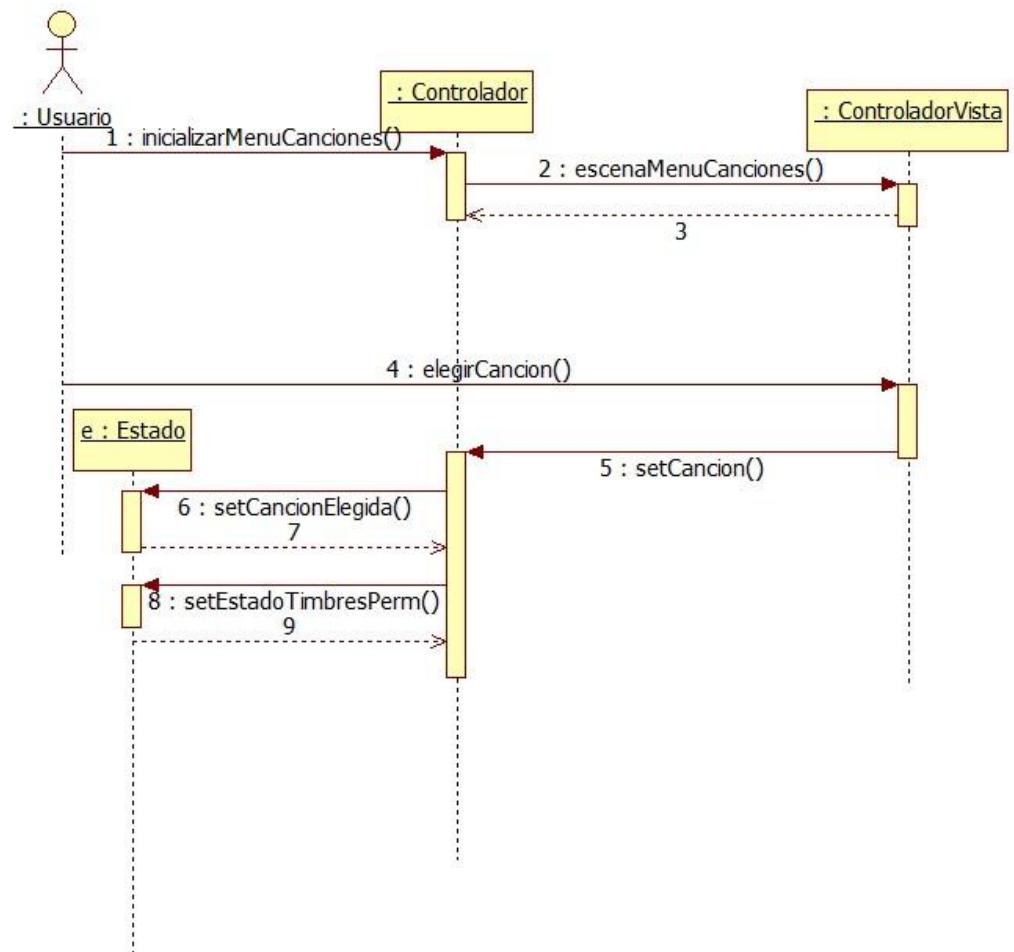


Figura 16: DS seleccionar canción

Juego:

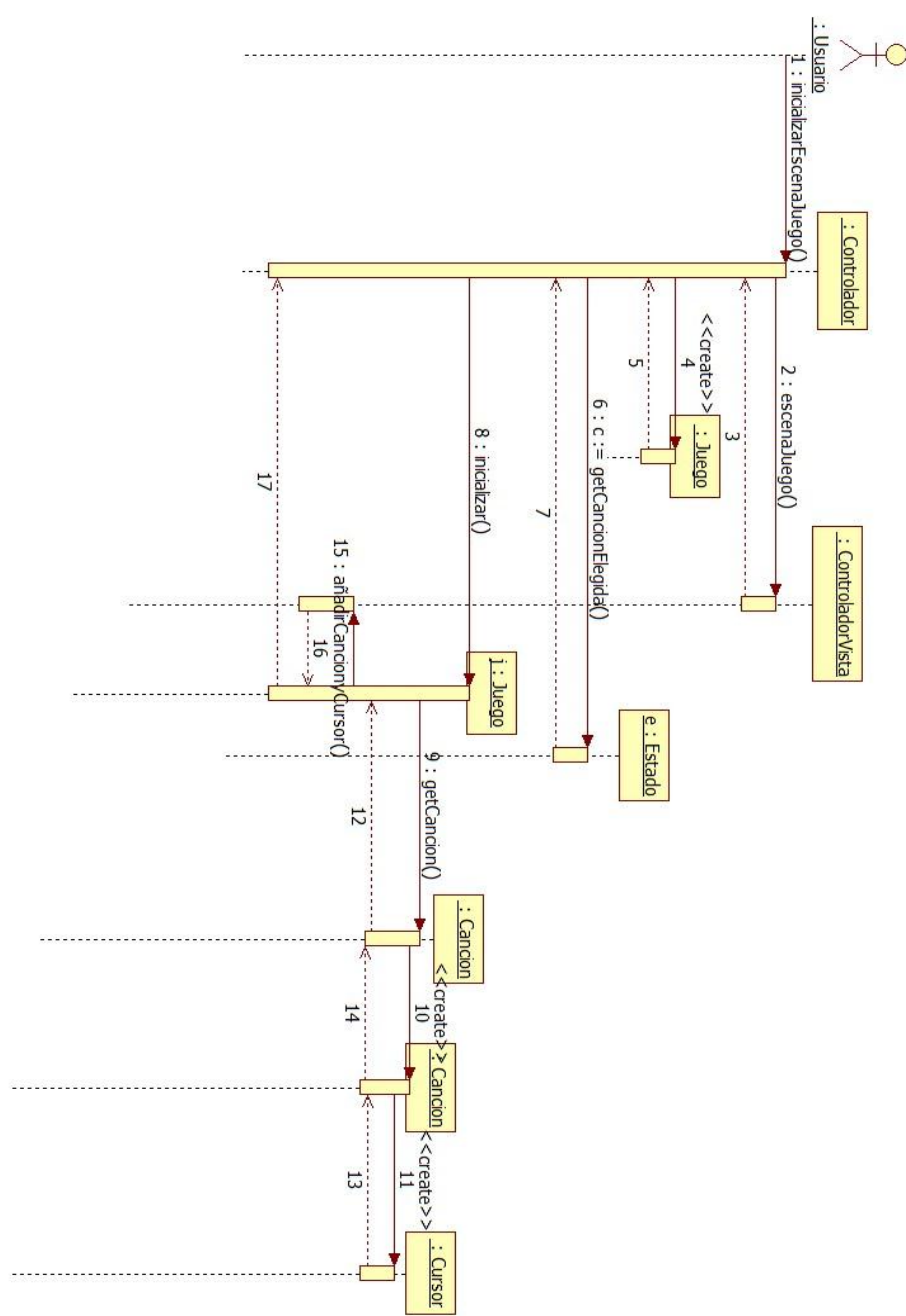


Figura 17: DS juego

Modificar velocidad:

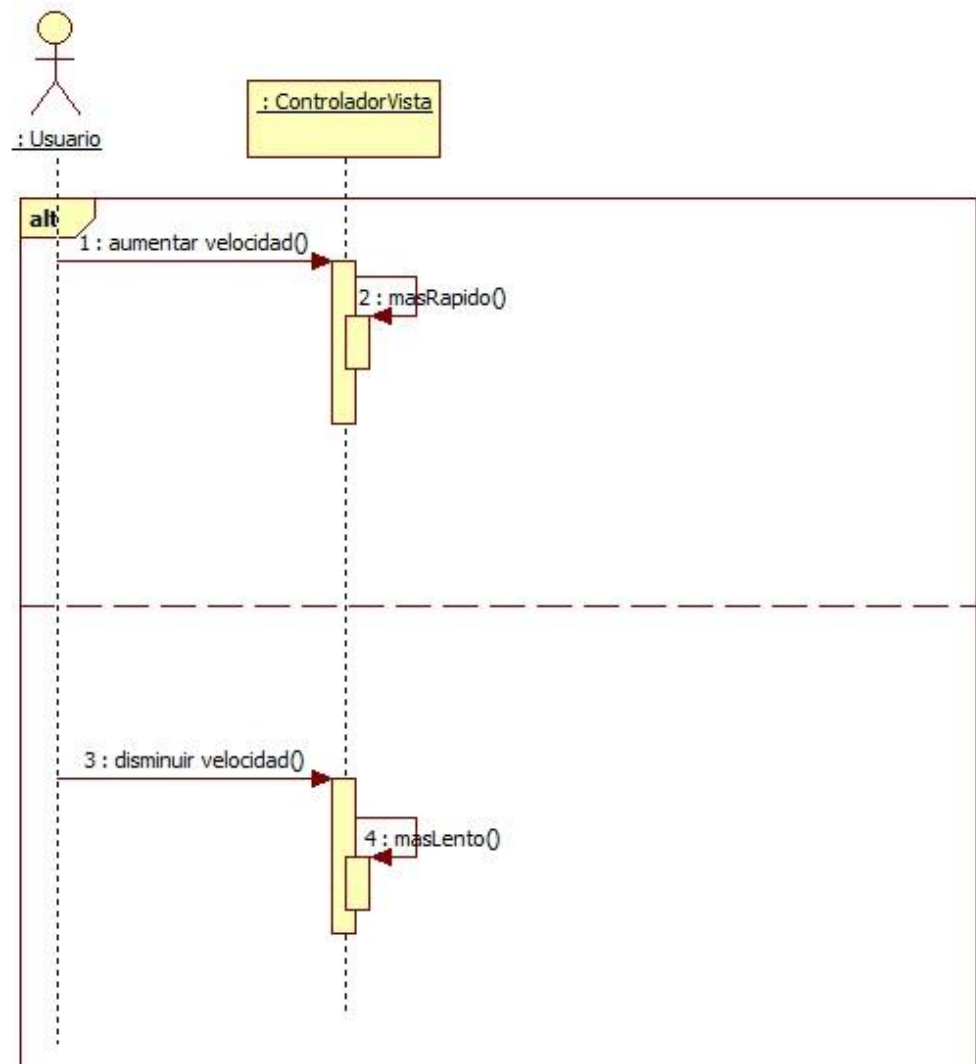


Figura 18: DS modificar velocidad

5.3 Diagrama de clases

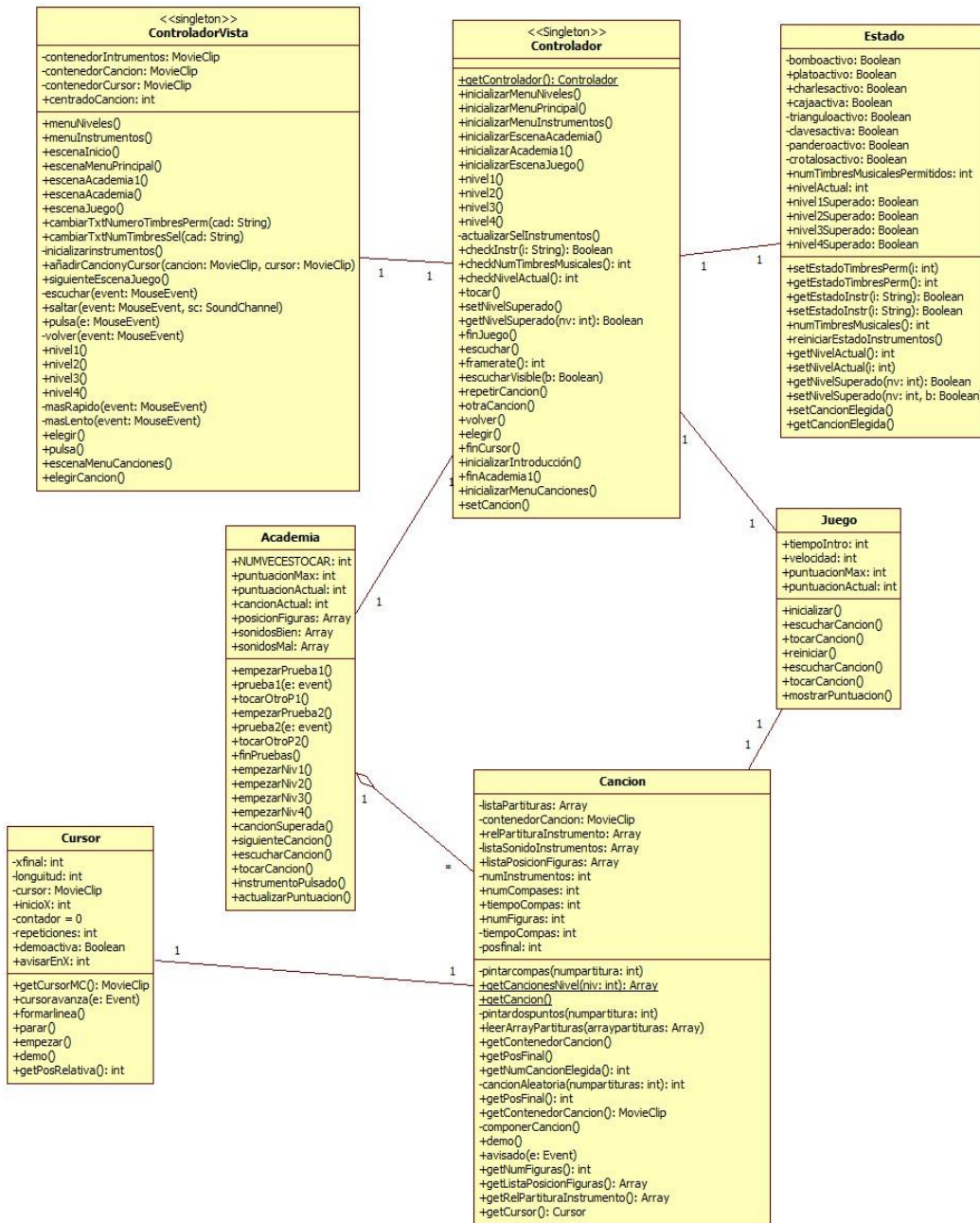


Figura 19: Diagrama de clases

5.4 Diagrama de paquetes

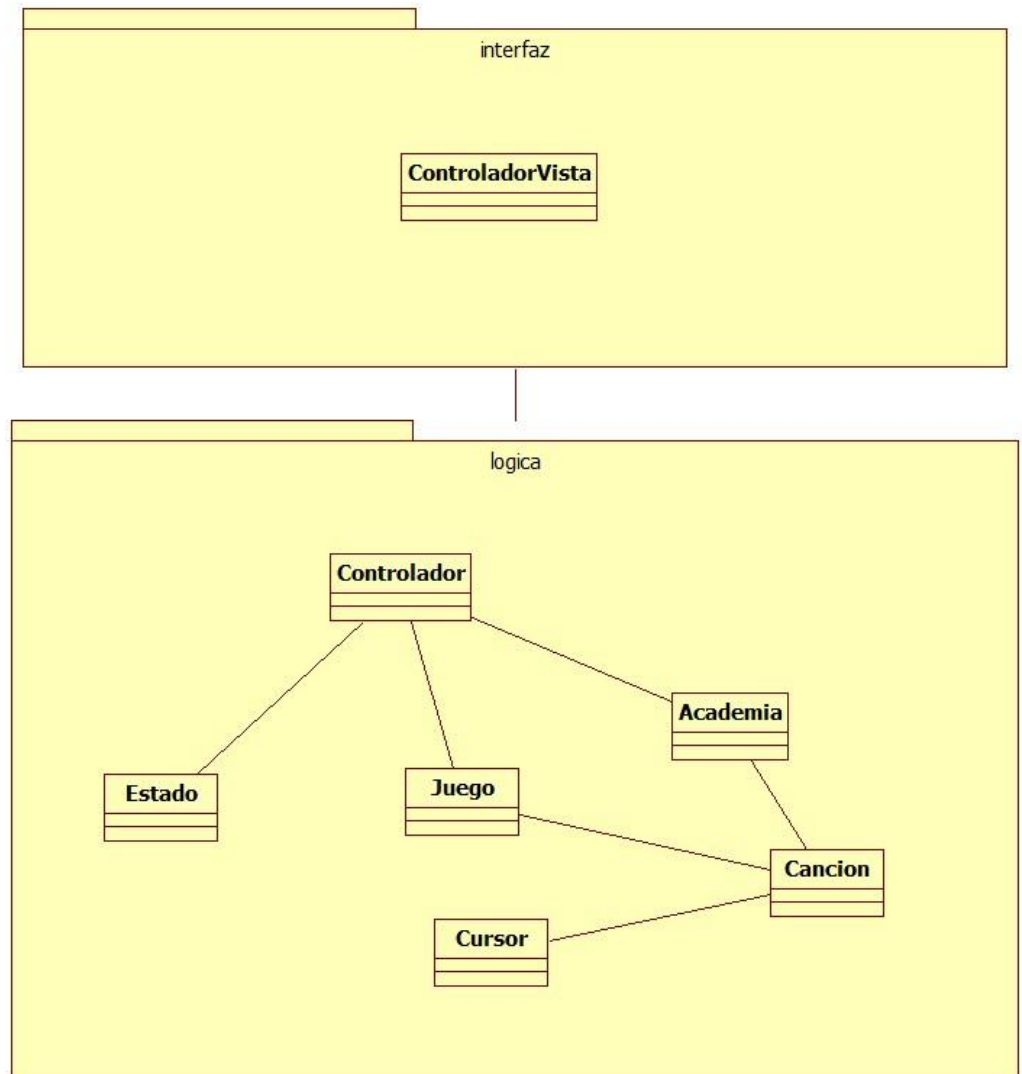


Figura 20: Diagrama de paquetes

Capítulo 6

Implementación y Pruebas

6. Implementación y pruebas

6.1. Implementación

En esta parte se explicará en detalle algunas de las funciones de cada clase que pueden ser más problemáticas a la hora de entender.

Antes de empezar, hay que aclarar que no todo el código escrito está en las clases. Se ha intentado que la mayor parte estuviera en archivos externos.

Flash permite escribir código en cada fotograma, en nuestro caso, solo se ha escrito la llamada al método de inicialización de ese fotograma. Como se puede ver utilizando Adobe Flash, todo está dividido en varias escenas, en concreto son las siguientes: EscenaInicio, EscenaInterpretar, EscenaInstrumentos, EscenaJuego, EscenaAcademia1, EscenaAcademiaJuego y EscenaNiveles. Para ver la ventana de escenas pulse (shift+f9).

Así que por ejemplo: Si se va a la escena de selección de niveles y vemos el código escrito (pulsar f9) nos encontraremos con:

```
stop();  
var co=Controlador.getControlador();  
co.inicializarMenuNiveles();
```

En la escena de instrumentos se tiene:

```
stop();  
co.inicializarMenuInstrumentos();
```

Así para todas. Cada vez que se entre en esa escena se ejecutara ese código.

Clase ControladorVista

Corresponde con la clase de documento de Adobe Flash es por eso por lo que debe estar en la misma carpeta donde está el archivo .fla.

La clase del documento tiene acceso directo a los elementos predefinidos que aparecen en los distintos fotogramas en Flash. Es por eso por lo que se pone esta clase como la clase del documento.

Esta clase se creará automáticamente cada vez que se lance la aplicación así que para utilizar el patrón Singleton en esta clase se necesita añadir esta línea en el método de creación.

```
public function ControladorVista(){  
    singleton=this;  
}
```

Los métodos de inicialización de escenas de esta clase básicamente lo que hacen es añadir los listeners a los distintos botones de dicha escena y añadir los textos a los cuadros de texto, no hay mucho más que comentar.

Respecto a los eventos de botón en esta Clase generalmente solo se avisa al Controlador de que se ha pulsado determinado botón y si ControladorVista tiene que realizar algún cambio en pantalla, lo hace.

Para intentar de tratar organizadamente los elementos en pantalla se utilizan contenedores. A continuación se explica brevemente cómo funcionan los contenedores en Flash.

En ActionScript 3.0, la clase DisplayObject es la clase base de todos los objetos que pueden situarse en la lista de visualización. La lista de visualización administra todos los objetos mostrados en Flash Player. Se utiliza la clase DisplayObjectContainer para organizar los objetos de visualización en la lista de visualización.

Hay varios tipos de contenedores: Sprite, MovieClip... en nuestro caso se ha utilizado MovieClip. Estos contenedores son hijos de DisplayObjectContainer. En la figura 21 se muestran las subclases de DisplayObject.

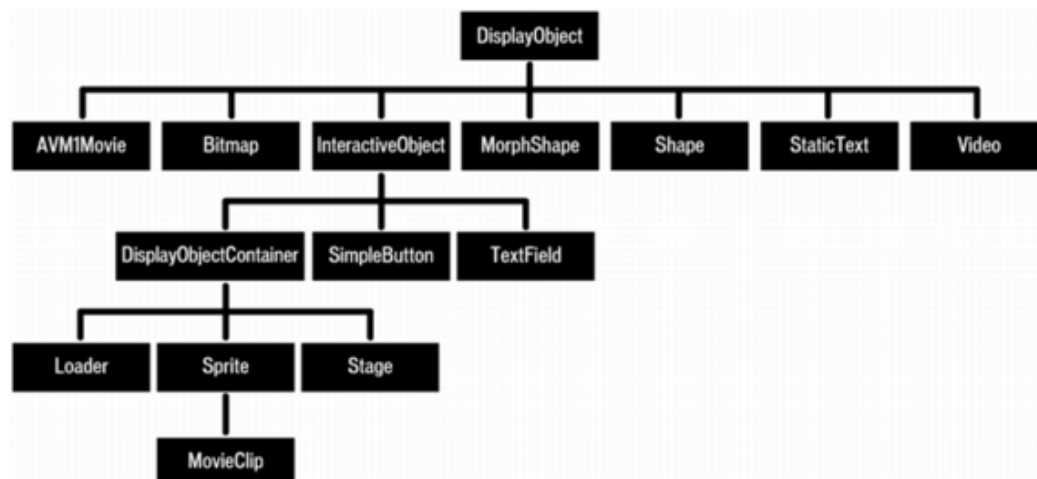


Figura 21: Subclases de DisplayObject

Se han colocado tres contenedores MovieClip en la clase ControladorVista (atributos): contenedorInstrumentos, contenedorCancion y contenedorCursor.

El método inicializarinstrumentos() va añadiendo los botones de los instrumentos en el contenedorInstrumentos. Para ello utiliza los distintos métodos de añadir instrumentos. Para que aparezcan los instrumentos centrados se tiene el array posicionInicial. Este array guarda la posición inicial donde se tiene que empezar a añadir los instrumentos. Los otros dos contenedores no les llena esta clase, sino que serán otras, principalmente la clase Academia y Juego las que le pasen a ControladorVista los MovieClips de las canciones y cursores utilizando el método añadirCancionyCursor().

El método framerate() se utiliza para el cambio de velocidad. Sobre el cambio de velocidad hay que tener en cuenta de que si esta aplicación se va a ejecutar desde navegador web los fps máximos permitidos son de 60 mientras que si se va a utilizar Flash Player no tenemos esta restricción. Para aumentar la compatibilidad de esta aplicación, se deja un intervalo de [20-50] fps (por defecto la velocidad es de 30fps).

Hay que tener una cosa en cuenta cuando se están utilizando las canciones infantiles. A pesar de que los navegadores web permiten hasta 60fps, surgen problemas de sincronización entre el ritmo de la canción y el cursor. Se recomienda fijar una velocidad de 30fps para estos casos.

Clase Controlador

Esta clase tiene asociada una clase auxiliar llamada clase Estado que, como su propio nombre indica, guarda la información asociada al estado de la aplicación en determinado momento (nivel elegido, instrumentos elegidos, los niveles que hay superados hasta el momento...).

También se puede encontrar métodos de inicialización de escena aquí. La diferencia con los de ControladorVista es que estos tratan más de la parte lógica de la aplicación.

El objeto Estado se crea cada vez que se entra a la aplicación (EscenaInicio).

Clase Estado

Comentada anteriormente.

Clase Canción

El método de creación de esta clase recibe 5 argumentos. El primero es el número de instrumentos que va a tener la canción, el segundo es el array de partituras que se comentará después, el tercero indica las repeticiones de la canción, el cuarto sirve para indicar el espacio al que se colocarán las figuras. Esto se indica pasándole la distancia de una corchea, las de las demás se calcularán a partir de esta. Por último se le indica la velocidad.

La primera funcionalidad que se encuentra en esta clase es la de dibujar las canciones utilizando MovieClips. Esto se hace en el método leerArrayPartituras() que se ejecuta en la creación. El array de partituras que se pasa en la creación no es más que un array que tiene las figuras codificadas de la partitura. A continuación se explica la codificación.

Estos arrays tienen varias partes: la partitura, el número de compases y el tiempo del compás. Las canciones que tengan varias partituras se separan en distintos ítems en el array mientras que las figuras de cada partitura se separan por el delimitador‘,’. Las figuras se codifican por su inicial excepto la blanca con puntillo que se codifica como ‘bp’ y la doble corchea que es ‘cc’. Si se trata de un silencio simplemente se pone una ‘s’ delante.

Por poner un ejemplo con un array de dos elementos:

`"n,sn,n,sn,n,sn,n,sn","sn,n,sn,n,sn,n,sn,n,4,2,"`

Es una canción cuyo tiempo de compas es 2/4, tiene 4 compases y está compuesta por 2 partituras donde la primera es: negra, silencio de negra, negra, silencio de negra, negra, silencio de negra, negra, silencio de negra y la segunda es: silencio de negra, negra, silencio de negra, negra, silencio de negra, negra. Como se puede observar, el tiempo de compas y el número de compases solo se indican en el último ítem del array.

Como las pizarras electrónicas tienen el problema de no ser multitáctiles las partituras están repletas de silencios y para que la lectura sea sencilla (no olvidemos que la aplicación es para niños de 3-5 años) estos silencios no se muestran. En ocasiones es necesario mostrar los silencios, para ello, se añade una v al final de la codificación del silencio. Por ejemplo:

`n,n,n,snv,2,`

Si solo tuviésemos tres negras el compás del final quedaría incompleto.

A parte de dibujar las partituras, el método `leerArrayPartituras()` va recabando información sobre la canción (número de figuras, posición final, posiciones de las figuras...)

Este método utiliza otra llamado `componerCancion()` que se encarga de poner los iconos de los instrumentos, el tiempo de compas, líneas horizontales...

El `MovieClip` se guarda en el atributo `contenedorCancion`. Una vez hecho esto se crea un objeto `Cursor`. Se hablará mas adelante sobre la clase `Cursor`.

Otra de las cosas que hace esta clase es la de ejecutar una demostración de la canción. Se utiliza eventos para ello. El método que empieza la demostración es el método `demo()` y el método manejador de eventos es `avisado()`.

La idea es la siguiente: El objeto Canción obtiene la posición de la siguiente figura que tiene que sonar y le dice al cursor que le avise cuando llegue a esa posición. Cuando Canción es avisado activa el sonido del instrumento correspondiente y busca la posición de la siguiente figura a tocar y la operación se repite. Hay que tener en cuenta que el

cursor tiene que estar en “estado” demostración, es decir, haber ejecutado el método de `Cursor demo()`.

En el caso de que estemos en la parte del juego, las canciones van acompañadas de una canción infantil. Para asignar canciones al objeto se utiliza el método `asignarMusica()` pasándole el objeto `Sound` correspondiente.

La clase `Canción` tiene dos métodos estáticos que nos devuelven las Canciones predefinidas para la academia y el juego. Estos son los métodos `getCancion()` para el juego y `getCancionNivel` para la academia.

Clase Cursor

El método de creación de `Cursor` tiene cuatro parámetros. El primero indica el número de instrumentos de la canción, se utiliza en el método `formarlinea()` para saber el tamaño que debe tener el cursor, el segundo es la posición final a la que tiene que llegar el cursor pues dependiendo de si la canción es 2/4, 3/4 o 4/4 y el número de compases esta es más larga o más corta. El tercero es el número de vueltas (repeticiones) que tiene que dar el cursor y el último la velocidad. Esta clase tiene los típicos métodos para que el cursor avance, se pare, se reinicie...

Para que el cursor se mueva de forma “continua” se utiliza el evento `ENTER_FRAME` de Flash. Este método hace que se llame al correspondiente método manejador del evento y que se ejecute en cada frame. Es justamente este método el que indica cuando se ha acabado una interpretación. La velocidad que se le pasa indica cuantos puntos avanza el cursor en cada frame.

Otro detalle que puede resultar confuso es el método `getPosRelativa()`. Podemos obtener la posición del cursor simplemente preguntando a su propiedad `x` (`cursor.x`), sin embargo, esta posición no nos sirve para nada pues lo mas seguro es que lo que queramos saber es en que posición se encuentra el cursor respecto a la partitura que aparece en pantalla. Es por eso por lo que se debe utilizar el método `getPosRelativa()`.

Clase Academia

La clase academia, como su propio nombre indica, se encarga de todo lo referente a la academia. Tanto la prueba 1 de la academia como la prueba 2 tienen un funcionamiento similar.

La prueba 1 comienza cuando es llamado el método `empezarPrueba1()`. Se usan eventos de tiempo para mover al muñeco durante la presentación de las pruebas. Cuando el muñeco acabe de explicar el funcionamiento de la prueba 1 se ejecutará el método `prueba1()`. Este método hace uso de `tocarOtroP1()` que elige aleatoriamente un instrumento para mandar al usuario que toque.

La prueba 2, como se dijo antes, es similar. Lo único diferente pueden ser la creación de los fuegos artificiales que aparecen en `tocarOtroP2()` cuando se acaban las pruebas. En ambos casos el método que se usa como manejador de eventos para los clicks en los instrumentos es `pboton()` que es donde se comprueba si la pulsación es correcta o no.

La academia trata además la parte de los niveles. Como se puede comprobar, hay un método para cada nivel denominados `empezarNivelX()` donde X es el número del nivel. Funcionan de forma parecida. Se pide a `Cancion` que cree las canciones del nivel correspondiente, se inicializan atributos y se pide a `ControladorVista` que pinte la canción y el cursor que corresponda.

Para el cálculo de puntuaciones mientras se está tocando una partitura la aplicación procede de la siguiente manera: Con cada pulsación de instrumento se obtiene cuál es la posición de la figura que se debería de haber tocado, hay que tener en cuenta que puede haber partituras con 2 instrumentos así que también se obtiene a qué instrumento corresponde. Se calcula la distancia entre la pulsación del usuario y la posición de la figura. Dependiendo del error, se subirán determinada cantidad de puntos. Si el error es muy pequeño, se dará la máxima cantidad de puntos.

La puntuación no se muestra en pantalla con números sino con estrellas (un total de 5) de forma que si el usuario ha dado correctamente a todas las figuras con el instrumento adecuado habrá obtenido 5. Sino, se calcula el cociente de la cantidad máxima de puntos que se podrían haber obtenido y los que se han obtenido y se iluminarán el número de estrellas correspondientes.

En esta clase se pueden encontrar las llamadas a los métodos que se comentaron antes en la clase `Cancion`, `canción.demo()` y `cursor.demo()`.

Hay un último detalle en esta clase. Cuando se da al botón de atrás o adelante hay que tener en cuenta que no se puede hacer sin mas. Hay que eliminar todos los eventos para que no den errores al cambiar de escena al no encontrar el objeto correspondiente. El método donde se contempla esto es `borrarEventos()`.

Clase Juego

Por último, tenemos la clase juego. Hemos comentado que en el juego las partituras van acompañadas de la canción infantil que se seleccione, esto provoca el tener que sincronizar la velocidad del cursor con el ritmo de la canción. Se empezará comentando como se ha realizado este cálculo por si en un futuro se decide el añadir mas canciones sea más sencillo.

Una vez editadas, las cuatro canciones infantiles que tiene el juego están compuestas por 4 compases de introducción y 8 de melodía. Lo primero es calcular lo más exactamente posible cuanto dura la introducción. Por experiencia propia, estas canciones tienen una velocidad similar así que los tiempos no deberían diferenciarse demasiado. En general, la introducción esta en torno a 8 segundos. Por añadir facilidad a la hora de tocar, el cursor se empieza a mover en el 3º compas de introducción así que a parte del tiempo de la introducción hay que calcular el tiempo en el que entra el 3º compás.

Calculado esto, ahora debemos centrarnos en la velocidad del cursor. Esta velocidad depende de cuantos fotogramas por segundo utilice la aplicación y la velocidad que le hayamos pasado al cursor en su creación. Como se comentó anteriormente, se recomienda trabajar a 30fps así que esto se fijara a 30 fotogramas por segundo. Para calcular la velocidad del cursor debemos tener en cuenta cuanto espacio debe recorrer y en que tiempo. En el juego, la aplicación crea las canciones con una distancia de corchea de 20 puntos. Un compas 4/4 estará compuesto por 4 tiempos de negra, es decir, cada compas ocupa 160 puntos. Como tenemos 8 compases el cursor debe recorrer 1280 puntos. Intentamos obtener cuantos segundos dura la melodía exactamente. Teniendo estas 2 cosas y sabiendo que nuestra aplicación tiene 30fps podemos calcular cuanto espacio debe recorrer el cursor en un frame. Seguramente el cursor vaya perdiendo poco a poco el ritmo así que, una vez hecho esto, deberemos ajustar lo que podamos la velocidad del cursor hasta obtener el resultado deseado.

Como en anteriores ocasiones, las canciones predeterminadas se cargan usando el método estático de Canción llamado getCancion() y pasándole el nombre de la canción infantil y el numero del nivel.

6.2. Pruebas

- Pruebas para Tabla UC 1: Introducción:

CP-1	
Objetivo	Comprobar el paso de escena desde la introducción hasta el menú principal.
Entrada	Esperar a la finalización de la explicación.
Salida	Paso de escena.
Resultado	Correcto

Tabla CP 1: Introducción

CP-2	
Objetivo	Comprobar el paso de escena desde la introducción hasta el menú principal.
Entrada	Se utiliza el botón de saltar escena.
Salida	Paso de escena.
Resultado	Correcto

Tabla CP 2: Introducción

- Pruebas para Tabla UC 2: Seleccionar nivel

CP-3	
Objetivo	Comprobar la selección correcta del nivel correspondiente.
Entrada	Se selecciona el nivel 1.
Salida	La partitura cargada es la esperada para el nivel 1.
Resultado	Correcto

Tabla CP 3: Selección nivel

CP-4	
Objetivo	Comprobar la selección correcta del nivel correspondiente.
Entrada	Se selecciona el nivel 3.
Salida	La partitura cargada es la esperada para el nivel 3.
Resultado	Correcto

Tabla CP 4: Selección nivel

- Pruebas para Tabla UC 3: Seleccionar los instrumentos

CP-5	
Objetivo	Comprobar el número de selección de instrumentos.
Prerrequisitos	Se ha elegido un nivel que requiere un instrumento (nivel 1 en academia).
Entrada	Se selecciona el triángulo.
Salida	Aparece el botón de listo. El triángulo aparece seleccionado.
Resultado	Correcto

Tabla CP 5: Selección Instrumentos

CP-6	
Objetivo	Comprobar el número de selección de instrumentos.
Prerrequisitos	Se ha elegido un nivel que requiere dos instrumentos (nivel 4 en academia).
Entrada	Se selecciona el tambor y el plato.
Salida	Aparece el botón de listo. El tambor y el plato aparecen seleccionados.
Resultado	Correcto

Tabla CP 6: Selección Instrumentos

CP-7	
Objetivo	Comprobar el número de selección de instrumentos.
Prerrequisitos	Se ha elegido un nivel que requiere dos instrumentos (nivel 4 en academia).
Entrada	Se selecciona el solo charles.
Salida	No aparece el botón de listo. El charles aparece seleccionado.
Resultado	Correcto

Tabla CP 7: Selección Instrumentos

- Pruebas para Tabla UC 4: Seleccionar canción

CP-8	
Objetivo	Comprobar la selección de una canción
Prerrequisitos	No hay.
Entrada	Se selecciona la canción de la muñeca vestida de azul.
Salida	Aparece la canción como seleccionada.
Resultado	Correcto

Tabla CP 8: Selección de canción

CP-9	
Objetivo	Comprobar la selección de otra canción
Prerrequisitos	Ya hay una canción seleccionada (la muñeca vestida de azul).
Entrada	Se selecciona la canción mi burro enfermo.
Salida	Aparece la canción de mi burro como seleccionada y se deselecta la de la muñeca.
Resultado	Correcto

Tabla CP 9: Selección de canción

- Pruebas para Tabla UC 5: Presentación academia

CP-10	
Objetivo	Selección de instrumento correcto
Prerrequisitos	El muñeco debe haber mandado tocar un instrumento.
Entrada	Se toca el instrumento correcto
Salida	Mensaje de acierto por parte del muñeco.
Resultado	Correcto

Tabla CP 10: Presentación academia

CP-11	
Objetivo	Selección de instrumento incorrecto
Prerrequisitos	El muñeco debe haber mandado tocar un instrumento.
Entrada	No se toca el instrumento correcto.
Salida	Mensaje de error por parte del muñeco.
Resultado	Correcto

Tabla CP 11: Presentación academia

CP-12	
Objetivo	Número de peticiones.
Prerrequisitos	El muñeco debe haber pedido tocar 10 veces.
Entrada	Se toca el instrumento correcto por 10ª vez.
Salida	La prueba finaliza.
Resultado	Correcto

Tabla CP 12: Presentación academia

- Pruebas para Tabla UC 6: Academia

CP-13	
Objetivo	Aparece la interfaz correcta.
Prerrequisitos	Se debe haber elegido nivel e instrumentos.
Entrada	El nivel elegido es el 4. Se eligen las claves y el pandero.
Salida	Aparecen los botones con las opciones. El muñeco explica el funcionamiento. Aparece la partitura correspondiente al nivel 4. Aparecen las claves y el pandero.
Resultado	Correcto

Tabla CP 13: Academia

- Pruebas para Tabla UC 7: Escuchar

CP-14	
Objetivo	Comprobar que suena la melodía que aparece en pantalla.
Prerrequisitos	Se debe haber elegido nivel e instrumentos.
Entrada	El nivel elegido es el 4. Se elige el bombo y el triángulo.
Salida	El cursor se empieza a mover y se escucha la interpretación de la partitura con tambor y triángulo.
Resultado	Correcto

Tabla CP 14: Escuchar

CP-15	
Objetivo	Comprobar que suena la melodía que aparece en pantalla acompañada de la canción.
Prerrequisitos	Se debe haber elegido nivel e instrumentos. Se elige la canción de la vaca lechera.
Entrada	El nivel elegido es el 3. Se elige el triángulo.
Salida	El cursor se empieza a mover y se escucha la interpretación de la partitura con el triángulo. Se escucha la canción de la vaca.
Resultado	Correcto

Tabla CP 15: Escuchar

- Pruebas para Tabla UC 8: Tocar

CP-16	
Objetivo	Comprobar que se puede interpretar la partitura.
Prerrequisitos	Se debe haber elegido nivel e instrumentos.
Entrada	El nivel elegido es el 1. Se elige el triángulo.
Salida	El cursor se empieza a mover y se puede pulsar el triángulo para interpretar la partitura.
Resultado	Correcto

Tabla CP 16: Tocar

CP-17	
Objetivo	Comprobar que se puede interpretar la partitura con una canción.
Prerrequisitos	Se debe haber elegido nivel e instrumentos. Se elige la canción de la muñeca.
Entrada	El nivel elegido es el 1. Se eligen las claves.
Salida	El cursor se empieza a mover y se puede pulsar las claves para interpretar la partitura al ritmo de la canción de la muñeca.
Resultado	Correcto

Tabla CP 17: Tocar

CP-18	
Objetivo	Comprobar que se obtiene la puntuación correctamente.
Prerrequisitos	Se debe haber elegido nivel e instrumentos. Se elige la canción de la muñeca.
Entrada	El nivel elegido es el 1. Se eligen las claves.
Salida	Después de haber realizado la interpretación, aparece la puntuación obtenida y el número de estrellas correspondientes.
Resultado	Incorrecto – El número de estrellas que ha aparecido (5) no es el correspondiente a la puntuación obtenida (104/240).
Modificación	Se ha detectado el problema en la clase Juego. No se realizaba la inicialización del MovieClip del contenedorEstrellas. Arreglado.

Tabla CP 18: Tocar

- Pruebas para Tabla UC 9: Modificar velocidad

CP-19	
Objetivo	Comprobar que se puede disminuir la velocidad
Prerrequisitos	Se debe haber elegido nivel e instrumentos.
Entrada	El nivel elegido es el 1. Se eligen las claves. Se pulsa el botón de aumento de velocidad.
Salida	El número que indica la velocidad aumenta.
Resultado	Correcto

Tabla CP 19: Modificar velocidad

CP-20	
Objetivo	Comprobar que se puede aumentar la velocidad.
Prerrequisitos	Se debe haber elegido nivel e instrumentos.
Entrada	El nivel elegido es el 1. Se eligen las claves. Se pulsa el botón de disminuir la velocidad.
Salida	El número que indica la velocidad disminuye.
Resultado	Correcto

Tabla CP 20: Modificar velocidad

- Pruebas para Tabla UC 10: Juego

CP-21	
Objetivo	Comprobar que aparece la interfaz correcta.
Prerrequisitos	Se debe haber elegido nivel, instrumentos y canción.
Entrada	El nivel elegido es el 3. Se elige el charles. Se elige la canción de la gallina.
Salida	Aparecen los botones con las opciones de tocar y escuchar. Aparece la partitura correspondiente al nivel 3. Aparece el charles.
Resultado	Correcto

Tabla CP 21: Juego

Capítulo 7

Conclusiones y posibles ampliaciones

7. Conclusiones y posibles ampliaciones

Tras la realización del trabajo presentado se puede concluir que se han cubierto los objetivos inicialmente planteados. Se ha logrado una herramienta sencilla de manejar y atractiva para niños, que creemos que permitirá mejorar y hacer más divertido el aprendizaje musical del ritmo.

Dada la tecnología empleada la herramienta desarrollada podrá ser utilizada tanto en ordenadores como pizarras electrónicas e incluso dispositivos móviles como tabletas.

Se ha elaborado la documentación necesaria para ayudar a quien quiera modificar o usar esta aplicación pueda hacerlo sin demasiadas dificultades.

El presente proyecto me ha permitido enfrentarme a la realización de una aplicación real, con un cliente. Lo cual, aparte de poder aplicar los conocimientos adquiridos durante la carrera, me ha enseñado los problemas que surgen en la comunicación con el cliente, sobre todo, cuando éste no es un experto en informática.

He aprendido a entender las necesidades del cliente mediante el uso de correos y de reuniones cada 1 o 2 meses para las revisiones del proyecto. Estas revisiones son las que más me han ayudado a dirigir el proyecto en la dirección esperada debatiendo los posibles cambios, mejoras y los problemas que podrían ir surgiendo con estos, como fue el caso de la restricción ya comentada de la pizarra electrónica. Además, dado que el cliente es experto en el área musical, me ayudó a la hora de comprender la parte teórica del ritmo la cual tenía un poco olvidada.

Algunas ideas de ampliación para este proyecto pueden ser:

- Modificar la aplicación para que el profesor que la utilice pueda variar las partituras que aparecen por los distintos niveles. Se conseguiría que la aplicación fuese un poco más dinámica.
- Mejora de la interfaz gráfica del juego. Al estar hecha en Flash, hay multitud de opciones en este campo para que algún diseñador gráfico cree animaciones más complejas e impactantes.
- Modificación para uso con canciones polifónicas. Al no ser las pantallas electrónicas multitáctiles no se ha podido desarrollar la aplicación de forma que las canciones fuesen polifónicas (varios instrumentos pueden tocarse a la vez). Se podría intentar buscar una solución para este problema.
- Añadir nueva funcionalidad: Componer. Se comentó con el cliente que, además de tener la academia y el juego, tener la posibilidad de componer canciones. El niño podría componer su canción para después poder tocarla. Esta parte se

desechó desde un primer momento pues no se esperaba que diese tiempo suficiente a implementarla.

- Guardar resultados. Podría ser interesante guardar los logros y puntuaciones que fuese consiguiendo el niño.

Capítulo 8

Bibliografía y Glosario

8. Bibliografía y Glosario

8.1 Bibliografía

Referencia del lenguaje y componentes ActionScript 3.0 (fecha última consulta: 06/08/2012)

http://livedocs.adobe.com/flash/9.0_es/ActionScriptLangRefV3/package-summary.html

Manual de programación de ActionScript 3.0 (fecha última consulta: 06/08/2012)

http://help.adobe.com/es_ES/ActionScript/3.0_ProgrammingAS3/flash_as3_programming.pdf

Manual Adobe Flash CS3 (fecha última consulta: 06/08/2012)

http://help.adobe.com/archive/es_ES/flash/cs3/flash_cs3_help.pdf

Teoría de la música (fecha última consulta: 06/08/2012)

http://imhaserri.files.wordpress.com/2007/08/curso_de_teoría.pdf

Banco de sonidos (fecha última consulta: 06/08/2012)

<http://www.findsounds.com/>

Sintetizador de voz (fecha última consulta: 06/08/2012)

<http://www.yakitome.com/>

Juegos analizados: (fecha última consulta: 23/04/2012)

<http://www.educa.jcyl.es/zonaalumnos/es/tipologia-recursos/juegos>

<http://childtopia.com/index.php?module=home&func=juegos&idphpx=juegos-educativos-divertidos>

<http://www.vedoque.com/>

<http://www.elpatinete.com/juegos/juegos-de-musica-educativos-para-ninos.html>

<http://www.kongregate.com>

8.2 Glosario de términos y siglas

Ostinato rítmico: Un ostinato es un patrón musical que se repite continua y persistentemente a lo largo de un fragmento o durante toda una pieza musical. Si el patrón que se utiliza es rítmico estaremos ante un ostinato rítmico.

Compás: Es la entidad métrica musical, compuesta por varias unidades de tiempo (como la negra o la corchea). Esta división se representa gráficamente por unas líneas verticales, llamadas «líneas divisorias» o «barras de compás» que se colocan perpendicularmente a las líneas del pentagrama.

Instrumentos básicos: Cuando se habla de instrumentos básicos se refiere a todos los instrumentos de la aplicación sin incluir los de la batería, es decir, las claves, los crótales, el pandero y el triángulo.

REM: (REquirements Management) es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software.

FRQ: Se refiere a los requisitos funcionales. Se siguen las siglas que utiliza el programa REM.

NFR: Se refiere a los requisitos no funcionales.

IRQ: Requisitos de información.

UC: Casos de uso.

DS: Diagramas de secuencia.

CP: Casos de pruebas.

Anexo A

Anexo A.1: Manual de usuario

En el siguiente anexo se describen los pasos que los usuarios deben realizar para manejar la aplicación. Se acompañará con ilustraciones para facilitar la comprensión.

Entrada al juego



Figura 22: Entrada al juego

Nos encontraremos en la escena de introducción. El muñeco nos presentará la aplicación, si queremos saltarnos esta parte pulsaremos la flecha azul que se encuentra abajo a la derecha. Entraremos en el menú principal del juego.

Menú Principal

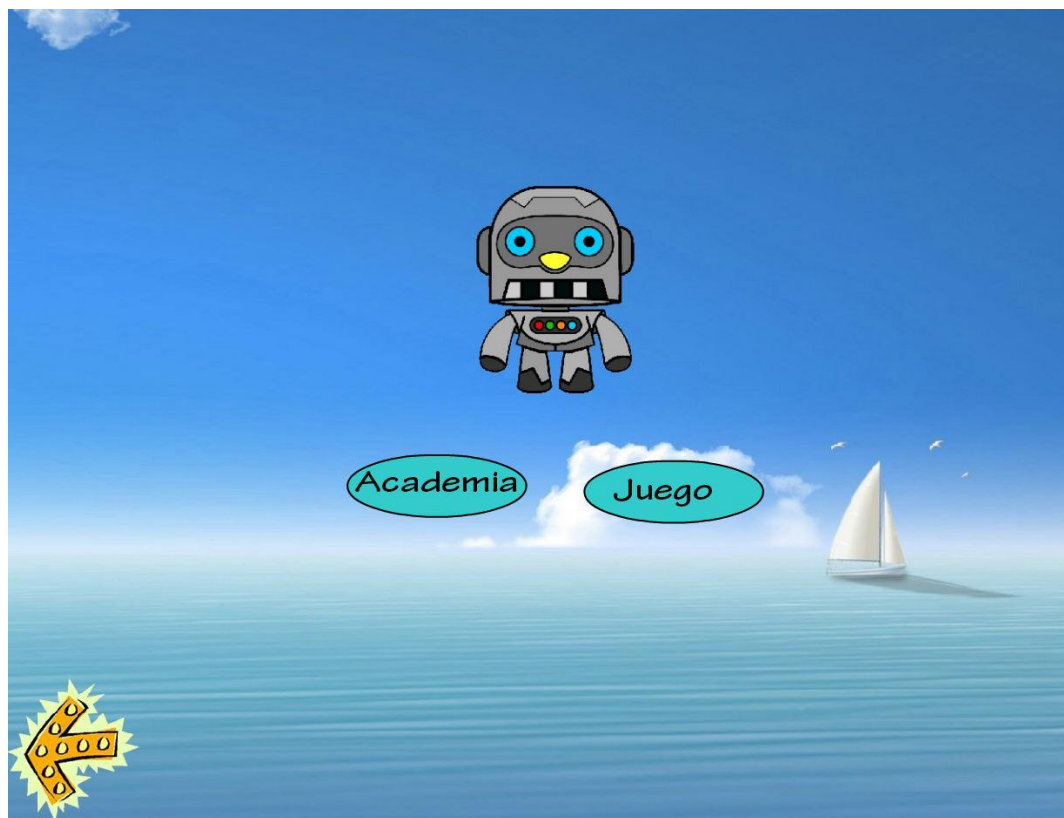


Figura 23: Menú principal

Aquí podremos decidir que parte del juego queremos usar. Si queremos entrar en la Academia pulsaremos el botón de la izquierda con el nombre “Academia”. Si queremos entrar al juego pulsaremos el botón de la derecha con el nombre “Juego”.

También se puede volver a escuchar la introducción pulsando la flecha azul que se encuentra abajo a la izquierda.

Academia

Antes de pasar a la academia tendremos que realizar unas pequeñas pruebas (opcional).



Figura 24: Presentación academia 1

El muñeco nos pedirá que toquemos un instrumento al azar. Nos aparecerá el nombre del instrumento en cuestión arriba en el centro. Cuando acertemos diez veces pasaremos a la segunda parte de las pruebas.



Figura 25: Presentación academia 2

El proceder de esta segunda parte será igual que antes a excepción de que ahora lo deberemos hacer con los instrumentos de la batería.

Cuando acabemos las pruebas pasaremos al menú de niveles de la Academia.

En cualquier momento podremos avanzar directamente a la Academia pulsando la flecha que apunta a la derecha. También podremos retroceder al menú principal pulsando la flecha izquierda.

Menú de niveles

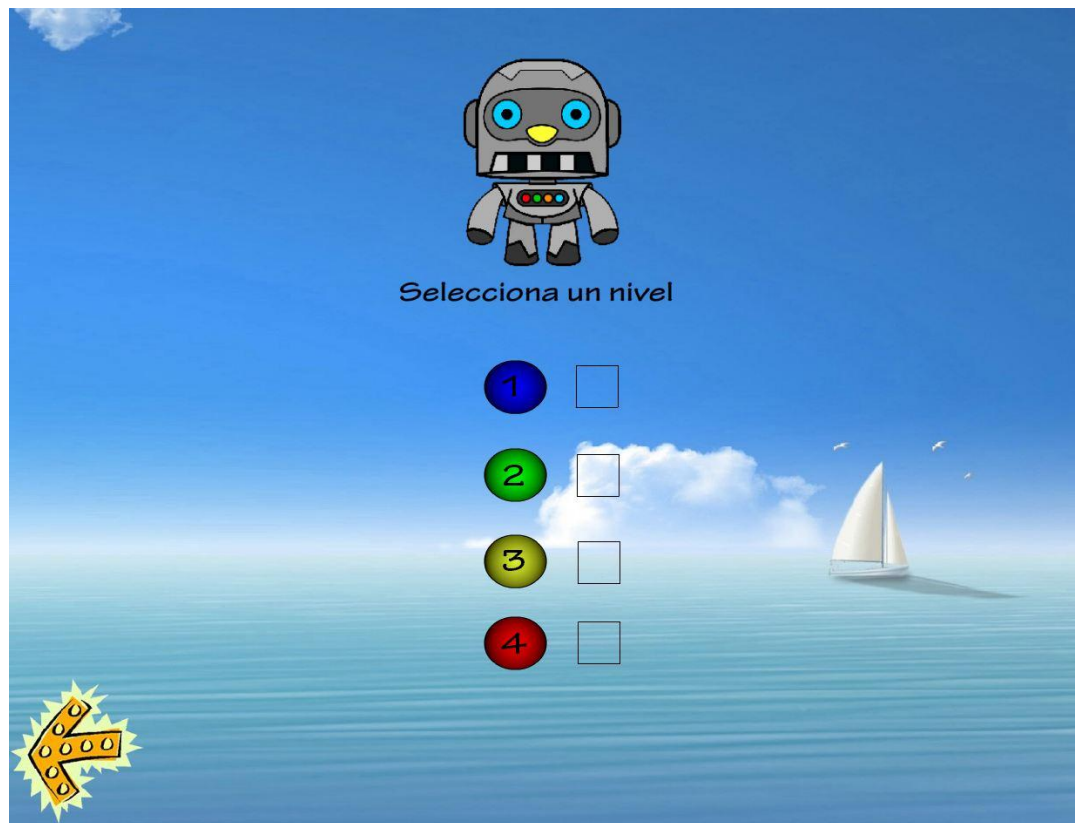


Figura 26: Menú de niveles

Aquí elegiremos el nivel de la Academia que queremos intentar pasar. Los niveles tienen el siguiente contenido:

Nivel 1. 2 ostinatos utilizando solo blancas y negras y solo un instrumento. El tiempo de compas es de 2/4.

Nivel 2. 1 ostinato donde se añadirá el uso de corcheas. Solo un instrumento. El tiempo de compas será 2/4.

Nivel 3. Se permite el uso de 2 instrumentos a la vez. El tiempo de compas es 2/4.

Nivel 4. Se permite el uso de 2 instrumentos a la vez. El tiempo de compas será 3/4.

Si queremos volver al menú principal pulsaremos la flecha que se encuentra abajo a la izquierda. A continuación se muestra un ejemplo seleccionando el nivel 3.

Selección de instrumento/s

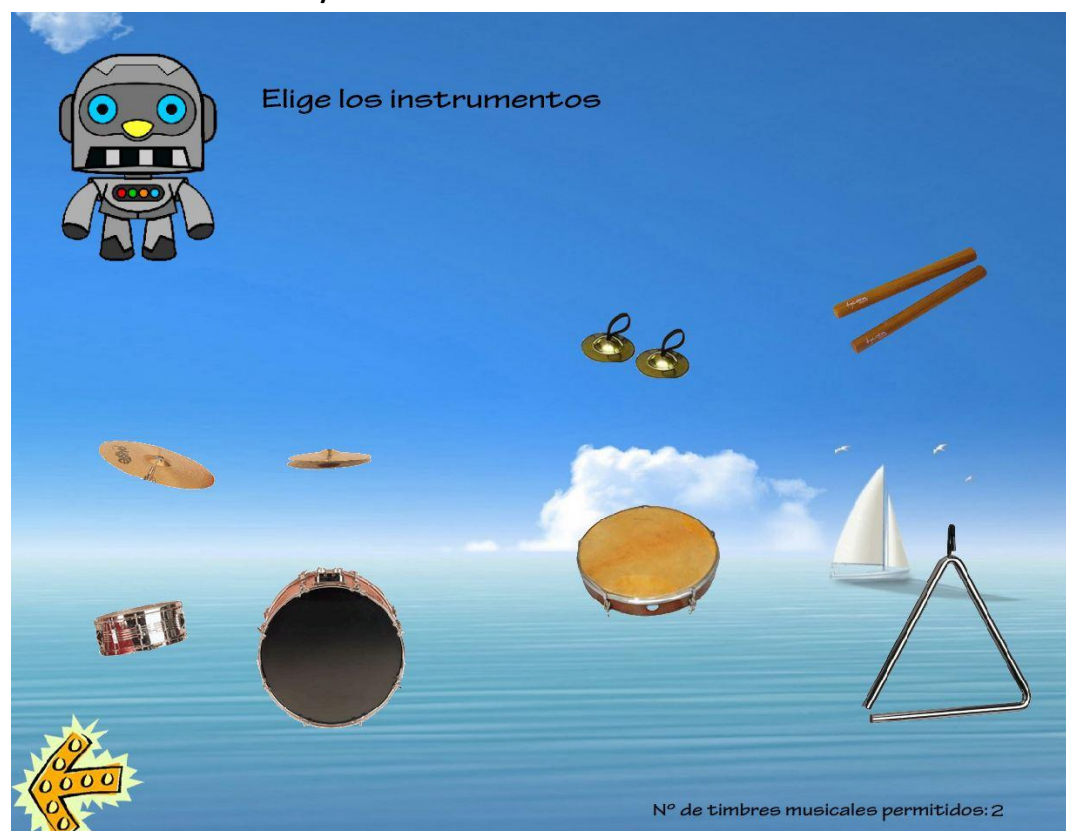


Figura 27: Menú de selección de instrumentos

En este caso, tendremos que seleccionar dos instrumentos. Para seleccionarlos haremos click en los instrumentos que queramos. Si aparece su nombre es que están seleccionados. Si queremos deseleccionarlos pulsaremos otra vez sobre ellos y el nombre desaparecerá. Cuando hayamos seleccionado los instrumentos nos aparecerá un cuadrado azul abajo que deberemos pulsar para pasar a la siguiente escena.

Si queremos volver al menú de niveles pulsaremos la flecha azul que aparece abajo a la izquierda.

Interpretar

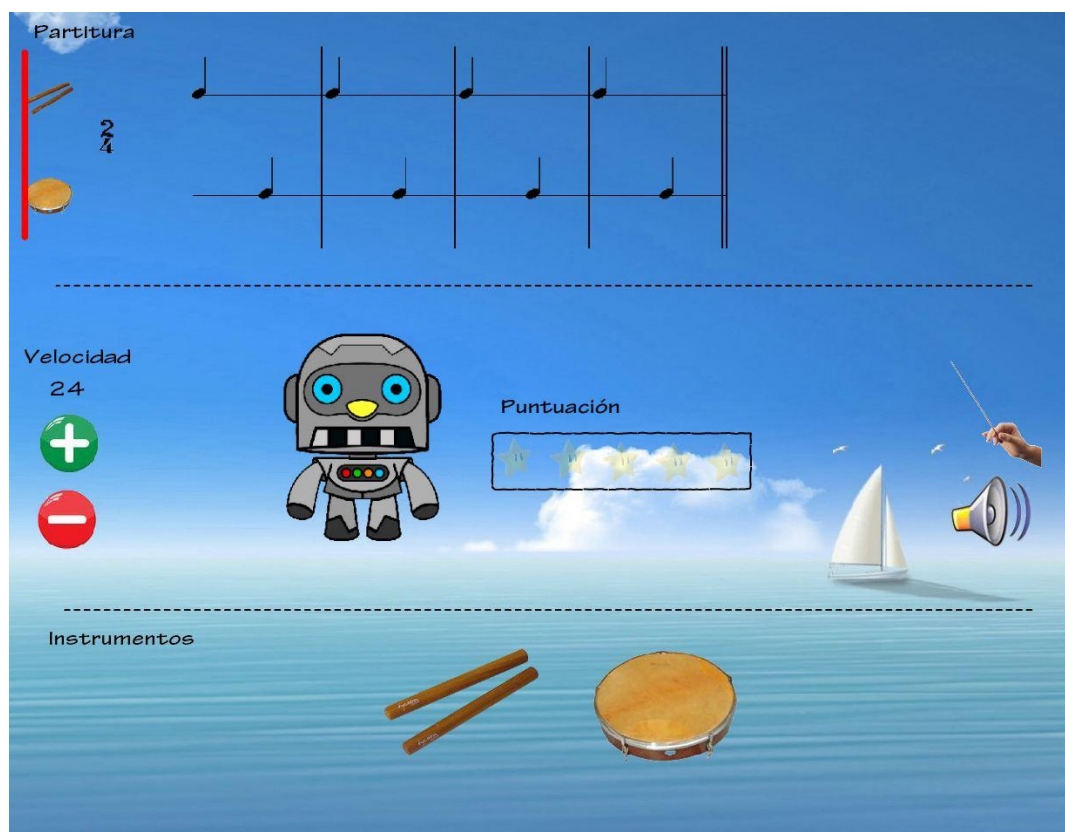


Figura 28: Escena de interpretación

Esta es la interfaz que utilizaremos para interpretar. En la parte de arriba nos aparece la partitura a interpretar. En este caso, al estar utilizando varios instrumentos hay que tener en cuenta que instrumento está asociado a cada una.

En la parte central tenemos un panel con distintas opciones:

A la izquierda vemos dos botones “+” y “-”. Estos nos sirven para cambiar la velocidad a la que se interpreta. Si queremos aumentar la velocidad pulsaremos sobre el botón “+”, si queremos disminuirla pulsaremos sobre “-”. Los cambios se verán reflejados sobre el número que aparece encima de los botones (representa los fps de la aplicación).

A la derecha del muñeco tenemos el cuadro de puntuaciones, dependiendo de cómo de bien interpretemos obtendremos más estrellas o menos. La idea es pulsar el instrumento correspondiente cuando la línea roja pase por encima de las figuras.

Por último, a la derecha nos encontramos con dos botones, uno tiene forma de batuta y otro de altavoz. Si pulsamos el altavoz podremos tener una demostración de la interpretación. Esto nos puede servir, a parte de como una guía, para ajustar la velocidad. Cuando estemos listos, pulsaremos la batuta. El cursor rojo se empezará a mover por encima de la partitura y tendremos que utilizar los instrumentos que elegimos antes para interpretar.

Los instrumentos aparecen abajo del todo de la pantalla.

Si al acabar, no hemos obtenido al menos una estrella tendremos que repetir la interpretación. Si lo hemos hecho bien, dependiendo de en que nivel estemos, habremos superado el nivel. Para indicar que hemos superado determinado nivel, aparecerá una estrella al lado del nivel en el menú niveles como se muestra a continuación.

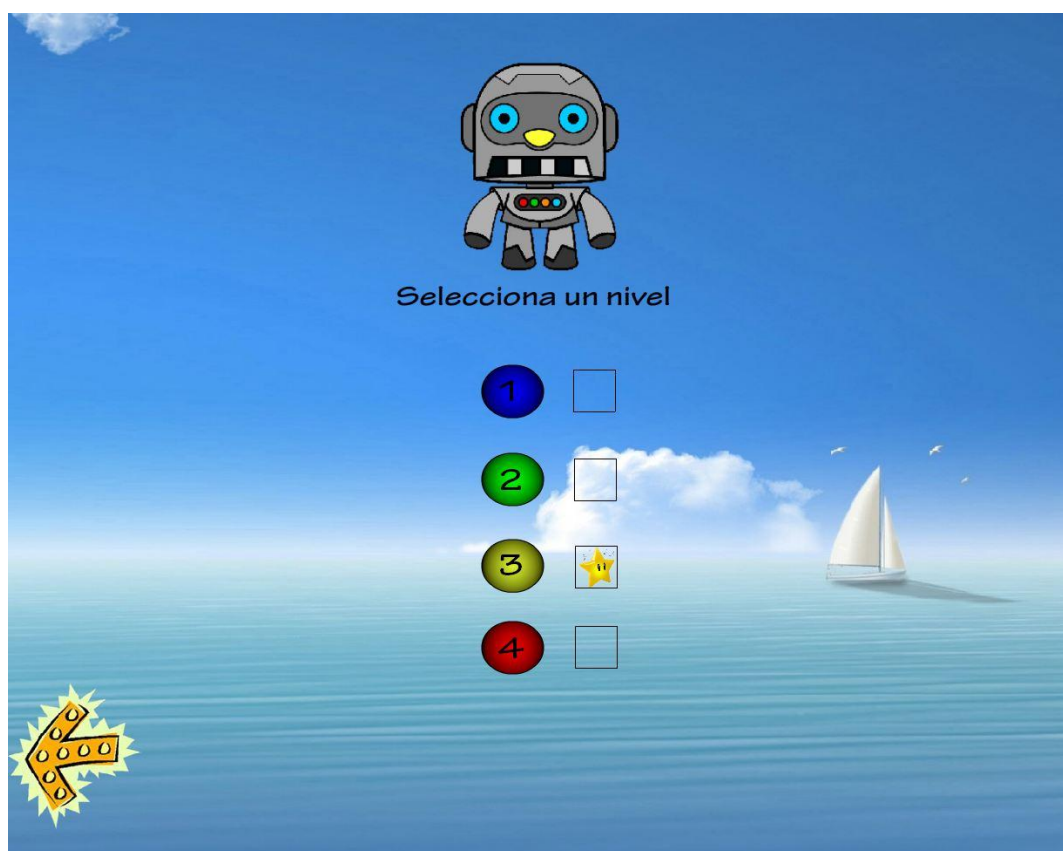


Figura 29: Trofeo

Juego

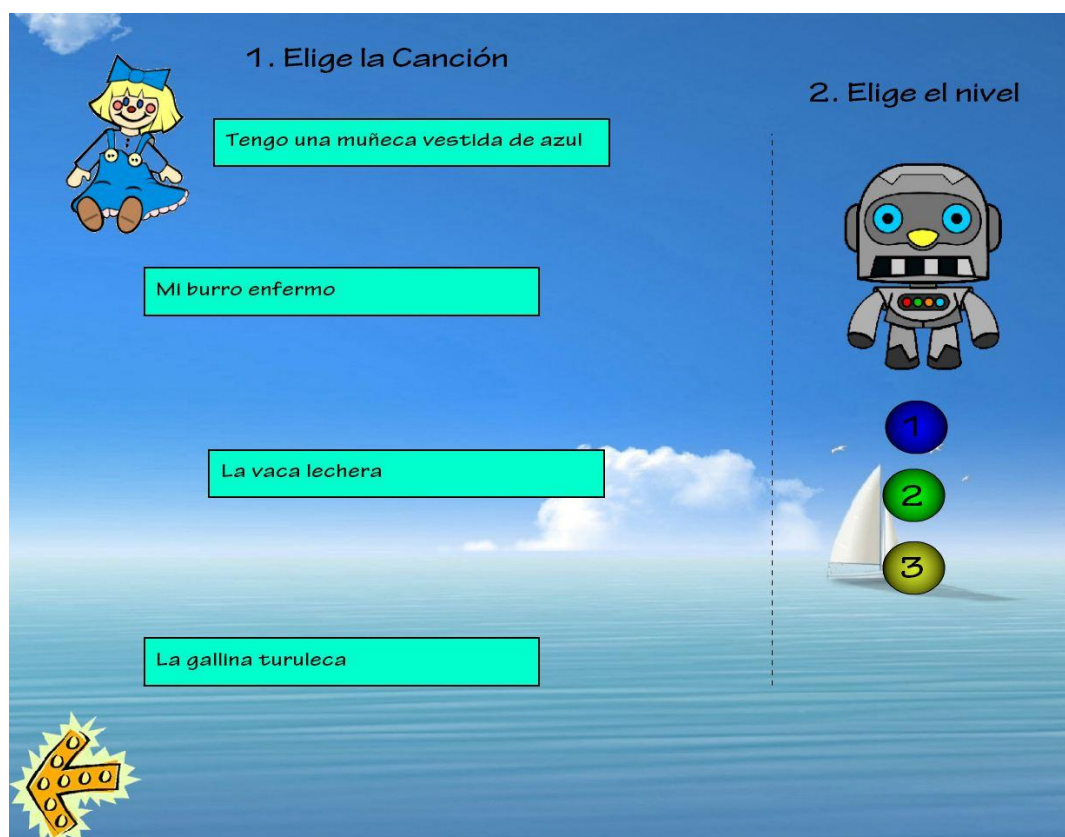


Figura 30: Menú de canciones

Si hemos entrado al juego nos encontraremos con esta pantalla. Primero deberemos elegir la canción que queremos y después el nivel. En este caso vamos a elegir la canción de la muñeca y el nivel 1. De la misma forma que en la academia, tendremos que seleccionar los instrumentos.

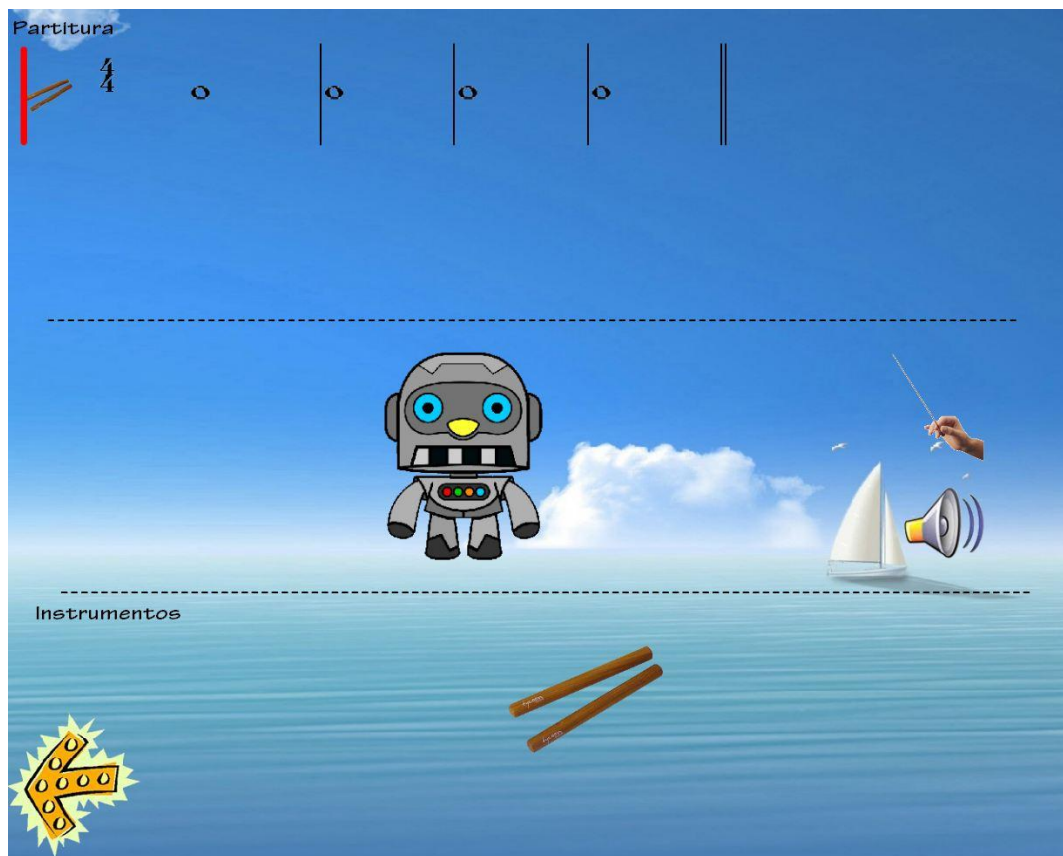


Figura 31: Interpretación de canción

Esta es la pantalla donde tendremos que interpretar. Es similar a la academia pero, en este caso, no disponemos de la modificación de la velocidad por razones obvias. La puntuación aparecerá al acabar en otra pantalla.

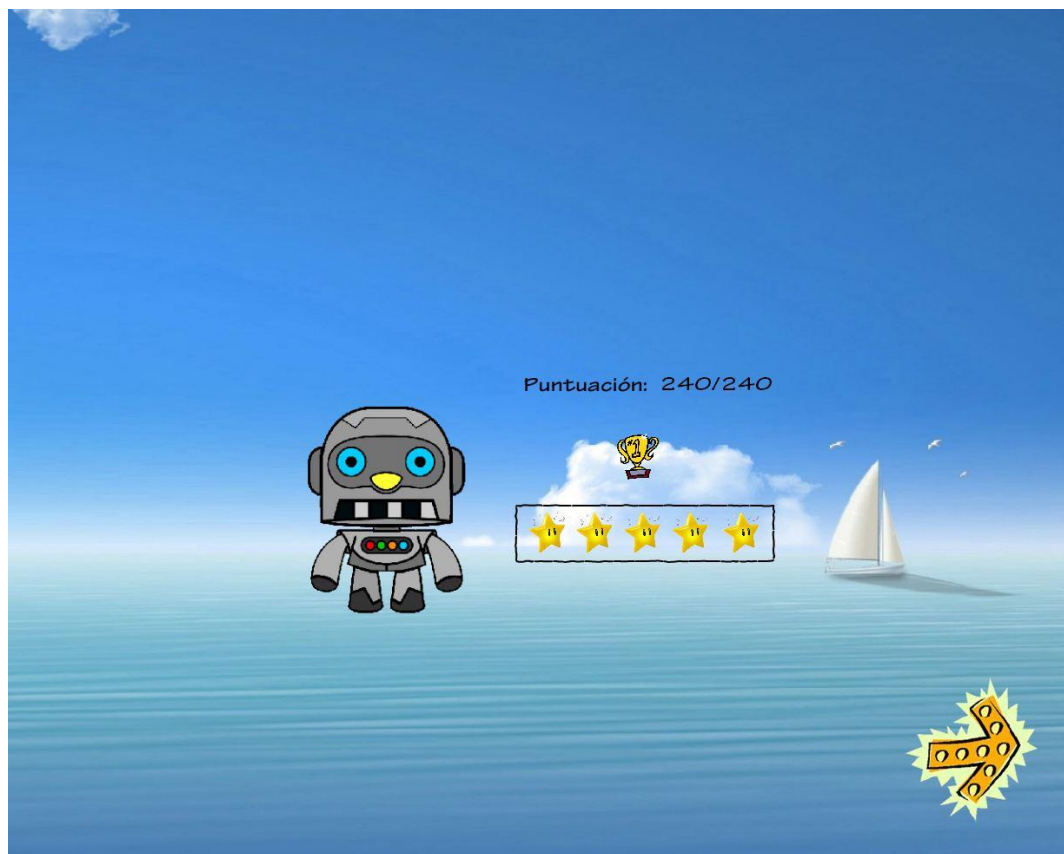


Figura 32: Escena puntuaciones

Si hemos conseguido 5 estrellas nos aparecerá un trofeo. Este trofeo se mantendrá en la pantalla de selección de canción como se muestra en la siguiente ilustración.

La puntuación que aparece arriba depende de como lo hayamos hecho y el número de figuras que tuviese la partitura. Cada figura son 30 puntos. En la Figura 31 vemos que la partitura esta compuesta por 4 figuras, 8 en realidad pues se repite, es por eso que en la Figura 32 podemos obtener como máximo 240 puntos.

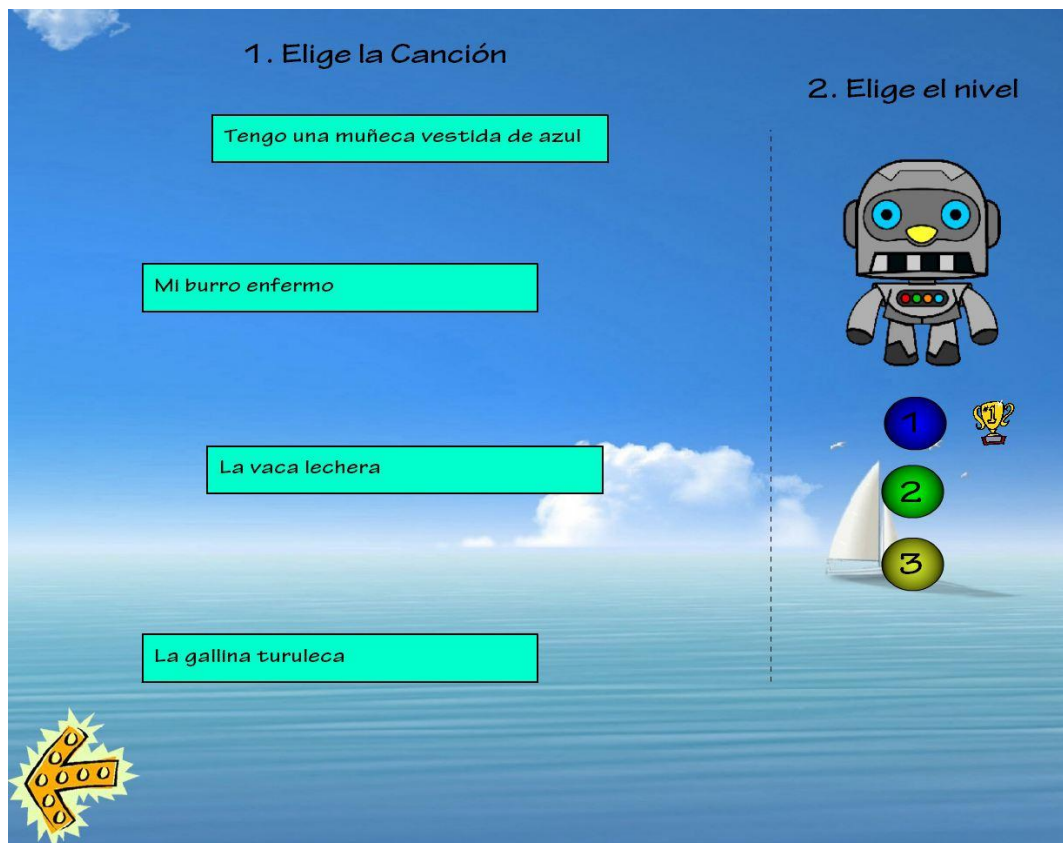


Figura 33: Trofeo canción

Anexo A.2: Manual de instalación

La instalación de los componentes requeridos para hacer funcionar la aplicación es muy sencilla. Como se comentó en la parte donde se habló de Flash, existen plugins para los navegadores web más utilizados que hacen que esta tecnología sea compatible con ellos. Simplemente, deberá dirigirse a la siguiente dirección utilizando su navegador favorito <http://get.adobe.com/es/flashplayer/>. Para ejecutar la aplicación arrastre el archivo *.swf sobre la ventana del navegador.

Anexo A.3: Contenido del CD

- Carpeta documentación: aquí se encuentra la memoria del proyecto junto el fichero de STARUML que contiene los diagramas realizados.
- Carpeta ejecutable: el fichero ejecutable es el que tiene extensión .swf.
- Carpeta fuente: el código fuente se encuentra en los distintos ficheros .as. El fichero .fla es el que abre el proyecto en Adobe Flash.
- Carpeta recursos: se han incluido todas las imágenes y sonidos que se utilizan en la aplicación.