

**MÁSTER UNIVERSITARIO EN
INGENIERÍA EN TECNOLOGÍA
DE TELECOMUNICACIÓN**

TRABAJO FIN DE MÁSTER

**AhoStage: Diseño y desarrollo de un
regidor de escena automático**

Estudiante	<i><Gonzalez, Gorostiaga, Peio></i>
Director/a	<i><Saratxaga, Couceiro, Ibon></i>
Curso académico	<i><2021-2022></i>

<Bilbao, a 14 de agosto de 2022>

Resumen

AhoStage es un software que automatizará todas las acciones que realiza un regidor escénico de una obra de teatro. El software tiene la capacidad de saber en todo momento en que punto del guion se encuentra la obra que se está representando, y en base a ello, lanzará una serie de mensajes para activar los diferentes recursos artísticos (luces, sonido, efectos especiales...) del teatro según se indique en el guion. De esta manera, AhoStage facilitará la labor a los técnicos de las obras de teatro, además de ayudar a reducir gastos a las compañías. Entre los recursos que se pueden controlar mediante el regidor, se encuentra un sistema de doblaje automático, el cual está basado en un software desarrollado anteriormente también por AHOLAB llamado AhoDubber, para que los asistentes a las obras de teatro puedan escucharlas en el idioma que ellos deseen. En este proyecto se trabaja con el estándar MIDI, que será el encargado de comunicar AhoStage con los diferentes recursos artísticos, siendo un lenguaje sencillo pero eficaz.

Palabras clave: MIDI, teatro, regidor automático, software

Laburpena

AhoStage antzezlan baten errejidore eszeniko batek egiten dituen ekintza guztiak automatizatuko dituen softwarea da. Softwareak une oro jakin dezake gidoiaren zein puntutan dagoen antzeztan ari den obra, eta, horretan oinarrituta, mezu batzuk zabalduko ditu antzerkiaren baliabide artistikoak (argiak, soinua, efektu bereziak) aktibatuzko, gidoian adierazten denaren arabera. Horrela, AhoStagek lana erraztuko die antzeztanetako teknikariei, konpainiei gastuak murrizten laguntzeaz gain. Errejidorearen bidez kontrola daitezkeen baliabideen artean bikoizketa automatikoko sistema bat dago. Sistema hori AHOLABek aurretik garatutako AhoDubber izeneko softwarean oinarrituta dago, antzeztanetara joaten direnek nahi duten hizkuntzan entzun ditzaten. Proiektu honetan MIDI estandarrekin lan egiten da, eta horrek komunikatuko du AhoStage baliabide artistikoekin, hizkuntza erraza baina eraginkorra izanik.

Hitz gakoak: MIDI, antzezlan, errejidore automatikoa, software

Abstract

AhoStage is a software that will automate all the actions performed by a stage manager of a play. The software has the ability to know at all times at what point of the script is the play being performed, and based on this, it will launch a series of messages to activate the different artistic resources (lights, sound, special effects ...) of the theater as indicated in the script. In this way, AhoStage will facilitate the work of theater technicians, in addition to helping companies to reduce costs. Among the resources that can be controlled by the ruler is an automatic dubbing system, which is based on a software previously developed also by AHOLAB called AhoDubber, so that theatergoers can listen to the plays in the language of their choice. This project works with the MIDI standard, which will be in charge of communicating AhoStage with the different artistic resources, being a simple but effective language.

Key words: MIDI, theatre, stage manager, software

Índice

Lista de ilustraciones	1
Lista de tablas	3
Lista de acrónimos	4
1. Introducción.....	5
2. Contexto y motivaciones	6
3. Objetivos y alcance del proyecto	8
3.1. Objetivo principal	8
3.2. Objetivos específicos	8
4. Beneficios	10
4.1. Beneficios sociales.....	10
4.2. Beneficios económicos	10
4.3. Beneficios técnicos	11
5. Estado del arte	12
5.1. AhoDubber	12
5.1.1. Sistema de adquisición de AhoDubber	12
5.1.2. Otros aspectos importantes	13
5.2. MIDI.....	15
5.2.1. Historia	15
5.2.2. Conexiones MIDI.....	16
5.2.3. Puertos MIDI.....	16
5.2.4. Mensajes MIDI.....	17
6. Análisis de alternativas.....	21
6.1. Protocolo de conexión	21
6.1.1. Alternativas	21
6.1.2. Criterios de selección	22
6.2. Plataforma de desarrollo	23
6.2.1. Alternativas	23
6.2.2. Criterios de selección	23
6.3. Lenguaje de programación	24
6.3.1. Alternativas	24
6.3.2. Criterios de selección	25
7. Diseño.....	27

7.1.	Requerimientos iniciales	27
7.1.1.	Microfonía	27
7.1.2.	Sistema de adquisición de audio	27
7.1.3.	Integración de AhoStage en el entorno de trabajo de control de escena	28
7.1.4.	Software de control	29
7.1.5.	Análisis detallado del interfaz lógico con las aplicaciones control	29
7.2.	Entorno de operación de AhoStage	29
7.2.1.	Análisis detallado del interfaz lógico con las aplicaciones control	31
7.3.	Módulos de AhoStage	31
7.3.1.	Módulo de adquisición y detección de actividad vocal en tiempo real	32
7.3.2.	Módulo de secuenciación	32
7.3.3.	Interfaz gráfico	33
7.4.	Diseño lógico del interfaz gráfico	33
7.4.1.	Funcionalidades	33
7.4.2.	Funcionamiento lógico	34
7.5.	Diseño gráfico	38
7.5.1.	Ventana principal de control	38
7.5.2.	Diálogo “Cargar guion”	43
7.5.3.	Diálogo “Guardar guion”	44
7.5.4.	Ventana de asignación de canales de entrada	44
7.5.5.	Ventana de asignación de puerto de salida	46
7.5.6.	Ventana de edición de línea	46
7.5.7.	Pestaña de ajustes	48
7.5.8.	Caja de mensajes “?”	49
7.6.	Formato del guion	49
8.	Planificación del proyecto	51
8.1.	PT1: Definición del proyecto	52
8.1.1.	A101: Planificación y metodología	53
8.1.2.	A102: Descripción de especificaciones	53
8.1.3.	A103: Análisis de AhoDubber	53
8.1.4.	A104: Reuniones con Eskena	54
8.2.	PT2: Diseño del interfaz gráfico	54
8.2.1.	A201: Familiarización con Windows Forms	55

8.2.2.	A202: Diseño físico del interfaz	55
8.2.3.	A203: Programación del interfaz	56
8.2.4.	A204: Pruebas.....	56
8.3.	PT3: Actualización del software AhoStage.....	56
8.3.1.	A301: Diseño e implementación de un módulo de generación de señales de secuenciación.....	57
8.3.2.	A302: Módulo para la asignación de canales de entrada	57
8.3.3.	A303: Cambios en el software AhoStage para adaptarlo a los nuevos requerimientos	58
8.4.	PT4: Unificación del módulo AhoStage y del interfaz grafico	58
8.5.	PT5: Pruebas del nuevo sistema en entorno no realista y realista	58
8.5.1.	A501: Periodo de pruebas en un entorno no realista.....	59
8.5.2.	A502: Primer periodo de corrección de errores	59
8.5.3.	A503: Periodo de pruebas en un entorno realista.....	60
8.5.4.	A504: Corrección de errores.....	60
8.6.	PT6: Documentación.....	60
8.6.1.	A601: Redacción de informes.....	61
8.6.2.	A602: Redacción manuales de usuario de la aplicación	61
8.7.	Diagrama de GANTT	62
9.	Presupuesto	63
9.1.	Recursos humanos.....	63
9.2.	Amortizaciones.....	63
9.3.	Gastos	64
9.4.	Otros	65
10.	Conclusiones	66
11.	Anexo I: Manual de usuario	67
	Introducción	70
	Cargado de un guion y reproducción de la obra.....	70
	Paso 1: Apertura de dialogo de selección de guion	70
	Paso 2: Selección de guion	71
	Paso 3: Configuración entrada de audio	72
	Paso 4: Configuración de salida de audio.....	74
	Paso 5: Comenzar la obra	75
	Control de reproducción	76

Botones de avance (>>) y retroceso (<<).....	77
Botón Pause/Play	77
Botón stop	77
Botones Mute In y Mute Out	77
Edición de guion.....	78
Editar línea.....	79
Añadir línea.....	79
Eliminar línea	80
Guarda los cambios en un archivo	81
12. Anexo II: Código	83
Archivos que componen AhoStage.....	83
Main	86
Variables globales	86
Form1	86
Otros formularios.....	87
RtMIDI	87
Take	87

Lista de ilustraciones

Ilustración 1: Eskena	6
Ilustración 2: AHOLAB	6
Ilustración 3: Sistema de adquisición de audio de AhoDubber	13
Ilustración 4: Señal audio VAD.....	13
Ilustración 5: Configuración entrada y salida de audio AhoDubber	14
Ilustración 6: Pantalla principal AhoDubber.....	14
Ilustración 7: Rótulo MIDI	15
Ilustración 8: Cable MIDI	16
Ilustración 9: Byte MIDI.....	17
Ilustración 10: Ejemplo mensaje Note ON	18
Ilustración 11: Byte control Note ON.....	18
Ilustración 12: Bytes información Note ON.....	19
Ilustración 13: Ejemplo mensaje Note OFF	20
Ilustración 14: Diagrama del sistema en una obra de teatro	30
Ilustración 15: Esquema de módulos de AhoStage	31
Ilustración 16: Funcionamiento básico de AhoStage.....	33
Ilustración 17: Diagrama de estados AhoStage	35
Ilustración 18: Pantalla principal de AhoStage	39
Ilustración 19: Menú archivo.....	39
Ilustración 20: Menu configuración	40
Ilustración 21: Menu de filtros	40
Ilustración 22: Tabla de visualización de guion	42
Ilustración 23: Menu de edición	42
Ilustración 24: Botones de control de secuenciación y control.....	43
Ilustración 25: Diálogo para cargar el guion.....	43
Ilustración 26: Diálogo para guardar el guion	44
Ilustración 27: Ventana de configuración de entrada de audio	45
Ilustración 28: Ventana de configuración MIDI	46
Ilustración 29: Ventana de edición de guion	47
Ilustración 30: Ventana para añadir línea al guion	47
Ilustración 31: Ventana para eliminar línea del guion	48
Ilustración 32: Ventana de ajustes.....	49
Ilustración 33: Ejemplo de guion	50
Ilustración 34: Pantalla inicio AhoStage.....	70
Ilustración 35: Botón cargar guion.....	71
Ilustración 36: Elección de guion	71
Ilustración 37: Configuración de entrada de audio	72
Ilustración 38: Asignación de personajes y micrófonos.....	73
Ilustración 39: Configuración de entrada de audio completada	73
Ilustración 40: Botón de configuración de salida de audio	74
Ilustración 41: Formulario de configuración de salida de audio	75

Ilustración 42: Botón de comienzo de obra	76
Ilustración 43: Botones de secuenciación y control	77
Ilustración 44: Menú de edición	78
Ilustración 45: Formulario edición de guion.....	79
Ilustración 46: Formulario añadir línea	80
Ilustración 47: Formulario eliminar línea	81
Ilustración 48: Botón guardar guion	81
Ilustración 49: Guardado de guion	82
Ilustración 50: Archivos de AhoStage	84
Ilustración 51: Esquema de la organización del código de AhoStage	85
Ilustración 52: Main de AhoStage	86

Lista de tablas

Tabla 1: Análisis de protocolos de conexión	22
Tabla 2: Análisis plataforma de desarrollo	24
Tabla 3: Análisis lenguaje de programación	26
Tabla 4: Participantes en el proyecto AhoStage	52
Tabla 5: Datos tarea PT1	52
Tabla 6: Datos subtarea A101.....	53
Tabla 7: Datos subtarea A102.....	53
Tabla 8: Datos subtarea A103.....	54
Tabla 9: Datos subtarea A104.....	54
Tabla 10: Datos tarea PT2	55
Tabla 11: Datos subtarea A201.....	55
Tabla 12: Datos subtarea A202.....	55
Tabla 13: Datos subtarea A203.....	56
Tabla 14: Datos subtarea A204.....	56
Tabla 15: Datos tarea PT3	57
Tabla 16: Datos subtarea A301.....	57
Tabla 17: Datos subtarea A302.....	57
Tabla 18: Datos subtarea A303.....	58
Tabla 19: Datos tarea PT4	58
Tabla 20: Datos tarea PT5	59
Tabla 21: Dato subtarea A501	59
Tabla 22: Datos subtarea A502.....	59
Tabla 23: Datos subtarea A503.....	60
Tabla 24: Datos subtarea A504.....	60
Tabla 25: Datos tarea PT6	61
Tabla 26: Datos subtarea A601.....	61
Tabla 27: Datos subtarea A602.....	61
Tabla 28: Diagrama de GANTT.....	62
Tabla 29: Gasto en recursos humanos del proyecto.....	63
Tabla 30: Amortizaciones del proyecto.....	64
Tabla 31: Gastos del proyecto	65
Tabla 32: Resumen del presupuesto.....	65

Lista de acrónimos

CPU	Central Processing Unit
DIN	Deutsches Institut für Normung
GB	Giga Byte
GPU	Graphics Processing unit
HDD	Hard Drive Disk
HTML	HyperText Markup Language
IDE	Integrate Development Enviroment
MIDI	Musical Instrument Digital Interface
NET	Network
RAM	Random Access Memory
SNR	Signal Noise Ratio
SSD	Solid State Drive
TB	Tera byte
TFM	Trabajo Fin de Master
USB	Universal Serial Bus
VAD	Voice Activity Detection

1. Introducción

Aunque habitualmente permanecen en la sombra y nunca se muestran cara al público, los regidores tienen un papel importantísimo y fundamental en el desarrollo de un espectáculo escénico, ya que un mínimo fallo de ellos desencadenaría en una grave alteración de la obra. Ellos son los encargados de seguir la evolución de la obra, y cuando llega el momento indicado, transmiten a los diferentes técnicos que disparen o activen los diferentes recursos, como pueden ser el encendido o apagado de las diferentes luces o sonido, la activación de los efectos especiales, la apertura o la clausura del telón... Todo esto requiere un seguimiento exhaustivo de la obra y una precisión milimétrica.

Tanto en la representación de una obra como en la gira de una compañía de teatro es necesario el trabajo de un equipo técnico de iluminación, de sonido... y en caso de que la obra necesite recursos extraordinarios como pueden ser los subtítulos sería necesario un técnico extra para que pueda gestionarlo. Estos técnicos deben trabajar en directo con varios sistemas de control independientes, ya que van siguiendo la evolución del espectáculo y van activando diferentes eventos de los diferentes sistemas. Es evidente que, la introducción de nuevos recursos a controlar requiere de personal adicional porque multiplica las acciones de control manual a realizar.

Uno de los posibles recursos escénicos a controlar es el doblaje automático, el cual consiste en lanzar la correspondiente grabación doblada en el momento adecuado. Por lo tanto, al tratarse de un recurso escénico totalmente independiente, necesitaría la gestión de un técnico extra. En relación con el doblaje automático, hace unos años AHOLAB, grupo de investigación que realiza el presente proyecto, realizó un sistema de doblaje automático para las compañías de teatro, pero dicho sistema seguía necesitando la participación de un técnico extra que fuese el encargado de gestionarlo. Es por ello, que el software de AHOLAB no haya sido muy utilizado ya que cuanto mayor cantidad de técnicos sean necesarios en una obra, mayor será el gasto de la compañía.

Con el objetivo de ayudar a los regidores de los espectáculos escénicos, la asociación vasca de teatro Eskena ha propuesto la posibilidad de crear un regidor de espectáculos automático, el cual centralice la gestión de todos los recursos escénicos en un único punto y de esta manera se reduzca el gasto en el equipo técnico. Es en este punto es donde nace el proyecto AhoStage, un software para monitorizar y controlar todos los recursos técnicos de un espectáculo escénico, con el fin de facilitar el trabajo al regidor de la obra. Además, los asistentes a las obras de teatro podrán seguir beneficiándose del doblaje automático, y las compañías podrán seguir representando las obras en diferentes países, aunque la lengua del espectáculo no coincida con el idioma de los asistentes.

En el presente documento se describirá detalladamente la solución propuesta por AHOLAB para el desarrollo del software AhoStage. También se explicará la metodología seguida durante el proyecto y el presupuesto necesario para poder llevarlo a cabo.

2. Contexto y motivaciones

En este capítulo se va especificar cuál era la situación inicial antes de comenzar el proyecto y cuáles han sido las necesidades y motivaciones que ha hecho que este trabajo se acabe llevando a cabo. Por otro lado, se realizará un repaso a las nuevas funcionalidades que se pretenden implementar en AhoStage.

Eskena (Euskadio Ekoizle Eszenikoen Elkarte) es una asociación de teatro que reúne a 23 empresas productoras de teatro de Euskadi. Entre los fines de Eskena se encuentran los siguientes:

- La promoción de Artes Escénicas.
- Impulsar el desarrollo empresarial del sector.
- Posibilitar una relación continuada con las instituciones y demás agentes del medio.
- La presentación, defensa y promoción de los intereses de sus afiliados.



Ilustración 1: Eskena

Eskena desarrolla proyectos tructores para todo el sector empresarial escénico vasco porque tiene los medios y la capacidad de aunar las diferentes visiones y las necesidades del sector privado al que representa.

En esta situación Eskena ha solicitado la necesidad de desarrollar un regidor de teatro automático el cuál pueda facilitar el trabajo a los técnicos durante el transcurso de las obras. Ante esta problemática, el grupo de investigación AHOLAB Signal Processing Laboratory de la UPV/EHU ha decidido aceptar el reto ya que dispone de una muy alta experiencia en todos los campos de las tecnologías del habla (síntesis y reconocimiento del habla, identificación y de locutores, detección y clasificación de audio, conversión y transformación de voces, etc.). AHOLAB tiene un demostrado prestigio internacional especialmente en el campo de la síntesis de voz y de la transformación y la conversión de voz, con muchas publicaciones científicas en estas áreas.



Ilustración 2: AHOLAB

Anteriormente, AHOLAB ya ha trabajado con la compañía Eskena en la creación de un sistema de doblaje automático llamado AhoDubber, el cual era capaz de detectar en todo momento en que punto del guion se encontraba la obra y dependiendo de ello, emitía la correspondiente intervención doblada. Gracias a un sistema de detección de voz (VAD, Voice Activity Deetection) era posible determinar si por un algún micrófono se había detectado habla, y en base a ello el programa decidía si debía lanzar o no la intervención doblada. Pero como se ha mencionado anteriormente, este software seguía necesitando el trabajo de un técnico adicional por lo que las compañías de teatro no han hecho gran uso de él.

Mediante el proyecto KSITek, Eskena y AHOLAB, buscaron el desarrollo de un nuevo software el cual fuese capaz de mejorar los defectos de AhoDubber. KSITek ha sido el proyecto que ha unido a Eskena y AHOLAB, a través del cual Eskena ha podido mostrar sus necesidades y los requerimientos que necesitaría el nuevo proyecto. AHOLAB, por su parte, ha mostrado las diferentes posibilidades que podría ofrecer para dar solución a dicha problemática. Ante la problemática presentada por Eskena, el grupo de investigación AHOLAB pretende resolverla, partiendo de la base del proyecto AhoDubber, se intentará mejorarlo y adaptarlo a las nuevas necesidades que se presentan.

Uno de los puntos más importante de KSITek era la necesidad de crear un regidor automático que sea capaz de seguir la evolución del guion e indicar a los diferentes técnicos (luces, sonido, efectos) los momentos en los que deben ir disparando sus diferentes recursos. De esta manera, se pretende simplificar y automatizar el trabajo del técnico de escena teniendo el control de todos los sistemas en un único software. Además, atendiendo a los requerimientos que necesitaba AHOLAB, AhoStage debe cumplir otro tipo de requerimientos como pueden ser un interfaz gráfico sencillo para el usuario, la posibilidad de edición de guion, la aplicación de filtros, la configuración de los parámetros de entrada de audio...

AhoDubber contaba con un interfaz gráfico muy simple y poco práctico, mediante el cual no se podía tener apenas control sobre la obra. Por ello, Eskena ve indispensable contar con un interfaz gráfico simple, sencillo e intuitivo para que pueda ser usado por cualquier técnico, pero que a su vez reúna todas las necesidades que un técnico precisa para poder tener un control total de la obra.

Otra de las principales funciones que ofrece AhoStage es la fácil creación y edición de guion, de esta manera los técnicos o guionistas podrán trabajar con los guiones directamente desde el programa, sin necesidad de operar con archivos de texto y tener que estar alerta de usar el formato adecuado. AhoStage, desde su interfaz gráfica ofrece la posibilidad de modificar los guiones de una forma sencilla sin tener que preocuparse por el formato, ya que el sistema se encarga por sí mismo de configurar el archivo correctamente.

Otro punto que requirió de un profundo estudio fue la comunicación entre AhoStage y las diferentes aplicaciones de control de los recursos escénicos, la cual, como queda indicado en KSTek, debía ser lo más eficiente posible. Tras analizar las diferentes aplicaciones de control que componen los recursos escénicos, se ha podido observar que todos ellos tienen algo en común: disponen de un interfaz MIDI. Por lo tanto, se ha llegado a la conclusión que la comunicación entre AhoStage y la aplicación de control correspondiente debe ser mediante notas MIDI, dependiendo la nota que envíe AhoStage la aplicación de control realizará una acción u otra. MIDI es un protocolo que permite a ordenadores, instrumentos musicales y hardware comunicarse entre sí. representarán la acción que debe realizar la aplicación de control (luces, sonido, efectos especiales...). De esta manera, cada aplicación escénica diferente será capaz de asociar el valor de una nota MIDI a una acción que debe realizar. Ejemplo de esto pueden ser las luces, ya que se puede relacionar el valor de una nota (como puede ser la 27) al apagado de las ellas.

3. Objetivos y alcance del proyecto

En este apartado se pretende explicar de una forma breve, clara y concisa cuáles son los objetivos de este proyecto. El proyecto tiene un objetivo principal que no es otro que obtener un prototipo del sistema AhoStage, integrado con una o varias de las aplicaciones típicas de control de escenario. Para lograr este objetivo principal se han planteado otros objetivos secundarios o específicos que sirven de apoyo para conseguir el objetivo principal. Logrando estos objetivos secundarios será estar más logrando el objetivo principal.

3.1. Objetivo principal

- **Obtener un prototipo del sistema AhoStage, integrado con una o varias de las aplicaciones típicas de control de escenario:** se desea lograr un software lo más robusto y perfecto posible, acorde a las pretensiones de los miembros de la asociación de teatro Eskena. AhoStage debe ser un sistema automático que sea capaz de identificar en cada momento el punto del guion en el que se encuentra el espectáculo escuchando las intervenciones de los actores y actrices, y notificarlo a los diferentes sistemas de escena que controlan la música, luces y otros efectos, para que estos puedan actuar según tengan planificado en sus respectivas escaletas. Para poder comunicarse con las aplicaciones de control de escenario que se usan en las obras se usará un estándar de comunicación llamado MIDI. El software debe ser lo más completo posible para cumplir con la mayoría de las acciones que desee realizar un técnico de la obra, dándole la posibilidad de tener el manejo completo de la obra tanto pudiendo avanzar, retroceder o parar la obra cuando el desee como pudiendo modificarla o configurarla de manera rápida y sencilla.

3.2. Objetivos específicos

- **Implementación de un módulo de envío de mensajes MIDI:** uno de los objetivos específicos más importantes es el de implementar la comunicación entre el software AhoStage y las aplicaciones de control de escenario mediante notas MIDI. El programa únicamente deberá detectar cuando hay un avance en el guion para enviar una nota MIDI a la aplicación de control. MIDI se implementará a través de un módulo específico el cual se deberá adaptar a las características del proyecto para poder enviar la nota MIDI como y cuando sea necesaria.
- **Creación de un módulo de secuenciación:** AhoStage contará con un módulo de secuenciación de guion, que será el encargado de ubicar en que punto del guion se

encuentra la obra. El objetivo es crearlo a partir de AhoDubber y reutilizando parte de él, poder crear un módulo el cual recibiendo la señal de entrada de audio sea capaz de comunicarse con el módulo MIDI para indicarle que nota debe emitir. Este módulo debe ser lo más eficiente posible.

- **Diseñar un interfaz gráfico:** uno de los puntos más importantes del proyecto es el diseño y desarrollo de un interfaz gráfico. El software anterior, AhoDubber, carecía de un interfaz gráfico por lo que se hacía muy complicado su manejo. Por lo tanto, el objetivo es diseñar un interfaz para el sistema operativo Windows, el cual sea sencillo e intuitivo de manejar, para que cualquier usuario pueda hacer uso de él. Además, se pretende que mediante este interfaz se pueda acceder a todas las funcionalidades que ofrece el software de una manera rápida, como pueden ser la edición de guion, la aplicación de diferentes filtros de búsqueda, la configuración de entrada y salida de audio...
- **Introducir nuevas funcionalidades para la edición del guion:** por medio de este objetivo específico se espera aplicar todas las nuevas funcionalidades que necesita disponer la asociación Eskena. Entre esas funcionalidades están la edición y guardado de guion, la configuración de entrada y salida de audio cuando el usuario lo desee, la configuración de los parámetros más importantes de la recogida de audio, la posibilidad de realizar búsquedas y filtrados en el guion...
- **Adquirir nuevos conocimientos:** el alumno o alumna adquirirá conocimientos sobre el procesado de señal de voz, la detección de actividad vocal, los sistemas de control de escena y los protocolos de comunicación entre los mismos. Igualmente, se intentará lograr nuevos conocimientos respecto al uso del estándar MIDI y a la programación en C++.

4. Beneficios

Este trabajo traerá múltiples beneficios en diferentes ámbitos, de esta manera, en esta sección se van a exponer los más importantes. Los ámbitos más destacados son los beneficios sociales, los beneficios económicos y los beneficios técnicos.

4.1. Beneficios sociales

Como ya se ha mencionado anteriormente este proyecto tiene como punto de partida el software AhoDubber, el cual consistía en un reproductor de tomas de doblaje automático de las obras de teatro. AhoStage es una evolución del mismo, pero logrando mejoras y consiguiendo que sea más eficiente. Por lo que los beneficios sociales siguen siendo los mismos, gracias a este software las compañías de teatro podrán actuar en diferentes países representando la obra en el idioma original, aunque ese idioma no sea la lengua principal de ese país, ya que los asistentes a la obra podrán escucharla en el idioma que ellos deseen (siempre y cuando se haya doblado previamente y esté disponible) gracias al doblaje automático. Además, este software posibilita que los amantes del teatro puedan acudir en cualquier país a él, sin ser necesario que la obra se represente en un idioma que ellos entiendan. Se podría dar la situación que una misma sala, cada asistente este escuchando la obra en un idioma diferente. Además, este doblaje podría ser en lengua de signos o subtítulos, con lo que se promociona la inclusión de personas con problemas auditivos. Por todo esto, los beneficios sociales son varios, tanto para los asistentes a las obras de teatro como para las compañías de teatro.

4.2. Beneficios económicos

En lo que a los beneficios económicos respecta, las compañías de teatro son claramente las grandes beneficiadas. Por un lado, gracias al doblaje automático, las compañías pueden representar sus obras en más países y de esta forma pueden aumentar el número de actuaciones anuales y así lograr mayores ingresos. Además, como el doblaje es automático, solo se necesitará doblar la obra una única vez, y no en cada actuación, y de esta manera las compañías pueden reducir personal de doblaje.

Por otro lado, y el punto más importante, el software automatiza la mayoría de las acciones que debe realizar un técnico en las actuaciones, de esta manera logrando un regidor automático. Todas las acciones que anteriormente realizaban los técnicos como el encendido y apagado de luces, el control del sonido o la gestión de los efectos especiales estarán automatizadas, por

lo que las compañías podrán reducir el número de técnicos necesarios que acuden a la obra ya que el técnico únicamente deberá revisar el correcto funcionamiento del software.

4.3. Beneficios técnicos

Son muy variados los beneficios técnicos que ofrece este trabajo. Este proyecto recoge la mayoría de las acciones en un único software, de modo que es mucho más sencillo tener el control de las obras. Gracias a la simplicidad del interfaz gráfico cualquier usuario puede hacer uso del programa sin casi necesidad tener conocimientos previos. En cuanto a la creación y edición de guion, es mucho más sencilla e intuitiva ya que se puede realizar directamente desde el programa.

Por otro lado, esta herramienta facilita la creación de obras escénicas de mayor complejidad y espectacularidad. Gran parte del atractivo de un espectáculo reside en los elementos espectaculares que lo componen: diseño de luz, sonido, audiovisuales, pirotecnia, escenografía.... AhoStage es una herramienta que centralizará los disparos de todos los efectos de un espectáculo, con lo que su control se facilita por parte de un único técnico. Al simplificar el disparo de efectos de un espectáculo posibilitaría a las compañías girar con efectos más complejos.

Por último, al usarse las notas MIDI, todo el sistema escénico se puede beneficiar de ellas, ya que la mayoría de las aplicaciones escénicas disponen de un interfaz MIDI, y será más sencillo interoperar con nuevos sistemas que dispongan de un interfaz MIDI.

5. Estado del arte

Este apartado se va centrar en dos aspectos fundamentales para la correcta elaboración del proyecto. Por un lado, se estudiará el funcionamiento de AhoDubber ya que a partir de él se va comenzar a realizar el nuevo proyecto. Muchas de las características y funcionalidades de AhoDubber pueden ser reutilizadas para AhoStage.

Por otro lado, es de vital importancia conocer el estándar tecnológico MIDI y su funcionamiento en relación a las notas MIDI, los canales, los mensajes de control... Es muy importante entender su funcionamiento ya que mediante conexiones MIDI se van a conectar el software de AhoStage y los demás sistemas de control.

5.1. AhoDubber

AhoDubber es el punto de partida de este proyecto, por lo que es básico entender su funcionamiento y características. Identificar sus debilidades o puntos de mejora también es clave para poder mejorarlas en el presente proyecto. Al ser la base de AhoStage, muchas de sus funcionalidades pueden ser adaptadas y reutilizadas.

AhoDubber utiliza la detección de voz (VAD, Voice Activity Detection) con múltiples micrófonos de entrada y en tiempo real, de forma que pueda detectar quién habla en cada momento. Además, el sistema emplea el guion codificado siguiendo un formato desarrollado ex profeso para situar cada intervención de los actores en el contexto de dicho guion y emitir la correspondiente intervención doblada.

5.1.1. Sistema de adquisición de AhoDubber

En el módulo de entrada se reciben las señales provenientes de los micrófonos de los actores y actrices y se determina quien es el que está hablando. Este módulo utiliza los micrófonos, y un interfaz audio para recibir las señales de audio en el PC, donde serán analizadas.

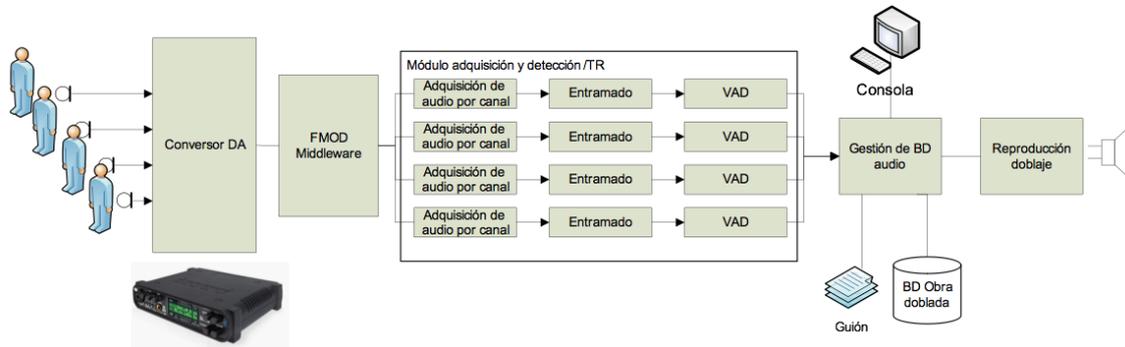


Ilustración 3: Sistema de adquisición de audio de AhoDubber

El sistema se basa en la detección de voz VAD, también conocida como detección de actividad del habla o detección del habla, es la detección de la presencia o ausencia del habla humana, utilizada en el procesamiento del habla.

Se podría pensar que el sistema es capaz de entender que está diciendo cada personaje en cada momento e identificarlo en el guion, pero el VAD únicamente detecta si el personaje está hablando o está en silencio y en base a eso determina si avanzar o no en el guion. En caso de que por un micrófono se detecte voz y ese micrófono corresponda al personaje que en ese momento debe hablar según el guion, el sistema lanza el doblaje automático y avanza a la siguiente línea del guion para volver a esperar el habla en el correspondiente micrófono.

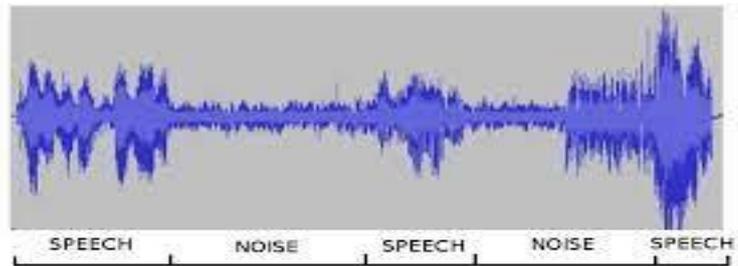


Ilustración 4: Señal audio VAD

5.1.2. Otros aspectos importantes

Uno de los aspectos que se va mejorar en AhoStage es el interfaz gráfico ya que AhoDubber carece de él, se ejecuta en la ventana de comandos y es manejado mediante el teclado. El manejo de AhoDubber es muy simple pero poco intuitivo ya que hay que realizarlo mediante teclado.

Nada más inicializar el programa se deben configurar tanto la entrada de audio como la salida de audio, no siendo posible cambiarlas más adelante durante la ejecución del programa. Todo esto se realiza mediante el teclado.

```
CANALES DE ENTRADA PARA LOS MICROFONOS
- Elige el dispositivo de entrada para los canales 2-3
  1 : Varios micrófonos (Realtek High Definition Audio)
ESC : Salir

El dispositivo de entrada elegido para los canales 2-3 es el numero: 1
-----

CANALES DE SALIDA PARA LA REPRODUCCION DE TOMAS:
- Elige el driver de salida para el canal 0
  1 : Altavoces (Realtek High Definition Audio)
ESC : Salir
```

Ilustración 5: Configuración entrada y salida de audio AhoDubber

Una vez configurada la entrada y la salida de audio, el programa inicializa la pantalla principal la cual tiene el siguiente formato:

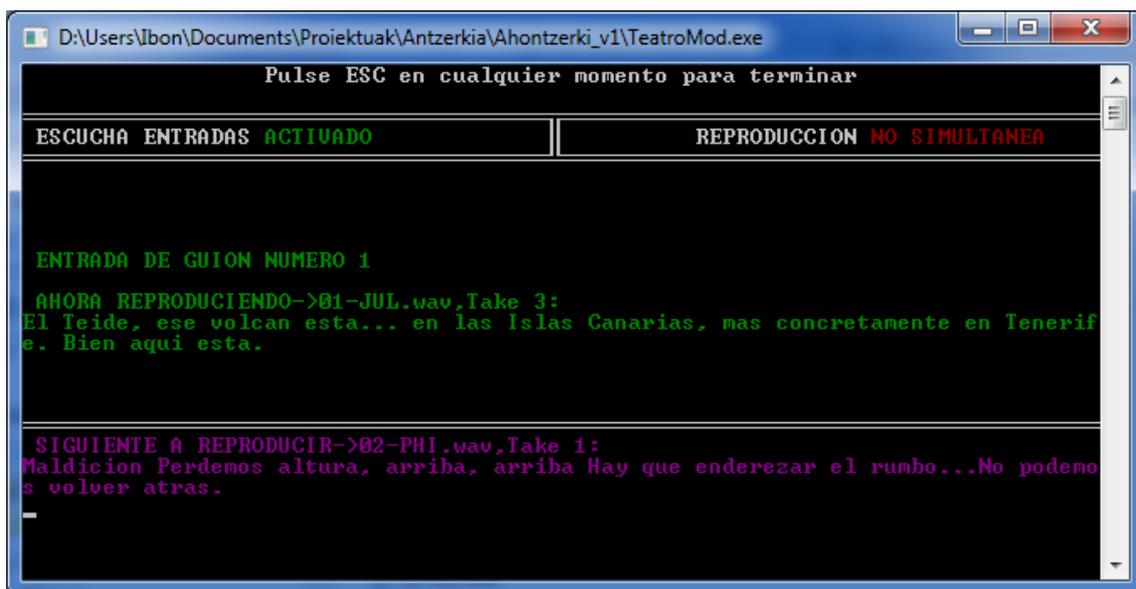


Ilustración 6: Pantalla principal AhoDubber

Las funcionalidades que ofrece el programa son muy limitadas, ya que el usuario no tiene ni la opción de escoger manualmente el guion de la obra que desea reproducir. Estas son las escasas funcionalidades que ofrece:

- Activar o desactivar la reproducción
- Activar o desactivar los canales de entrada de audio
- Avanzar o retroceder en el guion manualmente.

En conclusión, una vez analizado el software AhoDubber, se puede determinar que el módulo VAD se podría reutilizar para AhoStage pero que todo lo demás se deberá rehacer, haciendo especial hincapié en el interfaz gráfico.

5.2. MIDI

MIDI (*Musical Instrument Digital Interface*) es un estándar tecnológico que describe un protocolo, una interfaz digital y conectores que permiten que varios instrumentos musicales electrónicos, ordenadores y otros dispositivos relacionados se conecten y comuniquen entre sí. Una simple conexión MIDI puede transmitir hasta dieciséis canales de información que pueden ser conectados a diferentes dispositivos cada uno.



Ilustración 7: Rótulo MIDI

El sistema MIDI transporta mensajes de eventos que especifican notación musical, tono y velocidad (intensidad); señales de control para parámetros musicales como lo son la dinámica, el vibrato, panoramización en dos dimensiones, cues y señales de reloj que establecen y sincronizan el tempo entre varios dispositivos. Estos mensajes son enviados mediante un cable MIDI a otros dispositivos que controlan la generación de sonidos u otras características.

No obstante, hay que tener claro qué no es MIDI, MIDI no transmite señales de audio. El MIDI son únicamente datos. Un conjunto de instrucciones que las máquinas usan para hablar.

Las ventajas del uso de MIDI incluyen el pequeño tamaño de los mensajes y la fácil manipulación, modificación...

5.2.1. Historia

El MIDI fue desarrollado a principios de los 80 con el fin de estandarizar la comunicación entre las diferentes piezas de hardware musical.

Ikaturu Kakehashi, fundador de Roland, propuso en 1981 la idea de crear un lenguaje instrumental estándar al resto de grandes fabricantes, como Oberheim, Dave Smith Instruments y Moog.

Durante los años siguientes, representantes de todas las grandes compañías de hardware colaboraron para crear, modificar y perfeccionar el MIDI.

El MIDI estándar salió a la luz en 1982. Kakehashi y Dave Smith recibieron en 2013 el Grammy Técnico por su esencial papel en el desarrollo de este protocolo

Todas las compañías involucradas dejaron de lado sus intereses personales por el bien común de los músicos. Hoy en día, el MIDI sigue brillando, y gran parte de la música actual no existiría sin él.

5.2.2. Conexiones MIDI

Para sacar el máximo partido a la configuración MIDI se necesita comprender cada conexión MIDI que la conforma. La mayoría de conexiones MIDI están compuestas por un cable 5-pin DIN. Estos cables de audio son comúnmente conocidos como cables MIDI.

Aunque la mayoría de conexiones MIDI se hacen a través de cables MIDI, algunas configuraciones sólo necesitan cables USB a USB, o USB a MIDI, en función del material empleado o la situación de la interfaz MIDI.



Ilustración 8: Cable MIDI

Para enviar datos MIDI de la configuración al equipo de trabajo, es probable que se necesite una interfaz MIDI. Esta interfaz permitirá enviar y recibir datos MIDI entre el hardware y el equipo de trabajo deseado (u otro secuenciador).

5.2.3. Puertos MIDI

Existen tres tipos diferentes de puertos MIDI:

- **MIDI OUT:** el MIDI OUT transmite datos MIDI de un dispositivo, como un secuenciador o sintetizador, a otra fuente. El MIDI OUT del secuenciador irá al MIDI IN del dispositivo deseado.
- **MIDI IN:** el MIDI IN recibe datos MIDI de otra fuente. Se usará el MIDI IN del equipo para recibir instrucciones del secuenciador u otro dispositivo.
- **MIDI THRU:** el MIDI THRU duplica los datos provenientes del puerto MIDI IN. Esto permite conectar varios dispositivos sin necesidad de varios puertos en el secuenciador o interfaz MIDI.

5.2.4. Mensajes MIDI

El MIDI usa una interfaz serie asíncrona, a una velocidad un poco extraña de 31.250KHz. Cuando se dice que un dato es serie asíncrono se quiere decir que se mandan bit a bit, de uno en uno, y el receptor sincroniza estos datos una vez por cada byte (8 bits).

Las señales que se manejan son binarias, pero cuando se habla de comunicación serie asíncrona no se les llama HIGH y LOW. En su defecto se les llamará MARK cuando sea 1 y SPACE cuando sea 0.

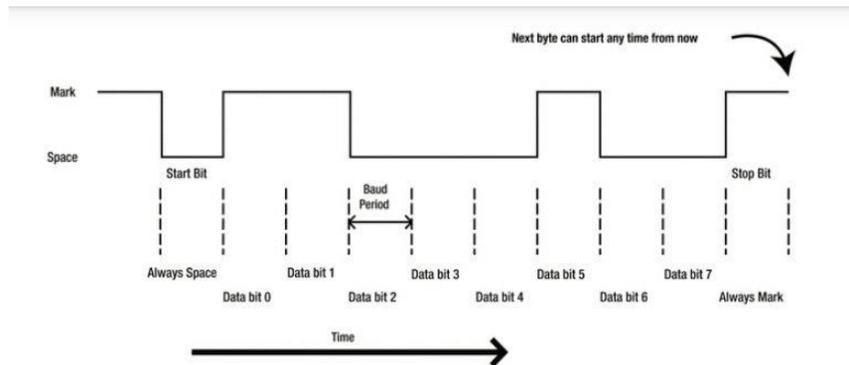


Ilustración 9: Byte MIDI

Cuando no se están enviando mensajes, la

señal permanece en MARK y la aparición de un SPACE marca el comienzo de un byte de datos (Start). El byte termina con la señal de Stop, que es MARK.

En resumen, la estructura de un mensaje MIDI se reduce a: 1 bit SPACE como señal de start, 1 byte con información y un 1 bit MARK como señal de Stop. Para controlar un dispositivo MIDI, lo que se hace es combinar varios de esos bytes de información, y su contenido y la cantidad de ellos dependerá del tipo de mensaje que se esté gestionando. Hay diferentes tipos de mensajes MIDI y cada uno de ellos controla un aspecto del sistema.

Un mensaje MIDI resulta de la combinación de varios bytes de información. El número de bytes que lo compongan depende del tipo de mensaje. Hay varios tipos de mensajes, y cada uno de ellos controla una parte de un sistema musical MIDI.

5.2.4.1. Note ON

Note ON, se utiliza cuando se genera una nota. Este tipo de mensajes está compuesto por 3 bytes. Y un ejemplo podría ser este:

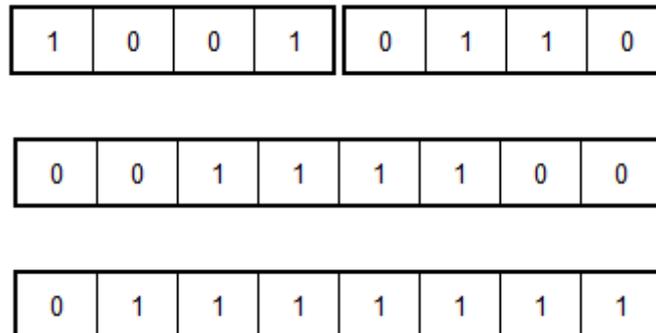


Ilustración 10: Ejemplo mensaje Note ON

El primer byte indica el tipo de mensaje y a qué canal va dirigido. Se le conoce como byte de Status.

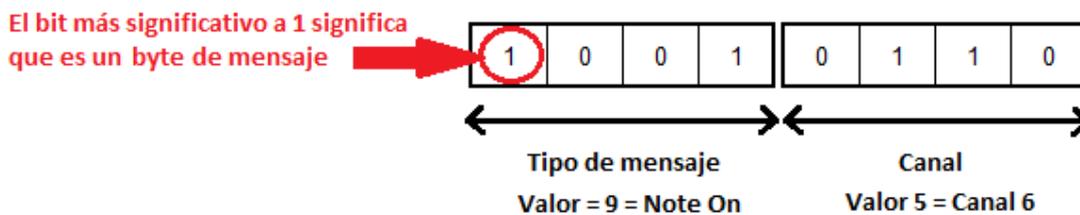


Ilustración 11: Byte controlo Note ON

Los 4 primeros bits indican qué tipo de mensaje es, en este caso Note ON, y el primero de esos bits siempre será 1 para indicar que es un byte de Status.

Normalmente un mensaje MIDI va dirigido a un canal y cada conexión MIDI puede comunicarse por 16 canales MIDI. De esta forma cada dispositivo MIDI puede utilizar uno, varios o incluso todos los canales.

De esta forma se pueden tener varios equipos conectados, y que cada uno de ellos reaccione cuando se le mande un mensaje por el canal que le hayamos seleccionado. Los canales en términos de programación se numeran del 0 al 15.

Como se ha dicho, los mensajes Note ON están compuestos de 3 bytes, y los otros dos que quedan contienen los datos de ese mensaje. Se les llama bytes de datos.



Ilustración 12: Bytes información Note ON

El primero de ellos contiene la información de la nota y el segundo la velocidad, que es la fuerza con la que se ha tocado. Como el primer bit tiene que estar a 0, el byte puede tener 128 valores que van del 0 al 127.

En el caso de la velocidad, cuanto mayor sea, más fuerte se habrá tocado la nota, siendo 127 el máximo. Normalmente la velocidad se traduce en el volumen de la nota.

5.2.4.2. Note OFF

Todo mensaje de Note ON debería estar acompañado en algún momento de un Note OFF, que indica que se deja de tocar una nota. Hay dos maneras indicar que se deja de tocar una nota:

- Mandar un mensaje Note OFF.
- Mandar un mensaje Note ON, indicando en el primer byte de datos la nota que queremos liberar y en la velocidad el valor 0.

Un mensaje Note OFF es muy parecido al Note ON, pero poniendo en el tipo de mensaje del byte de Status un 8. El primer byte de datos indica que nota hemos dejado de pulsar, y el segundo la velocidad lo rápido que se ha liberado la nota.

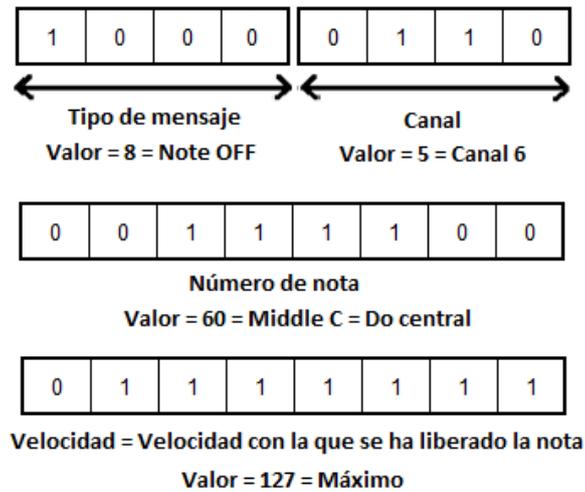


Ilustración 13: Ejemplo mensaje Note OFF

En realidad, la mayoría de dispositivos no tienen en cuenta la velocidad de los mensajes Note OFF. Por eso la mayoría de las veces se utiliza el mensaje Note ON con velocidad 0, y que entienden absolutamente todos los dispositivos.

6. Análisis de alternativas

En este apartado se van a escoger entre los diferentes recursos necesarios para la elaboración y desarrollo del trabajo, analizando las distintas alternativas de las que se disponen. Cierto es que AhoStage es un proyecto profundamente definido desde el comienzo ya que los miembros de Eskena detallaron perfectamente las características y necesidades del proyecto, por lo que a la hora de realizar el trabajo fueron pocas las decisiones que se tuvieron que tomar. No obstante, a pesar de ser pocas, se tuvieron que tomar decisiones muy importantes que pueden marcar el desarrollo del proyecto. Se van a describir las diferentes decisiones que se tuvieron que tomar, que alternativas había para cada una de ellas, el criterio de elección para cada una de las decisiones y cuál ha sido alternativa escogida finalmente.

6.1. Protocolo de conexión

Uno de los puntos más importantes y con más relevancia en el proyecto era como se iba a realizar la conexión entre AhoSatge y los recursos escénicos o aplicaciones de control de los teatros. Para tomar una decisión correcta, se analizaron una serie de protocolos que podían realizar esa función. Las opciones han sido varias, pero a continuación se van enumerar las más relevantes.

6.1.1. Alternativas

- MIDI: MIDI es la abreviatura de “Musical Instruments Digital Interface”, la cual básicamente es un protocolo de comunicación de datos que les permite a PC, sean Windows, Mac, Linux o cualquier otra plataforma, sintetizadores, controladores y muchos otros dispositivos e instrumentos desarrollados para la creación musical comunicarse sin problemas y en forma estándar.
- Ethernet: un estándar de red de comunicación capaz de manejar grandes cantidades de datos a velocidades de 10 Mbps o 100 Mbps, y hasta 1500 bytes por paquete. La especificación utiliza un protocolo abierto en la capa de aplicación. Se utiliza especialmente para aplicaciones de control. Ofrece una serie de ventajas, este tipo de red es fácil de configurar, operar, mantener y ampliar. Esta tecnología se utiliza con ordenadores personales, mainframes, robots, dispositivos y adaptadores de entrada/salida (E/S), controladores lógicos programables (PLC) y otros dispositivos
- USB (Universal Serial Bus): define un conjunto de paquetes (con sus formatos) y procedimientos para controlar y administrar el acceso al bus USB de los periféricos y
-

para permitir el entendimiento entre las dos partes en la comunicación. Es manejado por el controlador USB

6.1.2. Criterios de selección

Una vez expuestas las dos alternativas posibles, se van a mencionar los criterios de elección con el valor de importancia de cada uno representado en porcentaje:

- Facilidad de uso (40%): para realizar la conexión entre los dispositivos se debe diseñar un algoritmo para que sean capaces de entenderse entre ellos. Es por ello, que se necesita un protocolo fácil de entender, y, sobre todo, fácil de configurar.
- Eficiencia (40%): la conexión entre los dispositivos debe ser muy simple, e intentar que transportando la menor de datos posibles se envíe la mayor de información posible.
- Fiabilidad (5%): debe ser un protocolo que ofrezca una alta disponibilidad, y un porcentaje muy reducido de error ya que cualquier dato erróneo puede desencadenar en un fallo fatal en la obra.
- Alcance (5%): aunque pueda causar una impresión contraria, los recursos escénicos pueden encontrarse relativamente de AhoStage por lo que las conexiones deben ser relativamente largas. Es por ello, que el protocolo elegido debe tener una tolerancia de error relativamente alta a la distancia

Atendiendo a estos criterios se van a calificar las tres alternativas planteadas con una nota del 0 al 10, y la alternativa que consiga la nota más alta será la solución para el problema. Para conseguir la nota final se ponderarán por su porcentaje cada una de las notas parciales.

	Facilidad de uso (40%)	Eficiencia (40%)	Fiabilidad (5%)	Alcance (5%)	Resultado final
MIDI	9	9	7	8	8
Ethernet	5	7	10	10	5,8
USB	6	7	8	7	6

Tabla 1: Análisis de protocolos de conexión

Como se puede ver en la tabla anterior, una vez finalizado el análisis, la mejor opción es usar el estándar MIDI, ya que las aplicaciones de control y los recursos escénicos de los teatros cuentan con interfaz MIDI, y en el mundo musical está muy extendido. Además, su utilidad es muy sencilla y se puede adaptar fácilmente a las necesidades del proyecto excelentemente.

6.2. Plataforma de desarrollo

Una de las especificaciones para el proyecto era que debía ser ejecutable en el sistema operativo Windows. Partiendo de esa base, antes de comenzar el proyecto se barajaron varias plataformas de programación para llevar a cabo el trabajo. Otro punto importante del proyecto es que se debía diseñar un interfaz gráfico sencillo, por lo que la plataforma de desarrollo debe posibilitar esto.

6.2.1. Alternativas

- Netbeans: NetBeans IDE es un entorno de desarrollo integrado de código abierto y gratuito para el desarrollo de aplicaciones en los sistemas operativos Windows, Mac, Linux y Solaris. El IDE simplifica el desarrollo de aplicaciones web, empresariales, de escritorio y móviles que utilizan las plataformas Java y HTML5.
- VisualStudio: es un panel de inicio creativo que se puede usar para editar, depurar y compilar código. El editor que posee el programa es bastante inteligente; puede detectar errores y sugerir código según sea necesario. El programa incluye un diseñador para Windows Forms y formularios web el cual permite arrastrar y soltar de elementos de interfaz de usuario.
- Eclipse: es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma.

6.2.2. Criterios de selección

A continuación, se van a detallar los criterios con los que se tomará la decisión, dándoles a cada uno de ellos una relevancia diferente:

- Interfaz gráfica (60%): uno de los puntos más importantes del trabajo es la creación de un interfaz gráfico. Por ello, es extremadamente importante trabajar en una plataforma la cual permita diseñar el interfaz de una manera rápida e intuitiva.
- Conocimientos previos (10%): haber trabajado previamente con la plataforma puede ayudar durante el proyecto.
- Sencillez (30%): es importante que la plataforma sea fácil de utilizar y de aprender nuevos conceptos.

Atendiendo a estos criterios se van a calificar las tres alternativas planteadas con una nota del 0 al 10, y la alternativa que consiga la nota más alta será la solución para el problema. Para conseguir la nota final se ponderarán por su porcentaje cada una de las notas parciales.

	Interfaz gráfico (60%)	Conocimientos previos (10%)	Sencillez (30%)	Resultado final
Netbeans	6	6	7	6,3
Visual Studio	9	3	8	8,1
Eclipse	6	7	8	6,7

Tabla 2: Análisis plataforma de desarrollo

En definitiva, la mejor alternativa de las tres es usar la plataforma de desarrollo VisualStudio ya que ofrece unas muy buenas características para diseñar y desarrollar el interfaz gráfico. Ciertamente que no se tienen conocimientos previos de esta plataforma, pero es extremadamente sencilla de utilizar y muy intuitiva.

6.3. Lenguaje de programación

La correcta elección del lenguaje en el cual se va a desarrollar el software es muy importante, ya que una vez escogido el lenguaje de programación y comenzado el proyecto, es muy complicado cambiarlo durante el transcurso del mismo. Hay que mencionar, que AhoStage tiene como punto de partida el software AhoDubber el cual está desarrollado en C++, y del cual se pueden reutilizar bastantes partes. Como se ha analizado en el apartado anterior, la plataforma de desarrollo será Visual Studio, por lo que este puede ser otro factor determinante a la hora de la elección.

6.3.1. Alternativas

- **C++:** C++ es un ejemplo de lenguaje de programación compilado, multiparadigma, principalmente de tipo imperativo (se conoce en cada paso del programa su estado, determinado por el valor que toman las variables utilizadas) y orientado a objetos, incluyendo también programación genérica y funcional. Entre sus características principales están el control de excepciones, biblioteca estándar (suele venir con el compilador), compatibilidad con C, uso de punteros, es portátil (tiene un gran número de compiladores en diferentes plataformas y sistemas operativos), eficiencia con el hardware, tiene estándar ISO... Otros puntos importantes son que ya se tienen conocimientos previos del lenguaje y que AhoDubber está también programado en C++.

- C# (C Sharp): es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET. C# tiene sus raíces en la familia de lenguajes C. C# ha sido desarrollado por Microsoft con el objetivo de dar la posibilidad de crear una multitud de aplicaciones ejecutadas en .NET Framework. C# es un lenguaje sencillo, fácil integración con otros lenguajes y multihilo.
- Java: es el lenguaje de programación y la plataforma de desarrollo número uno. Reduce costos, acorta los plazos de desarrollo, impulsa la innovación y mejora los servicios de las aplicaciones. Es un lenguaje simple y robusto. Se tiene conocimientos previos de este lenguaje.

6.3.2. Criterios de selección

Una vez explicadas las soluciones posibles, se va proceder a detallar cuales van a ser los criterios de evaluación para escoger la mejor solución para el problema. Cada criterio tendrá un peso diferente a la hora de escoger la mejor solución.

- Conocimientos previos (25%): este criterio mide los conocimientos previos que tienen los diferentes miembros del equipo sobre el lenguaje de programación, ya sea porque han estudiado o han trabajado con él previamente. Tener unos buenos conocimientos previos ayuda a no tener que realizar un esfuerzo previo para aprender el lenguaje y a que durante el desarrollo del proyecto todo sea más sencillo.
- Sencillez (15%): que sea un lenguaje muy sencillo de aprender como de programar es vital. El uso de las bibliotecas es muy importante por lo que disponer de ellas es clave. Además, disponer de mucha información en internet es relevante, ya que durante el transcurso del proyecto será necesario usarla.
- Robustez (15%): un sistema robusto es un sistema fuerte, sin debilidades ni vacíos de seguridad. La robustez en un programa informático hace referencia a su capacidad para hacer frente a errores mientras se está ejecutando
- AhoDubber (30%): este criterio mida la capacidad de adaptar el sistema AhoDubber al nuevo sistema AhoStage. Hay que tener en cuenta que AhoDubber está escrito en C++. Esto podría ayudar y acelerar la programación, ya que habría partes que podrían ser reutilizables.
- Eficiencia (15%): es el conjunto de fases de un compilador que transforman un fragmento de código en otro fragmento con un comportamiento equivalente y que se ejecuta de forma más eficiente, es decir, usando menos recursos de cálculo como memoria o tiempo de ejecución.

Atendiendo a estos criterios se van a calificar las tres alternativas planteadas con una nota del 0 al 10, y la alternativa que consiga la nota más alta será la solución para el problema. Para conseguir la nota final se ponderarán por su porcentaje cada una de las notas parciales.

	Conocimientos previos (25%)	Sencillez (15%)	Robustez (15%)	AhoDubber (30%)	Eficiencia (15%)	Resultado final
C++	8	7	5	10	5	7,6
C#	4	7	5	7	5	5,7
Java	7	8	8	2	8	6

Tabla 3: Análisis lenguaje de programación

Atendiendo a la tabla anterior, el mejor lenguaje de programación para este caso es C++, principalmente porque AhoDubber estaba programado en este lenguaje, además de tener unos buenos conocimientos previos. Esta elección hace que la robustez y eficiencia del programa queden mermadas, por lo que se deberán buscar otros métodos para compensarlas.

7. Diseño

Durante este apartado se expondrá la solución propuesta para la problemática surgida. La exposición se hará de forma progresiva y ordenada, de manera que en primer lugar se explicarán temas más sencillos y superficiales para finalmente dar una explicación exhaustiva y detallada.

Primeramente, se abordarán los aspectos generales de equipamiento necesario para la plataforma de ejecución, integración con el resto de aplicaciones de los técnicos de control e interfaces de comunicación entre los diferentes elementos.

Posteriormente, se describirá el software esquemáticamente, mediante diagramas, de esta forma será más sencillo conocer las diferentes funcionalidades que tiene el programa y todas las capacidades que tiene.

Finalmente, se detallará como está compuesto pantalla a pantalla (también conocidos como formularios), es decir, cada formulario que funcionalidad tiene y como poder moverse entre ellas. Los formularios tienen diferentes botones, menús, submenús, tablas, *textboxes*, *comboxs*... por lo que se definirá la funcionalidad de cada uno de ellos.

7.1. Requerimientos iniciales

7.1.1. Microfonía

La microfonía a emplear debe asegurar que se capte el máximo de la voz del actor que la lleva y el mínimo del sonido circundante (de otros actores o de fondo). Se recomienda tecnología inalámbrica y la capsula o micrófono deberá ser del tipo “diadema” con diagrama polar omnidireccional, (facilitar el movimiento sin pérdida de sensibilidad) sensibilidad media para evitar capturar otros sonidos y una calidad alta en cuanto a su respuesta en frecuencias.

7.1.2. Sistema de adquisición de audio

El equipamiento de adquisición de audio debe ser capaz de recibir la señal de los micrófonos (o de las petacas de recepción en el caso de sistemas inalámbricos), preamplificarla, realizar la conversión A/D y transmitirla al ordenador independientemente para cada canal de entrada. La conexión con el PC se puede hacer usando diferentes sistemas, siempre que cada canal (micrófono) aparezca como un dispositivo de audio de entrada de cara al sistema operativo (Windows).

7.1.3. Integración de AhoStage en el entorno de trabajo de control de escena

En este apartado se definen los requerimientos para que AhoStage pueda funcionar de forma integrada con el resto de equipamiento y software que forman parte del entorno de trabajo de los técnicos de control de escenario.

7.1.3.1. Equipo de ejecución AhoStage

El software de AhoStage correrá en Windows. Para permitir la conexión del interfaz de adquisición de audio el PC debe contar con conectores USB (o Firewire). No hay requerimientos muy exigentes respecto a potencia de cálculo del hardware: un procesador de capacidad media-alta y memoria suficiente para evitar bloqueos temporales...

7.1.3.2. Conexión para el envío de mensajes de disparo

Según los diferentes análisis que se ha realizado, los mensajes de disparo (también se denominarán señalización externa) consistirán en *notas* MIDI que se enviarán desde el PC en el que se ejecuta AhoStage a través de un interfaz MIDI. A esta interfaz MIDI se conectarán en cascada los diferentes sistemas de control de luces, audio, efectos, etc.

Con respecto a la implementación física de este cableado, el escenario de trabajo será el siguiente:

- AhoStage se ejecutará en un ordenador diferente del resto de softwares de control.
 - Este ordenador podrá utilizar un interfaz MIDI independiente conectado al PC AhoStage vía USB.
 - También podrá utilizar la salida MIDI del interfaz de audio, en el caso de que disponga de la misma.
- El equipamiento de control de deberá tener entradas MIDI para recibir estos mensajes.
 - Mesas de luces.
 - Softwares de control de video. Deberían conectarse a un puerto MIDI a través de un USB (usando una tarjeta de sonido o una tarjeta MIDI externa)
 - Las mesas de sonido digitales

No se ha contemplado el caso que AhoStage corra en el mismo PC dónde se ejecuta algún otro software de control, porque esta configuración se considera desaconsejable para su uso en directo.

7.1.4. Software de control

En este apartado se detallará el diferente software de control que se podría utilizar para trabajar conjuntamente con AhoStage. Para ello se debe tener en cuenta que la comunicación del secuenciador (AhoStage) con el software de control se realizará mediante notas MIDI; por lo que el requerimiento mínimo necesario es que el software de control sea capaz de funcionar controlado vía estas notas MIDI.

Algunos softwares de control son los siguientes: QLab, Resolume, VVVV, Arkaos8...

7.1.5. Análisis detallado del interfaz lógico con las aplicaciones control

En este apartado se describen los requisitos iniciales para el diseño del protocolo que se utilizará para comunicarse con las aplicaciones de control. Requiere para su definición final el conocer en detalle qué aplicaciones de control serán utilizadas y las capacidades de las mismas. Los requisitos son los siguientes:

- Se utilizará un canal MIDI para distinguir cada una de las aplicaciones de control destinatarias de los mensajes (hasta 16 por lo tanto).
- Se utilizarán distintas notas (configurables) para que cada aplicación pueda disparar diferentes acciones en función de la nota concreta.
- Las notas tendrán una duración predeterminada y fija. El control de la duración de los eventos dependerá de la aplicación de control, no de AhoStage. Es decir, AhoStage sólo dispara.
- La “velocidad” de las notas será fija.

7.2. Entorno de operación de AhoStage

Con el objetivo de tener una visión general de AhoStage y del funcionamiento del sistema en su globalidad, en este subapartado se va presentar gráficamente como podría ser el esquema de una sesión de teatro, desde un punto vista técnico, cual serían la posición de AhoStage y como se le relacionaría con el resto del sistema como pueden ser los micrófonos de entrada de audio o con los diferentes recursos artísticos.

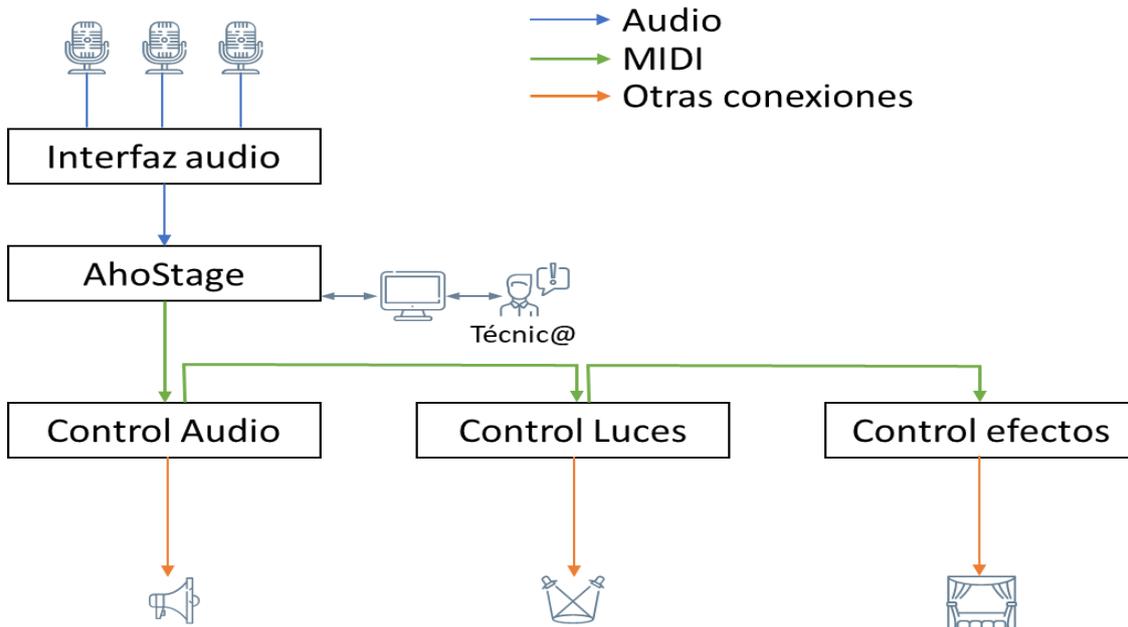


Ilustración 14: Diagrama del sistema en una obra de teatro

El entorno constará de tres partes subsistemas:

- Sistemas de adquisición de audio: esta parte será la encargada de la recogida del audio de los diferentes personajes de la obra. Estará compuesta por los diferentes componentes hardware necesarios como pueden ser micrófonos, cableados, receptores inalámbricos, las señales procedentes de los canales de entrada alimentan un interfaz de audio que realiza la conversión A/D y se conecta al ordenador donde se ejecuta AhoStage.
- AhoStage: será manejado desde el equipo del técnico de escena. En base al análisis de las señales de audio de los diferentes canales, lanzará o no una serie de notas MIDI hacia los diferentes recursos escénicos.
- Recursos escénicos: este subsistema está compuesto por los diferentes recursos escénicos que componen una obra de teatro como pueden ser luces, sonido... y sus correspondientes aplicaciones de control... Por lo tanto, estos recursos recibirán diferentes notas MIDI por su determinado canal, y en base al valor de dicha nota realizarán una acción u otra siguiendo lo indicado en su escaleta propia.

7.2.1. Análisis detallado del interfaz lógico con las aplicaciones control

En este apartado se describen los requisitos iniciales para el diseño del protocolo que se utilizará para comunicarse con las aplicaciones de control. Requiere para su definición final el conocer en detalle qué aplicaciones de control serán utilizadas y las capacidades de las mismas. Los requisitos son los siguientes:

- Se utilizará un canal MIDI para distinguir cada una de las aplicaciones de control destinatarias de los mensajes (hasta 16 por lo tanto).
- Se utilizarán distintas notas (configurables) para que cada aplicación pueda disparar diferentes acciones en función de la nota concreta.
- Las notas tendrán una duración predeterminada y fija. El control de la duración de los eventos dependerá de la aplicación de control, no de AhoStage. Es decir, AhoStage sólo dispara.
- La “velocidad” de las notas será fija.

7.3. Módulos de AhoStage

Una vez conocida la composición y el funcionamiento global del sistema, en este apartado se realizará una descripción de los diferentes módulos que componen AhoStage. En la siguiente imagen se puede observar gráficamente como está compuesto AhoStage. Hay tres módulos principales.

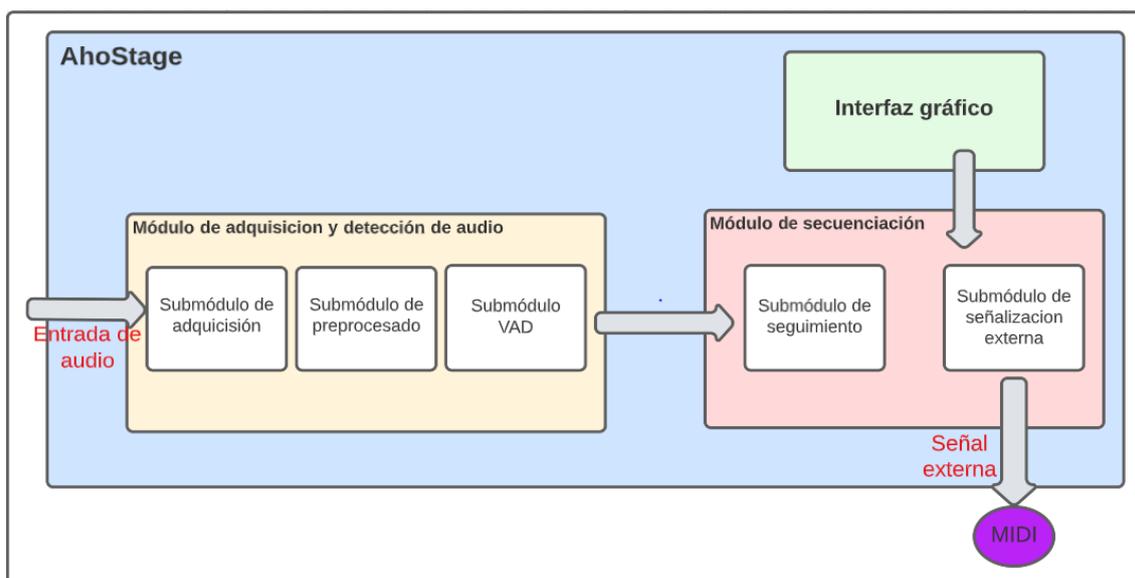


Ilustración 15: Esquema de módulos de AhoStage

7.3.1. Módulo de adquisición y detección de actividad vocal en tiempo real

Este módulo se encargará del manejo de las entradas de audio del sistema, permitiendo la adquisición simultánea en tiempo real de todos los canales de entrada. El audio de cada canal pasa por una etapa de procesamiento de entrada que lo preparará para ser entregado a un detector de actividad vocal. El resultado de la detección se pasará al módulo de secuenciación.

Este módulo constará de submódulos:

- Submódulo de adquisición: Deberá permitir a adquisición simultánea de hasta 8 canales de entrada, y deberá ser independiente del dispositivo conversor que se utilice en la entrada al PC. Utilizará el mezclador de sonido provisto por el sistema operativo para recoger los canales de audio.
- Submódulo de preprocesado: Se encargará de preparar las señales de audio para el algoritmo de detección de actividad vocal. Constará de un algoritmo de filtrado paso alto mediante la implementación de un filtro FIR digital que funciona en tiempo real.
- Submódulo VAD: Este submódulo implementará un algoritmo de detección de actividad vocal, basado en la estimación del espectro a largo plazo.

El nuevo sistema AhoStage tendrá un módulo de adquisición y procesamiento de audio que se basará en el del sistema AhoDubber que se desarrolló para la emisión automática de grabaciones de audio con el doblaje de la obra escénica. En el módulo de entrada se reciben las señales provenientes de los micrófonos de los actores y actrices y se determina quién es el que está hablando. Este módulo utiliza los micrófonos, y un interfaz audio para recibir las señales de audio en el PC, donde serán analizadas. Se describen a continuación las especificaciones de estos elementos.

7.3.2. Módulo de secuenciación

Este módulo recibirá los resultados de la detección de actividad vocal y determinará en qué punto del guion se encuentra el espectáculo. Para ello constará dos submódulos.

- Submódulo de seguimiento: Este submódulo recibirá la información de los VADs y la comparará con el guion del espectáculo que habrá sido formateado para el sistema. Con esta información irá siguiendo la secuencia de las intervenciones del espectáculo. Esta secuencia se podrá ir siguiendo mediante una consola de control, que permitirá su control manual para corregir desajustes.
- Submódulo de señalización externa: Este submódulo se encargará de determinar si un avance en el guion debe disparar algún evento (luz, sonido, video) y, en caso afirmativo, generar y transmitir la señal MIDI.

7.3.3. Interfaz gráfico

Es el encargado de relacionar al usuario con el sistema. En los siguientes apartados se detallarán más concretamente su funcionamiento y como está compuesto.

7.4. Diseño lógico del interfaz gráfico

El interfaz gráfico de AhoStage se articula en torno a una ventana principal en la que se dispondrá de toda la información necesaria para que el operador técnico de escenario pueda supervisar y gestionar el funcionamiento de la aplicación.

Desde esta ventana se accederá mediante las entradas de la barra de menú al diálogo de carga de una nueva obra, a la ventana de asignación de canales de entrada o a una caja de mensajes con la información de la aplicación. En la siguiente ilustración se puede observar el funcionamiento de una manera sencilla.

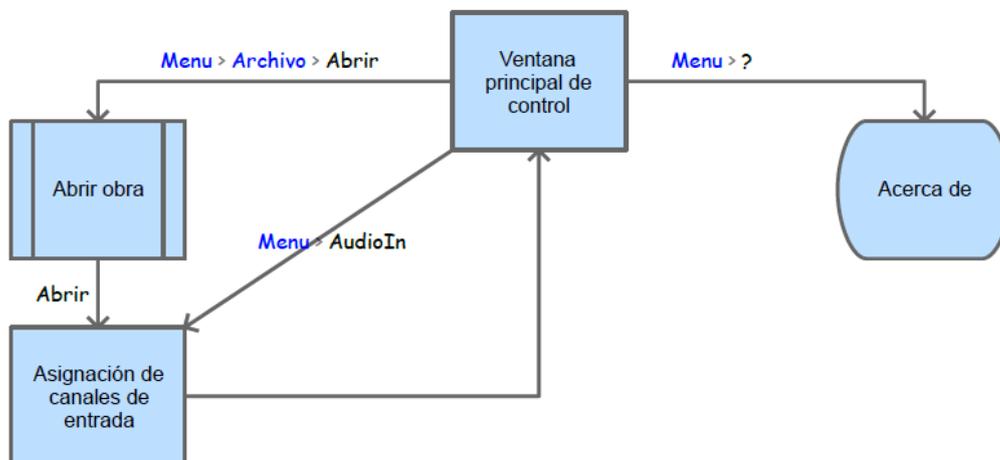


Ilustración 16: Funcionamiento básico de AhoStage

7.4.1. Funcionalidades

De cara al usuario, el software presenta diferentes funcionalidades que tienen como objetivo ayudar al técnico de la obra. A partir de esas necesidades se ha diseñado el interfaz gráfico, de modo que pueda completarlas todas de una forma sencilla e intuitiva para el usuario. A continuación, se van a describir las principales funcionalidades para luego poder entender fácilmente cada uno de los elementos de los formularios.

- Elegir la obra a representar de una manera sencilla y gráfica.
- Configurar los canales de entrada de audio de forma manual.
- Configurar el canal de salida MIDI de forma manual.
- Configurar los parámetros más importantes de la recogida de audio del VAD.
- Lograr un secuenciador automático y poder representarlo de forma gráfica para el usuario.
- Posibilidad de editar el guion desde el interfaz gráfico, editando, añadiendo o eliminando líneas.
- Relacionado con el anterior punto, la posibilidad de guardar los cambios realizados en un archivo .txt nuevo.
- Posibilidad de silenciar tanto la entrada de audio (MuteIn) como la salida MIDI (MuteOut), de forma independiente.
- Poder hacer una búsqueda o filtración en el guion en base al personaje, al canal MIDI o a la nota MIDI.
- Adelantar o retrasar el secuenciador de forma manual.
- Acabar la obra cuando se desee.

Estas son las funcionalidades principales del programa, algunas de ellas se podrán realizar en el mismo formulario o pantalla, y otras se tendrán que realizar en pantallas diferentes.

7.4.2. Funcionamiento lógico

En este apartado se va estudiar el programa desde un punto de vista lógico, es decir cómo, cuándo y de qué manera se pueden aplicar las funcionalidades previamente descritas. Para ello se muestra en la siguiente imagen un diagrama de estados, con cada estado en el que puede encontrarse el programa y como se pasa de un estado a otro. Además, se enumera en cada estado que acciones o funcionalidades están disponibles para el usuario.

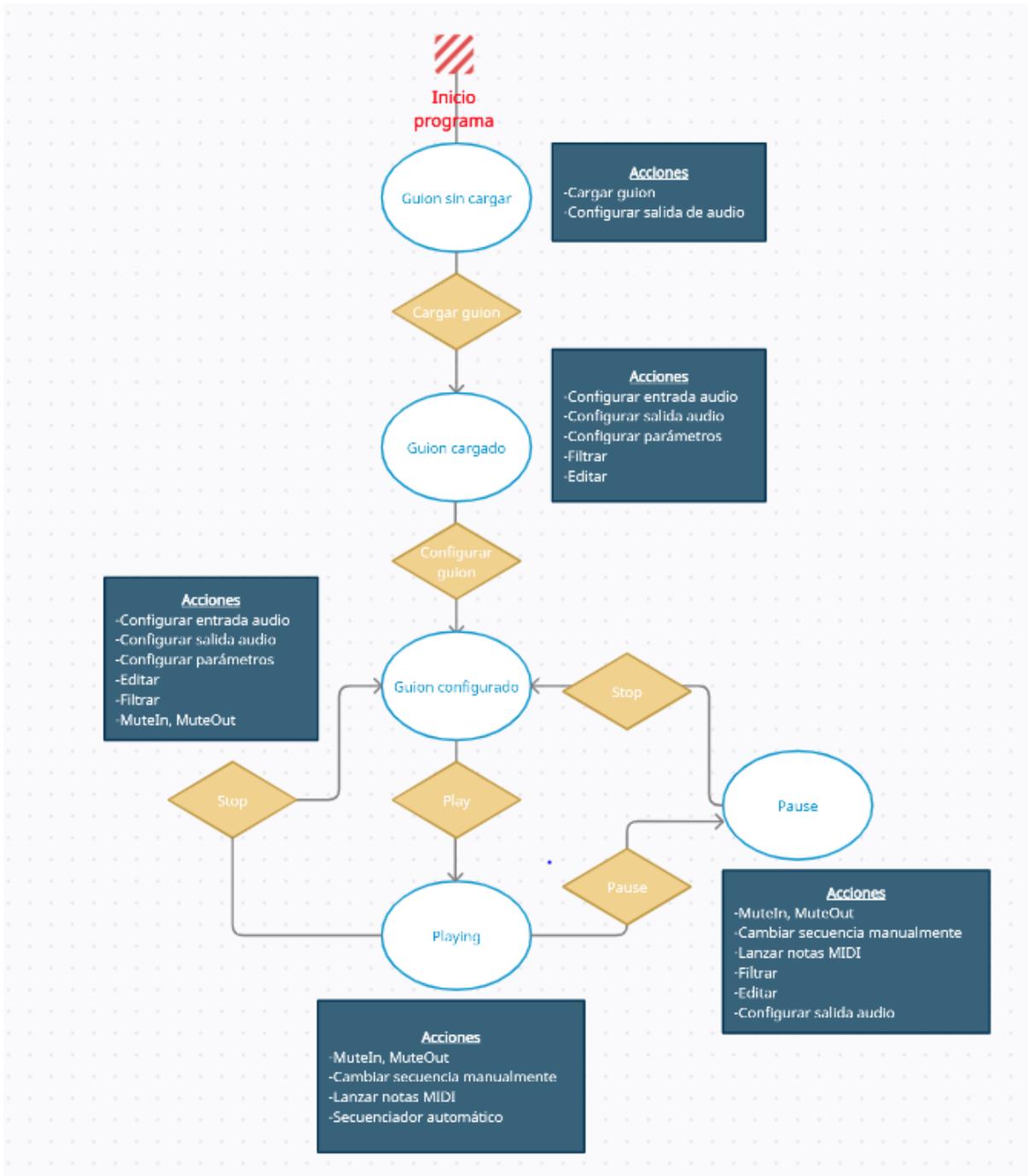


Ilustración 17: Diagrama de estados AhoStage

En la imagen se puede observar que el programa tiene cinco diferentes estados en los que se puede encontrar: guion sin cargar, guion cargado, guion configurado, playing y pause.

7.4.2.1. *Guion sin cargar*

Nada más inicializarse, el programa entra en este estado, siendo el estado base y siendo inevitable entrar en él nada más empezar. En este estado únicamente se pueden realizar dos acciones:

- Cargar guion nuevo: cargar un archivo modo txt.
- Configurar salida audio: aunque no haya ningún guion cargado, se puede configurar la salida MIDI, ya que la falta de guion no afecta a la salida.

Para poder pasar al siguiente estado se debe cargar un guion correctamente. No tiene por qué estar configurada la salida de audio.

7.4.2.2. *Guion cargado*

Una vez cargado el guion en el programa, las acciones que se pueden realizar aumentan. La mayoría de ellas están relacionadas con la configuración del guion.

- Configurar entrada de audio: a partir de este momento se podrán relacionar los personajes del guion con los micrófonos detectados por el sistema.
- Configurar salida de audio: igual que en el estado anterior, se puede elegir el puerto MIDI de salida.
- Configurar parámetros de entrada de audio
- Filtrar
- Editar: también se activará la opción de guardar los cambios realizados en un archivo txt.

El objetivo principal de este estado consiste en configurar correctamente todo lo necesario para poder comenzar la obra. Para ello, para poder dar comienzo a la obra, se deben haber completado dos acciones básicas: tener relacionados todos los personajes con algún dispositivo de entrada y haber elegido un puerto de salida MIDI.

7.4.2.3. Guion configurado

Cuando el guion este completamente configurado el programa estará preparado para dar comienzo a la obra. Este estado es el estado previo a que el software entre en playing. Es la última oportunidad para hacer las configuraciones necesarias. Estas son las acciones posibles de realizar:

- Configurar salida de audio: se podrá cambiar el puerto de salida MIDI.
- Configurar entrada de audio: se podrá cambiar la relación personajes- dispositivos de entrada.
- Configurar parámetros de entrada de audio
- Filtrar
- Editar: también estará disponible la opción de guardar los cambios realizados en un archivo txt.
- MuteIn-MuteOut

En el diagrama anterior se puede observar que después de este estado la única opción es darle al play y dar comienzo a la obra. No obstante, cabe la posibilidad de volver al estado anterior, aunque resulte extraño, si se cumple una de las dos siguientes condiciones: se carga un nuevo guion o se rompen todas las relaciones personajes-dispositivos de entrada. Por lo que, en tal caso se volverá al estado anterior y se deberá volver a realizar la configuración del guion. Pero lo más lógico y usual sería comenzar la obra después de estar en este estado mediante el botón play.

7.4.2.4. Playing

La obra ha comenzado, y se ha activado el secuenciador automático. En este estado la mayoría de las acciones están relacionadas con el manejo del secuenciador y las acciones de configuración y edición quedan desactivadas. Además, no se pueden aplicar los filtros y en caso de haberse aplicado algún filtro antes de entrar en playing, estos se restablecerán.

- MuteIn-MuteOut
- Cambiar secuencia manualmente: mediante unos botones se podrá adelantar la posición del guion manualmente en caso de ser necesario.
- Lanzar notas MIDI: cuando hay un salto de secuencia y la salida MIDI no este silenciada, se lanzarán notas MIDI.
- Secuenciador automático: siempre y cuando no estén silenciados los dispositivos de entrada.

Cuando una obra está el estado playing, es decir esta activa, el usuario tiene dos opciones: acabar la obra mediante el botón *stop* o parar la obra mediante el botón de *pause*. En caso de terminar la obra, el sistema volvería al estado previo (guion configurado) donde la obra está configurada y preparada para comenzar. Cuando se pulsa el botón *stop* el secuenciador vuelve a 0 y todos los cambios realizados (MuteIn, MuteOut...) se restablecen. Por otro lado, en caso de parar la obra, se entrará en otro estado diferente donde se podrán realizar más acciones.

7.4.2.5. *Pause*

Este estado sirve cuando se desea parar la obra, como puede ser para editar el guion, aplicar algún filtro, o porque es necesario parar la obra. Cuando se está en pause, se para el secuenciador y no puede avanzar ni retroceder, pero la posición del secuenciador no se reinicia. Por lo que estas son las acciones que se pueden realizar:

- MuteIn-MuteOut
- Filtrar
- Editar
- Configurar salida de audio MIDI

Destacar que no es posible reconfigurar la entrada de audio, es decir, la relación entre los personajes y los micrófonos. Cuando la obra está en pause el usuario puede tomar dos decisiones: por un lado, terminarla manualmente mediante el botón stop, y, por otro lado, reanudarla en el punto en el que se encontraba antes. Si se opta por dar por terminada la obra, el programa volverá al estado previo a iniciar la obra, con el guion ya cargado y la configuración de entrada y salida de audio ya realizada.

7.5. Diseño gráfico

En este apartado se define el diseño de cada una de las ventanas que forman el interfaz gráfico de la aplicación y se detalla el funcionamiento de cada uno de los controles que componen dichas ventanas.

7.5.1. Ventana principal de control

La ventana principal es el punto desde el que se controla todo el funcionamiento de la aplicación. La información y los controles más importantes se encuentran directamente visibles y accesibles desde esta misma ventana. Esta ventana es la que se abre al arrancar la aplicación y es también

a la que se retorna cuando se finalizan los diálogos particulares de las diferentes opciones. Su diseño esquemático se muestra en la siguiente ilustración.

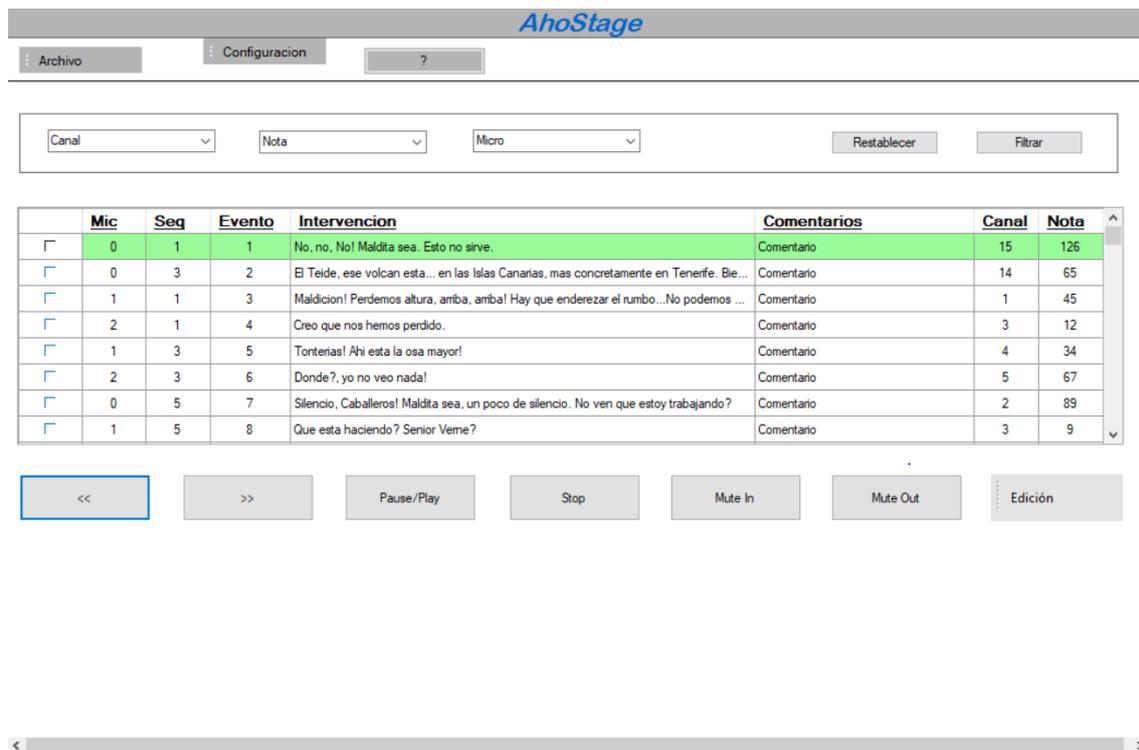


Ilustración 18: Pantalla principal de AhoStage

7.5.1.1. Barra de menú:

Las opciones de menú serán:

- Archivo:

Despliega un menú con dos entradas:

- Cargar guion: carga el guion de una obra escénica, para lo que inicia un diálogo de carga de fichero y configuración de audio.
- Guardar guion: Abre una ventana de diálogo para guardar el guion en lugar que se desee y con el nombre que se desee.

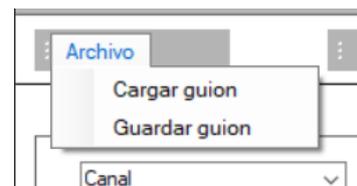


Ilustración 19: Menú archivo

- Configuración:

Despliega un menú con tres entradas:

- AudiIn: carga el formulario para configurar la entrada de audio. No se podrá usar este botón hasta que se haya cargado un guion.
- MIDIout: mediante este botón se carga el formulario de configuración del puerto de salida MIDI.
- Ajustes: con este botón se pueden ajustar los parámetros de recogida de audio.



Ilustración 20: Menu configuración

- ?:

Presenta una caja de mensajes con la información sobre la aplicación.

7.5.1.2. Filtro de visualización de guion

El operador puede filtrar las líneas de guion que aparecen en la ventana de visualización mediante tres campos, cada uno de ellos es un combo desplegable en los cuales se puede seleccionar una opción diferente:

- Micro: para filtrar los eventos producidos por un determinado micrófono.
- Canal (MIDI): corresponde a la aplicación externa que recibirá las notas MIDI generadas.
- Nota: filtrar mediante la nota MIDI.

La selección se realizará mediante combos desplegables, y habrá dos diferentes botones.

- Filtrar: aplica los filtros seleccionados.
- Restablecer filtros: borra cualquier filtro aplicado anteriormente y muestra todo el guion.

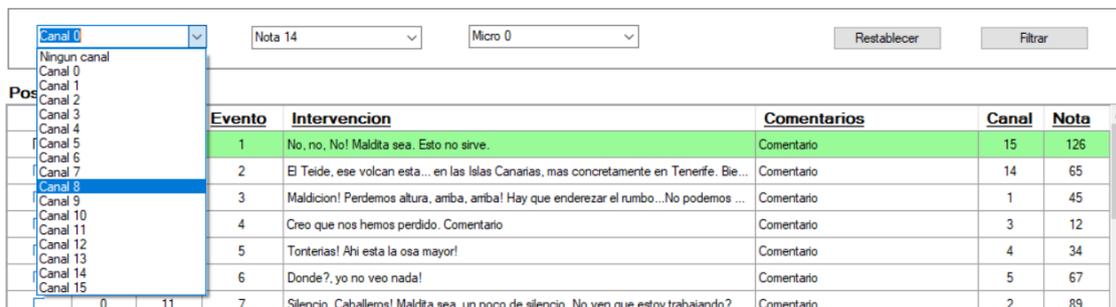


Ilustración 21: Menú de filtros

7.5.1.3. *Tabla de visualización de guion*

En esta tabla aparecerá el guion completo, aunque puede filtrarse mediante los filtros de visualización. Algunas columnas son ordenables, siempre y cuando la obra esté en estado de Pause. La tabla constará de 8 columnas:

- Columna de selección de línea de guion: Esta columna no es ordenable y contiene un Radiobutton que permite seleccionar una línea de guion para posicionarse en ella mediante el botón de “Play/Pause”. Para poder cambiar de línea la obra debe estar en pausa.
- Columna Mic: Indica el canal de entrada (que corresponde con el personaje del guion). Esta columna es ordenable y filtrable.
- Columna Seq: Indica el número de secuencia de la línea de guion en el conjunto de las intervenciones. Esta columna es ordenable y se utiliza como ordenación por defecto en la carga de la tabla.
- Columna evento: Indica el evento al que corresponde esa línea del guion. Esta columna es ordenable.
- Columna de texto: Contiene el texto de la intervención del guion. Esta columna no es ordenable.
- Columna de comentarios: Mediante esta columna se podrá añadir cualquier comentario extra relacionado con el evento con el objetivo de ayudar al técnico.
- Columna Canal: Indica el canal MIDI por el que se enviará la nota asociada a la intervención. Esta columna es ordenable y filtrable.
- Columna Nota: Indica la nota MIDI que se enviará. Esta columna es ordenable y filtrable.

Además, se sombreadá de forma inversa la siguiente fila de la tabla, es decir, quedara en tono más oscuro el siguiente evento de la obra. De esta forma, se podrá saber en todo momento en que punto de la obra se encuentra. También cabe destacar que la primera línea de la table será siempre la sombreada, de esta forma según vaya avanzando la obra, irá cambiando la primera línea de la tabla.

Posición: 0

	Mic	Seg	Evento	Intervencion	Comentarios	Canal	Nota
▢	0	1	1	No, no, No! Maldita sea. Esto no sirve.	Comentario	15	126
▢	0	2	2	El Teide, ese volcan esta... en las Islas Canarias, mas concretamente en Tenerife. Bie...	Comentario	14	65
▢	0	4	3	Maldicion! Perdemos altura, amba, amba! Hay que enderezar el rumbo...No podemos ...	Comentario	1	45
▢	0	6	4	Creo que nos hemos perdido. Comentario	Comentario	3	12
▢	0	8	5	Tonterias! Ahi esta la osa mayor!	Comentario	4	34
▢	0	10	6	Donde?, yo no veo nada!	Comentario	5	67
▢	0	11	7	Silencio, Caballeros! Maldita sea, un poco de silencio. No ven que estoy trabajando?	Comentario	2	89
▢	0	14	8	Que esta haciendo? Senior Verme?	Comentario	3	9

Ilustración 22: Tabla de visualización de guion

7.5.1.4. Botones de control de secuenciación y control

En esta parte de la ventana se encuentran los botones que permiten controlar el posicionamiento del secuenciador en el guion de la obra. Además, se encuentra el menú para tener la posibilidad de editar el guion.

- Botón atrás (<<): retrocede una intervención en la secuencia.
- Botón Play/Pause: para o reanuda la obra, dependiendo del estado del que venga.
- Botón stop: termina la obra, vuelve al estado principal
- Botón adelante (>>): adelanta una intervención en la secuencia.
- Botón MuteIn: deshabilita las entradas de los micrófonos con lo que el operador puede realizar un control totalmente manual de la secuenciación.
- Botón MuteOut: deshabilita la salida MIDI, de esta forma el secuenciador puede seguir avanzando, ya sea de forma automática o manual, pero el programa no emite notas MIDI en la salida
- Menú editar: mediante este menú se podrá editar el guion durante el transcurso de una obra, para ello la obra debe estar en pausa. Hay tres modos de edición, los cuales se eligen mediante el menú, y después de haberse seleccionado uno de ellos se abre un nuevo formulario de edición.
 - Opción editar línea: da la opción de editar una línea ya existente. Para ello, se debe haber seleccionado la línea mediante el RadioButton.
 - Opción añadir línea: se añade una nueva línea en el guion con los datos que el usuario desee.

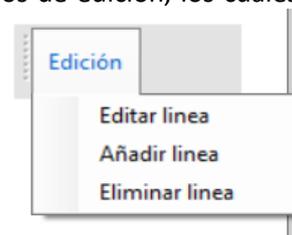


Ilustración 23: Menú de edición

- Opción eliminar línea: elimina la línea del guion seleccionada mediante el RadioButton.

Los botones Pause/Play, MuteIn y MuteOut pueden estar en dos colores diferentes: verde y rojo. El color rojo representa que dicha opción esta desactivada y el color verde representa que esta activada.



Ilustración 24: Botones de control de secuenciación y control

7.5.2. Diálogo “Cargar guion”

La entrada Menú>Archivo>Cargar Guion permite abrir y cargar un guion de una obra escénica y configurar las entradas correspondientes a cada personaje del guion.

El diálogo comienza con un diálogo estándar de apertura de fichero del sistema operativo, que se muestra en la siguiente ilustración. Desde éste se puede navegar por el árbol de directorios y filtrar los tipos de ficheros. El sistema mostrara únicamente los archivos .txt de forma automática, ya que los guiones deben tener esa extensión.

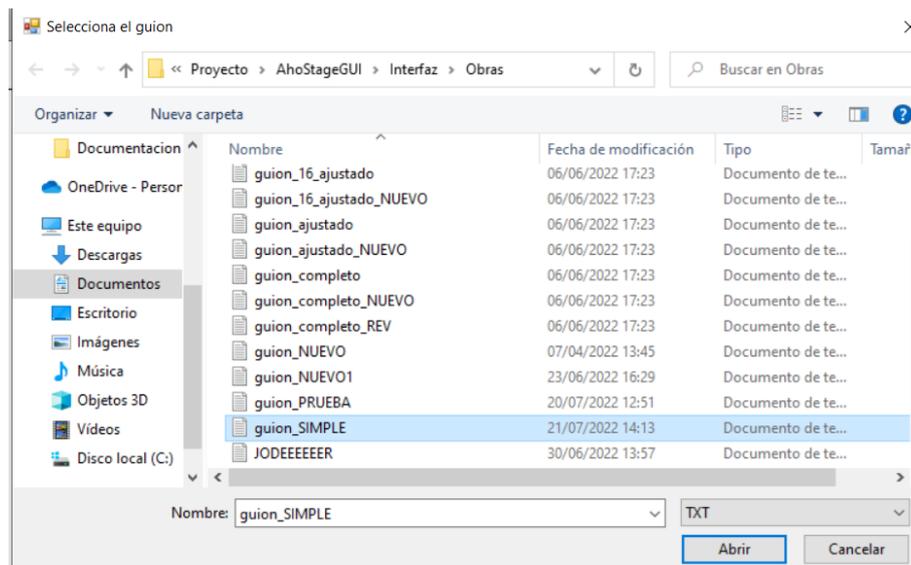


Ilustración 25: Diálogo para cargar el guion

Una vez abierto y cargado el guion se desplegará la ventana de asignación de canales de entrada para configurar la correspondencia entre los micrófonos del sistema y los personajes del guion. Esta ventana se detalla en apartados siguientes.

7.5.3. Diálogo “Guardar guion”

Al igual que el diálogo “cargar guion” se abre mediante Menú>Archivo>Cargar y permite guardar el guion editado con el nombre que el usuario desee. El sistema guardará el guion con el formato adecuado y con una extensión .txt.

El usuario podrá navegar a través del árbol de directorios y guardar el archivo en el lugar adecuado.

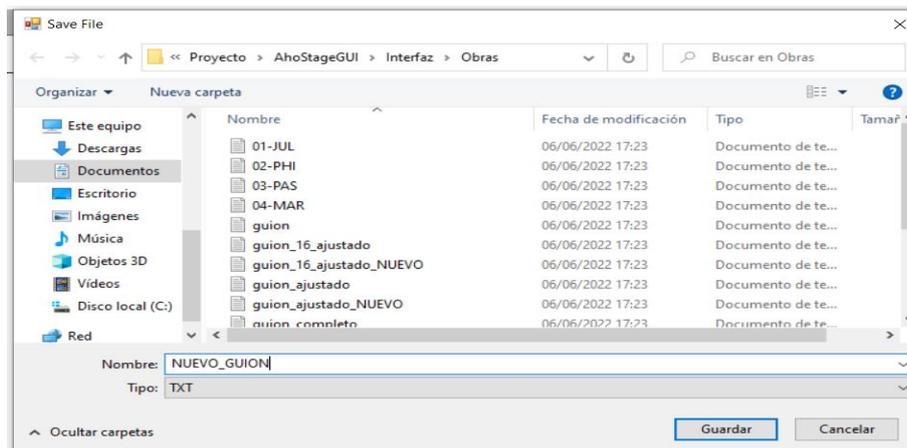


Ilustración 26: Diálogo para guardar el guion

7.5.4. Ventana de asignación de canales de entrada

La ventana de asignación de canales de entrada permite asociar los micrófonos (más propiamente los dispositivos de entrada de audio del sistema) a los diferentes personajes intervinientes en el guion cargado.

A esta ventana se accede como parte del diálogo de apertura de un guion, una vez seleccionado y abierto, y se puede también acceder desde la entrada de menú “AudioIn”, para modificar una configuración ya realizada. Si se intenta acceder a esta ventana sin haber cargado ningún guion se mostrará una caja de mensajes de error indicando que no hay guion disponible para hacer la asignación.

L

La ventana se muestra en la siguiente ilustración. En ella se muestran dos listas con desplazamiento vertical que permiten introducir la relación entre dispositivos de entrada de audio y personajes del guion.

Para asignarlos, se selecciona un dispositivo de la lista de dispositivos de entrada y un personaje de la lista de personajes del guion y se pulsa el botón “Asignar”. En ese momento las listas se actualizan y el dispositivo elegido desaparece de la lista de dispositivos y en la de personajes se incluye el dispositivo asignado junto con el personaje en la entrada. El botón “Asignar” sólo se habilitará cuando se hayan seleccionado tanto un dispositivo como un personaje sin dispositivo asignado en la segunda lista.

El botón desasignar se habilita cuando se ha seleccionado un personaje con asignación en la lista de personajes del guion. Al pulsarlo se deshace la asignación y ambas listas se actualizan.

Además, mediante el botón buscar dispositivos se podrá actualizar la tabla de micrófonos en caso de haberse conectado nuevos micrófonos al sistema.

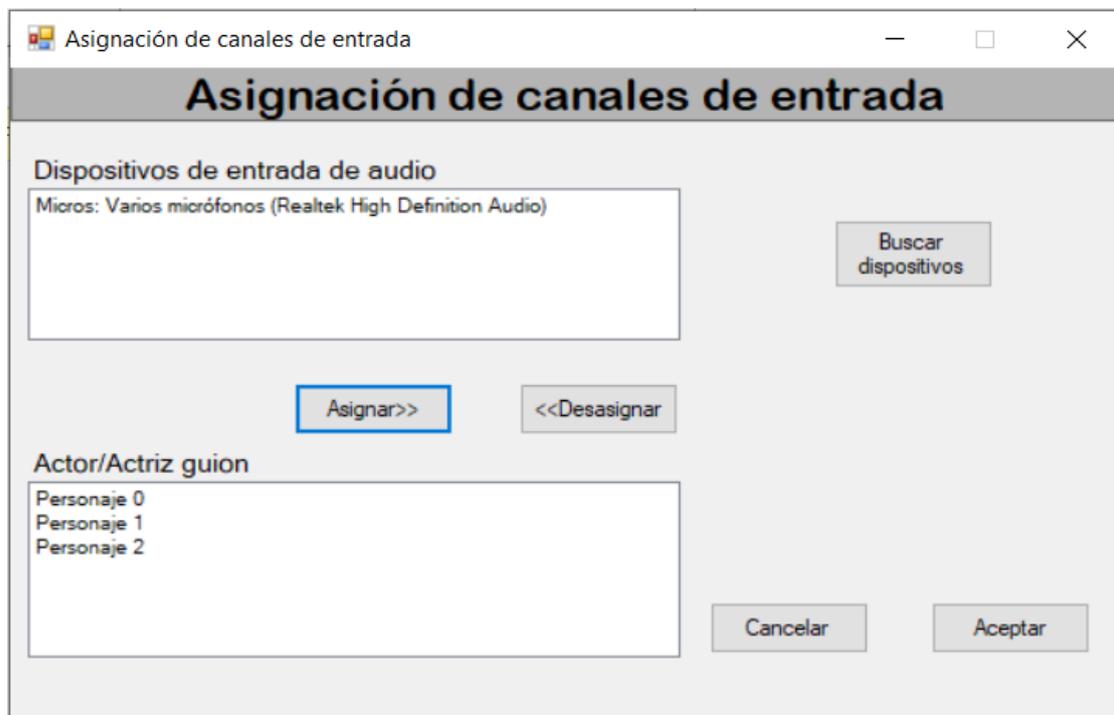


Ilustración 27: Ventana de configuración de entrada de audio

La ventana incluye botones de aceptar y cancelar. El botón de aceptar sólo se habilitará cuando todos los personajes del guion tengan algún dispositivo asignado.

7.5.5. Ventana de asignación de puerto de salida

A través de esta ventana se mostrará una lista con los diferentes puertos MIDI disponibles. A esta ventana se accede a través del menú desplegable del formulario principal “Configuración” y después “MIDIOut”. Para que la obra pueda comenzar se debe haber seleccionado un puerto MIDI anteriormente.

En la siguiente ilustración se muestra la ventana. En el centro del formulario aparecerá una lista con los diferentes puertos de salida MIDI y debajo dos botones: “Aceptar” y “Cancelar”. En caso de seleccionar un puerto y pulsar el botón aceptar, ese puerto quedaría asignado como puerto de salida.

El botón “Buscar puertos MIDI” buscará nuevos puertos MIDI conectados al sistema, en caso de que se hayan conectados nuevos.



Ilustración 28: Ventana de configuración MIDI

7.5.6. Ventana de edición de línea

Para que se abra esta ventana la obra debe estar en pausa y una línea seleccionada en la columna 1 de la tabla. En ese momento se abrirá un nuevo formulario, el cual dejará editar cualquier parámetro de dicha línea.

En caso de querer guardar los cambios, se pulsará el botón aceptar y se guardarán los cambios. En sistema realizará una comprobación para determinar si los cambios realizados son aceptables, y en caso de que no lo sean, mostrará un mensaje advirtiendo de que alguna modificación es incorrecta. Por el contrario, si la edición es correcta se volverá al formulario anterior y se actualizará la tabla con los cambios realizados. Para guardar los cambios realizados en un documento de texto se debe utilizar el botón “Guardar Guion” del menú “Archivo”

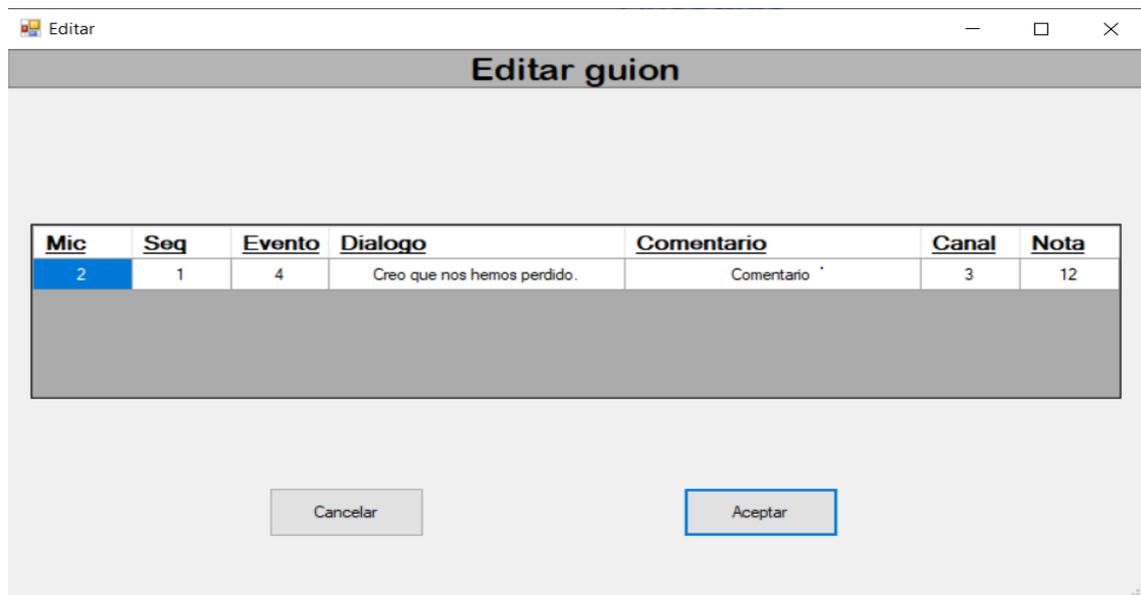


Ilustración 29: Ventana de edición de guion

Esta ventana también se abrirá en caso de querer añadir o eliminar una nueva línea al guion, con la opción de guardar o cancelar los cambios introducidos.

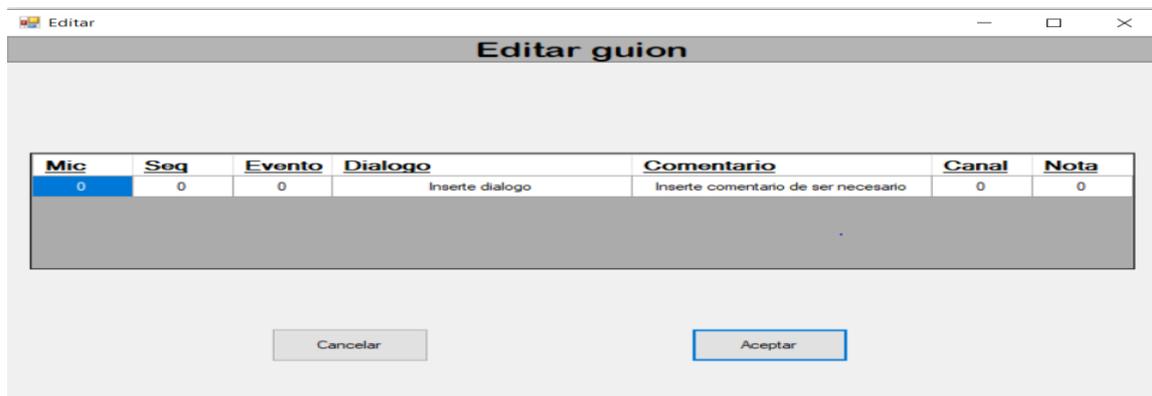


Ilustración 30: Ventana para añadir línea al guion

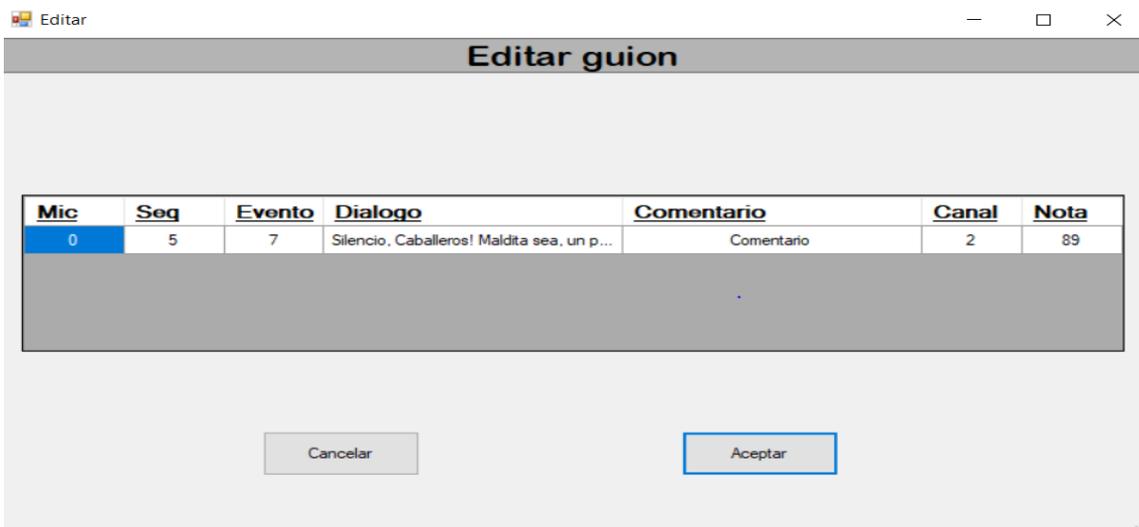


Ilustración 31: Ventana para eliminar línea del guion

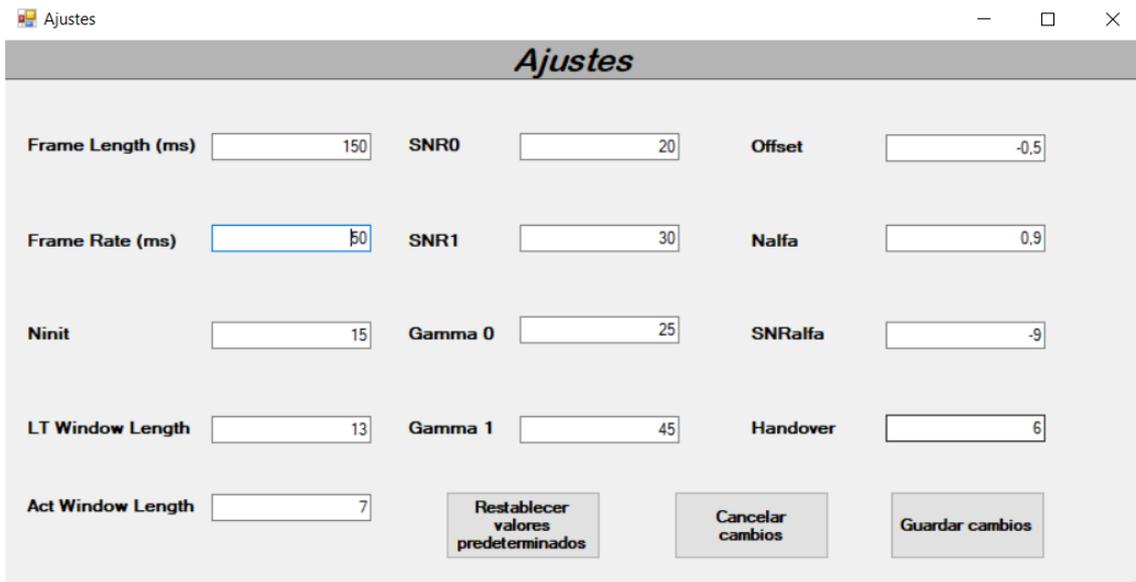
7.5.7. Pestaña de ajustes

A esta ventana se podrá acceder desde el menú de configuración, configuración-->Ajustes. En esta ventana se dará la posibilidad de modificar algunos parámetros o variables utilizados en el VAD, es decir, en la recogida de audio. De esta manera, el usuario tendrá la opción de cambiar de una manera sencilla y rápida los parámetros sin tener la necesidad de modificar ficheros o modificar el código.

Cabe destacar que estas modificaciones únicamente se podrán realizar antes de comenzar una obra, por lo que, una vez comenzada la obra cualquier modificación hecha no tendrá efecto en dicha obra.

Cuando se inicia el programa, estos parámetros toman unos valores estándar o predeterminados, de esta manera, si el usuario no desea realizar ningún cambio en ellos, cuando se inicie la obra los parámetros tomarán esos valores. Por otro lado, en esta ventana se dispone del botón "Restablecer valores predeterminados", que en caso de haber realizado alguna modificación y querer volver a los valores predeterminados del programa, mediante este botón se podrá volver a ellos. Además del botón "Restablecer valores predeterminados", en esta ventana se disponen de los botones "Cancelar cambios" y "Guardar cambios", el primero de ellos para no guardar los cambios realizados y el segundo de ellos para aplicar los cambios realizados.

La gran mayoría de los parámetros modificables tiene relación con la longitud de la ventana, la relación señal ruido (SNR), el handover, el offset... En la siguiente imagen se puede observar el formato de la ventana y los parámetros modificables:



Parameter	Value
Frame Length (ms)	150
Frame Rate (ms)	30
Ninit	15
LT Window Length	13
Act Window Length	7
SNR0	20
SNR1	30
Gamma 0	25
Gamma 1	45
Offset	-0.5
Nalfa	0.9
SNRalfa	-9
Handover	6

Ilustración 32: Ventana de ajustes

7.5.8. Caja de mensajes “?”

Esta caja de mensajes se mostrará mediante la entrada de menú “?” y presenta los créditos de la aplicación.

7.6. Formato del guion

El archivo de la obra que contiene el guion y el cual se cargará en el programa deberá tener un determinado formato para que de esta forma pueda ser interpretado. Deberá ser un .txt y contener la siguiente estructura:

0	1	1	15	126	No, no, No! Maldita sea. Esto no sirve. Comentario	
0	3	2	14	65	El Teide, ese volcan esta... en las Islas Canarias, mas concretamente en Tenerife. Bien aqui esta.	Comentario
1	1	3	1	45	Maldicion! Perdemos altura, arriba, arriba! Hay que enderezar el rumbo...No podemos volver atras.	Comentario
1	1	3	1	27	Maldicion! Perdemos altura, arriba, arriba! Hay que enderezar el rumbo...No podemos volver atras.	Comentario
2	1	4	3	12	Creo que nos hemos perdido. Comentario	
1	3	5	4	34	Tonterias! Ahi esta la osa mayor! Comentario	Comentario
2	3	6	5	67	Donde?, yo no veo nada! Comentario	
0	5	7	2	89	Silencio, Caballeros! Maldita sea, un poco de silencio. No ven que estoy trabajando? Comentario	Comentario
1	5	8	3	9	Que esta haciendo? Senior Verne? Comentario	Comentario
0	7	9	5	67	Un experimento. Comentario	
1	7	10	8	45	Y cual es el objetivo de dicho experimento? Comentario	Comentario
0	9	11	3	134	Quiero saber si una erupcion de gas podria propulsar un cuerpo hasta la Luna. Comentario	Comentario
2	5	12	10	22	Quiere lanzar un trozo de papel hasta la Luna? Comentario	Comentario
0	11	13	12	34	No, maldita sea, solo es un experimento. Este papel deberia llegar hasta esa papelera. Comentario	Comentario
1	9	14	3	11	Por que no prueba a ir a la Luna en globo? Comentario	Comentario
0	13	15	5	23	Eso es imposible. Comentario	Comentario
1	11	16	6	78	Pues deberia... Comentario	Comentario
0	15	17	2	67	Silencio, por favor! Comentario	Comentario
0	17	18	4	54	Tengan cuidado Comentario	Comentario

Ilustración 33: Ejemplo de guion

Como se puede observar en la imagen anterior, el documento de texto constará de siete columnas separadas por una tabulación. Este es el contenido de cada una de las columnas:

- Columna 1: tendrá un valor numérico, cada uno de ellos valdrá para identificar a un personaje de la obra.
- Columna 2: mostrará el número de secuencia de la frase del guion.
- Columna 3: representa el número de evento, deberá estar ordenado de forma ascendente.
- Columna 4: especifica el canal MIDI a utilizar.
- Columna 5: especifica la nota MIDI a lanzar.
- Columna 6: guarda la frase del guion.
- Columna 7: en esta última columna se podrá añadir un comentario adicional.

8. Planificación del proyecto

Se presenta una planificación aproximada del proyecto AhoStage, por lo que tanto la duración, la fecha de inicio y la fecha de entrega y los recursos utilizados para cada una de las tareas que posteriormente van a ser expuestas son orientativas.

Para realizar una correcta planificación, el proyecto se ha dividido en diferentes tareas de trabajo, y en cada una de ellas habrá una serie de subtareas a realizar. En cuanto a la fecha de inicio y la fecha final del proyecto, en este caso serán 1 de marzo de 2022~~0~~ y 14 de julio de 2022 respectivamente. Además, hay que tener en cuenta que este proyecto requiere una dedicación total de 600 horas.

Por un lado, se va a detallar la información principal de cada paquete de trabajo y de cada tarea, en que consiste, fecha de comienzo y final y horas dedicadas. Por otro lado, se insertará un diagrama de Gantt para poder ver gráficamente la planificación.

En cuanto al personal humano que ha tomado parte en este proyecto, destacan los ingenieros Ibon Saratxaga y Peio Gonzalez, además del personal vario de la compañía Eskena que puntualmente han ido colaborando.

- Ibon Saratxaga: director del trabajo de fin de master. Ibon es Ingeniero en Telecomunicaciones y actualmente profesor en la Escuela de Ingeniería de Bilbao. Además, forma parte del grupo de investigación AHOLAB (equipo con el que se ha realizado este proyecto) en el que trabaja en síntesis de voz y machine listening buscando el aprendizaje automático. Su función en el proyecto ha sido la de Ingeniero Senior, teniendo la labor de guiar y ayudar al Ingeniero Junior en sus tareas y en las tomas de decisiones.
- Peio Gonzalez: Ingeniero en Telecomunicaciones y terminando el master en Telecomunicaciones, especializado en el área telemática. En este proyecto tiene la función de Ingeniero Junior, encargado del desarrollar del software y de elaborar la documentación requerida. A su vez, el proyecto le servirá como el trabajo de fin de master.
- Personal de Eskena: durante el proyecto diferentes miembros de la asociación Eskena han colaborado puntualmente en el proyecto, mediante reuniones con los demás participantes. Los más destacados son el director, el encargado del presupuesto y, principalmente, el técnico de Eskena.

Participante	Abreviatura	Función
Ibon Saratxaga	IS	Ingeniero Senior
Peio Gonzalez	PG	Ingeniero Junior
Personal de Eskena	EK	Colaboradores

Tabla 4: Participantes en el proyecto AhoStage

A continuación, se va a describir el trabajo realizado en cada tarea y en cada una de sus subtareas. Las tareas se irán realizando secuencialmente, es decir, hasta que no concluya el anterior no se comenzará con la siguiente. En cuanto a las subtareas, algunas se realizarán secuencialmente y otras en paralelo (siempre y cuando las subtareas realizadas en paralelo pertenezcan a la misma tarea).

8.1. PT1: Definición del proyecto

Los objetivos principales de esta tarea son varios, entre los principales están analizar los requerimientos del proyecto, definir los objetivos principales y realizar una planificación aproximada del proyecto. Para ello se realizarán reuniones constantes con miembros de la asociación de teatro Eskena con el fin de acordar todos los puntos necesarios para llevar a cabo el proyecto, así como el método de trabajo, fechas de entrega, especificaciones del software o documentación a entregar.

En esta tarea también se hará un análisis del proyecto AhoDubber, con el objetivo de poder reutilizar las partes que se pueden adaptar a este nuevo proyecto.

Fecha de inicio	1 de marzo de 2022
Fecha final	18 de marzo de 2022
Duración(días)	14
Carga de trabajo(horas)	112
Participantes	IS-PG-EK

Tabla 5: Datos tarea PT1

Esta tarea constará de cuatro subtareas: planificación y metodología del proyecto, descripción de especificaciones, análisis de AhoDubber y reuniones con Eskena. Las tres primeras subtareas se realizarán secuencialmente, esto es, hasta que no termine la anterior no comenzará la siguiente. Mientras que la última subtask, se realizará simultáneamente con todas las anteriores.

8.1.1. A101: Planificación y metodología

Durante esta subtarea se realizará una planificación aproximada pero lo más detallada posible del proyecto. Se planificarán las siguientes tareas, sus objetivos, fechas de inicio y finalización... También se especificará que personal se necesita en el proyecto y que recursos materiales serán necesarios durante todo el proyecto, así como ordenadores, software, conexiones MIDI, micrófonos...

Fecha de inicio	1 de marzo de 2022
Fecha final	6 de marzo de 2022
Duración(días)	4
Carga de trabajo(horas)	30
Participantes	IS-PG

Tabla 6: Datos subtarea A101

8.1.2. A102: Descripción de especificaciones

Mediante esta tarea se definirán los objetivos del proyecto. Para ello, son indispensables las reuniones con diferentes miembros de Eskena, para que de esta forma entre ambas partes se pueda llegar a un acuerdo. Se definirá exactamente como debe funcionar el software y cómo será la interfaz gráfica del programa, tanto desde un punto de vista de visibilidad como desde un punto de vista de funcionalidad. Todas las especificaciones y acuerdos, deberán ser recogidos en diferentes documentos a entregar.

Fecha de inicio	7 de marzo de 2022
Fecha final	13 de marzo de 2022
Duración(días)	5
Carga de trabajo(horas)	32
Participantes	IS-PG

Tabla 7: Datos subtarea A102

8.1.3. A103: Análisis de AhoDubber

Esta subtarea consistirá en saber que partes del proyecto AhoDubber pueden ser de utilidad para el proyecto AhoStage. AhoDubber es un proyecto realizado años anteriores que consistía en un doblador de voz automático, por lo que muchas partes de ese proyecto pueden ser de utilidad para el actual. Además, es fundamental entender su código y el funcionamiento para

poder aplicarlo a AhoStage. Durante esta subtarea también se intentará ejecutar AhoDubber en diferentes ordenadores.

Fecha de inicio	14 de marzo de 2022
Fecha final	20 de marzo de 2022
Duración(días)	5
Carga de trabajo(horas)	38
Participantes	PG

Tabla 8: Datos subtarea A103

8.1.4. A104: Reuniones con Eskena

Como ya se ha dicho anteriormente, esta subtarea se realizará en paralelo con todas las anteriores, de esta forma el contacto con la asociación Eskena será constante. Las reuniones serán de online y en ellas es necesario que participen diferentes miembros de Eskena para que cada uno de ellos pueda aportar la visión de su ámbito de trabajo, como puede ser el técnico, el guionista, la tesorera, la directora... En estas reuniones ellos expondrán sus necesidades y requerimientos para el proyecto, los cuales serán recogidos en documentos. Además, se acordarán las fechas de entrega.

Fecha de inicio	1 de marzo de 2022
Fecha final	20 de marzo de 2022
Duración(días)	14
Carga de trabajo(horas)	12
Participantes	IS-PG-EK

Tabla 9: Datos subtarea A104

8.2. PT2: Diseño del interfaz gráfico

La premisa principal de esta tarea es diseñar y desarrollar la interfaz gráfica que utilizará el usuario final para interactuar con el software. En la tarea PT1, concretamente en la subtarea A102, se ha acordado con Eskena como debe ser este interfaz, tanto físicamente como técnicamente, esto es, como debe funcionar. Como el software va ser para Windows, el interfaz grafico se desarrollará mediante formularios Windows de Visual Studio.

Fecha de inicio	21 de marzo de 2022
Fecha final	20 de mayo de 2022
Duración(días)	44
Carga de trabajo(horas)	264
Participantes	IS-PG

Tabla 10: Datos tarea PT2

Esta tarea constara de cuatro diferentes subtareas, las cuales se realizarán consecutivamente.

8.2.1. A201: Familiarización con Windows Forms

Para poder desarrollar correctamente el interfaz necesario es de vital importancia entender el funcionamiento de Windows Forms para luego poder aplicar los conocimientos al desarrollo del interfaz. Por ello, durante esta subtask se conseguirán los conocimientos básicos de Windows Forms mediante cursos online, videos, foros...

Fecha de inicio	21 de marzo de 2022
Fecha final	23 de marzo de 2022
Duración(días)	3
Carga de trabajo(horas)	24
Participantes	PG

Tabla 11: Datos subtask A201

8.2.2. A202: Diseño físico del interfaz

En esta subtask se diseñará físicamente el interfaz, de esta forma se crearán los formularios necesarios y en ellos se introducirán los elementos requeridos, así como botones, tablas, labels, menús desplegables... Únicamente se colocarán en los formularios sin realizar la programación de cada uno de ellos.

Fecha de inicio	24 de marzo de 2022
Fecha final	13 de abril de 2022
Duración(días)	15
Carga de trabajo(horas)	54
Participantes	IS-PG

Tabla 12: Datos subtask A202

8.2.3. A203: Programación del interfaz

Partiendo del interfaz gráfico creado en la subtarea anterior, A302, en esta subtarea se programará el interfaz gráfico. Se programará cada elemento y cada evento del formulario.

Fecha de inicio	14 de abril de 2022
Fecha final	18 de mayo de 2022
Duración(días)	25
Carga de trabajo(horas)	170
Participantes	PG

Tabla 13: Datos subtarea A203

8.2.4. A204: Pruebas

Con esta subtarea se podrá comprobar que el funcionamiento del interfaz gráfico es el deseado. En caso de haber un error, se corregirá. Después de esta subtarea el interfaz grafico estará listo para ser implementado

Fecha de inicio	19 de mayo de 2022
Fecha final	20 de mayo de 2022
Duración(días)	2
Carga de trabajo(horas)	16
Participantes	IS-PG

Tabla 14: Datos subtarea A204

8.3. PT3: Actualización del software AhoStage

En toda esta tarea se adaptará el software de AhoDubber a las nuevas necesidades. De esta manera se creará un módulo de generación de señales MIDI para poder gestionar la salida. Además, se deberá modificar el código para descartarse de todas las partes sobrantes o modificar esas partes para el nuevo sistema, como pueden ser la entrada de audio o el bucle principal.

Fecha de inicio	23 de mayo de 2022
Fecha final	17 de junio de 2022
Duración(días)	20
Carga de trabajo(horas)	160
Participantes	PG

Tabla 15: Datos tarea PT3

8.3.1. A301: Diseño e implementación de un módulo de generación de señales de secuenciación

Esta subtarea es una de las más importantes del proyecto. El objetivo es crear un módulo capaz de leer el guion y a partir de ahí crear notas MIDI y mandarlas por la salida secuencialmente. El módulo deberá hacer toda la gestión de puertos MIDI, como buscarlos, abrirlos y cerrarlos, mandar los mensajes...

Fecha de inicio	23 de mayo de 2022
Fecha final	27 de mayo de 2022
Duración(días)	5
Carga de trabajo(horas)	40
Participantes	PG

Tabla 16: Datos subtarea A301

8.3.2. A302: Módulo para la asignación de canales de entrada

Un punto muy importante del proyecto es la gestión de los canales de entrada. El usuario es el que debe elegir cada personaje con que micrófono está relacionado, por lo que en el código se deben hacer modificaciones para que el usuario pueda ir haciendo esas relaciones dinámicamente. Este módulo se encargará de todo lo relacionado con la gestión de canales de entrada como buscar los micrófonos disponibles, asignarlos a personajes, desasignarlos, cerrarlos...

Fecha de inicio	30 de mayo de 2022
Fecha final	3 de junio de 2022
Duración(días)	5
Carga de trabajo(horas)	40
Participantes	PG

Tabla 17: Datos subtarea A302

8.3.3. A303: Cambios en el software AhoStage para adaptarlo a los nuevos requerimientos

Durante esta subtarea se modificará el código de AhoDubber para adaptarlo a las nuevas características. Una de las modificaciones más importantes es la del bucle principal. Además, muchas partes del programa ya no serán necesarias y podrán ser eliminadas.

Fecha de inicio	6 de junio de 2022
Fecha final	17 de junio de 2022
Duración(días)	10
Carga de trabajo(horas)	80
Participantes	PG

Tabla 18: Datos subtarea A303

8.4. PT4: Unificación del módulo AhoStage y del interfaz gráfico

Esta tarea no tendrá ninguna subtarea, será una única tarea. Tendrá un único objetivo: juntar el módulo de AhoStage y el módulo del interfaz gráfico. Antes de llegar a esta tarea, se tendrán los dos módulos independientemente. Durante esta tarea se corregirán todos los posibles errores de compatibilidad que puedan surgir durante la unión. Después de esta tarea se logrará un único programa el cual estará preparado para comenzar el testeo.

Fecha de inicio	20 de junio de 2022
Fecha final	24 de junio de 2022
Duración(días)	5
Carga de trabajo(horas)	40
Participantes	PG

Tabla 19: Datos tarea PT4

8.5. PT5: Pruebas del nuevo sistema en entorno no realista y realista

La premisa principal de la tarea es comprobar el correcto funcionamiento del programa y en caso de aparecer errores corregirlos. Para ello se llevarán a cabo diferentes métodos de testeo para corroborar el buen funcionamiento en diferentes situaciones. Se crearán dos entornos diferentes: uno no realista (en el laboratorio) y otro realista (una simulación real de una obra).

Fecha de inicio	27 de junio de 2022
Fecha final	8 de julio de 2022
Duración(días)	10
Carga de trabajo(horas)	80
Participantes	IS-PG-EK

Tabla 20: Datos tarea PT5

8.5.1. A501: Periodo de pruebas en un entorno no realista

Esta subtarea se desarrollará en el laboratorio, que consistirá en diferentes pruebas. Primeramente, se realizará un *debugeo* del código para comprobar que se compila y ejecuta correctamente. Posteriormente, se testeará cada módulo de forma independiente para acabar testeando el programa en su globalidad. Para llevar a cabo las pruebas, se dispondrá del material necesario como micrófonos, puertos de salida MIDI, guiones de obras de teatro... Durante todas las pruebas se irán apuntando los errores o posibles mejoras. En este periodo de pruebas únicamente tomarán parte personal del grupo de investigación AHOLAB.

Fecha de inicio	27 de junio de 2022
Fecha final	28 de junio de 2022
Duración(días)	2
Carga de trabajo(horas)	16
Participantes	IS-PG

Tabla 21: Dato subtarea A501

8.5.2. A502: Primer periodo de corrección de errores

Durante este periodo de corrección de errores se corregirán todos los errores detectados durante la subtarea A601. En el caso de las mejores sugeridas durante la tarea A601, se intentarán llevar a cabo de ser posibles.

Fecha de inicio	29 de junio de 2022
Fecha final	1 de julio de 2022
Duración(días)	3
Carga de trabajo(horas)	24
Participantes	PG

Tabla 22: Datos subtarea A502

8.5.3. A503: Periodo de pruebas en un entorno realista

Estas pruebas serán realizadas en un teatro junto al equipo técnico de Eskena, con el objetivo de conseguir una simulación lo más realista posible de una obra de teatro. Durante estas pruebas se detectarán posibles errores de compatibilidad de nuestro programa con el material o software que ellos disponen, como pueden ser micrófonos, puertos MIDI o el software QLAB. Además, se tendrán en cuenta posibles mejoras que puedan sugerir el equipo técnico de Eskena.

Fecha de inicio	4 de julio de 2022
Fecha final	5 de julio de 2022
Duración(días)	2
Carga de trabajo(horas)	16
Participantes	IS-PG-EK

Tabla 23: Datos subtask A503

8.5.4. A504: Corrección de errores

Durante este periodo de corrección de errores se corregirán todos los errores detectados durante la subtask A603. En el caso de las mejoras sugeridas durante la tarea A603, se intentarán llevar a cabo de ser posibles

Fecha de inicio	6 de julio de 2022
Fecha final	8 de julio de 2022
Duración(días)	3
Carga de trabajo(horas)	24
Participantes	PG

Tabla 24: Datos subtask A504

8.6. PT6: Documentación

La documentación es la última parte del proyecto, donde se escribirán todas las memorias e información requerida. Los informes a realizar son varios, tanto para la asociación Eskena, como para Gune+ como para nosotros mismos (AHOLAB).

Fecha de inicio	11 de julio de 2022
Fecha final	14 de julio de 2022
Duración(días)	4
Carga de trabajo(horas)	24
Participantes	IS-PG

Tabla 25: Datos tarea PT6

8.6.1. A601: Redacción de informes

Estos informes recogerán toda la información necesaria del proyecto. Además, se entregará a Eskena y Gune+ los informes que requieren.

Fecha de inicio	11 de julio de 2022
Fecha final	14 de julio de 2022
Duración(días)	2
Carga de trabajo(horas)	16
Participantes	IS-PG

Tabla 26: Datos subtarea A601

8.6.2. A602: Redacción manuales de usuario de la aplicación

Este manual servirá para recoger detalladamente cómo funciona el software, de esta manera cualquier nuevo usuario únicamente deberá leerlo para poder utilizar el programa. Además, se entregará una copia a la asociación Eskena.

Fecha de inicio	13 de julio de 2022
Fecha final	14 de julio de 2022
Duración(días)	2
Carga de trabajo(horas)	8
Participantes	PG

Tabla 27: Datos subtarea A602

8.7. Diagrama de GANTT

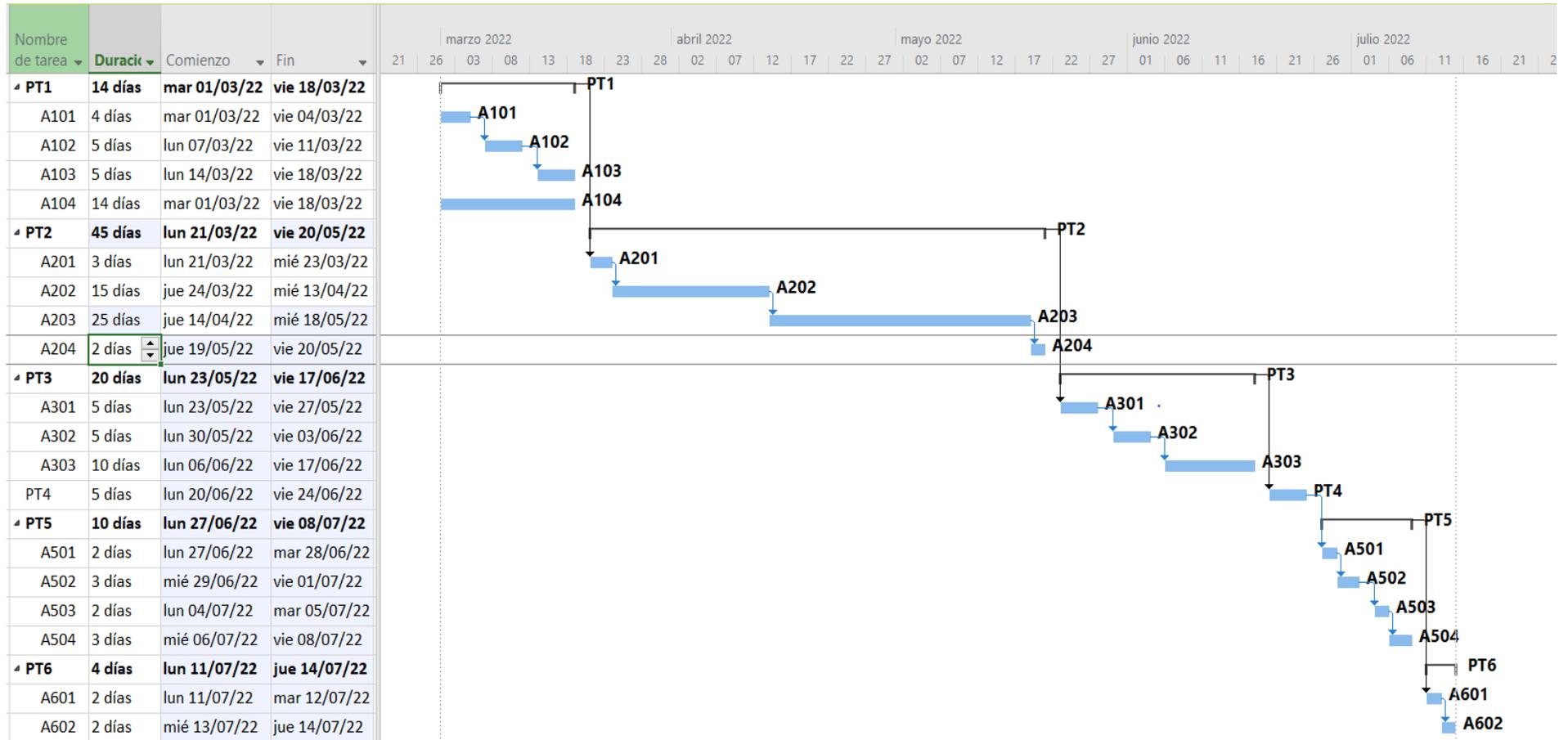


Tabla 28: Diagrama de GANTT

9. Presupuesto

En este apartado se detallará el presupuesto necesario para llevar a cabo todo el proyecto. Cabe destacar que este no ha sido un proyecto que haya requerido excesivos recursos materiales ya que la mayoría de él se ha podido realizar desde una computadora. Por lo que los recursos humanos copan la gran mayoría del presupuesto. A continuación, se va desgranar el presupuesto en diferentes secciones, para finalmente sumar cada una de ellas para lograr el coste total del proyecto.

9.1. Recursos humanos

Como recursos humanos se considera a las personas que han tomado parte en el proyecto. Anteriormente se ha mencionado que los participantes han sido Ibon Saratxaga, Peio Gonzalez y miembros del equipo Eskena. No obstante, para realizar el proyecto únicamente se van a tener en cuenta a los ingenieros Ibon Saratxaga y Peio Gonzalez, y no a los miembros de Eskena ya que han tenido una participación puntual y, además, ellos son los beneficiarios del proyecto. En la siguiente tabla se puede observar el coste de cada ingeniero, teniendo en cuenta el número de horas y la tasa horaria de cada uno de ellos.

	Puesto	Tasa horaria($\frac{\text{€}}{h}$)	Número de horas(h)	Total (€)
IS	Ingeniero senior	15	200	2500
ID	Ingeniero junior	10	600	6000
Total				8500

Tabla 29: Gasto en recursos humanos del proyecto

9.2. Amortizaciones

Las amortizaciones tienen en cuenta los recursos materiales que se han usado en el proyecto y que a su vez tienen una duración de vida larga. Dentro de estos recursos se pueden incluir tanto maquinaria, equipamientos, programas software como hardware específico. Para este proyecto se han utilizado dos tipos de equipamiento que requieren ser amortizados:

- Ordenador portátil: ordenador Huawei con un valor de 800 euros con una vida útil estimada de 5 años. Por lo que, si el proyecto dura 5 meses el coste ha sido de 67 euros
- PC de sobremesa AHOLAB: PC que se encuentra en el centro de investigación de AHOLAB el cual tiene un valor de 2000 euros con una vida útil de 5 años. El coste para este proyecto ha sido de 167 euros
- Interfaz MIDI: para poder realizar las diferentes pruebas durante el proyecto ha sido necesario adquirir un conector con diferentes puertos MIDI del modelo Behringer U-PHORIA UMC404HD. El valor de este aparato ha sido de 170 euros con una vida útil de 3 años. Por lo tanto, el coste ha sido únicamente de 7 euros.
- Receptores inalámbricos: estos receptores serán los encargados de enviar la señal de audio desde los micrófonos al punto que se desee. Para este proyecto se han utilizado los Evolution Wireless G3 SENNGEISSER, los cuales tiene un valor de 500 euros cada uno. Se han utilizado tres de ellos, y cada uno tiene una vida útil de 10 años

	Vida útil(años)	Coste (€)	Tiempo utilizado (meses)	Utilización simultanea	Total (€)
Ordenador portatil	5	2000	5	1	67
PC sobremesa AHOLAB	5	700	5	1	167
Puertos MIDI	3	170	5	1	24
Receptores inalámbricos	10	1500	5	1	63
Total					331

Tabla 30: Amortizaciones del proyecto

9.3. Gastos

En este apartado se incluyen las compras realizadas durante el proyecto y que han sido de uso exclusivo del proyecto, por lo que no se pueden incluir en la sección de amortizaciones. En este caso se han realizado dos gastos:

- Licencia Office: con la cual se han redactado todos los documentos necesarios.
- Micrófonos: para poder realizar todas las pruebas necesarias durante el proyecto ha sido necesario la compra de 3 micrófonos DPA Microphones, cada uno por un valor de 180 euros.

	Valor (€)
Licencia Office	46
Micrófonos	540
Total	586

Tabla 31: Gastos del proyecto

9.4. Otros

Para finalizar el presupuesto, se va añadir un último concepto al presupuesto llamado costes no directos que va ser el 10% de los costes directos (la suma de los anteriores apartados). A través de este apartado se representará el coste del consumo eléctrico de los aparatos, el mantenimiento de las instalaciones, la conexión a internet...

En la siguiente tabla se puede observar el coste total del proyecto teniendo en cuenta las anteriores secciones:

Concepto	Valor (€)
Recursos humanos	8500
Amortización	331
Gastos	586
Costes directos	9675
Costes indirectos	967
Total	10648

Tabla 32: Resumen del presupuesto

10. Conclusiones

A modo de cierre de este trabajo, se puede señalar que se ha logrado un regidor automático capaz de indicar en todo momento donde está la obra. Es por ello por lo que se ha conseguido completar el objetivo principal del proyecto. Más si cabe, y a pesar que durante el transcurso del mismo han surgido infinidad de problemas y contratiempos, se han conseguido mejorar notablemente los objetivos previos, ya que durante el transcurso del proyecto se han ido introduciendo mejoras que aumentan las funcionalidades y capacidades del programa.

La parte más visible del software AhoStage es su interfaz gráfico, por ello se han puesto todos los esfuerzos en diseñarlo lo mejor posible. Finalmente, como se ha podido ver en apartados anteriores, se ha conseguido un interfaz para el usuario muy simple, de forma que pueda familiarizarse con él rápidamente. No obstante, aunque el interfaz sea sencillo, se ha logrado que reúna todas las funcionalidades necesarias para un técnico de una obra de teatro.

Además del interfaz gráfico, otro de los puntos importantes del proyecto era la implementación de MIDI como método de comunicación entre AhoStage y las aplicaciones de control escénicas. Tras las diferentes pruebas realizadas durante el proyecto se puede concluir que es uno de los mejores métodos de comunicación ya que es un lenguaje bastante completo y eficiente, capaz de poder enviar toda la información necesaria utilizando los mínimos recursos posibles. Utilizando unos pocos bytes tiene la habilidad de controlar todo el sistema escénico de una obra de teatro, además de ser extremadamente rápido, y, como ha quedado demostrado, tiene una alta compatibilidad con la mayoría de las aplicaciones.

Como se ha mencionado anteriormente, durante el transcurso del proyecto han surgido una infinidad de contratiempos y problemas, la gran mayoría de ellos relacionados con la programación, ya que durante algunos tramos se requerían grandes conocimientos. En la implementación del módulo MIDI también surgieron problemas al ser una tecnología relativamente nueva para el equipo de trabajo. Pero ante estas adversidades, se ha sabido cómo reaccionar y poder resolver cada uno de los problemas. Es por ello, que este proyecto ha servido para adquirir una infinidad de nuevos conocimientos, y, sobre todo, ha servido para aprender a sobreponerse a cada contratiempo.

En definitiva, se puede concluir que el proyecto se ha realizado exitosamente. Se espera que los asistentes a las obras de teatro queden satisfechos con el resultado y puedan disfrutar de todas las nuevas implementaciones que incluye AhoStage.

11. Anexo I: Manual de usuario



AhoStage: Diseño y desarrollo de un regidor de escena automático

Entregable E4: Manual de usuario

Julio 2022

ahō LAB

Aholab Signal Processing Laboratory

Faculty of Engineering of Bilbao

University of the Basque Country (UPV/EHU)

Índice

Introducción	68
Cargado de un guion y reproducción de la obra	70
Paso 1: Apertura de dialogo de selección de guion	70
Paso 2: Selección de guion	71
Paso 3: Configuración entrada de audio	72
Paso 4: Configuración de salida de audio.....	74
Paso 5: Comenzar la obra	75
Control de reproducción	76
Botones de avance (>>) y retroceso (<<).....	77
Botón Pause/Play.....	77
Botón stop	77
Botones Mute In y Mute Out.....	77
Edición de guion	78
Editar línea.....	79
Añadir línea.....	79
Eliminar línea	80
Guarda los cambios en un archivo	81
Paso 1: Apertura de dialogo de guardado de guion.....	81
Paso 2: Guardar guion	82

Ilustraciones

Ilustración 1: Pantalla inicio AhoStage.....	¡Error! Marcador no definido.
Ilustración 2: Botón cargar guion.....	71
Ilustración 3: Elección de guion	71
Ilustración 4: Configuración de entrada de audio	72
Ilustración 5: Asignación de personajes y micrófonos.....	73
Ilustración 6: Configuración de entrada de audio completada	73
Ilustración 7: Botón de configuración de salida de audio	74
Ilustración 8: Formulario de configuración de salida de audio	75
Ilustración 9: Botón de comienzo de obra	76
Ilustración 10: Botones de secuenciación y control	77
Ilustración 11: Menú de edición	78
Ilustración 12: Formulario edición de guion.....	79
Ilustración 13: Formulario añadir línea.....	80



Universidad del País Vasco Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA ESCUELA DE INGENIERÍA DE BILBAO

Ilustración 14: Formulario eliminar línea	81
Ilustración 15: Botón guardar guion	81
Ilustración 16: Guardado de guion	82

Introducción

Este documento presenta una guía para adquirir los conocimientos básicos para poder hacer uso del software AhoStage. Las indicaciones se harán de una forma para que cualquier usuario pueda entenderlas de una manera rápida y sencilla. Estas son las funciones que se detallarán en el documento: cargado de un guion y comienzo de reproducción de la obra, edición de un guion y el control de reproducción de una obra.

Como punto de partida para cada la explicación de cada una de las funciones se tomará la pantalla principal del software, la cual carga nada más inicializar el programa.

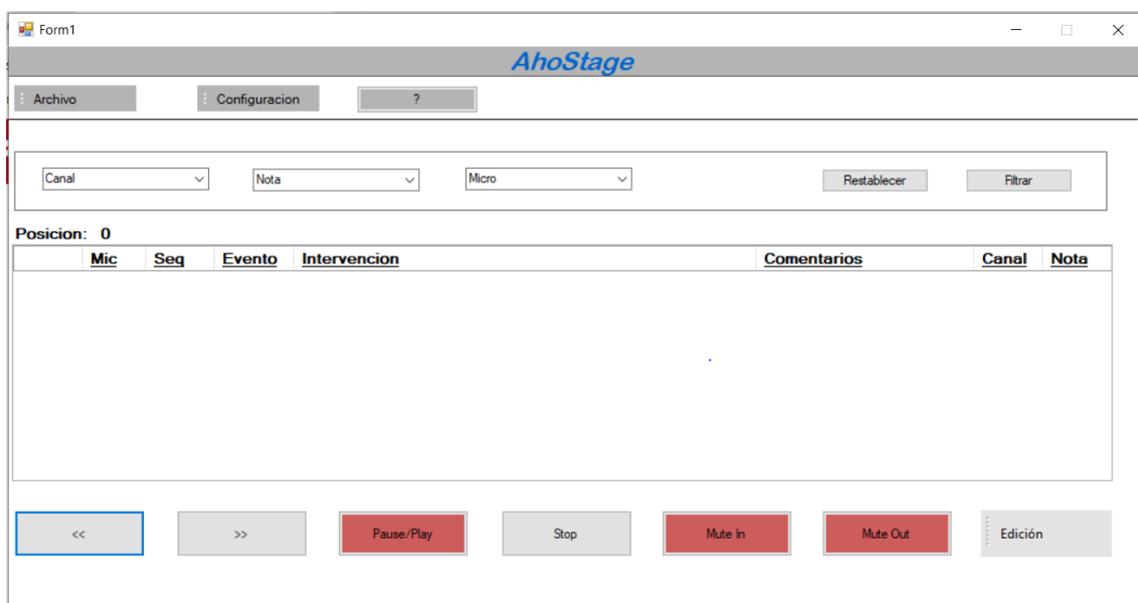


Ilustración 34: Pantalla inicio AhoStage

Cargado de un guion y reproducción de la obra

La función principal de AhoStage es la ejecución y reproducción de una obra. Durante este punto se explicará como cargar una obra al programa, realizar la configuración básica y ejecutarla. Los pasos a seguir son los siguientes:

Paso 1: Apertura de dialogo de selección de guion

Primeramente, se debe cargar el dialogo para poder elegir el guion a reproducir entre los archivos de la computadora. Para ello, se debe desplegar el menú archivo que se encuentra

arriba a la derecha de la pantalla principal. Una vez desplegado, se selecciona la opción cargar guion para abrir el dialogo.

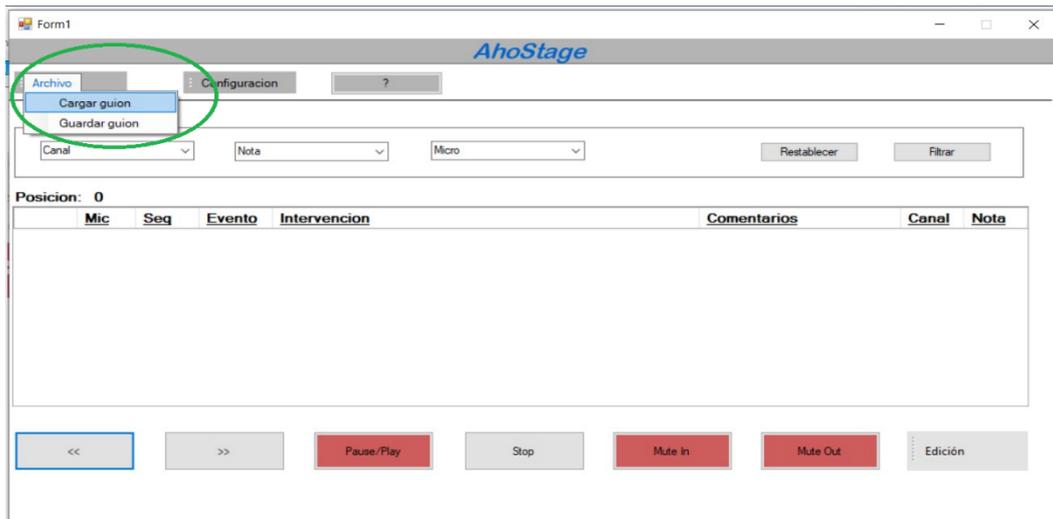


Ilustración 35: Botón cargar guion

Paso 2: Selección de guion

Una vez abierto el dialogo de selección de guion, se debe escoger el guion a reproducir, el cual debe estar en formato .txt. Para facilitar esta elección, el programa realiza un filtrado automático de archivos .txt para no mostrar en pantalla otro tipo de archivos.

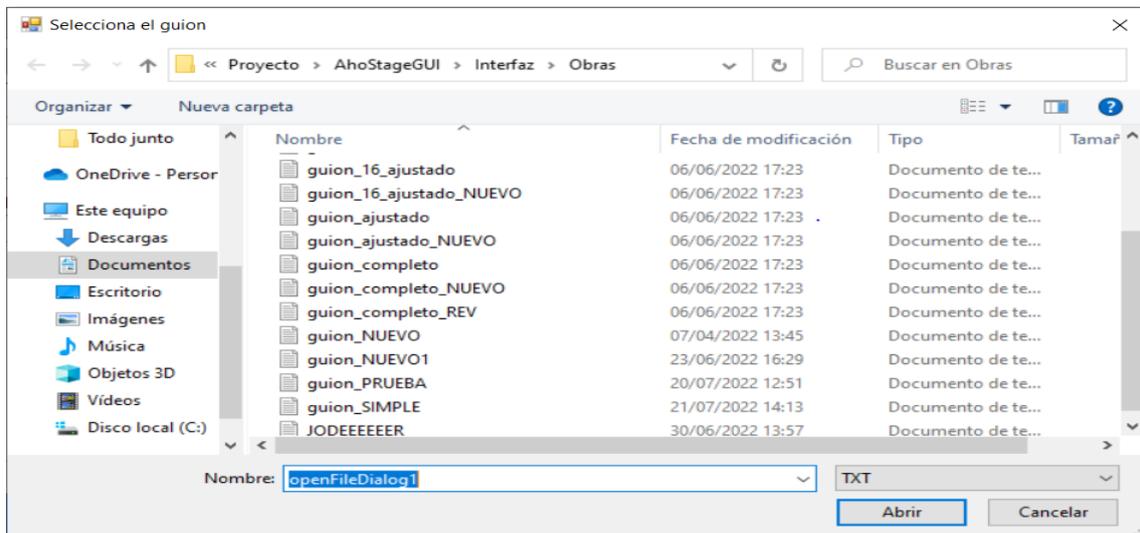


Ilustración 36: Elección de guion

Paso 3: Configuración entrada de audio

Cuando se carga un guion, automáticamente se carga la pestaña de configuración de entrada de audio, la cual tiene el siguiente formato:

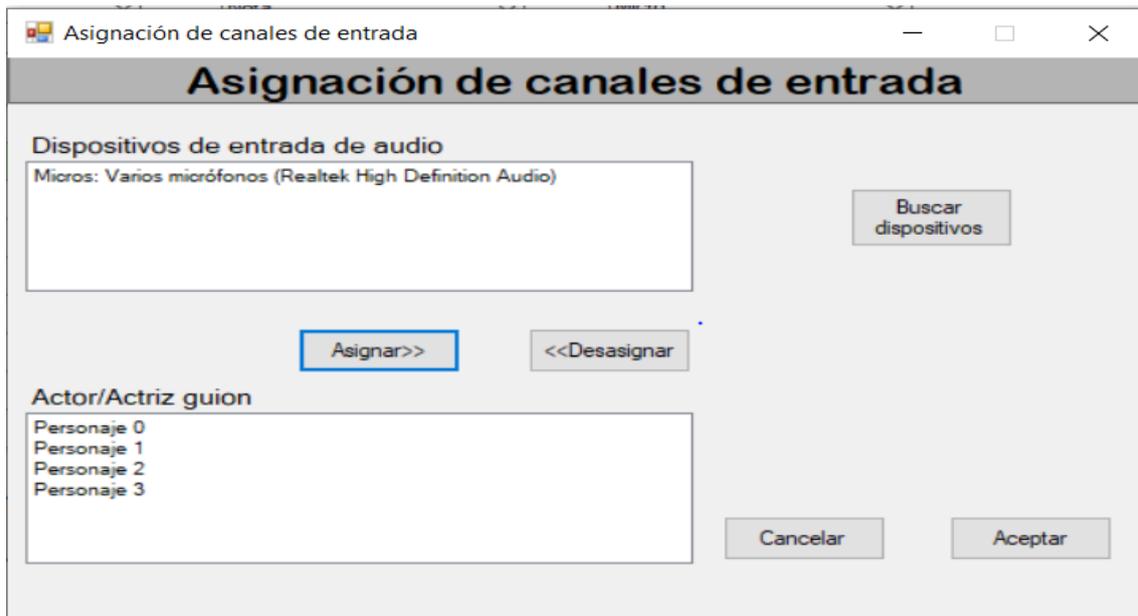


Ilustración 37: Configuración de entrada de audio

En esta pantalla, en el cuadro superior se muestran las entradas de audio disponibles detectadas por el sistema y en el cuadro inferior se muestran los personajes que componen la obra. Para poder comenzar la obra, todos los personajes deben tener un dispositivo asignado. Para unir un personaje con un micrófono, se debe seleccionar al mismo tiempo el personaje y micrófono que se quieren relacionar entre sí, y a continuación, pulsar el botón “asignar” que se encuentra en el centro de la pantalla.

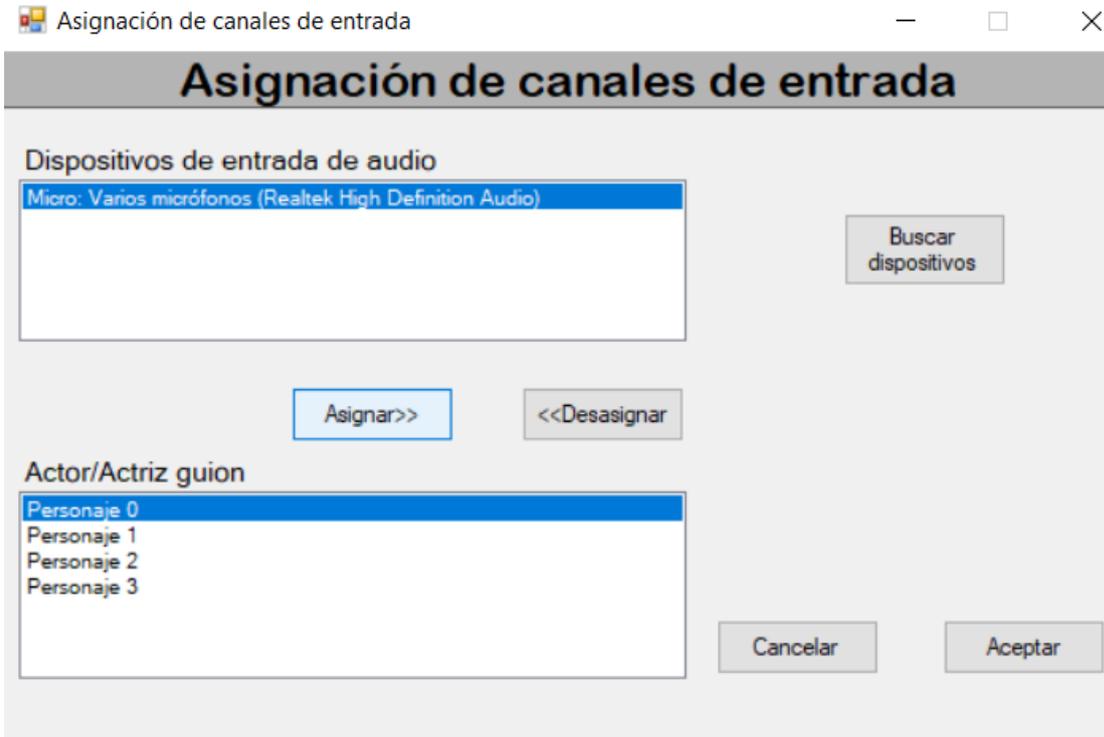


Ilustración 38: Asignación de personajes y micrófonos

Cuando todos los personajes tengan asignado un micrófono la pantalla tendría un aspecto similar al de la siguiente imagen:



Ilustración 39: Configuración de entrada de audio completada

En caso de querer romper una relación entre micrófono y personaje, únicamente se debe seleccionar en el cuadro de abajo y pulsar el botón “desasignar”.

Para terminar el proceso, y cerrar la pestaña, se pulsa el botón “Aceptar” para guardar los cambios en el sistema.

Paso 4: Configuración de salida de audio

Tras configurar y aceptar la entrada de audio, el programa vuelve a la pantalla de inicio. Tras esto, se debe configurar la salida de audio, es decir elegir el puerto por el que se van a lanzar las correspondientes notas MIDI. Para ello es necesario abrir el formulario de configuración de puerto de salida de audio. Para poder abrirlo, en el formulario principal, se debe desplegar el menú de configuración y seleccionar la opción “MIDIOut”.

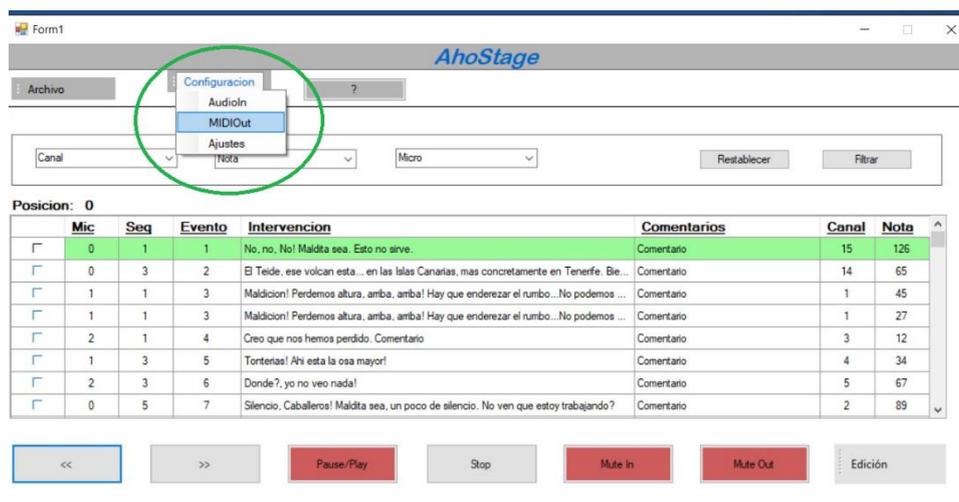


Ilustración 40: Botón de configuración de salida de audio

Una vez clicado en “MIDIOut” se abre el formulario de configuración de salida de audio, el cual tiene el siguiente formato:



Ilustración 41: Formulario de configuración de salida de audio

En el cuadro central del formulario se muestran todos los puertos de salida MIDI que ha detectado el sistema, entre los cuales se debe elegir uno. Para ello, con el ratón se selecciona el puerto que se desea utilizar y a continuación se pulsa el botón “Aceptar” para guardar la elección. De esta forma, queda configurada la salida MIDI y el software vuelve a la pantalla de inicio.

Paso 5: Comenzar la obra

Una vez llegado a este punto, estando todo correctamente configurado, únicamente queda dar la orden para comenzar la obra. En la pantalla de inicio de AhoStage, en la parte inferior se encuentra el botón “Pause/Play”, mediante el cual se da comienzo o se para la obra. En caso de que la obra se encuentre parada, el botón tendrá un color rojo, y en caso de que se esté reproduciendo tendrá un color verde.

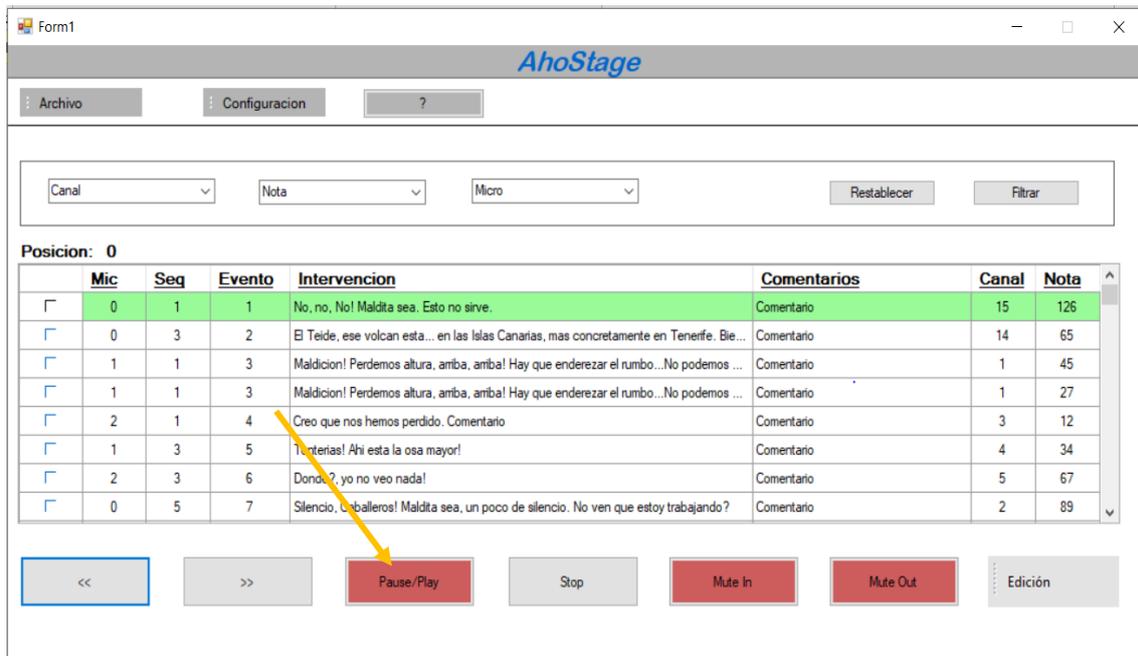


Ilustración 42: Botón de comienzo de obra

De esta forma, da comienzo la obra.

Control de reproducción

Relacionado con el punto anterior, ahora se van exponer las diferentes opciones que tiene el técnico de la obra para tener el control de la obra. Hay que recordar que cuando se pulsa el botón play en AhoStage el sistema es capaz de detectar en todo momento en que punto del guion se encuentra, y cuando detecta un cambio en el guion avanza el secuenciador. El control de reproducción se hace mediante los botones de secuenciación y de control.

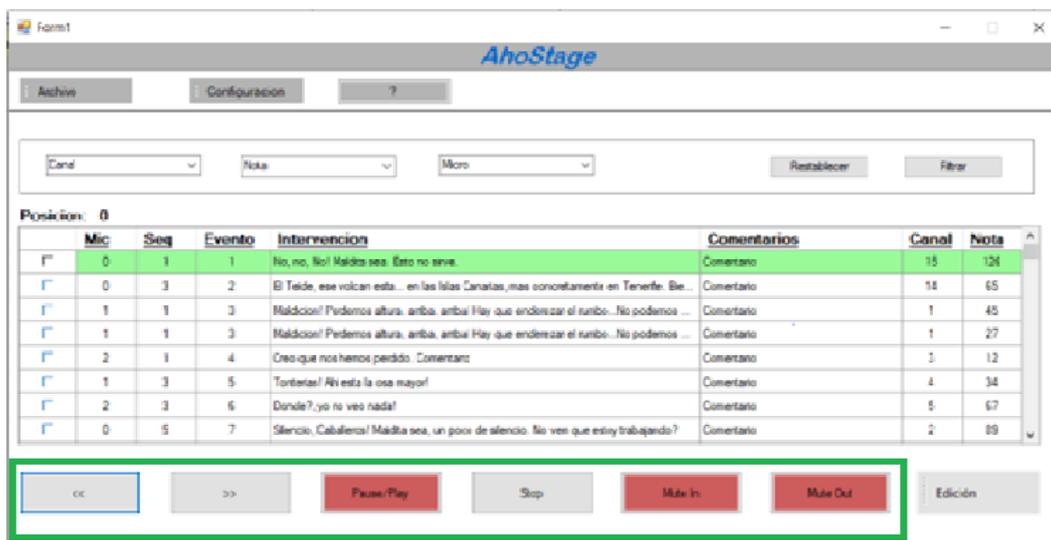


Ilustración 43: Botones de secuenciación y control

Botones de avance (>>) y retroceso (<<)

Mediante estos dos botones se avanza o retrocede en el guion línea a línea, únicamente pulsando en ellos, en caso de que el usuario necesite cambiar de posición en el guion.

Botón Pause/Play

Este botón pausará y continuará la obra, dependiendo la situación anterior, se estaba reproduciendo lo pausará y estaba en pausa continuará la obra. En caso de estar en verde significará que está en reproducción la obra y en caso de encontrarse en rojo significará que está en pausa.

Botón stop

Finaliza la obra.

Botones Mute In y Mute Out

Estos botones tienen el control sobre la relación entre la entrada y salida de audio y el secuenciador. A través de Mute In, se le indicará al secuenciador si debe de hacer caso a los

micrófonos de entrada o no. Y a través de Mute Out, se le indicará al secuenciador si debe o no lanzar notas MIDI por el puerto de salida cuando detecten un cambio en el guion.

En caso de que un botón se encuentre en color rojo significará que esa opción esta deshabilitada y en caso de que se encuentre en color verde querrá decir que se encuentra habilitada.

Edición de guion

Otra de las funcionalidades más importantes de AhoStage es la posibilidad de edición de guion. El usuario dispone de tres posibilidades: editar una línea ya creada del guion, añadir una nueva línea y eliminar una línea existente del guion.

Para tener la posibilidad de editar el guion, se debe tener cargado un guion previamente. Por ello, los primeros dos pasos a seguir son exactamente los dos primeros de la función anterior “cargado de guion y reproducción de obra”. Es decir, se deben seguir los pasos 1 “Apertura de dialogo de selección de guion” y 2 “selección de guion”. De esta manera, ya se tendría disponible un guion para poder editarlo.

Para escoger la forma de editar el guion, se debe desplegar el menú “Edición” que se encuentra en la parte inferior derecha de la pantalla. Una vez desplegado, se muestran las tres opciones que ofrece el programa y tras pinchar en una de ellas se abre el formulario de edición de guion:

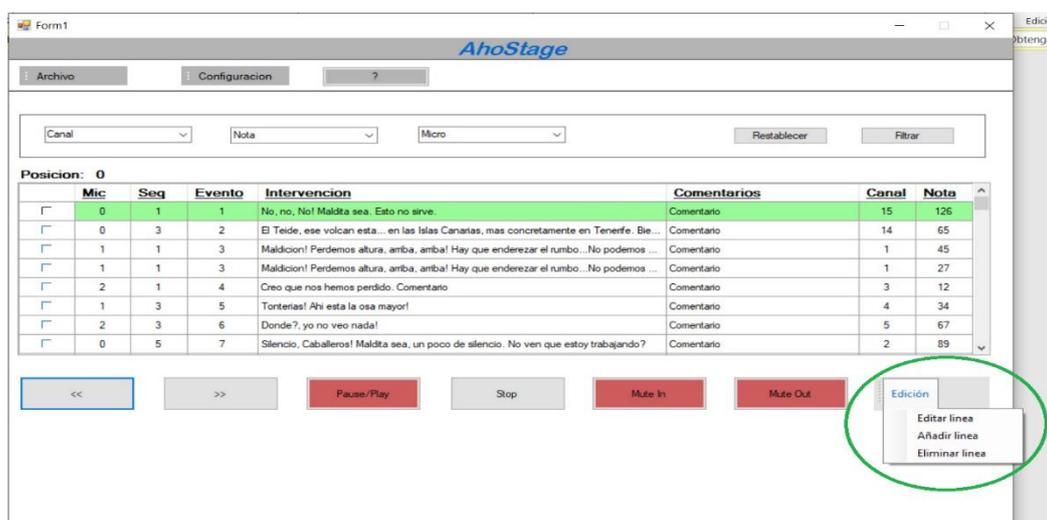


Ilustración 44: Menú de edición

Editar línea

Para que se abra este formulario, además de pulsar la opción “Editar línea”, se debe haber seleccionado la línea que se desea editar mediante el checkbox de la pantalla principal.

<u>Mic</u>	<u>Seq</u>	<u>Evento</u>	<u>Dialogo</u>	<u>Comentario</u>	<u>Canal</u>	<u>Nota</u>
0	1	1	No, no, No! Maldita sea. Esto no sirve.	Comentario	15	126

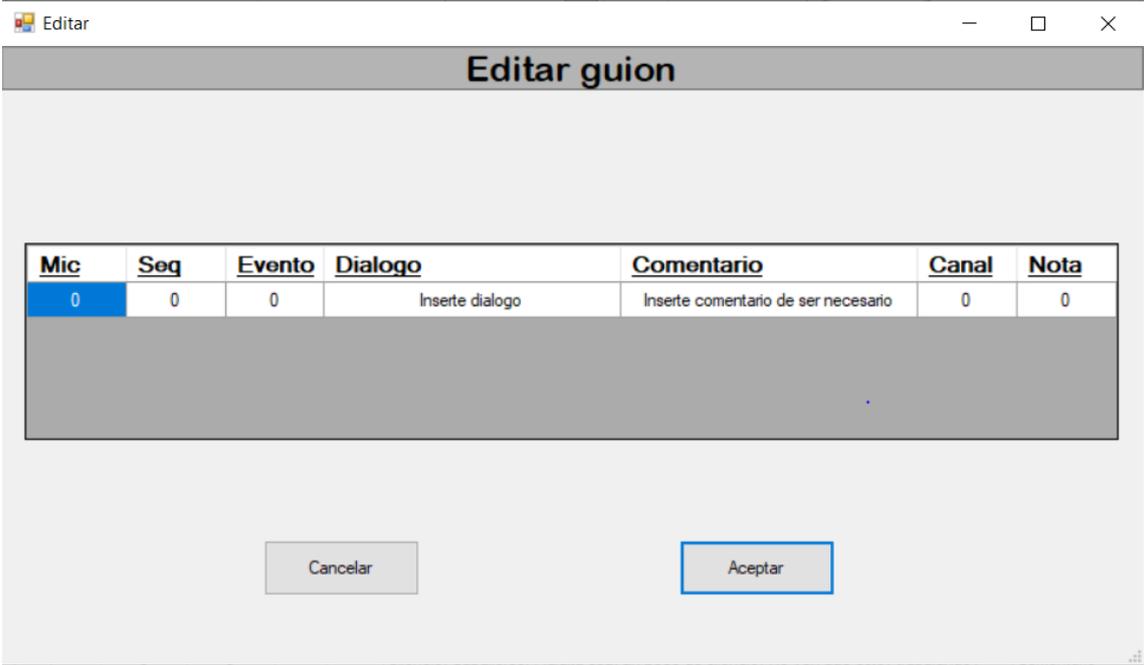
Cancelar Aceptar

Ilustración 45: Formulario edición de guion

En la imagen anterior, se puede observar la pantalla de edición de guion, en la cual se muestra la línea escogida. Para editar el campo de alguna de las casillas, se debe pinchar sobre ella y modificarla mediante el teclado. Para de querer guardar los cambios, se pulsará el botón “Aceptar”. En caso de que alguna modificación sea incorrecta, el software lo notificará mediante una pantalla emergente y se descartarán los cambios.

Añadir línea

En este caso, se abrirá la pantalla de edición de guion, pero con la línea totalmente vacía para que el usuario introduzca los datos que desee.



<u>Mic</u>	<u>Seg</u>	<u>Evento</u>	<u>Dialogo</u>	<u>Comentario</u>	<u>Canal</u>	<u>Nota</u>
0	0	0	Inserte dialogo	Inserte comentario de ser necesario	0	0

Ilustración 46: Formulario añadir línea

Como en el caso anterior, para guardar los cambios se pulsará el botón “Aceptar”. En caso de que alguna modificación sea incorrecta, el software lo notificará mediante una pantalla emergente y se descartarán los cambios.

Eliminar línea

Para poder eliminar una línea, antes de seleccionar la opción “Eliminar línea” se debe elegir qué línea se desea eliminar mediante el CheckBox de la pantalla principal. Tras ello se abrirá el formulario de edición de guion.



Ilustración 47: Formulario eliminar línea

Si definitivamente se quiere eliminar la línea, se debe pulsar el botón “Aceptar”.

Guarda los cambios en un archivo

En caso de querer guardar los cambios realizados en el guion en un archivo de texto, se deben seguir los siguientes pasos:

Paso 1: Apertura de dialogo de guardado de guion

Lo primero es abrir el dialogo de guardado de guion, para ello se debe seleccionar el menú desplegable “Archivo” y elegir la opción “Guardar guion”.

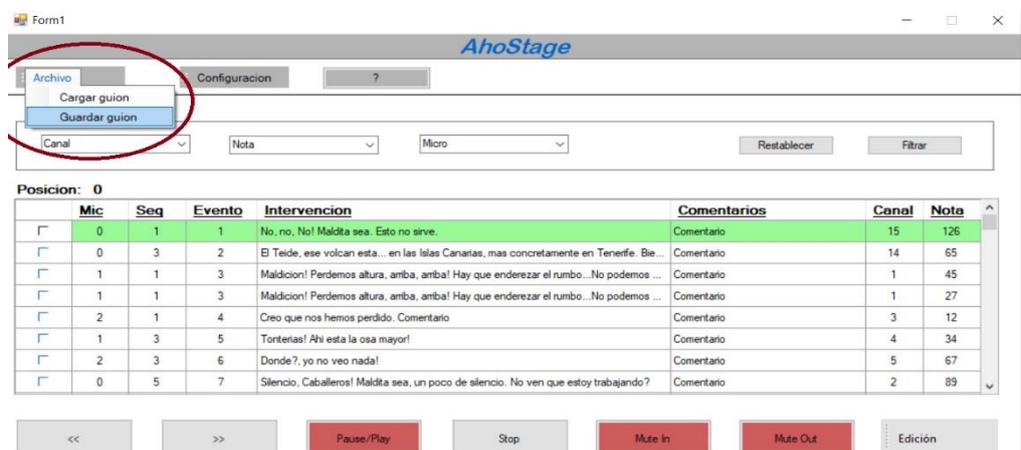


Ilustración 48: Botón guardar guion

Paso 2: Guardar guion

Inmediatamente, se abrirá el diálogo para poder elegir el lugar donde se desea guardar el guion. Para ello se debe seleccionar el directorio e introducir el nombre que se quiera. El archivo se guardará automáticamente en formato .txt.

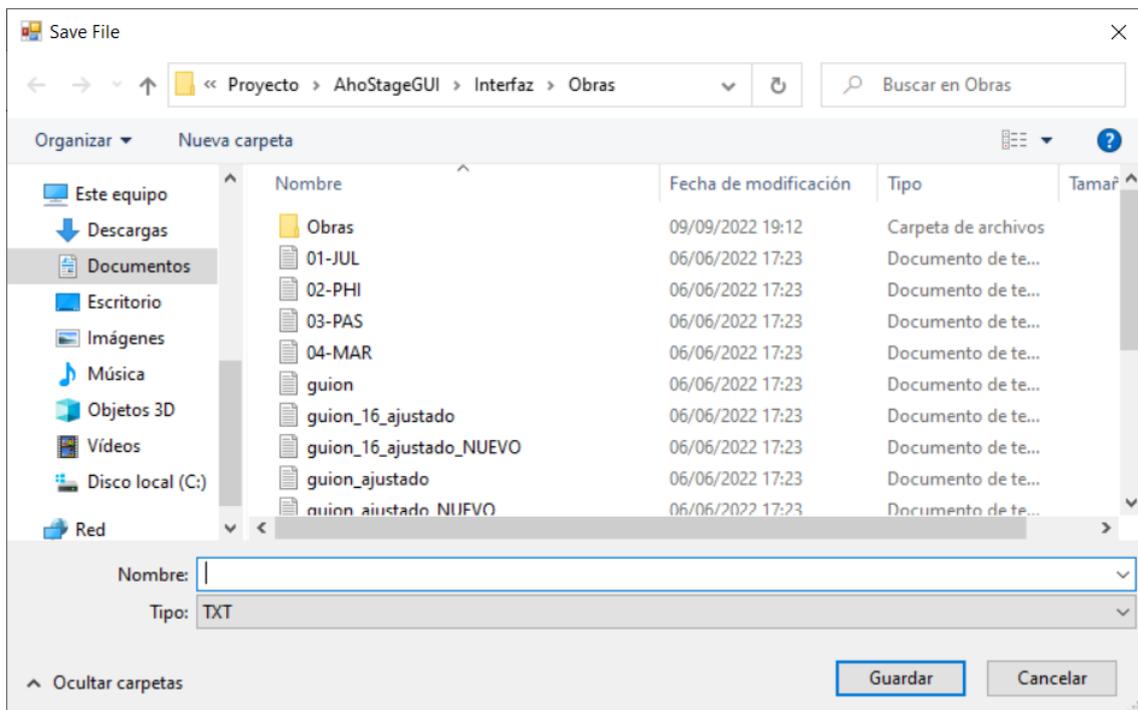


Ilustración 49: Guardado de guion

12. Anexo II: Código

El objetivo principal de este anexo es documentar el código desarrollado durante el proyecto, con el fin de tener una visión general de AhoStage desde un punto de vista más técnico. Aun así, no se entrará en detallar minuciosamente el código ni se insertará el mismo en el documento, sino que se dará una explicación global para conocer los diferentes módulos y archivos que componen el programa y qué relación tiene entre ellos.

En primero lugar, cabe recordar que el software ha sido desarrollado mediante la plataforma de Visual Studio, la cual ofrece una multitud de ventajas y facilidades, y que el lenguaje escogido ha sido C++. En segundo lugar, hay que tener en cuenta que para el diseño del interfaz gráfico se ha hecho uso de la programación de forma gráfica, es decir, son una serie de archivos los cuales se pueden modificar insertando o eliminando elementos (menús, tablas, labels, botones...) de forma dinámica, sin necesidad de escribir el código manualmente.

Para poder entender correctamente el funcionamiento de AhoStage, lo primero de todo será conocer los tipos de archivos o ficheros con los que se ha trabajado, y saber cuáles son cada uno de ellos. A continuación, se analizará la relación que tienen entre ellos para el correcto funcionamiento de AhoStage.

Archivos que componen AhoStage

En este subapartado se van a conocer los archivos que componen el programa tal y como se puede observar en la siguiente imagen. Principalmente existen tres tipos de archivos diferentes:

- .cpp: contienen código fuente escrito en el lenguaje de programación orientado a objetos C++. Como la información dentro del archivo CPP se almacena en forma de texto sin formato, los archivos pueden abrirse con cualquier editor de texto. Estos archivos guardarán el código necesario para el funcionamiento del programa.
- .h: son generalmente los archivos de encabezado utilizados con los lenguajes de programación C ++. Pueden consistir en definiciones de variables externas, prototipos de funciones y constantes. En este proyecto, tienen varias utilidades, como puede ser la programación del interfaz gráfico dinámicamente.
- .resx: son archivos XML que son utilizados para la versión .NET, la cual no es de interés en este proyecto.

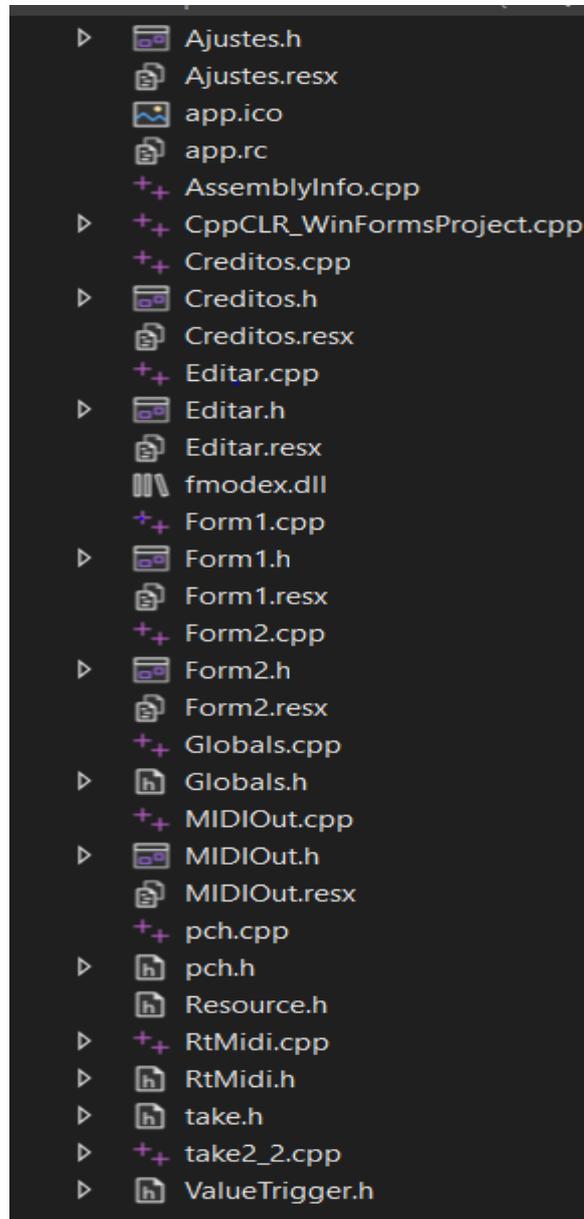


Ilustración 50: Archivos de AhoStage

Atendiendo a la imagen anterior, cada uno de los archivos se puede clasificar en una función de archivo diferente. Es decir, cada uno de ellos realiza una función diferente dentro del código. De esta manera, se pueden clasificar dentro de una de estas cinco funciones:

- Formularios: estos archivos son el código de un formulario diferente. Cada formulario consta de tres archivos un .h, un .resx y un .cpp. Con el conjunto de esos tres archivos se forma un formulario como son Form1, Form2, Editar, MIDIOut, ajustes y créditos
- Variables globales: es un archivo .cpp y .h que guarda las declaraciones de las variables globales del programa. Su nombre es Globals.

- Módulo RtMidi: módulo compuesto por un .cpp y .h que sirve para implementar el estándar MIDI en AhoStage. Su nombre es RtMidi.
- Módulo de funciones: otro módulo compuesto por un .cpp y .h, que contiene las funciones necesarias para el correcto funcionamiento de AhoStage. El .h tiene el nombre de take y el .cpp el de take2_2.
- Main: contiene el main del programa, con las correspondientes inicializaciones y declaraciones. Es un archivo .cpp con nombre CppCLR_WinFormsProject.

En la siguiente imagen, se puede observar un diagrama que representa la relación que tienen los diferentes archivos entre sí.

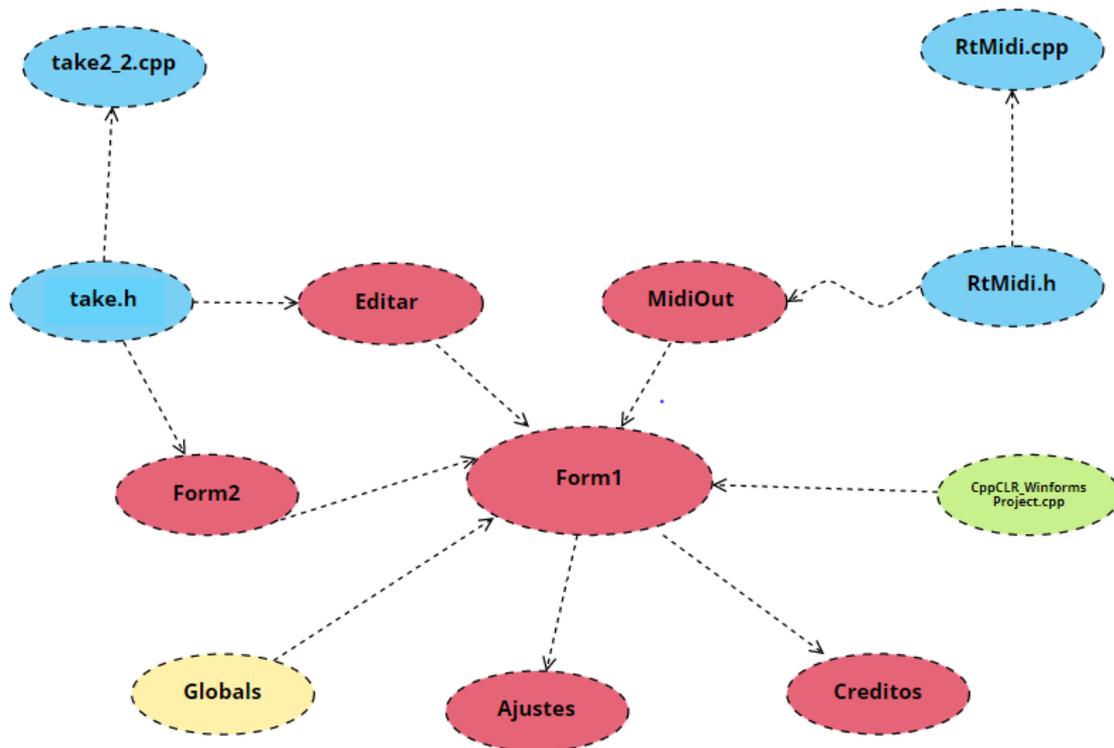


Ilustración 51: Esquema de la organización del código de AhoStage

A continuación, se describirá la composición de cada uno de los archivos explicando su función y enumerando las diferentes funciones que lo componen.

Main

El main se ubica en el archivo CppCLR_WinformsProject.cpp, su función principal es cargar el formulario de inicio, que en este caso es el Form1. Además de ello, declara el namespace System y que se va a hacer uso de threads. El código es muy corto y sencillo.

```
1  #include "pch.h"
2  #include "Form1.h"
3
4  using namespace System::Windows::Forms;
5  using namespace System;
6  [STAThread]
7  int main()
8  {
9      Application::EnableVisualStyles();
10     Application::SetCompatibleTextRenderingDefault(false);
11     Application::Run(gcnew CppCLRWinFormsProject::Form1());
12     return 0;
13 }
```

Ilustración 52: Main de AhoStage

Variables globales

Este archivo contiene las declaraciones de las variables globales de todo el programa. Pueden ser utilizadas desde cualquier otro archivo del programa. Gracias a esto, se pueden modificar el contenido de esta variable de forma sencilla. Cabe destacar que únicamente contiene las declaraciones, no las definiciones.

Form1

Este formulario contiene la información principal del formulario inicial. No obstante, se podría dividir en tres partes diferentes:

- Definición de variables globales: recoge las definiciones de las variables globales declaradas en Globals.h.
- Clase Form 1: en esta clase se encuentra el código que crea y desarrolla todo el funcionamiento necesario para el formulario de inicio.
- Bucle principal: contiene un thread que se lanza nada más comenzar a reproducirse una obra. De esta manera, se consigue que puedan trabajar paralelamente tanto el interfaz gráfico como el secuenciado automático.

Otros formularios

La información de los demás formularios de AhoStage quedan guardada en archivos específicos:

- Form 2: guarda la información del formulario de configuración de entrada de audio.
- Ajustes: contiene la información del formulario de ajustes.
- Editar: este archivo tiene la información del formulario para poder editar el guio tanto como para modificarlo como para añadir línea como para eliminar línea.
- MIDIOut: es el archivo que guarda la información del formulario de configuración de salida de audio.
- Creditos: contiene los créditos del software.

RtMIDI

En los archivos RtMidi.h y RtMidi.cpp está el código relacionado con MIDI. El archivo .h contiene las declaraciones de las funciones y en .cpp las definiciones de las funciones.

Take

En estos archivos se guardan las funciones principales de AhoStage. Entre las más destacadas están las de carga de guion, las de edición de guion, las de configuración... Estas funciones están declaradas y definidas en estos dos archivos, pero son utilizadas o llamadas por el interfaz gráfico.