



Universidad
Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

PROYECTO FIN DE CARRERA

Editor de Reproductores Multimedia:

Easy Player Editor

Autor: Héctor Monsalve Toledo

Tutor: María González García

Titulación: Ingeniería Técnica en Informática de Gestión

Leganés, Septiembre de 2015

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto fin de Carrera en Leganés el martes 29 de Septiembre de 2015, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la calificación de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Amigo lector, si por alguna razón estás leyendo estas breves líneas, o bien eres familia o amigo, o bien estás buscando información para tu propio proyecto. Si éste último es el caso, permíteme ofrecerte un sabio consejo: “acaba tu Proyecto Fin de Carrera antes de entrar al mundo laboral”.

Largo y arduo ha sido el camino que he tenido que recorrer para completar este trabajo, y aunque creo que de una forma u otra lo habría terminado, no habría sido lo mismo sin el apoyo, y los constantes y pesados (pero sólo un poquito) recordatorios de mi familia y mis amigos.

En especial he de agradecer a mis padres Luis y Julia y a mi tía Emi, que siempre me han apoyado en todos los momentos de mi vida.

A mi hermano, que aunque el tío tarde en ducharse por las mañanas medio siglo, es un tío cojonudo.

A todos mis amigos que siempre me apoyan, en especial Paco y David, habéis sido pesados de cojones con mi proyecto, pero siempre que se os necesita estáis ahí, tengo una gran suerte de tener unos amigos como vosotros.

A mi pareja y diseñadora gráfica personal Laura, gracias por todo el apoyo que me das siempre. Y gracias por los iconos que has diseñado para esta aplicación, son Mantequilla de Nápoles.

Y me gustaría agradecer el especial apoyo recibido por parte de dos personas de la Universidad que me propusieron este proyecto, y sin las que el mismo no habría sido posible: María y Julián.

¡Gracias a todos!

Resumen

En este proyecto fin de carrera se ha realizado un editor de reproductores multimedia que ofrezca características de accesibilidad, capaz de generar reproductores multimedia personalizados en función de las necesidades del usuario.

Con este objetivo, se ha realizado un estudio previo del estado actual de las tecnologías necesarias para la composición de la aplicación. A partir de los conocimientos adquiridos, se ha diseñado y desarrollado un software (programa informático) que cumple con los requisitos establecidos en el planteamiento inicial.

Posteriormente, se ha realizado la implementación de la aplicación y se ha evaluado detalladamente su funcionamiento, para comprobar que el resultado obtenido se ajusta a las necesidades iniciales.

La aplicación se ha diseñado para sistemas Windows de 64 bits, pero como se verá en capítulos posteriores, dicha limitación podría resolverse con la implementación de algunas mejoras especificadas en las líneas futuras propuestas.

Los resultados obtenidos en la fase de pruebas muestran como la aplicación es capaz de generar reproductores multimedia con funcionalidad completa, totalmente personalizables en la funcionalidad ofrecida al usuario final, así como capaces de reproducir todo tipo de vídeos multimedia con diferentes formatos en las pistas de vídeo, audio, o subtítulos.

Índice de Contenidos

Agradecimientos	V
Resumen	VII
Índice de Contenidos	IX
Índice de Ilustraciones	XIII
Índice de Tablas	XVII
1. Introducción	3
1.1 <i>Motivación del Proyecto</i>	3
1.2 <i>Ámbito del Sistema</i>	4
1.3 <i>Objetivos</i>	4
1.4 <i>Estructura del Documento</i>	5
2. Estado del Arte	9
2.1 <i>Tecnologías</i>	9
2.1.1 Java	9
2.2 <i>Frameworks</i>	11
2.2.1 Swing	11
2.2.2 AWT	12
2.2.3 Java Media Framework	13
2.2.4 JavaFX	13
2.2.5 VLCJ	14
2.3 <i>Estudio de Accesibilidad</i>	16
2.3.1 Tipos de discapacidad	17
2.3.2 Estándares de Accesibilidad	22
2.3.3 Requisitos de accesibilidad de un reproductor multimedia	23
2.4 <i>Trabajos relativos</i>	24
2.5 <i>Discusión del Estado de Arte</i>	25
3. Gestión del Proyecto	28
3.1 <i>Descomposición en Tareas</i>	28
3.2 <i>Planificación del Proyecto</i>	29
3.3 <i>Diagrama de Gantt</i>	30
3.4 <i>Presupuesto</i>	30
3.4.1 Costes de Personal	30
3.4.2 Costes de los equipos	31

3.4.3	Costes de subcontratación y otros costes directos	31
3.4.4	Resumen de costes	31
4.	Fases de la aplicación	35
4.1	<i>Diseño</i>	36
4.2	<i>Implementación</i>	36
4.3	<i>Evaluación</i>	37
5.	Diseño	41
5.2	<i>Diseño de la interfaz gráfica</i>	41
5.2.1	Interfaz del Editor	44
5.2.2	Interfaz del Reproductor	45
5.2.3	Interfaz de la Ayuda Contextual	46
5.3	<i>Diseño de la lógica de la aplicación</i>	47
5.2.1	Lógica de activación de funcionalidad en el reproductor	47
5.2.2	Lógica de la gestión de perfiles personalizados	48
5.2.3	Lógica de la reproducción multimedia	48
5.2.4	Lógica de la comprobación de compatibilidad	49
6.	Implementación	54
6.1	<i>Implementación de la interfaz gráfica</i>	55
6.1.1	Pantalla de Selección de Perfil Inicial	55
6.1.2	Paleta de Activación de Funcionalidad	57
6.1.3	Panel de Controles de Reproducción	59
6.1.4	Pantalla del Reproductor Multimedia	61
6.1.5	Pantalla de la Ayuda Contextual	65
6.1.6	Pantalla del Editor	66
6.2	<i>Implementación de la lógica de la aplicación</i>	71
6.2.1	Lógica de activación de funcionalidad en el reproductor	71
6.2.2	Lógica de gestión de perfiles	72
6.2.3	Lógica de la reproducción multimedia	73
6.2.4	Lógica de la comprobación de compatibilidad	73
7.	Evaluación	78
7.1	<i>Evaluación de la interfaz gráfica</i>	78
7.2	<i>Evaluación de la lógica de la aplicación</i>	78
7.2.1	Prueba 1: Easy Player Editor	78
7.2.2	Prueba 2: Reproducción en Easy Player	85

7.2.3	Prueba 3: Reproducción en Easy Player	92
8.	Conclusiones	100
9.	Futuras líneas de trabajo	104
10.	Referencias	108

Índice de Ilustraciones

Ilustración 1 – Plan de trabajo.....	29
Ilustración 2 – Diagrama de Gantt.....	30
Ilustración 3 – Costes de personal.....	30
Ilustración 4 – Costes de equipos.....	31
Ilustración 5 – Resumen de costes	31
Ilustración 6 - Modelo en Espiral.....	36
Ilustración 7 - Casos de Uso del Prototipo	42
Ilustración 8 - Esquema de la interfaz del Editor Multimedia.....	44
Ilustración 9 - Esquema de la Interfaz del Reproductor	46
Ilustración 10 - Esquema de la Interfaz de la Ayuda Contextual.....	47
Ilustración 11 - Comprobación de Compatibilidad de Accesibilidad.....	50
Ilustración 12 - Esquema de Clases de la Aplicación	55
Ilustración 13 - Selección inicial de Perfil por defecto	56
Ilustración 14 - Selección inicial de Perfil Visual.....	57
Ilustración 15 - Paleta de Opciones del Editor	58
Ilustración 16 - Ejemplo de Etiqueta en la Paleta de Opciones del Editor	59
Ilustración 17 - Ejemplo de Panel de Controles del Reproductor	60
Ilustración 18 - Ejemplo de Botones accesibles por combinación de teclado	61
Ilustración 19 - Ejemplo de Reproductor Multimedia.....	62
Ilustración 20 - Barra de Menú del Reproductor	62
Ilustración 21 - Menú Ayuda del Reproductor	63
Ilustración 22 - Ayuda del Reproductor.....	63
Ilustración 23 - Menú de Audio del Reproductor.....	64

Ilustración 24 - Menú de Subtítulos del Reproductor	64
Ilustración 25 - Menú de Capítulos del Reproductor	65
Ilustración 26 - Vista general del Editor	66
Ilustración 27 - Menú de Perfiles del Editor	67
Ilustración 28 - Asignación del nombre del perfil.....	67
Ilustración 29 - Ventana para guardar el fichero de propiedades del perfil en disco	68
Ilustración 30 - Mensaje de Ayuda Contextual al guardar perfil.....	68
Ilustración 31 - Fichero de propiedades del perfil almacenado en disco.....	69
Ilustración 32 - Ejemplo de fichero de propiedades de un Perfil	69
Ilustración 33 - Menú de perfiles del Editor	70
Ilustración 34 - Mensaje de ejemplo de la Ayuda Contextual al cargar perfil	70
Ilustración 35 - Menú de Ayuda del Editor.....	71
Ilustración 36 – Selección del perfil inicial	79
Ilustración 37 – Pantalla principal del Editor.....	79
Ilustración 38 – Reproductor modificado.....	80
Ilustración 39 – Comprobación de compatibilidad	80
Ilustración 40 – Guardar de perfil.....	81
Ilustración 41 – Nombre de perfil	81
Ilustración 42 – Guardar fichero en disco	82
Ilustración 43 – Perfil guardado	82
Ilustración 44 – Cargar perfil	83
Ilustración 45 – Cargar perfil	83
Ilustración 46 – Cargar perfil	84
Ilustración 47 – Atajo a la Ayuda	84

Ilustración 48 – Atajo a la información de los atajos	85
Ilustración 49 – Información del contenido multimedia.....	86
Ilustración 50 – Reproductor configurado	86
Ilustración 51 – Archivo multimedia abierto.....	87
Ilustración 52 – Selección de pista de audio en inglés	88
Ilustración 53 – Selección de pista de subtítulos en inglés	88
Ilustración 54 – Subtítulos habilitados	89
Ilustración 55 – Subtítulos en pantalla.....	89
Ilustración 56 – Desplazamiento en el vídeo.....	90
Ilustración 57 – Desplazamiento en el vídeo.....	91
Ilustración 58 – Ayuda del Reproductor.....	92
Ilustración 59 – Información del contenido multimedia.....	93
Ilustración 60 – Reproductor configurado	94
Ilustración 61 – Archivo multimedia abierto.....	95
Ilustración 62 – Selección de pistas.....	95
Ilustración 63 – Audiodescripción activada.....	96

Índice de Tablas

Tabla 1 Requisitos de accesibilidad de un reproductor multimedia [24] 24

Introducción

1. Introducción

En esta sección se expondrá una sencilla descripción del problema a tratar, así como la solución en líneas generales por la que se ha optado para resolverlo.

1.1 Motivación del Proyecto

En la actualidad muchas son las aplicaciones ofrecidas en el mercado del software libre para la reproducción de contenido multimedia, generalmente con capacidad de reproducción de la gran mayoría de formatos multimedia, con una buena optimización de recursos, aceleración por hardware, capacidad para realizar streaming de vídeo y una serie de características que completan aplicaciones multimedia realmente potentes.

Sin embargo, la mayoría de este tipo de aplicaciones no suelen estar enfocadas a proveer un contenido multimedia que pueda ser accedido por personas con discapacidad.

La idea es proporcionar una herramienta, un editor gráfico, que permita diseñar reproductores multimedia personalizados teniendo en cuenta las necesidades de acceso del usuario final, de tal forma que se pueda proporcionar un contenido multimedia que pueda ser consumido de una forma más sencilla y pueda ser accedido por dicho usuario. Este editor gráfico se enmarca dentro de la Tesis Doctoral titulada 'Integración de Requisitos de Accesibilidad en el Diseño de Interfaces de Agentes de Usuario Multimedia' realizada por María González García y dirigida por Lourdes Moreno.

Tener un reproductor potente capaz de enviar vídeo del ordenador a una TV a través de Chromecast vía WiFi está muy bien, pero, ¿y si además dicho reproductor se pudiera configurar totalmente en base a las necesidades de un usuario específico? ¿Y si un usuario en lugar de utilizar un reproductor multimedia cualquiera, utilizase “su” propio reproductor multimedia a medida?

1.2 *Ámbito del Sistema*

En base a las limitaciones que los reproductores multimedia pueden tener en cuanto a la accesibilidad, se va a proceder a diseñar un software capaz de editar y diseñar reproductores multimedia personalizados según las preferencias o necesidades de un usuario final.

Se tratará de ofrecer una herramienta capaz de generar reproductores multimedia que faciliten al usuario final la forma de consumir contenido multimedia proporcionándole alternativas según sus necesidades de acceso.

1.3 *Objetivos*

En base a la motivación y el ámbito comentado en las secciones anteriores, a continuación se detallarán los objetivos que deberán cumplirse para considerar el proyecto como finalizado satisfactoriamente:

1. Se deberá desarrollar un software capaz de diseñar un reproductor multimedia personalizado, añadiendo la funcionalidad deseada al mismo: rebobinar, avanzar, reproducir, detener, activar subtítulos, etc.
2. El reproductor multimedia personalizado, será capaz de reproducir todo tipo de contenido multimedia: avi, mp4, mkv, etc, así como reproducir contenido multimedia con múltiples pistas de subtítulos y/o audio, permitiendo seleccionar entre ellas.
3. Se implementará un sistema de guardado de perfiles que permita al usuario guardar dicha configuración, una vez configurado el reproductor personalizado con la funcionalidad deseada.
4. Se implementará un sistema de carga de perfiles (incluidos varios perfiles por defecto: para el acceso por un usuario con discapacidad visual, con discapacidad auditiva, así como funcionalidad completa), de tal forma que el usuario pueda partir de una "plantilla" inicial en base a sus necesidades, y agregar más funcionalidad a partir de ella.

5. La herramienta deberá proveer una ayuda contextual, de tal forma que el usuario sepa en todo momento la funcionalidad que ha activado o desactivado en el reproductor, así como otra información relevante.
6. El usuario podrá en todo momento activar o desactivar cualquier tipo de funcionalidad a través de atajos de teclado, si así lo desea.
7. El reproductor multimedia personalizado, a su vez, deberá ser capaz de ejecutar toda su funcionalidad a través de atajos de teclado.

1.4 Estructura del Documento

El resto del documento está estructurado en los siguientes capítulos:

- **Estado del Arte:** Presentación de los dispositivos y las tecnologías que han sido utilizadas en el proyecto. Se analiza su estado actual, tanto sus características como sus aplicaciones. Además, se realizará un estudio de accesibilidad, en el cual se explicará los tipos de discapacidad, estándares así como algún trabajo relativo.
- **Gestión del Proyecto:** En este capítulo se muestra la metodología empleada para llevar a cabo la realización del proyecto. En él se exponen las tareas en las que se ha dividido así como la duración de cada una y el consiguiente gasto.
- **Fases del Proyecto:** Breve descripción de las fases por las que ha pasado el proyecto así como del modelo de desarrollo utilizado.
- **Diseño:** Descripción de cómo se quiere que se realicen los desarrollos para la implementación de un prototipo que cumpla los requisitos expuestos anteriormente.
- **Implementación:** Descripción de cómo se ha desarrollado el prototipo diseñado en el apartado anterior.
- **Evaluación:** En este capítulo se observan las distintas pruebas y tests realizadas al prototipo con la intención de comprobar su funcionalidad.

- **Conclusiones:** Presentación de las conclusiones obtenidas como consecuencia del desarrollo de este proyecto.
- **Futuras Líneas de Trabajo:** Resumen de las principales ampliaciones posibles para llevar a cabo como continuación o alternativa a este proyecto.

Estado del Arte

2. Estado del Arte

Dado que este Proyecto Fin de Carrera sigue la propuesta establecida en la Tesis Doctoral titulada 'Integración de Requisitos de Accesibilidad en el Diseño de Interfaces de Agentes de Usuario Multimedia' realizada por María González García, en dicha Tesis, se seleccionó Java como tecnología para realizar el desarrollo del editor gráfico. Por lo tanto, en esta sección se realiza un análisis inicial respecto al estado actual de la tecnología Java y de diferentes frameworks que permiten desarrollar interfaces gráficas con Java. Este estudio, ha sido desarrollado previamente al diseño de la aplicación y ha servido de guía introductoria y referencia para las siguientes fases del proyecto.

A continuación, se realiza un estudio de accesibilidad, identificando: tipos de discapacidad, estándares, así como trabajos relativos.

La metodología empleada ha consistido en una primera etapa de investigación puramente teórica en la que se han identificado términos, agrupado conceptos y relacionado temas.

Posteriormente, el estudio se centra en un enfoque mucho más práctico.

La documentación de esta sección ha sido dividida en base a los diferentes conceptos sobre los que se ha realizado el estudio debido a que su uso es un requisito indispensable en este proyecto.

2.1 Tecnologías

2.1.1 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores

de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados [9].

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

Para que una aplicación creada en Java sea capaz de ejecutarse en un ordenador, servidor o dispositivo en general, es necesario que en dicho dispositivo exista el Java Runtime Environment (JRE), o en castellano Entorno de Ejecución de Java, para el sistema operativo del dispositivo en el que se vaya a ejecutar la aplicación.

La gran ventaja, es que el código Java de la aplicación no es necesario cambiarlo en ningún momento, independientemente de la plataforma en la que se ejecute. En cada dispositivo, se instalará su versión correspondiente de Java JRE, el cual incluye la

máquina virtual de Java para esa plataforma, y la aplicación, será ejecutada e interpretada directamente a través de dicha máquina virtual.

Ventajas

- La sintaxis del lenguaje es muy parecida a la de C/C++.
- Es un lenguaje orientado a objetos.
- Java permite crear programas multitarea.
- Permite el uso de excepciones para manejar errores.
- Es más difícil cometer errores de programación que en C/C++ dado que no existen los punteros.
- Existen entornos de programación muy maduros, como por ejemplo Eclipse o Netbeans.
- Existe una vasta comunidad de usuarios que comparten gran cantidad de código libre, por ejemplo: GitHub.

2.2 Frameworks

2.2.1 Swing

Es un framework Modelo-Vista-Controlador (MVC) para desarrollar interfaces gráficas para Java con independencia de la plataforma. Sigue un simple modelo de programación por hilos, y posee las siguientes características principales [\[10\]](#):

- Independencia de plataforma.
- Extensibilidad: es una arquitectura altamente particionada, es decir, los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto. Se puede extender clases existentes proveyendo alternativas de implementación para elementos esenciales.
- Personalizable: dado el modelo de representación programático del framework de Swing, el control permite representar diferentes estilos de apariencia "look

and feel" (desde apariencia MacOSX hasta apariencia Windows pasando por apariencia GTK+, IBM UNIX o HP UX entre otros). Además, los usuarios pueden proveer su propia implementación de apariencia, que permitirá cambios uniformes en la apariencia existente en las aplicaciones Swing sin efectuar ningún cambio al código de aplicación.

Ventajas

- El diseño en Java puro posee menos limitaciones de plataforma.
- El desarrollo de componentes Swing es más activo.
- Los componentes de Swing soportan más características.

2.2.2 AWT

La Abstract Window Toolkit (AWT, en español Kit de Herramientas de Ventana Abstracta) es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java [\[11\]](#).

Cuando Sun Microsystems liberó Java en 1995, AWT suministró solo un nivel de abstracción muy fino sobre la interfaz de usuario nativa subyacente. Por ejemplo, crear una caja de verificación AWT causaría que AWT directamente llame a la subrutina nativa subyacente que cree una caja de verificación. Sin embargo, una caja de verificación en Microsoft Windows no es exactamente lo mismo que una caja de verificación en Mac OS o en los distintos tipos de UNIX.

Desde sus inicios el entorno Java ya contaba con una biblioteca de componentes gráficos conocida como AWT. Esta biblioteca estaba concebida como una API estandarizada que permitía utilizar los componentes nativos de cada sistema operativo. Entonces una aplicación Java corriendo en Microsoft Windows usaría el botón estándar de Windows y una aplicación corriendo en UNIX usaría el botón estándar de dicho sistema operativo. En la actualidad dicha tecnología puede considerarse en desuso debido a los siguientes motivos:

- Depende fuertemente de los componentes nativos del sistema operativo, y el programador AWT se ve limitado a las funcionalidades comunes en todos los sistemas operativos.
- El comportamiento de los controles varía mucho de sistema a sistema y se vuelve muy difícil construir aplicaciones portables.

2.2.3 Java Media Framework

Java Media Framework (JMF) proporciona a los applets y aplicaciones Java la capacidad de reproducir, capturar y transmitir/recibir en tiempo real audio, vídeo y otros contenidos multimedia. Facilita una serie de codificadores y decodificadores para los formatos multimedia más relevantes siendo capaz además, de realizar transcodificación entre dichos formatos [12].

Entre sus principales características cabe destacar:

- Estabilidad, al operar sobre la máquina virtual Java (JVM).
- Sencillez, ya que permite realizar complejas tareas multimedia con unos pocos comandos.
- Potencia, permitiendo la manipulación de elementos multimedia de audio y vídeo, así como su retransmisión en tiempo real a través de la red, merced al protocolo RTP.

2.2.4 JavaFX

JavaFX es una familia de productos y tecnologías de Sun Microsystems, adquirida por Oracle Corporation, para la creación de aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX planeados [13].

La intención de Sun Microsystems respecto de JavaFX es competir en el espacio que ya ocupan Flash de Adobe, y Silverlight de Microsoft.

Las aplicaciones JavaFX pueden ser ejecutadas en una amplia variedad de dispositivos.

Ventajas

- Es una extensión a la plataforma Java, ofreciendo interactividad, animación y programación compatible con AJAX, Adobe Flash, y MS Silverlight.
- Permite crear contenido rico en medios.
- Se trata de un sistema de desarrollo para la web.
- Multiplataforma: ordenadores, móviles, consolas, etc.
- Tiene como objetivo que los navegadores operen mejor que con Javascript, el cual es más lento y con diferentes implementaciones en función del navegador.

2.2.5 VLCJ

El framework VLCJL [\[1\]](#) es un proyecto Open Source que proporciona funcionalidad multimedia y está basado en el conocido reproductor multimedia VLC, de VideoLAN.

Los mapeos que ofrecen las librerías VLCJ pueden ser utilizados para construir todo tipo de reproductores multimedia utilizando Java, desde una simple ventana para reproducir un vídeo, hasta una compleja aplicación de reproducción en streaming bajo demanda. El reproductor multimedia VLC [\[3\]](#) es un reproductor multiplataforma muy popular, disponible para Linux, MacOSX y Windows. La mayor diferencia respecto al resto de reproductores es que incluye soporte directo para la reproducción de contenido en prácticamente cualquier formato de vídeo codificado con cualquier tipo de códec, sin la necesidad de tener que instalar dichos códec en el sistema operativo independientemente.

El reproductor multimedia VLC a pesar de ser multiplataforma, está compilado específicamente para cada una de ellas, y no se puede usar directamente a través de Java. VLC facilita una Interfaz de Programación de Aplicación [\[4\]](#) (API) que se llama “libVLC”, la cual se incluye con la instalación del reproductor VLC. Por ello, es necesaria una capa Java para interactuar con esa librería, y eso es precisamente lo que la librería VLCJ ofrece.

Es posible utilizar VLCJ para construir aplicaciones usando diferentes kits de desarrollo, incluyendo: Swing, AWT, SWT, JavaFX y OpenGL.

Existen muchos tipos de aplicaciones que pueden desarrollarse a través de VLCJ, entre ellas se encuentran:

- Reproductor cliente enriquecido de vídeo/audio embebido en una aplicación Java.
- Vídeo embebido directamente en una aplicación Swing, SWT, JavaFX, OpenGL, JMonkeyEngine.
- Vídeo embebido en aplicaciones enriquecidas Eclipse/Netbeans.
- Reproductor de radio a través de Internet.
- Reproductor de YouTube.
- Transcodificador.
- Servidor de streaming a través de la red.
- Cliente de streaming.

Ventajas

- Acceso a prácticamente toda la funcionalidad nativa de la librería libVLC a través del API.
- Framework de alto nivel que simplifica el acceso a la funcionalidad de la librería libVLC, encapsulando altamente el acceso a la librería nativa, así como los comportamientos específicos en cada plataforma.
- Protección ante usos incorrectos de la librería nativa, librando así de posibles errores en tiempo de ejecución. Por ejemplo, VLCJ encapsula completamente el manejo de eventos asíncronos nativos del reproductor multimedia, registrando callbacks para ello, y despachándolos de forma serial a través del patrón de diseño “Observer” [\[5\]](#).
- Actualmente, el proyecto VLCJ se encuentra en una fase madura, y se pueden realizar aplicaciones multimedia Java avanzadas, capaces de reproducir

prácticamente cualquier tipo de contenido multimedia de forma local, en red local, o a través de Internet, así como ejercer de servidor de dicho contenido a modo de estación web o vídeo bajo demanda.

2.3 Estudio de Accesibilidad

Hay muchas razones por las cuales una persona puede experimentar diferentes grados de discapacidad en el campo auditivo, cognitivo, neurológico, físico, comunicativo y visual. Por ejemplo, algunas personas pueden tener discapacidades de nacimiento, como consecuencia de algún accidente o enfermedad, o en otros casos, haber desarrollado este tipo de problemas con el paso de los años. Muchas veces ocurre que una persona con algún tipo de discapacidad leve, no reconoce dicha discapacidad.

Cada individuo es único, y por ello, cabe considerar los siguientes puntos (basado en el estudio de la iniciativa de la W3C [\[6\]](#)):

- **Discapacidad debido a la edad**

Mucha gente desarrolla algún tipo de discapacidad con el paso de los años.

- **Múltiples discapacidades**

Algunas personas tienen combinaciones de varias discapacidades. Por ejemplo, una persona sorda y con limitaciones visuales, se puede beneficiar del uso de textos/subtítulos, pero únicamente si dicho texto es ajustable en tamaño y color.

- **Condiciones de salud**

En muchas ocasiones las condiciones de salud pueden afectar a la resistencia, destreza y concentración. Por ejemplo, alguien podría experimentar fatiga, dolor u otro tipo de síntoma que pudiera afectar a su capacidad física para estar delante del ordenador.

- **Limitaciones cambiantes**

En otras ocasiones, una persona puede experimentar limitaciones recurrentes

o progresivas. Por ejemplo, puede darse el caso en que una persona necesite una serie de características de accesibilidad un día, y otro día otras diferentes.

- **Limitaciones temporales**

Algunas personas pueden experimentar limitaciones temporales, como por ejemplo aquellas que se deban a algún tipo de accidente, cirugía o medicación. En estos casos, es frecuente que la persona no conozca soluciones de accesibilidad ni cómo usarlas, y en algunos casos, ni siquiera sea consciente de que necesite dichas soluciones de accesibilidad.

- **Limitaciones situacionales**

Algunas personas pueden experimentar algún tipo de limitación circunstancial. Por ejemplo, pueden encontrarse en un ambiente ruidoso dónde no se es capaz de escuchar audio, o en un lugar con mucha luz dónde no es capaz de ver la pantalla adecuadamente.

2.3.1 Tipos de discapacidad

2.3.1.1 Auditiva

El rango de la discapacidad auditiva va desde leve o moderado en uno o los dos oídos (“difícil de escuchar”), hasta substancial o incorregible en ambos oídos (“sordera total”).

Algunas personas con discapacidad auditiva, son capaces de escuchar sonidos, pero muchas veces es insuficiente para seguir una conversación, especialmente si además se encuentran en un ambiente con ruido de fondo. Aquí se incluye a las personas que utilizan dispositivos que mejoran el sonido.

- Las personas con este tipo de discapacidad, necesitan:
 - Transcripciones y textos/subtítulos del audio, incluyendo contenido en formato audio, o pistas de audio en contenido multimedia.
 - Reproductores multimedia capaces de mostrar subtítulos, y ajustar el tamaño y color de éstos.

- Opciones para parar, pausar y ajustar el volumen del audio.
- Gran calidad del audio en primer plano, claramente diferenciable del sonido de fondo.
- Ejemplos de discapacidad auditiva:
 - Dificil de escuchar: leve o moderado en uno o los dos oídos.
 - Sordera total: substancial e incorregible en ambos oídos.
 - Sordo-ciego: substancial e incorregible en el campo auditivo y visual.

2.3.1.2 Cognitiva y neurológica

Las discapacidades cognitivas y neurológicas implican algún tipo de desorden en el sistema nervioso, incluyendo el cerebro y el sistema nervioso periférico. Esto puede impactar directamente en cómo una persona escucha, se mueve, ve, habla y entiende la información. Este tipo de discapacidad no tiene por qué afectar a la inteligencia de la persona.

- Las personas con este tipo de discapacidad, necesitan:
 - Contenido claro y estructurado que facilite la orientación y visión general.
 - Etiquetado consistente de formularios, botones y resto de contenido.
 - Contenido y funcionalidad predecible.
 - Diferentes opciones de acceder a la misma funcionalidad (botones directos visibles, barras de menú, etc).
- Ejemplos de discapacidad cognitiva y/o neurológica:
 - Déficit de atención por desorden de hiperactividad: dificultad para centrarse en una única tarea, o dificultad para centrarse durante largos periodos, fácil de distraer.
 - Desorden por espectro de autismo: discapacidad de comunicación e interacción social.

- Intelectual: limitación de inteligencia, aprendizaje lento, o incapacidad para comprender conceptos complejos. Síndrome de down es uno de los casos más comunes para este tipo de discapacidad.
- De aprendizaje: es más un término funcional que una condición médica, y no está uniformemente definido.
- Salud mental: incluye ansiedad, delirio, depresión, paranoia, esquizofrenia y otro tipo de desórdenes. Estas condiciones causan dificultad para centrarse en procesar información o comprenderla. Particularmente, la medicación para paliar estos síntomas, pueden causar otros efectos colaterales como por ejemplo: visión nublada, temblores de mano, etc.
- De memoria: incluye limitación para memorizar en el corto plazo, en el largo plazo o dificultad para recordar palabras/lenguaje.
- Esclerosis múltiple: causa daño en las células nerviosas del cerebro y médula espinal, pudiendo afectar a las habilidades auditivas, cognitivas, físicas o visuales en periodos de brote o recaída.
- Perceptual: dificultad en el procesado de información auditiva, táctil visual u otro tipo de información sensorial. Puede influir en la lectura (dislexia), escritura (disortografía), procesamiento de números, u orientación espacio temporal.
- Ataques: incluye diferentes tipos de epilepsia y migraña, que pueden ser debido a una reacción visual parpadeante o señales de audio a ciertas frecuencias.

2.3.1.3 Física

Las discapacidades físicas incluyen: debilidad y limitación muscular (como por ejemplo movimientos involuntarios incluyendo temblores, falta de coordinación o parálisis), limitaciones sensoriales, problemas de articulaciones (artritis), fuertes dolores, o falta de extremidades.

- Las personas con este tipo de discapacidad, necesitan:
 - Periféricos ergonómicos adaptados.
 - Reconocimiento de voz, seguimiento de vista, y otro tipo de aproximaciones para tener control sobre la pantalla.
- Ejemplos de discapacidad física:
 - Amputación o deformidad.
 - Artritis.
 - Fibromialgia.
 - Reumatismo.
 - Distrofia muscular.
 - Temblores y espasmos.
 - Cuadriplejía.

2.3.1.4 Comunicativa

Las discapacidades comunicativas incluyen dificultad para producir una conversación entendible por otros o por software de reconocimiento de voz. Por ejemplo, una voz estrepitosamente alta o la falta de claridad pueden ser difíciles de entender.

- Las personas con este tipo de discapacidad, necesitan:
 - Modelos alternativos de interacción.
 - Chat basado en texto.
 - Atajos de teclado para interactuar con la aplicación.
- Ejemplos de discapacidad comunicativa:
 - Apraxia: inconsistencia en la articulación y reproducción de sonidos, así como errores en la generación de sonidos en el orden correcto.
 - Taquifemia: falta de ritmo en la conversación, entonación, y co-articulación de sonidos.
 - Disartria: debilitación o completa parálisis de los músculos necesarios para reproducir sonidos.

- Desorden en la generación de sonidos: incluye dificultad o incapacidad para reproducir ciertos sonidos o patrones de sonidos, muchas veces incurriendo en: agregaciones, distorsiones, omisiones o substracciones de esos sonidos.
- Tartamudez.
- Mudez.

2.3.1.5 Visual

El rango de las discapacidades visuales va desde leve o moderado en uno o ambos ojos (“baja visión” o “visión parcial”) hasta pérdida de visión sustancial e incorregible en ambos ojos (“ceguera total”). Algunas personas sufren una reducción en el rango de colores perceptibles (“daltonismo”), o una elevada sensibilidad en la intensidad de los colores. Esta variación en la percepción de los colores y brillo pueden ser independientes de la agudeza visual.

- Las personas con este tipo de discapacidad, necesitan:
 - Alargar o reducir el tamaño de los textos e imágenes.
 - Personalizar las características de fuentes, colores y tabulado.
 - Escuchar la audio descripción del vídeo multimedia.
- Ejemplos de discapacidad visual:
 - Daltonismo: incluye desde dificultad para distinguir colores entre rojo y verde, o entre amarillo y azul, y casualmente incapacitación para percibir color alguno.
 - Baja visión: incluye baja agudeza visual, visión túnel (capaz de ver únicamente el centro del campo visual), visión periférica (capaz de ver únicamente los lados del campo visual), y visión nublada.
 - Ceguera total: sustancial e incorregible pérdida de la visión en ambos ojos.
 - Sordo-ceguera: discapacidad incorregible tanto auditiva como visual.

2.3.2 Estándares de Accesibilidad

2.3.2.1 Web Content Accessibility Guidelines (WCAG 2.0)

El estándar Web Content Accessibility Guidelines (WCAG 2.0) [14] se ocupa de la accesibilidad al contenido Web para personas con discapacidad. De acuerdo con la pauta 1.2 de las WCAG 2.0 [14], el contenido de vídeo debe ser acompañado de diferentes alternativas multimedia, como por ejemplo la utilización de subtítulos (para personas sordas), audio descripción (para personas ciegas), etc. Desde 2012, este estándar es una norma ISO (ISO/IEC 40500:2012 [36], “Information technology – W3C Web Content Accessibility Guidelines (WCAG) 2.0” [14]), la cual cubre un amplio rango de recomendaciones para elaborar contenido Web más accesible. A parte de este estándar, existen una gran cantidad de otras iniciativas basadas en la WCAG 2.0:

- Technical standards Section 508 [15] en Estados Unidos.
- Barrierefreie-Informationstechnik-Verordnung (BITV) 2.0 [16] en Alemania.
- Référentiel Général d’Accessibilité pour les Administrations (RGAA) [17] en Francia.
- Accessibility for Ontarians with Disabilities Act (AODA) [18] en Canada.
- UNE 139803:2012 [19] en España.

2.3.2.2 UAAG 2.0

Este estándar internacional explica cómo crear agentes de usuario accesibles para personas con discapacidad y cómo aumentar la accesibilidad al contenido Web. Por ejemplo, los reproductores multimedia deberían asegurar que los controles de audio y vídeo sean accesibles a través de combinaciones de teclado o lectores de pantalla [20].

Este estándar actualmente dispone de dos versiones, UAAG 1.0 [21], y UAAG 2.0 [20]. La segunda soporta el estándar WCAG 2.0. Por lo tanto, de acuerdo con las UAAG 2.0, los reproductores multimedia deberán dar soporte a los controles multimedia alternativos requeridos en la guía 1.2 de las WCAG 2.0 (audio descripción, subtítulos, etc).

2.3.2.3 ATAG 2.0

Este estándar provee una línea que permite diseñar herramientas de creación de contenido Web, que sean por un lado más accesibles para los autores con discapacidades, y que además promocióne la producción de contenido Web más accesible para todos los autores [22].

ATAG 2.0 se divide en dos partes, A y B. La parte A se refiere a la accesibilidad de las interfaces de usuario de herramientas de autor para la creación de contenidos para autores con algún tipo de discapacidad, mientras que la parte B da soporte a la producción de contenido accesible.

2.3.2.4 ISO 9241-171

Esta normal, ergonomía de la interacción Hombre-Máquina, proporciona guías sobre la accesibilidad del software [23].

Este estándar proporciona guías ergonómicas y especificaciones para el diseño de software accesible para su uso en el trabajo, en casa, en educación y en lugares públicos.

2.3.3 Requisitos de accesibilidad de un reproductor multimedia

Como se ha comentado anteriormente, el trabajo realizado en este Proyecto Fin de Carrera se enmarca dentro de la Tesis Doctoral titulada 'Integración de Requisitos de Accesibilidad en el Diseño de Interfaces de Agentes de Usuario Multimedia'. Por lo tanto, en dicha Tesis, se ha realizado una revisión exhaustiva de los estándares de accesibilidad presentados en la sección Estándares de Accesibilidad y se han obtenido un conjunto de requisitos de accesibilidad que son necesarios para considerar un reproductor multimedia accesible. En la Tabla 1 se muestra este conjunto de requisitos. Como puede observarse, a cada requisito se le ha asignado un código, un grupo y un subgrupo [24].

Tabla 1 Requisitos de accesibilidad de un reproductor multimedia [24]

Código	Nombre	Descripción	Grupo	Subgrupo	Fuente	
					ISO	UAAG 2.0
NP01	Play	Reproducir contenido multimedia	Nativo	Reproducción	10.8.2	2.11.6
NP02	Stop	Detener contenido multimedia	Nativo	Reproducción	10.8.2	2.11.6
NP03	Pause	Pausar contenido multimedia	Nativo	Reproducción	10.8.2	2.11.6
NS01	Resize	Redimensionar ventana gráfica	Nativo	Tamaño	10.5.8	1.8.3
NV01	Mute	Habilitar/Deshabilitar audio	Nativo	Volumen	10.6.2	1.5.1
NV02	Volumen	Ajustar volumen	Nativo	Volumen	10.6.2	1.5.1
AP01	Rewind	Rebobinar segundos en la reproducción	Adicional	Reproducción	10.8.3	2.11.7
AP02	Forward	Avanzar segundos en la reproducción	Adicional	Reproducción	10.8.3	2.11.7
AA01	Caption	Habilitar / Deshabilitar subtítulos	Adicional	Alternativo	10.1.3, 10.7.2	1.1.2
AA02	AudioDescription	Habilitar / Deshabilitar audio descripción	Adicional	Alternativo	10.1.3	1.1.2
AA03	Size	Modificar tamaño de subtítulos	Adicional	Alternativo	10.7.3	1.4.1
AA04	Font	Modificar tipo de fuente de los subtítulos	Adicional	Alternativo	10.7.3	1.4.1
AA05	Color	Modificar color de los subtítulos	Adicional	Alternativo	10.7.3	1.4.1
AA06	LanguageCaption	Modificar el idioma de los subtítulos	Adicional	Alternativo	8.2.1	2.7.1
AA07	LanguageAudio	Modificar el idioma del audio descripción	Adicional	Alternativo	8.2.1	2.7.1
AH01	Help	Documentación de ayuda sobre las características de accesibilidad	Adicional	Ayuda	11.1.5	3.3.2
AF01	Find	Búsqueda en subtítulos	Adicional	Búsqueda	-	2.4.1

2.4 Trabajos relativos

En esta sección cabe destacar trabajos anteriores en el mismo ámbito que el proyecto que estamos desarrollando, como por ejemplo:

- **JW Player:** proporciona la posibilidad de incluir subtítulos y audio descripción [25].

- **BSPlayer:** proporciona la posibilidad de incluir subtítulos, así como cambiar el la fuente y su tamaño [26].
- **VideoLan Media Player:** proporciona la posibilidad de utilizar atajos de teclado, así cómo cambiar el tipo de fuente, tamaño y color de los subtítulos. Es uno de los reproductores más completos en la actualidad [27].
- **Acorn Media Player:** basado en HTML5, proporciona control total por teclado. Además, permite la integración con subtítulos externos en formato "SRT". También proporciona la posibilidad de generar transcripciones dinámicas de los subtítulos seleccionados [28].
- **LeanBack:** basado en HTML5, soporta la inclusión de elementos de audio y vídeo, permite incorporar subtítulos a través del uso de elementos de pista, y proporciona atajos de teclado para navegadores [29].
- **Video JS:** basado en HTML5, es un reproductor que implementa una solución sencilla y amigable al uso de subtítulos [30].
- **Easy Youtube Player:** proporciona la posibilidad de reproducir vídeos de YouTube de una forma más accesible a todos los públicos [31].
- **KMPlayer:** proporciona la posibilidad de incluir subtítulos, así como cambiar la fuente y su tamaño [32].
- **OZPlayer:** basado en los estándares de HTML5 para audio y vídeo, ofrece retro compatibilidad con vídeos antiguos Flash. Este reproductor es totalmente compatible con el estándar Web Content Accessibility Guidelines (WCAG 2.0) [14], ofreciendo reproducción de subtítulos así como de audio descripción [33].

2.5 Discusión del Estado de Arte

Este proyecto gira en torno a la reproducción de contenidos multimedia de forma más accesible para aquellas personas que tienen algún tipo de discapacidad.

En la actualidad, la tecnología vive un nivel de gran desarrollo, sin embargo, en este campo, aún queda trabajo por delante.

Por ello, en este proyecto, se intenta profundizar en las tecnologías actuales para realizar un prototipo que cumpla los estándares de accesibilidad, y proporcione a los usuarios con cualquier tipo de discapacidad, un acceso más cercano a los contenidos multimedia.

Gestión del Proyecto

3. Gestión del Proyecto

En esta sección se describe la gestión del proyecto que se presenta en este documento.

En diferentes secciones a continuación, se estimará el esfuerzo del proyecto en tiempo, costes de personal y equipos, etc. Esta estimación se realiza antes de comenzar las diferentes fases de las que consta dicho proyecto, y serán utilizadas para realizar un estudio de viabilidad del mismo.

3.1 *Descomposición en Tareas*

Estas son las diferentes tareas de las que consta el proyecto:

- Estudio inicial
 - Estado de la cuestión: visión inicial del proyecto a realizar.
 - Planteamiento del problema: en la actualidad existe gran variedad de reproductores multimedia, pero faltan reproductores de contenido centrados en la accesibilidad. Además, no hay herramientas para construir reproductores personalizados en base a los diferentes tipos de acceso o las diferentes formas en las que un usuario accede al contenido que se ofrece.
 - Croquis de la solución: planteamiento de una posible solución a grandes rasgos.
- Desarrollo
 - Toma de requisitos: definición de los requisitos necesarios para la herramienta de edición de reproductores multimedia personalizados que se desea construir.
 - Especificación de los requisitos: análisis en mayor detalle de los requisitos.
 - Diseño: distintas vías para obtener el resultado esperado.

- Programación de la solución: una vez fijado el diseño y la plataforma sobre la que se va a construir la solución, se procederá a la implementación.
 - Pruebas: una vez construida la solución, a través de una batería de pruebas se verificará que la misma cumple con los requisitos establecidos inicialmente.
- Memoria y documentación: en esta fase, se incluirán todas las tareas relacionadas con la documentación del proyecto, esto es, análisis, desarrollo, implementación y pruebas, así como el manual de usuario.

3.2 Planificación del Proyecto

En esta sección se procede a detallar las tareas y fechas planificadas para el desarrollo completo del proyecto.

Tareas	Planificación			Dedicación Real			
	Inicio	Fin	Dedicación (h)	Inicio	Fin	Dedicación (h)	Desviación (h)
Estudio Preliminar							
Estado de la cuestión	01/12/2014	18/12/2014	112	01/12/2014	18/12/2014	112	0
Planteamiento del problema	19/12/2014	19/12/2014	8	19/12/2014	19/12/2014	8	0
Croquis de la solución	22/12/2014	26/12/2014	40	22/12/2014	26/12/2014	40	0
Desarrollo							
Toma de requisitos	29/12/2014	01/01/2015	32	29/12/2014	01/01/2015	32	0
Especificación de requisitos	02/01/2015	05/01/2015	16	02/01/2015	06/01/2015	24	8
Diseño	06/01/2015	18/01/2015	72	07/01/2015	20/01/2015	80	8
Programación de la solución	19/01/2015	19/02/2015	192	21/01/2015	23/02/2015	192	0
Pruebas	20/02/2015	23/02/2015	24	24/02/2015	27/02/2015	32	8
Memoria							
Introducción	07/01/2015	12/01/2015	32	07/01/2015	12/01/2015	32	0
Estado del arte	27/01/2015	05/02/2015	64	27/01/2015	05/02/2015	64	0
Gestión del proyecto	06/02/2015	17/02/2015	64	06/02/2015	17/02/2015	64	0
Fases del proyecto	18/02/2015	13/03/2015	144	18/02/2015	13/03/2015	144	0
Conclusiones	16/03/2015	17/03/2015	16	16/03/2015	17/03/2015	16	0
Futuras líneas de trabajo	18/03/2015	20/03/2015	24	18/03/2015	20/03/2015	24	0
Anexos	23/03/2015	27/03/2015	40	23/03/2015	27/03/2015	40	0
Presentación							
Preparación de diapositivas	30/03/2015	01/04/2015	24	30/03/2015	01/04/2015	24	0
Preparación discurso	31/08/2015	02/08/2015	24	31/08/2015	02/08/2015	24	0

Ilustración 1 – Plan de trabajo

3.3 Diagrama de Gantt

En este apartado se muestra el diagrama de Gantt, en el cuál se pueden observar el orden de las tareas descritas en el apartado Descomposición en Tareas, así como la duración de las mismas:

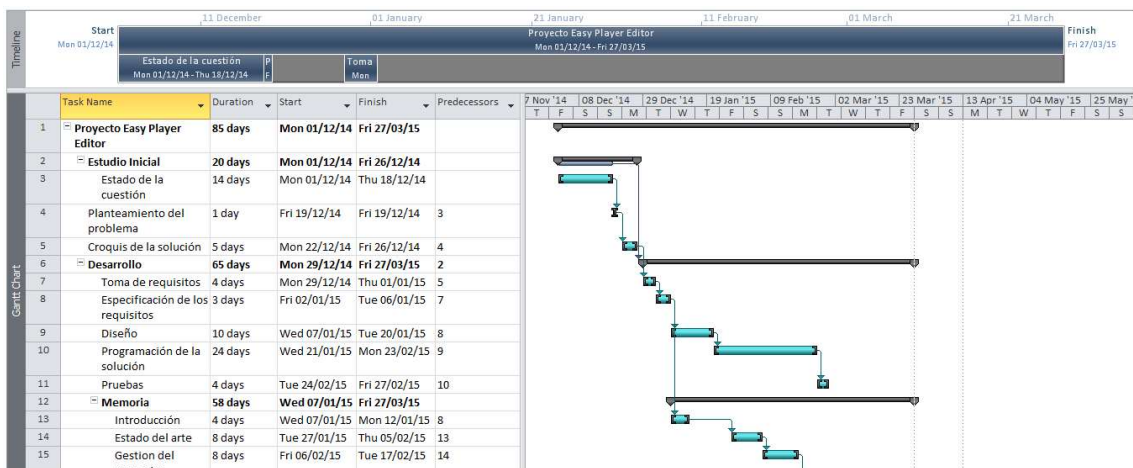


Ilustración 2 – Diagrama de Gantt

3.4 Presupuesto

3.4.1 Costes de Personal

Los costes mostrados a continuación hacen referencia a las horas invertidas por el equipo de trabajo que ha desarrollado el proyecto en las diferentes tareas de gestión, diseño, programación y evaluación.

FUNCIÓN	HORAS	PRECIO(€)/HORA	SUBTOTAL(€)
Diseño	80	30,5	2440
Gestion	56	48,75	2730
Programación	192	19,25	3696
Pruebas	32	22,5	720
TOTAL			9586

Ilustración 3 – Costes de personal

3.4.2 Costes de los equipos

A continuación se describen los costes presupuestados a raíz de los equipos hardware y software usados para la realización del proyecto.

DESCRIPCIÓN	COSTE ANUAL	USO AL PROYECTO	DEDICACIÓN	DEPRECIACIÓN	COSTE IMPUTABLE
Ordenador portátil	625 €	100	4 meses	12 meses	208,33 €
Licencia Windows	0 €	100	5 meses	12 meses	-
Licencia Java	0 €	100	6 meses	12 meses	-
Licencia VLCJ	0 €	100	7 meses	12 meses	-
				TOTAL	208,33

Ilustración 4 – Costes de equipos

3.4.3 Costes de subcontratación y otros costes directos

Puesto que se trata de la creación de un prototipo, se ha optado por no subcontratar tareas a otras empresas y por lo tanto, no ha habido otro tipo de gasto adicional a imputar al proyecto. El coste de subcontratación, por ende es 0€.

3.4.4 Resumen de costes

Teniendo en cuando los gastos detallados en los apartados anteriores, el resumen de gastos queda como sigue:

Personal	9.586 €
Equipos	208,33
Subcontratación	0 €
Otros	0
TOTAL	9794,33 €

Ilustración 5 – Resumen de costes

Fases de la aplicación

4. Fases de la aplicación

En este capítulo se van a presentar las diferentes fases del proyecto. El ciclo de vida del software elegido se basa en el modelo de desarrollo en espiral, por ese motivo, aunque la aplicación consta de tres módulos diferenciados como son diseño, implementación y evaluación, no implica que se hayan ejecutado en ese orden.

El modelo de desarrollo en espiral fue definido en el año 1986 por Barry Boehm [7].

Este modelo básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo en Cascada, pero no necesariamente debe ser así. El Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del Modelo de Prototipos con los aspectos controlados y sistemáticos del Modelo en Cascada, con el agregado de gestión de riesgos.

En cada iteración las tareas que se realizarán serán de cuatro tipos:

- Análisis de requerimientos.
- Diseño del sistema.
- Etapas de construcción.
- Test y evaluación.

En la siguiente ilustración, se puede ver de forma gráfica cómo se representaría el modelo de desarrollo en espiral:

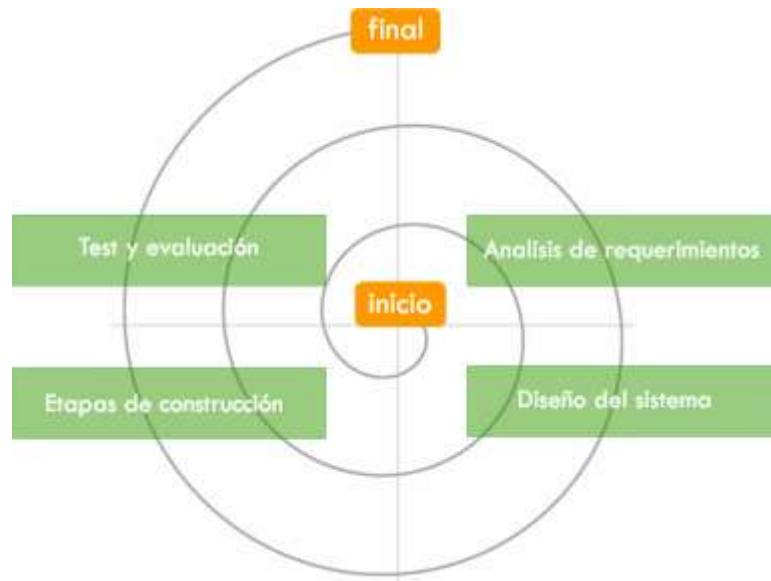


Ilustración 6 - Modelo en Espiral

4.1 *Diseño*

En esta sección se diseñará el concepto de la aplicación, es decir, básicamente se establecerán las bases para que la aplicación pueda ser posteriormente implementada.

En esta fase, se tratará de diseñar módulos independientes entre sí, para tratar de potenciar su reutilización en el futuro, ya sea para ampliaciones de la propia aplicación o incluso para futuras aplicaciones.

4.2 *Implementación*

En esta sección se partirá con las bases de la fase de diseño para proceder a la implementación real de la aplicación. Se explicarán el cómo y el por qué de la implementación que se ha llevado a cabo en este caso particular.

Se trata de una fase mucho más técnica que la fase de diseño, y entre otras cosas, en ella se explicarán el por qué de las tecnologías elegidas.

4.3 Evaluación

La fase de evaluación se corresponde con la última iteración del modelo de ciclo de vida del software en espiral. Esta fase es la responsable de asegurar que la aplicación funciona correctamente, y que obtiene el resultado esperado para cada caso de uso.

Es muy frecuente en proyectos mal planificados, o con requisitos poco claros, que la funcionalidad final de un software no sea exactamente la que se pensó en un principio. Por ello, a pesar de ser la última, es una de las fases más importantes.

Diseño

5. Diseño

En esta sección se establecerán las bases para posteriormente poder desarrollar una herramienta capaz de generar reproductores multimedia personalizados con funcionalidades de accesibilidad.

Se pueden diferenciar dos bloques principales a la hora de diseñar el prototipo:

- Interfaz gráfica
 - Editor gráfico que permite diseñar reproductores multimedia personalizados
 - Reproductor multimedia personalizado según características de acceso
- Lógica de la aplicación

El prototipo recibirá el nombre de ***Easy Player Editor***, puesto que la idea es crear un editor de reproductores multimedia de fácil manejo, que permita habilitar o deshabilitar funcionalidades específicas en un reproductor y que a su vez sea totalmente funcional.

5.2 *Diseño de la interfaz gráfica*

Para poder realizar un buen diseño de la interfaz gráfica, inicialmente cabe considerar los posibles casos de uso que posteriormente se desean implementar.

Dadas las características del prototipo a diseñar, hay que remarcar que habrá una serie de casos de uso para el editor, y otro conjunto de casos de uso para el reproductor multimedia personalizado.

El siguiente diagrama muestra los casos de uso identificados:

CASOS DE USO

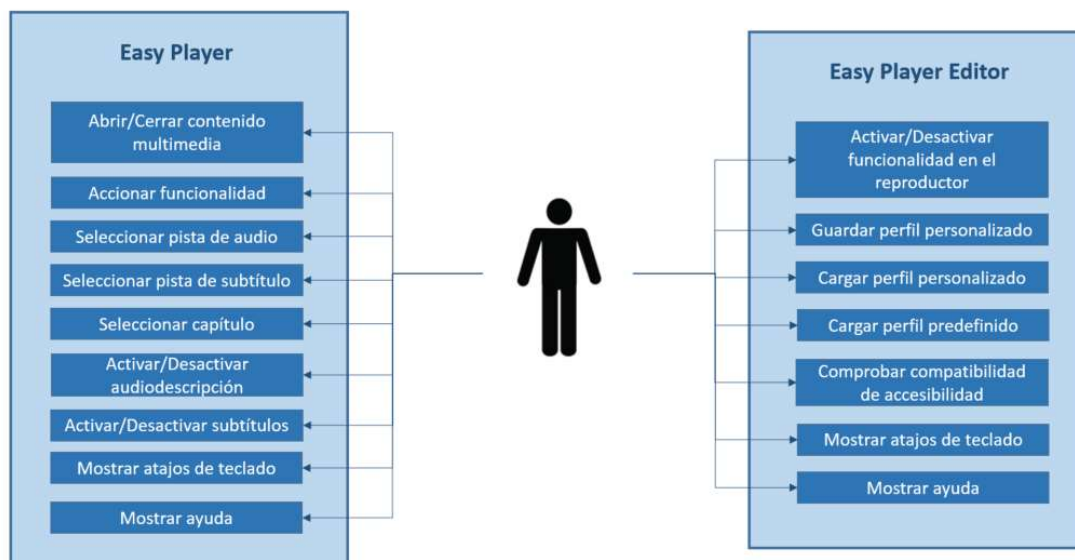


Ilustración 7 - Casos de Uso del Prototipo

A continuación, se explicará brevemente cada uno de estos casos de uso.

- Casos de uso para el editor Easy Player Editor:
 - Activar/Desactivar funcionalidad: será posible activar y desactivar desde el editor funcionalidades de reproducción, como por ejemplo: detener, reproducir, rebobinar, avanzar, ajustar volumen, etc.
 - Guardar perfil personalizado: una vez el usuario tiene un reproductor con la funcionalidad deseada, podrá guardar el perfil en disco.
 - Cargar perfil personalizado: en todo momento, el usuario podrá cargar un perfil personalizado existente en disco, para poder trabajar a partir del mismo.
 - Cargar perfil predefinido: en todo momento, el usuario podrá cargar cualquiera de los perfiles predefinidos incluidos en el editor.
 - Comprobar compatibilidad de accesibilidad: se contrastará la funcionalidad configurada actualmente, con los perfiles de accesibilidad pre-configurados, indicando si se cumplen o no.

- Mostrar atajos de teclado: aparecerá un pop up con todos los atajos de teclado disponibles en el editor.
 - Mostrar ayuda: aparecerá un pop up en el que se mostrará información general de la aplicación.
 - Casos de uso para el reproductor Easy Player:
 - Abrir/Cerrar contenido multimedia: el usuario podrá en todo momento abrir un archivo con contenido multimedia, en cualquier formato.
 - Accionar funcionalidad: en este caso de uso se engloba cualquier tipo de interacción del usuario final con cualquier tipo de funcionalidad de reproducción puramente dicha: reproducir, detener, rebobinar, avanzar, ajustar volumen, silenciar, etc.
 - Seleccionar pista de audio: el usuario podrá en todo momento cambiar la pista de audio, siempre y cuando el contenido multimedia disponga de múltiples pistas de audio.
 - Seleccionar pista de subtítulo: el usuario podrá en todo momento cambiar la pista de subtítulo, siempre y cuando el contenido multimedia disponga de múltiples pistas de subtítulo.
 - Seleccionar capítulo: el usuario podrá en todo momento cambiar de un capítulo a otro, siempre y cuando el contenido multimedia contenga capítulos configurados.
 - Activar/Desactivar audiodescripción: si dicha funcionalidad está agregada en el reproductor, cuando el botón de la audiodescripción es activado, pasará a reproducirse automáticamente dicha pista (siempre por defecto será la pista número 1 del contenido multimedia), sin importar qué pista de audio esté seleccionada. Cuando se desactiva la audiodescripción, automáticamente pasará a reproducirse la pista de audio que estuviera seleccionada antes de activarla.
 - Activar/Desactivar subtítulos: si dicha funcionalidad está agregada en el reproductor, cuando el botón de subtítulos es activado, la pista de
-

subtitulado seleccionada actualmente se mostrará en pantalla. Cuando se desactivan los subtítulos, ningún subtítulo será mostrado en pantalla, a pesar de que haya alguno seleccionado.

- Mostrar atajos de teclado: aparecerá un pop up con todos los atajos de teclado disponibles en el reproductor.
- Mostrar ayuda: aparecerá un pop up en el que se mostrará información general de la aplicación.

5.2.1 Interfaz del Editor

La interfaz del editor englobará el conjunto de vistas que se muestra en la Ilustración 8:

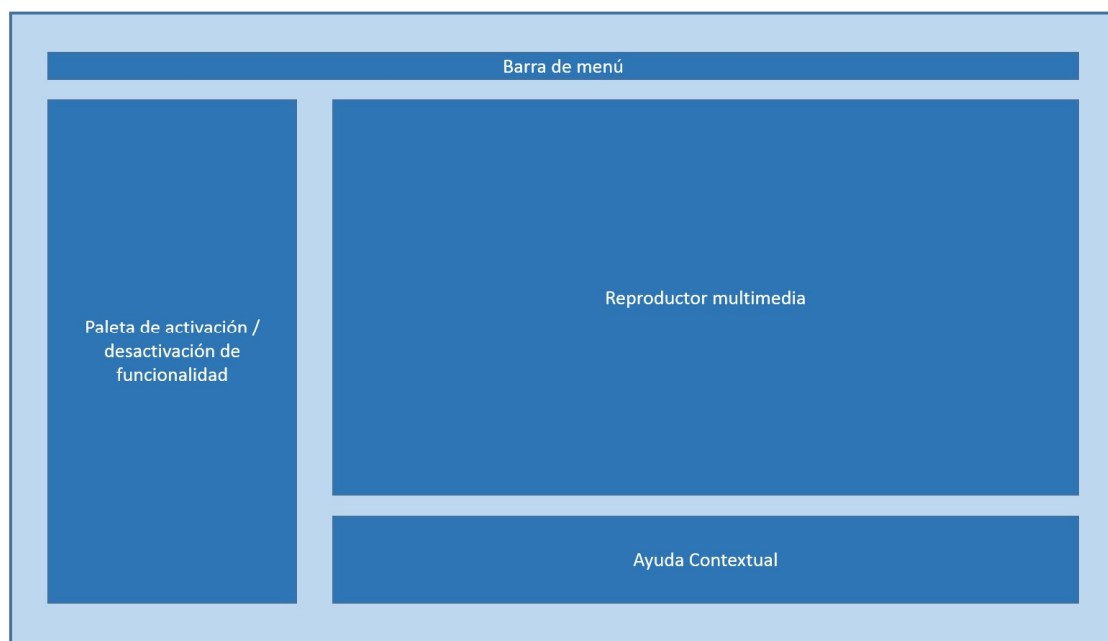


Ilustración 8 - Esquema de la interfaz del Editor Multimedia

- Barra de menú: a través de esta vista se facilitarán al usuario opciones para comprobar la compatibilidad de la configuración actual del reproductor con los perfiles personalizados, gestionar los perfiles, y demás casos de uso mencionados anteriormente.

- Paleta de activación/desactivación de funcionalidad: este componente permitirá activar y desactivar en el reproductor cualquier funcionalidad, como por ejemplo: detener, reproducir, rebobinar, avanzar, ajustar volumen, etc. Si por ejemplo se pulsa el botón de reproducir y éste no estaba en el reproductor, aparecerá, en caso contrario, desaparecerá del reproductor.
- Reproductor multimedia: vista previa de la configuración actual del reproductor. Dicha vista será totalmente funcional (ver sección Interfaz del Reproductor).
- Ayuda Contextual: vista en la cual se mostrará al usuario información acerca de las últimas acciones realizadas en el editor (ver sección Interfaz de la Ayuda Contextual).

5.2.2 Interfaz del Reproductor

La interfaz del reproductor probablemente sea la más compleja del prototipo, puesto que debe contener todos los elementos del reproductor multimedia. Además, debe ser capaz de proporcionar la funcionalidad completa para la reproducción del contenido multimedia.

La pantalla deberá tener un aspecto similar al que se muestra a continuación en la Ilustración 9:

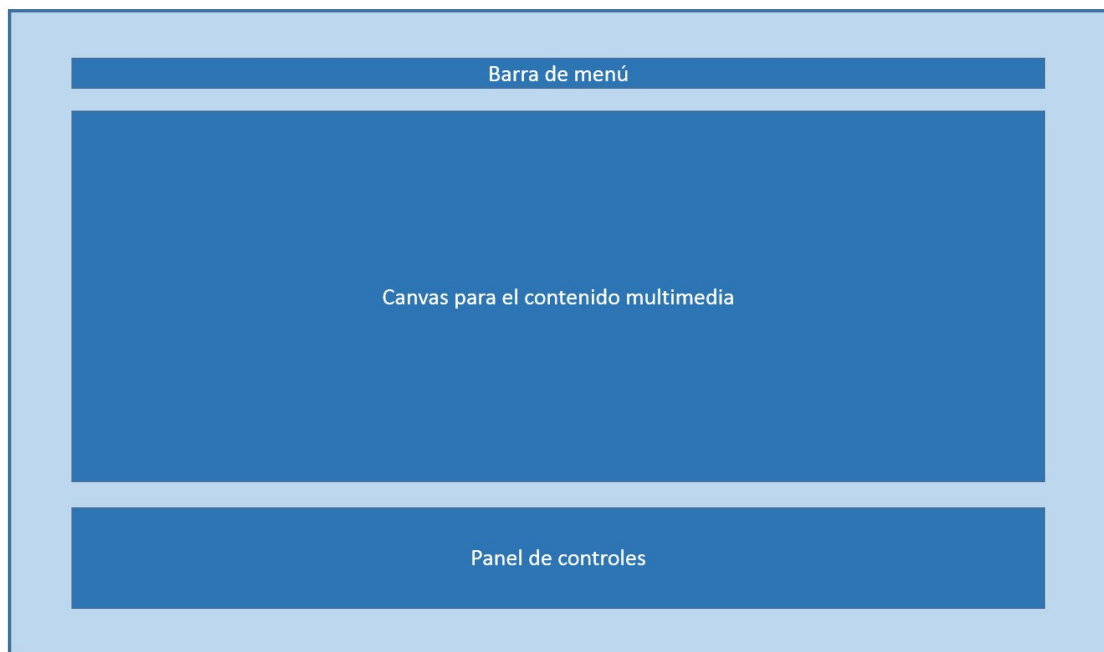


Ilustración 9 - Esquema de la Interfaz del Reproductor

Como puede observarse, el reproductor constará de tres partes principales:

- Barra de menú: en ella se facilitarán al usuario opciones para abrir y cerrar contenido multimedia, así como la selección de pistas de audio, subtítulos y capítulos, y también opciones de ayuda, como mostrar los atajos de teclado del reproductor o información general del mismo.
- Canvas para el contenido multimedia: en este componente se mostrará el contenido multimedia.
- Panel de controles: constará de todos los controles necesarios para manejar la funcionalidad completa del reproductor.

5.2.3 Interfaz de la Ayuda Contextual

La interfaz de la Ayuda Contextual es un panel en el cual se mostrará al usuario información relevante.

Para este prototipo no es necesario que este componente tenga mucha más funcionalidad, en cambio, se propondrán al respecto una serie de líneas futuras para potenciar esta vista sustancialmente, en el apartado Futuras líneas de trabajo.

La Ilustración 10 mostrada a continuación muestra el aspecto de esta vista:

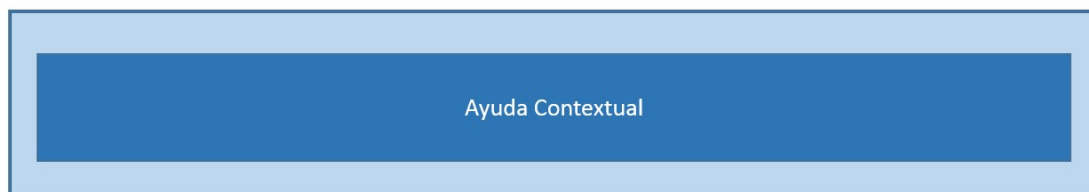


Ilustración 10 - Esquema de la Interfaz de la Ayuda Contextual

Como puede observarse, la ayuda constará de un único panel, dónde se mostrarán las últimas acciones del usuario, así como información relevante durante la utilización del editor.

5.3 Diseño de la lógica de la aplicación

En esta sección se tratará de identificar todas aquellas operaciones que corren por debajo de la interfaz, para que la aplicación funcione cómo se espera.

A continuación se detalla la lógica de los procesos identificados:

- Activación de funcionalidad en el reproductor
- Gestión de perfiles personalizados
- Reproducción multimedia
- Comprobación de compatibilidad con perfiles que contienen los elementos necesarios para permitir el acceso dependiendo de las características del usuario final.

5.2.1 Lógica de activación de funcionalidad en el reproductor

Esta parte de la lógica de la aplicación se centra en la interacción con el usuario del editor, gracias a la cual éste será capaz de añadir o quitar funcionalidad para cumplir con sus necesidades.

Para ello, será necesario crear una paleta que permita dicho manejo, y que estará ligada directamente con el panel de controles del reproductor, de tal forma que cuándo se active por ejemplo el botón de “reproducir”, aparezca en el reproductor en caso que no estuviera, o bien desaparezca, en caso que ya estuviera presente en el mismo.

Dicha paleta, a parte de la funcionalidad básica, además incluirá la opción de activar todos los botones a la vez, o bien desactivarlos.

5.2.2 Lógica de la gestión de perfiles personalizados

Para poder brindar al usuario la posibilidad de guardar y cargar sus propias configuraciones para el reproductor multimedia, será necesario crear un sistema que permita la gestión de perfiles personalizados.

Será necesario también proporcionar algún método o función mediante el cual la interfaz gráfica pueda cargar los valores de un perfil, o bien volcar la información a disco en caso que se desee guardar un perfil para su futura utilización o edición.

5.2.3 Lógica de la reproducción multimedia

La lógica de la reproducción multimedia es el pilar básico en el que se basa el prototipo, puesto que al fin y al cabo, el resultado será un reproductor multimedia personalizado y capaz de mostrar contenido multimedia de la forma más accesible posible.

La capacidad de reproducción de contenido multimedia se basará en la librería *libVLC*, librería nativa que en el caso del prototipo se centra en sistemas Windows de 64 bits.

La gran ventaja de esta librería, en la cual se basa el popular reproductor multimedia VLC, es que incluye soporte directo para la reproducción de contenido en prácticamente cualquier formato de vídeo codificado con cualquier tipo de códec, sin la necesidad de tener que instalar dichos códec independiente en el sistema operativo.

Para el acceso a esta librería nativa de Windows, se hará uso del Proyecto VLCJ. Este proyecto proporciona acceso a prácticamente toda la funcionalidad nativa de la librería *libVLC* a través del API, gracias a una interfaz nativa de Java (JNI). Es un framework de alto nivel, maduro y robusto que encapsula el acceso a la librería nativa, que además da protección frente a usos incorrectos de la misma, librando así de posibles errores en tiempo de ejecución.

5.2.4 Lógica de la comprobación de compatibilidad

La idea de este prototipo se basa en brindar al usuario la posibilidad de crear reproductores multimedia personalizados con características de accesibilidad.

Por ello, en todo momento el usuario podrá comprobar si la configuración actual es compatible con algunos de los perfiles preestablecidos teniendo en cuenta la edad o el tipo de discapacidad del usuario final, como por ejemplo:

- Reproductor para usuarios con discapacidad auditiva.
- Reproductor para usuarios con discapacidad visual.
- Reproductor simplificado: únicamente funcionalidades básicas.
- Reproductor general: todas las funciones.
- Reproductor por defecto: únicamente barra de progreso.

Esta comprobación se realizará de forma sencilla, a través de la clase gestora de perfiles de usuario, chequeando que los parámetros necesarios para cumplir estos perfiles estén presentes o no en el reproductor personalizado. Para ello, se le proporcionará la opción “Check Accessibility” al usuario en el menú principal.

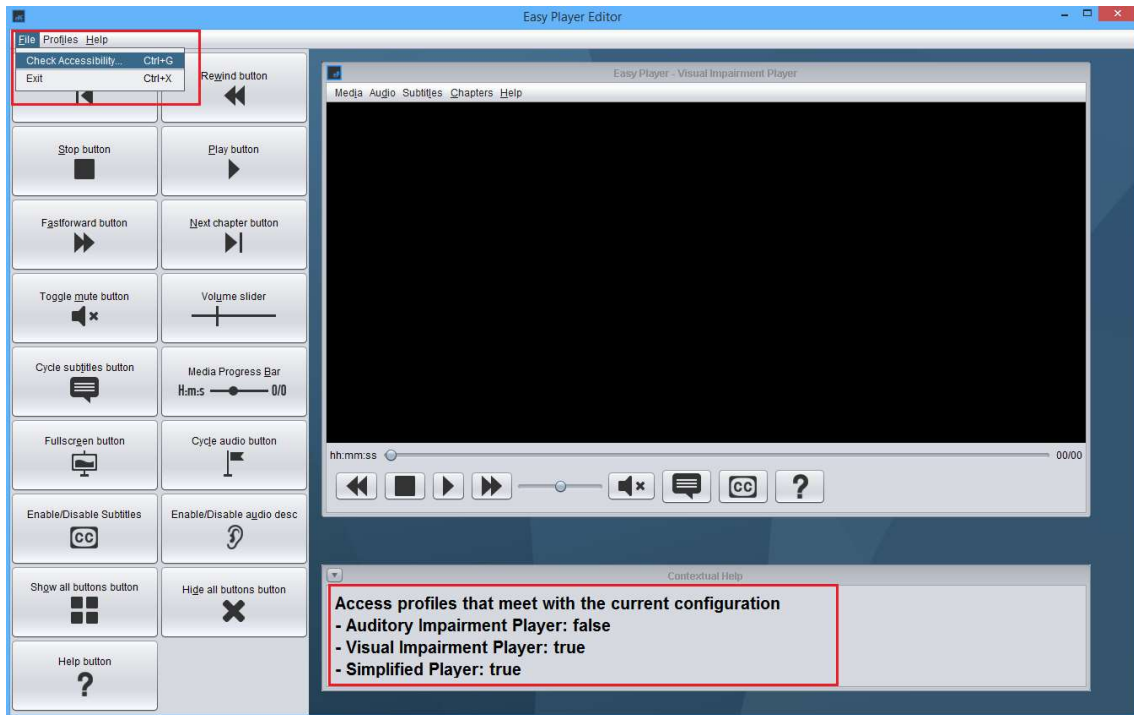


Ilustración 11 - Comprobación de Compatibilidad de Accesibilidad

Implementación

6. Implementación

En este capítulo se detallará la fase de implementación del prototipo para el editor de reproductores personalizados según las características de acceso **Easy Player Editor**, a partir de los diseños especificados anteriormente.

Llegados a este punto, es muy importante elegir la tecnología y frameworks más adecuados, en base al estudio que se realizó en el Estado del Arte.

El lenguaje de programación elegido es *Java*, el cual permitirá que simplemente retocando una pequeña parte del código, este prototipo sea portable, por ejemplo, a otras plataformas, así como tiene capacidad para desarrollar de forma sencilla futuras actualizaciones de la aplicación.

Para la interfaz gráfica se utilizará el framework *Swing* de Java, dado que es altamente extensible y personalizable, permitiendo un desarrollo de componentes más activo.

Una vez elegido el lenguaje y el framework del apartado gráfico, es vital elegir un motor potente y funcional para la reproducción del contenido multimedia. Por ello, y basándonos en el apartado del estado del arte Tecnologías, se ha decidido que toda la funcionalidad de reproducción multimedia se llevará a cabo a través del API proporcionado por el framework *VLJ*. El motivo de esta decisión es el siguiente, el soporte para reproducir contenido multimedia, especialmente vídeo, generalmente es pobre en la plataforma Java. Java Media Framework se puede utilizar para reproducir contenidos, pero es un framework antiguo con pocas actualizaciones, además de contar con bastantes limitaciones en cuanto a reproducción de contenidos de diferentes formatos y códec. Por otro lado, el framework JavaFX para reproducir contenido multimedia, también cuenta con bastantes limitaciones.

Por estos motivos, la solución más recurrente se basa en utilizar frameworks multimedia nativos y mapearlos a través de librerías nativas utilizando Java Native Interface [\[2\]](#) (JNI), y por ello, la mejor opción en este caso reside en el

framework VLCJ, el cual proporciona gran potencia para la reproducción de contenido multimedia de una forma eficaz y segura.

6.1 Implementación de la interfaz gráfica

La implementación de la interfaz gráfica del Easy Player Editor se ha desarrollado en el paquete java llamado `es.uc3m.labda.easyplayereditor`.

pkg `es.uc3m.labda.easyplayereditor`

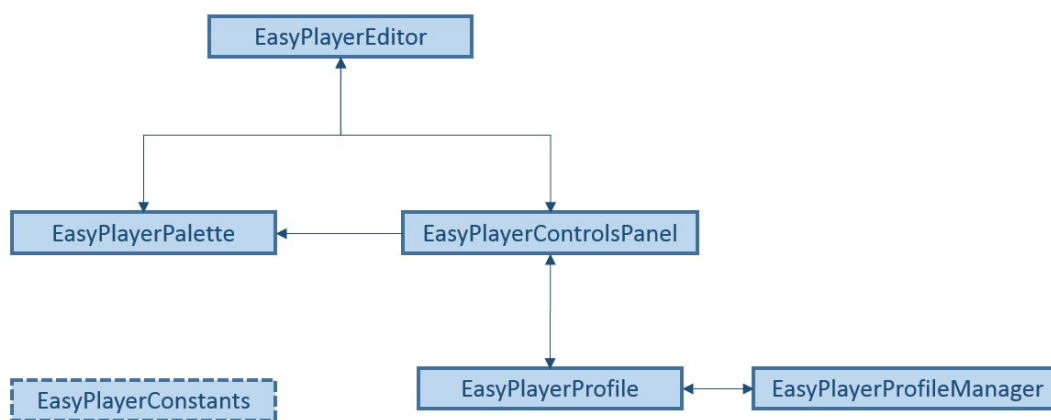


Ilustración 12 - Esquema de Clases de la Aplicación

A continuación, se procederá a detallar más técnicamente las diferentes pantallas de la aplicación.

6.1.1 Pantalla de Selección de Perfil Inicial

Tal y cómo se indica en el diseño, se han especificado en la aplicación una serie de perfiles preestablecidos:

- Reproductor para usuarios con discapacidad auditiva.
- Reproductor para usuarios con discapacidad visual.
- Reproductor simplificado: únicamente funciones básicas.
- Reproductor general: todas las funciones.

- Reproductor por defecto: únicamente barra de progreso.

Al arrancar la aplicación, se le da al usuario la posibilidad de cargar uno de estos perfiles, para poder editar a partir de dicha funcionalidad, o bien, cargar el perfil por defecto, y que sea el usuario el que active la funcionalidad deseada.

En la siguiente ilustración se muestra la ventana de selección del perfil por defecto, y cómo el editor es presentado a continuación:

Selección inicial de perfil por defecto

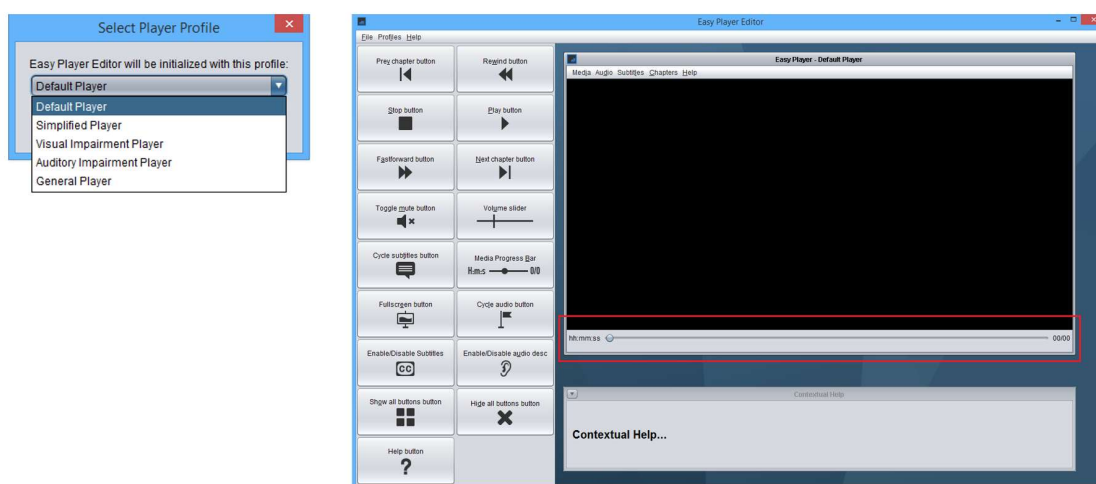


Ilustración 13 - Selección inicial de Perfil por defecto

Como puede observarse, cuando se selecciona inicialmente el perfil por defecto, el reproductor multimedia del editor se carga únicamente con la barra de progreso, elemento que a pesar de aparecer por defecto en todos los perfiles preestablecidos, se puede eliminar si el usuario lo desea, puesto que la paleta de controles ofrece dicha posibilidad. A partir de ahí, el usuario podrá incluir toda la funcionalidad que necesite. En cambio, cuándo se selecciona por ejemplo el perfil para usuarios con discapacidad visual, el panel de control que aparece en el reproductor está compuesto por aquellas funciones básicas mínimas para que dichos usuarios puedan utilizar el reproductor de forma más accesible (ver Ilustración 14).

Selección inicial de perfil para usuarios con limitación visual

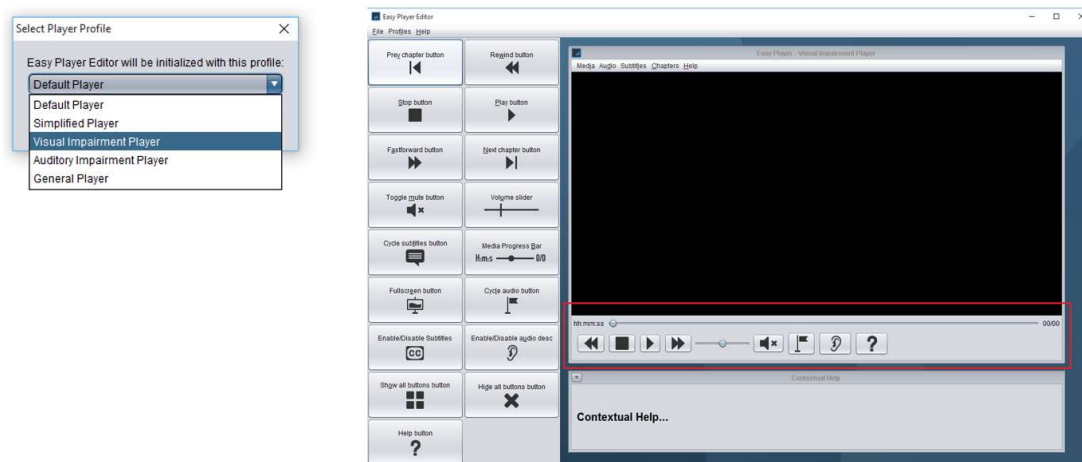


Ilustración 14 - Selección inicial de Perfil Visual

6.1.2 Paleta de Activación de Funcionalidad

Para la implementación de este componente básico del editor se ha desarrollado la clase Java `EasyPlayerControlsPalette.java`.

Cuando se construye una instancia de esta clase, se pasa como parámetro un objeto de la clase `EasyPlayerControlsPanel` de tal forma que podamos manejar las operaciones de activación o desactivación de la funcionalidad que necesite el usuario sobre ese propio panel de controles embebido en el reproductor multimedia.

Este es el aspecto de la paleta, con todas las opciones que ofrece:



Ilustración 15 - Paleta de Opciones del Editor

Cuándo se pulsa un botón de esta paleta, por ejemplo el botón de reproducir (“Play button”), pueden ocurrir dos cosas:

- Si en el reproductor ya estaba el botón de reproducir, éste desaparece.
- Si en el reproductor no estaba el botón de reproducir, éste aparece.

Y así sucesivamente con cada una de las opciones que ofrece dicha paleta.

Cabe destacar dos opciones:

- Existe un botón para activar todas las opciones de reproducción (“Show all buttons button”)
- Existe un botón para desactivar todas las opciones de reproducción (“Hide all buttons button”)

Todas y cada una de las opciones de la paleta, son accesibles adicionalmente a través de atajos de teclado presionando `Alt+<Tecla>`. Dicha funcionalidad puede apreciarse en la etiqueta que aparece al dejar el ratón detenido sobre un botón, o bien, se corresponderá con la letra subrayada del título del botón:

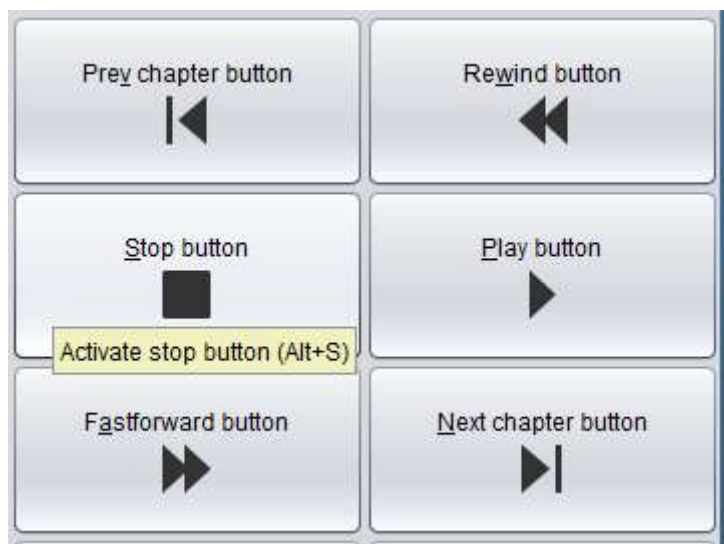


Ilustración 16 - Ejemplo de Etiqueta en la Paleta de Opciones del Editor

6.1.3 Panel de Controles de Reproducción

Para la implementación de este componente básico del reproductor, se ha desarrollado la clase Java `EasyPlayerControlsPanel.java`.

Cuando se construye una instancia de esta clase, se pasa como parámetro un objeto de la clase `EasyPlayerProfile`, así como un objeto de la clase `EmbeddedMediaPlayer`.

La clase `EasyPlayerProfile` indica si cada una de las posibles opciones de reproducción está activada o no.

Por otro lado, la clase `EmbeddedMediaPlayer` (que pertenece al framework VLCJ) permitirá operar directamente sobre el contenido multimedia que se cargue en el reproductor cuando el usuario accione los botones del reproductor multimedia, por ejemplo, cuando pulse el botón de "Silenciar", el valor del volumen tomará el valor de

0, y se dejará de escuchar el audio. Cuando se vuelva a pulsar, el audio se restablecerá a su valor anterior.

El funcionamiento del API ofrecido por VLCJ para interactuar con el contenido multimedia cargado, se basa en callbacks asíncronas, por ello, cuando se pulsa un botón del panel de controles del reproductor, saltará el listener asociado a dicho botón, y es ahí donde se interactúa con el contenido multimedia a través de VLCJ, mandando el evento correspondiente.

Cabe destacar que cuando un contenido multimedia está en reproducción y se pulsa de nuevo el botón de reproducir, el contenido quedará pausado, y cuando se vuelva a pulsar dicho botón, la reproducción continuará.

En la Ilustración 17 se muestra el aspecto del panel de controles del reproductor con la funcionalidad al completo:



Ilustración 17 - Ejemplo de Panel de Controles del Reproductor

Del panel de controles del reproductor se destaca lo siguiente:

- Cuando el botón de activación de subtítulos (“CC”) se activa, se marca de color azul y es entonces cuando se habilitan los subtítulos en pantalla, si no, los subtítulos no son mostrados.
- Cuando el botón de activación de audiodescripción (“oreja”) se activa, se marca de color azul, y automáticamente se cambia a la pista de audiodescripción. Cuando se desactiva, se vuelve a la pista que estuviera seleccionada anteriormente.
- En el prototipo no se ha implementado la funcionalidad de reproducción en pantalla completa, funcionalidad que será comentada en la sección Futuras líneas de trabajo.

Todas y cada una de las opciones de la paleta, son accesibles adicionalmente a través de atajos de teclado presionando `Alt+<Tecla>`. Dicha funcionalidad puede apreciarse en la etiqueta que aparecerá al dejar el ratón detenido sobre un botón, como muestra la ilustración 18:



Ilustración 18 - Ejemplo de Botones accesibles por combinación de teclado

6.1.4 Pantalla del Reproductor Multimedia

Si bien se podría haber creado una clase específica para el reproductor, puesto que tiene entidad propia, para este prototipo se ha decidido desarrollar esta parte directamente en la clase `EasyPlayerEditor.java`.

Para ello se ha empleado un `JInternalFrame` de Java (o frame secundario dentro del frame del editor), el cual va a contener los siguientes componentes:

- Un objeto `Canvas` de Java, como superficie en la que se mostrará el contenido multimedia.
- Un objeto `EasyPlayerControlsPanel` del paquete `es.uc3m.labda.easyplayereditor`, el cual proporciona al usuario la capacidad de operar sobre el contenido multimedia.
- Un objeto `JMenuBar` de Java, gracias al cual se ofrecerá otro tipo de funcionalidad.

El aspecto del reproductor multimedia en funcionamiento puede observarse en la Ilustración 19:



Ilustración 19 - Ejemplo de Reproductor Multimedia

A través de la barra de menú, el usuario puede abrir y cerrar contenido multimedia cuando así lo desee (ver ilustración 20):

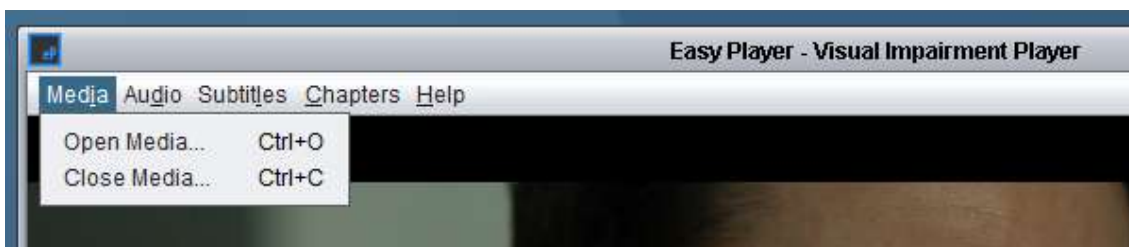


Ilustración 20 - Barra de Menú del Reproductor

Todas y cada una de las opciones de la barra de menú del reproductor, son accesibles adicionalmente a través de atajos de teclado, dichos atajos pueden verse al lado del texto de cada opción. Para ello se utiliza el método Java para los elementos de menú “setAccelerator” a través del cual se puede indicar la combinación de teclas que acciona dicho componente.

En el menú de ayuda del reproductor, se puede ver las combinaciones disponibles (ver ilustración 21):



Ilustración 21 - Menú Ayuda del Reproductor

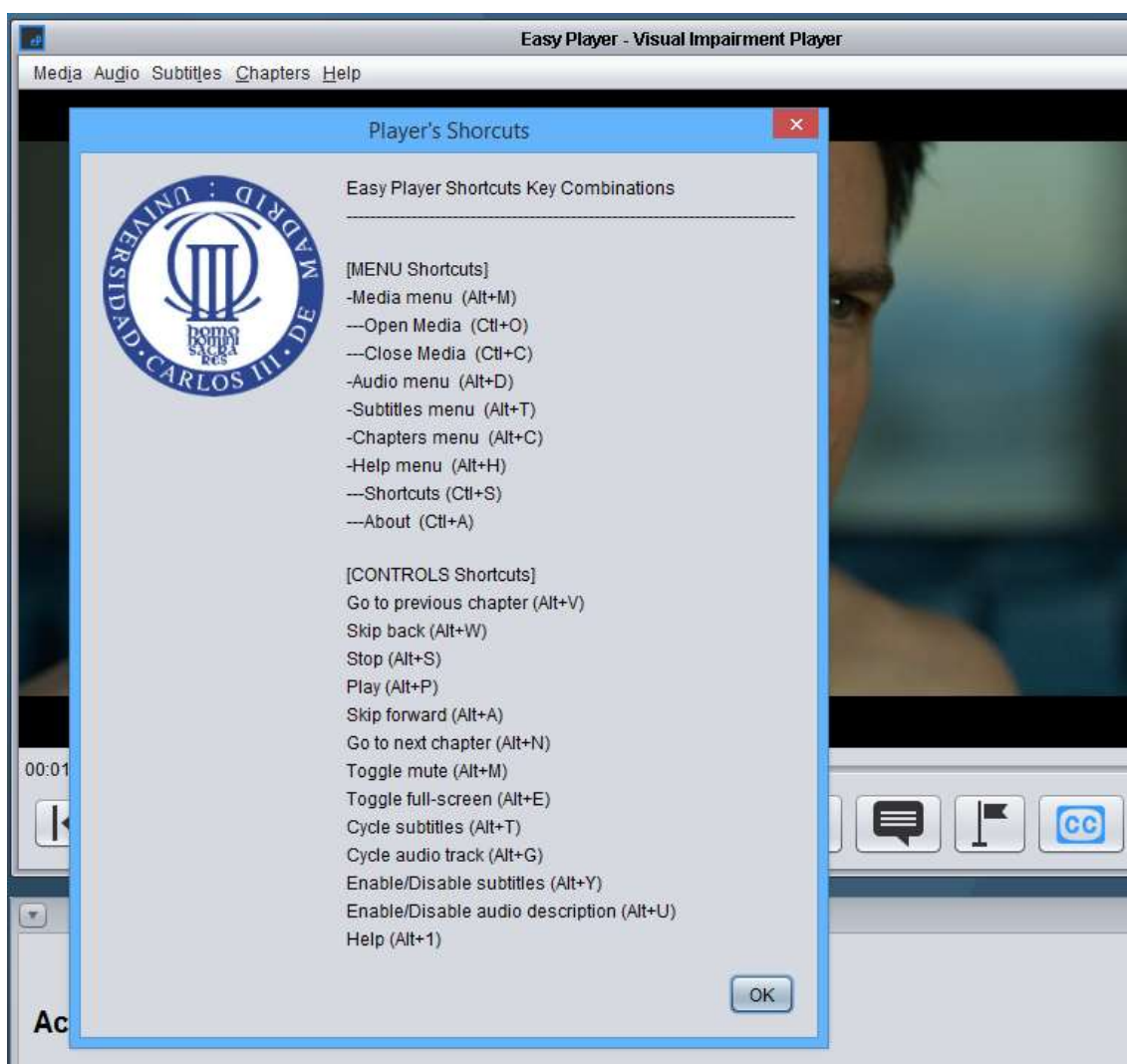


Ilustración 22 - Ayuda del Reproductor

Cuando se abre un nuevo contenido multimedia con múltiples pistas de audio, subtítulos o capítulos, se recorre cada pista, y se reconstruye la barra de menú del reproductor desde cero, de tal forma que los submenús de Audio, Subtítulos y Capítulos muestran los valores asociados al contenido del vídeo y se reconfigura la interacción de dichos botones con el objeto multimedia `EmbeddedMediaPlayer` de VLCJ, que es el objeto que contiene el contenido. En las siguientes ilustraciones se muestra el menú de Audio, Subtítulos y Capítulos descritos anteriormente:

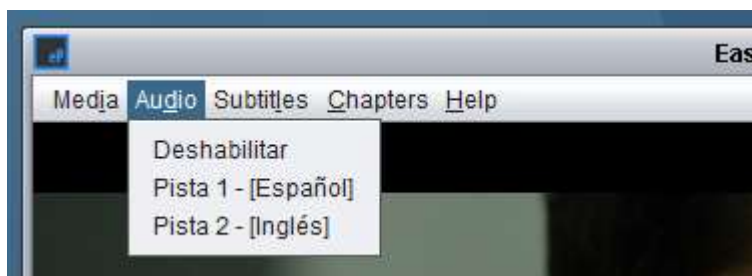


Ilustración 23 - Menú de Audio del Reproductor



Ilustración 24 - Menú de Subtítulos del Reproductor

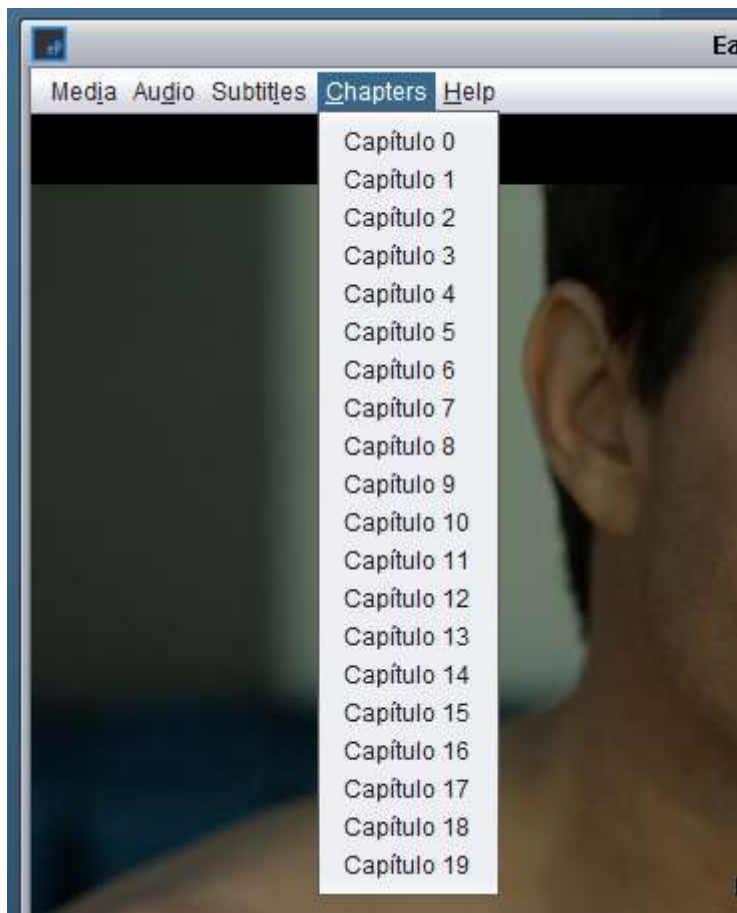


Ilustración 25 - Menú de Capítulos del Reproductor

6.1.5 Pantalla de la Ayuda Contextual

Al igual que con la pantalla del reproductor, en este caso se podría haber diseñado una clase específica para la Ayuda Contextual, pero para este prototipo se ha decidido desarrollar esta parte directamente en la clase `EasyPlayerEditor.java`.

Para ello se ha empleado otro `JInternalFrame` de Java, el cual contiene un panel con una etiqueta que va actualizando su valor a medida que el usuario interactúa con la aplicación.

En este prototipo, la funcionalidad de la ayuda contextual es muy sencilla, pero esta funcionalidad podría potenciarse con alguna idea que será incluida en la sección Futuras líneas de trabajo.

6.1.6 Pantalla del Editor

La pantalla del editor **Easy Player Editor** es la vista de la aplicación de forma completa.

Esta vista se basa en la clase `JFrame` de Java, y maqueta de forma completa todos los componentes, o vistas, de la aplicación:

- Una barra de menú basada en la clase `JMenuBar` de Java.
- Una paleta para la activación de funcionalidad, a través de una instanciación de la clase `EasyPlayerControlsPalette`.
- Una ventana interna para el reproductor multimedia, basada en la clase `JInternalFrame` de Java.
- Una ventana para la ayuda contextual, basada también en la clase `JInternalFrame` de Java.

En la Ilustración 26 se muestra el aspecto general del editor:

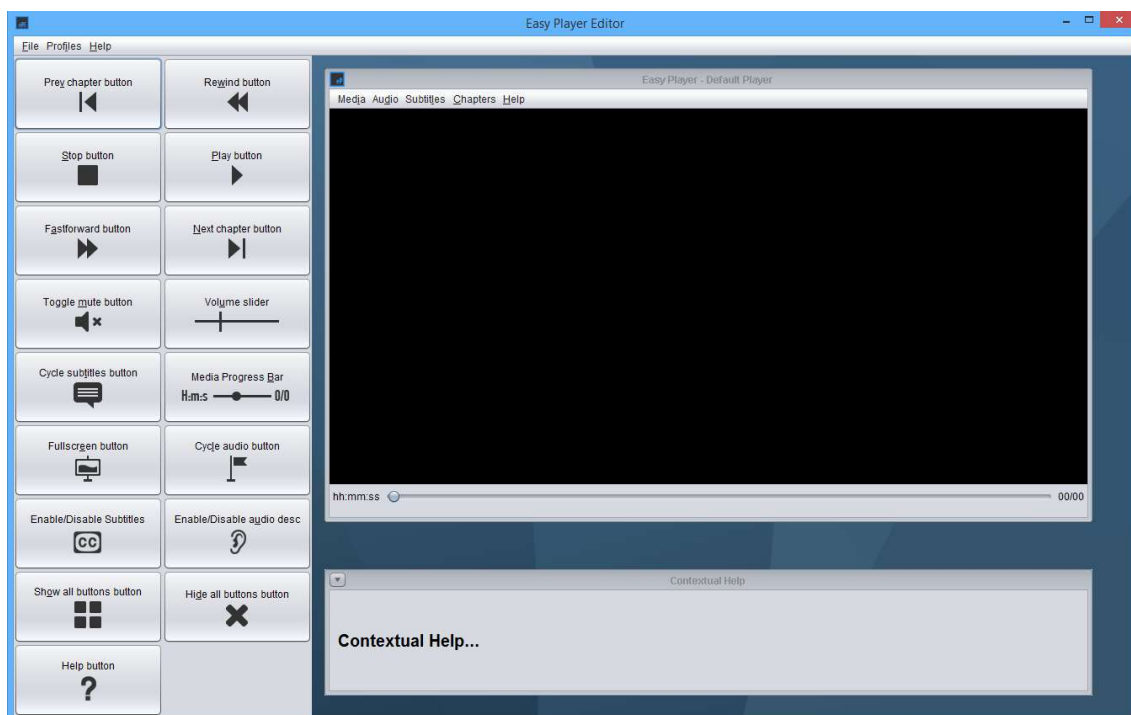


Ilustración 26 - Vista general del Editor

En los apartados anteriores se ha comentado la implementación de cada una de las vistas de la interfaz, por lo que no se hará mayor hincapié en ello aquí.

El editor permite, tal y como se especifica en la fase de diseño del prototipo, la gestión de perfiles personalizados. Es decir, un usuario puede configurar un reproductor, y guardarlo en disco, para compartirlo con otros usuarios, o bien para su futura modificación. Del mismo modo, se permite la carga de perfiles personalizados a partir de ficheros en disco guardados previamente desde el editor.

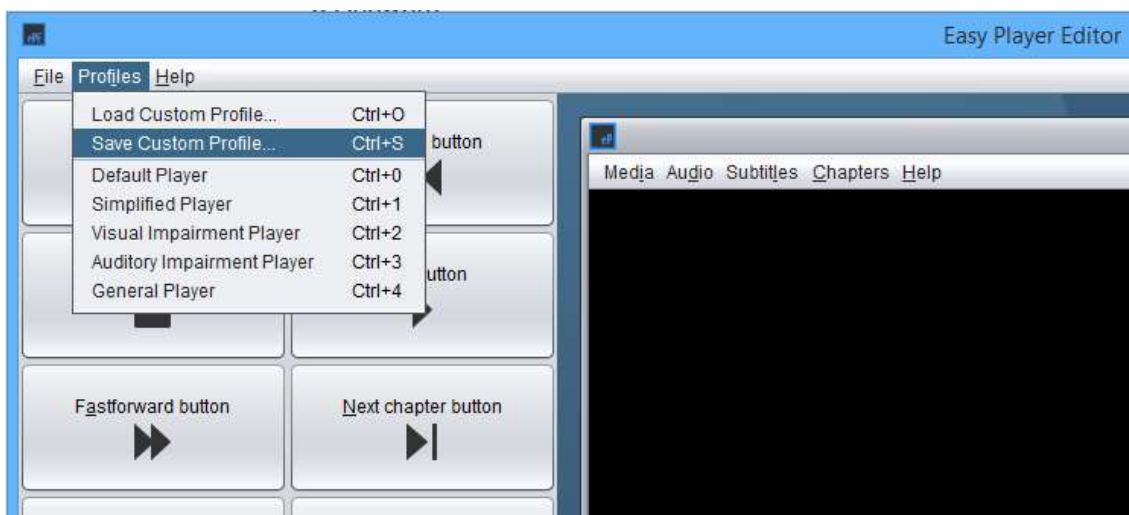


Ilustración 27 - Menú de Perfiles del Editor

Cuando se desea guardar un perfil, aparecerá una ventana en la que se pide el nombre del Perfil, el cual no tiene por qué coincidir con el nombre del fichero que se guardará en disco. El nombre del Perfil es el nombre que se le dará a una variable interna del perfil para identificarlo dentro de la aplicación:



Ilustración 28 - Asignación del nombre del perfil

Y a continuación, una vez pulsado el botón "OK", aparecerá un explorador para guardar el fichero de propiedades del perfil en disco, con el nombre que queramos:

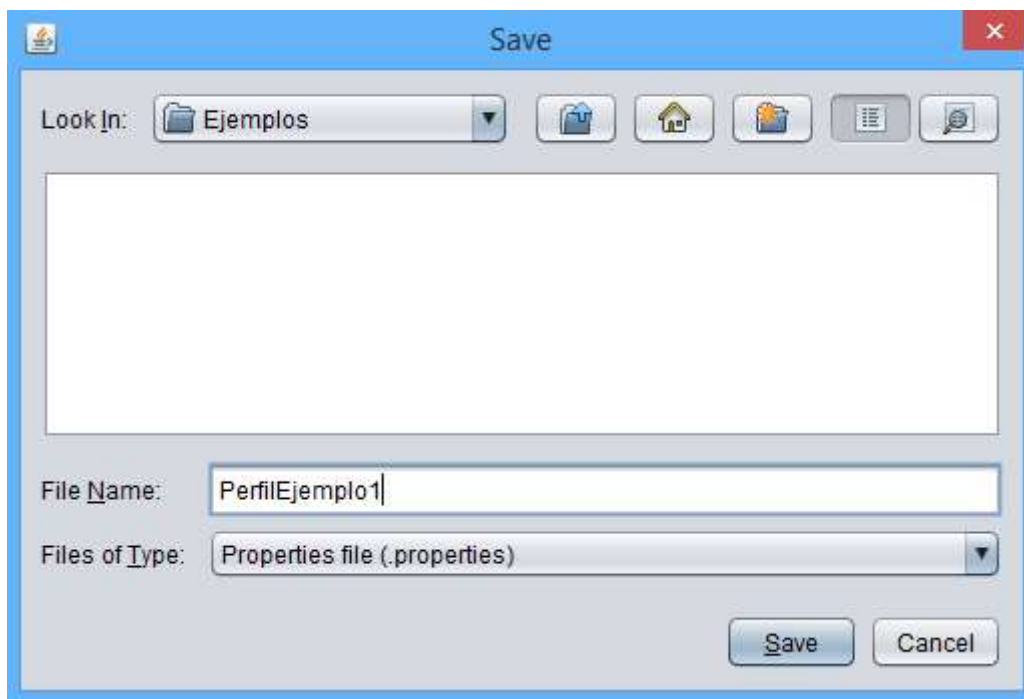


Ilustración 29 - Ventana para guardar el fichero de propiedades del perfil en disco

En caso de no introducir en el nombre del fichero la extensión .properties, la aplicación la incluye automáticamente.

En la ayuda contextual se mostrará el siguiente mensaje:



Ilustración 30 - Mensaje de Ayuda Contextual al guardar perfil

Y como resultado, en la carpeta de destino aparecerá dicho fichero:

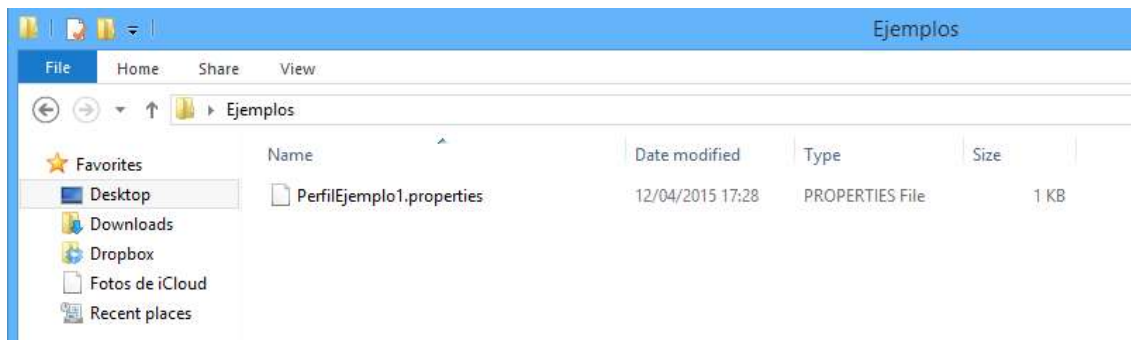


Ilustración 31 - Fichero de propiedades del perfil almacenado en disco

Con el siguiente contenido:

```
PerfilEjemplo1.properties x
1 #Sun Apr 12 17:28:07 CEST 2015
2 profile.help=true
3 profile.subtitles.cycle=true
4 profile.next.chapter=true
5 profile.toggle.mute=true
6 profile.toggle.subtitles=true
7 profile.fullscreen=true
8 profile.stop=true
9 profile.toggle.audioDesc=true
10 profile.previous.chapter=true
11 profile.fastforward=true
12 profile.audio.cycle=true
13 profile.name=Ejemplo1
14 profile.volume.slider=true
15 profile.rewind=true
16 profile.play=true
17
```

Ilustración 32 - Ejemplo de fichero de propiedades de un Perfil

Se trata de un fichero Properties de Java que guarda los atributos que necesita el objeto `EasyPlayerProfile` para ser gestionado por la aplicación.

Del mismo modo, dicho perfil podrá ser cargado más adelante a través de la opción “Load Custom Profile” del menú Profiles (ver ilustración 33):

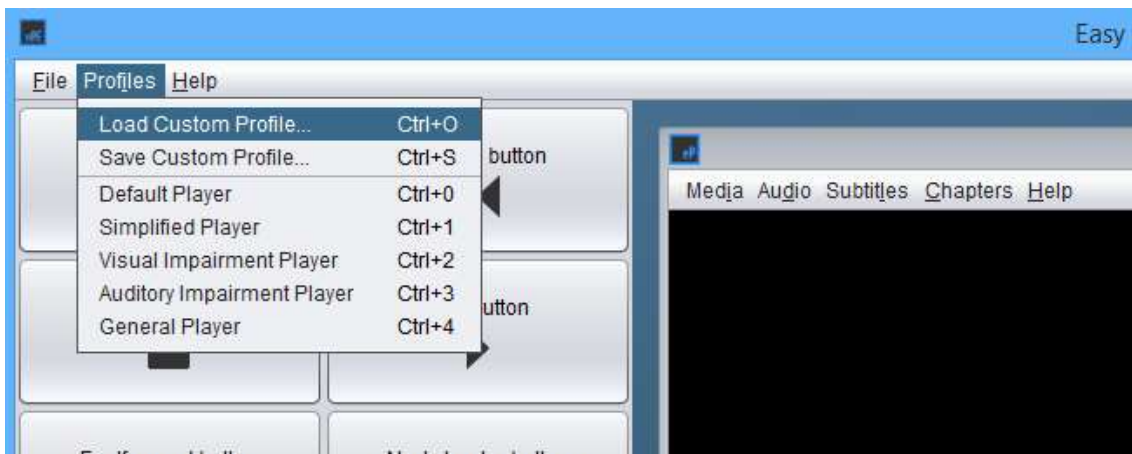


Ilustración 33 - Menú de perfiles del Editor

Una vez cargado el perfil, la ayuda contextual mostrará el siguiente mensaje:

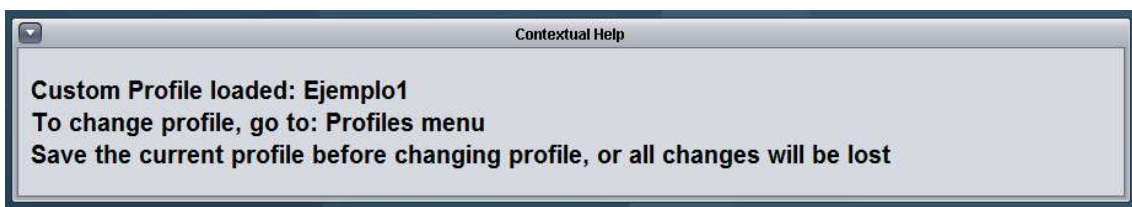


Ilustración 34 - Mensaje de ejemplo de la Ayuda Contextual al cargar perfil

Cuándo se carga un perfil nuevo, siempre se muestra en la ayuda un mensaje indicando que antes de cambiar de perfil, se guarde el contenido, o de lo contrario se perderá.

Todas y cada una de las opciones de la barra de menú del editor, son accesibles adicionalmente a través de atajos de teclado, dichos atajos pueden verse al lado del texto de cada opción.

Para ello, y al igual que en el caso de la ventana del reproductor, se utiliza el método Java para los elementos de menú "setAccelerator", a través del cual se puede indicar la combinación de teclas que acciona dicho componente.

También se ha incluido la opción de mostrar todos los atajos disponibles en el reproductor en el menú de ayuda del editor.



Ilustración 35 - Menú de Ayuda del Editor

En este caso, se ha optado por mostrar una ventana de pop up con toda la información a través del método “showMessageDialog” de la clase JOptionPane, método del framework Swing muy utilizado para este tipo de mensajes.

6.2 Implementación de la lógica de la aplicación

La lógica de la aplicación se divide en:

- Lógica de activación de funcionalidad en el reproductor.
- Lógica de gestión de perfiles
- Lógica de la reproducción multimedia
- Lógica de la comprobación de compatibilidad

Cabe destacar, que no se ha creado un paquete Java diferente para separar entre lógica e interfaz gráfica a la hora de desarrollar este prototipo, a pesar de que hubiera sido una buena idea. Este punto será incluido en la sección de Futuras líneas de trabajo, para sugerir realizar una refactorización del código en futuras versiones.

6.2.1 Lógica de activación de funcionalidad en el reproductor

Puesto que la lógica de la activación de la funcionalidad del reproductor en el editor va muy ligada a la interfaz gráfica, esta sección queda cubierta en su totalidad en el apartado Paleta de Activación de Funcionalidad.

6.2.2 Lógica de gestión de perfiles

Para gestionar los perfiles personalizados, se han desarrollado las clases `EasyPlayerProfile` y `EasyPlayerProfileManager`.

La primera clase (`EasyPlayerProfile`) cuenta con un booleano para cada funcionalidad del reproductor. Cuando se crea una instancia de la clase `EasyPlayerControlsPanel` (panel de controles para el reproductor) para construir el reproductor, se le pasa como parámetro un objeto de tipo `EasyPlayerProfile`, y cada vez que se activa o desactiva funcionalidad en el reproductor multimedia, dicho perfil es actualizado.

La clase `EasyPlayerProfileManager` ofrece más funcionalidad relacionada con los perfiles. Esta clase es un gestor de perfiles, que cuando es instanciada guarda un mapa con todos los perfiles por defecto, a los cuales se puede acceder a través del método `getProfile`, que devuelve una instancia de la clase `EasyPlayerProfile`, con la configuración de ese perfil personalizado.

Adicionalmente, permite transformar de un fichero de `properties` (con el formato adecuado) a un objeto de tipo `EasyPlayerProfile`, y viceversa. De esta forma, en todo momento se puede editar un reproductor multimedia, y guardar un perfil con dicha configuración, o del mismo modo, cargar un perfil determinado presente en disco.

Los perfiles personalizados se guardan en el disco duro en forma de un fichero de `properties` de Java, tal y como se describe en la sección de implementación Pantalla del Editor.

6.2.3 Lógica de la reproducción multimedia

Esta parte de la lógica se centra en el reproductor multimedia, cuya funcionalidad, como se ha comentado en secciones anteriores, está basada en la utilización del framework VLCJ.

Cada vez que se selecciona un archivo multimedia, se instancia un objeto de tipo `EmbeddedMediaPlayer` proporcionado por el framework de VLCJ, el cual es capaz de ejecutar acciones sobre el contenido multimedia.

La clase `EasyPlayerControlsPanel` tiene un listener por cada funcionalidad proporcionada por el reproductor: reproducir, rebobinar, parar, seleccionar pista de subtítulo, etc. Cuando un listener es accionado por el usuario, se dispara un evento asíncrono que es gestionado por la instancia `EmbeddedMediaPlayer` ejecutando la acción deseada sobre el contenido multimedia.

Para conseguir alguna funcionalidad concreta, como la modificación del volumen de audio con una barra de selección de tipo `JSlider` de Java, ha sido necesario añadir código para gestionar el desplazamiento con el ratón por la barra. Una vez se selecciona el valor deseado, se manda el evento a la instancia del objeto `EmbeddedMediaPlayer`.

Gracias al framework VLCJ, todas las acciones se ejecutan de forma asíncrona y controlada.

6.2.4 Lógica de la comprobación de compatibilidad

Cuando se crea toda la interfaz gráfica, se instancian una serie de objetos, entre ellos uno de los más importantes es el `EasyPlayerControlsPanel`.

Este objeto contiene a su vez un atributo principal, el cual es una instancia del objeto `EasyPlayerProfile`. Como se ha comentado anteriormente, dicho objeto es una clase básica de Java que guarda un atributo booleano por cada funcionalidad capaz de ser suministrada por el reproductor, así como el nombre del perfil actualmente

cargado en el reproductor. Cada vez que se activa o desactiva una funcionalidad, el booleano correspondiente a dicha funcionalidad, refleja dicha elección.

En la clase `EasyPlayerProfileManager`, se han desarrollado los siguientes métodos que permiten realizar la comprobación de compatibilidad con los tres perfiles básicos preestablecidos en el editor:

- `isAuditoryImpairmentPlayer` devuelve true si la actual configuración del reproductor contiene los componentes necesarios para poder ser accedido por personas con discapacidades auditivas.
- `isVisualImpairmentPlayer` devuelve true si la actual configuración del reproductor contiene los componentes necesarios para poder ser accedido por personas con discapacidades visuales.
- `isSimplifiedPlayer` devuelve true si la actual configuración del reproductor es una configuración simple (solo con los componentes básicos).

Evaluación

7. Evaluación

7.1 Evaluación de la interfaz gráfica

Para este proyecto, y la elaboración del prototipo, no se ha realizado una evaluación de la interfaz gráfica, debido a la falta de usuarios finales, y a que dicha evaluación no forma parte de los objetivos iniciales.

Esta evaluación se podría haber realizado con unos pocos usuarios, pero no habría sido una muestra suficientemente significativa.

Pese a no haber podido realizar esta evaluación objetiva sí se ha realizado una pequeña evaluación personal y subjetiva llegando a la conclusión personal de que la interfaz realizada ha facilitado en gran medida la personalización de reproductores multimedia a medida desde la interfaz gráfica del editor, puesto que se trata de una interfaz muy intuitiva y sencilla, con atajos de teclado para la ejecución de todas las funcionalidades disponibles.

De igual modo, la experiencia con la interfaz gráfica del reproductor, personalmente resulta sencilla e intuitiva, y la reproducción multimedia resulta satisfactoria.

Esto ha sido posible porque es una interfaz sencilla y funcional que cumple con los requisitos que se le piden en el diseño sin agregar otras funciones que puedan hacerla más enrevesada y complicada al uso.

7.2 Evaluación de la lógica de la aplicación

7.2.1 Prueba 1: Easy Player Editor

Esta prueba consiste en la utilización del Easy Player Editor para crear un reproductor multimedia que tenga las funciones básicas de reproducción. La idea es añadir alguna funcionalidad extra, y después guardar dicho perfil. En todo el proceso, se comprobará que la ayudara contextual muestra las acciones realizadas.

Para ello, ejecutamos el programa, el cual inicialmente nos pide seleccionar un perfil preconfigurado (ver Ilustración 36).



Ilustración 36 – Selección del perfil inicial

Seleccionamos el perfil simplificado, y se abre la interfaz del editor, con los controles de dicho perfil activos en el reproductor (ver Ilustración 37).

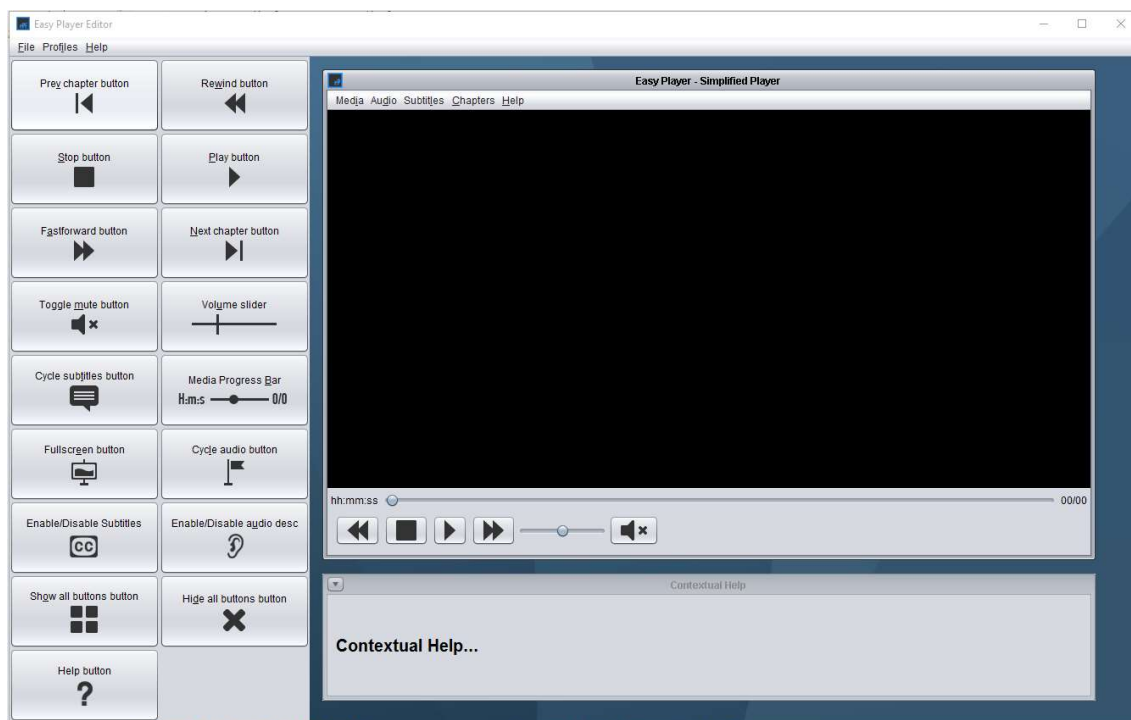


Ilustración 37 – Pantalla principal del Editor

Añadimos el botón de Ayuda, pulsando en la paleta del editor, y vemos cómo el botón se añade al reproductor correctamente, y la ayuda contextual muestra la ejecución de dicha acción (ver Ilustración 38).

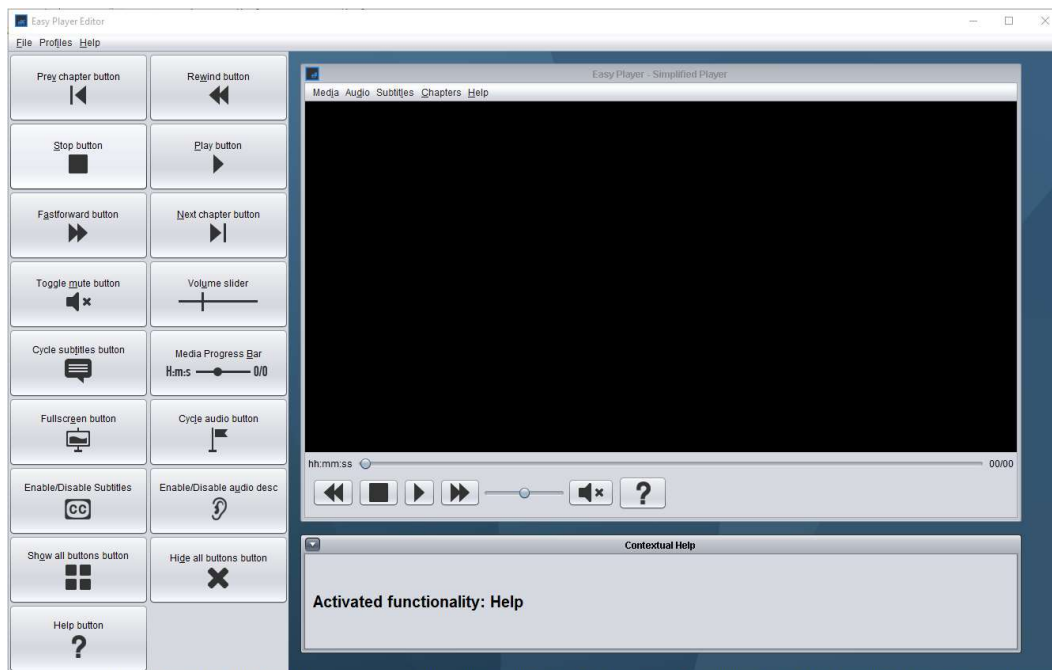


Ilustración 38 – Reproductor modificado

Comprobamos la compatibilidad de la configuración actual para personas con algún tipo de discapacidad, en la ayuda contextual se muestra el resultado (ver Ilustración 39).

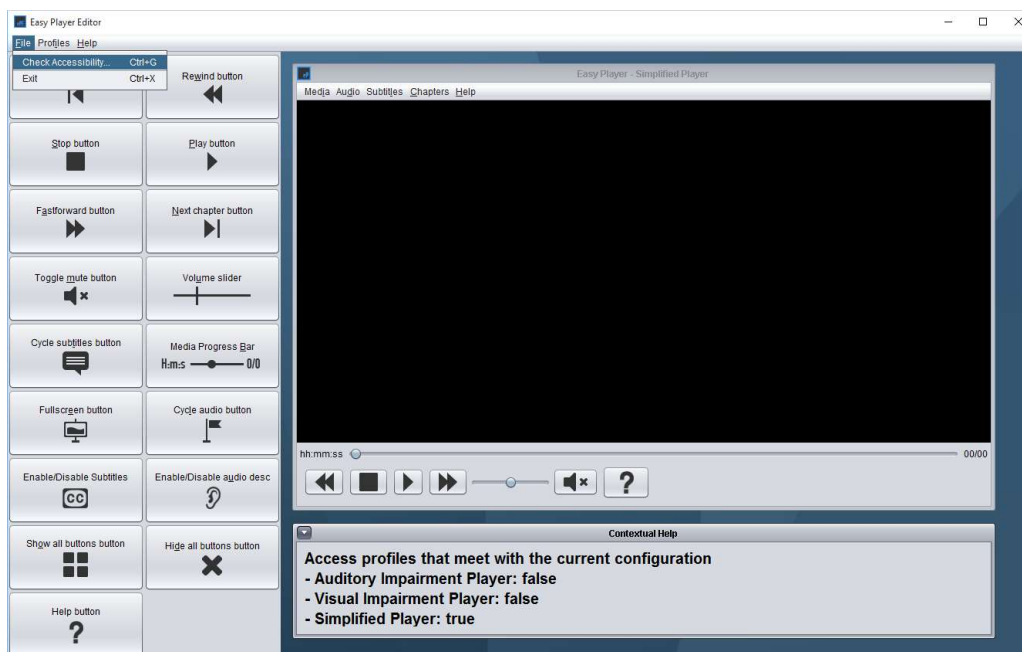


Ilustración 39 – Comprobación de compatibilidad

Una vez añadida la funcionalidad adicional (el botón de Ayuda), guardamos el perfil en el disco duro, para poder cargarlo más adelante. Para ello, pulsamos en el menú de perfiles y seleccionamos la opción “Save Custom Profile...” (ver Ilustración 40).

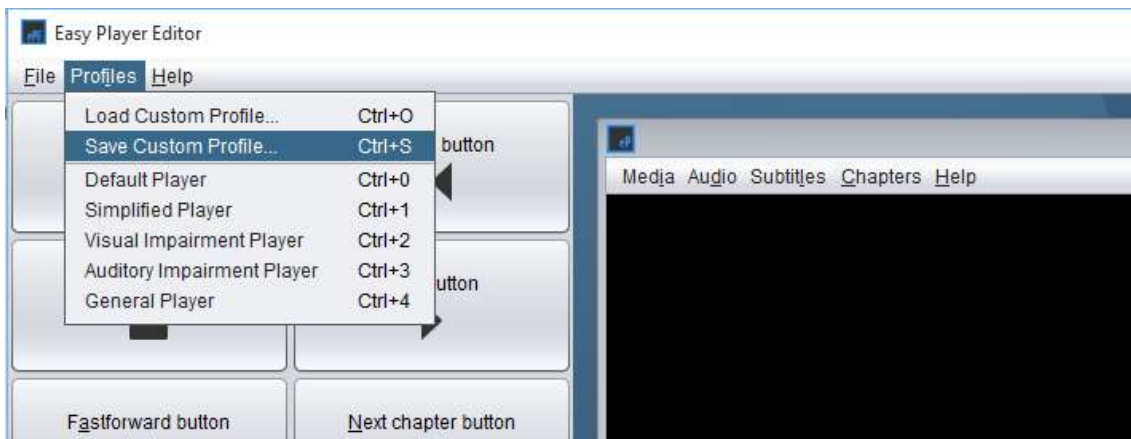


Ilustración 40 – Guardar de perfil

Después de seleccionar dicha opción, se establece un nombre para el perfil (ver Ilustración 41).

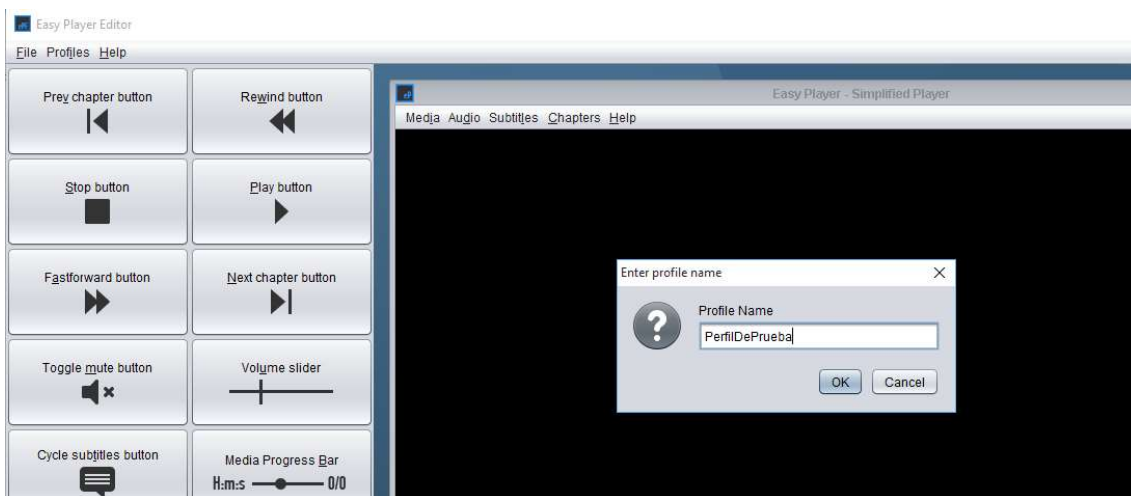


Ilustración 41 – Nombre de perfil

A continuación, se introduce el nombre del fichero de propiedades donde se guardará la configuración del perfil y su localización en disco (ver Ilustración 42).

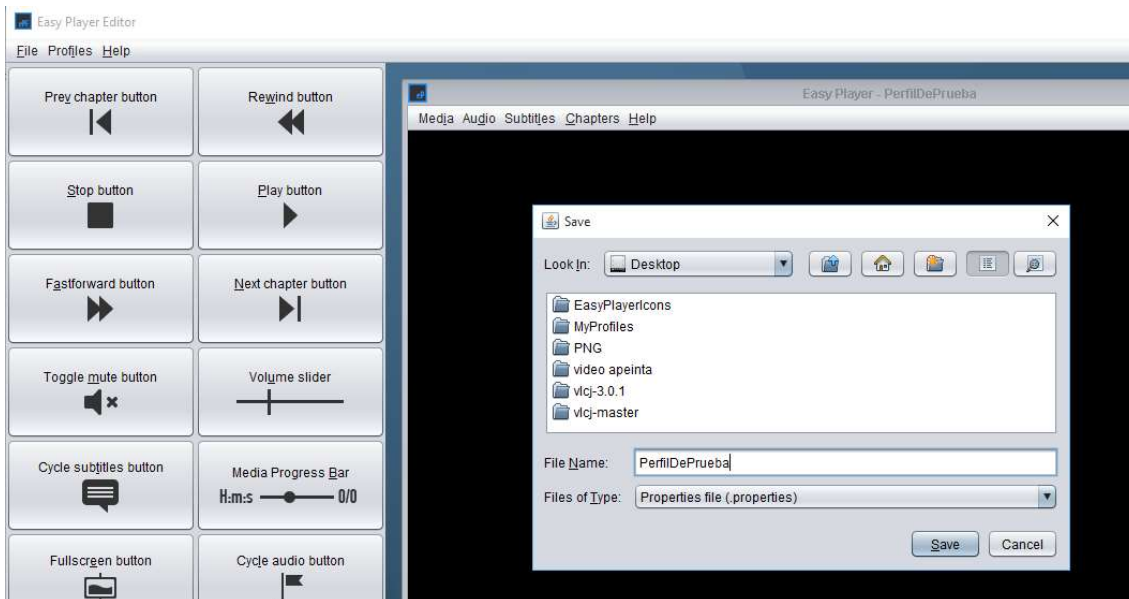


Ilustración 42 – Guardar fichero en disco

Una vez guardado el fichero, la ventana del reproductor se actualiza con el nombre de perfil introducido, y la ayuda contextual muestra el mensaje correspondiente (ver Ilustración 43).

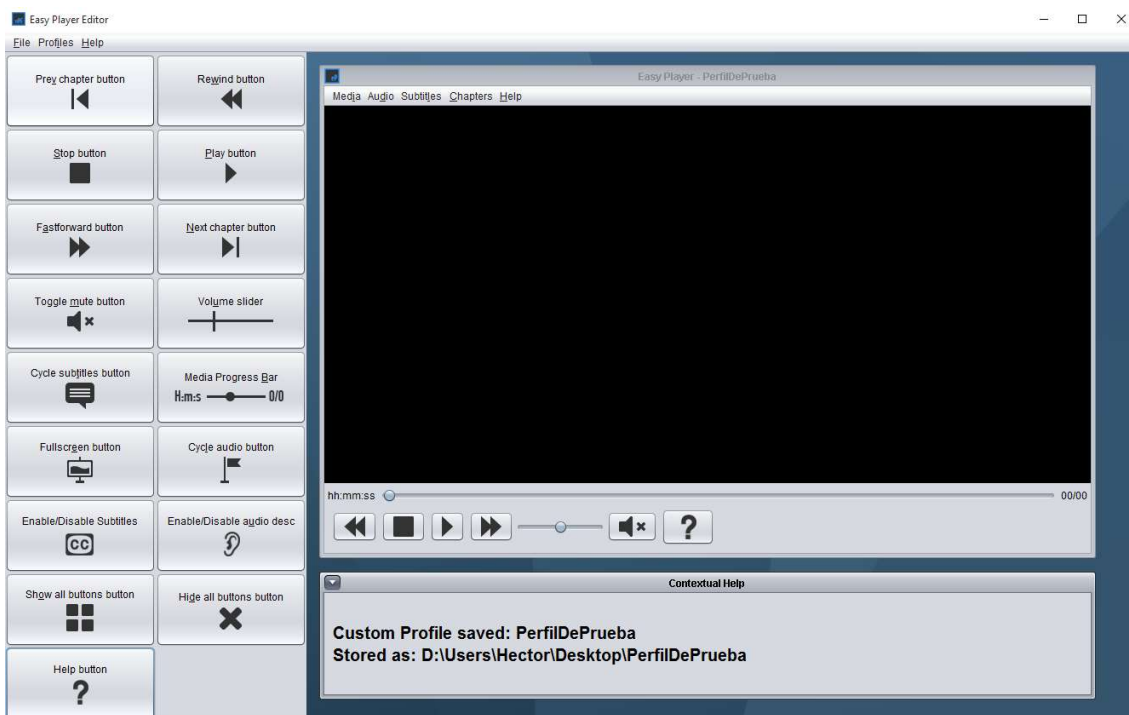


Ilustración 43 – Perfil guardado

Para probar el correcto funcionamiento, se selecciona de nuevo el perfil simplificado del menú de perfiles (ver Ilustración 44).

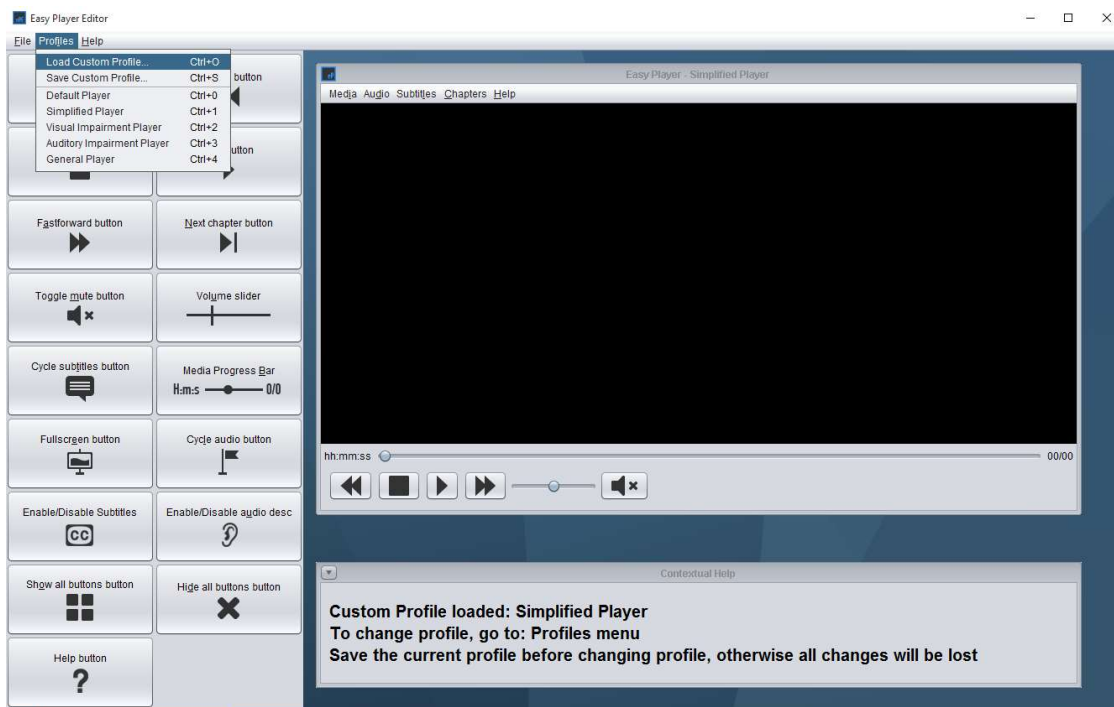


Ilustración 44 – Cargar perfil

Se carga el perfil personalizado guardado anteriormente (ver Ilustración 45).

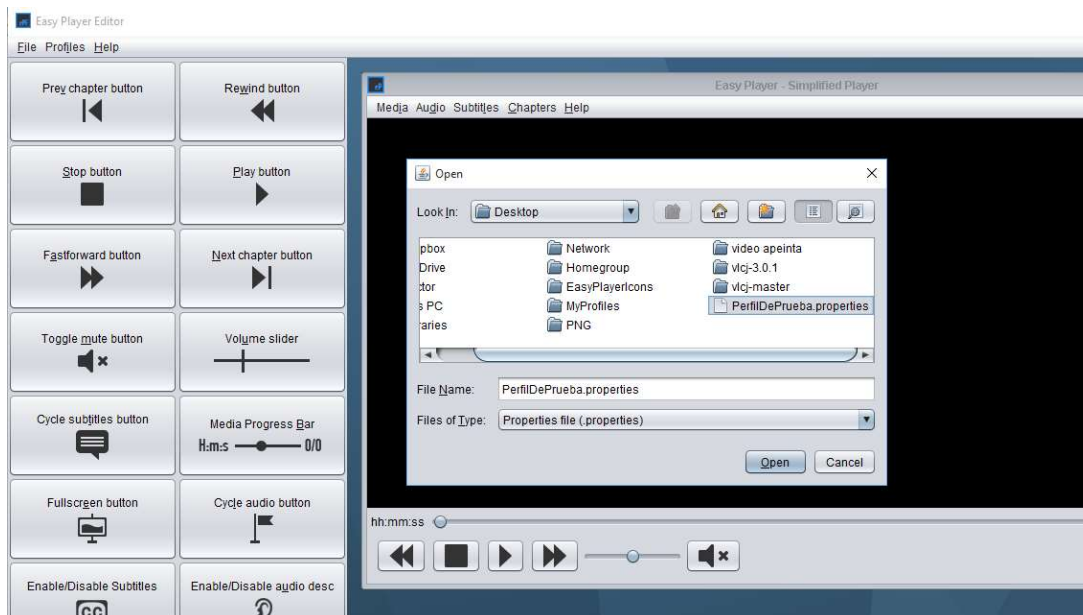


Ilustración 45 – Cargar perfil

Al cargar el perfil personalizado se puede comprobar que se guardó de forma correcta (ver Ilustración 46).

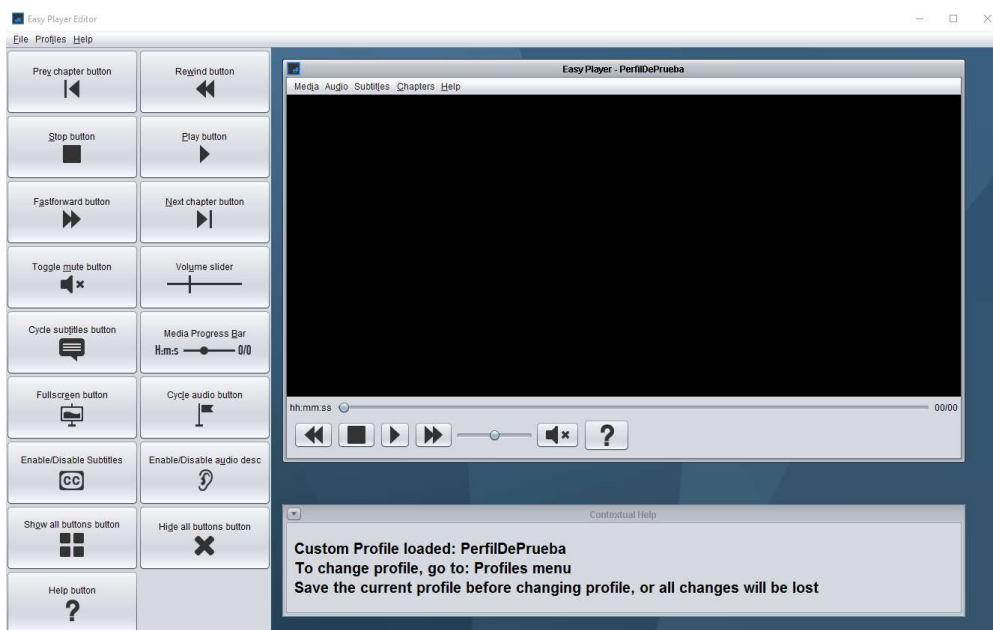


Ilustración 46 – Cargar perfil

Todas las acciones realizadas anteriormente, también podrían haber sido ejecutadas mediante combinaciones de teclado.

Para comprobar que los atajos de teclado funcionan de forma correcta, a continuación se muestra el resultado de ejecutar el atajo de teclado que muestra la ayuda del editor (Alt+H), y acto seguido, se ejecuta el atajo de teclado que muestra todos los atajos de teclado proporcionados por el editor (Ctrl+S) (ver Ilustración 47 y 48).

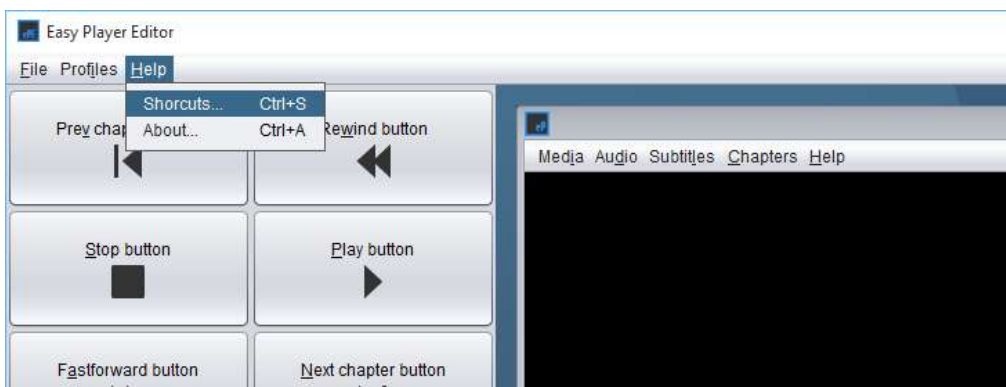


Ilustración 47 – Atajo a la Ayuda

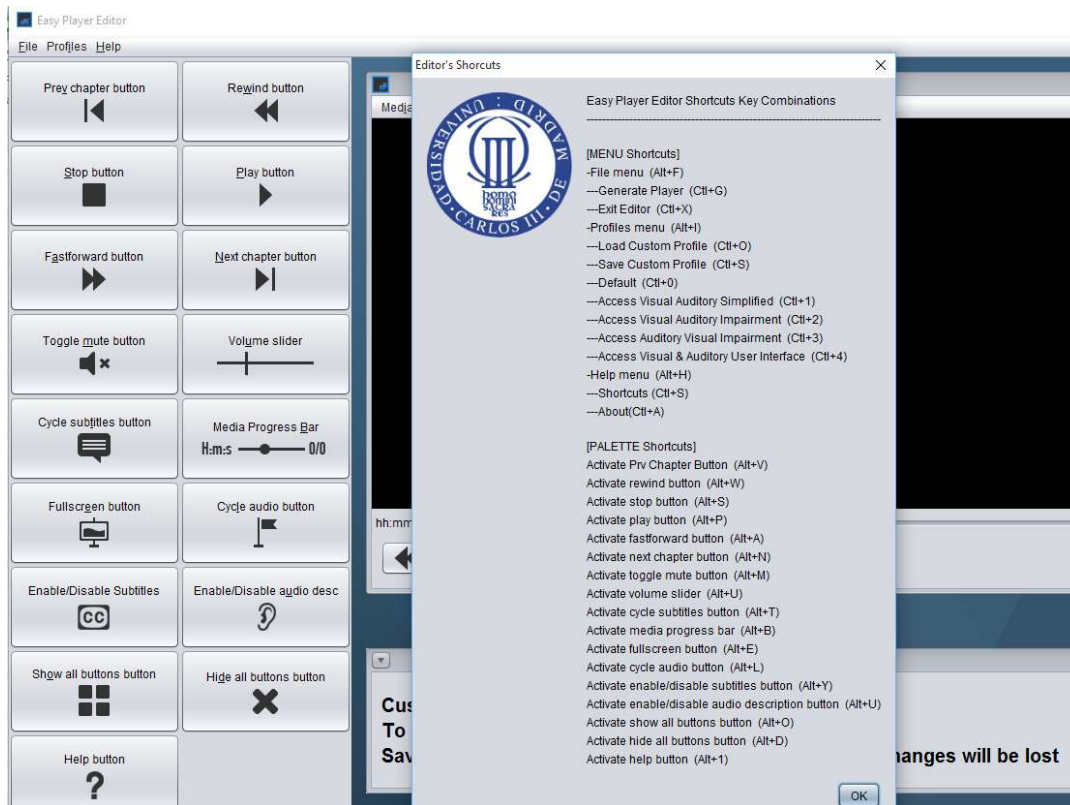


Ilustración 48 – Atajo a la información de los atajos

Como puede comprobarse, el funcionamiento del editor supera las pruebas ejecutadas satisfactoriamente, cumpliendo con los requisitos establecidos en el planteamiento inicial.

7.2.2 Prueba 2: Reproducción en Easy Player

Para esta prueba, se ha seleccionado un vídeo con las siguientes especificaciones multimedia (ver Ilustración 49).

Container and general information Matroska: 4.97 GiB, 2h 4mn 1 video stream: AVC 2 audio streams: DTS /AC-3 2 text streams: ASS /ASS		Overall bit rate: 5 704 Kbps Encoded date: UTC 2015-01-24 19:08:07 Writing application: mkvmerge v7.5.0 ('Glass Culture') 32bit built on J Writing library: libebml v1.3.1 + libmatroska v1.4.2 Attachements: RaspoutineMedium_TB.otf	
Go to the web site of a player for this file			
First video stream English, 4 500 Kbps, 1920*820 (2.35:1), at 23.976 fps, AVC (High@L4.1) (CABAC / 4 Ref Frames)			
Go to the web site of this video codec			
First audio stream Spanish, 755 Kbps, 48.0 KHz, 24 bits, 6 channels, DTS		Second audio stream English, 448 Kbps, 48.0 KHz, 6 channels, AC-3	
First text stream Spanish, ASS		Second text stream English, ASS	
Go to the web site of this text codec		Go to the web site of this text codec	

Ilustración 49 – Información del contenido multimedia

Se trata de la película Oblivion, con una pista de vídeo, dos pistas de audio (español e inglés), dos pistas de subtítulos (español e inglés), así como el contenido de capítulos.

Inicialmente, abrimos el editor y cargamos el perfil predefinido para personas con discapacidades auditivas (ver Ilustración 50).

The screenshot shows the Easy Player Editor window. On the left is a control panel with buttons for: Prey chapter button, Rewind button, Stop button, Play button, Fastforward button, Next chapter button, Toggle mute button, Volume slider, Cycle subtitles button, Media Progress Bar (H:m:s 0/0), Fullscreen button, Cycle audio button, Enable/Disable Subtitles, Enable/Disable audio desc, Show all buttons button, Hide all buttons button, and Help button. The main area displays the 'Easy Player - Auditory Impairment Player' window with a video player and a contextual help message: 'Custom Profile loaded: Auditory Impairment Player. To change profile, go to: Profiles menu. Save the current profile before changing profile, otherwise all changes will be lost'.

Ilustración 50 – Reproductor configurado

Para la realización de la prueba se abre el fichero multimedia, el cual inicia la reproducción con la pista de audio en español (puesto que el fichero así está configurado en sus especificaciones MKV), y sin mostrar subtítulos (ver Ilustración 51).

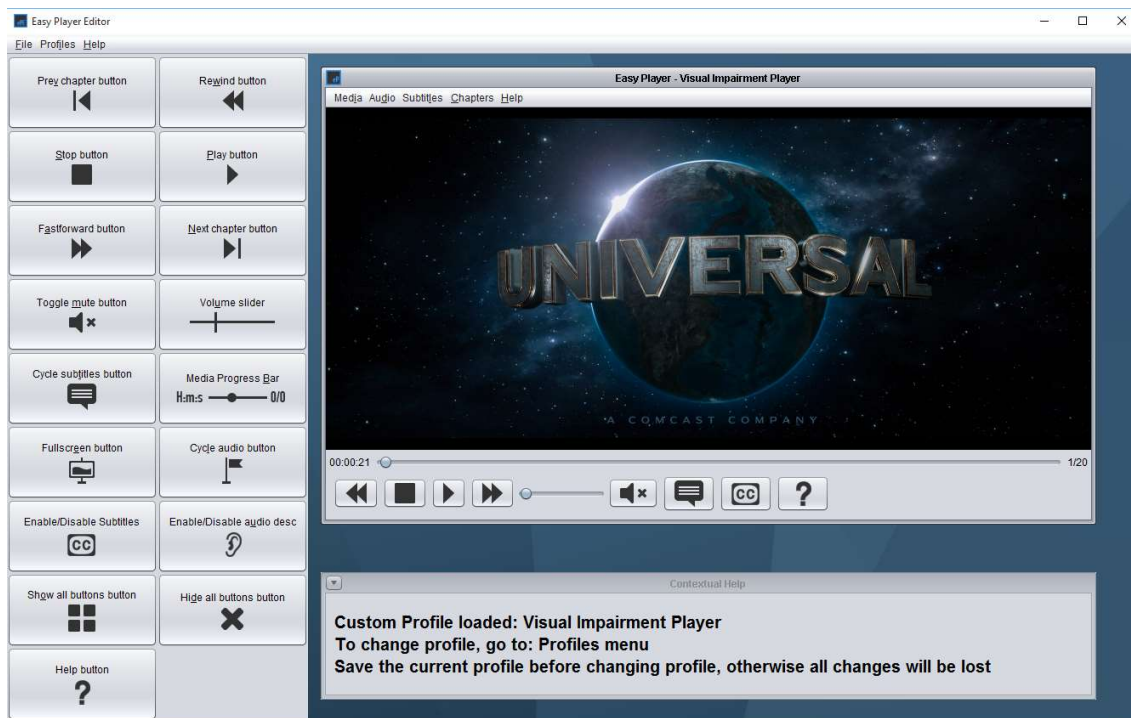


Ilustración 51 – Archivo multimedia abierto

Una vez abierto el fichero multimedia, cambiamos a la pista de audio en inglés, y el audio cambia correctamente (ver Ilustración 52).

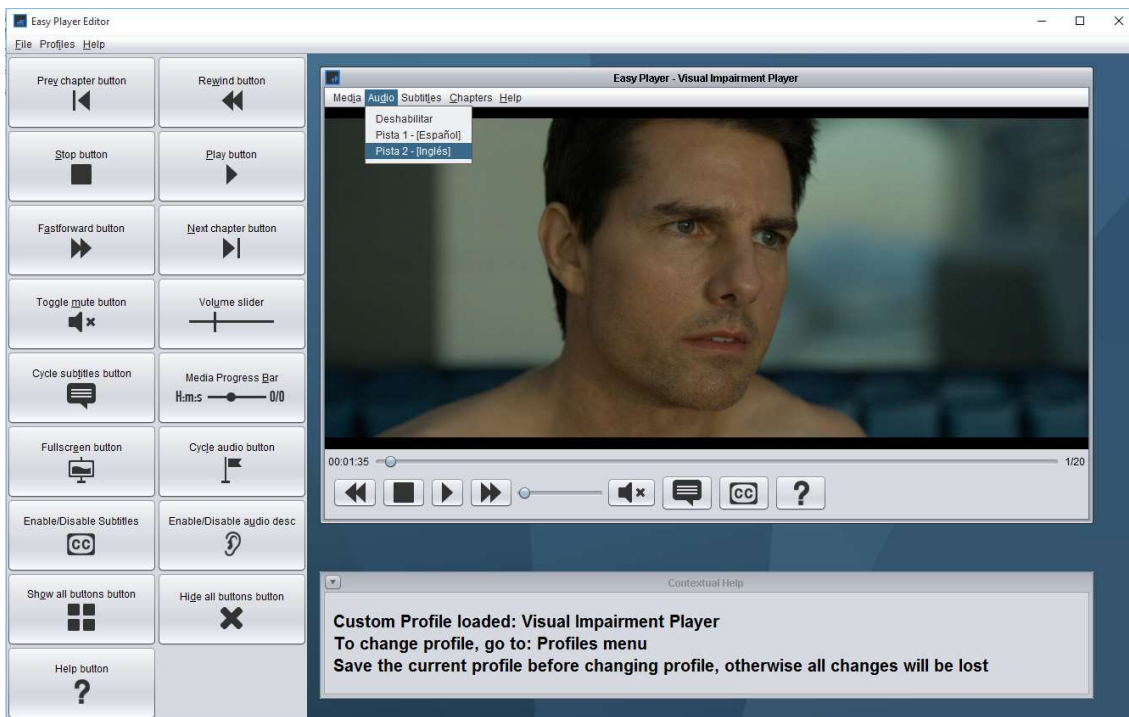


Ilustración 52 – Selección de pista de audio en inglés

Se selecciona la pista de subtítulos en inglés (ver Ilustración 53).

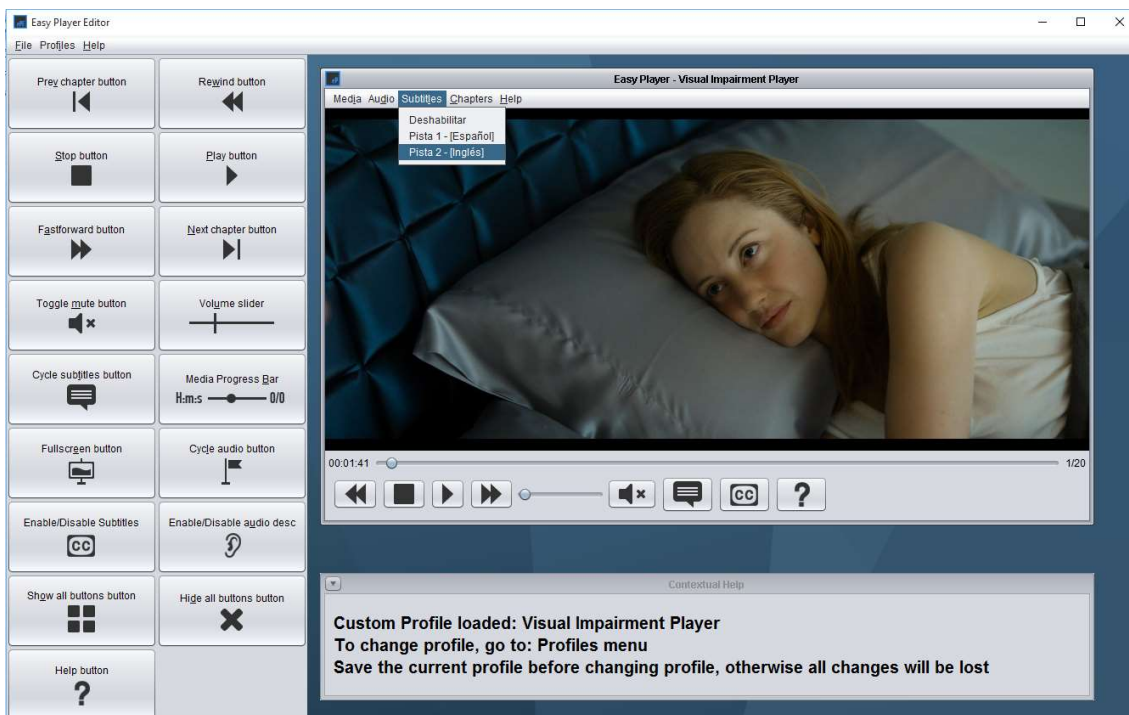


Ilustración 53 – Selección de pista de subtítulos en inglés

Para que se muestren los subtítulos, se habilita el botón “CC” (ver Ilustración 54).

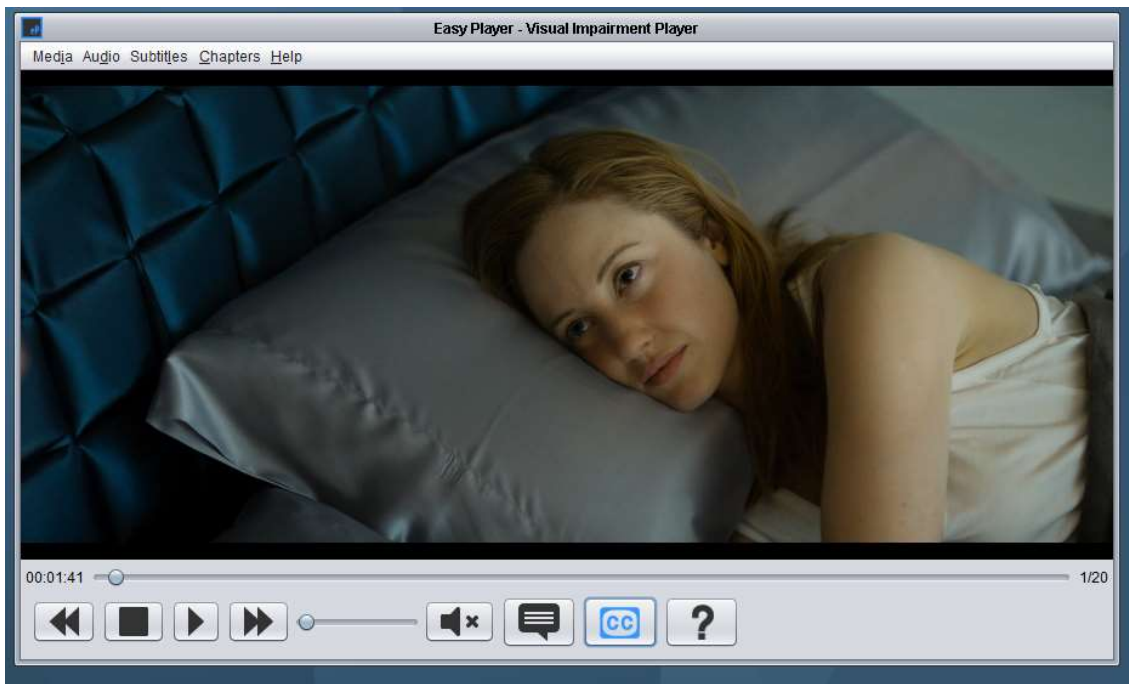


Ilustración 54 – Subtítulos habilitados

Al cabo de un par de segundos, los subtítulos se comienzan a mostrar (ver Ilustración 55).



Ilustración 55 – Subtítulos en pantalla

Se comprueba que las funciones básicas se comportan tal y como se espera: play (si se pulsa este control, el vídeo se pausa, si se vuelve a pulsar, el vídeo se reanuda), el stop detiene el vídeo, la barra de menú modifica el audio correctamente, la barra de progreso del vídeo permite movernos por el vídeo hasta el punto deseado (y automáticamente se actualiza el instante del vídeo así como el capítulo en el que nos encontramos, etc.).

Por ejemplo, si nos desplazamos hasta al minuto 43:31 del vídeo, se puede observar (ver Ilustración 56) como se muestra correctamente el instante del vídeo al que nos hemos movido, así como el capítulo en el que nos encontramos, en este caso, el capítulo 6 (el primer capítulo en este vídeo es el capítulo 0, por ello, en el reproductor aparece el número 7 de 20).



Ilustración 56 – Desplazamiento en el vídeo

Si se selecciona el capítulo 10 (como se ha comentado anteriormente, en la Ilustración 57 se muestra el texto 11 de 20, puesto que los capítulos en el reproductor se cuentan desde el 1, y en este caso, el primero capítulo del vídeo es el número 0) para

comprobar la selección de capítulos, se observa como el vídeo se desplaza al comienzo de dicho capítulo (ver Ilustración 57).



Ilustración 57 – Desplazamiento en el vídeo

Como ocurría en la prueba anterior, todas las acciones realizadas anteriormente, también se podrían haber ejecutado mediante combinaciones de teclado.

Como en la sección anterior (Prueba 1: Easy Player Editor), se han realizado las mismas pruebas para comprobar el funcionamiento de los atajos de teclado, por lo que, se accede primero al menú Ayuda del reproductor a través de la combinación Alt+H, y posteriormente se utiliza la combinación Ctrl+A para acceder al submenú About que contiene información sobre el editor (ver Ilustración 58).

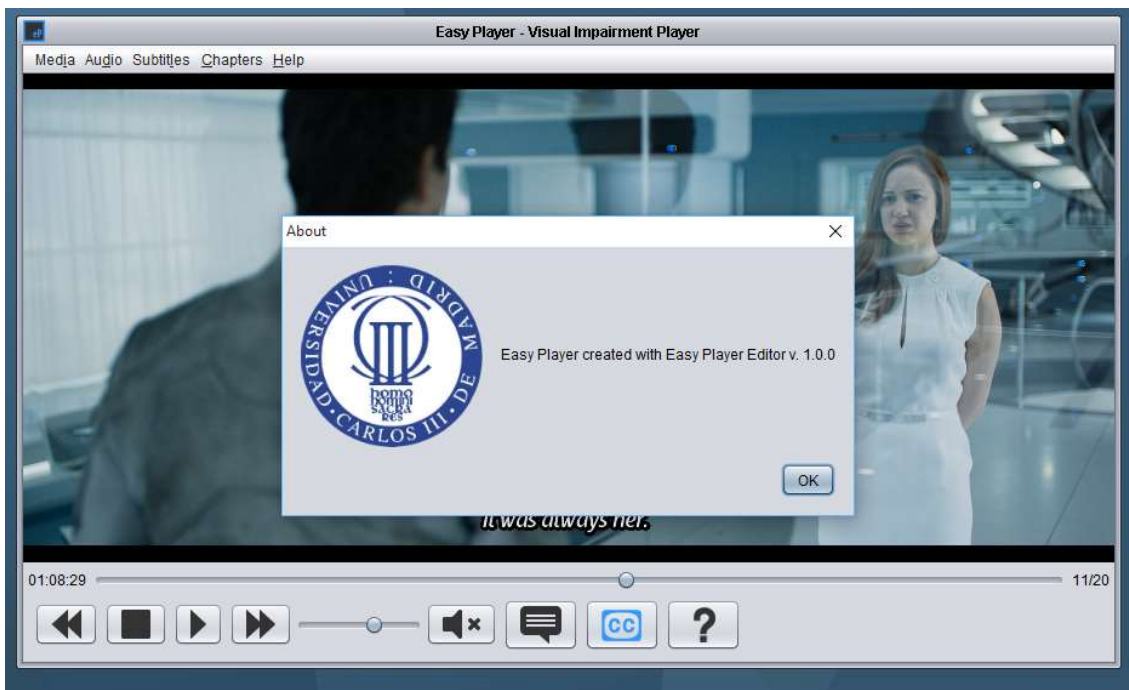


Ilustración 58 – Ayuda del Reproductor

Como puede comprobarse, el funcionamiento del editor supera las pruebas ejecutadas satisfactoriamente, cumpliendo con los requisitos establecidos en el planteamiento inicial.

7.2.3 Prueba 3: Reproducción en Easy Player

Para esta prueba, se ha seleccionado un vídeo con las siguientes especificaciones multimedia (ver Ilustración 59).

Container and general information	
MPEG-4 (Base Media): 74.0 MiB, 10mn 40s	Overall bit rate mode: Variable
1 video stream: MPEG-4 Visual	Overall bit rate: 969 Kbps
3 audio streams: AAC / AAC / AAC	Encoded date: UTC 2015-01-19 12:38:37
1 text stream: Timed Text	Tagged date: UTC 2015-01-19 12:38:37
	Writing application: My MP4Box GUI 0.6.0.6 < http://my-mp4box-gui.zyt
Go to the web site of a player for this file	
First video stream	
810 Kbps, 650*365 (16:9), at 25.000 fps, MPEG-4 Visual (Simple@L1) (BVOP)	
First audio stream	Second audio stream
English, 62.7 Kbps, 44.1 KHz, 2 channels, AAC (LC)	Spanish, 61.6 Kbps, 44.1 KHz, 2 channels, AAC (LC)
First text stream	
English, Timed Text	
Go to the web site of this text codec	

Ilustración 59 – Información del contenido multimedia

Se trata de un vídeo multimedia con una pista de vídeo, una de subtítulos y tres de audio:

- Pista 1: Inglés con Audiodescripción
- Pista 2: Español con Audiodescripción
- Pista 3: Español

Inicialmente, se abre el editor y se carga el perfil predefinido para personas con discapacidad visual, así como el botón de subtulado (activamos este último) (ver Ilustración 60).

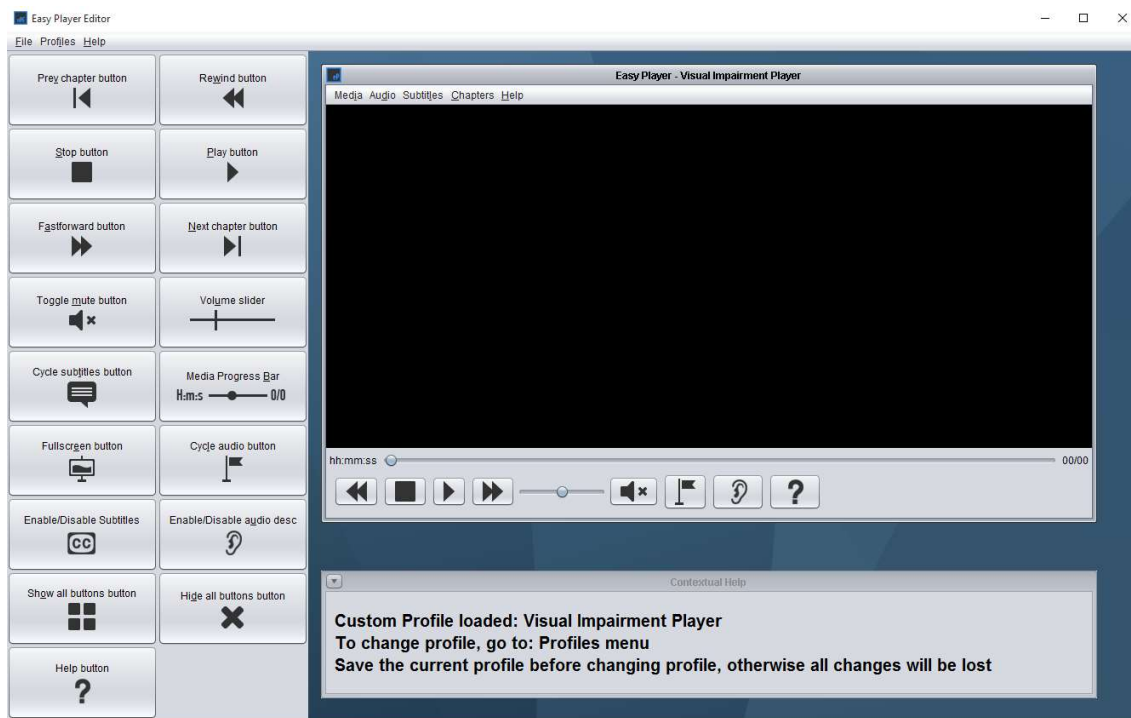


Ilustración 60 – Reproductor configurado

En este punto, es importante destacar que no existe un estándar sobre qué pista de audio debería ser considerada la audiodescripción en un medio multimedia (DVD, Bluray, etc.). Adicionalmente, en este prototipo se ha optado por seleccionar la pista número 1 como la audiodescripción, lo que quiere decir, que cada vez que se active el botón de la audiodescripción, únicamente se reproducirá la pista número 1, aunque se cambie de pista. Una vez se desactive el botón de audiodescripción, automáticamente se reproducirá la pista que estuviera seleccionada. Para más información sobre este mecanismo, consultar el apartado Panel de Controles de Reproducción. Para la realización de la prueba se abre el fichero multimedia, el cual inicia la reproducción con la pista de audio número 1, que en este caso es justamente la audiodescripción (ver Ilustración 61).

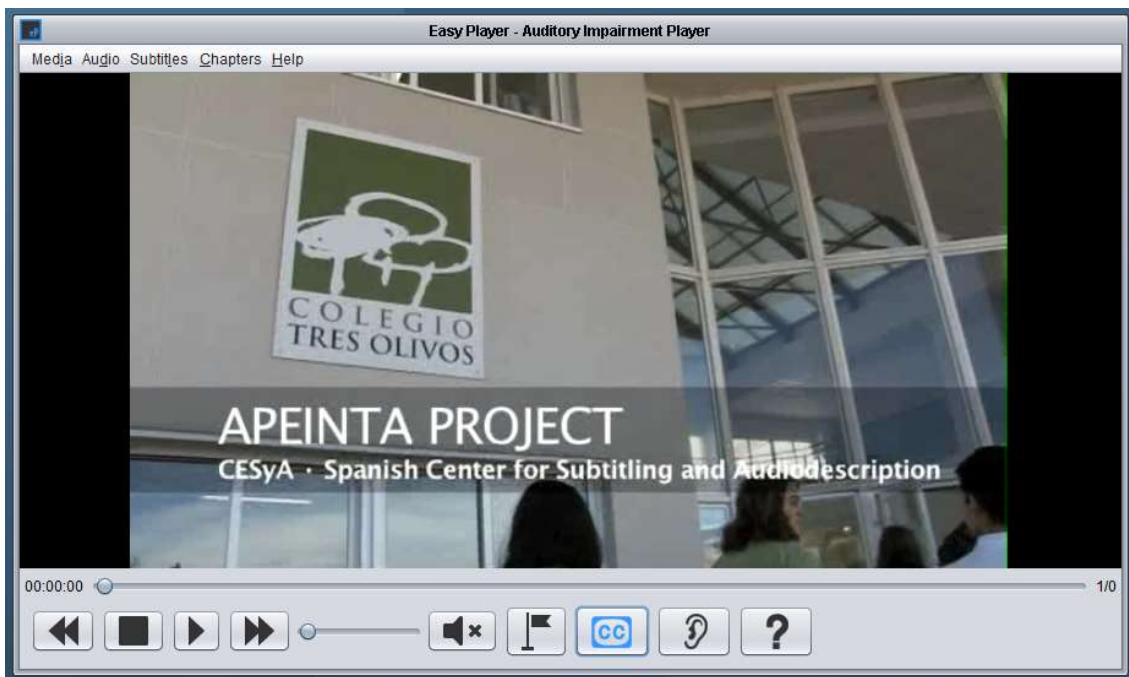


Ilustración 61 – Archivo multimedia abierto

A continuación se selecciona la pista de audio número 2, y la pista de subtítulos (ver Ilustración 62).



Ilustración 62 – Selección de pistas

El audio comienza a reproducirse en español y se muestra la única pista de subtítulos disponible en el vídeo correctamente.

A continuación, se activa la audiodescripción, y se comprueba como automáticamente comienza a reproducirse la pista número 1 que es la pista de audio en inglés con audiodescripción (ver Ilustración 63).

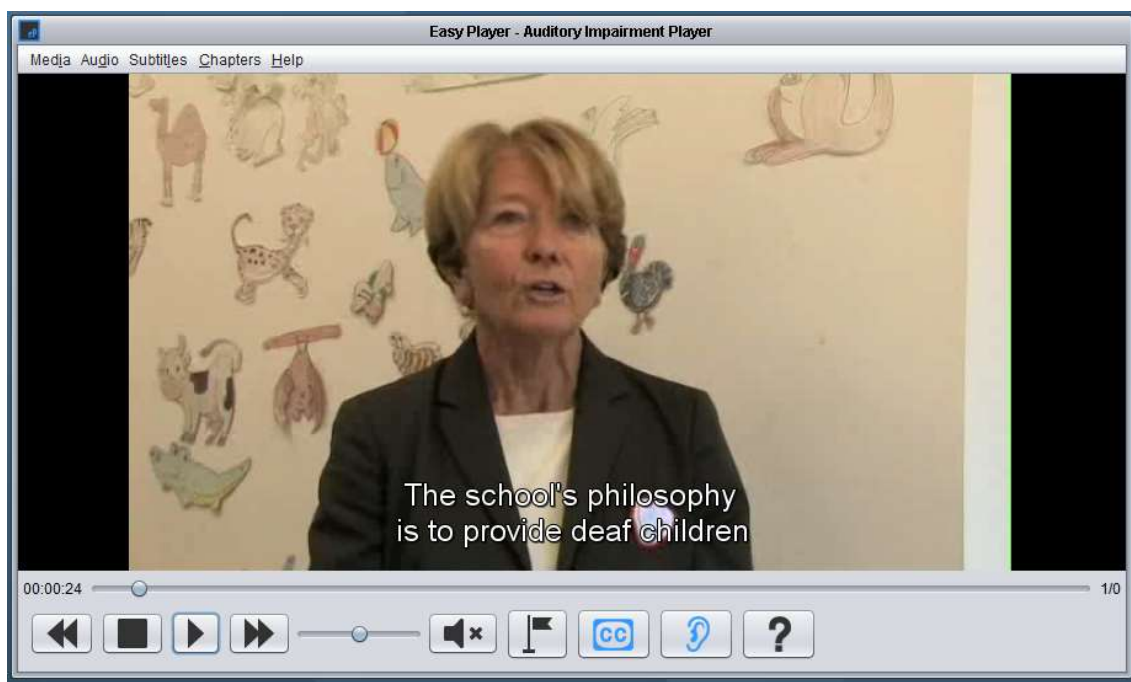


Ilustración 63 – Audiodescripción activada

Si se desactiva la audiodescripción, automáticamente, vuelve a reproducirse la pista de audio número 2, que es la que estaba seleccionada antes de activar la audiodescripción.

Por tanto, el funcionamiento del reproductor supera las pruebas ejecutadas, cumpliendo con los requisitos establecidos en el planteamiento inicial.

Conclusiones

8. Conclusiones

De acuerdo con los resultados obtenidos en las pruebas ejecutadas en el apartado de Evaluación de este documento, se puede concluir que el prototipo desarrollado en este proyecto funciona correctamente, puesto que es capaz de diseñar reproductores multimedia adecuados a las necesidades del usuario final, gestionar perfiles personalizados, así como reproducir archivos multimedia con diferentes formatos y contenido.

Cabe comentar que aunque se ha conseguido introducir gran cantidad de funcionalidad de reproducción en el prototipo, para los archivos multimedia de tipo MKV se ha conseguido reproducir pistas de subtítulos en formato avanzado como ASS, pero no en formato SRT. Esto probablemente se debe a un bug en el framework de VLCJ al reproducir archivos MKV que incluyen subtítulos SRT con codificación UTF-8. Tampoco fue posible incorporar la funcionalidad del cambio de tipo de fuente, tamaño y color, presumiblemente por un bug en el framework. Para tratar de depurar cada uno de estos problemas, se abrió una discusión con el creador del framework VLCJ [\[34\]](#) [\[35\]](#).

Si bien el comportamiento general del prototipo es satisfactorio, es importante remarcar que la inclusión de la funcionalidad indicada en la sección Futuras líneas de trabajo significaría una solución más completa y accesible. Ya que por el momento, ni el editor gráfico ni el reproductor diseñado pueden ser considerados accesibles, debido a que no cumplen con todos los requisitos de accesibilidad.

A pesar de ello, y como conclusión final se puede decir que el prototipo funciona correctamente según el planteamiento inicial, y cumple perfectamente su cometido.

Futuras líneas de trabajo

9. Futuras líneas de trabajo

Dentro de las futuras líneas de trabajo destinadas a mejorar el prototipo implementado en este Proyecto Fin de Carrera se pueden distinguir las siguientes líneas de trabajo:

- Añadir funcionalidad al reproductor para poder cargar ficheros de subtítulos de forma externa.
- Añadir funcionalidad al reproductor para modificar la fuente, el tamaño y el color de los subtítulos [\[35\]](#).
- Añadir funcionalidad al reproductor para realizar una búsqueda en los subtítulos y que el vídeo avance hasta la posición donde se encuentran dicho subtítulo.
- Añadir funcionalidad al reproductor para reproducir el vídeo a pantalla completa.
- Añadir funcionalidad al editor para generar código HTML5 basado en la configuración del perfil seleccionado.
- Añadir funcionalidad a la ayuda contextual, para almacenar un histórico de acciones realizadas por el usuario, y que este pueda desplazarse por ellas.
- Agregar las librerías del framework VLCJ dentro del paquete Java, para que se carguen del mismo, y no de un directorio externo.

Referencias

10. Referencias

[1] Proyecto VLCJ

Referencia al proyecto Open Source vlcj de Caprica Software.

<http://capricasoftware.co.uk/index.html#/projects/vlcj>

[2] Java Native Interface (JNI)

Referencia al framework de programación KNI.

http://es.wikipedia.org/wiki/Java_Native_Interface

[3] Reproductor multimedia VLC

Referencia al popular reproductor VLC.

<http://www.videolan.org/vlc/index.html>

[4] Definición de Interfaz de Programación de Aplicación

Referencia a la definición de API.

http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

[5] Definición de Patrón de diseño "Observer"

Referencia a la definición del patrón de diseño "Observer".

http://es.wikipedia.org/wiki/Observer_%28patr%C3%B3n_de_dise%C3%B1o%29

[6] W3C Web Accesibility Initiative

Referencia a W3C Web Accesibility Initiative

<http://www.w3.org/WAI/intro/people-use-web/Overview.html>

[7] Modelo en Espiral

Referencia a la documentación a cerca del modelo en espiral diseñado por Barry Boehm.

https://en.wikipedia.org/wiki/Spiral_model

[8] MKV

Referencia a la documentación a cerca del contenedor multimedia Matroska.

<https://en.wikipedia.org/wiki/Matroska>

[9] Tecnología Java

Varias referencias en las que se ha basado este proyecto para documentación a cerca de la historia del lenguaje de programación Java:

<https://www.fayerwayer.com/2012/01/los-lenguajes-de-programacion-mas-populares-de-2011/>

[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

<https://blog.udemy.com/los-mejores-lenguajes-de-programacion-para-aprender-en-2013-los-ocho-magnificos/>

[10] Swing

Enlaces de referencia sobre el framework Swing:

<http://docs.oracle.com/javase/tutorial/uiswing/start/>

[https://en.wikipedia.org/wiki/Swing_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))

[11] AWT

Enlaces de referencia sobre el framework AWT:

<http://www.javaworld.com/article/2077188/core-java/introduction-to-the-awt.html>

https://en.wikipedia.org/wiki/Abstract_Window_Toolkit

[12] Java Media Framework

Enlaces de referencia sobre el framework Java Media Framework:

https://en.wikipedia.org/wiki/Java_Media_Framework

[13] JavaFX

Enlaces de referencia sobre el framework *JavaFX*:

<https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>

[14] WCAG 2.0

Enlaces de referencia sobre las Web Content Accessibility Guidelines:<http://www.w3.org/TR/WCAG20/>

[15] Technical standards Section 508

<http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/guide-to-the-section-508-standards>

[16] Barrierefreie-Informationstechnik-Verordnung (BITV) 2.0

http://www.gesetze-im-internet.de/bitv_2_0/BJNR184300011.html

[17] Référentiel Général d'Accessibilité pour les Administrations (RGAA)

<https://references.modernisation.gouv.fr/referentiel>

[18] Accessibility for Ontarians with Disabilities Act (AODA)

<http://www.aoda.ca/>

[19] Norma UNE 139803:2012

<http://accesibilidadweb.dlsi.ua.es/?menu=une139803-2012>

[20] UAAG 2.0

<http://www.w3.org/TR/2014/WD-UAAG20-20140925/>

[21] UAAG 1.0

<http://www.w3.org/TR/WAI-USERAGENT/>

[22] ATAG 2.0

<http://www.w3.org/TR/ATAG20/>

[23] ISO 9241-171

http://www.iso.org/iso/catalogue_detail.htm?csnumber=39080

[24] An approach to User Interface Design of an accessible user agent

Referencia al estudio de accesibilidad previo a este proyecto:

González-García, M., Moreno, L., and Martínez, P. 2012. An approach to User Interface Design of an accessible user agent. 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion, DSAI 2012, Douro Region, Portugal, July, Elsevier, Volume: 14, pp. 254-262.

[25] JW Player

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

[26] BSPlayer

<http://www.bsplayer.org/>

[27] VideoLAN Media Player

<https://www.videolan.org/>

[28] Acorn Media Player

<https://github.com/ghinda/acornmediaplayer>

[29] LeanBack Player

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

[30] VideoJS Player

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

[31] Easy Youtube Player

<http://hiantonia.com/journal/2009/06/17/easy-youtube-player-making-it-easier/>

[32] KMPlayer

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

[33] OZ Player

<http://www.accessibilityoz.com/ozplayer/>

[34] Framework VLCJ: problema subtítulos

Discusión con el creador del framework sobre un problema con el formato de los subtítulos:

<http://stackoverflow.com/questions/27805970/mediaplayer-setspu-not-working>

[35] Framework VLCJ: cambiar tipo de fuente, color y tamaño

Discusión con el creador del framework sobre un problema con el formato de los subtítulos:

<http://stackoverflow.com/questions/27931586/set-font-name-size-and-color-for-selected-subtitle-with-vlcj>

[36] Framework VLCJ: cambiar tipo de fuente, color y tamaño

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625