



Universidad de Jaén

Escuela Politécnica Superior
de Jaén

TRABAJO FIN DE GRADO

CASSIOPEIA, SISTEMA DE GESTIÓN DISCOGRÁFICA

Alumno

Orestes Colomina Monsalve

Tutor

Carlos Javier Ogáyar Anguita

(Departamento de Informática)

septiembre, 2022

(Página intencionalmente en blanco)



Universidad de Jaén

Departamento de Informática

Don Carlos Javier Ogáyar Anguita, tutor del Trabajo Fin de Grado titulado: '**Cassiopeia, Sistema de Gestión discográfica**', que presenta Don Orestes Colomina Monsalve, otorga el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, septiembre de 2022

El alumno:

01/09/2022

El tutor:

X Orestes Colomina Monsalve

Orestes Colomina Monsalve

Signed by: COLOMINA MONSALVE ORESTES - 26526901N

Orestes Colomina Monsalve

Carlos Javier Ogáyar Anguita

(Página intencionalmente en blanco)

Agradecimientos

A mi familia, por permitirme trastear con un ordenador desde pequeño y desde entonces querer estudiar esta titulación. A mis amigos y amigas, Carlos, José, Jaime, Dulce, Ana y Andrea por estar siempre ahí y aguantarme durante todo este eterno proceso.

Y por último a mi tutor Carlos Javier por mostrar gran interés en el proyecto y darme la oportunidad de que se haga realidad este proyecto.

Gracias a todos, de verdad.

FICHA DEL TRABAJO FIN DE TÍTULO

Titulación	Grado en Ingeniería Informática
Modalidad	Proyecto de Ingeniería
Especialidad <small>(solo TFG)</small>	Tecnologías de la Información
Mención <small>(solo TFG)</small>	Tecnologías para la Información y la Comunicación
Idioma	Español
Tipo	General
TFT en equipo	No
Autor/a	Orestes Colomina Monsalve
Fecha de asignación	10/11/2021
Descripción corta	<p>Este trabajo consiste en la realización de un software de gestión discográfica integral para la indexación y tratamiento de colecciones de música en formatos tanto físicos como digitales.</p> <p>Aunque ya existen soluciones para algunas de las necesidades de gestión de los audiófilos, este software busca ofrecer una solución integrada que incluya funcionalidades como la indexación de música y discos de fuentes físicas (CDs, vinilos, casetes, etc.) y digitales (ficheros, streaming, etc.), la gestión y búsqueda de metadatos y letras de canciones, una reproducción integrada de archivos en los formatos más extendidos (incluyendo listas de reproducción), así como la sincronización con algún sistema de streaming como Spotify o Tidal.</p>

NORMAS APLICADAS EN ESTE DOCUMENTO

LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA

NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
<i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i>	

Contenidos

1	Especificación del trabajo	15
1.1	Introducción	15
1.2	Objetivos del trabajo	15
1.3	Antecedentes y estado del arte	16
1.3.1	Situación actual	17
1.3.2	Resumen de las deficiencias y carencias identificadas	26
1.4	Requisitos iniciales	29
1.5	Alcance	31
1.6	Hipótesis y restricciones	31
1.7	Estudio de alternativas y viabilidad	31
1.8	Descripción de la solución propuesta	34
1.9	Tecnologías utilizadas	35
1.9.1	Librerías	35
1.9.2	Software	36
1.10	Metodología de desarrollo de software	38
1.11	Estimación del tamaño y esfuerzo	38
1.12	Planificación temporal	40
1.13	Presupuesto	41
2	Diseño de la aplicación	42
2.1	Especificaciones del sistema	42
2.2	Análisis y diseño del sistema	43
2.2.1	Diseño del sistema	43
2.2.2	Diseño de las interfaces	44
2.3	Diagramas de clases	45
2.3.1	Diagrama de casos de uso	45
2.3.2	Diagramas de secuencia	46
2.4	Implementación del sistema	49
2.4.1	Núcleo	49

2.4.2	Clase Program	50
2.4.3	Configuración	51
3	Desarrollo	52
3.1	Primera Iteración	52
3.1.1	Conjunto de pruebas	57
3.2	Segunda iteración	57
3.2.1	Conjunto de pruebas	61
3.3	Tercera iteración	61
3.3.1	Conjunto de pruebas	64
3.4	Cuarta iteración	64
3.4.1	Conjunto de pruebas	70
3.5	Quinta iteración	70
3.5.1	Conjunto de pruebas	74
3.6	Sexta iteración	74
3.6.1	Conjunto de pruebas	81
3.7	Séptima iteración	81
3.7.1	Conjuntos de pruebas	84
3.8	Pruebas finales	84
3.8.1	Pruebas de verificación del sistema	85
3.8.2	Pruebas de validación del sistema	86
4	Modelo de negocio	88
4.1	Objetivos	88
4.1.1	Objetivos cualitativos	88
4.1.2	Objetivos cuantitativos	88
4.2	Análisis del entorno	89
4.3	Forma jurídica de la empresa	90
5	Conclusiones y trabajos futuros	93
6	Apéndices	94
6.1	Guía original del Trabajo Fin de Título	94

6.2	Instalación y configuración del sistema	97
6.3	Requisitos del sistema	97
6.4	Manuales de usuario	98
6.4.1	Argumentos de la línea de comandos	98
6.4.2	Ventana principal	100
6.4.3	Tareas comunes	103
6.4.4	Gestión de discos físicos	113
6.4.5	Uso del reproductor integrado	117
6.4.6	Integración con Spotify	121
6.4.7	Configuración	125
6.4.8	Anexo: Estándar Goldmine	129
7	Definiciones y abreviaturas	131
8	Bibliografía	133
8.1	Sitios web utilizados	133

Índice de ilustraciones

Ilustración 1.3-1 - Interfaz de Spotify, mostrando una lista de reproducción creada por el usuario	18
Ilustración 1.3-2 - Captura de TIDAL, en su versión Web	19
Ilustración 1.3-3 - Captura de Apple Music, sin cuenta	20
Ilustración 1.3-4 - Captura de YouTube Music, versión web	20
Ilustración 1.3-5 - Interfaz de Deezer	21
Ilustración 1.3-6 - Captura de Discogs en un álbum, mostrando todas sus variaciones	23
Ilustración 1.3-7 - Listado de ventas para un álbum	24
Ilustración 1.3-8 - Vista de la colección de Discogs	26
Ilustración 1.3-9 - Este álbum de Pink Floyd contiene etiquetas de tipo "remasterizado"	27
Ilustración 1.3-10 - Ejemplo de una canción con partes que podría ser juntada	27
Ilustración 1.3-11 - Captura de un álbum no disponible en Spotify por un problema de derechos de autor	28
Ilustración 1.7-1 - qBittorrent, aplicación hecha con Qt	32
Ilustración 1.7-2 - Audacity, hecho con WxWidgets	32
Ilustración 1.7-3 - Ejemplo de ventana con Java Swing	33
Ilustración 1.7-4 - La calculadora de Windows 10 está hecha con UWP	33
Ilustración 1.7-5 - Una aplicación hecha con WinForms sobre pájaros	34
Ilustración 1.9-1 - Microsoft Visual Studio 22	37
Ilustración 1.9-2 - Paint.net	37
Ilustración 1.12-1 - Cronograma del proyecto	40
Ilustración 2.1-1 - Una canción dividida en partes	43
Ilustración 2.3-1 - Diagrama de clases del núcleo del software	45
Ilustración 2.3-2 - Diagrama de casos de uso	46
Ilustración 2.3-3 - Diagrama de secuencia para añadir un álbum	46
Ilustración 2.3-4 - Diagrama de secuencia para vincular una cuenta de Spotify	47
Ilustración 2.3-5 - Diagrama de secuencia para descargar letras de internet	47
Ilustración 2.3-6 - Diagrama de secuencia para reproducir un archivo de sonido	48
Ilustración 2.3-7 - Diagrama de creación de disco físico	48
Ilustración 3.1-1 - Sketch del formulario principal	52
Ilustración 3.1-2 - Esbozo para los formularios de crear un álbum de manera manual.	54
Ilustración 3.1-3 - Boceto para la ventana de visualización	54
Ilustración 3.1-4 - Boceto del formulario para editar un álbum	55
Ilustración 3.1-5 - Ventana principal vacía	55
Ilustración 3.1-6 - Resultado del formulario de crear álbum	56
Ilustración 3.1-7 - Resultado del formulario de crear canción	56
Ilustración 3.1-8 - Resultado de la implementación de la visualización	56
Ilustración 3.1-9 - Resultado de la implementación de editar álbum	57
Ilustración 3.2-1 - Resultado del formulario de búsqueda	60
Ilustración 3.2-2 - Formulario de resultados	60
Ilustración 3.2-3 - Visualizado de canciones largas	60
Ilustración 3.3-1 - Asistente	62
Ilustración 3.3-2 - Configuración de foobar2000	63
Ilustración 3.3-3 - Formulario de configuración para cambiar idioma	63

Ilustración 3.4-1 - Boceto del reproductor y la lista de reproducción	65
Ilustración 3.4-2 – Resultado final del reproductor.....	66
Ilustración 3.4-3 - Reproductor sincronizando con Spotify	68
Ilustración 3.4-4 - Twit para compartir lo que se está escuchando, con opción a modificar el contenido.....	69
Ilustración 3.4-5 – Interfaz de la lista de reproducción.....	70
Ilustración 3.5-1 - Formulario de elegir disquetera.....	72
Ilustración 3.5-2 - Boceto del formulario de las letras	73
Ilustración 3.5-3 - Resultado del formulario de visor de letras	74
Ilustración 3.6-1 - Esbozo de la barra lateral	75
Ilustración 3.6-2 - Barra lateral	76
Ilustración 3.6-3 - Barra de herramientas de Notepad++	76
Ilustración 3.6-4 - La barra de herramientas implementada.....	77
Ilustración 3.6-6 - Filtrado en la barra de herramientas.....	79
Ilustración 3.6-5 - Resultado del formulario de filtros	79
Ilustración 3.6-7 - Formulario para crear discos de vinilo.....	79
Ilustración 3.6-8 - Resultado de añadir un álbum de vinilo con dos discos.	80
Ilustración 3.7-1 - Resultado de hacer clic en el botón de compartir un fragmento de una letra	83
Ilustración 3.7-2 - Álbum creado desde ficheros de música.....	84
Ilustración 4.2-1 - Fragmento de Google Maps en Madrid resaltando las tiendas de discos.....	89
Ilustración 4.3-1 - Herramienta para elegir la forma jurídica	91
Ilustración 6.4-1 - Directorio raíz del programa (donde reside el ejecutable)	99
Ilustración 6.4-2 - Menú contextual de Windows.....	99
Ilustración 6.4-3 - Ventana de propiedades con los parámetros.	99
Ilustración 6.4-4 - Ventana principal de Cassiopeia sin álbumes	100
Ilustración 6.4-5 - Ventana principal con discos y sin vista lateral.....	101
Ilustración 6.4-6 - La barra de herramientas con un álbum seleccionado	101
Ilustración 6.4-7 - Formulario para crear un álbum con datos de ejemplo.....	103
Ilustración 6.4-8 - Formulario para crear una canción.....	103
Ilustración 6.4-9 - El formulario transformado para crear una canción con partes.....	103
Ilustración 6.4-10 - Visualizando un álbum de ejemplo.	104
Ilustración 6.4-11 - Formulario para editar un álbum	105
Ilustración 6.4-12 - Un álbum con canciones bonus	106
Ilustración 6.4-13 - Álbum con una canción con partes.....	107
Ilustración 6.4-14 - Mensaje si se han buscado correctamente las canciones	108
Ilustración 6.4-15 - Log de la aplicación	108
Ilustración 6.4-16 - Mensaje de advertencia de que no se han podido vincular los archivos	108
Ilustración 6.4-17 - Una carpeta con un álbum, que vamos a meter en el sistema	109
Ilustración 6.4-18 - Ventana de letras.....	110
Ilustración 6.4-19 - Ventana para filtrar.....	111
Ilustración 6.4-20 - Demostración del filtro de canciones.....	112
Ilustración 6.4-21 - Demostración del filtro con la barra de búsqueda.....	112
Ilustración 6.4-22 - Asistente para la creación de CD.	113
Ilustración 6.4-23 - Asistente para la creación de un vinilo	114

Ilustración 6.4-24 - Ventana de diálogo preguntando si hace falta crear otro disco.	114
Ilustración 6.4-25 - Asistente de la creación de vinilos, en este caso con el segundo.....	115
Ilustración 6.4-26 - Visualizando la información de un vinilo.....	115
Ilustración 6.4-27 - Ventana de anotaciones.....	116
Ilustración 6.4-28 - Reproductor.	117
Ilustración 6.4-29 - Reproductor reproduciendo una canción.....	118
Ilustración 6.4-30 - Ventana de la lista de reproducción.	118
Ilustración 6.4-31 - Reproductor sincronizando con Spotify	119
Ilustración 6.4-32 - Ventana para elegir disquetera.	120
Ilustración 6.4-33 - Ventana para permitir los permisos en el navegador.....	122
Ilustración 6.4-34 - Ventana para buscar un álbum en Spotify con datos de ejemplo.	123
Ilustración 6.4-35 - Ventana con resultados de la búsqueda.	123
Ilustración 6.4-36 - Entrada de una aplicación vinculada en Spotify	124
Ilustración 6.4-37 - Opciones de idioma con una selección disponible durante el desarrollo de la aplicación.	125
Ilustración 6.4-38 - Configuración del portapapeles.....	126
Ilustración 6.4-39 - Configuración de la cadena del historial.....	126
Ilustración 6.4-40 - Ventana de personalización para la cadena de retransmisión.....	127
Ilustración 6.4-41 - Botones para configurar el tipo de letra de la aplicación. Tanto en el visor de letras como en las tablas.....	127
Ilustración 6.4-42 - Ventana para cambiar los colores con los valores predeterminados	128

Índice de tablas

Tabla 1 – Ecuaciones del modelo básico COCOMO	39
Tabla 2 - Costes en hardware y software	41
Tabla 3 - Conjunto de pruebas 1	57
Tabla 4 - Conjunto de pruebas 2	61
Tabla 5 - Conjunto de pruebas 3	64
Tabla 6 - Conjunto de pruebas 4	70
Tabla 7 - Conjunto de pruebas 5	74
Tabla 8 - Conjunto de pruebas 6	81
Tabla 9 - Matriz DAFO.....	89

1 ESPECIFICACIÓN DEL TRABAJO

En este capítulo se presenta la especificación del trabajo, con una estructura y contenidos **inspirados** en los criterios y recomendaciones que establece la norma UNE 157801:2007 - “*Criterios Generales para la elaboración de proyectos de Sistemas de Información*”.

A lo largo del documento se utilizarán términos y acrónimos cuya descripción aparecen en el apartado 7 (Definiciones y abreviaturas).

1.1 Introducción

Este trabajo consiste en la realización de un software de gestión discográfica para la indexación y tratamiento de colecciones de música en formatos tanto físicos como digitales, ya sean discos de vinilo o discos compactos. Este software busca ofrecer una solución integrada que incluya funcionalidades como la indexación de música y discos de fuentes físicas, la gestión y búsqueda de metadatos y letras de canciones, reproducción integrada en los formatos más extendidos y usados como la sincronización con algún sistema de *streaming* como Spotify.

Aunque ya existen esta clase de aplicaciones, existen de forma on-line. Se busca una solución fuera de línea para asegurar el problema de la privacidad del usuario y para dejar de depender de servicios en la nube ya que son de terceros y es posible que el usuario no le guste o su cobertura de Internet sea limitada, e incluso los servicios dejen de funcionar, haciendo imposible el trabajo del usuario.

1.2 Objetivos del trabajo

Los objetivos son:

- Diseñar e implementar el software anteriormente descrito.
- Analizar las virtudes y carencias del mercado *streaming* y obtener lo mejor.
- Observar el mercado de tiendas en línea y entender cómo organizar los álbumes.

- Crear un manual de usuario extenso para usuarios noveles de fácil seguimiento y lo más completo posible.

1.3 Antecedentes y estado del arte

Hay muchas personas que disfrutan la música, unos lo hacen de manera casual, otros se lo toman más en serio y se dedican a describir y dar su opinión en medios especializados (ej. *Pitchfork*) o incluso vendedores de discos con mucha tradición viajando entre ciudades organizando ferias locales o en su propia tienda.

Tarde o temprano llega el problema de organizar la música. Una colección musical puede llegar a crecer hasta límites insospechados, ya que cada semana salen nuevos álbumes, históricamente hubo muchísimos lanzamientos y habrá más joyas enterradas. Mantener todo esto de manera organizada es un problema complejo.

Supongamos un usuario que tiene en su haber una colección de discos en su estantería:

Existen varias soluciones, desde la propia persona que con su propia cabeza organiza toda su colección de memoria, no es la mejor ya que a veces la memoria sufre de fallos (olvido o dificultad al recordar) y llega a ser imposible recordar todos los discos que puede llegar a tener y dónde están localizados.

Otra es la clásica manera de mantener todo con papel y lápiz, apuntando minuciosamente (o brevemente) cada uno de los álbumes de la colección. Es eficiente ya que no consume electricidad, pero al final es poco práctico ya que las búsquedas en esa lista pueden demorarse bastante y también el consumo de papel no es la mejor solución en el ámbito medioambiental. Otra cosa que puede pasar, es que el usuario pierda el papel y por ello pierda la base de datos.

La tecnología avanza a ojos vistas y con los ordenadores ya se empezaron a desarrollar aplicaciones multimedia para escuchar música en formato digital. Todo esto más la fusión de la computadora con el disco compacto, terminó por mezclar la música y la informática.

Originalmente, los primeros programas para reproducir / gestionar música eran Winamp o el propio Windows Media Player de Microsoft. Fueron programas muy populares durante la década de los años 00. En los años 90, sólo se preocupaba sobre el problema de reproducir el contenido en el ordenador. Luego, surgió la plataforma iTunes de Apple, lanzado en 2001, que permitía comprar

música en línea y añadirla a tu colección virtual. Permitía sincronizar los contenidos entre los dispositivos de Apple.

Por último, los servicios en *streaming* están revolucionando la manera en la que el usuario escucha la música y ahora todo se realiza de forma on-line.

1.3.1 Situación actual

Si realizamos un análisis de la situación actual, no ha sido posible encontrar un software que resuelva los requisitos planteados, la idea es hacer un programa estilo *navaja suiza*, que haga de manera básica un poco de todo y presentar este prototipo.

Realmente no interesa depender de una base de datos externa. Aunque es la mejor opción por la cantidad abrumadora de datos, se ha optado por una solución fuera de línea para que el usuario personalice al máximo su colección y así añadir, la posibilidad de tener varias bases de datos a la vez.

1.3.1.1 Servicios de streaming

Hoy en día, la opción más popular son los servicios en *streaming*, que hacen de todo en uno, simplemente. Las opciones más conocidas (y que se hacen competencia entre sí) son *Spotify*, *TIDAL*, *Apple Music*, *YouTube Music*, *Deezer*, *Amazon Prime Music*. Todas estas plataformas ofrecen prácticamente el mismo contenido a un precio muy parecido, por no decir el mismo. Música grabada reproducida en directo, opción de poder guardar los álbumes en una colección personal, crear ilimitadas listas de reproducción, que la propia plataforma te sugiera contenidos... etc.

Estos servicios son muy parecidos entre sí, pero tienen sutiles diferencias entre ellos que al final son determinantes para los consumidores a la hora de elegir un servicio y a realizar el propio pago.

Spotify

Parte de la ventaja es que, es el que lleva más años en el sector (2008) y cuenta con cierta veteranía. Es bastante asequible (9,99 € al mes, 4,99 € opción de estudiantes) y cuenta con una calidad de sonido bastante aceptable. OGG a 320kbps. También tiene un catálogo interesante de podcasts. A mi juicio, es el mejor servicio ya que me satisface por el catálogo de discos que tiene. También, desde hace poco, dispone de la posibilidad de ver las letras de las canciones a tiempo real. Las

desventajas son, que es el único servicio sin calidad de alta definición y que sus aplicaciones para escritorio, son bastante ineficientes, ya que son aplicaciones web.

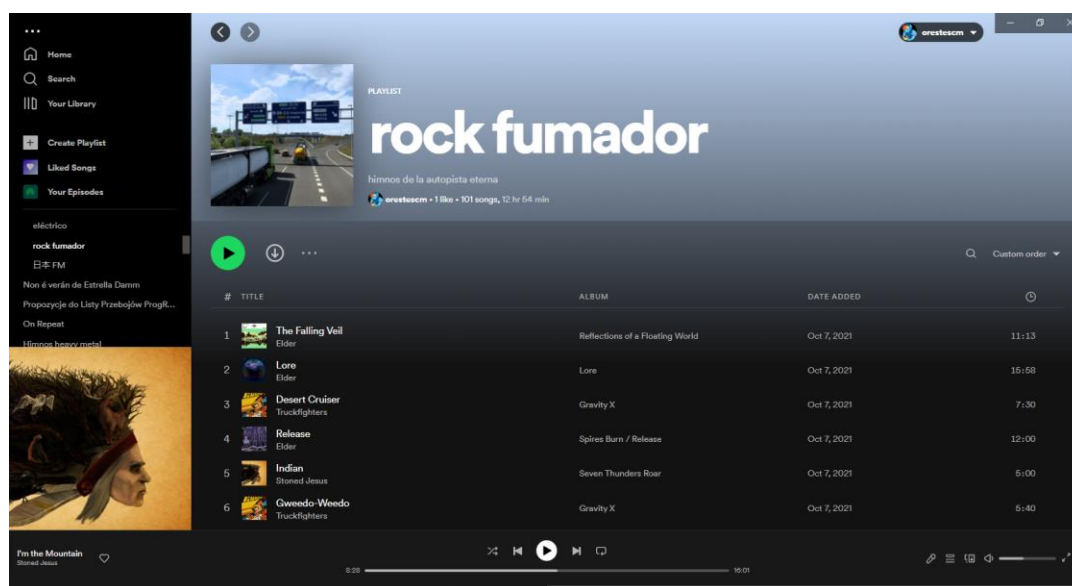


Ilustración 1.3-1 - Interfaz de Spotify, mostrando una lista de reproducción creada por el usuario

Dispone de una versión gratuita, pero es bastante tediosa ya que contiene muchos anuncios, no te deja descargar los contenidos sin conexión y en el teléfono, no te deja seleccionar las canciones al gusto, estás limitado con el modo aleatorio.

La interfaz es bastante cómoda de usar.

TIDAL

TIDAL es un servicio más reciente que se hace publicitar por su calidad en alta definición que puede atraer a usuarios más audiófilos. El precio es 9,99 € para el servicio estándar, calidad AAC a 160kbps, con *HiFi* normal, FLAC a 1411 kbps, calidad CD y 19,99 € si quieres la opción más *premium* con la calidad de sonido MQA¹, que ya puede llegar hasta los 10 Mbps. También es posible descargar los contenidos. No dispone de versión gratuita. También disponen de una oferta especial con Vodaphone. Contiene un catálogo bastante extenso con más de 80 millones de canciones.

También tiene la opción de descargar cualquier contenido para reproducirlo sin conexión, crear ilimitadas listas de reproducción...

¹ MQA es tecnología de sonido que ofrece audio a calidad de estudio directamente para el consumidor con un tiempo de descarga razonable.

Una clara desventaja es, que, no tiene el mismo catálogo que Spotify, ya que hay álbumes que echo en falta, aunque eso es tema de derechos y pasa en el resto de servicios.

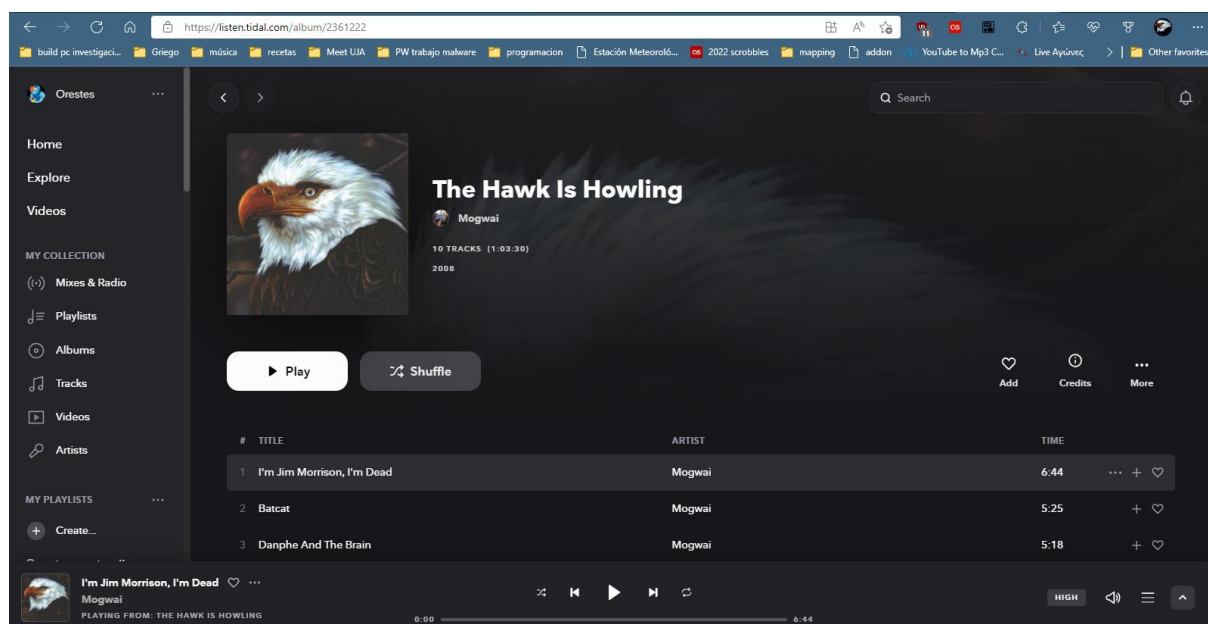


Ilustración 1.3-2 - Captura de TIDAL, en su versión Web

Apple Music

Es la versión en *streaming* de la compañía de Cupertino, California, aunque éste sigue existiendo². Apple Music es la competencia a los otros servicios del sector, con su propio catálogo.

El precio estándar es de 9,99 € al mes, cuyas funcionalidades incluyen la reproducción ilimitada de canciones, conciertos y contenidos exclusivos, audio espacial con Dolby Atmos más descarga de contenidos para reproducirlos sin conexión.³ Dispone de calidad especial de master en 192kHz de hasta 24 bits por muestreo. También disponen de una suscripción reducida llamada Voice, que dispone de la reproducción sin límites, pero te impide la opción de descargar la música y de investigar las letras de las canciones. Sin embargo, esta suscripción es exclusiva para aquellos usuarios que ya disponen de un dispositivo Apple como un iPhone. Cuesta 4,99 € al mes.

² Sólo está disponible en sistemas Windows.

³ Apple Music desde hace poco ofrece audio espacial con Dolby Atmos, pero se requiere un hardware y un software compatibles y no todo el contenido está disponible.

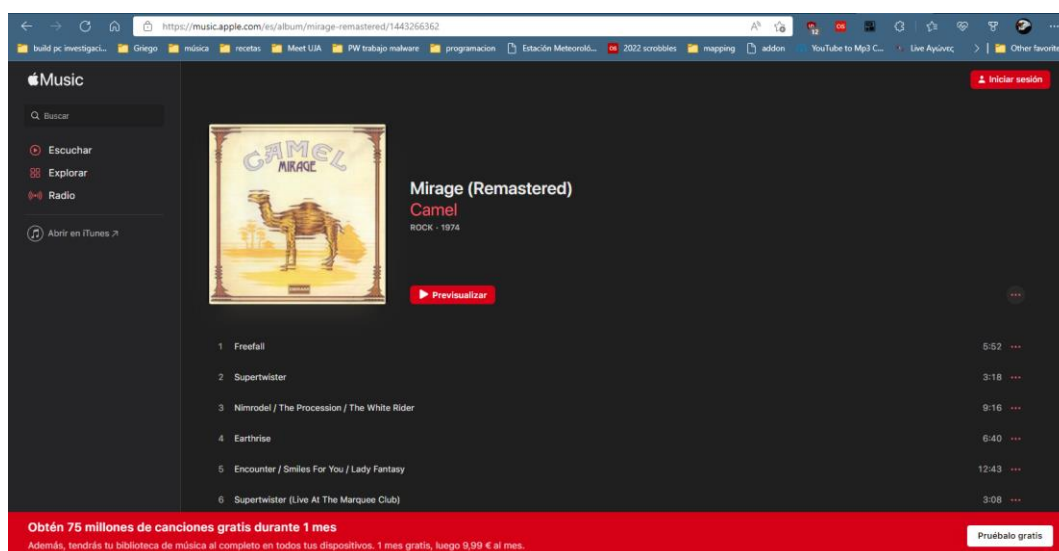


Ilustración 1.3-3 - Captura de Apple Music, sin cuenta

YouTube Music

Es la opción de Google tras la desaparición de Google Play Music.

Está completamente integrado en YouTube, lo cual, a priori sugiere que tiene una versión gratuita. Y es que, YouTube está plagado de canciones autogeneradas por Google, que es el propio catálogo de este servicio. Lo malo es que, en el teléfono es imposible escuchar música ya que, si lo bloqueas o apagas la pantalla, la música se detiene, incluso si se cambia de aplicación. Te obliga a mantener la pantalla encendida lo cual gasta mucha batería. En cambio, en la web está bastante bien ya que la reproducción en segundo plano es implícita, cambias de pestaña y la música no se detiene. (y el bloqueador de anuncios funciona)

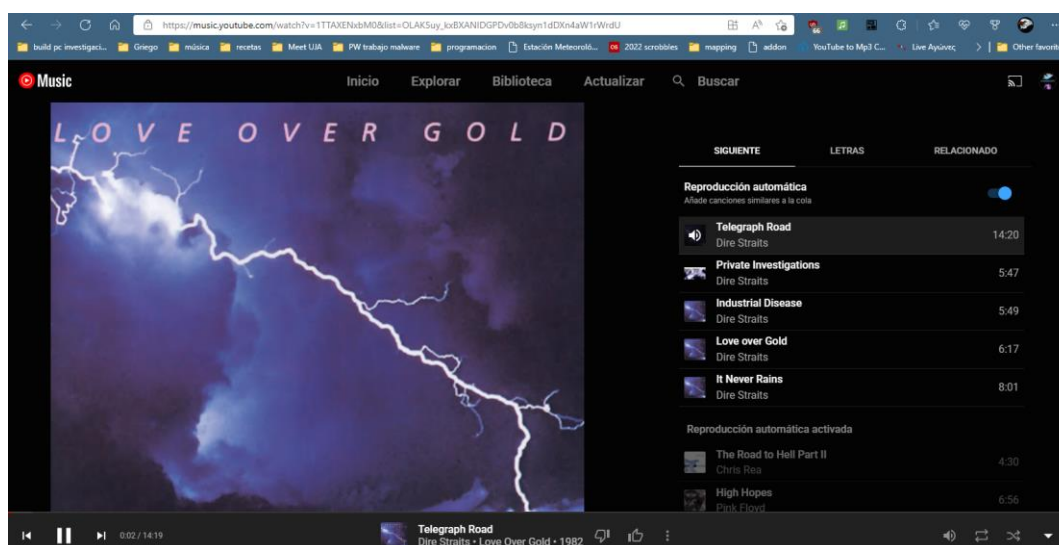


Ilustración 1.3-4 - Captura de YouTube Music, versión web

El catálogo puede ser la mejor opción ya que, vídeo que está subido a YouTube, sea autogenerado o no, puede ser reproducido en YouTube Music. A no ser que el propio contenido esté bloqueado por derechos de autor en nuestro país.

El precio de la suscripción mensual son 9,99 € y ofrece la descarga de contenidos al teléfono, reproducción en segundo plano en el teléfono y la eliminación de anuncios sin necesidad de un bloqueador de terceros.

Deezer

Es una compañía francesa que, a pesar de haberse creado antes que Spotify, nunca gozó de la popularidad de la ya mencionada, pero en los últimos años ha experimentado un notable crecimiento en cuanto a número de usuarios y a catálogo. Dispone de suscripción gratuita con publicidad con bastantes funcionalidades como disponibilidad de podcasts, reconocedor de canciones al estilo de Shazam integrado, un visor de letras que se sincroniza con la canción que estés escuchando más integración con aparatos interactivos como el Amazon Echo.

Su suscripción de pago, que cuesta 10,99 € al mes, ofrece reproducción a la carta sin límites, eliminación de la publicidad, descarga sin conexión de contenidos y sonido de alta fidelidad con el códec FLAC a calidad CD.

También tienen un servicio que aprenden del usuario y a partir de lo que escucha te recomienda nuevos estilos musicales o nuevos artistas afines al gusto del usuario, basados en colores.

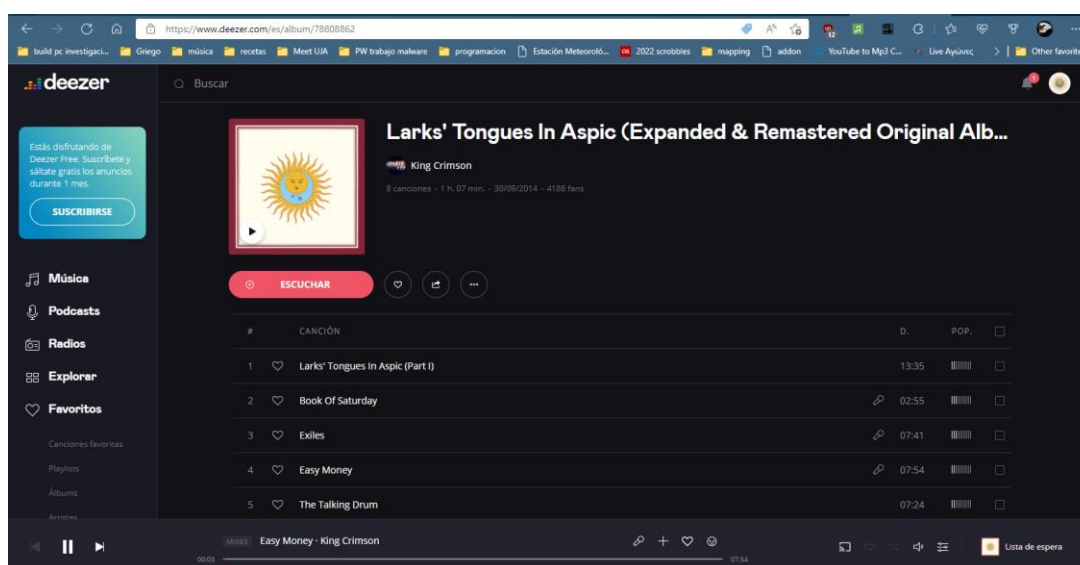


Ilustración 1.3-5 - Interfaz de Deezer

Amazon Prime Music

Es una extensión de la suscripción de Amazon Prime. El problema que presenta es que, si no deseas pagar más, estás limitado a 2 millones de canciones. Es un catálogo bastante limitado en comparación con el resto de servicios vistos anteriormente. No puedes crear listas de reproducción (tienes unas por defecto de Amazon) pero en cambio, puedes descargar las canciones disponibles para escucharlas sin conexión.

Ahora, si se desea acceder a la suscripción *Unlimited*, desbloqueas el resto del catálogo (90 millones de canciones), calidades de audio superiores, creación de listas de reproducción, podcasts... entre otros. Destaca en la calidad de audio de *Ultra Alta Definición* que ofrece audio de 24 bits a 192kHz, que es muy buena calidad y es considerada de audiófilos. Cuesta 9,99 € al mes. Si uno es cliente de Amazon y no le importa gastar más dinero, pienso que es la opción ideal.

1.3.1.2 Mercado en línea - Discogs

Ya finalizando con el mercado *streaming*, vamos a pasar al mercado en línea y pienso que uno de los sitios más populares entre vendedores y coleccionistas de segunda mano es *Discogs*.

Discogs es un sitio web de compra-venta en línea de discos de segunda (y primera) mano similar a *Ebay*. Hay vendedores y directamente compras a un particular. En ocasiones puedes toparte con tiendas de discos profesionales que venden por ese medio como alternativa a una venta en línea.

A su misma vez, actúa como base de datos comunitaria. Esto es, que un usuario puede añadir una variación de un disco en físico de un álbum. Un álbum puede sufrir innumerables reediciones y remasterizados. Cada una puede tener ligeras variaciones en el sonido y calidad de la música, pueden venir de diferentes *masters*⁴.

Esto, tiene gran valor de coleccionista y algunas copias pueden llegar a venderse por miles y miles de euros.

⁴ Los álbumes, por lo menos antiguamente, se grababan y se guardaban en cintas magnéticas de varias pistas y de muy buena calidad. Luego hacían copias y las mandaban a plantas de producción, donde se estampan los discos de vinilo o los discos compactos.

King Crimson – In The Court Of The Crimson King (An Observation By King Crimson)

Genre: Rock
Style: Prog Rock, Symphonic Rock
Year: 1969
Notes: Official release date as per Melody Maker: October 10, 1969.

Tracklist

21st Century Schizoid Man (Including "Mirrors")	6:52
I Talk To The Wind	5:40
Epitaph (Including "March For No Reason" And "Tomorrow And Tomorrow")	8:30
Moonchild (Including "The Dream" And "The Illusion")	12:09
The Court Of The Crimson King (Including "The Return Of The Fire Witch" And "The Dance Of The Puppets")	8:48

Credits (10)

- Greg Lake: Bass Guitar, Lead Vocals
- Barry Godber: Cover
- Michael Giles: Drums, Percussion, Vocals
- Robin Thompson: Engineer
- Tony Page: Engineer
- Robert Fripp: Guitar

Versions (384)

Title (Format)	Label	Cat#	Country	Year
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album)	Island Records, Island Records	ILPS 9111, ILPS-9111	UK	1969
In The Court Of The Crimson King An Observation By King Crimson (LP, Album, Club)	Atlantic	SD 8245	US	1969
In The Court Of The Crimson King An Observation By King Crimson (LP, Album, Mono, Promo, Alt)	Atlantic	8245	US	1969
In The Court Of The Crimson King An Observation By King Crimson (LP, Album, Promo, CTH)	Atlantic	SD 8245	US	1969
In The Court Of The Crimson King An Observation By King Crimson (LP, Album, RI)	Atlantic	SD 8245	US	1969
In The Court Of The Crimson King (An Observation By King Crimson) (8-Trk, Album)	Atlantic	A 8TC 8245	Canada	1969
In The Court Of The Crimson King (An Observation By King Crimson) (8-Trk, Album)	Atlantic	M 88245	US	1969
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album)	Atlantic	SD 8245	US	1969
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album)	Atlantic	SD 8245	Israel	1969
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album)	Island Records, Island Records	ILPS 9111, ILPS-9111	UK	1969
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album)	Island Records, Island Records	ILPS 9111, ILPS-9111	UK	1969
In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, CTH)	Atlantic	SD 8245	US	1969

Ilustración 1.3-6 - Captura de Discogs en un álbum, mostrando todas sus variaciones.

Como podemos ver en la imagen, están disponibles las múltiples variantes de un álbum, su país de origen, año de salida y códigos internos. Todos estos parámetros distinguen un disco de otro. Luego, arriba, podemos acceder a la venta.

Accederemos a una página donde podemos ver la venta y los distintos discos que hay a la venta, su precio, vendedor, estado.. etc.

De esta manera, estamos dentro de una tienda de discos online enorme donde cada vendedor tiene su propio “*stand*” y podemos mirar todo lo que queremos.

También tenemos la disponibilidad de hablar por privado con los distintos vendedores para acordar temas como el envío o la disponibilidad de los artículos.

You Selected: *
 Year: 1969
 Format Description: LP
 Release: King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (All Versions)

Ships From
 United States 76
 United Kingdom 43
 Italy 14
 Canada 6
 France 6
 Show more...

Currency
 USD (\$) 87
 EUR (€) 48
 GBP (£) 43
 AUD (A\$) 5
 CAD (CA\$) 5
 Show more...

Style
 Prog Rock 193
 Symphonic Rock 90

Format Description
 Album 193
 Stereo 98
 Club Edition 16
 Promo 2
 Mono 1
 Show more...

Media Condition
 Very Good Plus (VG+) 83
 Very Good (VG) 56
 Near Mint (NM or M-) 23
 Good Plus (G+) 20
 Mint (M) 5
 Show more...

More Filters
 Make an Offer 70
 Seller...

Shop King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP)
 1 - 26 of 193 < Prev Next > Sort: Listed, Condition, Artist, Title, Label Seller Price Show: 25

Item	Price	Shipping	Total	Actions
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, Pre) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good (VG) ⊕ Sleeve Condition: Very Good (VG) seam split View Release Page	\$20.00	+\$63.00 shipping	about €76.04 total	Add to Cart Details
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, Pit) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good (VG) ⊕ Sleeve Condition: Very Good Plus (VG+) ⊕ Just a few light marks on the vinyl that may cause some noise. Pitman press (CP on the labels). Includes the original Atlantic inner sleeve. The cover has very little wear. (Just a bit on the inside of the gatefold and a bit on the lower back left.) View Release Page	\$35.00	Unavailable in Spain	about €32.32 +shipping	
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, SP) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good (VG) ⊕ Sleeve Condition: Very Good (VG) Really nice all around condition. Original Atlantic inner pic sleeve. Some minor shelf wear. \$5 unlimited US shipping View Release Page	\$44.99	+\$28.00 shipping	about €67.39 total	Add to Cart Make Offer / Details
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, CTH) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good Plus (VG+) ⊕ Sleeve Condition: Very Good Plus (VG+) ⊕ Nice copy just off NM, minor shelfwear on cover. \$5 unlimited US shipping View Release Page	\$60.00	+\$28.00 shipping	about €99.72 total	Add to Cart Make Offer / Details
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, MO) Label: Atlantic Cat#: SD 8245 Media Condition: Good Plus (G+) ⊕ Sleeve Condition: Very Good Plus (VG+) ⊕ LP is visually VG- with scuffs and light marks, but has great luster and plays pretty well overall. Sleeve is VG+ with light wear and a name lightly penned in the top left corner. View Release Page	\$27.99	+\$26.00 shipping	about €49.85 total	Add to Cart Details
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, Pit) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good (VG) ⊕ Sleeve Condition: Good Plus (G+) ⊕ LP is a nice VG with some light marks, but has great luster and plays through quite well. Outer sleeve is VG- with some staining and tearing along bottom seam. But looks solid otherwise. JAM CITY!!! View Release Page	\$24.99	+\$26.00 shipping	about €47.08 total	Add to Cart Details
King Crimson - In The Court Of The Crimson King (An Observation By King Crimson) (LP, Album, CTH) Label: Atlantic Cat#: SD 8245 Media Condition: Very Good Plus (VG+) ⊕ Sleeve Condition: Very Good (VG) disc has light marks and scuffs; plays great; jacket has mild wear to edges (small paper tear at bottom right corner); corners, and spine; company inner included; great copy overall. View Release Page	\$45.00	+\$30.00 shipping	about €69.25 total	Add to Cart Make Offer / Details

Ilustración 1.3-7 - Listado de ventas para un álbum

Algo que es crucial es, indicar la condición del artículo. Hay una escala *de facto* que usa Discogs para indicar la condición del medio a escuchar y de las carátulas y folletos, se llama estándar Goldmine.

- **Mint (M)**

Se trata de un artículo nuevo, sin abrir, todavía con el plástico que envuelve de protección.

- **Near Mint (NM o M-)**

Es un artículo con muy poco uso, apenas hay rasgos de desgaste. Igual que los coches Km 0 en los concesionarios.

- **Very Good Plus o Excellent (VG+ o E)**

Es un artículo que está en excelente estado, está usado, pero ha tenido mucho mimo por parte del anterior dueño.

- **Very Good (VG)**

No tiene la *frescura* de un disco nuevo de fábrica, es obvio que ha sido usado, al reproducirlo tiene ruido audible, pero no molesta a la experiencia. Se notarán rasgos de uso, pero menores.

- **Good (G)**

Estos artículos costarán en torno a un 20% menos que un artículo NM. No es malo, está en buen estado, pero es claramente obvio que es de segunda mano. El vinilo tendrá claros rasgos de uso y desgaste. Si es un disco común puedes intentar encontrar algo en mejores condiciones, pero si tienes prisa... no es mala opción. En cuanto a la calidad de sonido, habrá ruidos claros y pops en el vinilo y puede que sea un poco molesto. En cuanto a un CD, es raro que salte, pero tenlo en cuenta. Los libretos puede que estén escritos y dañados levemente.

- **Fair (F)**

Está en mal estado, pero está justo que llega a ser aceptable mantenerlo en la colección. El vinilo tendrá bastantes *pops* y ruido de polvo. Existe una posibilidad de que esté deformado. Quizás hasta falten libretos y otros añadidos con el CD. El CD puede que tenga problemas al reproducirse.

- **Poor (P)**

Este artículo está en muy mal estado, que apenas tiene valor. Es raro ver un artículo en estas condiciones porque nadie te lo va a comprar.

Por último, vamos a comentar la función de coleccionismo de *Discogs*. Se puede añadir a la colección cualquier álbum de la base de datos, con el estado que el usuario considere oportuno. De esta manera, ve de un plumazo los discos que tiene en la colección y de manera detallada de qué versión y año es el disco. También se puede consultar un valor aproximado de la colección en euros, basados en las últimas ventas.

Dashboard **Collection** Wantlist Lists Submissions Drafts Export Settings

Search Collection Manage Folders Manage Custom Fields Change Currency Export My Collection

Collection Value: Min €986.96 Med €1.439.53 Max €2.389.43 Sort Newest Additions Folder All (105) Show 25

Remove Selected cd (18) Move Selected

Artist, Title, Label, Year, Format	Min	Median	Max	Added	Folder	Rating	Notes
Museo Rosenbach - Zarathustra CD, Album, RE Contempo Records CONTEDISC 004 Italy	€18.02	€23.08	€28.15	11 months ago	Uncategorized	☆☆☆☆ Very Good Plus (VG+) Near Mint (NM or M-)	
Dream Theater - Distance Over Time CD, Album Inside Out Music, Sony Music IOMCD 523, 19075915292 2019 Europe	€4.62	€8.03	€8.82	about 1 year ago	cd	☆☆☆☆ Near Mint (NM or M-) Near Mint (NM or M-)	comprado ecl 17-12-20 gratis
Talking Heads - Remain in Light CD, Album, RE Sire 7599-26995-2 1997 Europe	€1.89	€3.95	€11.91	about 1 year ago	cd que he comprado	☆☆☆☆ Very Good Plus (VG+) Good (G)	jewel case rota
Camilo Sesto / Teddy Bautista / Angela Carrasco - Jesucristo Superstar (Versión Original En Español) 2xCD, Album, RE SONY BMG Music Entertainment, Ariola 82876 759012 2005 Spain	€8.05	€12.46	€17.82	about 1 year ago	cd	☆☆☆☆ Near Mint (NM or M-) Near Mint (NM or M-)	comprado ecl 25/09/20 gratis.
Haken (3) - Virus 2xLP, Album + CD, Album Inside Out Music, Sony Music IOMLP 549, 19439744731 2020 Europe	€23.46	€26.01	€36.52	about 1 year ago	lp que he comprado yo	☆☆☆☆ Near Mint (NM or M-) Near Mint (NM or M-)	recibido 07-08-20
Elio E Le Storie Tese - Italian, Rum Casusu Çikti CD, Album Hukapan Records, Hukapan Records HUK 471553 2, 471553 2 1992 Italy	€4.00	€8.03	€20.06	about 1 year ago	cd que he comprado	☆☆☆☆ Very Good Plus (VG+) Very Good Plus (VG+)	todo bien
Elio E Le Storie Tese - Eat The Phikis CD, Album Aspiene 74321-36094-2 1996 Italy	€3.91	€5.02	€10.94	about 1 year ago	cd que he comprado	☆☆☆☆ Very Good (VG) Very Good (VG)	todo bien
Le Orme - Felona E Sorona CD, Album Philips 842 507-2 1990 Italy	€6.02	€10.49	€15.05	about 1 year ago	cd que he comprado	☆☆☆☆ Very Good Plus (VG+) Very Good Plus (VG+)	todo bien
Pink Floyd - Obscured By Clouds CD, Album, RE Pink Floyd Records PFR7 2020 Spain	€11.26	€11.26	€11.26	over 2 years ago	cd	☆☆☆☆ Near Mint (NM or M-) Near Mint (NM or M-)	
Fruteria Toñi - El Porvenir Está En Las Huevas CD, Album Not On Label (Fruteria Toñi Self-released) none 2020 Spain	€14.10	€14.61	€15.69	over 2 years ago	cd que he comprado	★★★★★ Near Mint (NM or M-) Near Mint (NM or M-)	
Fruteria Toñi - Tengo Mis Dias Buenos CD, Album Not On Label (Fruteria Toñi Self-released) none 2017 Spain	€16.61	€16.78	€21.28	over 2 years ago	cd que he comprado	★★★★★ Near Mint (NM or M-) Near Mint (NM or M-)	comprado en crowdfunding

Ilustración 1.3-8 - Vista de la colección de Discogs.

1.3.2 Resumen de las deficiencias y carencias identificadas

En cuanto al mercado *streaming*, existe una clara deficiencia común a todos los servicios, sean gratuitos o de pago y es que, son en línea. Ningún servicio dispone de una disponibilidad del 100% y puede pasar que en un momento determinado pase cualquier cosa y el servicio se quede sin disponibilidad durante un tiempo. Aunque existe la posibilidad de descargar las canciones, no son realmente tuyas y las estás “alquilando”. Es imposible acceder a esas canciones descargadas si no es a través de ese *software*.

Otra deficiencia que he notado es la dificultad de organizar ciertos contenidos, ya sean los propios álbumes o las canciones.

Muchas veces, cuando un artista publica un álbum, más tarde puede que la discográfica lo *remasterice* para otro medio, la mayoría de veces añadiendo canciones extra. Esto está bien, el problema es cuando la discográfica le añade etiquetas al nombre del álbum y de las canciones y el resultado final es bastante molesto y poco práctico.

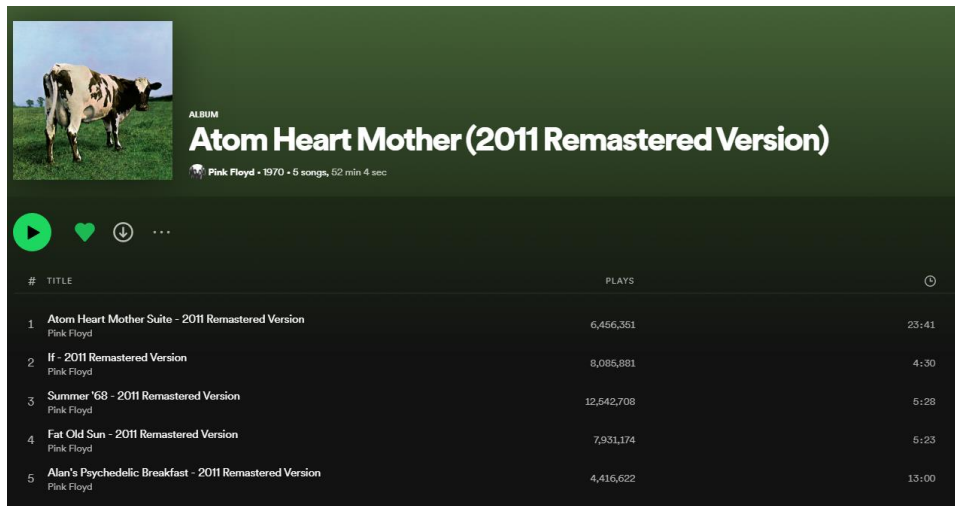


Ilustración 1.3-9 - Este álbum de Pink Floyd contiene etiquetas de tipo "remasterizado"

También existen etiquetas de tipo *Edición Deluxe*, *Edición Super Deluxe*, *Edición 20 aniversario Digital...* etc. Al final, resulta irrisorio y muy molesto tener este tipo de etiquetas. Lo ideal es deshacerse de todas estas etiquetas y si fuera necesario, añadir una nota aparte indicando este tipo de cosas. Personalizar los metadatos de las canciones sería una solución interesante y sólo tener una versión de cada álbum, sin distinguir que sean reediciones.

Otra carencia que he detectado, ya es algo más personal, es sobre las canciones *con partes*. Hay canciones que se dividen en varios actos y hay servicios de *streaming* que las separan y otros que las juntan. Lo mejor es dejarlo decidir a cada usuario, aunque lo más común es escuchar la canción entera por muchas partes que tenga y saber la duración total.

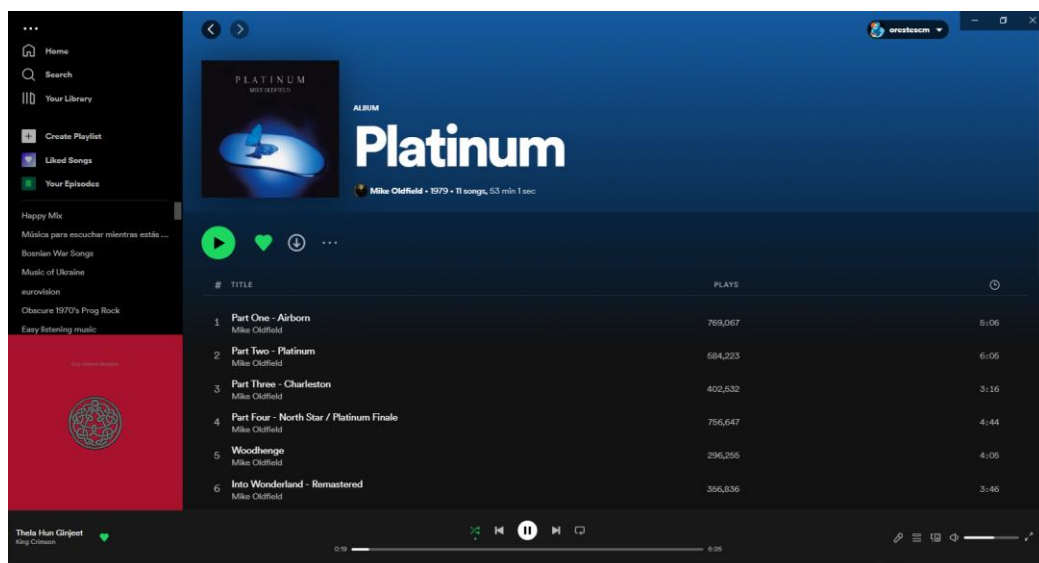


Ilustración 1.3-10 - Ejemplo de una canción con partes que podría ser junta

Por último, una de las carencias más importantes que se han identificado, es la inconsistencia de la disponibilidad de cierto contenido musical. En algunos servicios puede que esté un álbum, pero en otro no. Esto ocurre porque los derechos de un álbum pertenecen a una discográfica e impidan la venta o la reproducción *streaming*. O con otra calidad distinta. Sería interesante buscar una solución donde esté todo unificado.

A veces pasa, que directamente un álbum no está disponible en ningún servicio de *streaming*, así imposibilitando la escucha desde cualquier servicio y al final, la única manera de disfrutar el contenido es, o comprando el álbum o descargándolo por Internet.

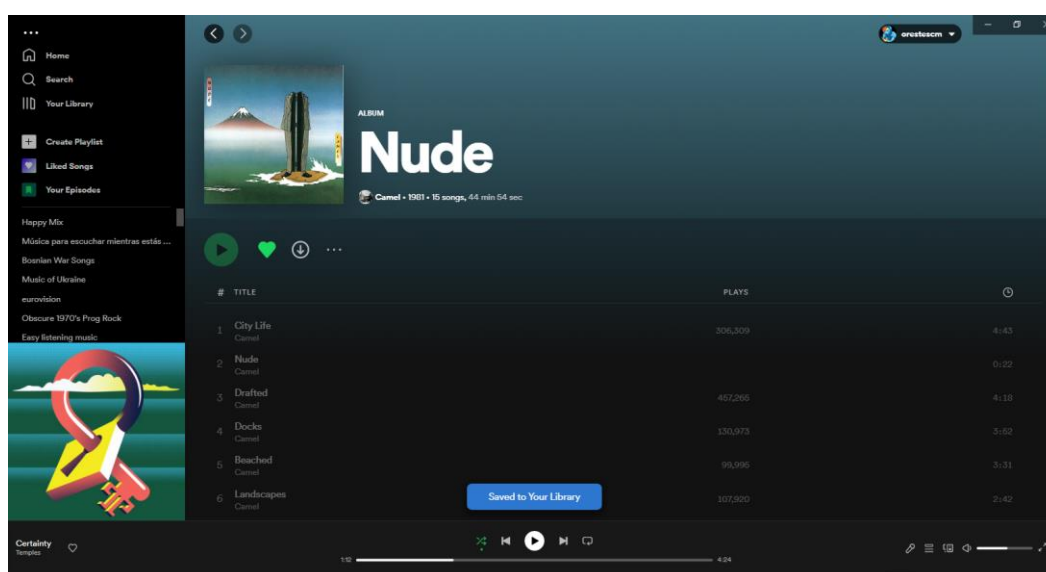


Ilustración 1.3-11 - Captura de un álbum no disponible en Spotify por un problema de derechos de autor.

1.4 Requisitos iniciales

Los requisitos iniciales del software son los siguientes:

- **Crear una colección musical.** El usuario debe poder crear y observar su colección de una vista rápida, viendo así los álbumes que tiene registrados.
- **Crear un álbum.** Debe haber una funcionalidad para crear álbumes paso a paso con todos los datos posibles, más todas sus canciones y dar la mayor flexibilidad al usuario posible, permitiendo borrar y editar lo que sea necesario.
- **Descargar metadatos de Internet.** Se debe poder descargar los metadatos de Internet, de una base de datos, por ejemplo, Spotify, para evitar la acción monótona de añadir manualmente todos los álbumes uno a uno.
- **Sincronización con Spotify.** En estos tiempos recientes, los servicios de *streaming* han conquistado casi todo el mercado musical, dejado casi apartados al mercado de álbumes físicos y de gestión. Por lo tanto, debería existir la posibilidad de vincular una cuenta de Spotify al *software* para sincronizar la reproducción como la biblioteca del usuario para que tenga todos sus álbumes disponibles.
- **Interfaz familiar.** El *software*, al ser de gestión, debe tener una interfaz sencilla para un usuario casual, que no tenga demasiados conocimientos técnicos. A más familiar sea la interfaz, más atractivo tendrá para el usuario final. Se valora la sencillez de uso.
- **Reproducción.** El *software* deberá ser capaz de reproducir ficheros de audio para que el usuario tenga sincronizada su colección local con el programa. Si se incluye este requisito, se deberá tener un sistema básico de gestión de listas de reproducción.
- **Reproducción de CD.** Debería existir la posibilidad de reproducir discos compactos con la ayuda de las funciones del sistema operativo.
- **Crear un símil de discos físicos.** El *software* debe de ser capaz de crear una representación de un disco en físico con los metadatos disponibles. Es necesario que también se implemente el estándar *Goldmine* para

representar el estado físico del medio, así determinando si está en buen estado o está en un estado deplorable.

- **Registro de letras de canciones y buscador.** Se debe poder manejar las letras de las canciones de manera local y, por ende, incluir un buscador para conseguir las letras automáticamente de manera transparente al usuario.
- **Filtrado de la vista.** La vista principal debe poder filtrarse en el caso de que existan muchos álbumes en el sistema, poder filtrarlos y así buscar y seleccionar los que en un momento interesen.
- **Configuración del software.** El *software* debe poder configurarse al gusto del usuario y debe de tener numerosas opciones.
- **Idiomas.** El *software* debe ser capaz de soportar varias lenguas para abarcar el mayor público posible.
- **Posibilidad de crear canciones con partes.** Hay canciones que tienen varias partes y aunque suenen juntas seguidas, siempre se separan, el programa debe permitir la gestión para fusionar esas mini-canciones y así crear una canción con partes.

1.5 Alcance

Contenido del trabajo:

- **Memoria del trabajo.** Es la memoria asociada a este trabajo donde se explican los requisitos, presupuesto, precedentes del arte, documentación, etc.
- **Ejecutable.** Ejecutable del software que cumpla con los requisitos mencionados anteriormente, con las comprobaciones pertinentes de que funciona correctamente.
- **Manual de usuario.** Manual de usuario donde se explique con detalle el funcionamiento del programa, de todas sus funciones, como apoyo al usuario en el caso de que desconozca el funcionamiento.

1.6 Hipótesis y restricciones

El TFT se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto será de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Por consiguiente, la principal restricción aplicable es la limitación de la duración del trabajo.

Se aplica más restricciones por el presupuesto, se usará la mayor cantidad de software libre y con licencias gratuitas para limitar el número de costes.

1.7 Estudio de alternativas y viabilidad

Para el desarrollo del proyecto se han tenido en cuenta varias herramientas, librerías, *frameworks* y lenguajes de programación.

- *Qt con C++*
- *WinForms con C#*
- *WxWidgets con C++*
- *Swing con Java*
- *UWP con C#*

Todos los *frameworks* mencionados se usan para desarrollar aplicaciones que utilizan una interfaz gráfica de usuario.

Qt parte de la ventaja de que es multiplataforma y también es software libre, lo cual cumple con el requisito de que este proyecto sea *software* libre. La desventaja es

que es complejo de programar, ya que funciona bajo C++ y suele dar bastantes problemas a la hora de implementar y su *framework* ocupa muchos recursos de espacio de almacenamiento, por lo tanto, ocuparía mucho tiempo para instalarse.

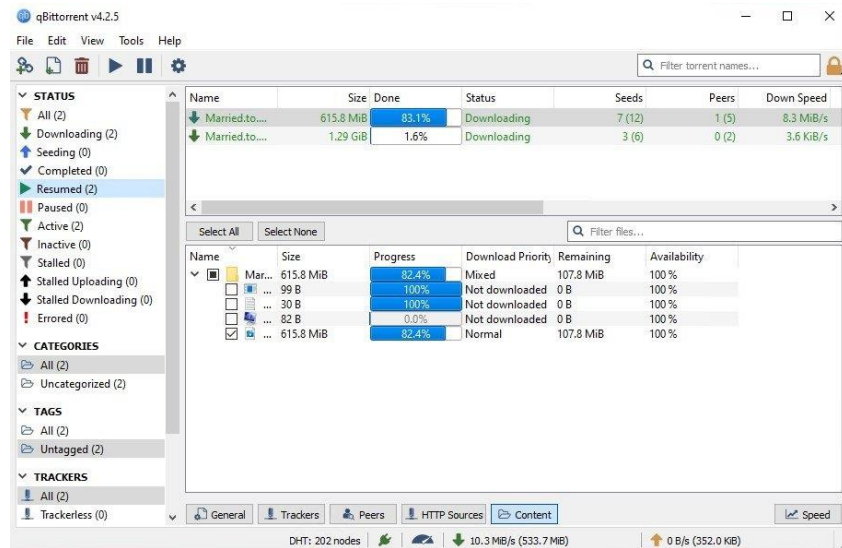


Ilustración 1.7-1 - qBittorrent, aplicación hecha con Qt

WxWidgets son una serie de bibliotecas libres para producir interfaces gráficas nativas de forma bastante eficiente y es muy portable a otros sistemas operativos. La desventaja, es que toda la programación del diseño de las ventanas es manual, carece de diseñador, lo cual hace que el proceso de implementación sea mucho más laborioso. También, hay que tener en cuenta que funciona con C++, ergo la programación vuelve a ser compleja.

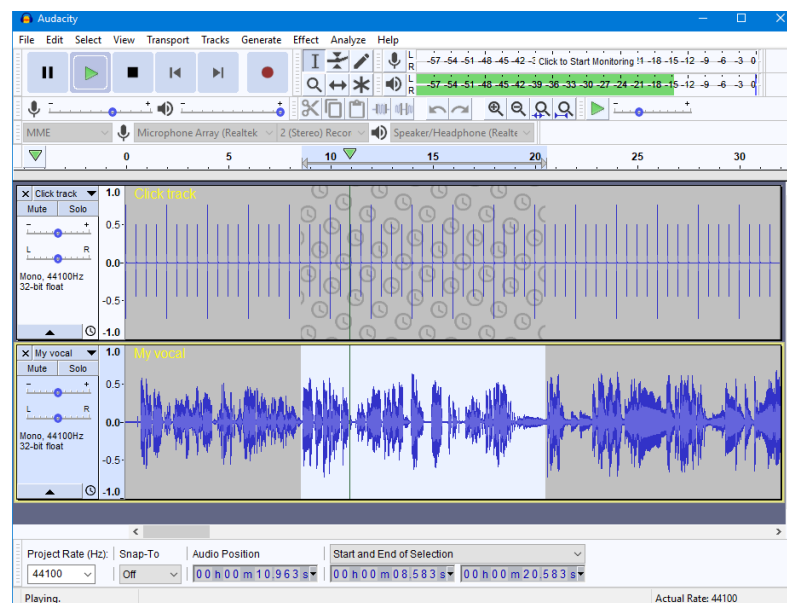


Ilustración 1.7-2 - Audacity, hecho con WxWidgets

Swing es una opción muy conocida en el mundo de la programación y se le conoce como el clásico. Es robusto y puede generar aplicaciones multiplataforma. La principal desventaja es que, el estilo de interfaz es muy antiguo y a día de hoy está muy anticuado. Es sólido, pero no es la mejor opción para este proyecto, ya que se busca algo familiar en el mundo Windows.

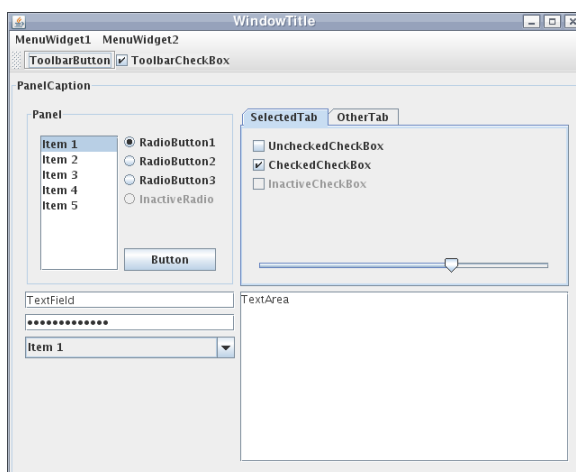


Ilustración 1.7-3 - Ejemplo de ventana con Java Swing

UWP es otra muy interesante opción, pues tiene un diseño muy moderno. Se programa con C# y puede crear interfaces bastante vistosas. El problema, es que sólo está limitado a las versiones de Microsoft Windows 10 y 11. Sería aceptable que el *software* fuera compatible como mínimo, con Windows 7. Además, pronto quedará completamente obsoleto con la aparición de WinUI.



Ilustración 1.7-4 - La calculadora de Windows 10 está hecha con UWP

WinForms es el *framework* de Microsoft para .NET que permite crear interfaces de forma nativa en *Windows* con poco código y con diseñador. Es fácil de programar y puedes sacar aplicaciones funcionales y familiares en muy poco tiempo. La desventaja es que es un *framework* bastante antiguo y el diseño puede ser considerado como algunos por anticuado y de la vieja escuela. Se ha elegido este *framework* al final por la familiaridad que poseo, la facilidad que es implementar un diseño y porque el resultado final sale bastante bien. Por último, Microsoft publicó *WinForms* como código libre.

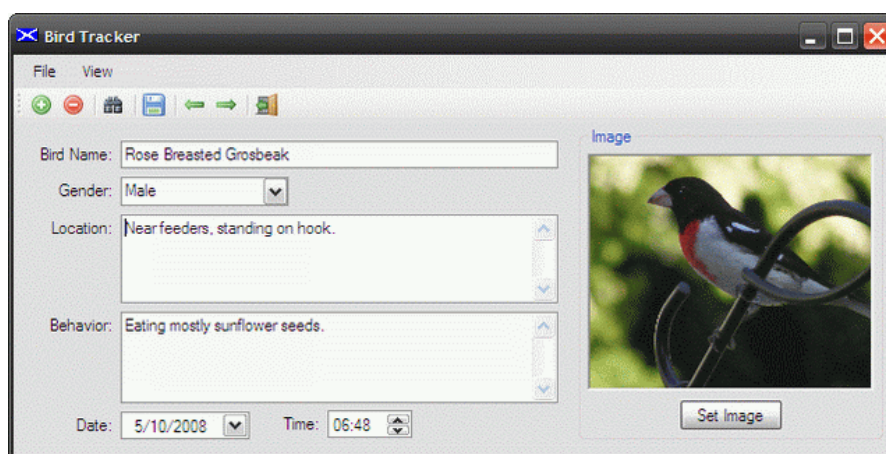


Ilustración 1.7-5 - Una aplicación hecha con WinForms sobre pájaros

1.8 Descripción de la solución propuesta

Al final, se propone una aplicación de escritorio para Windows, a priori sin conexión a Internet, que permita tener disponible una colección de álbumes con numerosos metadatos con canciones bonus y *con partes*, que se pueda diferenciar las numerosas ediciones de un distinto álbum sin demasiado esfuerzo.

La interfaz es sencilla, con máximo parecido a las aplicaciones nativas de Windows, para que sea fácil de utilizar por un usuario que no sea un experto. Contendrá múltiples idiomas por si el usuario no hablase español.

Se incluirá un reproductor integrado para reproducir de manera básica las canciones que tenga el usuario albergadas en su computadora. Ese mismo reproductor se reutilizará como *mando a distancia* para controlar su cuenta de Spotify.

Como ayuda al usuario, la aplicación es capaz de conectar con Spotify y descargar los metadatos completos de algún álbum que solicite el usuario en cualquier momento.

Por último, incluye una ventana de configuración para configurar la aplicación al gusto del usuario, con varias opciones a configurar como tipografías, colores, portapapeles... etc.

1.9 Tecnologías utilizadas

El lenguaje de programación elegido es C#, por la experiencia que tengo en el lenguaje y por su gran velocidad para desarrollar aplicaciones funcionales. Es bastante sencillo implementar aplicaciones y la programación asíncrona es bastante fácil comparada con otros lenguajes. Además, ofrece un gran rendimiento en las últimas versiones del lenguaje y contiene una gestión de memoria automática mediante recolectores de basura. Aparte, para programas de estilo multimedia funciona muy bien porque soporta imágenes, tipografías, caracteres Unicode sin instalar ninguna librería, todo viene de serie. Por último, instalar librerías es bastante fácil y existen un montón de código abierto listas para su uso.

1.9.1 Librerías

WinForms.

Es un conjunto de librerías de Microsoft para desarrollar aplicaciones para Windows con ventanas con C#. Es muy rápido diseñar e implementar una aplicación, Visual Studio tiene un diseñador integrador creado con ese propósito y ayuda mucho a la hora de diseñar los formularios y los eventos que la acompañan. Posee muchos componentes para crear interfaces intuitivas y familiares, ya que se asemejan mucho a los componentes que se ven y usan en Windows. Ejemplo de estos componentes son botones, listas, barras con iconos, menús, paneles... etc.

Newtonsoft.Json

Es una librería para C# de alto rendimiento para trabajar con JSON. Permite serializar y deserializar cualquier objeto. Funciona con LINQ y es bastante rápida y sencilla de utilizar. En la propia aplicación no se hace uso, pero la librería de Spotify sí depende de ella, ya que la API de Spotify trabaja con JSON y la librería de Spotify para C# la incluye automáticamente.

CsCore.

Es una librería de audio programada en C# gratuita y libre avanzada que ofrece un gran rendimiento y permite reproducción y captura de sonido sin mucho esfuerzo

de programación. Soporta los formatos más usados como MP3, FLAC, WAVE (audio PCM), WMA y OGG-Vorbis (mediante NVorbis).

TagLib-Sharp.

Es una librería de metadatos para C# sin dependencia de plataformas, es decir, funciona tanto en Windows, como en Linux como en Mac OS. Con ella se vuelve trivial el acceso a los metadatos de ficheros de audio, ya que a veces puede ser una tarea bastante compleja. Permite lectura, escritura y tiene un rango bastante amplio de formatos de audio y vídeo e imágenes.

SpotifyAPI-NET.

Esta librería de código abierto permite comunicación con la API Web de Spotify como cliente y ofrece clases y métodos para la autenticación, consultas a la base de datos de Spotify para conseguir metadatos de discos y carátulas y control sobre la reproducción del usuario. También, si se disponen de los permisos del usuario, se pueden editar las listas de reproducción del usuario.

NVorbis

Es una librería para decodificar ficheros Vorbis. No requiere P/Invoke⁵ o código inseguro. Ha sido perfecta para combinarla con la librería CsCore.

¿Qué es el código inseguro en C#?

Básicamente, es código que maneja punteros al estilo C/C++. En C# no es necesario usar punteros, pero hay ocasiones donde es necesario, sobre todo cuando trabajas con la API de Windows.

HtmlAgilityPack

Sirve para examinar documentos HTML y hacer “*scrapping*”⁶. Se usa a la hora de buscar letras de canciones por Internet.

1.9.2 Software

Visual Studio Community Edition

Un completo IDE extensible y gratuito para crear aplicaciones modernas para Windows, Android e, iOS, además de aplicaciones web y servicios en la nube.

Tiene numerosas funciones para desarrollar aplicaciones y soporta muchos lenguajes de programación como C++, C#, F#, Python, Visual Basic... etc. Ejemplos

⁵ P/Invoke es una tecnología que permite acceder a funciones y estructuras en otras librerías como la API de Win32.

⁶ Es una técnica para extraer información de una página web

de las funciones que son de gran utilidad son, por ejemplo, su depurador JIT que permite comprobar el estado de la aplicación justo en el momento de sufrir un cuelgue o una excepción no controlada.

También contiene medidores de rendimiento por si una parte de la aplicación tiene problemas graves de rendimiento (va muy lento o consume demasiada memoria), permite saber y determinar la línea de código donde más tiempo hace más procesamiento.

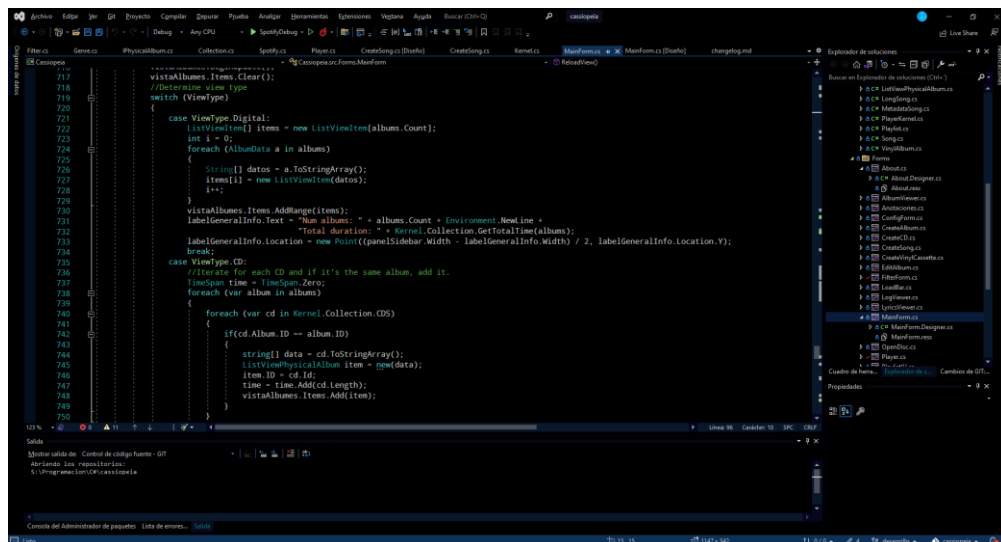


Ilustración 1.9-1 - Microsoft Visual Studio 22

Paint.net

Un editor de imágenes sencillo para Windows *freeware* que soporta edición de imágenes con capas. Usado para crear las imágenes del manual.

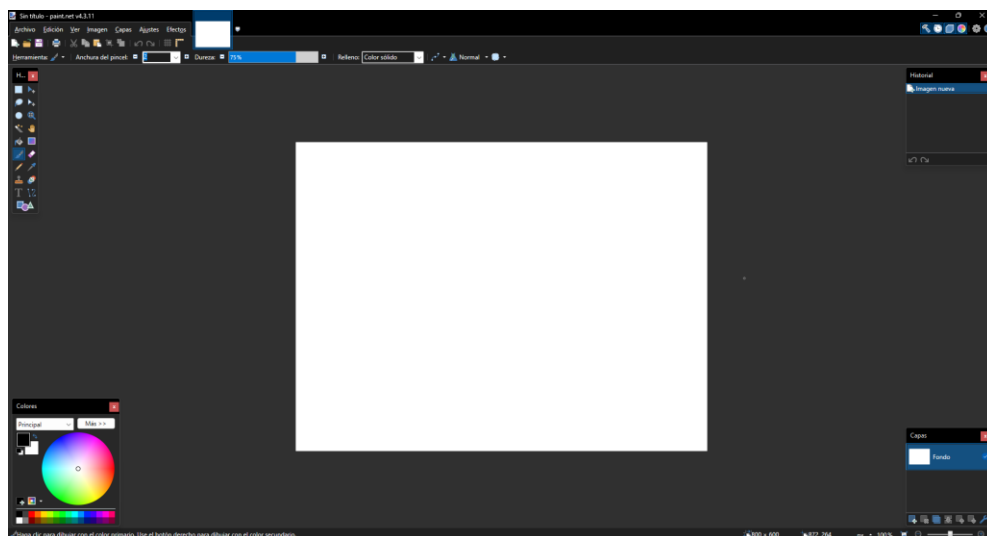


Ilustración 1.9-2 - Paint.net

1.10 Metodología de desarrollo de software.

Se ha decidido por usar una metodología de desarrollo incremental.

Es la más sencilla y consiste en descomponer las distintas etapas del desarrollo y una vez que estén todas finalizadas, hacer el incremento de la versión. Estas etapas son: análisis, diseño, implementación y pruebas. En cada incremento se repiten cada una de las etapas en orden.

Qué se hace en cada etapa:

Análisis.

Determinar los requisitos para el incremento y dejar claro el objetivo a diseñar e implementar. También, se piensa en la dificultad y en la viabilidad del propio incremento

Diseño.

Consiste en diseñar los componentes atendiendo a los requisitos planteados en la fase anterior. Antes de ello, primero hay que entender y dejar fijo el requisito para evitar hacer demasiados cambios en la etapa de diseño e implementación.

En esta parte, se junta el diseño de / los formularios implicados para resolver el problema planteado, los eventos a desarrollar y por último, pensar cómo implementarlo.

Implementación.

Consiste en implementar lo previamente diseñado y acordado a los requisitos. Se atenderán a los estándares del proyecto para evitar incongruencias en la base de código y ahorrar problemas en un futuro.

Pruebas.

Se realizan numerosas pruebas para determinar que el código implementado funcione a lo precisado en el requisito. Es la etapa más costosa, ya que a veces, algún error puede llevar demasiado tiempo de determinar y de resolver.

1.11 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFG, no existen restricciones de tipo económico, sino de tipo temporal (un número aproximado de horas). Por consiguiente, los cálculos de tamaño del proyecto están supeditados el tiempo disponible. En cuanto al esfuerzo, se dispone de tan un solo efectivo (la persona autora del trabajo).

Para la estimación, se ha recurrido al uso de COCOMO (Idri, 2000) publicado por Barry Boehm en 1981 en su libro “Software Engineering Economics” como herramienta para la obtención, de manera aproximada, del esfuerzo y del tamaño del proyecto. Más concretamente, se ha usado el modelo intermedio, ya que así obtenemos la estimación suficiente *grosso modo* para este proyecto.

Antes de calcular los datos, deberemos de especificar de qué tipo de proyecto tenemos. Hay tres modalidades:

- **Modo orgánico:** son proyectos desarrollados en un ambiente bastante conocido, el proyecto es pequeño y los programadores conocen la tecnología con la que trabajan y se tiene una idea bastante conservadora.
- **Modo semiacoplado:** son proyectos desarrollados con más complejidad. Es más grande y los programadores ya no conocen tanto las tecnologías con las que van a implementar el proyecto y requiere cierto tiempo a la hora de aprender e investigar.
- **Modo empotrado:** son proyectos muy complejos, grandes. Requieren mucha investigación, aprender las tecnologías y más personal.

Este proyecto entra dentro del modo orgánico, pues como se ha expuesto anteriormente, se han elegido las tecnologías por su sencillez y familiarización.

Modo de desarrollo	Esfuerzo nominal	Esfuerzo ajustado	Cronograma
Orgánico	$3.2 \times \text{KSLOC}^{1.05}$	$3.2 \times \text{EAFx}(\text{KSLOC})^{1.05}$	$\text{TDEV} = 2.5 \times (\text{PM})^{0.38}$
Semiaplado	$3.0 \times \text{KSLOC}^{1.12}$	$3.0 \times \text{EAFx}(\text{KSLOC})^{1.12}$	$\text{TDEV} = 2.5 \times (\text{PM})^{0.35}$
Empotrado	$2.8 \times \text{KSLOC}^{1.20}$	$2.8 \times \text{EAFx}(\text{KSLOC})^{1.20}$	$\text{TDEV} = 2.5 \times (\text{PM})^{0.32}$

Tabla 1 – Ecuaciones del modelo básico COCOMO

KSLOC se corresponde con la medida de miles de líneas de código y **EAF** es el multiplicador de esfuerzo. Nos ayudará a obtener un esfuerzo ajustado según las características del proyecto. Los siguientes coeficientes son:

- Complejidad del producto: nominal (1.0)
- Capacidad del programador: alto (0.86)
- Experiencia en el lenguaje: alto (0.95)
- Uso de herramientas software: muy alto (0.83)

- Fiabilidad del software: alta (1.15)

También deberemos estimar cuántas líneas de código habrá en el proyecto. Se ha puesto, así *grosso modo* que cada formulario puede tener 400 líneas de media y las clases auxiliares cerca de 150. Si debemos implementar 15 ventanas y otras 15 clases auxiliares, obtenemos 8250 líneas de código.

Sabiendo esto, el esfuerzo del proyecto es:

$$PM_{nominal} = 3.2 * 8.25^{1.05} = 29.34$$

Ese dato, indica la estimación de persona/mes. Para obtener el PM ajustado, hay que multiplicar por su EAF.

Por lo tanto:

$$PM_{ajustado} = PM_{nominal} = (1.0 * 0.86 * 0.95 * 0.83 * 1.15) = 22.88$$

Una vez obtenido el PM ajustado, pasaremos a obtener lo que nos interesa.

- Tiempo del proyecto:

$$TDEV = 2.5 * 22.88^{0.38} = 8.21$$

La duración aproximada, sería de 8 meses y 6 días.

- Número medio de personas:

$$\text{Número medio de personas} = \frac{22.88}{8.21} = 2.78 \text{ personas}$$

Por lo tanto, la media será de casi 3 personas.

1.12 Planificación temporal

A continuación, se presentará la planificación temporal del proyecto según la estimación del esfuerzo. Se ha considerado trabajar ocasionalmente en fin de semana.

Iteración	Empieza	Termina	Duración en días																												
				04/10/21	11/10/21	18/10/21	25/10/21	01/11/21	08/11/21	15/11/21	22/11/21	29/11/21	06/12/21	13/12/21	20/12/21	27/12/21	03/01/22	10/01/22	17/01/22	24/01/22	31/01/22	07/02/22	14/02/22	21/02/22	28/02/22	07/03/22	14/03/22	21/03/22	28/03/22	04/04/22	11/04/22
1	03/10/21	18/10/21	15	■																											
2	19/10/21	01/11/21	13		■																										
3	02/11/21	22/11/21	20			■																									
4	23/11/21	31/01/22	69				■																								
5	05/02/22	07/03/22	30																												
6	08/03/22	21/03/22	13																												
7	22/03/22	21/04/22	30																												

Ilustración 1.12-1 - Cronograma del proyecto

1.13 Presupuesto

El presupuesto engloba todos los costes relacionados al proyecto, tales como el ordenador elegido para desarrollar, las personas implicadas en el proyecto, licencias de software, iconos, música de prueba, cuentas en la nube, servicios como electricidad o internet, comida.

Se expondrá un coste estimado de un total de 30 000 €.

Presupuesto en hardware y software

El ordenador se ha comprado con una proyección a 3 años, entonces la amortización se calcula de la siguiente manera: $750 / (12 * 3) \Rightarrow 750 / 36 = 20,83$ € al mes durante tres años.

El precio de Spotify Premium es ajustado y no es posible cambiarlo, pues es la oferta que dispone la empresa. Ajustando, serían 10 € al mes. El resto de software usado será gratuito y de libre uso para ahorrar el máximo de costes posible.

Concepto	Coste total	Coste real / mes
<i>Computadora</i>	750 €	20,83 €
<i>Spotify Premium</i>	119,88 €	9,99 €
<i>Microsoft Windows 11 Pro</i>	10 €	0,28 €
<i>Microsoft Visual Studio Community</i>	0 €	0 €
Total	879,88 €	31,1 €

Tabla 2 - Costes en hardware y software

Coste individual.

Vamos a exponer el coste referido a salarios del personal involucrado en el proyecto. Dado que el proyecto ha sido realizado por una única persona realizando la labor de análisis e implementación, el sueldo base será el mínimo del convenio colectivo de analista programador. Serán 22 993,74 € brutos al año, ajustándose así al mínimo del convenio colectivo establecido en el BOE del 6 de marzo de 2018. La duración del proyecto aproximadamente será de un año.

Otros gastos relacionados con la electricidad, Internet, comida, entre otros, se relega al 10% del presupuesto total.

2 DISEÑO DE LA APLICACIÓN

Vamos a entrar en profundidad respecto al diseño del programa. Comentaremos los aspectos más importantes a tener en cuenta a la hora de diseñar las funcionalidades para cumplir con los requisitos que se presentaron.

2.1 Especificaciones del sistema

A la hora de determinar las especificaciones del sistema, deberemos definir los requisitos funcionales y no funcionales.

Requisitos funcionales. Son aquellos que puede y debe hacer el sistema.

- El programa debe permitir visualizar toda la información relacionada con un álbum en concreto, desde sus canciones hasta su letra.
- El programa debe gestionar correctamente las canciones con partes, que permitan su fusión y su defusión.
- El programa debe reproducir correctamente los ficheros de audio soportados que se encuentren almacenados en la computadora. También, deberá reproducir CD de audio.
- El programa debe poder soportar añadir distintos álbumes en formatos físicos, determinando su estado del medio y deben ser diferenciados con facilidad.
- El programa debe de conectar con una cuenta de Spotify y poder descargar todos los metadatos que fueran necesarios, para agilizar el proceso de añadido de álbumes.

Requisitos no funcionales. Son aquellos que describen la capacidad del sistema.

- El programa debe permitir tener una cantidad arbitraria de álbumes cargados, todos con sus debidos metadatos.
- El programa debe de ser robusto y gozar de cierta estabilidad al ser usado.
- El programa debe de ser extensible sin presentar complicaciones.

A la hora de gestionar las canciones bonus, se deberán diferenciar ya que originalmente, no se encuentran en el álbum original, se suelen hallar en reediciones.

Sobre las canciones largas, es una abstracción, ya que existen canciones de larga duración que se suelen cortar en aplicaciones de *streaming* o en su versión de CD para la comodidad del usuario. En cambio, es posible que el usuario quiera organizarlas como si fuera una sola canción conjunta y luego consultar las distintas partes. Un ejemplo de esto, es la canción Messiah Complex de Haken, que dura alrededor de 17 minutos y se encuentra dividida en 5 partes.

6	Messiah Complex i: Ivory Tower Haken	1,056,162	3:57
7	Messiah Complex ii: A Glutton for Punishment Haken	953,681	3:38
8	Messiah Complex iii: Marigold Haken	893,419	2:24
9	Messiah Complex iv: The Sect Haken	872,500	2:02
10	Messiah Complex v: Ectobius Rex Haken	901,310	4:57

Ilustración 2.1-1 - Una canción dividida en partes

A la hora de organizar este tipo de canciones, será elección del usuario elegir si dejarlas como están como si estuvieran en el medio original o *fusionarlas* en una canción sola y mostrar su duración completa. Es lo ideal relegar esto al usuario, para que así tenga la mayor flexibilidad a la hora de organizar su colección.

2.2 Análisis y diseño del sistema

2.2.1 Diseño del sistema

La aplicación consiste de formularios (que a partir de ahora llamaremos ventanas) y de clases internas auxiliares. Estas clases auxiliares son necesarias para representar la información. Las clases base son el núcleo duro y sobre ellas se ejecutan procedimientos que sostienen al programa.

Clases base.

- **Kernel.** Núcleo del programa. Contiene funciones para el cargado, guardado de datos, cambiar idioma, conseguir actualizaciones, ventana principal... etc.
- **Program:** Contiene el método principal y se limita a llamar al núcleo para lanzar el programa con los parámetros necesarios
- **Spotify:** Contiene todo el código para la integración con Spotify.

- **Config:** Representa la configuración del sistema, se encarga de guardarlo y cargarlo.

Clases auxiliares.

- **AlbumData:** Representa el concepto de álbum con todos sus metadatos.
- **Collection:** Representa una colección de discos.
- **Song:** Representa una canción con su duración
- **LongSong:** Representa una canción con partes. Hereda de canción.
- **PlayerKernel:** Contiene funciones internas para el reproductor para simplificar la reproducción de CD y de ficheros.
- **Playlist:** Representa una lista de reproducción con canciones locales
- **MetadataSong:** Representa un fichero de sonido como un MP3.

A priori, estas son las clases que componen el programa. A medida que se vaya avanzando en el desarrollo, se irán añadiendo más clases conforme sea necesario.

2.2.2 Diseño de las interfaces

El diseño de las interfaces se ha realizado conforme a los requisitos del sistema. Se ha necesitado, por lo menos, un formulario por cada requisito funcional del programa. Partimos de una ventana principal donde veremos todos los discos en una tabla y a partir de esta ventana, podremos llegar al resto de funcionalidades de la aplicación con los distintos menús y botones.

Atendiendo a los requisitos, tendremos que comprobar cuáles de ellos requieren algún formulario. A continuación, se exponen unos ejemplos:

- **Crear un álbum**

Para crear un álbum se necesitará de un asistente para que la tarea de crear una entrada sea lo menos tediosa posible, aunque haya que escribir todos los títulos necesarios.

- **Configuración del software**

Para que la tarea de personalización sea lo más sencilla posible, será necesaria una ventana con todos los parámetros personalizables.

- **Registro de letras de canciones**

Es necesario que para visualizar y editar las letras de las canciones se necesite una ventana aparte para que sea lo más cómodo posible. Este mismo, tendrá la posibilidad de aumentar o disminuir la letra.

- **Reproducción**

La reproducción de música requiere de un formulario aparte con sus botones para reproducir o pausar, saltar adelante o hacia atrás...

2.3 Diagramas de clases

Los diagramas de clases son representaciones UML de las clases que componen el proyecto y cómo se relacionan. El diagrama que engloba el proyecto es el siguiente:

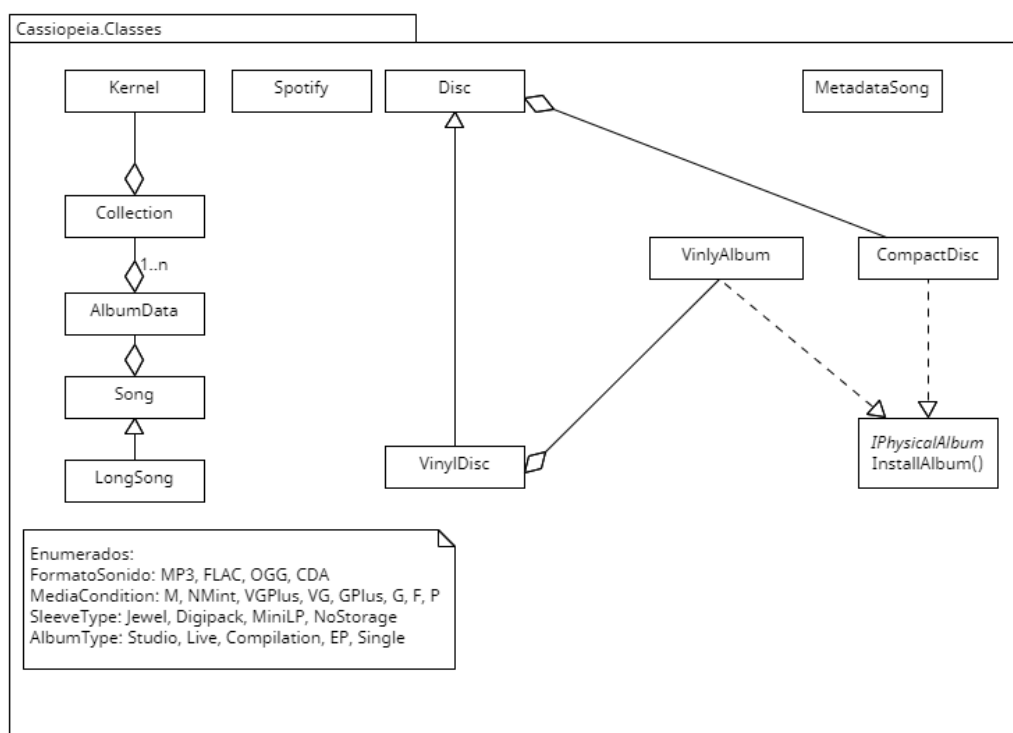


Ilustración 2.3-1 - Diagrama de clases del núcleo del software

2.3.1 Diagrama de casos de uso

Los diagramas de caso de uso muestran brevemente las interacciones que hay entre el usuario y la aplicación, viendo claramente los tipos de interacciones que se pueden hacer. En este proyecto, el diagrama de casos de uso ha sido el presente:

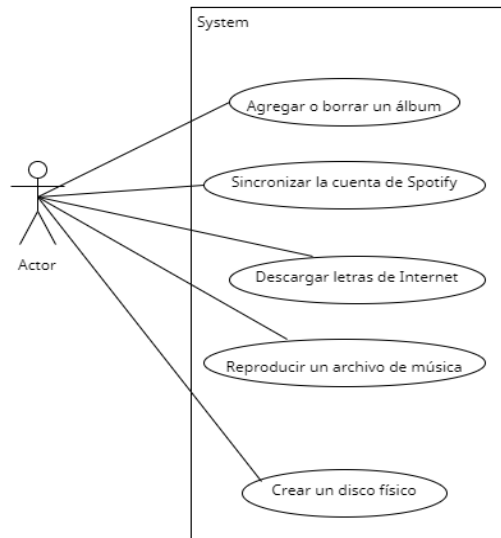


Ilustración 2.3-2 - Diagrama de casos de uso

2.3.2 Diagramas de secuencia

Estos diagramas muestran el proceso por el que pasa la aplicación desde que el usuario realiza una acción.

Añadir un álbum.

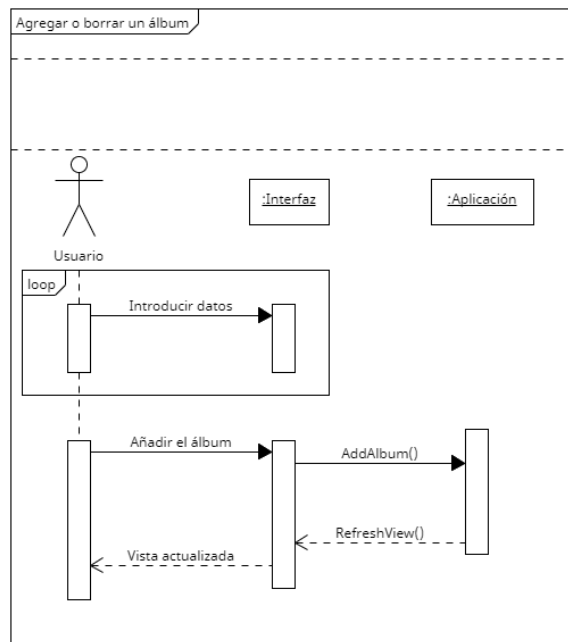


Ilustración 2.3-3 - Diagrama de secuencia para añadir un álbum

Vincular cuenta de Spotify

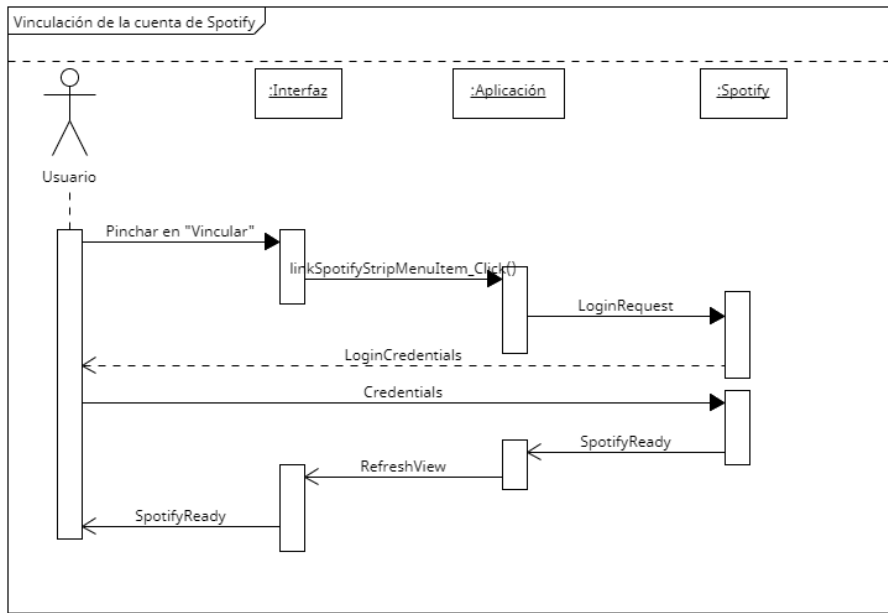


Ilustración 2.3-4 - Diagrama de secuencia para vincular una cuenta de Spotify

Descargar letras de internet.

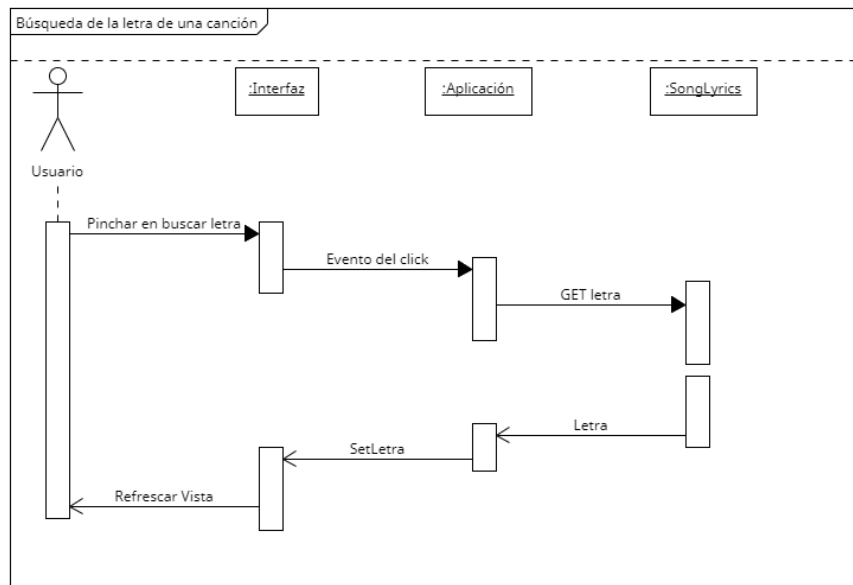


Ilustración 2.3-5 - Diagrama de secuencia para descargar letras de internet

Reproducir un archivo de música

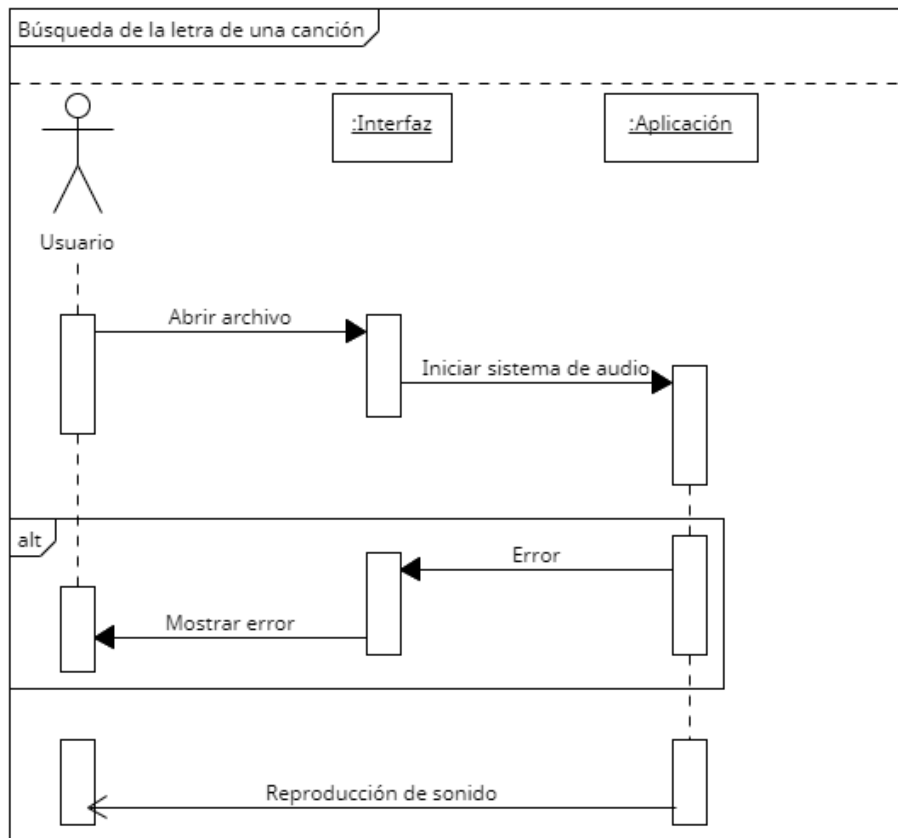


Ilustración 2.3-6 - Diagrama de secuencia para reproducir un archivo de sonido

Crear un disco físico

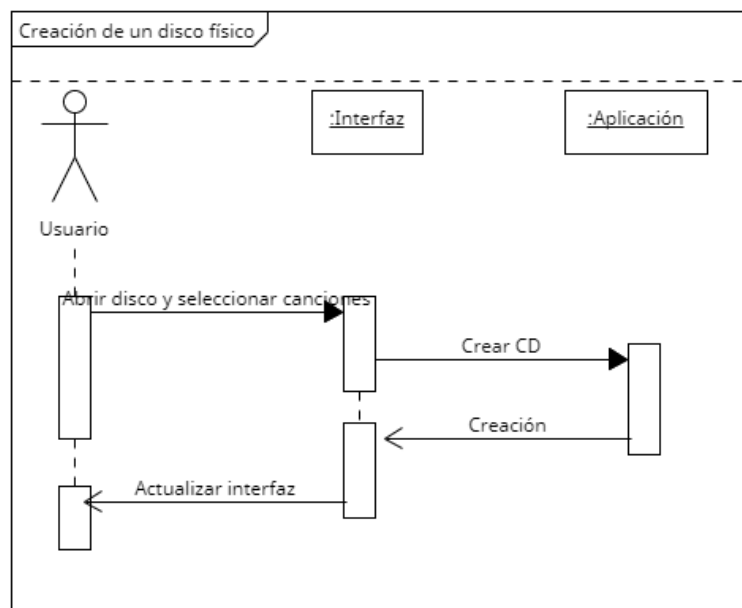


Ilustración 2.3-7 - Diagrama de creación de disco físico

2.4 Implementación del sistema

Una vez especificados los diagramas necesarios para todo el proyecto, vamos a entrar en profundidad en los aspectos de la implementación de las clases. Al ser el modelo de desarrollo incremental, se explicará en este apartado la base del proyecto.

2.4.1 Núcleo

El núcleo del programa consiste en una clase estática que agrupa todos los métodos necesarios para guardar y cargar los datos, inicializar todo el programa, levantarlo, el sistema de idiomas y el sistema de Spotify

También contiene el formulario principal que, a través de él, se accede al resto del programa.

- **Guardar y cargar los datos.**

En función de los datos a guardar, se usa un sistema u otro. Para los álbumes digitales, los metadatos, se usa CSV. Para los discos físicos se usa Json, para las letras se guarda la información de la canción y posteriormente la letra entera. El núcleo contiene todas las funciones necesarias para realizar esto.

- **Levantar el programa**

El núcleo debe encargarse de crear todo lo relacionado con el programa para poder lanzarlo correctamente. Todos los métodos que empiezan por Init cumplen este propósito. También se deben comprobar los argumentos de inicio.

- **Sistema de idiomas**

El sistema de idiomas funciona gracias al sistema de recursos interno que usa C#, estos archivos pueden contener cualquier dato, desde cadenas de texto hasta imágenes, sonido o datos binarios. El sistema de idiomas funciona de tal manera que disponemos de una cadena que funciona como llave y luego su traducción. En el programa, llamamos al núcleo pasando la llave y nos devuelve la cadena con el texto que necesitamos.

- **Spotify**

El sistema de Spotify se lanza de una manera muy concreta. Funciona de manera distinto porque es asíncrono. Se evaluaron varias opciones, antes

del lanzamiento del programa, después...

La mejor manera es de la siguiente manera: lanzarlo en el momento que cargue el formulario principal. ¿Cómo funciona esto?

```
//Load Spotify
Load += async (sender, args) => await Task.Run(() => Kernel.InitSpotify());
```

Lo que hacemos en esta línea de código es añadir un comportamiento al evento Load, que salta antes de que un formulario se cargue por primera vez. Se aprovecha para lanzar el sistema de Spotify ya que tardará un tiempo (tiene que mandar solicitudes vía Internet) y así evitar bloqueos en la interfaz.

- **Estructuras de datos**

Para almacenar los discos en memoria, la mejor estructura de datos es un Diccionario. Actúa como una tabla hash, donde se guarda una clave relacionada con un dato y no permite repeticiones. Es lo ideal ya que, sólo necesitamos una copia de un álbum y en ella todas las canciones que necesitemos. Si en algún momento se necesita otra copia, se puede repetir, pero con otro nombre. Además, es extremadamente eficiente, a la hora de conseguir un álbum a partir de una clave, siendo la eficiencia $O(1)$.

2.4.2 Clase Program

Esta clase contiene sólo el método principal y se dedica a lanzar la aplicación.

El proceso de cargado es el siguiente:

1. Se comprueban los parámetros de lanzamiento
2. Se carga el archivo de idiomas
3. Se carga y se aplica la configuración.
4. Se inicializa las estructuras de datos cargando los ficheros de datos y el formulario principal.
5. Se crea el reproductor
6. Se termina lanzando la aplicación con *Application.Run()*.

Dado a que la implementación está contenida en el núcleo, el método principal no contiene más de 100 líneas.

2.4.3 Configuración

El programa debe de ser capaz de guardar su estado para un próximo inicio para no volver a empezar de 0. Se ha creado una clase aparte, también estática, que sólo se limita a almacenar información para personalizar el programa.

Esta información es:

- El idioma
- Si está sincronizado con Spotify
- Portapapeles
- Tamaño de la ventana
- Colores
- Si está visible la barra lateral

Se vuelve a usar un archivo resx ya que, como se mencionó anteriormente, nos permite asociar una clave con un dato. Asociamos cada configuración con una cadena de texto y cada vez que se aplique un cambio en el programa se guarda fácilmente. Luego, en cada inicio, se revisan y se cargan todos los valores.

¿Y si el programa nunca ha sido lanzado?

No es un problema, pues se define una serie de valores por defecto para que el programa inicie correctamente, por ejemplo, el idioma se saca comprobando el idioma del sistema operativo. Si no se encuentra, se usa el inglés. O unos colores predeterminados. O los tipos de letra que se usan para visualizar los textos del sistema.

3 DESARROLLO

En este apartado se describirá con detalle el proceso de implementación. Tal como se ha especificado antes, la metodología a seguir será incremental. Se resumirá cada una de las iteraciones que ha habido en el proyecto, tales como el diseño del requisito y detalles sobre la implementación.

3.1 Primera Iteración

En esta primera iteración, se ha limitado a crear la mínima funcionalidad básica para crear álbumes, editar y borrar. Se ha creado el proyecto nuevo y se ha creado las primeras ventanas y los primeros diseños de la IU.

Los resultados de la primera iteración han sido una aplicación muy primitiva, donde se pueden añadir álbumes manualmente, visualizar, editar y borrar. También guardar toda la estructura a disco para evitar tener que escribir todo desde cero. Es crudo, pero funciona y se consigue las bases para el futuro de la aplicación.

Los objetivos para esta primera iteración son:

- Crear la interfaz principal
- Crear las ventanas necesarias para visualizar, añadir y editar álbumes
- Crear el sistema de guardado y el formato de guardado
- Realizar pruebas y ver que todo funciona correctamente

Diseño de las interfaces.

Para cumplir con las funcionalidades requeridas, se han necesitado crear varios formularios. El principal, para manejar la vista principal, creación de álbum, creación de canción, visualizador de álbum y editor de álbum.

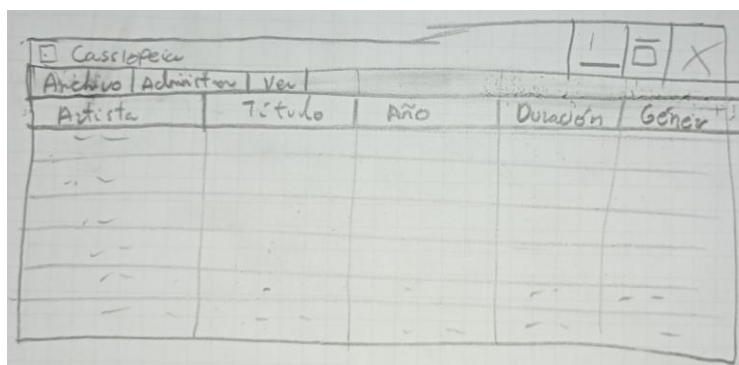


Ilustración 3.1-1 - Sketch del formulario principal

Para el algoritmo de guardado, se ha optado por el formato CSV, ya que es lo más sencillo de implementar.

¿Cómo se realiza el guardado?

1. Por cada álbum, se escribe en el fichero una línea con los parámetros del álbum. El orden es el siguiente:

Titulo,Artista,Año,NumCanciones,Genero,RutaAPortada

2. Para cada canción en ese álbum, se crea una nueva línea de la siguiente manera:

Titulo,DuracionEnSegundos

3. Se crea una línea en blanco para separar.

4. Vuelta al paso uno.

A continuación, se detallará un ejemplo de cómo quedaría:

```
Paracletus;Deathspell Omega;2010;10;blackmetal;covers\Paracletus_Deathspell  
Omega.jpg  
Epiklesis I;102  
Wings of Predation;221  
Abscission;363  
Dearth;225  
Phosphene;419  
Epiklesis II;184  
Malconfort;295  
Have You Beheld the Fevers ?;177  
Devouring Famine;306  
Apokatastasis Pantôn;244
```

A la hora de cargar, se hace el proceso inverso. Primero se crea el álbum con la información y a continuación, se crea cada canción y se añade. Se conoce siempre el número de canciones ya que está guardado en la primera línea.

Creación de álbumes manualmente.

Al ser uno de los requisitos principales, es lo primero que se tendría que implementar en esta iteración, la habilidad para añadir discos de manera manual. Para ello, se ha creado un asistente con dos formularios para que el usuario vaya metiendo los datos según haga falta.

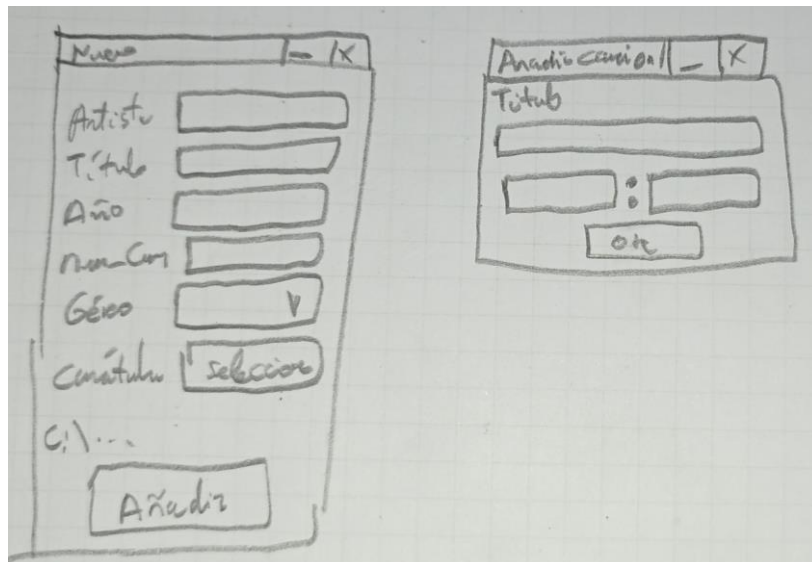


Ilustración 3.1-2 - Esbozo para los formularios de crear un álbum de manera manual.

Primero, se introducen los metadatos del álbum y se especifica el número de canciones. Luego, se abre el segundo formulario un número n de veces por cada canción que se tenga que introducir. Cuando finaliza, se añade. Si en algún momento se cierra la ventana, el proceso se dará por cancelado.

Visualización de un álbum.

En algún momento, tendrá que ser posible consultar la ficha de los álbumes que hayamos insertado en algún momento, es parte de la funcionalidad básica del programa. Se ha tenido que diseñar un formulario para cumplir con dicho requisito. Debe de mostrarse la información del álbum, nombre, título, duración, género, carátula y la lista de canciones.

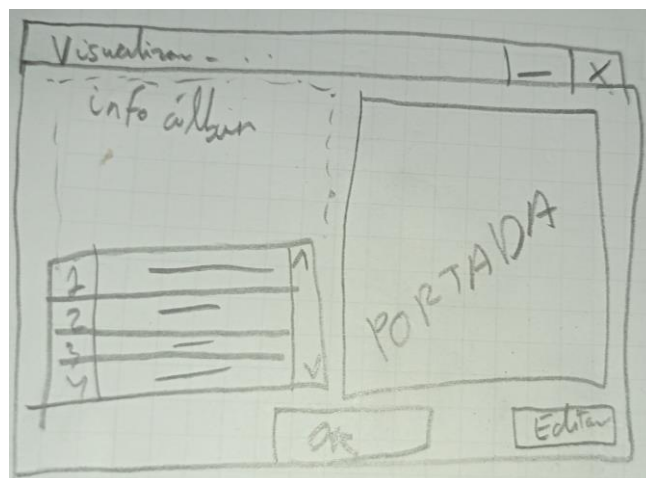


Ilustración 3.1-3 - Boceto para la ventana de visualización

Edición de los álbumes existentes.

Como en cualquier programa de gestión, se debe de poder editar cualquier cosa por si en algún momento se requiere una actualización o la corrección de alguna errata. Este formulario será parecido al de la creación, también incluirá una lista de las canciones existentes en el caso de que se requiera editar una canción. Se pensó de esta manera para que el usuario tenga familiaridad con el estilo del programa y que sea lo más sencillo de usar.

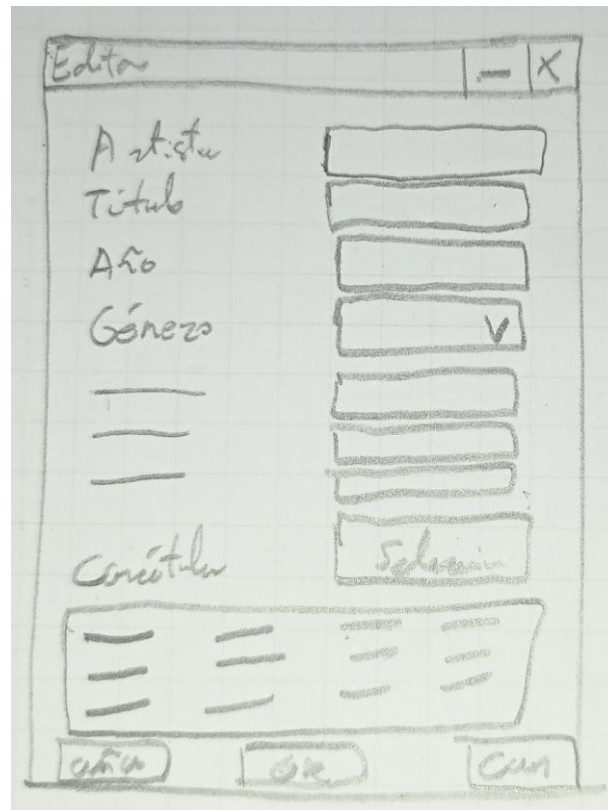


Ilustración 3.1-4 - Boceto del formulario para editar un álbum

Resultado de la iteración

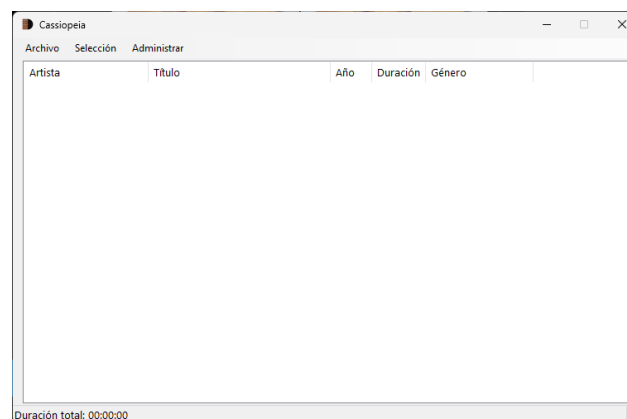


Ilustración 3.1-5 - Ventana principal vacía

Nuevo álbum

Artista

Título

Año

Número de canciones

Género

Tipo

Carátula

Ilustración 3.1-6 - Resultado del formulario de crear álbum

Añadir canción 1

Introduce el título de la canción

minutos:segundos

:

Es una canción bonus

Ilustración 3.1-7 - Resultado del formulario de crear canción

Visualizando Caligula's Horse - In Contact

Artista: Caligula's Horse
 Título: In Contact
 Año: 2017
 Duración: 01:07:04
 Género: -
 Tipo: Estudio

#	Título	Duración
1	Dream the Dead	08:10
2	Will's Song (Let the Colours Run)	04:43
3	The Hands Are the Hardest	04:26
4	Love Conquers All	02:21
5	Songs for No One	07:44
6	Capulet	03:24
7	Fill My Heart	06:42
8	Inertia and the Weapon of the Wall	02:57
9	The Cannon's Mouth	05:56

Duración total: 00:00:00

Ilustración 3.1-8 - Resultado de la implementación de la visualización

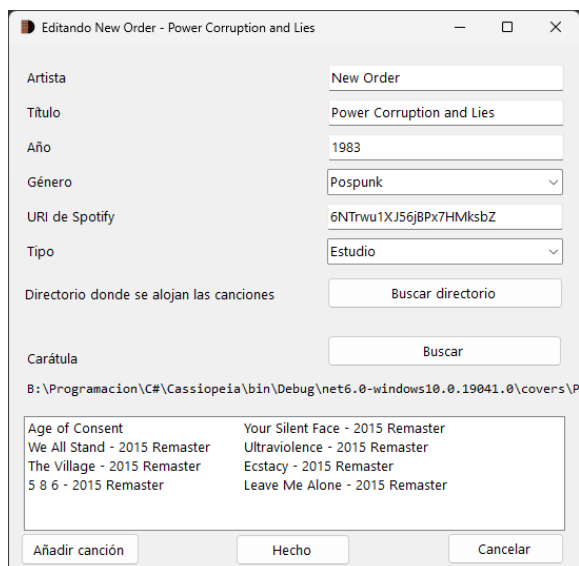


Ilustración 3.1-9 - Resultado de la implementación de editar álbum

3.1.1 Conjunto de pruebas

Prueba	Resultado
Añadir un álbum de manera manual usando el asistente.	Se añade correctamente y aparece en la vista principal.
Borrar un álbum.	No se realizaba correctamente el borrado ya que no se sincronizaba la vista con la E.D. Para solucionarlo, se ha tenido que reimplementar el método de borrado.
Visualizar un álbum.	Tras el doble clic, se abre la ficha correctamente.
Edición de un álbum, cambiando todos los metadatos.	Todos los cambios se aplican correctamente, incluida la duración.

Tabla 3 - Conjunto de pruebas 1

3.2 Segunda iteración

En esta iteración se agrega la primera integración con Spotify, que es el buscador de metadatos, simplificando el acto de añadir álbumes al sistema.

Fue necesario crear una clave en la página web de Spotify para tener acceso a las funciones API y la introducción de nuevas librerías en el proyecto para simplificar las tareas de comunicación.

Los objetivos para esta segunda iteración son:

- Programar el evento para copiar al portapapeles
- Implementar la búsqueda de álbumes mediante Spotify
- Arreglar errores
- Implementar la distinción de canciones bonus y canciones con partes.

Implementación de la conexión con Spotify.

El proceso para crear una aplicación en Spotify es muy sencillo, sólo es necesario generar un par de claves en la página <http://developer.spotify.com>. Una vez realizado esto, tenemos que generar un token para tener acceso a la API de Spotify y así obtener los permisos para descargar metadatos. Se realiza de manera fácil gracias a la librería que facilita las peticiones.

Se hace con tan sólo dos líneas de código:

```
SpotifyConfig = SpotifyClientConfig.CreateDefault().WithAuthenticator(new  
ClientCredentialsAuthenticator(PublicKey, PrivateKey));  
SpotifyClient = new SpotifyClient(SpotifyConfig);
```

Con la primera línea configuramos el inicio de sesión con los el par de claves que obtenemos de la página web y con la segunda montamos el cliente que hará las llamadas API correspondientes, tales como buscar álbumes y su información.

Para continuar, se ha tenido que diseñar un formulario para permitir al usuario insertar álbumes realizando una búsqueda en Spotify o directamente usando la URI del álbum⁷.

No se ha requerido un boceto, pues es un formulario muy sencillo, ya que sólo contiene dos entradas de texto, una para buscar y otra para poner la URI y sólo se activa con una entrada.

A la hora de realizar la búsqueda, se envía una petición a la API de Spotify y nos devuelve unos resultados, una estructura de datos. Para procesar esos resultados, los mostramos en una tabla, que es un formulario aparte, para que el usuario seleccione los álbumes que quiera agregar al sistema.

⁷ Una URI de Spotify tiene el formato `spotify:album:codigo`

Funcionamiento del portapapeles.

Se decidió implementar esto porque es bastante común copiar información de la aplicación para luego insertarla en otro lado. Se consigue capturando el evento cuando se pulsa una tecla. Se comprueba si se ha pulsado Ctrl y C y después, con la clase *Clipboard*, se invoca al método *SetText()* e insertar el texto que necesitamos.

Implementación de la distinción de canciones bonus.

Simplemente consiste en meter una variable booleana con valor predeterminado a falso en la clase *Song*. Si se cambia su valor a verdadero, aparecerá reflejada de otro color y esa duración *no* se sumará a la total del álbum.

A la hora de guardar, se añade en el CSV un 0 o un 1 para determinar el valor de esta variable.

Implementación de las canciones con partes.

Se ha necesitado una clase nueva, que herede de la clase *Song*. Esta clase, contiene una lista de canciones. Una canción con partes, en el fondo, es una canción con varias canciones. ¿Cómo se guarda? El procedimiento clásico cambia.

Primero se guarda el nombre de la canción y el número de partes que contiene. A continuación, por cada línea, se guarda el nombre de la parte y su duración en segundos.

```
Messiah Complex;5
Messiah Complex i: Ivory Tower;237
Messiah Complex ii: A Glutton for Punishment;219
Messiah Complex iii: Marigold;145
Messiah Complex iv: The Sect;122
Messiah Complex v: Ectobius Rex;298
```

¿Cómo crear una canción larga a partir de una lista de canciones? Para ello, se ha tenido que plantear un algoritmo. El objetivo es crear la canción larga conservando el orden. Pero, antes de nada, necesitaremos un número *n* de canciones seleccionadas, las que queremos *fusionar*.

1. Se crea la estructura vacía. Necesitaremos otra auxiliar para canciones que vayamos a borrar. También se guarda el número de pista de la primera canción seleccionada.
2. Para cada canción, se añade a la estructura y *se apunta para borrar*.
3. Para cada canción seleccionada, se borra.

4. Por último, se inserta la canción larga en el sitio que se guardó anteriormente. Dado que es un vector, se puede insertar por el medio.

Para deshacer esto, simplemente se agregan las partes a partir donde esté la canción larga y por último se borra la canción larga, dejando todo como estaba antes. En la interfaz, se ha optado por un color salmón para distinguirlo, ya que resalta.

Resultado de la iteración

Ilustración 3.2-1 - Resultado del formulario de búsqueda

#	Artista	Título	Año	Número...
1	Truckfighters	Gravity X	2005	13
2	Gravity XL	Sky Window	2021	11
3	Truckfighters	Gravity X	2005	13
4	Tom Wilson	Zero Gravity	2021	1
5	Zion.T	Zero Gravity	2015	1
6	DJ Xquizit	Gravity Save Me	2021	5
7	SKYL34D	Gravity X	2020	1
8	Xenia	Gravity	2021	1
9	Nocturnal Sunshine	Gravity (feat. RY X)	2020	2
10	xKore	Dead (feat. Gravity)	2016	1
11	DJ Xquizit	Dear Gravity	2017	6

Ilustración 3.2-2 - Formulario de resultados

Artista: Haken
 Título: Virus
 Año: 2020
 Duración: 00:52:00
 Género:
 Tipo: Estudio

#	Título	Duración
1	Prosthetic	05:58
2	Invasion	06:42
3	Carousel	10:30
4	The Strain	05:24
5	Canary Yellow	04:15
6	Messiah Complex I: Ivory Tower	17:01
7	Only Stars	02:10

Duración total: 00:00:00

Ilustración 3.2-3 - Visualizado de canciones largas

3.2.1 Conjunto de pruebas

Prueba	Resultado
Buscar un álbum en Spotify y agregarlo	Todo ha funcionado correctamente.
Crear una canción con partes	Todo ha funcionado correctamente.
Configurar canciones como bonus	Todo ha funcionado correctamente.
Guardado y cargado	Todo ha funcionado correctamente.

Tabla 4 - Conjunto de pruebas 2

Todo lo implementado en esta iteración ha funcionado a la primera sin encontrar fallos graves.

3.3 Tercera iteración

En esta iteración se ha diseñado el sistema de los discos físicos y en un primer paso se ha optado por implementar la creación de los CD. También se ha pensado por implementar el sistema de idiomas antes de que sea demasiado tarde ya que, hasta ahora, el programa no tiene muchas líneas de texto y conforme se vayan trabajando en las sucesivas iteraciones, el número de líneas puede crecer de manera bastante acelerada.

Los objetivos para esta tercera iteración son:

- Crear el sistema de idiomas con ayuda de los recursos de C#
- Implementar un mini asistente para crear CD basados en un álbum del sistema
- Crear una ventana de configuración del sistema para retocar los parámetros
- Realizar pruebas y arreglar errores

Implementación del sistema de idiomas.

C# dispone de un sistema de recursos que permite juntar una cadena de texto con otra. Esto permite crear un sistema de traducciones sencillo y es bastante rápido a la hora de cargar los textos. Con la clase **ResXResourceSet** y un archivo de recursos cargado, podemos pedir una cadena de texto y traducir el programa en el idioma se solicite. El proceso para cambiar de idioma es trivial, sólo hay que cargar el

fichero .resx con las cadenas nuevas y recargar los textos. No se pierde mucho tiempo de ejecución a la hora de aplicar los textos ya que no suelen haber más de 20 cadenas por ventana.

Un inconveniente de esto es, si un idioma le faltan traducciones, veremos cadenas vacías. No causa ningún problema al programa, pero da la sensación de que el programa tiene errores. La traducción al inglés estará siempre completa.

Creación del mini asistente para crear CD.

A la hora de registrar un CD en el sistema, se deberá de introducir más información para distinguir uno de otro. Se necesita un *mini asistente* para ayudar a la creación de estos, ya que las ediciones pueden variar por muchas cosas. Para crear un CD a partir de un álbum se deberá de especificar los siguientes datos:

- **Año de salida**
- **País de salida**
- **Número de canciones por disco**
- **Estado de la carátula**
- **Estado del medio**

Se hace la distinción por año de manufacturación y país de origen.

No debería existir problemas con las canciones, pues para eso ya se tienen los metadatos correctamente almacenados. Si el álbum dura más de 80 minutos, existe la opción de crear varios discos. También es posible seleccionar menos canciones de las que haya disponibles en el álbum original.

Es posible tener varias copias de un mismo CD en el sistema. Internamente se genera una ID aleatoria para diferenciarlas.

Creación de la ventana de configuración.

Fue necesario crear una ventana aparte para configurar los parámetros tales como el contenido del portapapeles, el idioma de la aplicación o incluso lo visual de la aplicación. A la hora de realizar el diseño, se ha optado por un diseño conservador basado en la aplicación foobar2000, debido a que es familiar para el usuario y queda bastante vistoso.

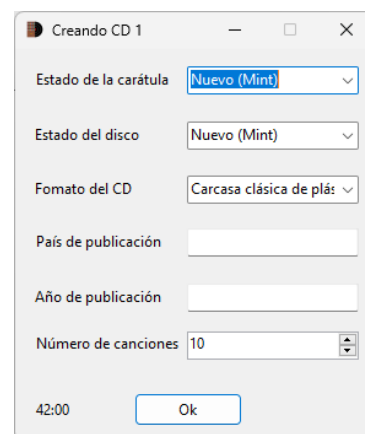


Ilustración 3.3-1 - Asistente

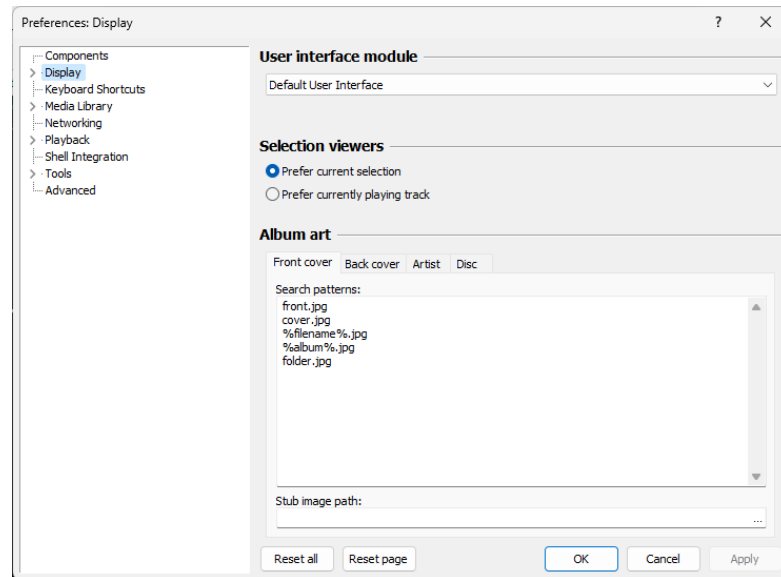


Ilustración 3.3-2 - Configuración de foobar2000

La idea final es crear una ventana similar a la mostrada en la figura anterior y poner todos los apartados configurables a la izquierda y que el contenido vaya cambiando según se seleccione la opción.

Los apartados de configuración que se han implementado son los siguientes: idioma, visualización (colores y tipografías), portapapeles y cadena de retransmisión.

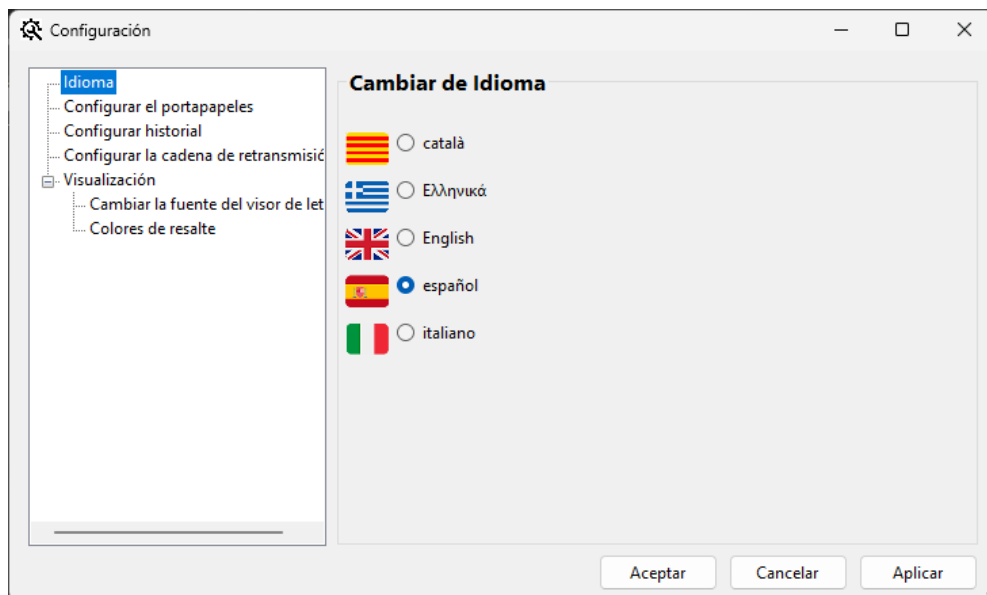


Ilustración 3.3-3 - Formulario de configuración para cambiar idioma

A la hora de guardar la configuración es inmediato, pues para ello se diseñó la clase **Config**. Cambiar un valor desde esta ventana automáticamente cambia los valores en esa clase estática y se guarda en el momento, para no perder ningún valor.

3.3.1 Conjunto de pruebas

Prueba	Resultado
Cambiar el idioma en la configuración	Los nuevos textos se aplican de manera inmediata.
Crear un CD	El CD se crea correctamente en el sistema
Crear un doble CD	Hubo algunos problemas a la hora de gestionar las canciones restantes, ya que hay que relanzar el mismo formulario. Se resolvió llevando la cuenta de las canciones añadidas.

Tabla 5 - Conjunto de pruebas 3

3.4 Cuarta iteración

Para esta iteración, se creó el reproductor como estaba estipulado en los requisitos. También, en mejorar la integración con Spotify, permitiendo conectar una cuenta de Spotify con el programa. Así mismo, el reproductor tiene una doble funcionalidad, reproducción de archivos locales y sincronización con Spotify.

También, se ha diseñado e implementado un sistema sencillo de listas de reproducción para los ficheros locales.

Los objetivos para esta cuarta iteración son los siguientes:

- Diseñar e implementar el reproductor
- Sincronizar una cuenta de Spotify con la aplicación
- Hacer que el reproductor “*tenga dos modos*”, local e Spotify.
- Diseñar el sistema de listas de reproducción.
- Realizar pruebas con el nuevo reproductor
- Arreglar todos los errores posibles que se vean y afecten a la funcionalidad

Reproductor, diseño.

El diseño del reproductor fue sencillo, pero luego a la hora de su implementación surgieron distintas complicaciones. Se optó, de nuevo, por un diseño conservador, con la portada y los botones de función básicos. Pero debía de cumplir una doble funcionalidad: reproducir archivos locales y luego sincronizarse con la reproducción de Spotify.

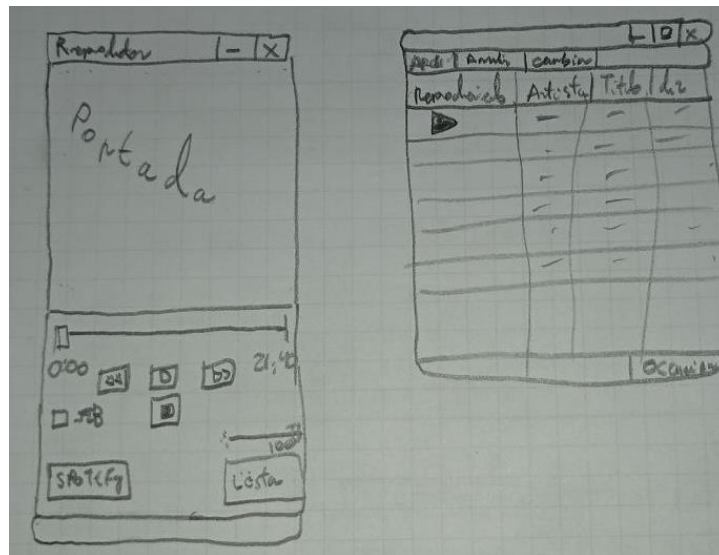


Ilustración 3.4-1 - Boceto del reproductor y la lista de reproducción

Primero, se implementó la funcionalidad de reproducción de archivos locales para dejarlo funcionando. Fue posible gracias a la librería **CSCore**, que permite reproducción de archivos de sonido de formatos famosos tales como MP3, FLAC o WAV. Más tarde, se implementó la reproducción OGG. Todo esto, se ha hecho en una clase aparte llamada *PlayerKernel* para simplificar el código.

Para implementar el reproductor, hay que pensar *en el tiempo*. Una canción al final es un flujo de audio que dura un tiempo determinado. Hay que actualizar cada un tiempo x la interfaz para refrescar los valores como la posición de reproducción o el volumen. El proceso para reproducir una canción es el siguiente:

1. Cargar el archivo de audio.
2. Creamos una “fuente de sonido”. Una fuente de sonido es un flujo de audio que proporciona *bytes*.
3. En una línea creamos el flujo de audio. Primero tenemos que saber el códec que requiere el fichero de audio y lo pasamos a flujo de audio.

```
_sound =
CSCore.Codecs.CodecFactory.Instance.GetCodec(cual).ToSampleSource().ToStereo().ToWaveSource(16);
```

4. Crear la salida de sonido. Esto en Windows, se suele hacer con WASAPI⁸.

⁸ Windows Audio Session API

Empezar a reproducir y a refrescar la interfaz, se consigue con un componente que se llama *Timer*. Con cada refresco llamamos al núcleo y actualizamos la posición de la canción.

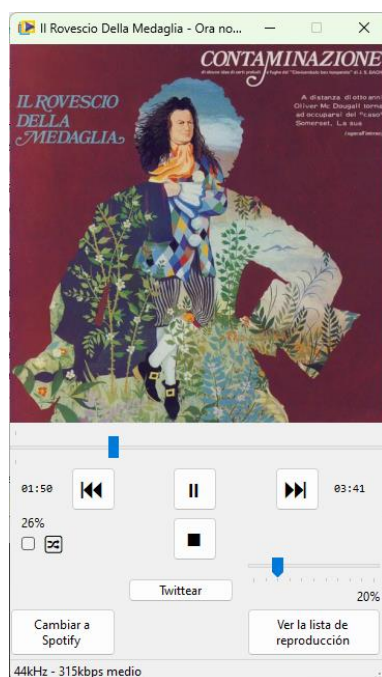


Ilustración 3.4-2 – Resultado final del reproductor

Implementación de la reproducción OGG.

CSCore no soporta de manera nativa el formato OGG, aunque sea libre. Para implementar su reproducción, deberemos crear una clase aparte que implemente la interfaz *ISampleSource*. Se necesitó una librería aparte, *NVorbis* que permite la lectura de ficheros OGG. A continuación, deberemos de implementar el comportamiento de las propiedades como la posición, duración, tasa de bits y los métodos de lectura. Una vez creada la clase, a la hora de cargar el fichero, usaremos esa misma clase para cargar el fichero y la reproducción funciona de manera normal.

Sincronización de una cuenta de Spotify.

Sincronizar una cuenta de Spotify con el programa fue un desafío ya que, requiere programación con Internet. Primero, se debe de conseguir el token con la autenticación del usuario. Se usa PKCE porque es la mejor opción para aplicaciones donde guardar el código secreto es difícil y también dispone de la ventaja de que el token se refresca automáticamente si se desea.

El proceso para iniciar sesión es el siguiente:

1. Antes de nada, comprobamos si tenemos un token existente en el sistema. Si existe el token, vamos al paso 3.
2. Si no existe el token, tenemos que generar uno nuevo. Creamos unos códigos de desafío PKCE, que es un mecanismo que funciona con OAuth 2.0. Seguidamente, generamos un enlace para iniciar sesión. La librería nos crea un mini servidor HTML, definimos los permisos de la aplicación y lanzamos el navegador para que el usuario inicie sesión. Cuando finaliza la autenticación, se ejecuta un *callback* que contiene el token enviado por Spotify y lanzamos el método para terminar de vincular la aplicación.
3. Conseguimos el nombre de usuario y el tipo de cuenta. Esto es importante para saber si se puede enviar órdenes de reproducción. Se guarda el token en un archivo aparte para así evitar el paso 2. También se indica que, cuando se refresque el token, lo se guarde de nuevo en el archivo.

```
//Leer token existente
var json = await File.ReadAllTextAsync(AuthPath);
//Pasar a objeto token
var token = JsonConvert.DeserializeObject<PKCETokenResponse>(json);
//Crear autenticador
var auth = new PKCEAuthenticator(PublicKey, token);
//Definir el evento para refrescar el token automáticamente
auth.TokenRefreshed += (sender, token) => File.WriteAllText(AuthPath,
JsonConvert.SerializeObject(token));
//Crear los objetos Spotify
SpotifyConfig =
SpotifyClientConfig.CreateDefault().WithAuthenticator(auth);
SpotifyClient = new SpotifyClient(SpotifyConfig);
```

Todo este código se ejecuta de manera asíncrona para no bloquear la interfaz mientras se realiza la conexión.

Implementación del “*doble modo*” del Reproductor.

Como se ha mencionado antes, se ha diseñado el reproductor con doble función, para escuchar canciones locales y como *mando a distancia* para controlar Spotify (si el usuario fuera Premium). Lo único a hay que tener en cuenta, es saber el modo en el que está la aplicación, pues a la hora de refrescar la ventana el procedimiento cambia. Cuando se refresca la ventana estando en el modo Spotify, se descarga el *PlayingContext*, que indica el contexto actual de Spotify de un usuario. Esto debe de realizarse de manera asíncrona, para evitar bloqueos en la interfaz de la aplicación. Si el contexto es válido, se actualiza toda la ventana, posición, volumen, aleatorio, portada... etc.

Esto se debe de hacer cada cierto tiempo no muy grande, con equilibrio entre uso de CPU, ya que no queremos saturar el sistema y de red (debemos vigilar la cantidad de datos que descargamos). Estos tiempos son de 150ms si la ventana es visible y 1000ms si la ventana no es visible. Se requiere una tasa de refresco medianamente alta para que no haya demasiada latencia a la hora de enviar comandos a Spotify tales como modificar el volumen, saltar parte de la canción o pausar y reproducir. También, se guardan dos valores *string* que corresponden a URI de Spotify, para saber si se ha cambiado de canción. En cada refresco se comprueba si la URI anterior y la URI actual han cambiado y si es así, se descarga la portada nueva y se actualiza la barra con el nuevo máximo. La descarga de la portada no es muy costosa, ya que se selecciona una versión con menor resolución y no suele pasar de los 50kb, es barato computacionalmente y en la carga de la red.

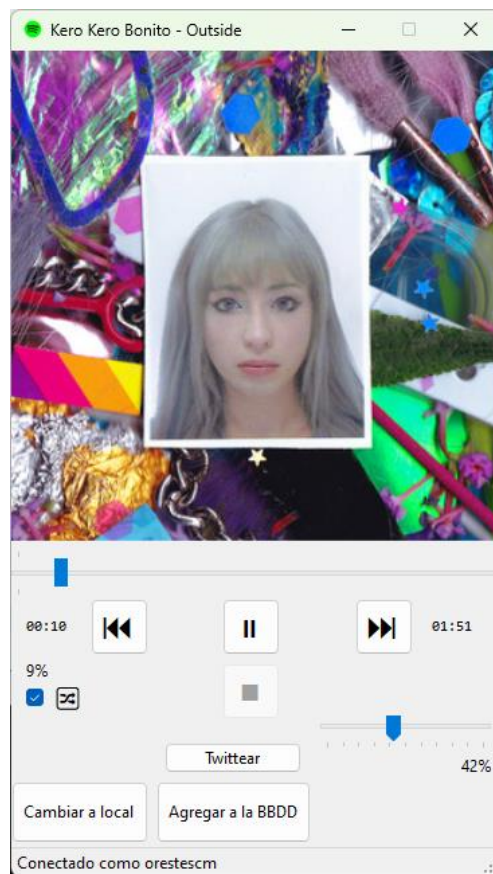


Ilustración 3.4-3 - Reproductor sincronizando con Spotify

Botón de compartir en Twitter.

Este botón se creó con la idea de que el usuario comparta por esta popular red social lo que está escuchando en ese momento. La idea consiste en pulsar el botón y

lanzar una nueva pestaña con el *twit* redactado para que el usuario pueda cambiar algo o directamente publicarlo en su perfil.

El procedimiento es el siguiente. Se recoge la información de la canción actual, que es conocida, y se crea la URL para crear el *twit*. El formato es el siguiente:

<https://twitter.com/intent/tweet?text=texto>.

Donde pone *texto* ponemos el fragmento seleccionado. Reemplazamos las nuevas líneas con los caracteres %0a%0c para que tengan efecto. También los espacios con %20. Una vez creada la URL, la lanzamos con la clase *Process*, que contiene un método para lanzar procesos. Hacemos una llamada del sistema a *cmd* con el comando *start*, más el parámetro /c y luego la URL. Windows detecta que es una página web y nos abre el navegador directamente.

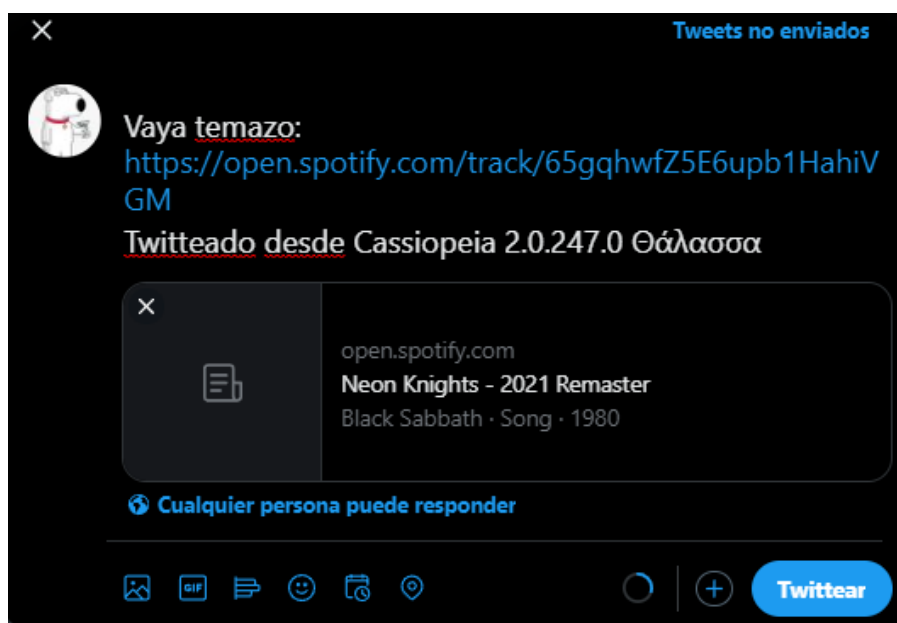


Ilustración 3.4-4 - Twit para compartir lo que se está escuchando, con opción a modificar el contenido.

Sistema de listas de reproducción

Se ha tenido que diseñar y crear otra ventana para almacenar la lista de reproducción para canciones locales. No es necesario crear una IU muy compleja, pues si muestra el número de canciones y cuáles son, es más que suficiente. Internamente contiene una lista de canciones y la interfaz sólo se limita a mostrarlas. Si se presiona doble clic sobre una entrada, se fuerza la reproducción. Está pensado para que sea lo más minimalista posible y que sea fácil de entender.

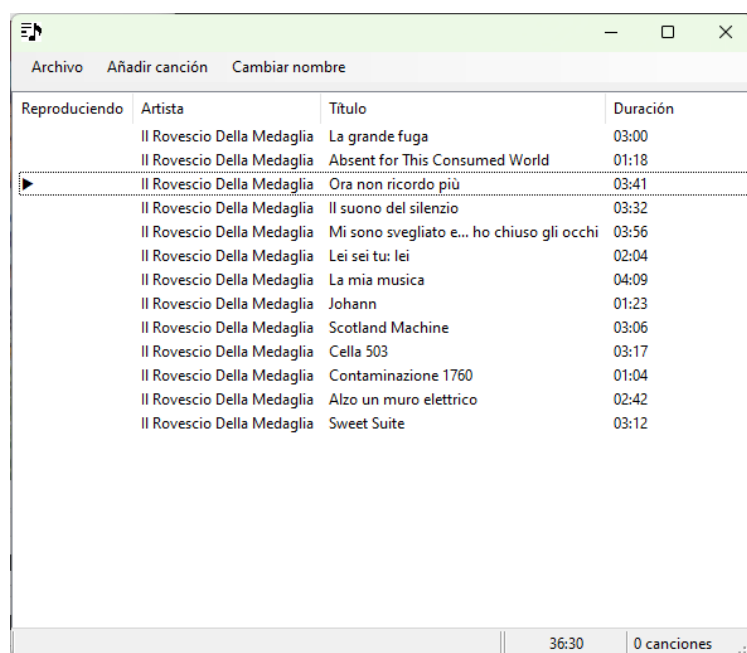


Ilustración 3.4-5 – Interfaz de la lista de reproducción.

3.4.1 Conjunto de pruebas

Prueba	Resultado
Reproducir un archivo local	Se reproduce sin problemas
Crear una lista de reproducción	Se crea la ventana, pero hay problemas a la hora de acceder a los índices. Se corrige comprobando siempre el puntero que recibe.
Sincronizar con Spotify	La sincronización va sin problemas y apenas tiene latencia, sin muchos problemas
Comprobar la estabilidad	El reproductor no goza de demasiada estabilidad, pues puede llegar a colgarse por un mal puntero de lista o por saltar dentro de la canción. Se realizaron mejoras y es un poco más estable.

Tabla 6 - Conjunto de pruebas 4

3.5 Quinta iteración

En esta iteración se ha planteado añadir algo menos complejo que en iteraciones anteriores, que fuera lo suficientemente sencillo y rápido.

Se optó por crear el visor de letras más una funcionalidad menor, añadir un álbum desde ficheros y como actualización, hacer que el reproductor reproduzca CD de audio.

Los objetivos para esta quinta iteración son:

- Implementar la reproducción de CD audio
- Diseñar la IU del visor de letras
- Hacer que se puedan buscar letras por internet
- Realizar pruebas
- Arreglar todos los errores posibles que se vean y afecten a la funcionalidad

Implementación de la reproducción de CD audio.

Para complementar el programa y pensando a la vez en el reproductor, se pensó en añadir una funcionalidad un poco *de la vieja escuela* y es reproducción de CD. La ventaja es que *CSCore* ya provee la salida de audio y el códec para audio PCM, que es igual que WAV. La desventaja es, hay que sacar el flujo de audio a mano y hay que usar las funciones de la API de Windows. Para ello, se creó una clase aparte para recopilar todas las funciones y estructuras necesarias, ya que la API de Windows está en C y el proyecto es en C#, entonces se han de importar las funciones a mano. También es necesario crear clases aparte para representar una pista de un CD, la propia disquetera (CDDrive) y la clase para ir leyendo los sectores y pasarlo a un flujo de audio (CDWave).

También se deberá de detectar las disqueteras en el sistema para saber si el usuario tiene una, ninguna o varias, para ello se ha diseñado otra ventana para representar las disqueteras disponibles. Fortuitamente, el propio *framework* nos proporciona una clase que representa una disquetera, *DriveInfo*, que proporciona información sobre una unidad de disco. Se obtiene la lista de todos los discos del sistema con *GetDrives()*, comprobamos si es de tipo CDROM y si fuera así, se añade a la lista. En el caso contrario, se cierra la ventana avisando al usuario, ya que imposible continuar con el proceso. Si existiera alguna, dejamos que el usuario seleccione alguna y que pulse "*Reproducir*". Ahí comienza el proceso.

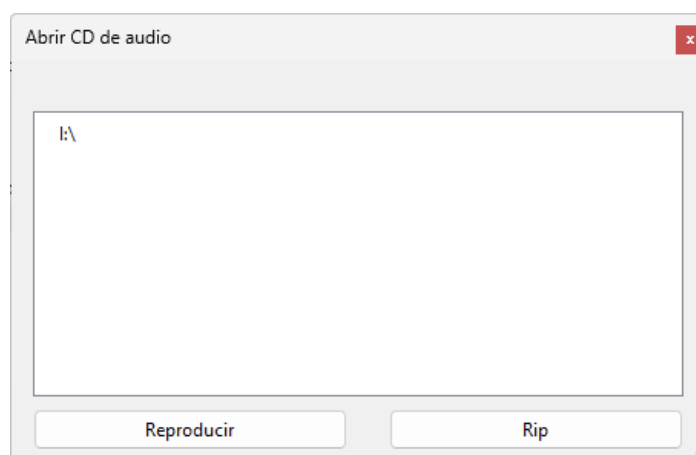


Ilustración 3.5-1 - Formulario de elegir disquetera

1. Con la letra del dispositivo, lo primero que se ha de hacer es abrir el dispositivo con la función *CreateFile*⁹ de la API de Windows. Es una función a más bajo nivel y nos permite conseguir un puntero (en inglés, HANDLE) al dispositivo. HANDLE es un void*. Con ese puntero, creamos el objeto disquetera.
2. Se leen las pistas del CD o el TOC¹⁰. El TOC es importante leerlo ya que un CD de audio no contiene archivos, es una secuencia de audio seguida. El TOC nos indica las pistas existentes en el CD y dónde empiezan físicamente en el disco.
3. Se crea la salida de audio y el flujo de sonido. El flujo de sonido se crea con ayuda de la clase auxiliar CDWaveProvider, que va leyendo el CD en bloques de sectores, los guarda como un buffer y lo reproduce. Se consigue con la función DeviceIoControl, que logra que Windows controle el dispositivo exactamente como le digamos.
4. Ya solo queda ordenar a la salida que reproduzca el sonido.

Diseño de la IU del visor de letras y la búsqueda por Internet.

Tras un diseño rápido se llegó a la conclusión de que no debe de ser muy difícil implementarlo. Contiene botones para ir iterando sobre las pistas del álbum y se deshabilita correctamente si nos encontramos en la primera o en la última.

⁹ Esta función crea un fichero o un dispositivo de entrada/salida

¹⁰ Table of Contents

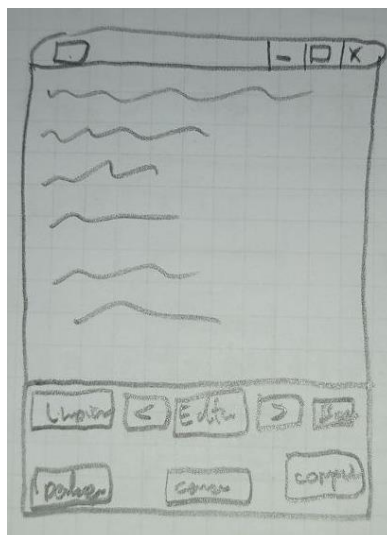


Ilustración 3.5-2 - Boceto del formulario de las letras

Dentro, existe un cuadro donde podemos editar la letra. Al principio se encuentra en modo sólo lectura, pero se puede editar en cualquier momento. También se incorpora la habilidad de ampliar el tamaño de la letra por si hubiera algún problema de visión.

Para terminar de complementar el requisito, se ha implementado la búsqueda por internet. Esto no fue del todo sencillo, ya que las páginas web tienen implementadas protecciones para evitar que se extraiga contenido de ellas, para evitar problemas de derechos de autor. Sin embargo, la página <http://www.songlyrics.com/> permite una extracción bastante sencilla de las letras. Se ha requerido una librería adicional para hacer la técnica denominada "*Html scrapping*", que consiste en extraer y buscar contenido de un documento HTML en función de sus etiquetas.

La dirección para buscar una canción sigue el formato "http://www.songlyrics.com/" + artista + "/" + título + "-lyrics" y es consistente. Se busca el *div* "*songLyricsDiv*" y lo insertamos en la letra.

En el caso de que nos devuelva "Sorry, we have no lyrics", directamente se deja en blanco y se envía un mensaje al usuario ya que no se han encontrado resultados.

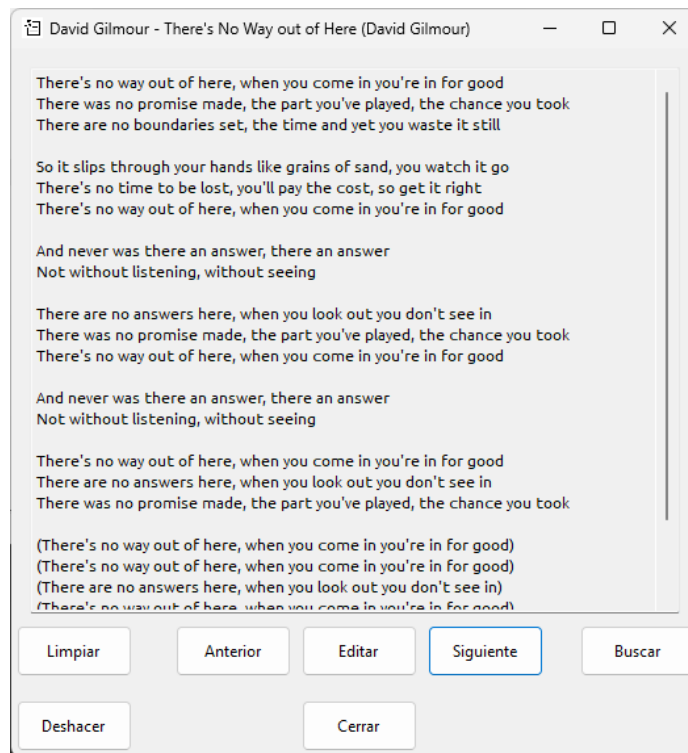


Ilustración 3.5-3 - Resultado del formulario de visor de letras

3.5.1 Conjunto de pruebas

Pruebas	Resultados
Reproducir un CD	La lista se crea bien y el CD empieza a reproducir sin problemas
Crear y editar una letra	La letra se guarda correctamente, y se puede editar. El zoom va correctamente.
Descargar una letra de internet	La letra se descarga y reporta si no se ha encontrado

Tabla 7 - Conjunto de pruebas 5

3.6 Sexta iteración

Para esta iteración se ha tenido en cuenta mejorar la usabilidad del programa y añadir funcionalidades para que sea más cómodo de usar. Se ha diseñado el sistema de filtros y la creación de una barra lateral para que sea más atractivo de ver. También, añadir la funcionalidad de representar discos de vinilo en el sistema con su correcta enumeración.

Los objetivos para esta sexta iteración son:

- Crear una barra lateral “*sidebar*” para que el programa sea más vistoso y atractivo para el usuario.
- Crear una barra de herramientas y colocar iconos de acciones frecuentes para simplificar el programa.
- Crear un sistema de filtros para filtrar las distintas vistas del programa y así facilitar la búsqueda de un artículo dentro del sistema.
- Implementar el añadido de discos de vinilo con su asistente.
- Realizar pruebas
- Arreglar errores y problemas que se encuentren por el camino

Diseño e implementación de la barra lateral.

Esta barra lateral debe de mostrar información básica sobre el álbum seleccionado. Su portada (si tuviera), título, artista y su duración. También se puede incluir un resumen sobre la colección como el número de álbumes totales y su duración total. El objetivo es aumentar la vistosidad del programa y que sea más atractivo, en lugar de que sólo se vea la tabla.

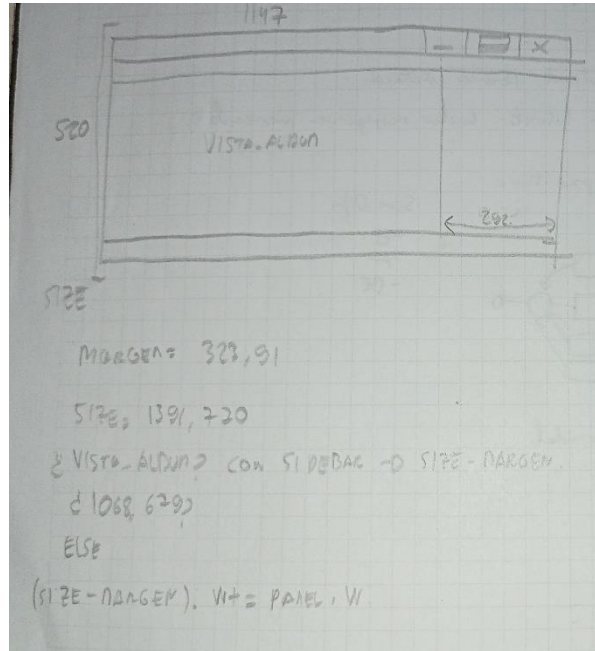


Ilustración 3.6-1 - Esbozo de la barra lateral

Para su implementación, se ha usado un Panel. Dentro de ese Panel, se pueden poner varios componentes como etiquetas para mostrar texto y un *PictureBox* para mostrar imágenes. Todos esos controles se agrupan dentro del Panel para su fácil manejo a la hora de ocultar y mostrar. ¿Cuándo se debe de actualizar? En el momento que cambie la selección. Si se seleccionan varios álbumes, sólo se muestra el primero que se ha seleccionado.

A la hora de mostrar u ocultar la barra lateral, se ha de cambiar el tamaño de la ventana principal.



Ilustración 3.6-2 - Barra lateral

- Si se oculta, se comprueba primero si está maximizada la ventana, si lo está, se suma el ancho de la ventana con el ancho de la barra, en caso contrario, restamos el tamaño menos un umbral determinado.
- Si se muestra, se aumenta el tamaño de ventana para que quepa la barra, excepto si está maximizada, que se reduce el tamaño de la ventana.

Implementación de la barra de herramientas.

En la mayoría de programas de *Microsoft Windows*, existe una barra de herramientas con numerosos y vistosos iconos que permiten realizar acciones comunes sin necesidad de acceder por los numerosos menús. Aunque, hoy en día, con las aplicaciones modernas, es un estilo que se está perdiendo. Un ejemplo de barra de herramientas es el siguiente.

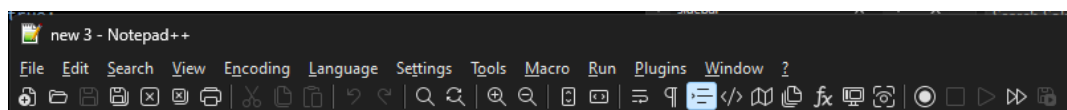


Ilustración 3.6-3 - Barra de herramientas de Notepad++

Primero, se ha tenido que añadir una *ToolStrip*. Es una barra que permite añadir iconos con acciones y cajas de texto. Primero se añaden los iconos que se quedan fijos. Estos son acciones que no dependen de ningún elemento seleccionado. Estas acciones son:

- Crear un nuevo álbum manualmente
- Crear una base de datos vacía

- Abrir una base de datos existente
- Guardar

Cuando se seleccione un álbum, se añaden iconos automáticamente que representan acciones como:

- Ver la ficha
- Editar los metadatos
- Borrar
- Ver la letra
- Crear un CD
- Crear un vinilo

Si seleccionamos varios, se borrarán todos los botones excepto el de borrar.



Ilustración 3.6-4 - La barra de herramientas implementada

Sistema de filtros.

Cuando una colección tiene cientos de álbumes se vuelve impracticable buscar los discos uno a uno. Implementar una función de búsqueda sería lo ideal. C# tiene disponible una tecnología llamada LINQ que permite hacer filtrados y búsquedas sobre una estructura de datos en tiempos bastante rápidos. Una colección se puede filtrar de varias maneras. Por artista, por título de álbum, por nombres de canciones... etc. A la hora de implementar el sistema de filtrado, se ha creado una ventana auxiliar muy sencilla para que el usuario escriba su filtro personalizado y lo aplique.

También, se ha aprovechado la nueva barra de herramientas y se ha puesto una caja de texto para que se aplique un filtro "global", que busque todo con la entrada que se ponga.

A la hora de aplicar los filtros, primero se aplica el del título de la canción. Primero se crea una estructura de datos auxiliar, luego, por cada álbum y luego por cada canción. Si la cadena de búsqueda está contenida, se añade a la lista y se pasa

al siguiente. Es eficiente, aunque haya muchos discos porque normalmente un álbum no contiene más de 15 canciones.

LINQ (Language-Integrated Query) es un conjunto de extensiones de C# que permiten trabajar con secuencias de datos, por ejemplo, vectores, listas enlazadas, diccionarios y luego filtrar, ordenar, agrupar... etcétera. Tradicionalmente, las consultas se realizaban con una cadena de texto que no se comprobaba en tiempo de compilación, dando lugar a numerosos errores de ejecución. Además, se tenía que aprender un lenguaje por cada tipo de origen, ya sea SQL, XML, entre otros. Con LINQ escribes la consulta con objetos fuertemente tipados usando funciones y operadores que ya resultan familiares. Esto simplifica mucho el proceso, ya que se trabaja con objetos y con la estructura de datos en memoria.

LINQ usa ejecución diferida. Es importante entender que llamar a estos métodos de extensión no ejecuta la consulta y muestra los resultados. Estos métodos más bien, presentan la pregunta y no la respuesta. Esto mejora el rendimiento debido a que no se ejecutan consultas continuamente y sólo se ejecutan cuando son estrictamente necesarias. Otra ventaja de esta ejecución diferida es que siempre devuelve los últimos datos, es decir, que si creas una consulta y la dejas en memoria y luego se actualizan los datos y vuelves a ejecutar la consulta, siempre estará actualizada.

Se crea una *query* para seleccionar los álbumes que contengan el artista que buscamos. Query es la estructura de datos donde realizamos el filtrado, que en este caso son los álbumes.

```
query = from album in query where  
album.Artist.ToLower().Contains(filter.Artist) select album;
```

Esto selecciona los álbumes y se elige cuando el artista contenga la palabra que buscamos. Similar a algo que haríamos en SQL.

Una vez se filtra la vista, la cargamos en la vista principal con un proceso idéntico al que hacíamos originalmente, pero haciendo saber al programa que estamos trabajando con una vista filtrada.

Ilustración 3.6-5 - Resultado del formulario de filtros

Por último, como se ha mencionado anteriormente, se aprovechó la nueva barra de herramientas para añadir un pequeño cuadro de texto para que el usuario haga un filtrado general de la colección. También se ha añadido un botón a la barra de herramientas para su fácil acceso.

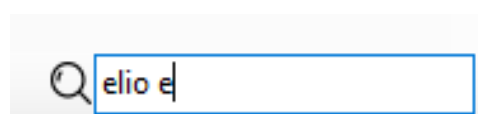


Ilustración 3.6-6 - Filtrado en la barra de herramientas

Creación de discos de vinilo, asistente.

Este proceso es similar al de los discos compactos, pero con un par de diferencias. Como ya se sabe, un disco de vinilo tiene una peculiaridad, y es que contiene música en las dos caras, por ende, un disco puede contener un número distinto de canciones en un lado que en otro.

También las duraciones son bastantes menores comparadas a los CD. Varía entre 30 y 70 minutos¹¹ entre las dos caras. El diseño del formulario se recicla del anterior, pero con un cambio notable y es el número de canciones en la cara frontal (A) y la trasera (B). También se debe de tener en cuenta que no se exceda sobre el total de las canciones del álbum. Si faltan canciones del álbum por poner en el disco, el programa lanzará una pregunta

Ilustración 3.6-7 - Formulario para crear discos de vinilo

¹¹ No es normal que un disco alcance los 70 minutos, pero se conocen ciertas excepciones. Pasa también en la música clásica.

si se desea crear otro disco. Esto se repetirá hasta que no queden canciones.

A la hora de visualizar, se debe de cambiar la creación de la tabla de visualización. En vez de mostrar los números de canciones de forma secuencial, se debe poner la cara y su número de pista relativo. Para conseguir esto, se ha hecho lo siguiente.

1. Para cada disco de vinilo:
2. Creamos un grupo con la cara frontal, empezando por la A. Se crea un número auxiliar que indica el número de canción relativa y otra variable de tipo carácter para el lado actual, inicializada a. Ahora, para cada canción, se carga en la tabla pero con la numeración A1, A2, A3...
3. Toca cambiar a la cara trasera. Se reinicia la variable auxiliar anterior a 1 y se aumenta el lado, en este caso la B. Se repite el proceso.
4. Si hubiera más discos, aumentamos el lado, pasando al C, E y los que fueran necesarios y volvemos al paso 1.

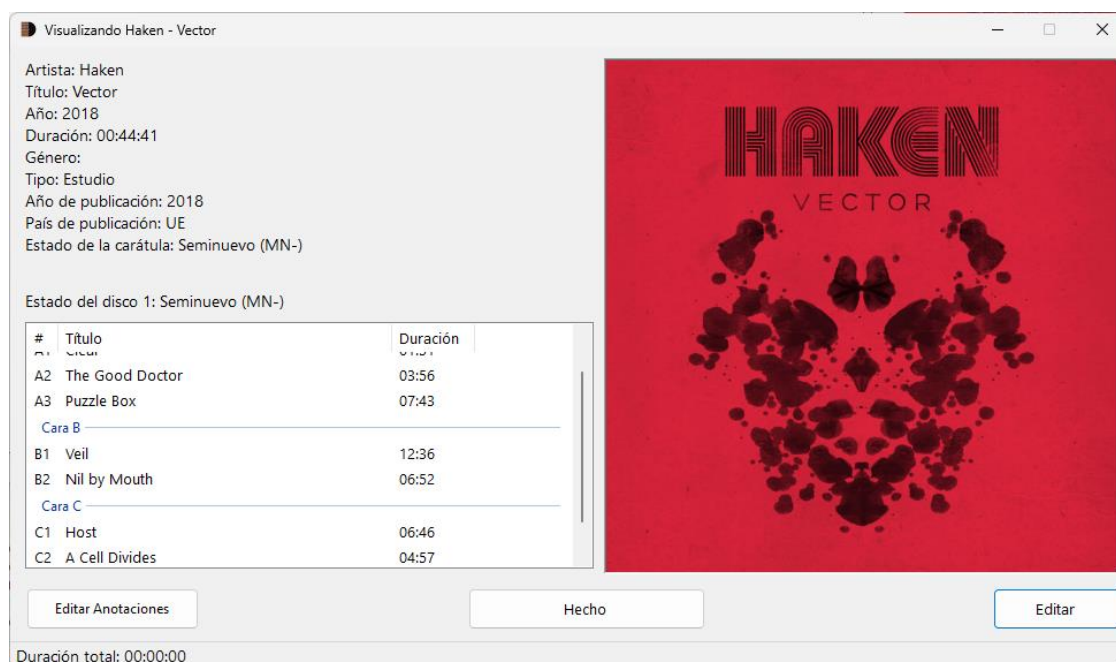


Ilustración 3.6-8 - Resultado de añadir un álbum de vinilo con dos discos.

Importación de una colección de Spotify.

Esta funcionalidad permite importar toda la colección de álbumes de un usuario de Spotify, en este caso, de la cuenta que se encuentre vinculada. El objetivo de esto

es aumentar la velocidad de la creación de la base de datos. El proceso para implementar esta funcionalidad es el siguiente:

1. Creamos una estructura de datos auxiliar para ir insertando los álbumes desde Spotify.
2. Obtenemos la lista de álbumes de la biblioteca del usuario
3. Si contiene más de 100 entradas, avisamos al usuario de que podría tardar un tiempo.
4. Vamos descargando los discos en bloques de 20 ya que Spotify no nos permite descargarlos todos de una vez. Se consigue enviando una solicitud de librería, para ir pidiendo el bloque que se necesite en ese momento.

Con esta funcionalidad, se pueden descargar alrededor de 1000 álbumes en poco más de unos 5 minutos.

3.6.1 Conjunto de pruebas

Prueba	Resultado
Crear un vinilo doble (con dos discos)	El disco es creado correctamente, pero a la hora de visualizar había un fallo cuando reportaba el número de canción. Se arregló.
Seleccionar discos en la vista para ver la barra lateral	Al principio funcionaba bien pero un inconveniente era que acumulaba mucha memoria por la foto. Se arregla forzando la llamada al recolector de basura.
Probar distintos filtros	Los filtros se aplican correctamente y muy rápido.
Pulsar los botones de la barra de herramientas	Las acciones se llaman correctamente.
Importar una colección de Spotify	Todos los álbumes se insertan correctamente con todos sus metadatos y portadas.

Tabla 8 - Conjunto de pruebas 6

3.7 Séptima iteración

Para la última iteración, se ha implementado el sistema de gestión de cintas de casete, muy típicas durante los años 80 y 90 cuando el CD no estaba muy

popularizado en España. También, se han implementado mejoras en distintos formularios para dejarlo todo pulido.

Los objetivos para esta última iteración son:

- Implementar el sistema de cintas de casete
- Mejorar la usabilidad del programa añadiendo mejoras de calidad, como opciones para compartir y añadir un álbum a partir de ficheros de sonido.
- Realizar pruebas
- Arreglar errores y problemas que se encuentren por el camino

Cintas de casete

El formulario de la creación de discos de vinilos se creó con vistas a ser reusado para la creación de las cintas. Esto es, debido a que, por muy distintos que sean los formatos, comparten lo más importante que es el lado frontal y trasero. Luego, se puede reutilizar la ventana anterior con unas ligeras modificaciones para crear cintas.

La clase para representar la propia cinta también implementa la interfaz de disco físico, con la diferencia de que se indica cuántas canciones tiene el lado frontal y el lado trasero. Normalmente, es raro encontrar un casete doble, ya que soportan más duración que los propios discos de vinilo. En este caso, si faltan canciones por asignar, se termina el proceso.

Mejora de la usabilidad – Compartir letras en Twitter

Para mejorar el formulario de visualización de letras se barajó la posibilidad de compartir las letras mediante un *twit* en la red social Twitter. Para ello, se recoge el texto seleccionado y se crea la URL para crear el *twit*. El proceso para lanzar Twitter es idéntico a cuando se comparte la canción que se está escuchando mediante el Reproductor (o se está sincronizando con Spotify).

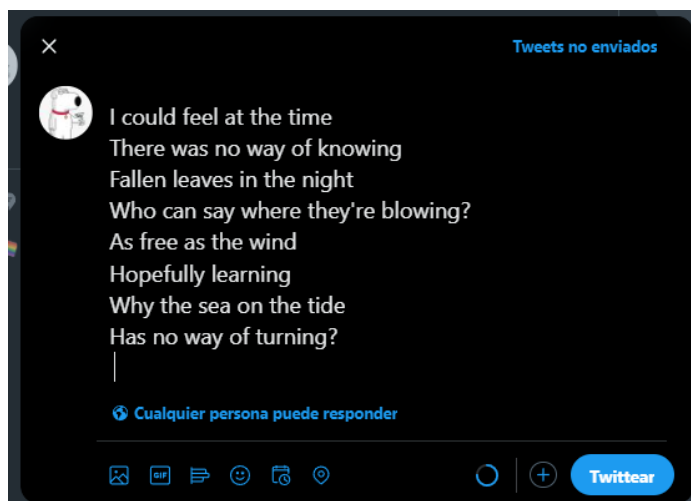


Ilustración 3.7-1 - Resultado de hacer clic en el botón de compartir un fragmento de una letra

Mejora de la usabilidad – Creación de un álbum a partir de ficheros de sonido.

Con mucha probabilidad, el usuario tendrá música almacenada en su computadora. En vez de añadir el álbum de manera manual con el asistente, puede ser que prefiera indicar los ficheros existentes con metadatos y que se añada el álbum al sistema de manera automática. Para ello, se ha implementado una función que cumpla con este requisito que mejora la usabilidad del programa de manera notable.

Para ello, se ha tenido que plantear un algoritmo que escanee los ficheros de sonido. El requisito principal es, que todos los ficheros del álbum se encuentren en el mismo directorio. Primero, creamos un array de 150 canciones para preservar el orden. 150 ha sido un número arbitrario y debería ser suficiente. Se crea un álbum vacío.

1. Para cada fichero de sonido
2. Se escanean los metadatos. Si el álbum carece de información se guarda el artista, título, año, portada si la hubiera y su duración.
3. Se crea la canción. Si dispone de número de pista, se guarda en su posición en el array, en caso contrario, se añade directamente al álbum.
4. Si detectamos un fichero .jpg, determinamos que será la portada del álbum.
5. Al final, si la canción tenía un número de pista, se inserta por su posición indicada.
6. Vuelta al paso 1 hasta procesar todos los archivos.

Con esta funcionalidad, podemos añadir un álbum a partir de archivos con metadatos, simplificando el proceso si el usuario dispone de copias de los discos en formato digital.

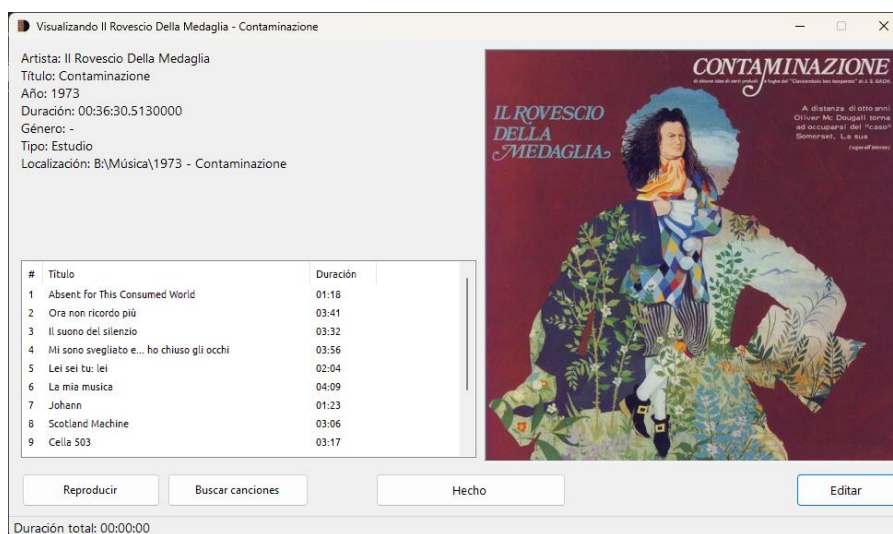


Ilustración 3.7-2 - Álbum creado desde ficheros de música.

3.7.1 Conjuntos de pruebas

Prueba	Resultado de la prueba
Crear una cinta de casete	Todo funcionaba correctamente, aunque se detectaron ciertos fallos a la hora de la visualización de la ficha.
Compartir letras por Twitter	Todo funcionó como se esperaba y Windows abre Twitter y el <i>twit</i> preparado con la letra.
Crear un álbum desde una carpeta	Se inserta correctamente. Se detectaron problemas a la hora de la lectura de los metadatos que conllevaron a implementar una librería nueva.

3.8 Pruebas finales

En este apartado se presentarán las pruebas finales del programa una vez llega a su finalización y comprobar que realiza todas las tareas como se ha estipulado en los requisitos.

3.8.1 Pruebas de verificación del sistema

Como se ha mencionado en el apartado anterior, se han de hacer pruebas una vez con el programa finalizado atendiendo a los requisitos estipulados. Aparte de las pruebas realizadas durante el desarrollo de las iteraciones, aparte, se pensaron en nuevas pruebas para complementar las anteriores y así asegurando de que el *software* funciona correctamente.

- **Usar el reproductor sincronizando con Spotify durante un largo período de tiempo.**
Es importante comprobar la estabilidad del reproductor en este modo, ya que un problema que puede ocurrir es que el token caduque y pueda causar problemas en el programa. En efecto, al caducar el token, las peticiones a la API son rechazadas y el programa se queda bloqueado. Se arregla cambiando el modo de refresco del token y con el sistema PKCE se refresca automáticamente sin problema.
- **Borrar álbumes aleatorios.**
Borrar algo de la base de datos forma parte del ciclo de vida de ésta. Se borran varios como prueba y observar si se sincroniza bien con la vista. Se detectaron problemas, pero fueron arreglados.
- **Borrar un conjunto de álbumes**
Un conjunto de álbumes se refiere a seleccionar varios. El programa podía colgarse ya que intentaba mostrar información del álbum recién borrado. Se arregló insertando comprobaciones previas a la hora de visualizar.
- **Crear álbum manualmente**
Sin cambios desde la última prueba, todo funciona correctamente.
- **Agregar varios álbumes desde Spotify.**
Funciona como se espera. Se añaden todos correctamente.
- **Compartir lo que se está escuchando por Twitter.**
Se espera una ventana de Twitter con el nombre de la canción que se está escuchado. Se encontraron ciertos problemas con caracteres especiales y con los espacios. Se arregló y ahora se crea el twit correctamente.

- **Uso de la lista de reproducción, añadir y borrar canciones**
La lista daba problemas a la hora de borrar canciones y al visualizar el triángulo de reproducción cuando se da doble clic en una entrada. Ambos se encuentran arreglados.
- **Guardado de las bases de datos**
Ningún error a reportar, todo funciona según lo estipulado.
- **Vincular y desvincular cuenta de Spotify**
El programa al principio no detectaba cuando se desvinculaba una cuenta de Spotify, simplemente desactivaba toda conexión para evitar problemas. Ahora sí lo detecta. La vinculación siempre ha funcionado.
- **Reproducción de un CD**
Funciona igual desde la prueba inicial.
- **Probar los filtros de manera exhaustiva**
No se ha detectado ningún problema a la hora del uso de los filtros, la visualización funciona correctamente.
- **Crear distintos discos físicos y visualizar su ficha**
Todas las fichas se generan correctamente en función del artículo que se desee observar.
- **Cambiar distintos parámetros de configuración, como el idioma.**
El idioma se cambia instantáneamente sin necesidad de un reinicio. El resto de parámetros se pueden cambiar sin problema y se guardan para el siguiente inicio.
- **Crear un álbum desde una carpeta de ficheros de sonido**
Al principio se detectaban problemas a la hora de escanear los metadatos. En el momento que se decidió cambiar de librería, todo funciona como es debido.

3.8.2 Pruebas de validación del sistema

Para validar la solución correctamente se han realizado pruebas correspondiendo a los requisitos estipulados inicialmente.

- **Crear una colección musical.** Se puede tener una colección musical sin un límite determinado.
- **Crear un álbum.** Se puede crear un álbum de manera personalizada con el asistente disponible.
- **Descargar metadatos de Internet.** Se puede descargar cualquier metadato disponible en Spotify.
- **Sincronización con Spotify.** Se puede vincular una cuenta, ya tenga suscripción *premium* o no, permitiendo importar la colección de álbumes y el control de la reproducción desde el reproductor.
- **Interfaz familiar.** El *software* dispone de una interfaz que recuerda a los programas clásicos y es familiar para el usuario novel.
- **Reproducción.** El programa puede reproducir una variedad de formatos de sonido de manera correcta.
- **Reproducción de CD.** Complementando con el requisito anterior, el programa también reproduce con éxito los discos compactos de audio.
- **Crear un símil de discos físicos.** El programa permite crear discos físicos de tres tipos, CD, Vinilo y Cassette y dispone de vistas diferenciadas para separarlos. También se pueden visualizar su ficha técnica.
- **Registro de letras de canciones y buscador.** Se puede guardar la letra de todas las canciones de manera personalizada e incluye un buscador de letras muy sencillo y básico.
- **Filtrado de la vista.** La vista principal se puede filtrar mediante la propia opción o desde la barra de texto disponible en la barra de herramientas. Funciona de manera muy rápida.
- **Configuración del software.** El *software* permite la configuración personalizada del usuario. Se puede cambiar la tipografía del sistema, colores, cadenas de texto y el idioma.
- **Idiomas.** El programa dispone de varios idiomas, entre ellos español e inglés.

- **Posibilidad de crear canciones con partes.** Se permite crear canciones con partes que representa una canción resaltada dentro del álbum.

4 MODELO DE NEGOCIO

El desarrollo de este TFG se puede orientar a la creación de una empresa que comercialice el presente *software*. Esta empresa, se dedicaría al desarrollo y constante actualización y estaría orientada a su uso en las tiendas de música donde vendan discos.

Por lo menos, en la provincia de Jaén, donde reside el alumno y el tutor, no existe una empresa de estas características, pero tampoco existe en el resto del país, sólo programas de gestión más genéricos.

4.1 Objetivos

Tenemos que tener en cuenta dos tipos de objetivos, cualitativos y cuantitativos:

4.1.1 Objetivos cualitativos

- Ofrecer soporte profesional con actualizaciones
- Conseguir que los clientes estén felices con el programa y atraer a nuevos clientes
- Obtener reconocimiento y popularidad

4.1.2 Objetivos cuantitativos

- Maximizar beneficios
- Crecer en el mercado
- Mantener la satisfacción de los clientes
- Mejorar en todo lo posible

4.2 Análisis del entorno

El mercado de la compraventa de discos es muy poco común en la provincia de Jaén, dado a que apenas existen este tipo de tiendas y prácticamente sólo se limita al ámbito particular. En cambio, si salimos de la provincia y vamos, por ejemplo, a la capital, podemos ver que tenemos un potencial mercado.

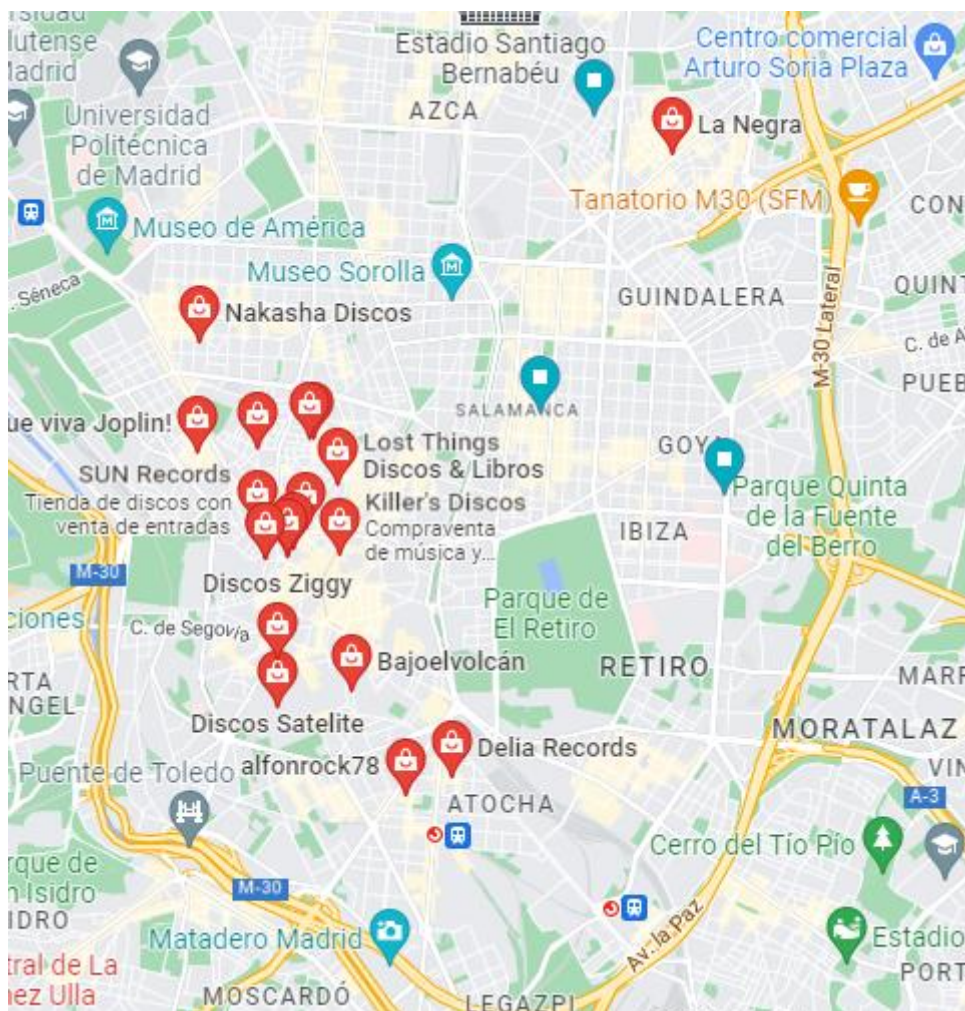


Ilustración 4.2-1 - Fragmento de Google Maps en Madrid resaltando las tiendas de discos

Todas estas tiendas podrían hacer uso de un programa gestor para almacenar todos los artículos de manera organizada para hacer inventario y para crear facturas por si el comprador las necesitara. Además, estas tiendas hacen uso de *Discogs* para la venta en línea de productos. Podría ser una opción interesante para este tipo de tiendas que usen un *software* como el presentado aquí.

No sólo eso, este programa también puede estar orientado a particulares que también quieran vender sus productos y requieran soporte profesional o solicitud de funcionalidades a demanda. Esto nos indica que tenemos bastante clientela potencial.

En cuanto a la posible competencia, apenas tenemos en el sector, aunque exista *Discogs*, sabemos que está orientado a prestar servicios en Internet y este *software* funciona de manera local. Se puede considerar competencia al resto de *software* que gestiona inventarios, pero de manera más genérica.

Una herramienta para evaluar una idea emprendedora es la matriz DAFO. (Debilidades, amenazas, fortalezas, oportunidades), que comprende los factores del entorno.

Interno	Externo
Debilidades	Amenazas
Empresa sin reconocimiento inicial Poca experiencia a la hora de emprender	Clientela conservadora. Dificultad para intentar vender el producto.
Fortalezas	Oportunidades
Personal cualificado para prestar el servicio Producto exclusivo y especializado Conocimiento en el mundo musical y multimedia	Abrirse paso en el mercado con un producto único Crecimiento de la sociedad

Tabla 9 - Matriz DAFO

A la vista de la matriz, se pueden ver que se disponen de bastantes fortalezas, pero las debilidades y las amenazas ya suponen que la creación de la sociedad tenga bastante riesgo.

4.3 Forma jurídica de la empresa

En España existen varias formas jurídicas de empresas, cada una con sus pros y sus contras. El Ministerio de Industria, Comercio y Turismo para ello ha creado una herramienta en Internet para ayudar a elegir la forma jurídica en función de los objetivos y de las condiciones de la empresa.

Elección de la forma jurídica

Tipo de empresa	Nº socios	Capital	Responsabilidad
Emprendedor de Responsabilidad Limitada	1	No existe mínimo legal	Ilimitada con excepciones
Sociedad Limitada de Formación Sucesiva	Mínimo 1	No existe mínimo legal	Limitada al capital aportado en la sociedad
Sociedades Profesionales	Mínimo 1	Según la forma social que adopte	Limitada al capital aportado en la sociedad

Ilustración 4.3-1 - Herramienta para elegir la forma jurídica

La herramienta da información relevante a la empresa como:

- Descripción.
- Normativa.
- Características.
- Órganos sociales.
- Número de socios.
- Responsabilidad.
- Capital.
- Fiscalidad.
- Socios
- Cómo se crea y dónde acudir

En este caso la empresa tendrá como forma jurídica la Sociedad Profesional. Es un tipo de sociedad que se constituye para el ejercicio en común de una actividad profesional. Una actividad profesional necesita titulación universitaria oficial o profesional. Será necesaria su inscripción en el Registro Mercantil. No es necesario realizar ninguna aportación para su constitución. La fiscalidad dependerá de la forma social que se adopte.

La normativa a aplicar para este tipo de empresa es la siguiente:

- Ley 2/2007 de sociedades profesionales (supletoriamente, por las normas correspondientes a la forma social adoptada).
- Ley 25/2009 de modificación de diversas leyes para su adaptación a la Ley sobre el libre acceso a las actividades de servicios y su ejercicio.

5 CONCLUSIONES Y TRABAJOS FUTUROS

Para concluir, se ha visto que la realización de un prototipo para gestionar una colección musical de manera local es una idea bastante innovadora y con vistas a crear un programa serio y bastante complejo. Aunque existan alternativas en línea, en vez de que sean competidoras, se pueden usar a nuestro favor y complementar el programa con datos que dispongan.

Al ser un programa con interfaces, las posibilidades de extensión son prácticamente ilimitadas. Siempre se puede cambiar o mejorar algo dentro del proyecto.

Como trabajos futuros, primero siempre es incrementar la estabilidad y usabilidad del programa. Errores de programación siempre se van a encontrar. Luego, implementar funcionalidades interesantes como las siguientes:

- Posibilidad de hacer una copia de un CD a ficheros de sonido locales
- Buscar los metadatos de un CD por Internet
- Aumentar la sincronización con Spotify, permitiendo crear listas de reproducción y/o descargarlas.
- Sincronización con Discogs para poder buscar metadatos
- Mejorar la búsqueda de letras.
- Intentar conectar con más servicios de *streaming*.
- Explorar más formatos de sonido desconocidos, como SPC o MOD para reproducir *chiptune*.¹²

Por último, comentar que también existe la posibilidad de mudar el programa a otro lenguaje de programación y librería para desbloquear la portabilidad entre sistemas operativos.

¹² La música *chiptune* se refiere a música de 8 o 16 bits, muy típica en los años 90 y que recuerda a los videojuegos.

6 APÉNDICES

6.1 Guía original del Trabajo Fin de Título

Identificación: [21/22-3276].

Escuela Politécnica Superior de Jaén

Propuesta de Trabajo de Fin de Grado

Grado en:

Especialidad:

Mención:

Título del Trabajo:

Idioma:

Descripción Corta del TFG:

Este trabajo consiste en la realización de un **software de gestión discográfica integral** para la indexación y tratamiento de colecciones de música en formatos tanto físicos como digitales. Aunque ya existen soluciones para algunas de las necesidades de gestión de los audiófilos, este software busca ofrecer una solución integrada que incluya funcionalidades como la indexación de música y discos de fuentes físicas (CDs, vinilos, casetes, etc.) y digitales (ficheros, streaming, etc.), la gestión y búsqueda de metadatos y letras de canciones, una reproducción integrada de archivos en los formatos más extendidos (incluyendo listas de reproducción), así como la sincronización con algún sistema de streaming como Spotify o Tidal.

Datos del Trabajo Fin de Grado:

Modalidad del TFG

Proyecto de Ingeniería

Tipo de TFG

TFG General

TFG Específico

TFG en equipo

(Debe adjuntarse memoria justificativa)

Número máximo de estudiantes

Tutor del Trabajo Fin de Grado:

Tutor/a

Nombre:

Departamento:

Área de Conocimiento:

- Este TFG tiene un segundo tutor
- El tutor es de la Escuela
- El tutor es ajeno al Centro

Segundo Tutor

Datos del Alumno Asignado

Alumno/a asignado/a

Nombre:

DNI:

Dirección:

Teléfono:

Correo-E:

Conocimientos/Requisitos Previos Recomendados:

No es necesario tener conocimientos específicos previos.

Objetivos del TFG:

- Estudiar el estado del arte de sistemas de gestión discográfica.
- Diseñar e implementar un software para la gestión discográfica que incluya las opciones enumeradas en apartados anteriores.
- Realizar pruebas de utilización con colecciones reales y hacer pruebas de sincronización con sistemas de streaming como Spotify o Tidal.
- Documentar convenientemente el desarrollo del proyecto y sus resultados.

Metodología a Desarrollar:

- Realizar un breve estado del arte de sistemas de gestión discográfica existentes.

- Seleccionar y utilizar una metodología basada en Ingeniería del Software para el análisis de requisitos, diseño, implementación, prueba y documentación. Se recomienda utilizar una metodología incremental.
- Seleccionar todas las herramientas necesarias, incluyendo lenguajes, SDKs, toolkits y librerías de terceros que pudieran servir para el desarrollo.
- Realizar el análisis del sistema, detallando requisitos funcionales y no funcionales.
- Detallar la estimación temporal y de costes de desarrollo.
- Realizar el desarrollo del software, así como de los conjuntos de datos de prueba.
- Realizar las pruebas necesarias para la evaluación del software y documentar los resultados.
- Redacción de la documentación final requerida.

Documentos y formatos de entrega:

- Documentación generada durante el proceso, de acuerdo a las buenas prácticas de la ingeniería del software. Esto se aplicará tanto en los proyectos de Ingeniería como en los trabajos teóricos o experimentales, es decir, siempre que se genere algún tipo de desarrollo, tanto software como webs, scripts, etc.
- Para el formato de documentación de los **Proyectos de Ingeniería** se utilizará la norma **UNE 157801**. Tal y como especifican las normas de estilo de la EPSJ:
 - *'Los Proyectos de Ingeniería deberán presentar una estructura y contenido adaptados a la norma **UNE 157001** - Criterios generales para la elaboración de proyectos - u otra derivada de ésta, en función de la naturaleza del proyecto'*.
 - *'...para el caso de los sistemas de información informatizados (sistemas informáticos, sistemas de información geográfica, etc.) está la norma derivada de la anterior **UNE 157801** - Criterios generales para la elaboración de proyectos de sistemas de información'*.
 - Para ello, se puede utilizar también la Norma CCII-N2016-02 (**Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática**) del Consejo General de Colegios Profesionales de Ingeniería en Informática, que incluye e implementa la norma **UNE 157801**.
- Memoria del trabajo, incluyendo manuales y anexos, en formato PDF.

- Código fuente completo y ejecutables del prototipo o software desarrollado.
- Documentos y archivos adicionales a los que se haga mención expresa en la memoria.
- Vídeo de demostración de la aplicación o de la experimentación realizada.
- Anexos para la presentación del trabajo:
 - Informe favorable del tutor.
 - Autorización de publicación, si procede.

6.2 Instalación y configuración del sistema

Se debe descargar el archivo .zip. Dentro del .zip, tendremos el ejecutable y un directorio llamado “*idiomas*”. Es recomendable dejarlo en una carpeta aparte.

NOTA: la carpeta *idiomas* es imprescindible, ya que, sin ella, la aplicación no podrá iniciar y no se lanzará nada.

Una vez realizado esto, para iniciar el programa, pulse doble *clic* sobre el ejecutable.

6.3 Requisitos del sistema

La aplicación sólo es compatible en los sistemas operativos Windows, versiones 7, 8, 8.1, 10, 11 (recomendable versión 1904x¹³) y 11. (compilación 22000 o 22651). Es indispensable contar con NET 6 instalado en el sistema.

No se requiere ningún componente hardware específico, pues si el ordenador ejecuta Windows sin problemas, el programa se ejecutará correctamente.

No obstante, es obligatorio una conexión a internet para ciertas funcionalidades como la integración con Spotify, aunque no es completamente necesaria.

Al principio se recomiendan unos 50mb de espacio en disco, no importa si es disco duro mecánico o en estado sólido (SSD). También se puede usar una unidad móvil para ejecutar el programa, ya sea una unidad USB o una tarjeta SD. Este espacio podría verse aumentado en función del número de carátulas descargadas y de lo grande que sea la base de datos.

En cuanto a requerimientos de memoria RAM, no son necesarias más de 150mb disponibles.

¹³ Es recomendable usar las versiones, 19042, 19043 o 19044. (20H2, 21H1 o 21H2), ya que son las más recientes.

6.4 Manuales de usuario

En este apartado se expondrá el manual de usuario, donde se explica con detalle y simpleza todas las funcionalidades del programa a disposición del usuario, por si se necesita su consulta en cualquier momento.

6.4.1 Argumentos de la línea de comandos

Cassiopeia dispone de varios argumentos opcionales para lanzar el programa de una manera específica. Se pueden lanzar con la memoria de comandos o mediante un acceso directo. (explicación en apartado siguiente)

- -reproductor / -player
Abre el programa con sólo el reproductor.
- -modoStream
No carga la interfaz y sólo ejecuta una tarea en segundo plano que recoge lo que el usuario esté escuchando en Spotify y lo saca a un fichero llamado np.txt con la cadena configurada. Más detalles en el apartado [Configuración - Cadena retransmisión](#)
- -consola
Lanza una consola con el programa principal para mostrar detalles sobre la ejecución con más detalle. (en inglés)
- -noSpotify
Deshabilita la funcionalidad de Spotify. Es un modo sin conexión.

6.4.1.1 Configuración de los argumentos con un acceso directo

Primero vamos al directorio donde se encuentra el ejecutable del programa.

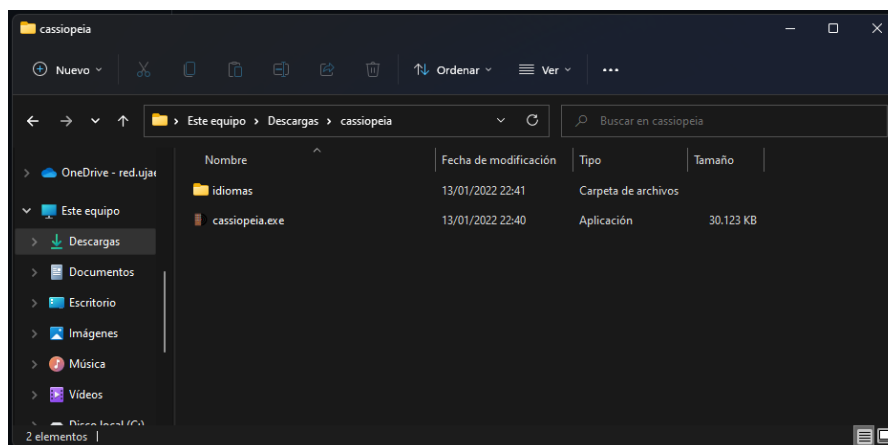


Ilustración 6.4-1 - Directorio raíz del programa (donde reside el ejecutable)

A continuación, de clic derecho sobre Cassiopeia. Si se encuentra en Windows 11, de clic izquierdo y luego presione Mayúsculas + F10. Aparecerá un menú similar al siguiente:

Haz clic en Crear acceso directo.

Aparecerá un nuevo fichero que es el acceso directo que hemos creado. Seguidamente, de clic derecho en ese acceso directo y luego en *Propiedades*. Escriba al final del cuadro de *Destino* los parámetros que necesite separados con un espacio. Ejemplo en la ilustración 6.4-3.

De esta manera, puede introducir todos los parámetros que necesite. También, si fuera necesario, se puede mover el acceso directo de sitio para tenerlo en el escritorio.

Un caso de uso puede ser lanzar el programa sólo con la interacción de Spotify en segundo plano para imprimir lo que está escuchando el usuario para ponerlo en un programa de creación de contenidos en *streaming*, por ejemplo, [OBS](#).

En el momento de escribir este manual (23/07/22) sólo están disponibles esos cuatro parámetros.

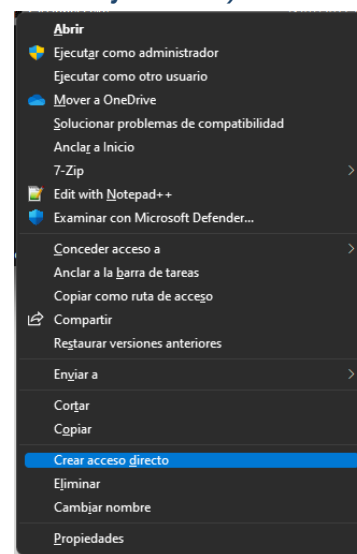


Ilustración 6.4-2 - Menú contextual de Windows

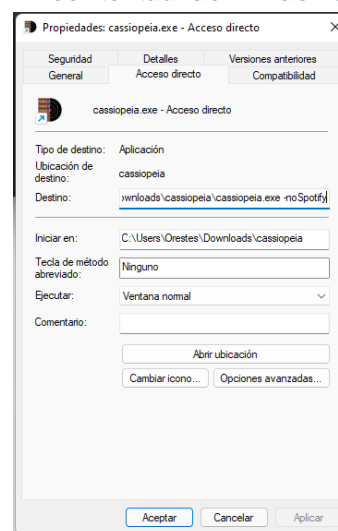


Ilustración 6.4-3 - Ventana de propiedades con los parámetros.

6.4.2 Ventana principal

Al ejecutar el programa por primera vez, verá una ventana como esta:

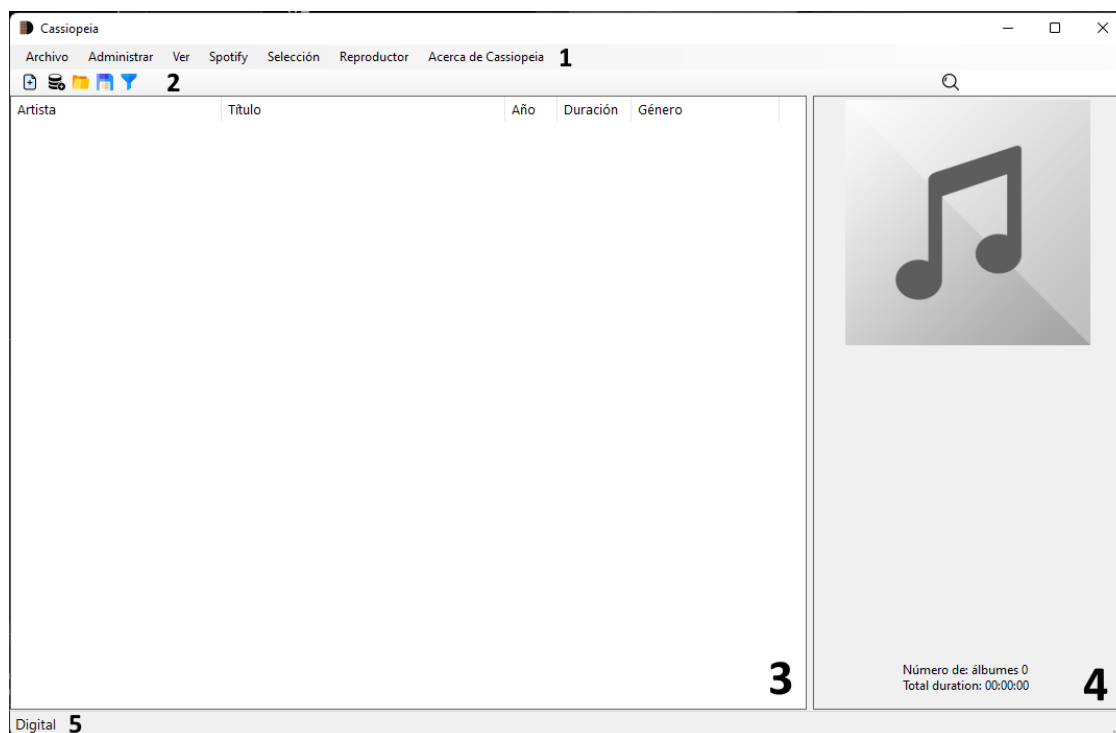


Ilustración 6.4-4 - Ventana principal de Cassiopeia sin álbumes

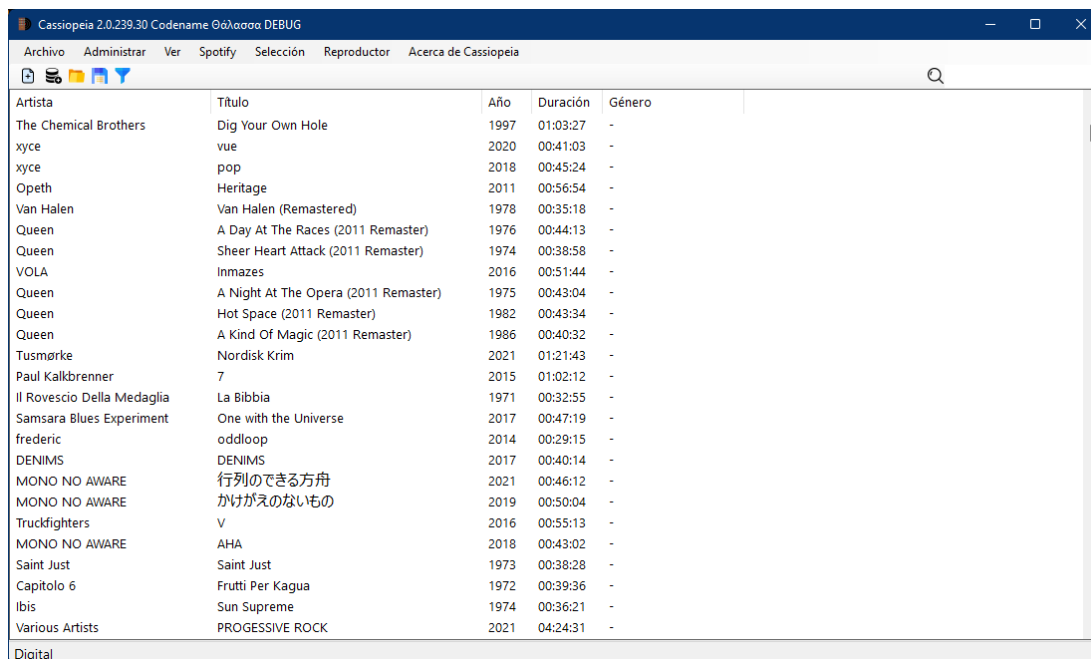
Una ventana con unos cuantos iconos y una vista completamente vacía. Los elementos numerados son los siguientes:

1. La barra de menús.
Contienen botones para realizar acciones en el programa como crear bases de datos nuevas, configurar la vista, lanzar el reproductor y configurar la herramienta.
2. Barra de iconos.
Contienen iconos para el acceso rápido a tareas comunes de la herramienta. A la derecha, hay una barra de búsqueda para filtrar los álbumes.
3. Vista de los álbumes.
Contiene la lista de todos los discos disponibles en la base de datos en función de la vista seleccionada.
4. Vista lateral.
Contiene una vista rápida del álbum seleccionado, su nombre, duración y su portada. Abajo resume la vista principal enseñando la cantidad de discos y la duración de todos ellos. Desde aquí se puede copiar la portada.

5. Barra inferior.

Muestra el tipo de vista actual. Puede ser Digital, CD o Vinilo.

Si abrimos el programa con una base de datos más completa, tendremos una vista similar a la siguiente:



Artista	Título	Año	Duración	Género
The Chemical Brothers	Dig Your Own Hole	1997	01:03:27	-
xyce	vue	2020	00:41:03	-
xyce	pop	2018	00:45:24	-
Opeth	Heritage	2011	00:56:54	-
Van Halen	Van Halen (Remastered)	1978	00:35:18	-
Queen	A Day At The Races (2011 Remaster)	1976	00:44:13	-
Queen	Sheer Heart Attack (2011 Remaster)	1974	00:38:58	-
VOLA	Inmazes	2016	00:51:44	-
Queen	A Night At The Opera (2011 Remaster)	1975	00:43:04	-
Queen	Hot Space (2011 Remaster)	1982	00:43:34	-
Queen	A Kind Of Magic (2011 Remaster)	1986	00:40:32	-
Tusmørke	Nordisk Krim	2021	01:21:43	-
Paul Kalkbrenner	7	2015	01:02:12	-
Il Rovescio Della Medaglia	La Bibbia	1971	00:32:55	-
Samsara Blues Experiment	One with the Universe	2017	00:47:19	-
frederic	oddlloop	2014	00:29:15	-
DENIMS	DENIMS	2017	00:40:14	-
MONO NO AWARE	行列のできる方舟	2021	00:46:12	-
MONO NO AWARE	かけがえのないもの	2019	00:50:04	-
Truckfighters	V	2016	00:55:13	-
MONO NO AWARE	AHA	2018	00:43:02	-
Saint Just	Saint Just	1973	00:38:28	-
Capitolo 6	Frutti Per Kagua	1972	00:39:36	-
Ibis	Sun Supreme	1974	00:36:21	-
Various Artists	PROGRESSIVE ROCK	2021	04:24:31	-

Ilustración 6.4-5 - Ventana principal con discos y sin vista lateral.

Esta es la ventana que verá durante la mayor parte del tiempo y es donde podrá observar todos los álbumes insertados en el sistema.

Podrá filtrarlos, crear los medios físicos y hacer todo lo posible con el programa.

6.4.2.1 La barra de herramientas

La barra de herramientas contiene botones para realizar acciones comunes evitando tener que buscar por menús y por ventanas anidadas.

Vamos a comentar la barra de herramientas completa (sólo cuando seleccionas un álbum)

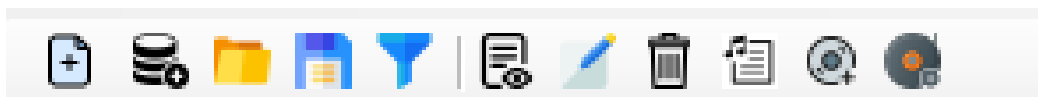


Ilustración 6.4-6 - La barra de herramientas con un álbum seleccionado

En orden de izquierda a derecha:


- Añadir nuevo álbum

- Crear nueva base de datos
- Abrir un fichero que contiene discos.
- Guardar
- Filtrar la vista.
- Separador para opciones con un álbum seleccionado.
- Ver el álbum.
- Editar el álbum.
- Borrar el álbum
- Ver las letras.
- Crear un CD.
- Crear un vinilo.

6.4.3 Tareas comunes

En este apartado se explicarán tareas comunes al usar el programa como crear nuevos discos, manejarlos... entre otros.

6.4.3.1 Crear un nuevo álbum

Para crear un nuevo álbum desde cero vaya al menú Archivo → Nuevo álbum o pulsando sobre el icono  en la barra de herramientas o pulsar Ctrl + N.

Se abrirá una ventana con varios campos a rellenar.

Rellene todos los metadatos del álbum como en la figura de ejemplo.

Si hace clic en Añadir, se abrirá otra ventana para crear las canciones del álbum.

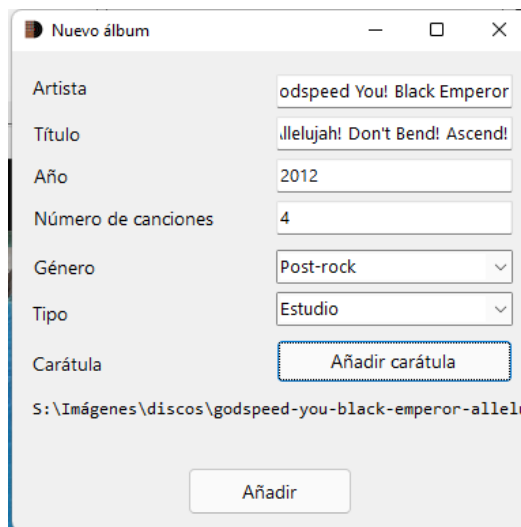
El formulario será similar como la mostrada en la Ilustración 7.5-8. Introduzca el nombre de la canción seguidamente de su duración en minutos y segundos.

Se crearán tantas ventanas como veces haya indicado en el apartado Número de canciones en la ventana anterior.

Tenga en cuenta de que no podrá volver al programa principal hasta que no haya terminado este proceso, a no ser que cancele el proceso o pulse la X roja en la esquina superior derecha, que también cancelaría el proceso.

6.4.3.1.1 Crear una canción con partes

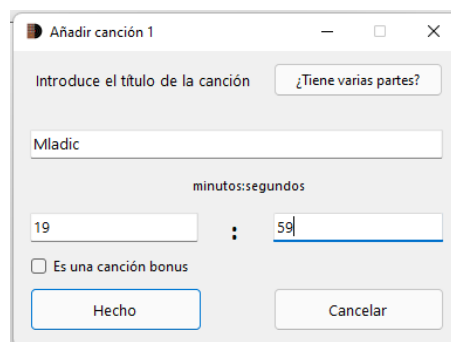
Existe en el mundo de la música canciones largas con partes que se dividen en partes para que no sea demasiado tedioso al escuchar en una lista de reproducción o por el simple hecho de que cada parte de la canción es distinta entre las demás.



Formulario 'Nuevo álbum' con los siguientes campos:

- Artista: odspeed You! Black Emperor
- Título: lilelujah! Don't Bend! Ascend!
- Año: 2012
- Número de canciones: 4
- Género: Post-rock
- Tipo: Estudio
- Carátula: Añadir carátula
- Carpeta: S:\Imágenes\discos\godspeed-you-black-emperor-allelu
- Botón: Añadir

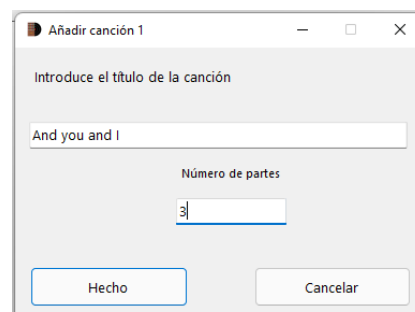
Ilustración 6.4-7 - Formulario para crear un álbum con datos de ejemplo.



Formulario 'Añadir canción 1' con los siguientes campos:

- Introduce el título de la canción: Mladic
- ¿Tiene varias partes?: (checkbox no marcado)
- Duración: 19 minutos, 59 segundos
- Es una canción bonus: (checkbox no marcado)
- Botones: Hecho, Cancelar

Ilustración 6.4-8 - Formulario para crear una canción.



Formulario 'Añadir canción 1' transformado para crear una canción con partes:

- Introduce el título de la canción: And you and I
- Número de partes: 3
- Botones: Hecho, Cancelar

Ilustración 6.4-9 - El formulario transformado para crear una canción con partes

En el mundo del rock progresivo¹⁴ hay muchos ejemplos de estas canciones. Si desea crear una canción así, pulse en el botón de ¿Tiene varias partes?

El formulario se volverá a abrir con menos opciones, donde nos pedirá el número de partes a crear.

Una vez indicado el número de partes, se añaden el resto de partes como si fuera una canción normal y una vez terminado con el proceso de añadir las partes, se resume a añadir una canción normal hasta terminar de añadir el álbum.

En cualquier momento se puede cancelar todo el proceso.

6.4.3.2 Visualizar un álbum

Una vez que tengamos algún álbum instalado en el sistema, ya se pueden realizar acciones con él.

Para visualizar un álbum, simplemente haz doble clic o selecciónalo y luego presione ENTER.

También puede presionar sobre el botón  en la barra de herramientas.

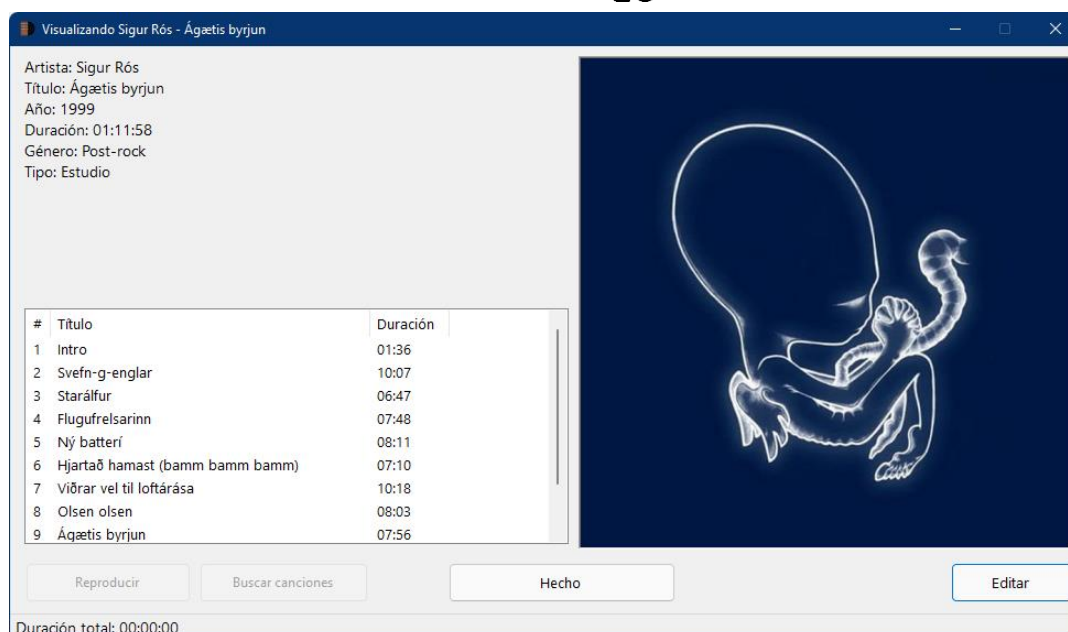


Ilustración 6.4-10 - Visualizando un álbum de ejemplo.

¹⁴ Un par de ejemplos de canciones con partes son *And You and I* de *Yes* o *Messiah Complex* de *Haken*.

6.4.3.3 Editando la información de un álbum

Para editar los datos de un álbum puede presionar en el botón Editar en el formulario de visualizado o bien puede seleccionar un álbum de la vista principal y pinchar en el icono de la barra de herramientas.

En la ventana que aparece, podemos personalizar y cambiar los parámetros del álbum. También puede añadir o quitar canciones.

Vamos a comentar las opciones resaltadas:

The screenshot shows a window titled "Editando Sigur Rós - Ágætis byrjun". The form contains the following fields and buttons:

- Artista: Sigur Rós
- Título: Ágætis byrjun
- Año: 1999
- Género: Post-rock
- URI de Spotify: 1DMMvY1KmolI3Y9fVEZDUVC (labeled 5)
- Tipo: Estudio (labeled 6)
- Directorio donde se alojan las canciones: Buscar directorio (labeled 1)
- Carátula: Buscar (labeled 2)
- Lista de canciones (labeled 3):

Intro	Ný batterí	Ágætis
Svefn-g-englar	Hjartað hamast (bamm bamm bamm)	Avalon
Starálfur	Viðrar vel til loftárása	
Flugufrelsarinn	Olsen olsen	
- Buttons: Añadir canción (labeled 4), Hecho, Cancelar

Ilustración 6.4-11 - Formulario para editar un álbum

1. Directorio donde se alojan las canciones.

Este directorio se especifica cuando tiene el disco en formato digital en una carpeta aparte, para poder reproducirlo con el programa

2. Carátula.

Permite seleccionar una imagen para elegirla como portada del álbum.

3. Lista de canciones.

Muestra todas las canciones del álbum.

4. Añadir canción.

Permite añadir una canción al álbum.

5. URI de Spotify.

Es un código propio de Spotify que identifica al álbum dentro de la plataforma. Imprescindible si se quiere dar la orden de escuchar el álbum desde la aplicación.

6. Tipo de álbum.

Especifica si es un álbum de estudio, recopilatorio... etc.

Si necesita editar el título de una canción, tan sólo debe de hacer doble clic en cualquier canción de la lista.

6.4.3.3.1 Configurar una canción como canción bonus.

Muchas veces los álbumes sufren reediciones como los libros, para volver a sacar una tirada y vender más o bien porque sale una versión remasterizada para un nuevo formato, como el disco compacto.

Como el disco compacto tiene más capacidad que un elepé clásico, las discográficas optaron por meter más canciones en el disco para tener más contenido.

Hoy en día es más exagerado, ya que con la era digital y del streaming, la cantidad de canciones extra no tiene límite.

En el programa, para configurar una canción como bonus puede, en la vista de visualizar, dar clic derecho a una canción y luego seleccionar “Configurar como canción bonus y viceversa” o en la ventana de editar álbum marcar la etiqueta de “Es una canción bonus”.

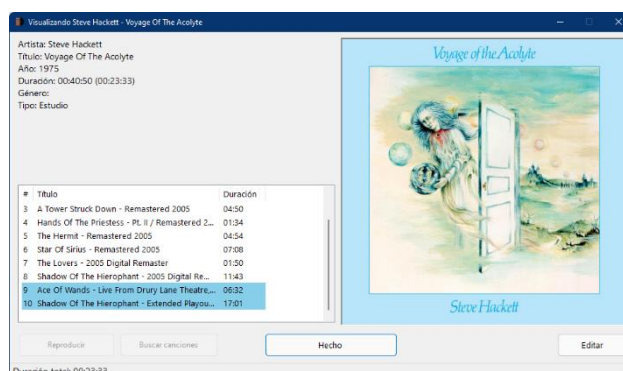


Ilustración 6.4-12 - Un álbum con canciones bonus

Una vez realizado esto, la vista cambiará, siendo recalculada la propia duración del álbum y con filas azules en las canciones, que representan las canciones extra.

Es buena idea tener una copia del álbum con todas las canciones bonus, a no ser que tenga un millón de ediciones distintas, que puede pasar, pero no es lo común.

6.4.3.3.2 Cómo crear una canción con partes a partir de canciones existentes.

A veces nos interesa fusionar una canción porque es muy larga, siempre apetece escucharla de una sentada, pero en nuestra edición aparece como canciones separadas. Esto es molesto y para ello existe un mecanismo para tratar de juntar las canciones separadas en una mega canción.

Para crear una canción con partes, selecciona canciones **consecutivas** y con el menú del clic derecho, aprieta sobre “*Fusionar en canción con partes*”. Aparecerá una entrada con color salmón indicando que es una canción con partes. Tendrá el nombre de la primera parte de la canción. Luego, en el formulario de editar canción podrá restaurar su nombre original. Si da doble clic, podrá ver con detalle las partes que la componen.

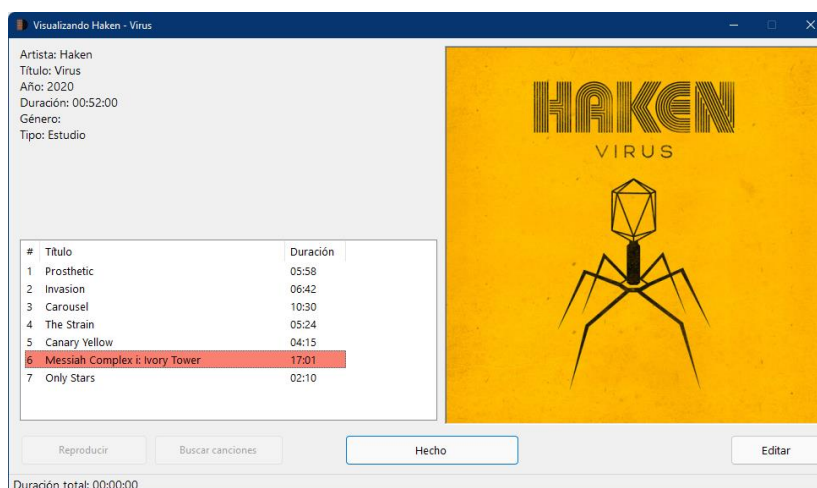


Ilustración 6.4-13 - Álbum con una canción con partes

6.4.3.4 Vincular un directorio con un álbum del sistema

Cuando disponga de la versión digital con los ficheros de audio en su computadora, puede vincular esos archivos a los álbumes del sistema. Este sistema se hace muy sencillo si realiza una organización previa y simplemente se trata de juntar los ficheros de un álbum en una carpeta separada.

Hay que resaltar que para que esto funcione, los metadatos o título del fichero **debe contener** al nombre de la canción. Por ejemplo, si tenemos una canción que se llame “*Carousel*”, con un fichero llamado 03-carousel.mp3 funcionará.

Para asociar el directorio, debemos abrir la vista de edición de un álbum como ya se explicó.

Posteriormente, hacemos clic en *Buscar directorio*.

Volvemos a la vista anterior y pinchamos en *Buscar canciones*. Si todo ha ido bien, se mostrará el siguiente mensaje (figura 7.5 - 14).

Si no ha sido posible vincular los archivos con las canciones del álbum, se mostrará el siguiente mensaje (figura 7.5 - 15)

En caso de que aparezca este mensaje de error, lo mejor es comprobar el log para obtener más detalles. Se puede acceder con el menú Administrar → Ver log.

Obtendremos información sobre las canciones que están dando problemas para ser vinculadas. En cambio, si todo ha ido correctamente, podremos reproducir el álbum.

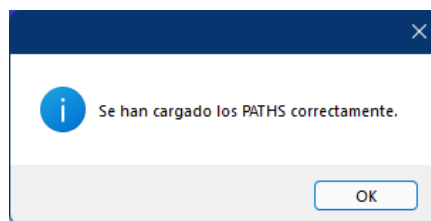


Ilustración 6.4-14 - Mensaje si se han buscado correctamente las canciones

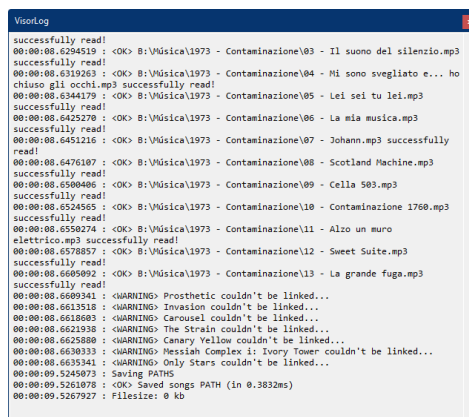


Ilustración 6.4-15 - Log de la aplicación

6.4.3.5 Añadir un álbum desde un directorio

Repitiendo el caso anterior, es el concepto inverso. Tenemos un álbum en ficheros de sonido y queremos importarlo en el sistema. Podemos hacerlo manualmente, pero tardaríamos bastante rato. Existe una forma automatizada de hacerlo.

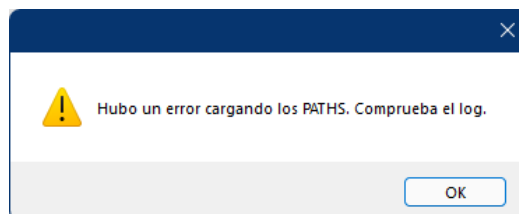


Ilustración 6.4-16 - Mensaje de advertencia de que no se han podido vincular los archivos

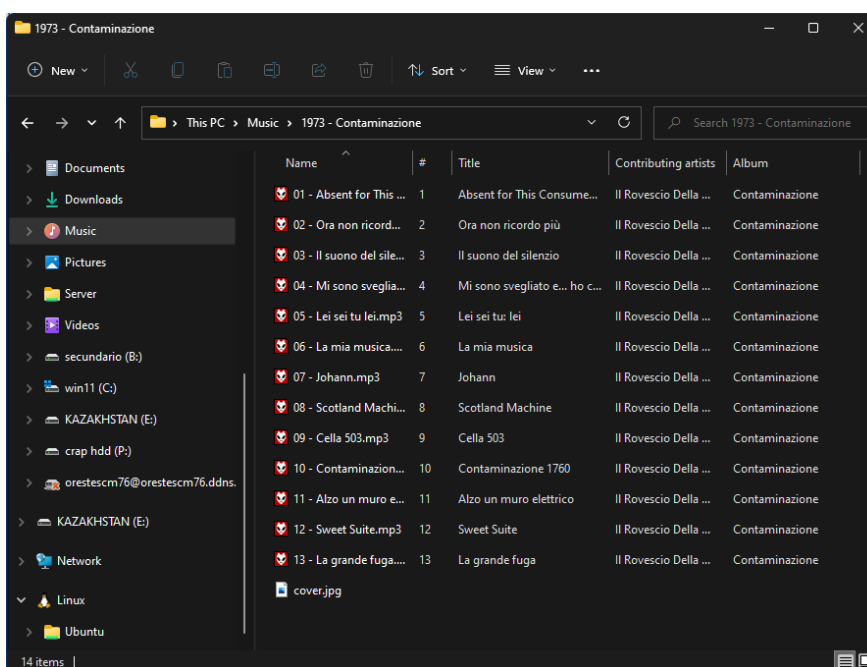



Ilustración 6.4-17 - Una carpeta con un álbum, que vamos a meter en el sistema

La opción está dentro del menú Archivo → Nuevo álbum desde un directorio.

Se nos abre un selector de carpetas y con este navegamos hasta el directorio que queremos seleccionar.

Una vez seleccionado el directorio, se creará el álbum inmediatamente y aparecerá con su directorio asociado y será posible reproducirlo con el reproductor integrado.

6.4.3.6 Visualización y edición de letras

Cassiopeia dispone de un visor y editor de letras de canciones bastante básico que cumple con la función de consultar la letra de una canción si se tiene registrada. Para acceder a esa funcionalidad, tenemos que seleccionar un álbum y luego presionar clic derecho y después en Ver la letra o bien pulsando en el botón  en la barra de herramientas.

Se abrirá la siguiente ventana:

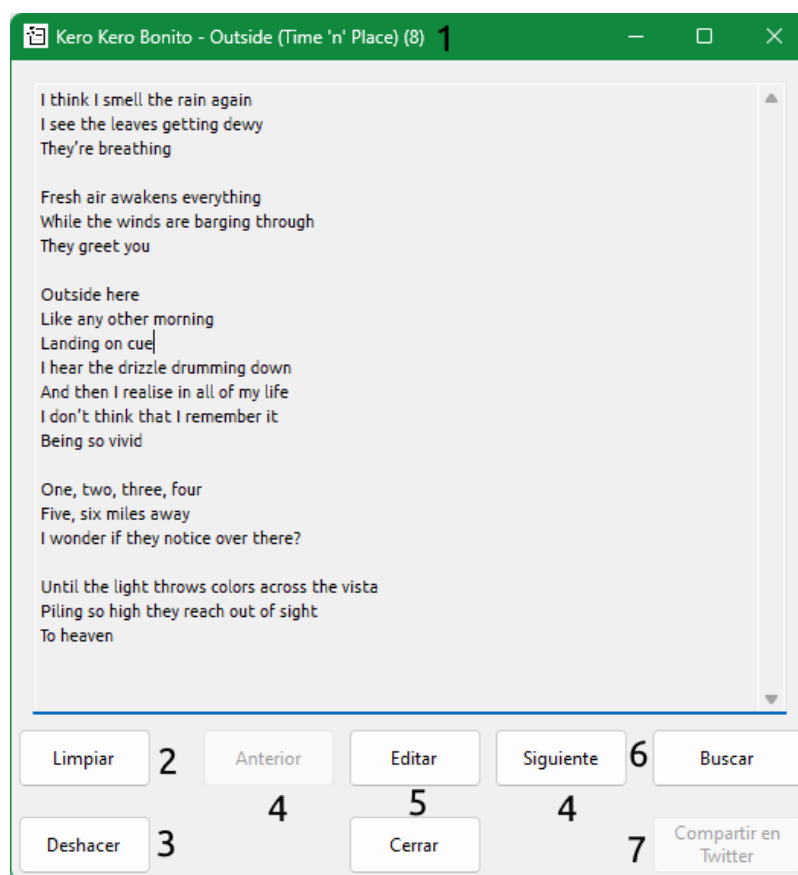


Ilustración 6.4-18 - Ventana de letras

1. Título de la ventana. Contiene la canción seleccionada, el álbum y el tamaño de la letra con la que se está visualizando.
2. Este botón borra la letra
3. Deshace los cambios a su estado anterior antes de guardar
4. Botones de interacción. Cada vez que se presione alguno, se guardará la letra actual.
5. Permite levantar la restricción de sólo lectura para editar el contenido de la letra.
6. Intenta buscar la letra por internet. Usa songlyrics.com como proveedor.
7. Compartir en Twitter. Si se selecciona un fragmento, se habilitará el botón. Copia el texto y abre Twitter para directamente publicar el *twit*.


El contenido que haya dentro no tiene porqué seguir ningún formato ni nada. Es completamente personalizable por el usuario y cada uno puede escribir la letra como le plazca.

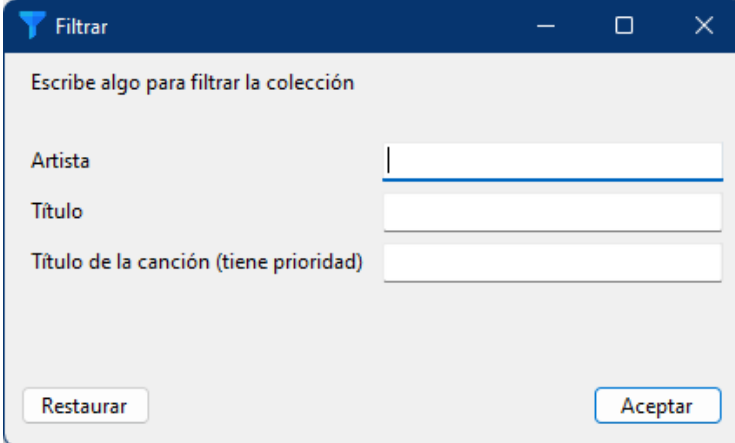
Si se presiona la tecla control mientras se usa la rueda del ratón, se cambia el tamaño de la letra. Para compartir una letra simplemente seleccione el fragmento que desee. El botón se habilitará. Lanzará Twitter con el fragmento copiado automáticamente.

6.4.3.7 Filtrar la vista principal.

Puede pasar que tengas una base de datos amplia, que contenga cientos de álbumes. Puede volverse tedioso el hecho de tener que buscar manualmente algún álbum. Cassiopeia incluye una opción de filtrar álbumes que se ejecuta desde dos secciones. Desde su propio menú y desde la barra de búsqueda.

Desde el menú.

Para acceder a la funcionalidad de filtrar, basta con simplemente pinchar en el botón  en la barra de herramientas.



La imagen muestra una ventana de diálogo titulada "Filtrar". El título de la ventana incluye un icono de embudo. El contenido de la ventana es el siguiente:

Escribe algo para filtrar la colección

Artista

Título

Título de la canción (tiene prioridad)

Restaurar

Ilustración 6.4-19 - Ventana para filtrar

En esta ventana podemos filtrar la vista principal. Podemos escribir lo que queramos y en el momento que presionemos "Aceptar", se aplicará el filtro. El primer filtro corresponde a el artista, el segundo al título del álbum y, por último, a título de la canción. Este último tiene prioridad porque es el primero que ejecuta, ya que, si especifica un título de canción, primero elimina los discos que no contengan ese título en sus canciones.

Si por ejemplo escribimos "Houdini" en el filtro de canción, obtendremos un resultado:

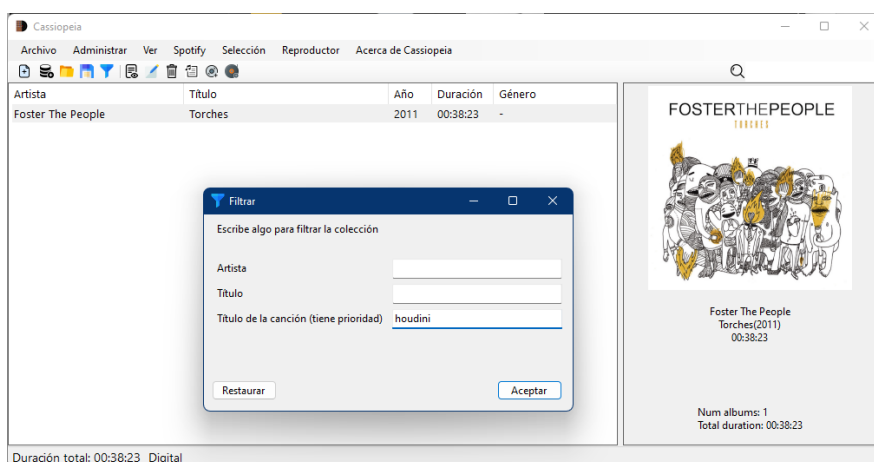


Ilustración 6.4-20 - Demostración del filtro de canciones

También, podemos escribir en la barra de texto a la derecha de la lupa. Lo que escribamos, aplicará los tres filtros a toda la base de datos y nos mostrará el resultado.

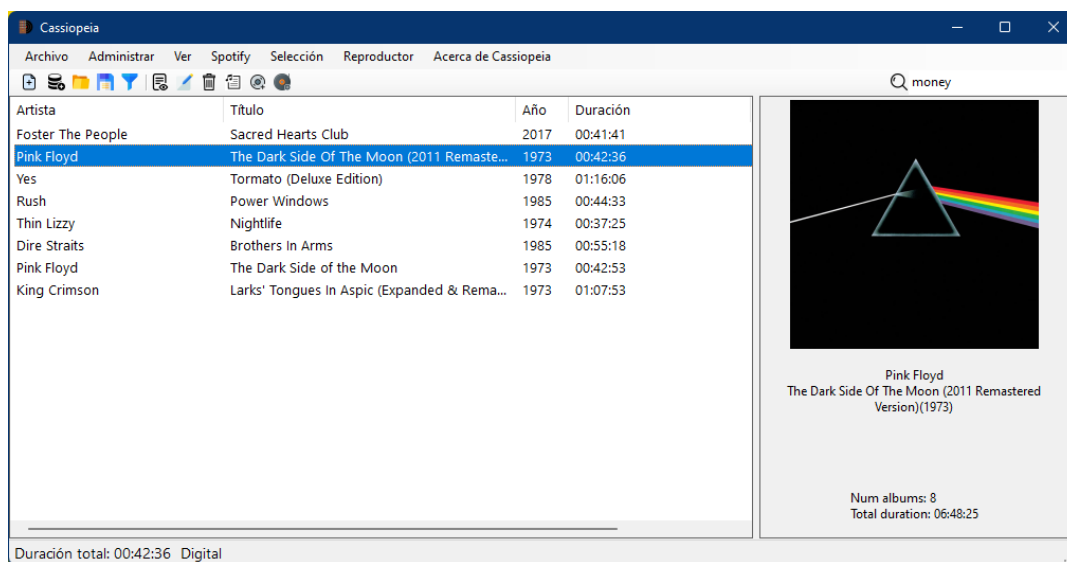


Ilustración 6.4-21 - Demostración del filtro con la barra de búsqueda


Como podemos apreciar, nos busca todos los discos con “money” en la base de datos. En este caso, sólo aparecen discos donde en alguna de sus canciones incluye la palabra “money”.

Todo el proceso de filtrado es extremadamente rápido, llegando a ejecutar en menos de 1 milisegundo, pudiendo decir así, que van a tiempo real.

6.4.4 Gestión de discos físicos

Este programa permite la gestión de discos físicos, tanto discos de vinilo como discos compactos (CD). Cada formato posee su vista única para su cómoda visualización. Antes de crear un medio físico, deberá asegurarse de que tenga los metadatos correctos del álbum.

6.4.4.1 Creación de un Disco Compacto.

Para crear un CD puede pinchar en el botón  en la barra de herramientas o bien en el menú contextual tras pulsar clic derecho en un álbum de la vista principal, en *Crear CD a partir de este álbum*.

Se abrirá una ventana nueva con varias opciones.

1. Estado de la carátula. Determina el estado físico de la carátula y del exterior, como la “*jewel case*” o la caja de cartón que tenga.
2. Estado del disco. Determina lo degradado que está el disco. Ambas se soportan en el estándar Goldmine.
3. Formato del CD. Especifica en qué formato viene empaquetado el CD, si es una “*jewel case*”, suelto, una caja de cartón...
4. País de publicación. En qué parte del mundo se ha puesto a la venta.
5. Año de publicación. En qué año se ha publicado esta edición.
6. Número de canciones. Determina el número de canciones **del disco actual**.
7. Duración. Muestra **la duración del disco actual**. No puede exceder de 80 minutos (1:20:00).

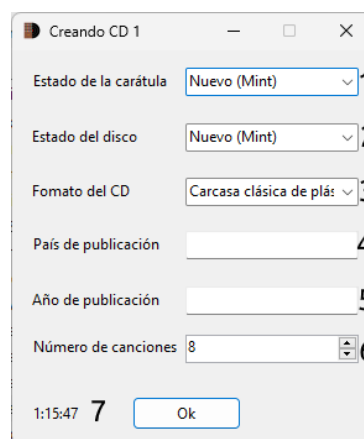



Ilustración 6.4-22 - Asistente para la creación de CD.

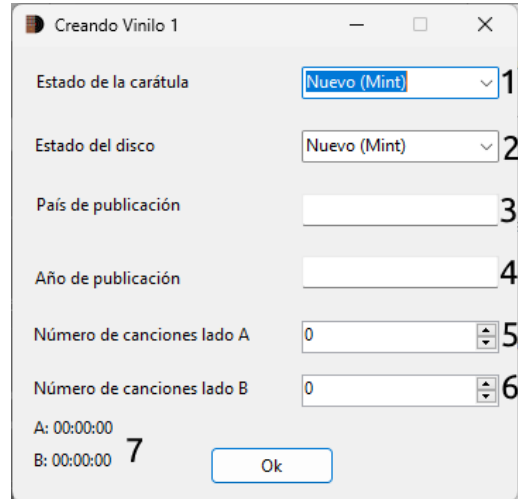
Si se pulsa **Ok** y el número de canciones elegidas es igual al total disponibles en el álbum, se cerrará el formulario y se creará el disco. En caso contrario, se preguntará si se desea crear otro disco y cuántas canciones quedan. Ocurrirá tantas veces mientras queden canciones disponibles, pudiéndose cancelar el proceso en cualquier momento y terminar de crear el disco.

6.4.4.2 Creación de un disco de vinilo.

Para crear un disco de vinilo se tomará el mismo procedimiento. Primero selecciona un álbum y luego presiona en el botón  de la barra de herramientas o haz clic derecho sobre él y luego a “*Crear disco de vinilo desde la selección*”.

Aparecerá una ventana como la mostrada a la derecha. Es parecida al asistente de creación de CD, pero con un campo más.

1. Estado de la carátula. Cómo está el disco de vinilo por fuera.
2. Estado del disco. Cómo está el propio disco y cómo se reproduce, si bien o mal.
3. País de publicación. En qué parte del mundo se ha puesto a la venta.
4. Año de publicación. En qué año se ha publicado esta edición.
5. Número de canciones del lado A. Cuántas canciones tiene el lado frontal.
6. Número de canciones del lado B. Cuántas canciones tiene el lado trasero.
7. Duración de cada lado para comprobar si la entrada proporcionada es correcta. Normalmente, no deben de exceder los 25 minutos por lado.



Creando Vinilo 1

Estado de la carátula: Nuevo (Mint) 1

Estado del disco: Nuevo (Mint) 2

País de publicación: 3

Año de publicación: 4

Número de canciones lado A: 0 5

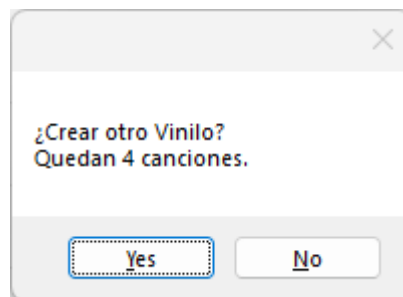
Número de canciones lado B: 0 6

A: 00:00:00
B: 00:00:00 7

Ok

Ilustración 6.4-23 - Asistente para la creación de un vinilo

Como ocurre antes con los CD, es posible que se necesite más de un disco. Si es el caso, aparecerá esta ventana de diálogo.



¿Crear otro Vinilo?
Quedan 4 canciones.

Yes No

Ilustración 6.4-24 - Ventana de diálogo preguntando si hace falta crear otro disco.

Si pulsamos que sí, se reabrirá el mismo formulario, pero con menos opciones. Sólo tendremos que introducir el estado del segundo disco y el número de canciones que dispone.

Como podemos apreciar en la figura, sólo aparecen tres opciones. No es necesario ser redundantes. Este proceso se irá repitiendo hasta que todas las canciones estén incluidas o el usuario pulse “no” a la hora de crear otro disco. Una vez tengamos creado el disco, podemos visualizar su información con el mismo procedimiento que hemos visto anteriormente, pero con la información ampliada.

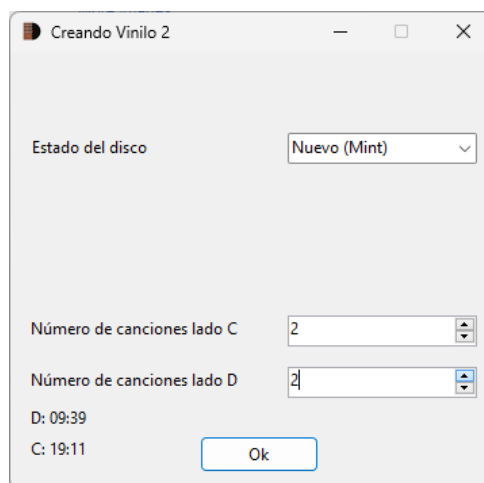


Ilustración 6.4-25 - Asistente de la creación de vinilos, en este caso con el segundo.

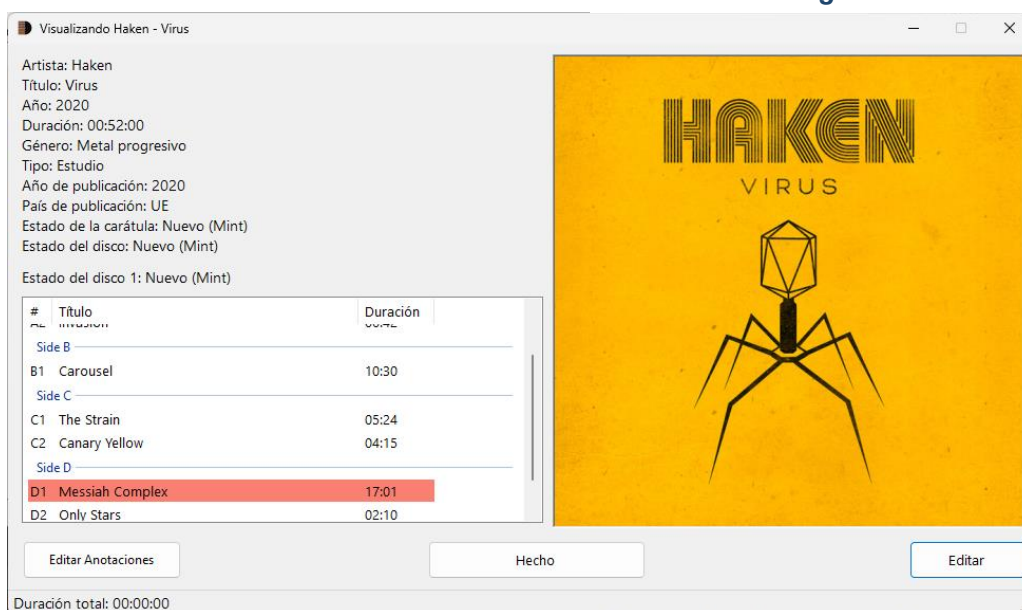



Ilustración 6.4-26 - Visualizando la información de un vinilo.

6.4.4.3 Creación de una cinta de casete

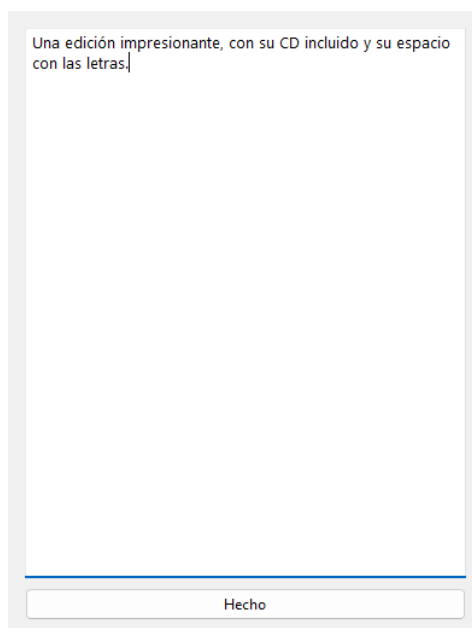
Para crear una cinta de casete, el proceso es prácticamente idéntico al de crear un disco de vinilo. Para acceder a la ventana basta con apretar en el botón  y seguir el mismo proceso explicado en el apartado anterior.

6.4.4.4 Ver y editar las anotaciones. Visualizar en la vista principal.

Si se ha fijado en la captura anterior, habrá visto un botón de “*Editar anotaciones*”. Es un botón que sólo está disponible si está visualizando un medio físico, no importa si es un vinilo o un CD. Si pulsa sobre él, se abrirá una ventana en blanco donde puede redactar lo que quiera sobre el artículo que está visualizando. No hay un límite de escritura y cada artículo dispone de sus anotaciones privadas.

Visualizar los CD y Vinilos en el sistema.

Para cambiar la vista principal, tan sólo debe de acudir al menú “Ver” -> “Cambiar de vista” y luego seleccionar el tipo de vista deseado.



**Ilustración 6.4-27 -
Ventana de
anotaciones**

6.4.5 Uso del reproductor integrado

Cassiopeia dispone de un reproductor integrado que es capaz de reproducir ficheros en MP3, FLAC y OGG. También puede reproducir CD de audio y, por último, sincronizarse con Spotify como si fuera un mando a distancia.

Para abrir el reproductor, basta con pulsar F11 en la ventana principal y veremos una ventana como esta:

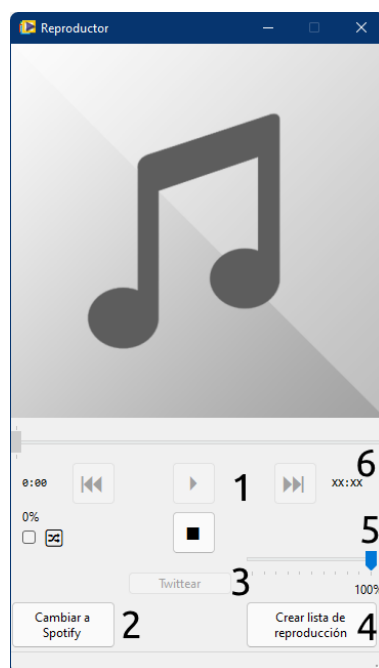


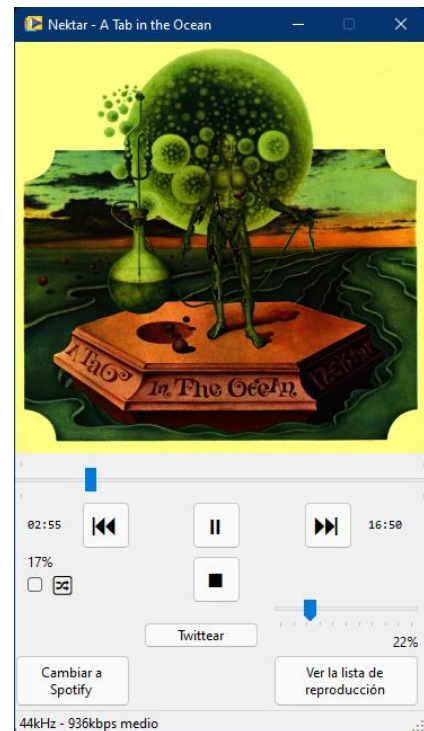
Ilustración 6.4-28 - Reproductor.

1. Botones de control. Permiten saltar canción, pausar, detener... etc.
2. Cambia entre el modo sincronización de Spotify o el local.
3. Abre la página de Twitter para redactar un tweet con la canción que se esté escuchando.
4. Crea una lista de reproducción
5. Volumen
6. Duración de la canción y si se pincha, se intercambia al tiempo restante de la canción. La barra superior es la barra deslizante de la posición de la canción.
7. A la izquierda, tenemos el botón de aleatorio, la marca de tiempo actual y el porcentaje de canción escuchada.

6.4.5.1 Reproducir una canción local (o varias)

Para reproducir una canción local (o varias) basta simplemente con arrastrar y soltar archivos compatibles a la ventana del reproductor. Si hay éxito, empezará a reproducir.

Como se puede apreciar, el botón de la lista de reproducción ha cambiado y ha pasado de “crear” a “ver”. Esto es, porque al reproducir una canción, se crea una lista de reproducción automáticamente para la comodidad del usuario. Si hacemos clic, podremos visualizar la propia lista de reproducción:



6.4.5.2 Manejando el gestor de listas de reproducción.

Ilustración 6.4-29 - Reproductor reproduciendo una canción

Podemos hacer varias cosas con esta ventana. Podemos cambiar el nombre de la lista, añadir canciones y eliminarlas. También, como es de suponer, podemos hacer doble clic en una canción para reproducir cualquiera a nuestro gusto.

Por último, para guardar la lista de reproducción, vamos al menú Archivo -> Guardar como.

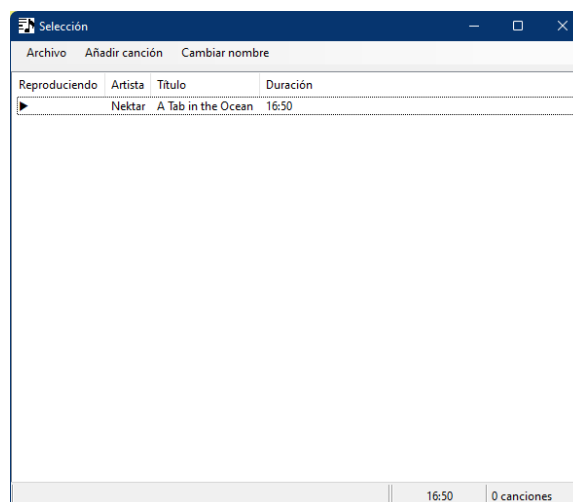


Ilustración 6.4-30 - Ventana de la lista de reproducción.

6.4.5.3 Sincronización con Spotify.

Como se ha mencionado anteriormente, una de las capacidades es sincronizarse con Spotify y en el caso de tener una cuenta premium, controlar la reproducción desde el programa. Para acceder a este modo, basta con pinchar en “Cambiar a Spotify”

Como podemos ver en la figura adjunta, no cambia mucho en la apariencia, pero desaparece el botón de la lista de reproducción y aparece un botón nuevo, que es de “Agregar a la BBDD”.

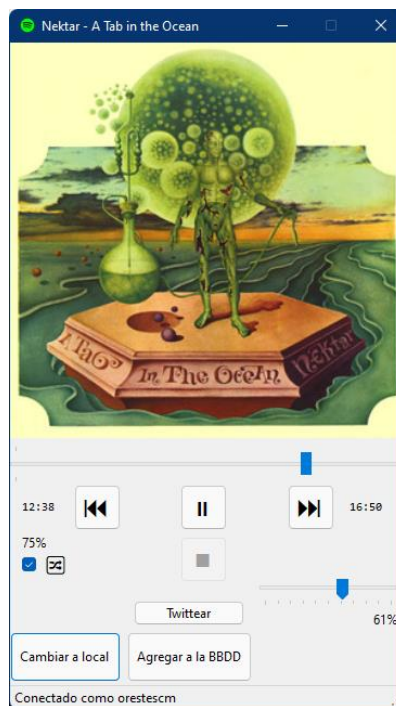
Este botón, añade el álbum de la canción que estás escuchando a la base de datos. Si ya estuviera insertado, lanzaría una advertencia y no seguiría con el proceso. En el caso de procesarse correctamente, observaremos una ventanita avisándonos de que todo ha ido bien.

La caja de aleatorio y las barras deslizantes de volumen y posición, están sincronizadas con Spotify. Si se cambian, se notarán inmediatamente.

NOTA: esto no es un *dispositivo* aparte de Spotify. Es un *mando a distancia*, es decir, sólo dicta órdenes a Spotify al dispositivo que se esté reproduciendo en ese momento. Ese dispositivo puede ser un teléfono con la aplicación, un ordenador o la propia página web de Spotify. Todos estos dispositivos sacan el sonido de la música. Cassiopeia no reproduce, dicta órdenes.

Como podemos observar, abajo sale que aparecemos conectados con nuestro nombre de usuario. Sirve para verificar que estamos con el usuario correcto.

Por último, si el usuario asociado no es de tipo *premium*, los botones de control se deshabilitarán ya que Spotify no permite estas peticiones si el usuario no es *premium*.



**Ilustración 6.4-31 -
Reproductor sincronizando
con Spotify**

6.4.5.4 Reproducir un CD de audio.

El Reproductor permite reproducir un CD de audio como un reproductor de la vieja escuela. Para eso, primero debemos dirigirnos al formulario principal y luego navegar al menú Archivo -> Abrir CD de audio.

Aparecerá una ventana donde saldrán las disqueteras que tenemos disponibles:

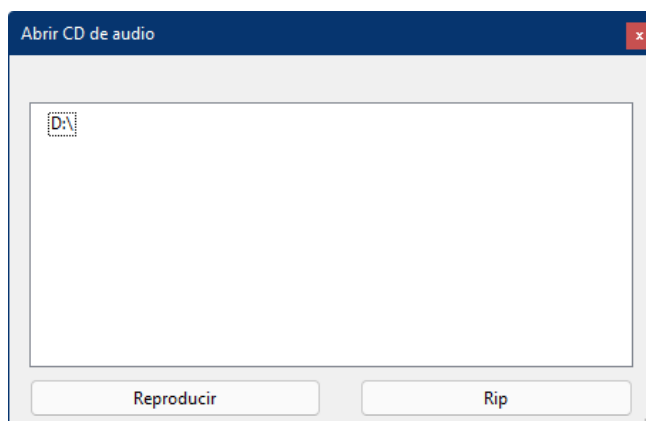


Ilustración 6.4-32 - Ventana para elegir disquetera.

Normalmente en los ordenadores sólo hay uno, pero en el caso de tener varios, selecciona el que vayas a usar. En mi caso, es la D:\.

Tenemos dos opciones, Reproducir o Rip. Rip saca las canciones del disco a ficheros digitales a nuestro disco duro.

Si pinchamos en reproducir, se cerrará la ventana y empezará a sonar la música y se creará una lista de reproducción nueva con las canciones del CD para que podamos saltar de una pista a otra.

6.4.6 Integración con Spotify

Como ya se ha mencionado anteriormente, este software tiene la capacidad de vincularse con su cuenta de Spotify, habilitando nuevas funcionalidades. La vinculación es realmente sencilla y rápida.

6.4.6.1 Vinculación de la cuenta

Para vincular su cuenta, diríjase al menú Spotify → Vincular con Spotify.

Aparecerá un mensaje de información diciendo que tiene 20 segundos para realizar la vinculación. Esto se hace para evitar bloqueos y volver al uso de la aplicación si por algún motivo la vinculación falla.

Si aceptamos, se nos abrirá el explorador con un enlace de Spotify pidiéndonos permisos que usará la aplicación. Estos permisos son:

- user-read-email
Permite saber el correo electrónico del usuario, para que el usuario observe que la autenticación ha ido correctamente.
- user-read-private
Con este permiso, el programa sabrá el tipo de cuenta que tiene el usuario, para distinguir si es Premium o Free, así se podrá usar la funcionalidad de *mando a distancia*.
- Streaming
Permite el control de Spotify si el usuario es Premium.
- user-read-playback-state
Permite leer el estado de reproducción del usuario en Spotify.
- user-library-read
Permite el acceso a la librería del usuario, los álbumes guardados.

Una vez realizada la vinculación, se nos abre la puerta a nuevas funcionalidades que usan nuestra cuenta de Spotify.

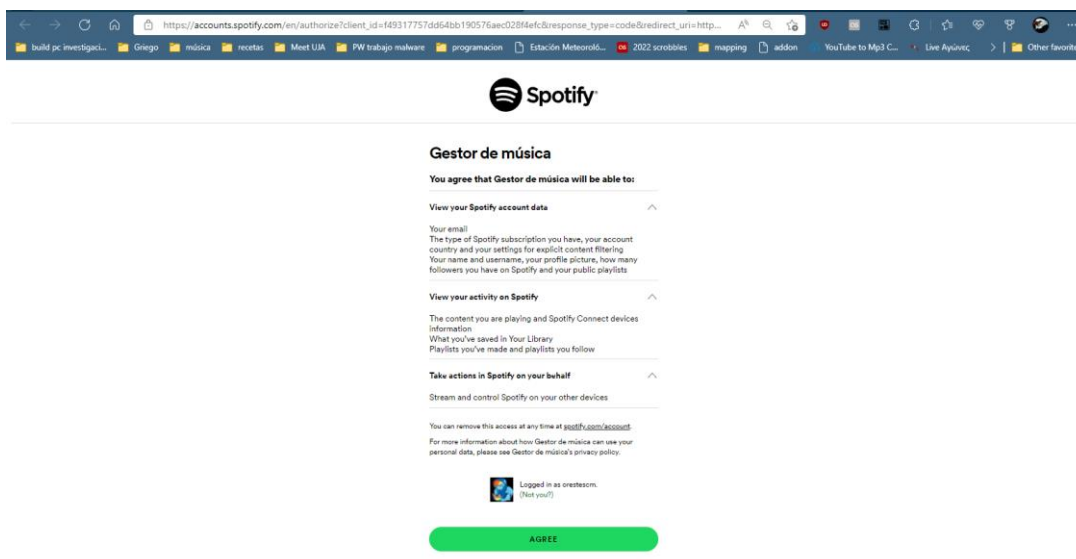


Ilustración 6.4-33 - Ventana para permitir los permisos en el navegador.

6.4.6.2 Importar álbumes desde su cuenta de Spotify

Con cierta probabilidad, eres un ávido escuchador de música y tengas guardados múltiples discos en tu cuenta de Spotify. Cassiopeia puede importar los metadatos de esos discos guardados para inicializar su base de datos en un tiempo mínimo.

Acceda al menú Spotify → Importar colección de álbumes.

Si tiene muchos, el programa le advertirá de que, si descarga las carátulas, puede demorarse un par de minutos, siempre dependiendo de su velocidad de conexión a Internet.

Nota: para realizar esta acción debe vincular antes su cuenta.

6.4.6.3 Insertar un álbum desde Spotify

Puede darse el caso de que el usuario no tenga Spotify y quiera insertar álbumes rápidamente. Existe una función que permite buscar álbumes en Spotify y posteriormente, descargar los metadatos.

Para acceder a este menú, vaya a Spotify → Buscar en Spotify.

Nota: para obtener la URI de un álbum de Spotify, debe de ir primero a la página del álbum, luego pinchar sobre los 3 puntos, luego en compartir, mantenga pulsado Control y aparecerá la opción de copiar la URI al portapapeles.

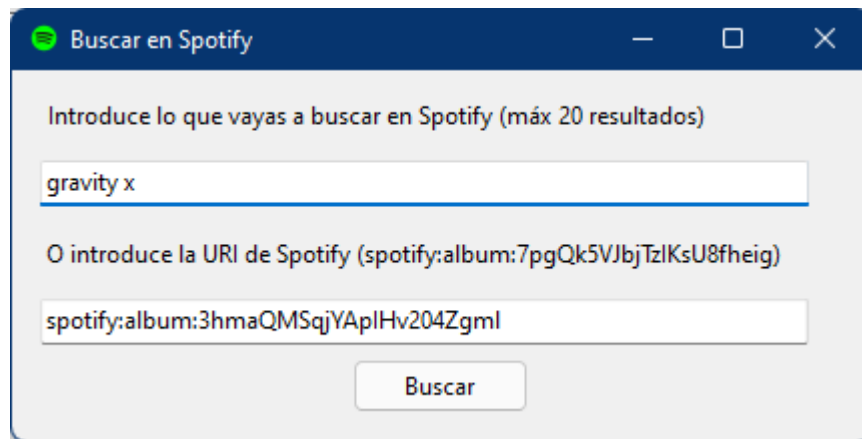


Ilustración 6.4-34 - Ventana para buscar un álbum en Spotify con datos de ejemplo.

Tendremos dos opciones y sólo podemos rellenar una. Aunque si rellena las dos, tendrá prioridad la búsqueda.

El menú de búsqueda abrirá otra ventana como la siguiente:



Ilustración 6.4-35 - Ventana con resultados de la búsqueda.

Simplemente seleccione los álbumes que le interese añadir al sistema y se añadirán completos, con la portada incluida.¹⁵

En cambio, podemos añadir una URI de Spotify. Es un código único que identifica al álbum con formato `spotify:album:uri`. Un ejemplo de este código único sería: `spotify:album:1jWmEhn3ggaL6isoyLfwBn`

¹⁵ Hay un fallo conocido y es que no se añade el género. Es por cómo funciona Spotify, que asocia un artista a varios géneros y, aunque sea válido, el software trabaja con la asociación 1 álbum 1 género.

Si añadimos un URI, el formulario se cerrará y tendremos el álbum insertado automáticamente.

6.4.6.4 Ordenar la reproducción de un álbum desde Cassiopeia a Spotify.

Si tiene una cuenta Premium, puede aprovecharse de esta opción. Para reproducir un álbum en Spotify desde el programa, basta con dar clic derecho al álbum en la lista principal y luego en “Reproducir en Spotify”. Sólo funcionará si está configurado el URI correctamente.

También se puede ordenar la reproducción de una canción arbitraria. En la vista de álbum haga clic derecho en “Reproducir en Spotify”.

NOTA: sólo funcionará en el caso de que haya un dispositivo Spotify disponible, estos pueden ser una computadora, un teléfono con la aplicación instalada, una televisión inteligente o un *Chromecast*, entre otros. Si no es el caso, no ocurrirá nada.

6.4.6.5 Desvincular Spotify de la aplicación

Para desvincular la aplicación de Spotify debemos de ir al siguiente enlace: <https://www.spotify.com/es/account/apps/> y luego buscar la entrada “Gestor de música”. Por último, pinchamos en el botón “retirar acceso”. El programa automáticamente detectará esta desvinculación y volverá a su *estatus* anterior.

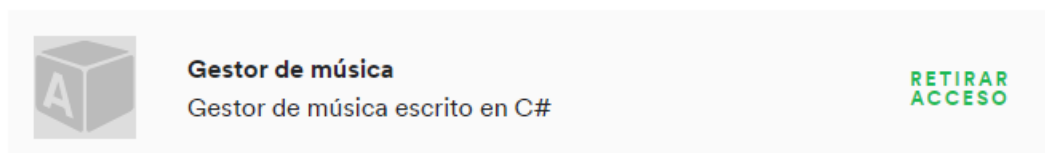


Ilustración 6.4-36 - Entrada de una aplicación vinculada en Spotify

6.4.7 Configuración

Cassiopeia dispone de un apartado de configuración para cambiar y personalizar distintas funcionalidades de la herramienta. Entre estas opciones, se incluyen:

- Idioma de la aplicación
- Portapapeles
- Historial
- Cadena de retransmisión
- Visualización (colores y tipografías)

Cambiar el idioma.

Es la primera opción que aparece en el árbol de opciones a mano izquierda.

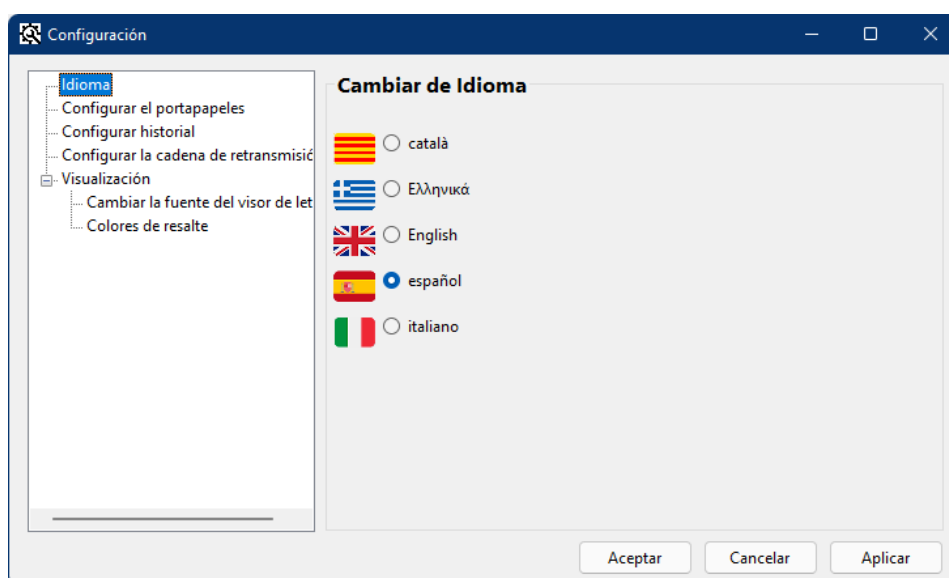


Ilustración 6.4-37 - Opciones de idioma con una selección disponible durante el desarrollo de la aplicación.

Para cambiar el idioma, seleccione un idioma entre los disponibles y luego presione en Aceptar o en Aplicar. No requerirá de reinicio y la aplicación cambiará los textos inmediatamente.

Portapapeles

Permite configurar el texto que copiará al portapapeles cuando copia un álbum desde la vista principal. Contiene una selección de filtros.

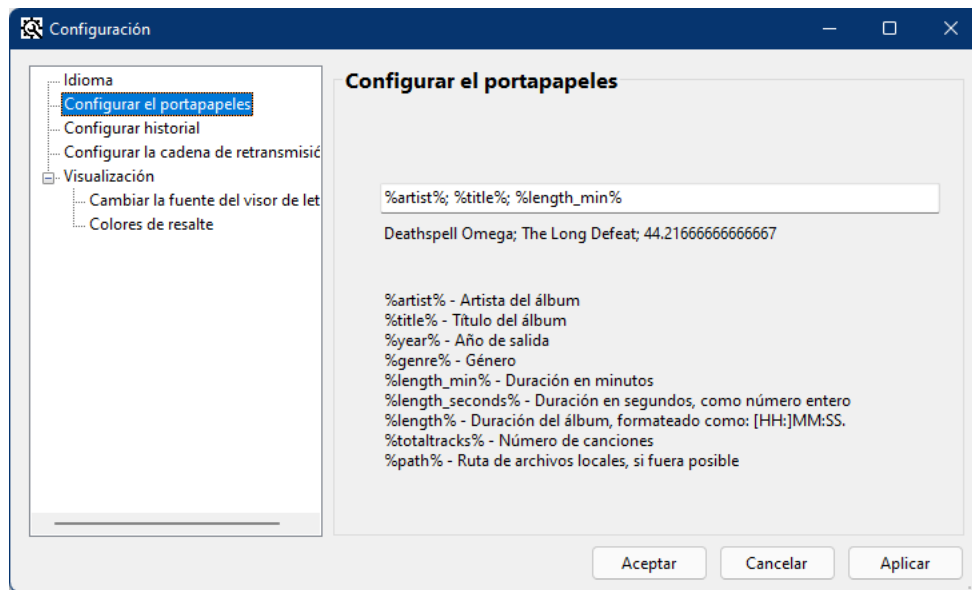


Ilustración 6.4-38 - Configuración del portapapeles

Historial

Cassiopeia puede registrar un historial de las canciones que el usuario haya escuchado tanto localmente como en Spotify. Se puede configurar su activación o el formato de la salida. Se puede modificar cada entrada del historial.

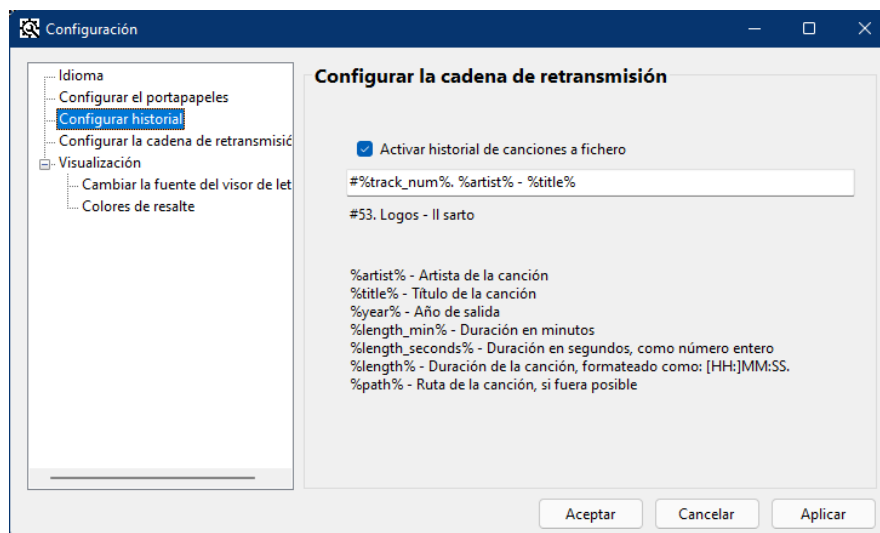


Ilustración 6.4-39 - Configuración de la cadena del historial

Cadena de retransmisión

Es posible configurar también esa salida a tiempo real. Es útil para mostrar a tiempo real qué está escuchando el usuario, como, por ejemplo, una salida a una retransmisión en directo. Esa salida se guardará a un fichero llamado np.txt en el directorio raíz del programa. Se actualiza cada segundo.

Un ejemplo de salida de cadena sería:

#1. Deep Purple - Super Trouper - Remastered (01:27)

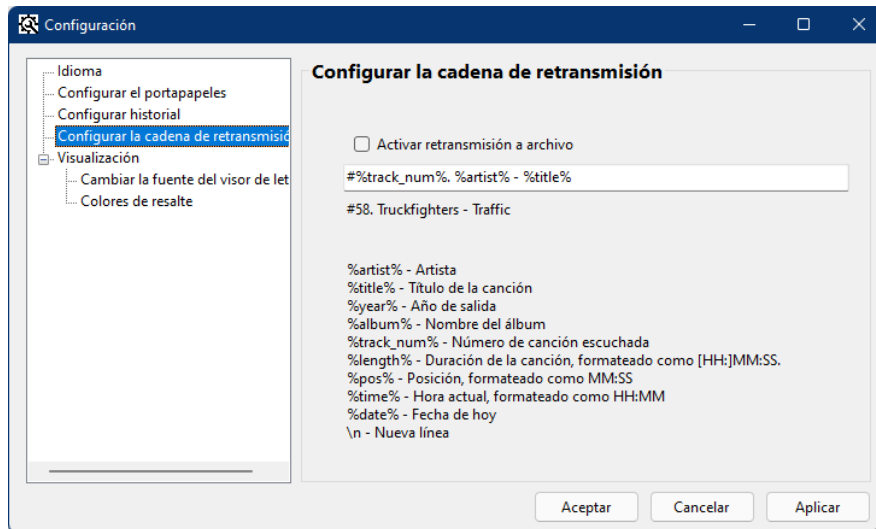


Ilustración 6.4-40 - Ventana de personalización para la cadena de retransmisión

Visualización

Por último, se puede cambiar la visualización del programa, empezando por el tipo de letra.

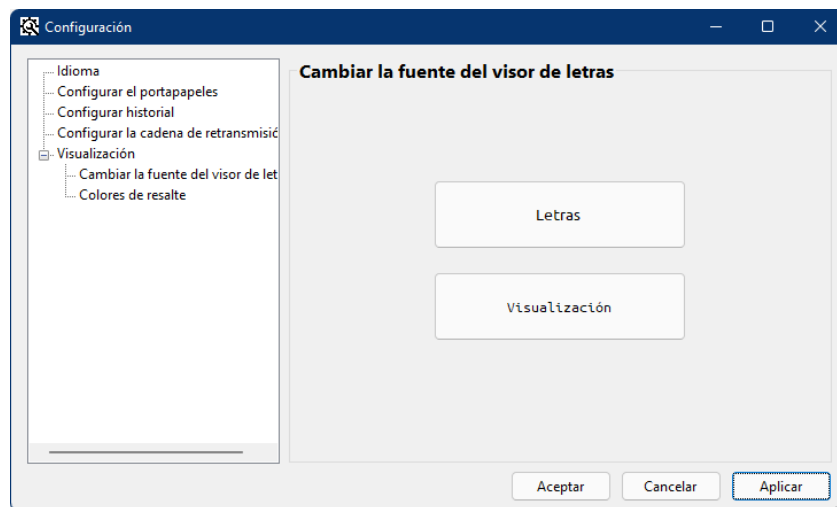


Ilustración 6.4-41 - Botones para configurar el tipo de letra de la aplicación. Tanto en el visor de letras como en las tablas.

Los colores de resalte que se pueden configurar son cuando una canción es bonus o contiene partes. Se abrirá un selector de colores familiar y podrá elegir el color que más le guste.

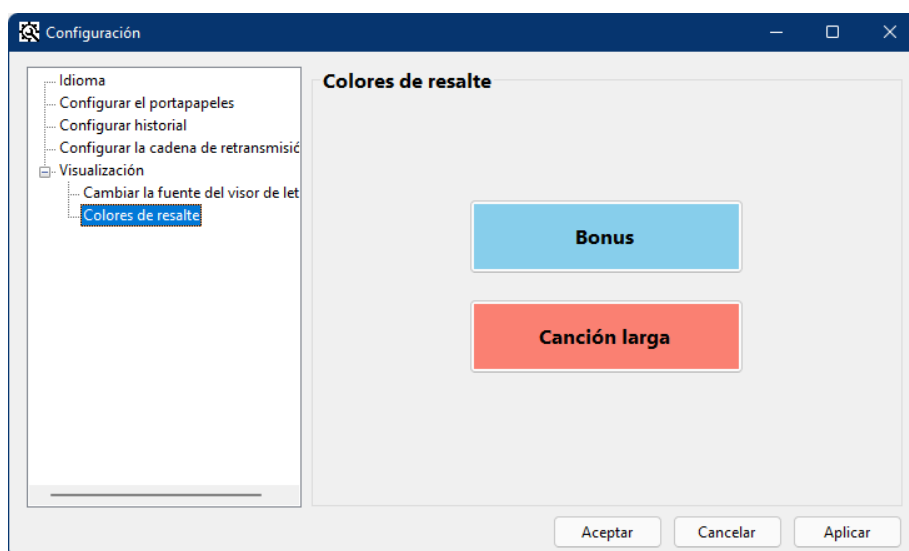


Ilustración 6.4-42 - Ventana para cambiar los colores con los valores predeterminados

6.4.8 Anexo: Estándar Goldmine

El programa usa el estándar *de facto Goldmine* para clasificar los discos en función de su estado, así creando una medida objetiva a la hora de vender artículos en un mercado online, para que el comprador sepa de antemano en qué estado se encuentra el objeto y así evitar sorpresas de última hora. En este apartado, se recogerá el significado de cada uno de las siglas.

- **Mint (M)**

Se trata de un artículo nuevo, posiblemente sin abrir y / o estrenar, incluso todavía contiene el plástico envolvente para protegerlo. Esta calificación debe usarse con suma cautela, si es que se debe de usar acaso.

- **Near Mint (NM o M-)**

Es un artículo con muy poco uso, apenas hay rasgos de desgaste. Es seminuevo, con un parecido a un coche Km 0 en los concesionarios. Un disco casi perfecto. Es muy probable que apenas tenga uso y se reproducirá sin imperfecciones. No debe mostrar signos obvios de desgaste ni huellas. La portada no debe tener ni pliegues, ni defectos visibles. Todo debe venir incluido.

- **Very Good Plus o Excellent (VG+ o E)**

Suele valer un 50% menos que NM. Es un artículo que está en excelente estado, está usado, pero ha tenido mucho mimo por parte del viejo vendedor. En cuanto a los vinilos, la superficie puede mostrar algún signo de desgaste y se pueden escuchar algunas rozaduras, pero no afecta a la experiencia.

- **Very Good (VG)**

Vale el 25% del valor de NM. No tiene la *frescura* de un disco nuevo de fábrica, es obvio que ha sido usado, pero no está demasiado desgastado. El artículo puede tener escritos.

Vinilo.

Todos los defectos encontrados en un artículo VG+ serán más pronunciados. Se notará más el ruido en pasajes suaves y el desgaste del surco será más audible, pero no dominará a la música.

- **Good (G)**

Estos artículos costarán en torno a un 20% menos que un artículo NM. No es que sea malo, está en buen estado, pero es claramente obvio que es de segunda mano.

Vinilo

Tendrá claros rasgos de uso y desgaste, se reproducirá sin saltos. Se oirá ruido y rasguños en la superficie. Si es un disco común puedes intentar encontrar algo en mejores condiciones, pero si tienes prisa... no es mala opción.

CD.

Hay muchos desgastes y rayones, pero se reproduce sin ningún problema. El dueño anterior no ha tenido mucho cuidado. Los libretos puede que estén rotos y muy escritos, con poco cuidado, e incluso, una jewel case rota.

- **Fair (F)**

Vale el 5% del precio NM.

Vinilo.

Está en mal estado, pero está justo que llega a ser aceptable mantenerlo en la colección. El vinilo tendrá bastantes *pops* y ruido de polvo. Puede que incluso esté deformado (*warped*) y la carátula puede estar dañada por humedades.

CD.

El CD no puede reproducirse bien. Tendrá saltos y algunas canciones no podrán oírse. Lo mejor es contactar con el vendedor. La portada está muy desgastada y dañada.

- **Poor (P)**

Este artículo se encuentra en un estado deplorable y apenas tiene valor monetario. El vinilo será casi irreproducible, con muchas probabilidades de estar deformado y con muchos ruidos, arañazos y suciedades. Los CD no se reproducirán y será muy difícil obtener una copia en condiciones. Los libretos con las letras estarán con toda probabilidad rotos. Es raro ver un artículo en estas condiciones, ya que es imposible venderlo.

7 DEFINICIONES Y ABREVIATURAS

CTRL: LA TECLA CONTROL.

PREMIUM: ES UN TIPO DE SUSCRIPCIÓN A SPOTIFY QUE DESBLOQUEA CIERTAS VENTAJAS, SIENDO UNA DE ELLAS QUE LA REPRODUCCIÓN PUEDA SER CONTROLADA POR APLICACIONES EXTERNAS.

HIFI: HIGH FIDELITY, ALTA FIDELIDAD EN SONIDO. REFERIDO A LA CALIDAD DE SONIDO DE UN DISPOSITIVO O DE UN FICHERO, PUEDE O NO ESTAR COMPRIMIDO. POR REGLA GENERAL, UN CD ES AUDIO HIFI.

FLAC: FREE LOSSLESS AUDIO CODEC, UN CÓDEC DE AUDIO DE COMPRESIÓN SIN PÉRDIDAS.

JIT: JUST IN TIME

IU: INTERFAZ DE USUARIO

CSV: COMMA SEPARATED VALUES.

TOC: TABLE OF CONTENTS, UNA TABLA QUE SE ENCUENTRA EN EL INTERIOR DE LOS CD E INDICA TODAS LAS PISTAS Y DÓNDE SE ENCUENTRAN FÍSICAMENTE EN EL DISCO.

LINQ: LANGUAGE-INTEGRATED QUERY

WASAPI: WINDOWS AUDIO SESSION API, PERMITE A LAS APLICACIONES CLIENTE ADMINISTRAR EL FLUJO DE DATOS DE AUDIO ENTRE LA APLICACIÓN Y UN DISPOSITIVO DE PUNTO DE CONEXIÓN DE AUDIO.

WAV: WAVEFORM AUDIO FORMAT

MOD: FICHERO MÓDULO. ES UN FORMATO ORIGINARIO DE LAS COMPUTADORAS AMIGA.

SPC: ES UN FORMATO DE SONIDO QUE REPRESENTA MÚSICA DE LA SUPER NINTENDO, DADO QUE EL CHIP DE SONIDO SE LLAMA SPC700.

8 BIBLIOGRAFÍA

- Albahari, J. (s.f.). *C# 10 in a Nutshell*. O'Reilly Media, Inc.
- Ben Albahari, J. A. (2008). *LINQ Pocket Reference*. O'Reilly Media, Inc.
- Cleary, S. (2019). *Concurrency in C# Cookbook*. O'Reilly Media, Inc.
- Li, Z.-N., Drew, M., & Li, J. (2014). *Fundamentals of Multimedia*. Springer.
- Pressman, R. (2010). *Ingeniería del Software*. McGraw-Hill.
- Price, M. J. (2021). *C# 10 and .NET 6 - Modern Cross-Platform Development*. Packt Publishing.
- Sells, C. (2003). *Windows Forms Programming in C#*. Addison-Wesley Professional.
- Williams, T. (2020). *C# Console and Windows Forms Development with LINQ and ADO.NET*. Packt Publishing.

8.1 Sitios web utilizados

- Visual Studio
<https://visualstudio.microsoft.com/es/vs/community/>
- Documentación de Windows Forms
<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>
- CSCore
<https://github.com/filoe/cscore>
- TagLib Sharp
<https://github.com/mono/taglib-sharp/>
- SpotifyAPI
<https://github.com/JohnnyCrazy/SpotifyAPI-NET>
- Sitio oficial de Pitchfork
<https://pitchfork.com/>
- Sitio oficial de Discogs
<https://www.discogs.com/>

- Información sobre Spotify Premium
<https://www.spotify.com/es/premium/>
- BOE usado para los salarios
[Disposición 3156 del BOE núm. 57 de 2018](#)
- Documentación de WASAPI
<https://docs.microsoft.com/es-es/windows/win32/coreaudio/wasapi>
- Sitio oficial de Foobar2000
<https://www.foobar2000.org>
- Explicación del estándar Goldmine
<https://support.discogs.com/hc/es/articles/360001566193-Como-calificar-art%C3%ADculos>
- Elección de la forma jurídica de la empresa
<http://www.ipyme.org/es-ES/DecisionEmprender/FormasJuridicas/Paginas/FormasJuridicas.aspx>
- Información sobre el formato de audio SPC
<http://www.vgmpf.com/Wiki/index.php?title=SPC>
- Información sobre los ficheros de audio MOD
https://en.wikipedia.org/wiki/Module_file
- Información sobre generar una URL para lanzar un *tweet*
<https://developer.twitter.com/en/docs/twitter-for-websites/tweet-button/guides/web-intent>
- Documentación de la API de Spotify
<https://developer.spotify.com/documentation/web-api/>
- Sitio oficial de Deezer
<https://www.deezer.com/es/>
- Sitio oficial de TIDAL
<https://tidal.com/>
- Sitio oficial de YouTube Music
<https://music.youtube.com/>

- Sitio oficial de Apple Music
<https://music.apple.com/es/browse>
- Sitio oficial de Amazon Prime Music
<https://www.amazon.es/music/prime>
- Sitio oficial de Winamp
<https://www.winamp.com/>