UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA


# Informe Proyecto

## Easy BPM


| **Estudiante** | **ROL** |
|---|---|
| Josschua Machuca | 202030005-4 |
| Daniela Stuven | 202030014-3 |
| Diego Tobar | 201930050-4 |
| Javier Zamora | 202004084-2 |


**Asignatura**
Diseño y Programación Orientados a Objetos - ELO329

| **Profesor** | **Paralelo** |
|---|---|
| Agustín González | 1 |

**Fecha**
6 de julio de 2023

# Índice

# 1. Descripción del problema

Este proyecto se basó en la problemática para encontrar los BPM (o beats por minuto) de una o varias canciones y aparte, tenerlas guardadas en un archivo de forma ordenada.
Esto se debe a la complejidad que existe para calcular este dato manualmente y los pocos repositorios que se encuentran en internet, por lo tanto, la gente debe realizarlos manualmente buscando la información necesaria en un lugar (internet) y almacenándola en otro (ya sea Excel o un archivo de texto simple).

# 2. Análisis del problema

En muchas áreas relacionadas con el ámbito musical es muy importante tener la cantidad de beats que tiene una canción en cierta cantidad de tiempo ya que esto indica la velocidad de la canción. Esto sirve desde la coordinación de un grupo musical hasta ayudar a un DJ a calcular en que momento debe poner la nueva canción para que el ritmo siga sin romper el tempo.

A pesar de que esta información se puede sacar de forma manual, requiere mucha experiencia y habilidad auditiva y de reacción, pues necesita que el oyente siga los pulsos de una canción al mismo tiempo que va contando por cierto periodo de tiempo para luego sacar la media aproximada para un minuto. Esto, por la complejidad, lo vuelve muy inexacto al depender de factores externos y humanos. También, se pueden encontrar páginas de internet con bases de datos de distintas canciones que pueden llegar a ser útiles pero aparte de tener que buscar canción por canción y ser poco confiables ya que distintas páginas tienen distintos valores para cada una de éstas, se debe buscar una forma para almacenar la información obtenida.

# 3. Diseño de solución

Para solucionar los problemas anteriormente mencionados, se realizó un programa escrito en lenguaje C++ con distintas funciones que facilitan la obtención, organización, accesibilidad y visualización de distintas canciones con sus respectivos BPM.

Entre las funciones principales del programa se encuentran obtener el BPM de forma automática de un archivo de audio seleccionado por el usuario, añadir el BPM obtenido, junto con la información pertinente de la canción, al archivo de texto que contiene la información de las canciones que se despliega en pantalla,y la eliminación del archivo de canciones indicando su numero de indexación en la pantalla. Cabe destacar, que para agregar una canción es necesario rellenar todos los campos (BPM, artista y nombre canción) o este entrega un mensaje de error indicando que el ingreso fue inválido. Entre las funciones menos importantes y menos mencionadas dentro de este informe, se encuentra un botón para filtrar el archivo según BPM, artista o canción y su respectivo botón para eliminar el filtro.

En cuanto a la obtención de los BPM, el principal enfoque fue realizar todo el desarrollo en C++, sin embargo, no encontramos librerías/códigos que nos permitieran realizar esto fácilmente, por lo que buscamos otros lenguajes que nos fueran útiles. Python, a través de una librería externa de libre uso [1] nos permitió obtener los BPM de una canción , ya sea, en formato wav o mp3, y además, permite realizar una grabación con el micrófono y crear un archivo mp3 a partir de la grabación.

Posteriormente, intentamos ejecutar el script de Python dentro de C++, sin embargo, tampoco logramos hacerlo de forma exitosa, por lo que la solución a este problema fue realizar un programa en Python que se ejecute en paralelo con el programa principal que se comunicara con este a través de archivos .txt para

obtener el nombre del archivo del cual se obtendrán los BPM y posteriormente colocar en otro .txt el valor obtenido para que el programa principal lo lea y lo coloque en la tabla con su artista y nombre de canción correspondiente. Una vez se cierra la ventana principal, también se cerrará el script de Python automáticamente.

# 4. Definición de requerimientos

Esta sección juega un papel crucial al comprender los casos de uso de nuestro proyecto y determinar el carácter funcional necesario para garantizar su desarrollo exitoso.
Nos adentraremos en una descripción detallada de los casos de uso clave. Estos detalles servirán como guía para entender como funciona el programa y los distintos usos que se le puede dar.

## 4.1. Obtención de BPM Manual

La primera funcionalidad que incluimos en nuestro proyecto fue la opción de obtener manualmente los BPM de una canción a través de la pulsación de un botón al ritmo de una canción.
Este caso está especialmente pensado para quienes no tienen un archivo de audio o la posibilidad de prender micrófono para que el programa calcule esta información automáticamente. Este método puede causar problemas de exactitud ya que depende de factores externos al programa, como la capacidad del oyente de reconocer las pulsaciones y/o su capacidad de "clickear" en el momento correcto.
El usuario notará que ha conseguido los BPM de la canción cuando los valores que aparezcan al lado del botón se estabilicen.

## 4.2. Obtención de BPM Mediante archivo de audio

Esta función es especial para los casos en donde se tiene un archivo de audio (ya sea .mp3, .wav o .m4a) y necesitamos los BPM de la canción que contiene.
Al presionar el botón "ADD SONG (AUTO BPM INPUT)" el programa comienza a correr un programa python (que es el que calcula los BPM), abre el explorador de archivos del usuario para que seleccione el archivo de la canción y el programa copie este archivo en el directorio para luego ser utilizado en el archivo .py obteniendo la información solicitada. Finalmente guarda esta información en el archivo con los datos ya guardados y solicita información extra como el artista y el nombre de la canción para terminar el formato necesario para añadir una canción al archivo.

## 4.3. Obtención de BPM mediante micrófono

Es una función creada para poder detectar a través del micrófono un audio y luego lo guarda como .mp3 para poder analizarlo y devolver los BPM al programa la información necesaria para poder guardar dentro del archivo de texto que se muestra en la interfaz con un sistema de casillas separando BPM, artista y nombre de la canción.
Este método es ideal para los casos en que el usuario no tiene el archivo de la canción, no lo quiere hacer de forma manual pero si puede reproducir la canción desde un dispositivo externo.

# 5. Diagramas

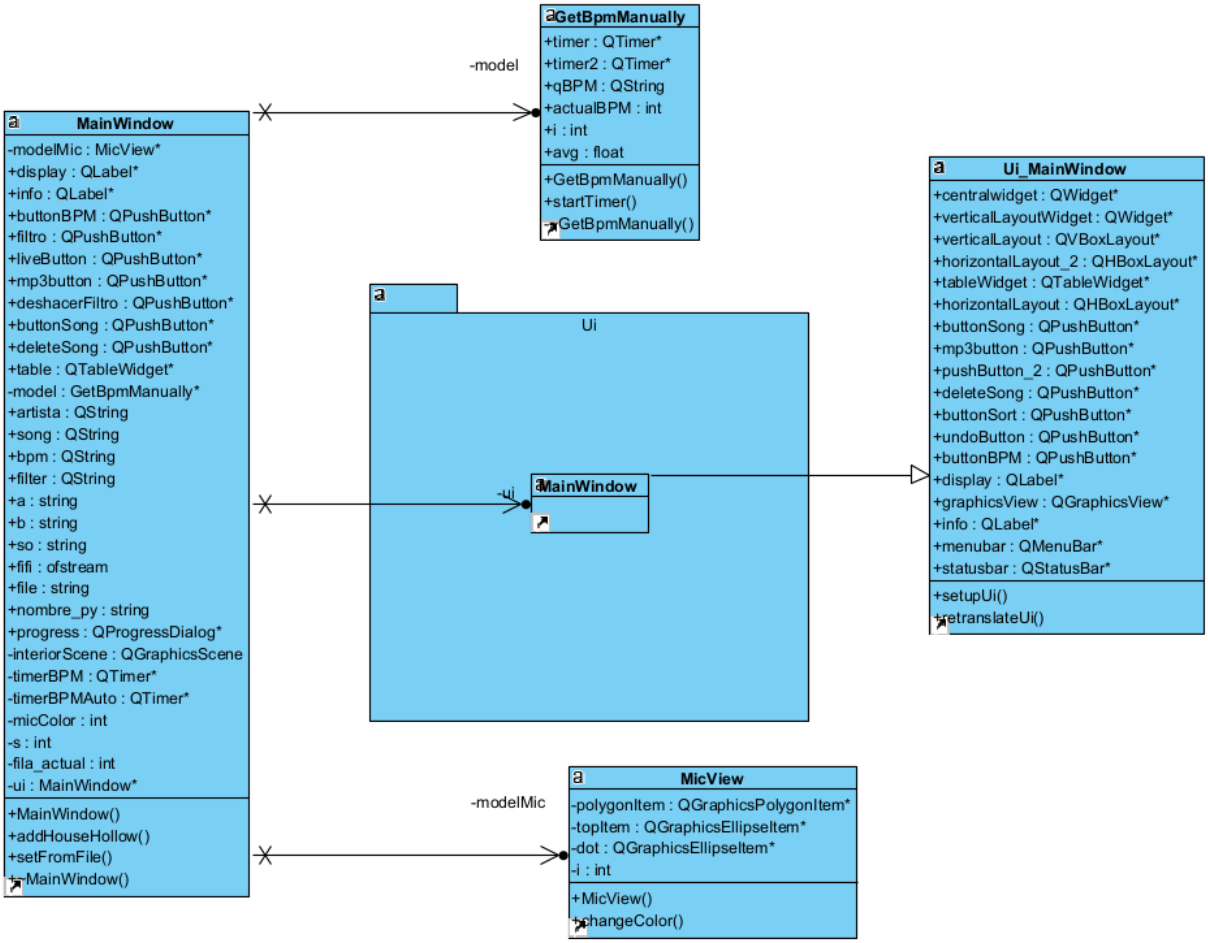



Figura 1: Diagrama UML de clases

## 5.1. Diagrama Caso de uso 4.2

**Nombre**: Obtener BPM mediante archivo de audio
**Propósito**: Verificar el correcto funcionamiento de la interfaz en conjunto con la lógica utilizada para obtener los BPM
**Actores**: Usuario, Sistema.
**Pre-condiciones**: El archivo de audio seleccionado esta en formato mp3 o wav.
**Post-condiciones**: El sistema obtiene correctamente los bpm y el usuario al ingresar artista y nombre de la canción se registra exitosamente en el sistema.
**Evento**: Usuario presiona el boton ADD SONG (AUTO INPUT)

| Usuario | Sistema |
|---|---|
| 1. Selecciona "ADD SONG (AUTO INPUT) | 2. Abre explorador de archivos |
| 3. Selecciona canción | 4. Obtiene BPM |
| | 5. Abre ventana dilogo |
| 6. Completa información necesaria | 7. Guarda os datos en el archivo |

Figura 2: Curso normal de eventos

| Usuario | Sistema |
|---|---|
| 1. Selecciona "ADD SONG (AUTO INPUT)" | 2. Abre explorador de archivo |
| 3. Seleciona canción | 4. Obtener el BPM |
| | 5. Abre ventana de dialogo |
| 6. No ingresa la informacion | 7. Ingreso invalido |
| | 8. Regresa a la interfaz principal |

Figura 3: Curso alternativo de eventos

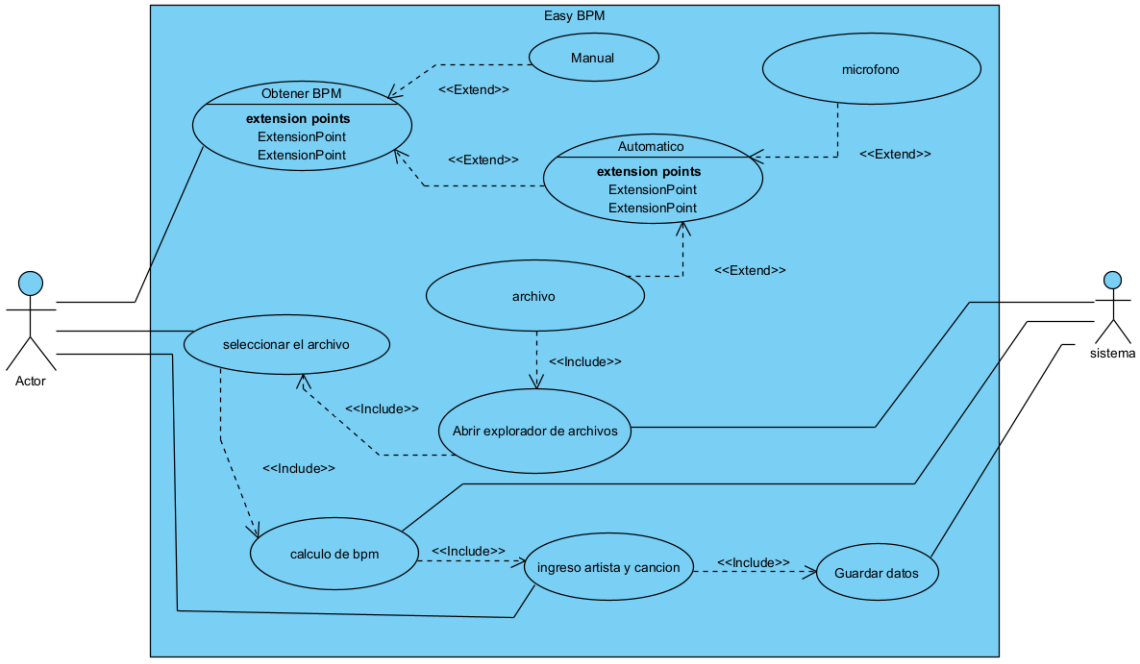



Figura 4: Diagrama UML de Secuencia

# 6. Resultados

## 6.1. Caso Uso: BPM manual

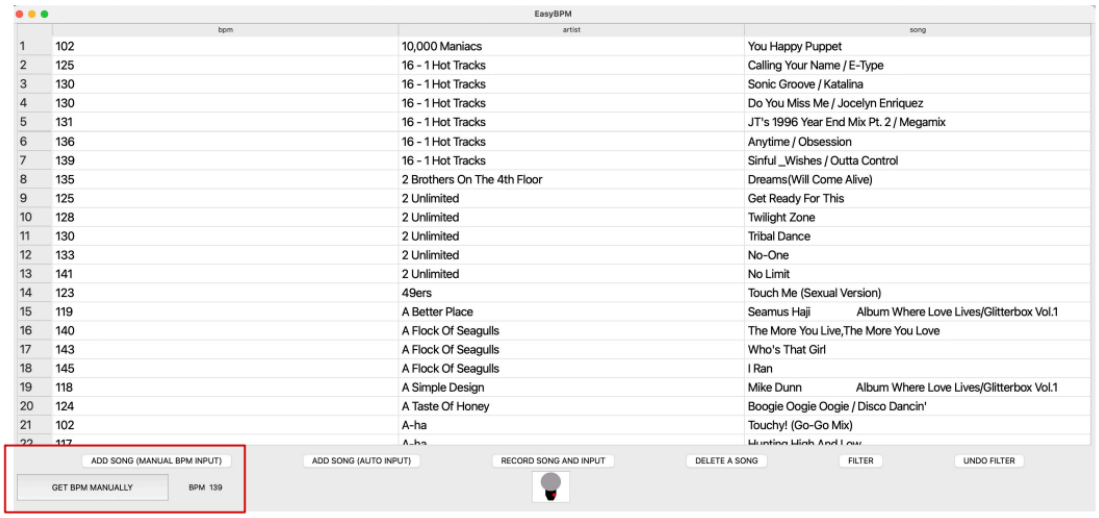



Figura 5: Página principal programa indicando sector GET BPM MANUALLY

Se obtienen los bpm manualmente con el botón indicado en Figura 4, se presiona ADD SONG (MANUAL BPM INPUT). Luego, se abren la ventanas de dialogo.

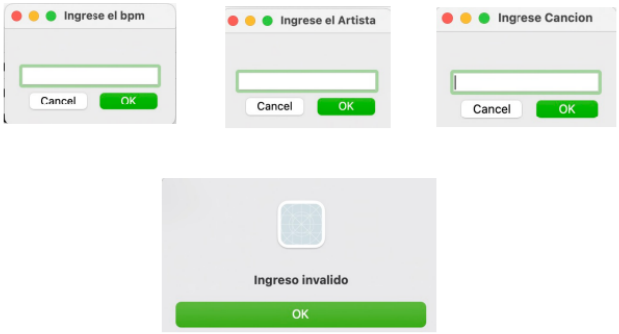



Figura 6: Ventanas de diálogo para ingresar datos/ingreso inválido

Se rellenan las ventanas de diálogos con los datos solicitados, si existe algún campo en blanco, la operación sera invalida y no se agregara a la tabla. En caso de completar los campos correctamente el resultado de la operación sera a la que se muestra en la Figura 7.

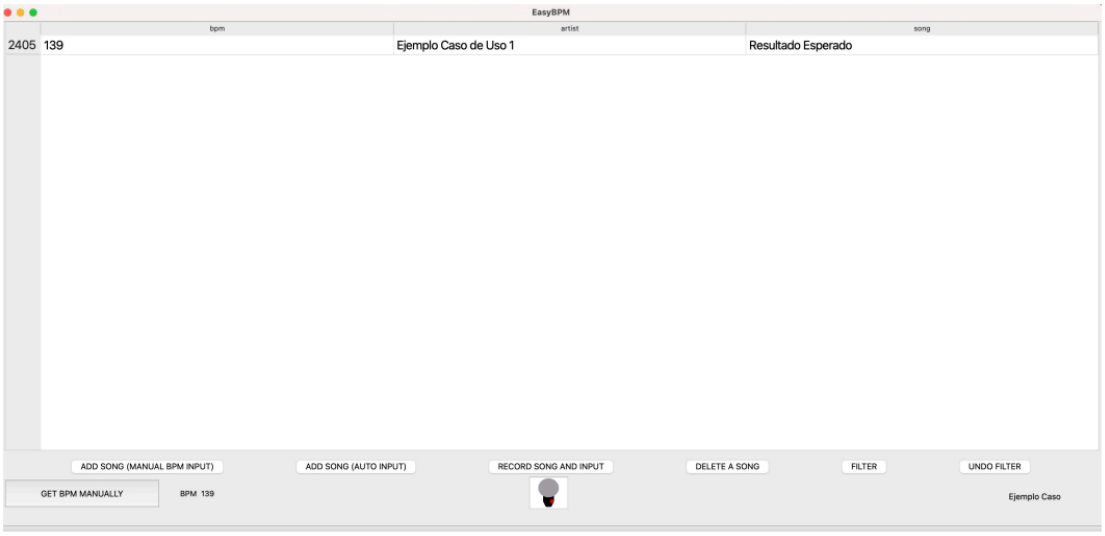



Figura 7: Resultado de BPM manually e ingreso de datos al archivo

## 6.2. Caso de Uso: BPM Automático (Mediante archivo)

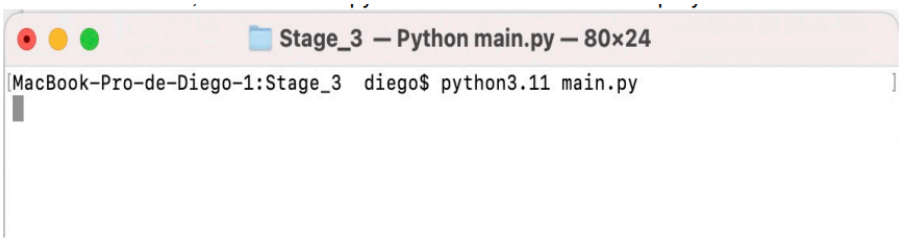



Figura 8: Terminal, ejecutando main.py ubicado en directorio del proyecto

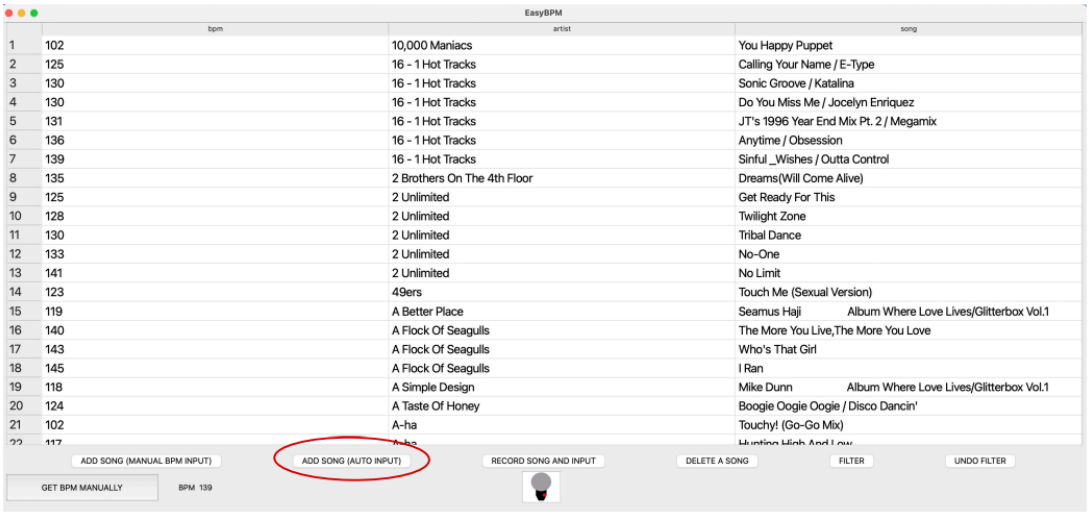



Figura 9: Página principal programa indicando sector ADD SONG (AUTO INPUT)

Al seleccionar esta opción, se abrirá el administrador de archivos, donde el usuario tendrá que elegir algún archivo de con extinción mp3, mp4 o wav. Ver figura 10.

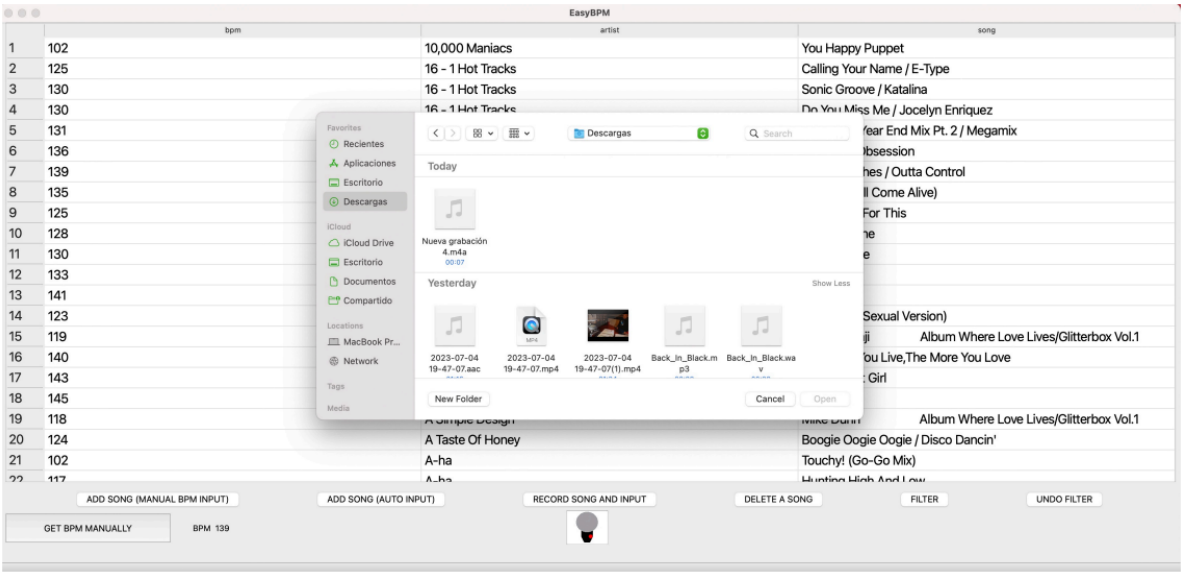



Figura 10: Administrador de archivos del sistema operativo

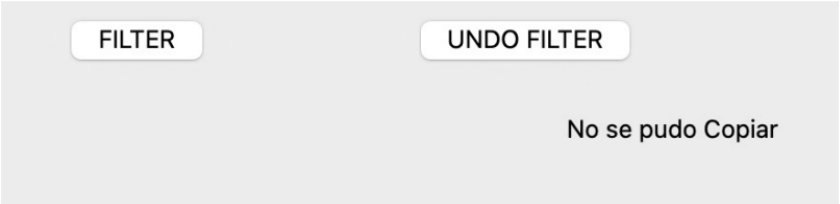



Figura 11: Página principal del programa

En el caso de que el archivo ya se utilizó, es decir, ya se copió, se debe reiniciar el programa y volver a ejecutar el main.py. Ver Figura 11.

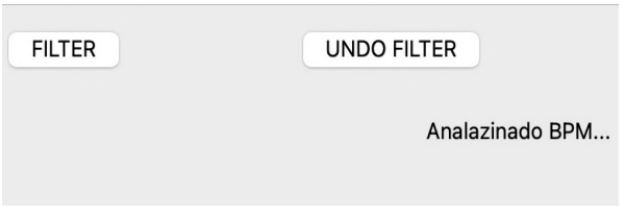



Figura 12: Página principal del programa

En el caso de que el archivo no se ha analizado anteriormente, se mostrará por pantalla que el bpm esta siendo analizado por el .py. Ver Figura 12. Pasando el tiempo de espera del análisis se abrirán pestañas de diálogos, preguntando sobre el nombre del artista y la canción tal como se muestra en la Figura 13.

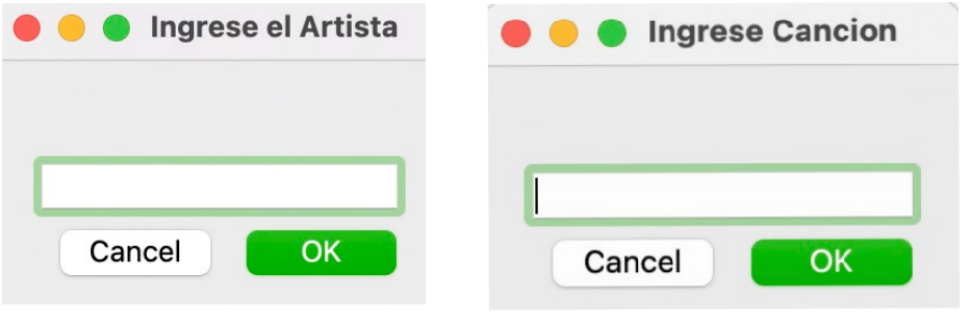



Figura 13: Ventanas de diálogos

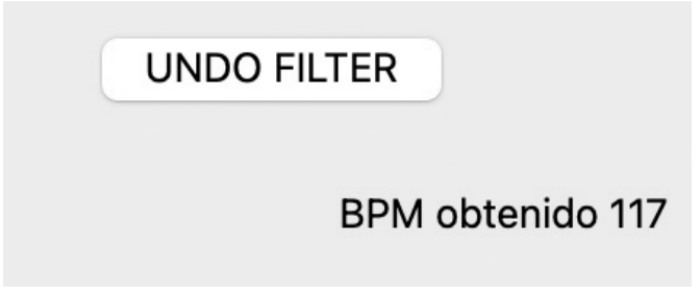



Figura 14: Página principal del programa, Muestra obtención del BPM desde python

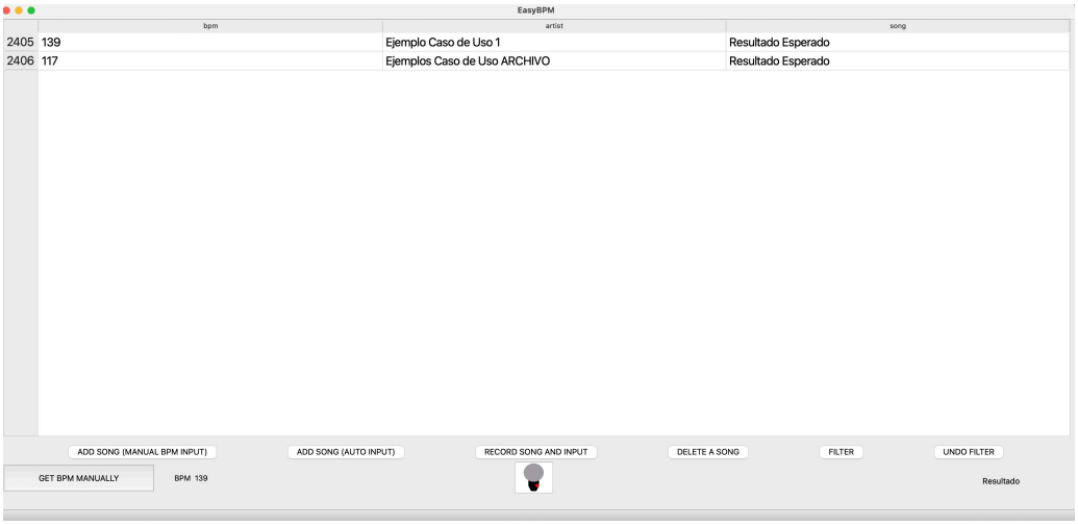



Figura 15: Resultados de la Obtención del bpm por medio de un archivo de audio

### 6.3. Caso de Uso: BPM automático (Mediante micrófono)

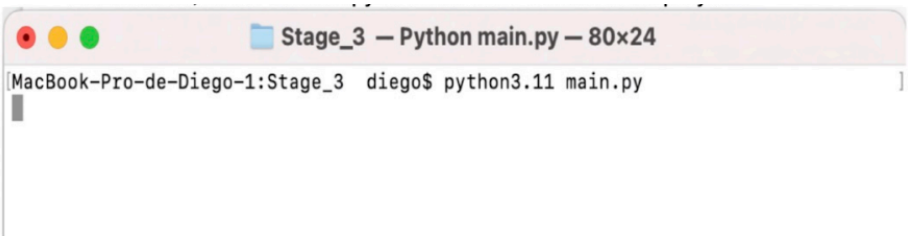
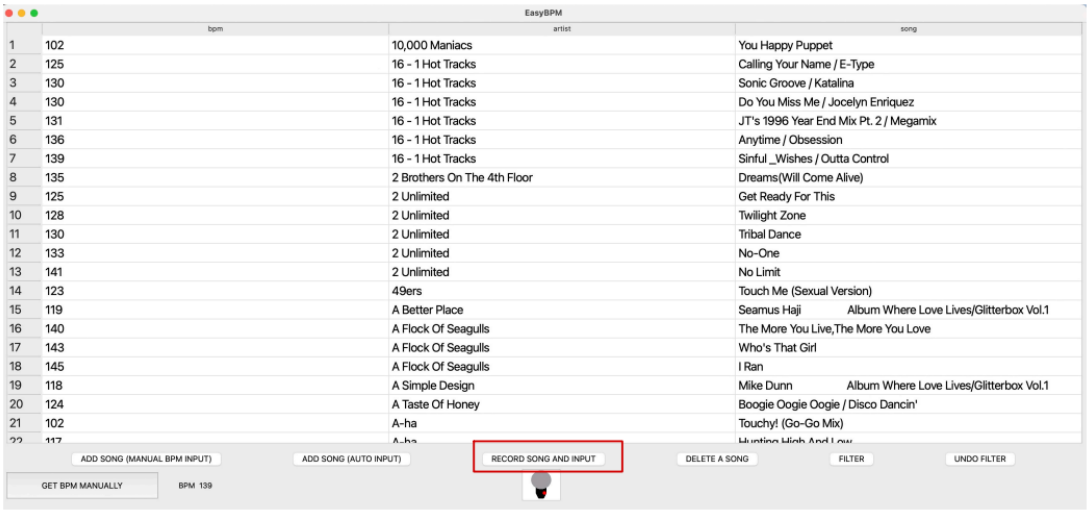


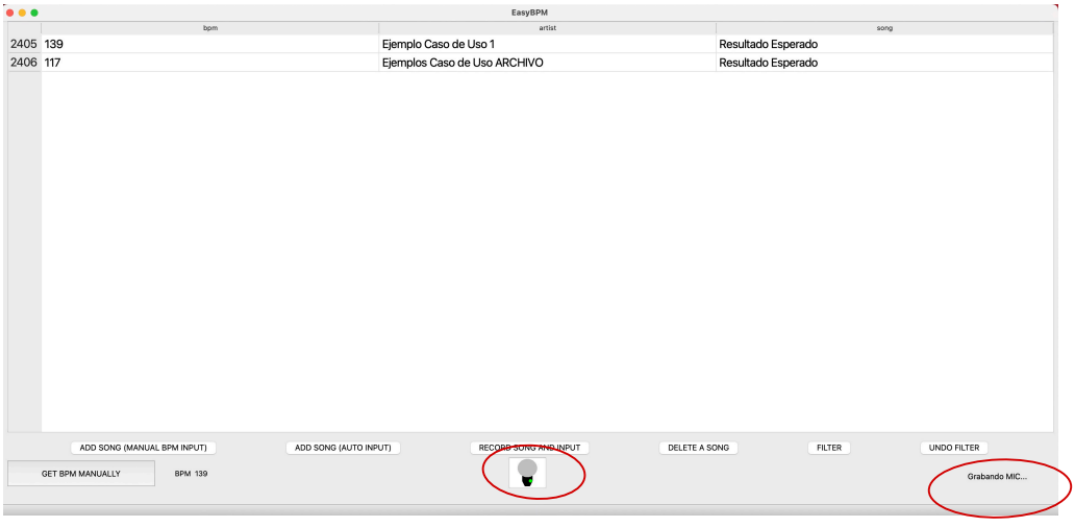

Figura 16: Terminal, ejecutando main.py ubicado en directorio del proyecto




| | bpm | artist | song |
|---|---|---|---|
| 1 | 102 | 10,000 Maniacs | You Happy Puppet |
| 2 | 125 | 16 - 1 Hot Tracks | Calling Your Name / E-Type |
| 3 | 130 | 16 - 1 Hot Tracks | Sonic Groove / Katalina |
| 4 | 130 | 16 - 1 Hot Tracks | Do You Miss Me / Jocelyn Enriquez |
| 5 | 131 | 16 - 1 Hot Tracks | JT's 1996 Year End Mix Pt. 2 / Megamix |
| 6 | 136 | 16 - 1 Hot Tracks | Anytime / Obsession |
| 7 | 139 | 16 - 1 Hot Tracks | Sinful _Wishes / Outta Control |
| 8 | 135 | 2 Brothers On The 4th Floor | Dreams(Will Come Alive) |
| 9 | 125 | 2 Unlimited | Get Ready For This |
| 10 | 128 | 2 Unlimited | Twilight Zone |
| 11 | 130 | 2 Unlimited | Tribal Dance |
| 12 | 133 | 2 Unlimited | No-One |
| 13 | 141 | 2 Unlimited | No Limit |
| 14 | 123 | 49ers | Touch Me (Sexual Version) |
| 15 | 119 | A Better Place | Seamus Haji Album Where Love Lives/Glitterbox Vol.1 |
| 16 | 140 | A Flock Of Seagulls | The More You Live,The More You Love |
| 17 | 143 | A Flock Of Seagulls | Who's That Girl |
| 18 | 145 | A Flock Of Seagulls | I Ran |
| 19 | 118 | A Simple Design | Mike Dunn Album Where Love Lives/Glitterbox Vol.1 |
| 20 | 124 | A Taste Of Honey | Boogie Oogie Oogie / Disco Dancin' |
| 21 | 102 | A-ha | Touchy! (Go-Go Mix) |
| 22 | 117 | A-ha | Hunting High And Low |

Figura 17: Página principal del programa indicando sector RECORD SONG AND INPUT

Una vez seleccionada esta opción, se indica por medio de la interfaz los efectos que indicaran que el programa esta utilizando el micrófono y por ende grabando el audio. Ver figura 18.




| | bpm | artist | song |
|---|---|---|---|
| 2405 | 139 | Ejemplo Caso de Uso 1 | Resultado Esperado |
| 2406 | 117 | Ejemplos Caso de Uso ARCHIVO | Resultado Esperado |

Figura 18: Página principal del programa indicando que el micrófono esta siendo utilizado

.

Finalizando la captura del audio, se abrirán las mismas ventanas de diálogos mostradas en la Figura 13. En estas, al igual que en los casos anteriores, el no ingresar los datos lo considerara como una operación invalida. Luego de esto los efectos que indican que se esta utilizando el micrófono del programa volverán a su estado inicial.

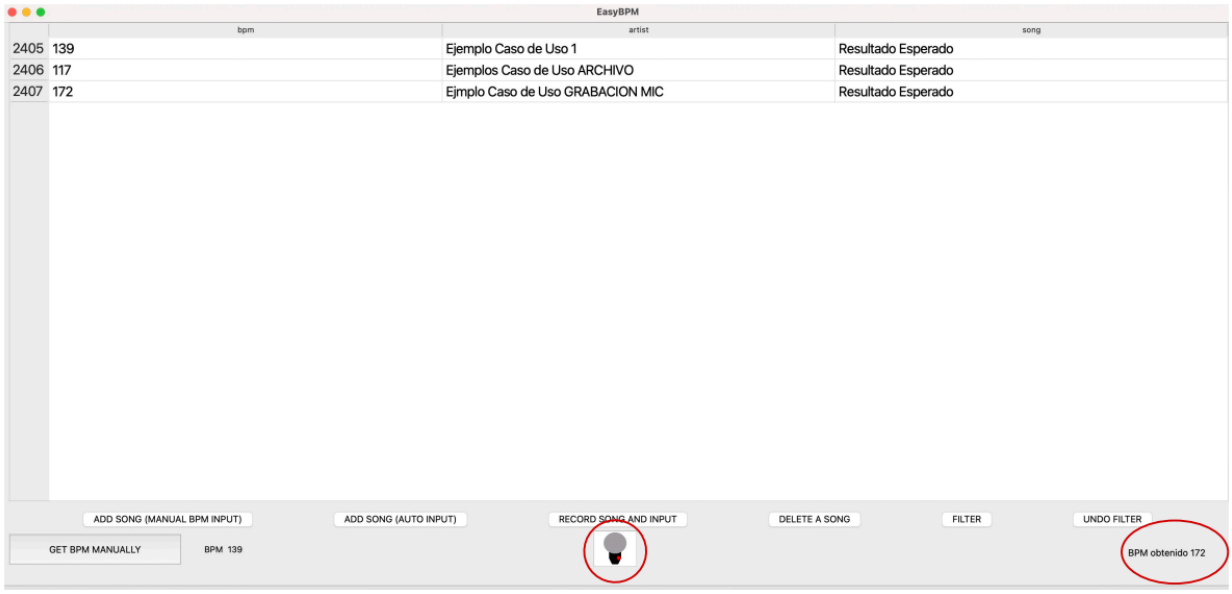



Figura 19: Página principal del programa muestra resultados y se indica que el micrófono deja de ser utilizado

## 6.4. Caso de Uso: Manejo de Excel

En esta sección ya no se requiere el uso externo del main.py.

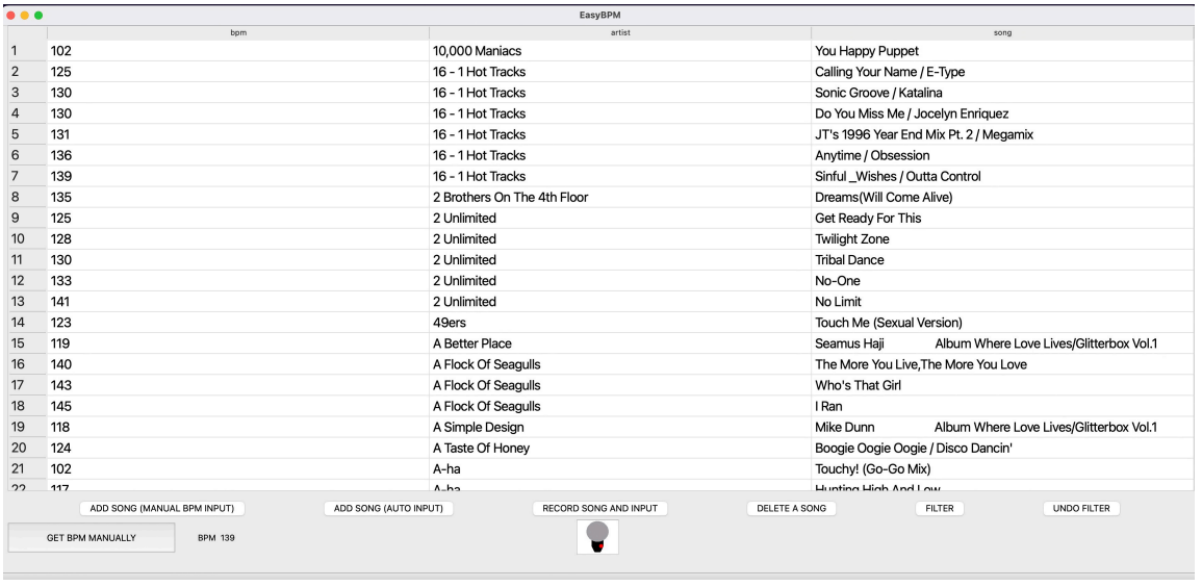



Figura 20: Página principal del programa (default)

En la Figura 20 en la parte superior podemos notar varias opciones tales como la pestaña BPM, artist o song, las cuales si se presionan ordenan de manera ascendente o descendente tomando como referencia la columna seleccionada.

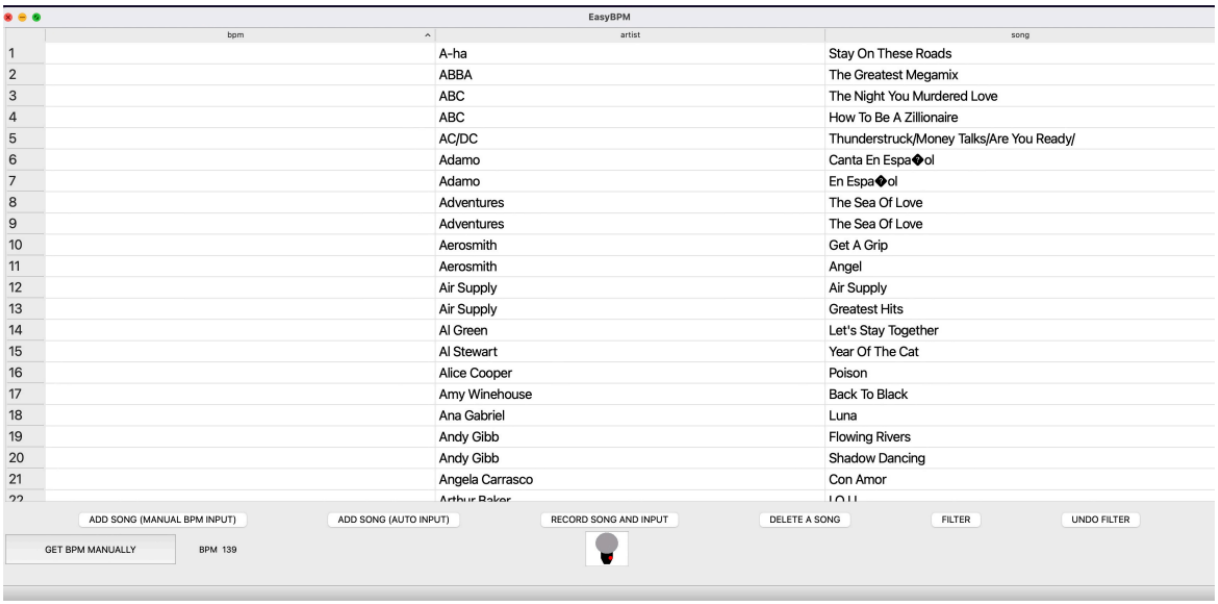



Figura 21: Página principal del programa (ordenanda por bpm)

Los primeros datos en aparecer son los bpm sin ingresar (por esta razon en la Figura 21 parte con los datos en blanco) Pero al continuar bajando se notaran las canciones ordenadas por bpm como se muestra en la Figura 22.

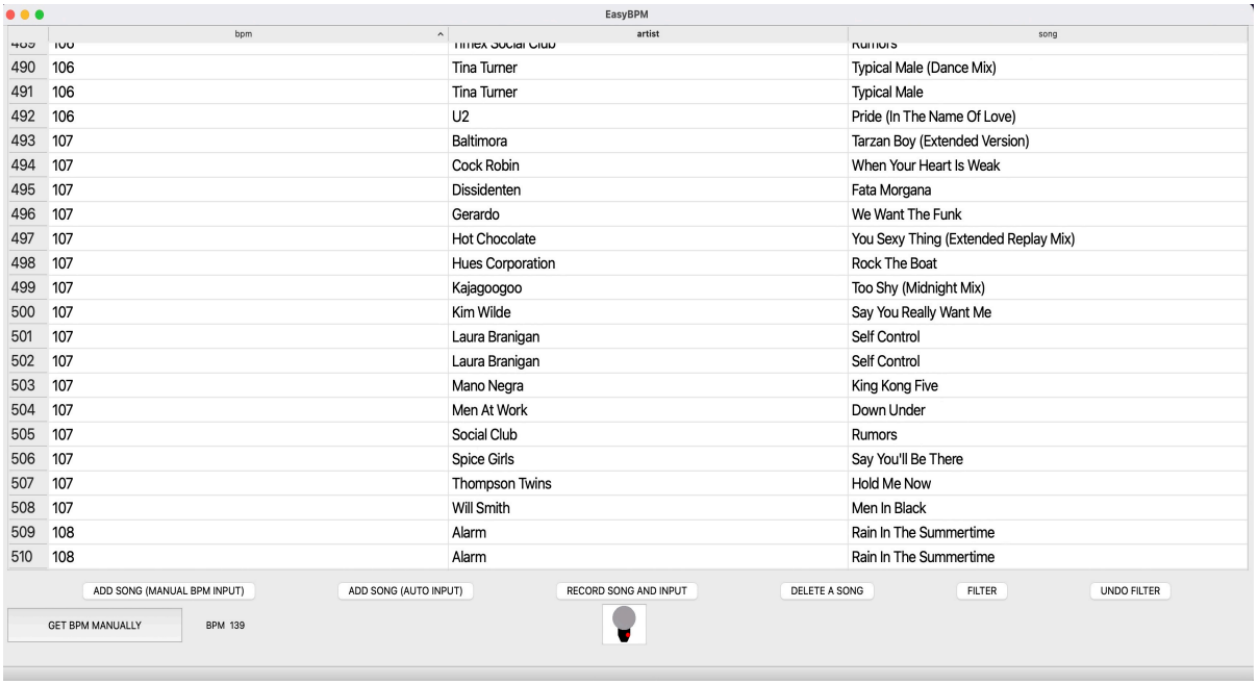



Figura 22: Página principal del programa (ordenanda por bpm)

### 6.5. Caso de Uso: delete a song

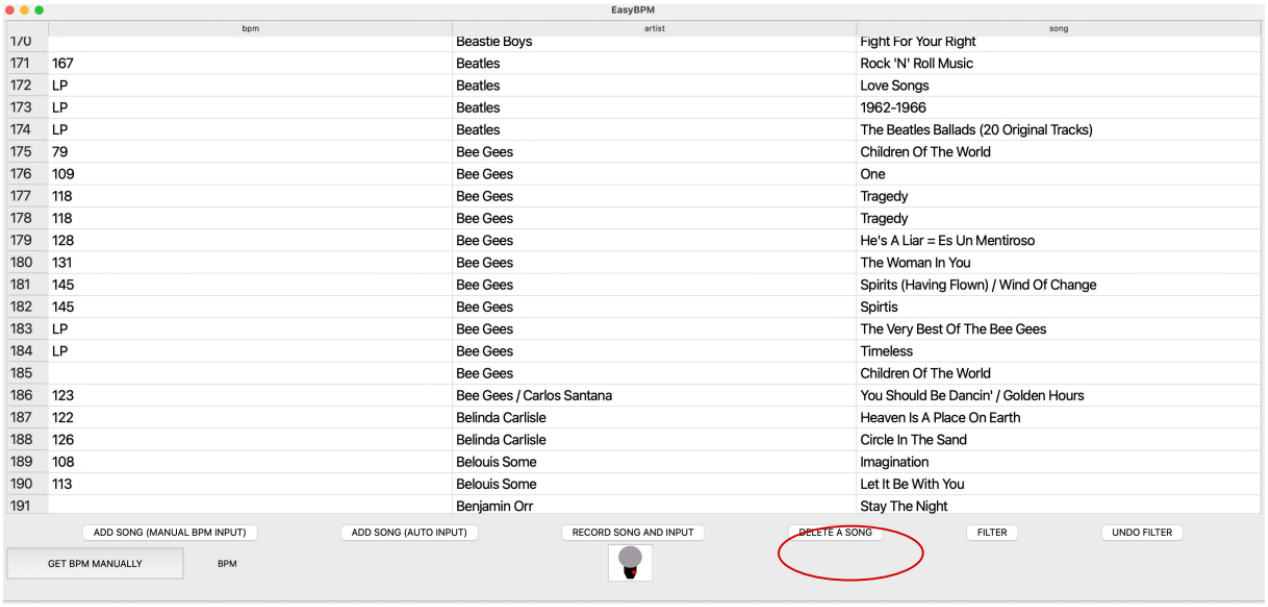



Figura 23: Página principal del programa indicando sector DELETE A SONG

Al seleccionar esta opción se abrirá una ventana la cual nos pedirá ingresar la fila a eliminar, cabe destacar que al hacer esta operación el programa no debe tener activo el filtro.

### 6.6. Caso de Uso: Filter/Undo filter

Por último, tenemos la opción de filtrar o eliminar un filtro ya establecido, seleccionando el botón FILTER o UNDO FILTER respectivamente, presentes en la página principal. En el caso del botón FILTER, mostrará por pantalla un ventana de diálogo (ver Figura 24) en donde se debe indicar por que palabra se desea filtrar.

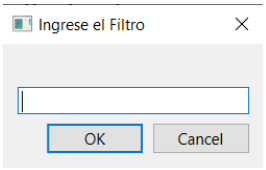



Figura 24: ventana de diálogo

## 7. Referencias

## Referencias

[1] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python.Ïn Proceedings of the 14th python in science conference, pp. 18-25. 2015.