

Proyecto: Generación de un  
archivo .wav a partir de archivo .txt

Versión: 1.0

Descripción y modelo de análisis

Elaborado por:  
Chong Hernández Samuel  
López Campillo Francisco Daniel  
Mendoza Hernández Carlos  
Emiliano  
Roa Díaz Vanessa

Fecha: viernes 13 de enero del 2023

# 1. Índice

<b>1. Índice .....</b>	<b>2</b>
<b>2. Introducción .....</b>	<b>3</b>
<b>3. Casos de Uso-Análisis .....</b>	<b>4</b>
<b>4. Diagrama de Clases.....</b>	<b>5</b>
<b>5. Diagramas de Secuencia .....</b>	<b>6</b>
<b>6. Diagramas de Estado.....</b>	<b>7</b>
<b>7. Descripción de Arquitectura.....</b>	<b>8</b>

## 2. Introducción.

En la digitalización existen diversas formas de identificar archivos, sea esta por su terminación, por ejemplo: .pdf, .png, .docx, etc. Esto para diversos ambientes, sea para un archivo de texto plano, un archivo word, una imagen o un sonido. Para este último existen diversos formatos en los que se pueden generar, sea a través de formatos de audio comprimido sin pérdida como los archivos ".FLAC", ".ALAC", entre otros; y los formatos de audio sin compresión como los ".aiff", ".pcm" o en los que se enfoca este proyecto, los archivos ".Wav".

Los archivos de sonido WAV o por su nombre en inglés Waveform Audio Format, son un formato de archivos de audio creado por IBM y Microsoft. Estos contienen sonidos tales como efectos, música o grabaciones de voz. Dado que no están comprimidos y su tamaño es mayor al de un archivo MP3, no son muy populares, sin embargo, al ser archivos muy grandes representan archivos de alta calidad. Su característica principal radica en que son una aplicación del método de almacenamiento de datos Resource Interchange File Format, una forma de guardar los datos en porciones (chunks), los cuales contienen una etiqueta de cuatro caracteres y el número de bytes de cada porción, ofreciendo la posibilidad de extender el formato fácilmente en el futuro.

El proyecto que se presenta a continuación tiene como objetivo la generación de archivos WAV por medio de un archivo de texto plano (.txt) el cual contiene los datos necesarios para la generación de este.

Para este se requirió de una amplia investigación para saber cómo formar el archivo WAV, resultando en el conocimiento de la codificación de flujo de bits; además del requerimiento de ciertas librerías para la manipulación de archivos en el lenguaje de programación Java.

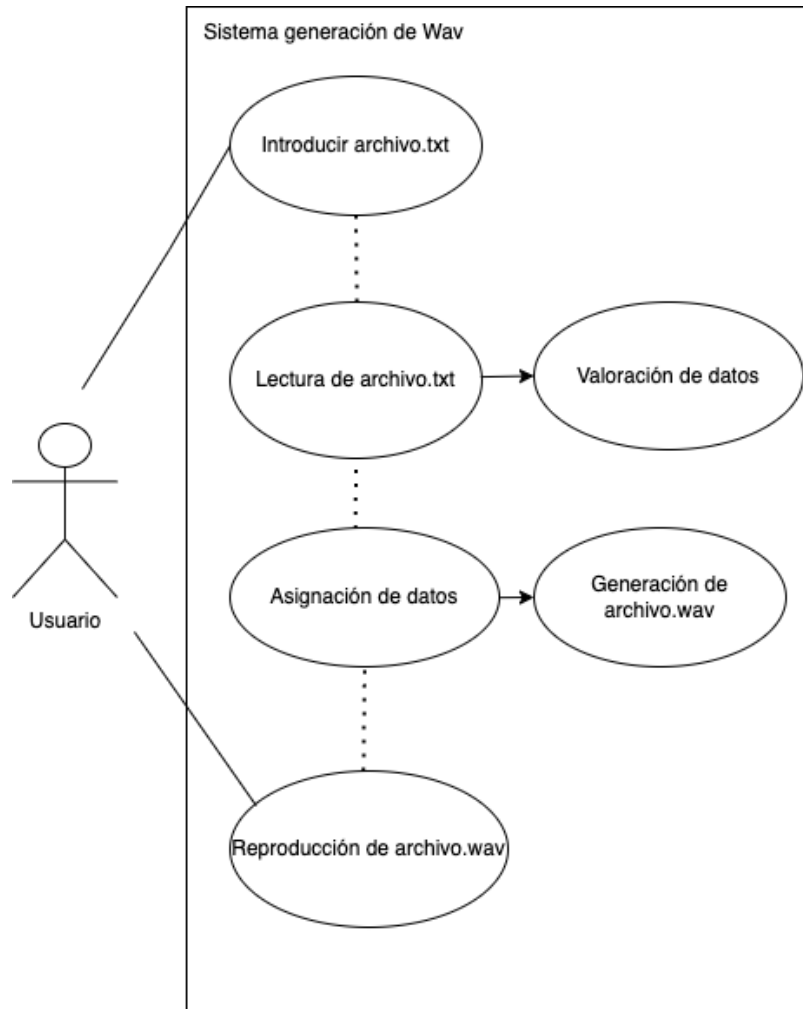
Profundizando en la solución, algunos aspectos fundamentales son el tamaño de archivo, número de canales o la frecuencia de muestreo.

Se tomó la decisión de que las excepciones manejadas a lo largo del desarrollo son bastante precisas y minuciosas, ya que esto propicia que el producto sea compatible con cualquier reproductor.

A continuación, se muestra la estructura del proyecto resultante, representada gráficamente por medio de UML.

### 3. Casos de Uso-Análisis.

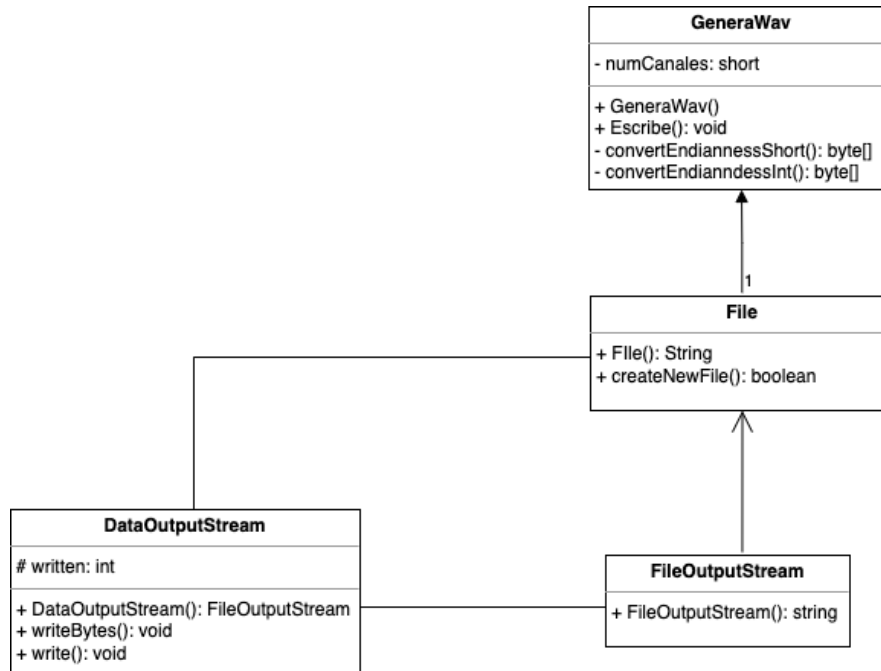
A través de la gráfica de casos de usos, podemos notar la forma en la que el usuario interactúa con el sistema creado. Dado que es un sistema en el que se trabaja sobre consola, esta forma resulta sencilla.



Al ser un sistema sencillo, el usuario únicamente interactúa con el sistema por medio del ingreso del archivo de texto plano con el contenido necesario para la generación del WAV. El sistema es el que se encarga de realizar las operaciones necesarias para hacer la conversión, resultando en el archivo WAV. Hecho esto, el usuario puede decidir si interactuar con el sistema para reproducir su archivo WAV generado.

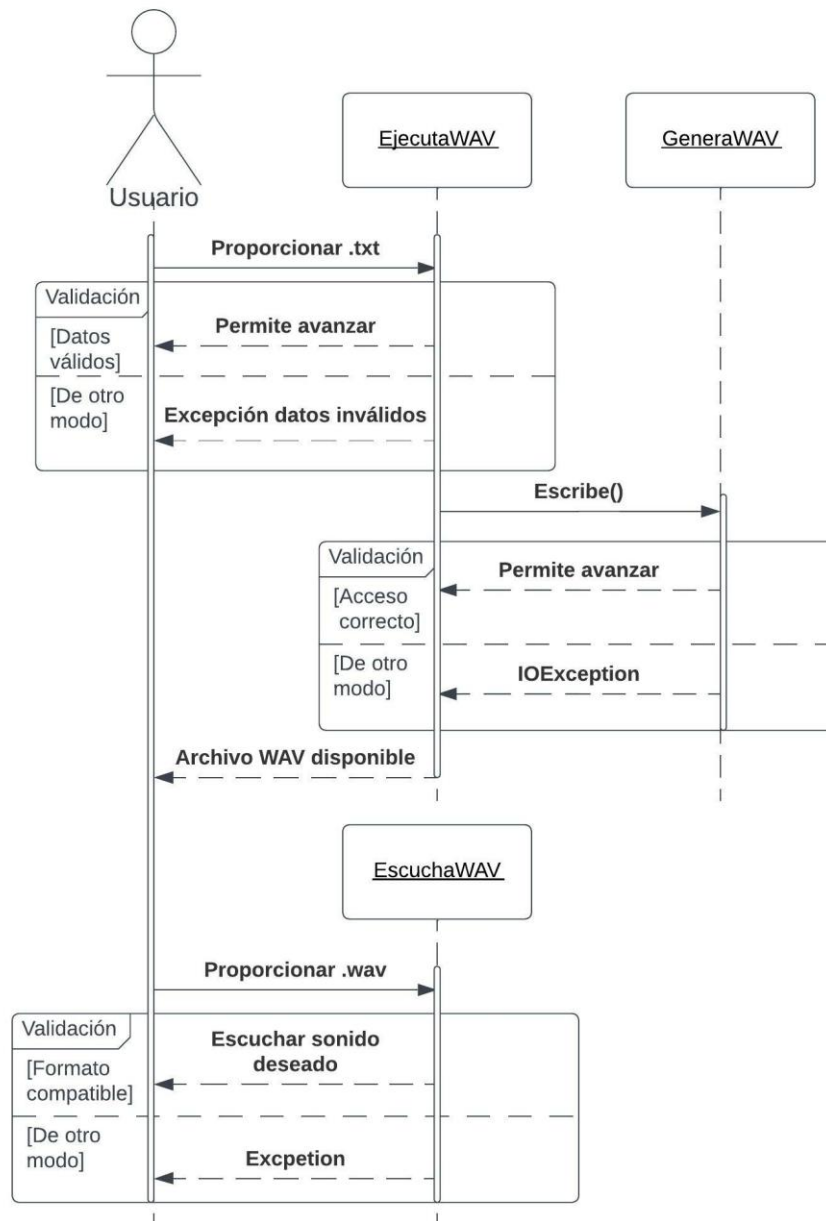
## 4. Diagrama de clases

El diagrama de clase es sencillo, ya que es una única clase la que se encarga de realizar la conversión del archivo. Como podemos ver a continuación:



Se muestra la relación entre clases, no solo con la encargada de generar el archivo WAV si no con las que fueron necesarias. Del diagrama podemos ver la relación que se presenta entre las clases encargadas del manejo de archivos, viendo su asociación debido a que trabajan sobre un mismo archivo. Posteriormente vemos la relación entre la clase **File** y la clase encargada de generar el WAV, mostrando una métrica de 1 debido a que es en un solo archivo sobre el que se trabaja y realiza la conversión.

## 5. Diagrama de Secuencia



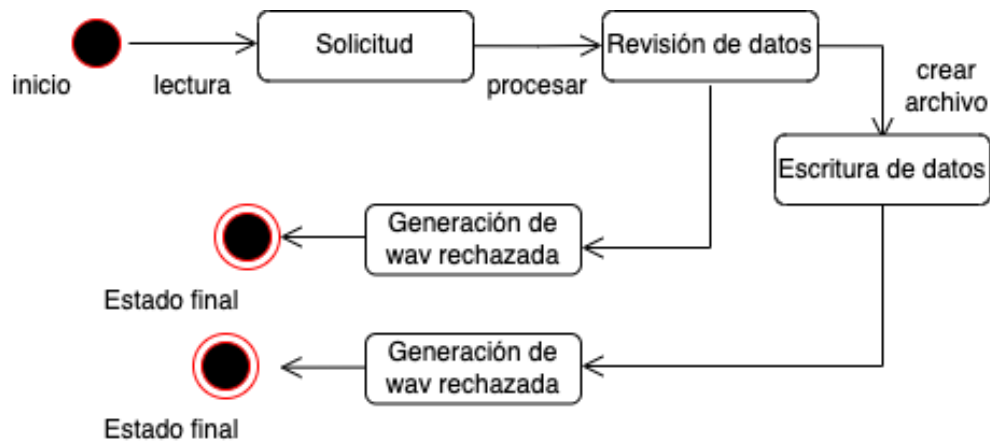
En este diagrama podemos observar las interacciones que existe entre el usuario y las clases (y las clases entre ellas):

1. Se proporciona el archivo **.txt** para validar su información de manera que sea compatible con los requerimientos que solicita la función **Escribe()**.
2. Esta debe acceder a dicha información con base en los nombres dados por el usuario, por lo que en caso de cometer algún error, retornará una excepción.
3. Si los datos y el acceso son exitosos, se genera el archivo **.wav** deseado, donde el usuario puede visualizarlo almacenado en su computadora.
4. Finalmente se usa la clase **EscuchaWAV** para poder escuchar el sonido deseado, proporcionando el nombre de este.

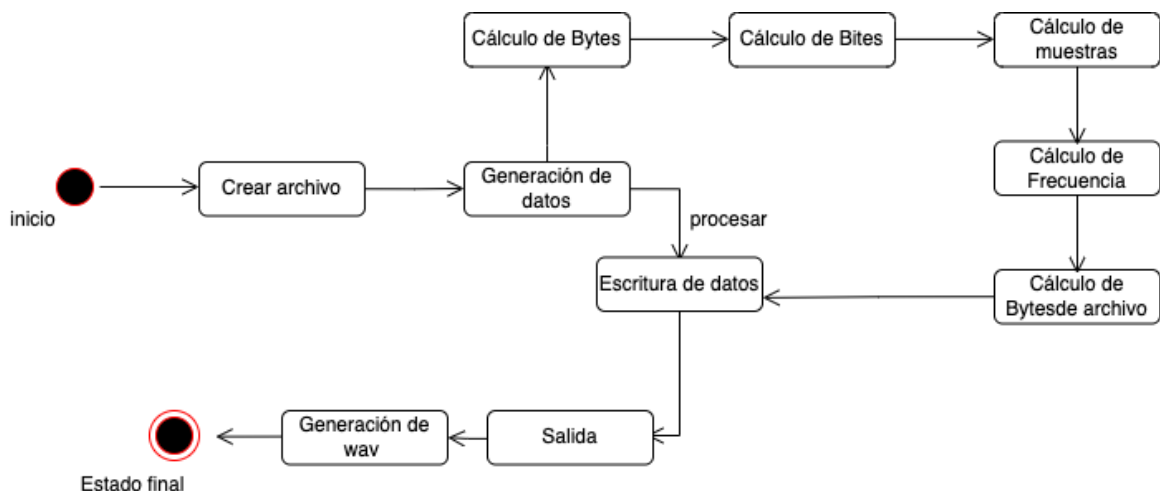
## 6. Diagrama de estado

A continuación se muestra el diagrama de estado tanto del sistema en general, como de la función principal. Este nos permite visualizar la secuencia por la que pasa nuestro archivo para llegar a su conversión.

Primeramente tenemos el diagrama de nuestro sistema completo, mirando las etapas por las que debe pasar el archivo para finalmente ser generado en un archivo WAV. Siendo estas la solicitud para ser leído, ser procesado y así realizar la revisión de sus datos para ser comprobados con el fin de saber si son los correctos para realizar la generación del archivo WAV. Depende de esta última etapa si se permitirá la generación del archivo WAV; de ser correctos los datos, se realizará la escritura y así la generación del WAV; de lo contrario se rechaza la generación del WAV.

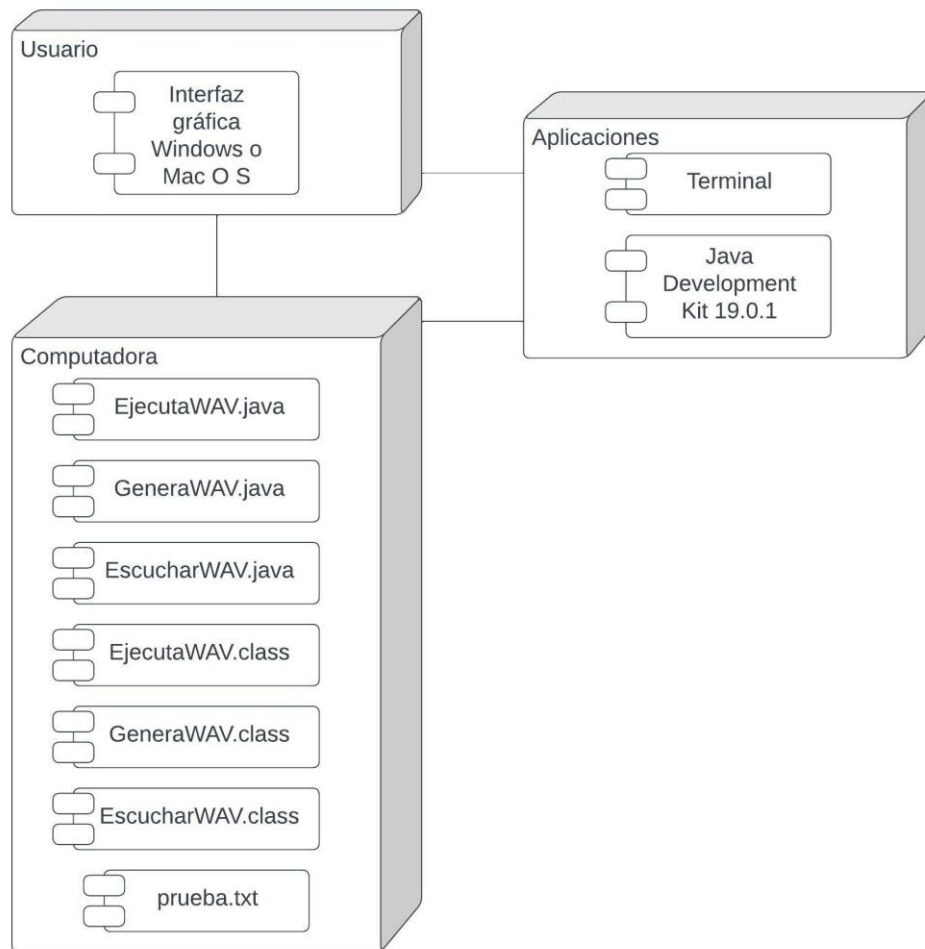


Por otro lado, tenemos el diagrama de estado de la función principal, aquella de generar el WAV. Se muestra el flujo de los datos que se introducen, mirando el proceso que debe transcurrir para la generación del WAV.



## 7. Descripción de Arquitectura

A continuación, se presenta el diagrama de arquitectura general de nuestro sistema:



En la parte de Usuario, encontramos el medio a través del cual el usuario tiene acceso directo al programa y su utilización. También identificamos los archivos que debe contener la computadora para poder ejecutar de manera exitosa la aplicación (recordando que la información de los archivos .java y el archivo .txt varía dependiendo del sonido que se desee implementar). Por último, es necesario tener acceso a la terminal, que permitirá seguir los pasos necesarios para compilar y ejecutar los programas (a través de las herramientas del Java Development Kit) exitosamente. Esta es una vista general de los componentes necesarios para poder emplear el proyecto desarrollado.