Práctica 7

Datos generales:

Nombre de la Práctica	Arreglos unidimensionales
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Laboratorio de Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	1. Introducción a las estructuras de datos y estructuras fundamentales.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	2-septiembre-2021 7:00 am
Fecha de entrega con extensión y penalización	3-septiembre-2021 7:00 am
Fecha de elaboración	1-septiembre-2021

Objetivo de la Práctica	Comprender el uso de los arreglos unidimensionales para la resolución de un problema real.
Tiempo aproximado de realización	4 horas
Introducción	La facilidad que los arreglos tienen para permitir guardar más de un dato con una sola variable lo hace pertinentes para la resolución de muchos problemas dónde se requiere esta situación. El único detalle con esta cuestión es que es poco flexible el número de elementos que podemos manipular, ya que se requiere conocer a priori la cantidad de elementos a guardar.

Referencias que debe consultar el alumno (si se requieren):

Referencia 1:

1.Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.

Referencia 2:

2.Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.

Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.

Actividades que debe realizar el alumno:

Actividad inicial:

Lea primero toda la práctica. No inicie a programar sin leer todo cuidadosamente primero. Recuerde que debe generar el reporte en formato IDC.

Actividad 1:

Primero genere la **Introducción**.

Actividad 2:

El sonido es una onda de presión que requiere un transductor de presión (un micrófono) para convertir las ondas de presión de aire (ondas sonoras) en señales eléctricas (señales analógicas). La conversión contraria se realiza mediante un altavoz (bocina), el cual convierte las señales eléctricas en ondas de presión de aire.

Sele pide que grabe un frase auditiva usando el programa Wavesurfer (http://www.speech.kth.se/wavesurfer/). En la figura 1 se puede ver este programa.

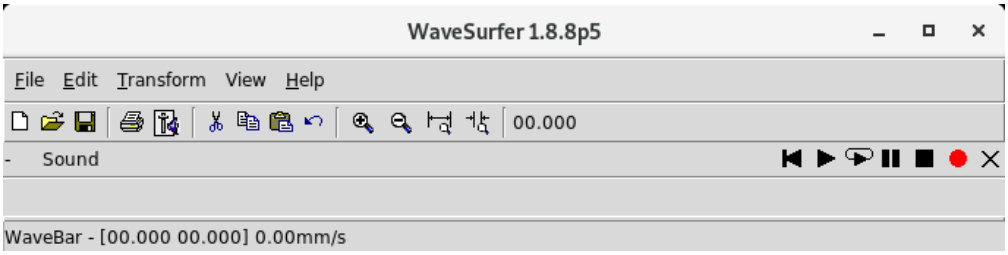

Figura 1. Wavesurfer.

Oprimiendo el botón de botón rojo (circular) comienza la grabación de voz, sin embargo, antes de eso se requiere realizar lo siguiente:

- Oprima botón derecho sobre la zona que se encuentra debajo de la palabra "Sound", aparecerá un menú como el de la figura 2, ahí debe seleccionar agregar panel de "waveform".

- Después puede oprimir el botón rojo para comenzar la grabación, se le pide que inmediatamente después de presionar el botón diga su "nombre completo" con todo y apellidos y la frase indicada abajo (después de completar oprima el botón negro cuadrado para detener el proceso). Por ejemplo el audio debe quedar grabado con "Aldonso Becerra Sánchez, quiero que sepan que mi mamá me mima y amanecer, ocaso, amanecer, ocaso, los pastores presurosos llevan de tanto correr los zapatos rotos". Y quedará grabado algo como lo de la figura 3.
- Posteriormente grabe el archivo como archivo .wav.

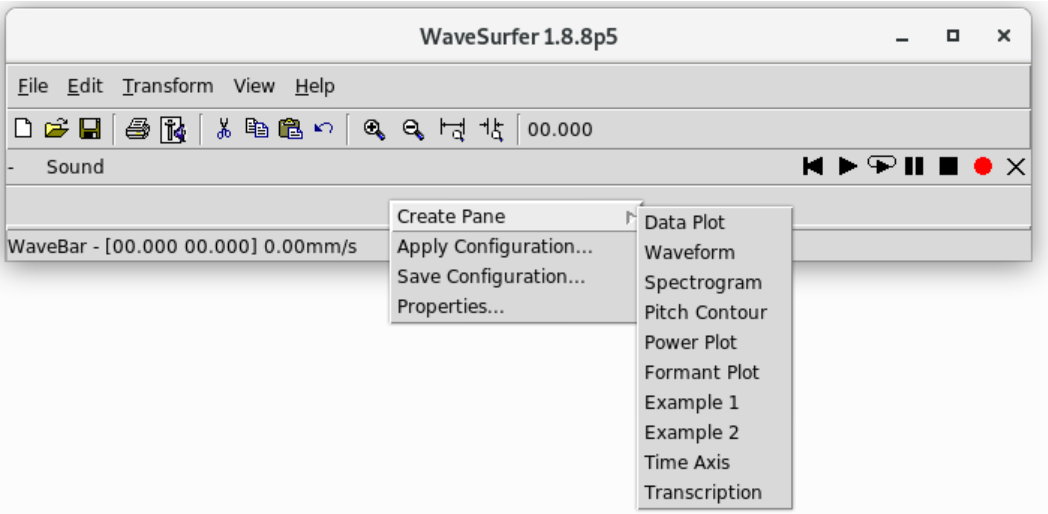

Figura 2.

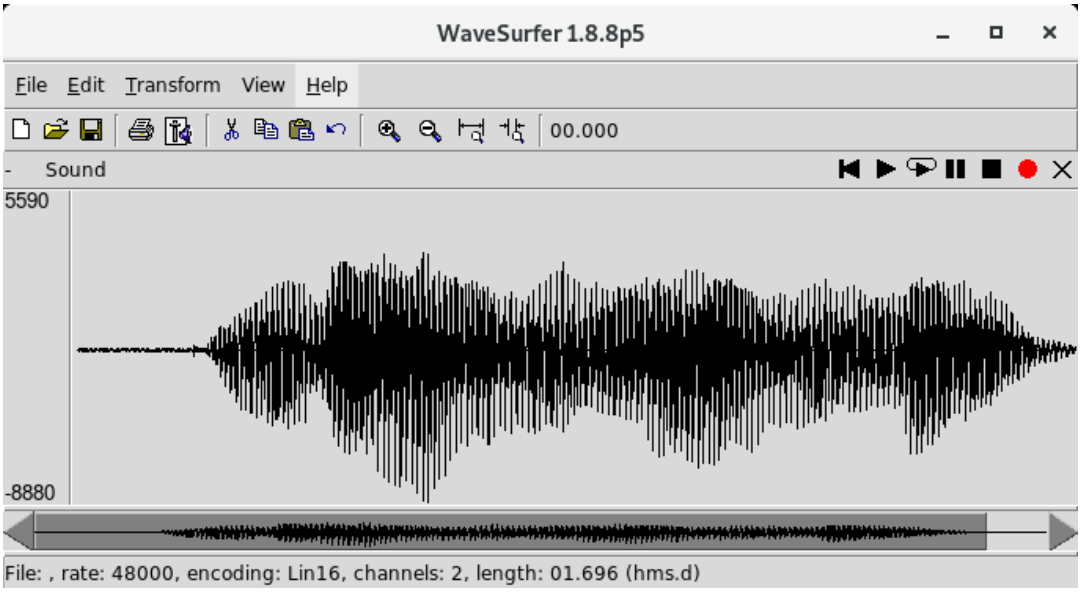

Figura 3.

Si se oprime botón derecho sobre la zona de la forma de onda aparece un menú, ahí oprima el botón del ratón sobre propiedades (figura 4). Ahí se pueden apreciar las características de la señal grabada; por ejemplo, la tasa de muestreo (el número de puntos que son obtenidos/discretizados (ver figura 5) de la señal analógica continua [forma de onda], el número de canales [generalmente se llama sonido estereofónico o estéreo al sonido grabado y reproducido en dos canales. Los discos de vinilo, los CD audio, la mayoría de las estaciones de radio FM, casetes y la totalidad de canales de TV y televisión vía satélite transmiten señales de audio estéreo. El propósito de grabar en sonido estereofónico es el de recrear una experiencia más natural al escucharlo, y donde, al menos en parte, se reproducen las direcciones izquierda y derecha de las que proviene cada fuente de sonido grabada], y el número de bis usados para codificar/discretizar la señal numérica que es convertida por el micrófono).

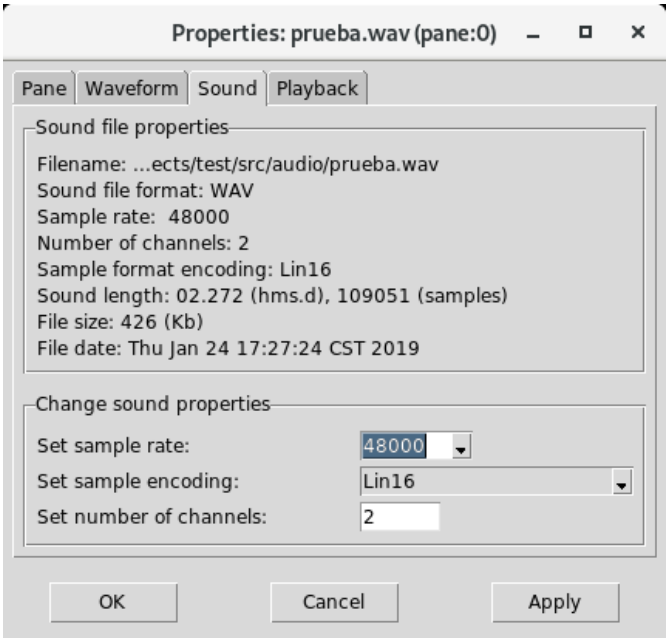

Figura 4.

Figura 5.

Tomando como base el programa que se proporciona en Java, el cual hace una lectura de un archivo de audio (wav), se le pide lo siguiente:

1. Acople el programa de la clase AudioFileRecord para que los datos del buffer2[] del método leerAudio ahora queden guardados en la clase ArregloNumeros. Para esto debe crear un método, en la clase ArregloDatos, que reciba el arreglo buffer[] (para que se pueda tener esta funcionalidad desde la super clase, no solo con valores numéricos) y lo guarde en el contenido del arreglo interno de la clase ArregloDatos (en la variable lote[], sobreescriba la variable interna del arreglo de los datos para que no se dupliquen los almacenamientos de memoria). La idea es que todo procesamiento se haga sobre una variable de tipo ArregloNumeros.
2. Acople el programa para que el método escribirAudio pueda guardar el buffer2[] como un arreglo[], para eso debe crear un método en la clase ArregloDatos (por ejemplo, public Object[] leerArreglo()) que proporcione una copia de los datos (no debe regresar el arreglo original, sino sólo una copia, esto garantiza la encapsulación) para después usar esa copia y almacenarla en buffer2[] como double, para que así pueda ser usado el método writeFrames. La idea es que la información guardada en la variable de tipo ArregloNumeros sea extraída para que wirteFrames pueda hacer su trabajo como arreglo con [].
3. A partir de aquí todas las modificaciones que se hagan sobre el archivo de audio deberán ser sobre la variable temporal de tipo ArregloNumeros.
4. Se pide que haga una serie de métodos de manipulación del audio. Por lo cual deberá elegir el lugar correcto para guardarlos.
5. Con la señal de voz discretizada (x[n], valor obtenido ya en el archivo .wav, y que el método leerAudio() ya proporciona) aplicar un filtro FIR de primer orden en las altas frecuencias, dado por $y[n] = x[n] + \alpha x[n-1]$, donde α es un valor configurable, habitualmente con 0.95 o 0.97. Este proceso se llamará void preEnfasis(). Verifique en qué cambio el audio, documento este hecho.
6. Un método en la clase AudioFileRecord que incremente el volumen del archivo de audio almacenado en la variable temporal buffer2 de tipo ArregloNumeros: public void subirVolumen(int intensidad). Intensidad es la cantidad de volumen a subir, la cual debe ser proporcionada por el usuario.
7. Se pide que haga un método en la clase AudioFileRecord que decremente el volumen del archivo de audio almacenado en la variable temporal buffer2 de tipo ArregloNumeros: public void bajarVolumen(int intensidad). Intensidad es la cantidad de volumen a bajar, la cual es proporcionada por el usuario.
8. Pruebe la funcionalidad de estos métodos guardando el archivo de audio (variable temporal de tipo ArregloNumeros convertida a arreglo [] para ser pasada a writeFrames) cada vez que hace una modificación del archivo temporal de audio.
9. Haga un método para acelerar la pista de audio (sonido tipo Alvin y las ardillas). Para hacer esto existen varias maneras, la que se pide que use es reducir el número

de muestras por unidad de tiempo (por ejemplo a la mitad, las muestras como en la figura 5). Por ejemplo, si originalmente eran 10000 muestras (tramas*número de canales), el nuevo archivo de audio quedará de 5000. El proceso para conseguir esto es agarrar dos muestras consecutivas y promediarlas (esto es acelerar el sonido al doble), creando un nuevo arreglo ahora con la mitad de muestras. Recuerde que todo este proceso es sobre una variable de tipo ArregloNumeros. El incremento en la velocidad debe ser un parámetro, No siempre debe ser el doble, este valor debe ser configurable en el método.

10. Haga un método para retrasar la pista de audio (sonido fantasmal). Para hacer esto existen varias maneras, la que se pide que use es aumentar el número de muestras por unidad de tiempo (ver figura 5). Por ejemplo, si originalmente eran 10000 muestras (tramas*número de canales), el nuevo archivo de audio quedará de 20000 (en este caso fue al doble). El proceso para conseguir esto es insertar la muestra "n" en un nuevo arreglo de tipo ArregloNumeros, y después de cada muestra "n" (o antes) se debe insertar una muestra inventada que será el relleno, la cual se obtiene como un promedio de la muestra actual "n" más la muestra "n+1" (o n-1, dependiendo). El decremento en la velocidad debe ser un parámetro, el cual debe ser configurado por el usuario.
11. Haga un método para que usted diseñe la manera de eliminar el silencio existente en un archivo de audio. Por ejemplo el mostrado en la figura 6:

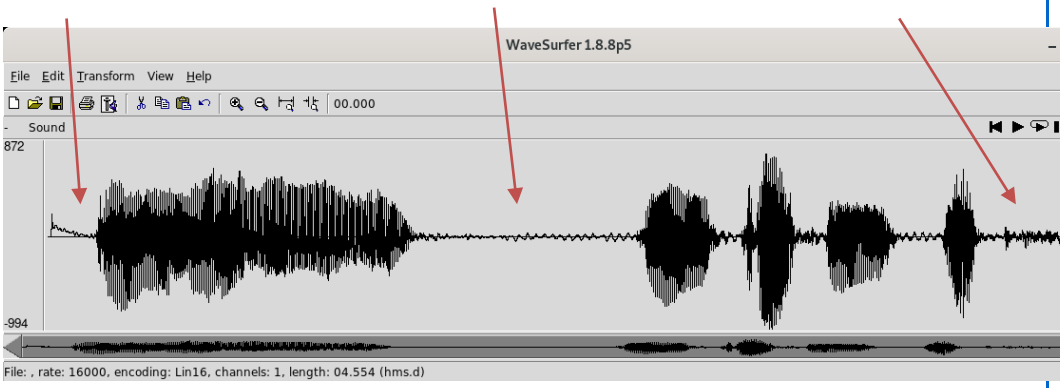

Figura 6.

Fíjese en el silencio puede existir en cualquier posición del audio, normalmente es antes o después de palabras pronunciadas (flechas en rojo). Ese silencio debe ser removido.

12. Haga que el audio grabado sufra una inversión en el eje x (se volteé al revés en tiempo, es decir lo que estaba a la izquierda ahora estará a la derecha, y así sucesivamente). Tal como se muestra la figura 7 y 8.

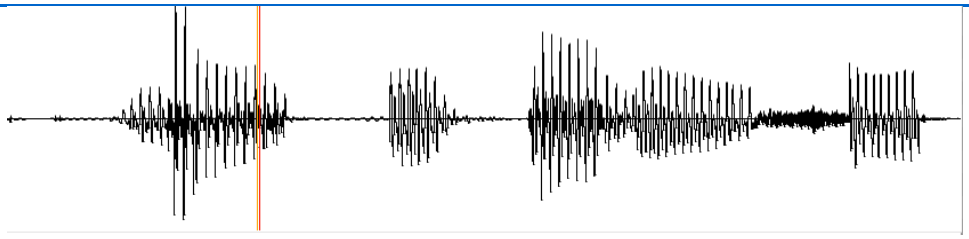

Figura 7.

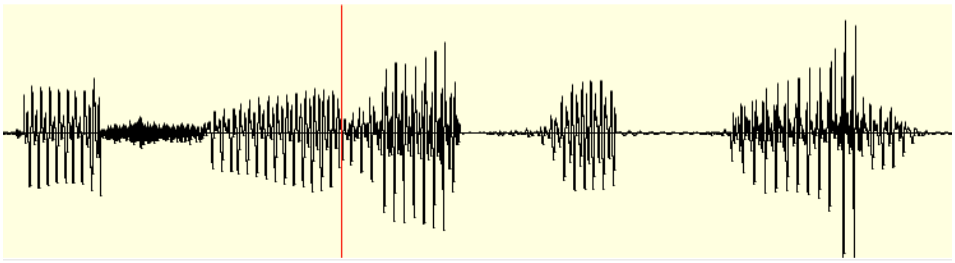

Figura 8.

13. Haga que el audio grabado sufra una inversión en el eje y (se volteé de cabeza en intensidad, es decir lo que estaba arriba ahora estará abajo, y así sucesivamente). Tal como se muestra la figura 9 y 10.

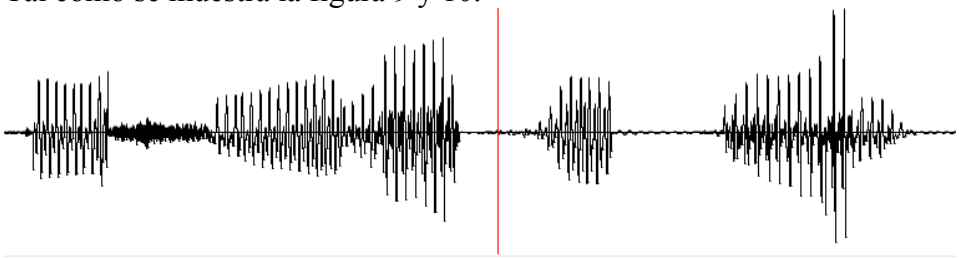

Figura 9.

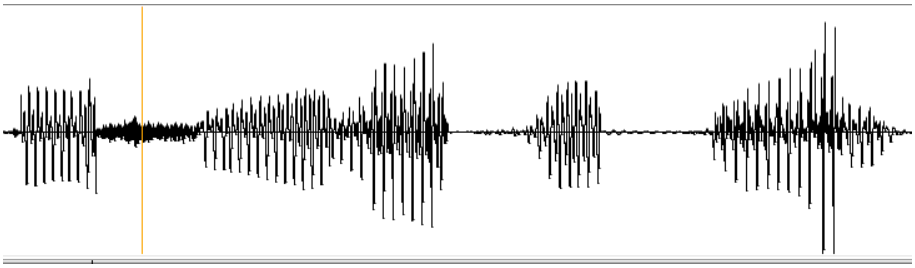

Figura 10.

14. Haga unas grabaciones de vocales (pronunciadas una vocal sola por cada archivo). Grabe varias veces una misma vocal, por ejemplo unas 5 veces. Posteriormente obtenga la energía que producen cada una de ellas (la energía de

una señal de audio se obtiene elevando al cuadrado cada una de las muestras individuales, posteriormente se suman todas). Verifique si a través de la energía puede clasificar vocales. Haga pruebas y documente los resultados.

15. Grafique los resultados en Excel (una pequeña parte de muestras, ya que son muchas) para que pueda ver que cada manipulación de toda la práctica ha sufrido un cambio, y agregue las imágenes de esto al reporte.
16. Pruebe el funcionamiento del programa grabando el arreglo resultante ArregloNumeros en un archivo .wav correspondiente en disco.

Haga el programa, la cual es el **Desarrollo** de reporte.

Actividad 3:

Pruebe el funcionamiento del programa de la actividad 2 con todo y sus capturas de pantalla.

Actividad 4:

Realice la sección de **Código agregado** (diagrama de clases UML).

Actividad 5:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).

Actividad 6:

Finalmente haga las **Conclusiones**.

Actividad 5:

Enviar en http://ingsoftware.reduaz.mx/moodle

Archivo anexo que se requiere para esta tarea (opcional):

Dudas o comentarios: a7donso@gmail.com