Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software.

Nombre de la Práctica Arreglos unidimensionales

Numero de Práctica 7

Nombre de la carrera Ingeniería de Software

Nombre de la materia Lab. Estructuras de Datos

Nombre del alumno Jesús Manuel Juárez Pasillas

Nombre del docente Aldonso Becerra Sánchez

Fecha: 02/09/2021

Práctica 7: Arreglos unidimensionales

Introducción:

Los arreglos unidimensionales nos permiten realizar todo tipo de operaciones sobre un conjunto de datos utilizando solo una variable para almacenar un numero finito de elementos, las operaciones se vuelven fácil de realizarlas por que se puede acceder a los elementos de una forma muy fácil (con un índice).

Desarrollo:

En esta actividad se estará haciendo uso de arreglos para modificar audios. Los datos que se almacenaran en el arreglo son numéricos por lo que se hará uso de la clase ArregloNumeros para almacenar el audio y modificarlo con este.

Para grabar el audio se hizo uso del programa “WaveSurfer #3” con el que se llevo a cabo un proceso para grabar un audio diciendo una frase indicada.
Algunos de los métodos creados son:

- public void preEnfasis(): Este método lo que hará es aumentar la frecuencia.
  El audio se escucha muy distorsionado como si se hubiera saturado el micrófono a la hora de grabar el audio.
- public void subirVolumen(int intensidad): Sube el volume del audio.
  Aumenta ligeramente el volume del audio.
- public void bajarVolumen(int intensidad): Baja el volumen del audio.
  Decrementa ligeramente el volumen del audio.
- public void acelerar(int velocidad): Hace que el audio sea mucho más rápido.
  Aumenta la rapidez del audio tanto como se lo indique el usuario.
- public void relentizar(int decremento): Hace que el audio sea mucho más lento.
  Ralentiza el audio tanto como el usuario se lo indique.
- public void invertirEjeX(): Invierte el audio para que se reproduzca de final a principio.
  Invierte el audio en el eje x, para que el audio se reproduzca en orden inverso al normal.

Nota: Toda la documentación esta agregada en la carpeta “doc” dentro de la carpeta del proyecto (“edylab_2021_7/doc”).

Capturas del programa funcionando:

PruebaAudio:

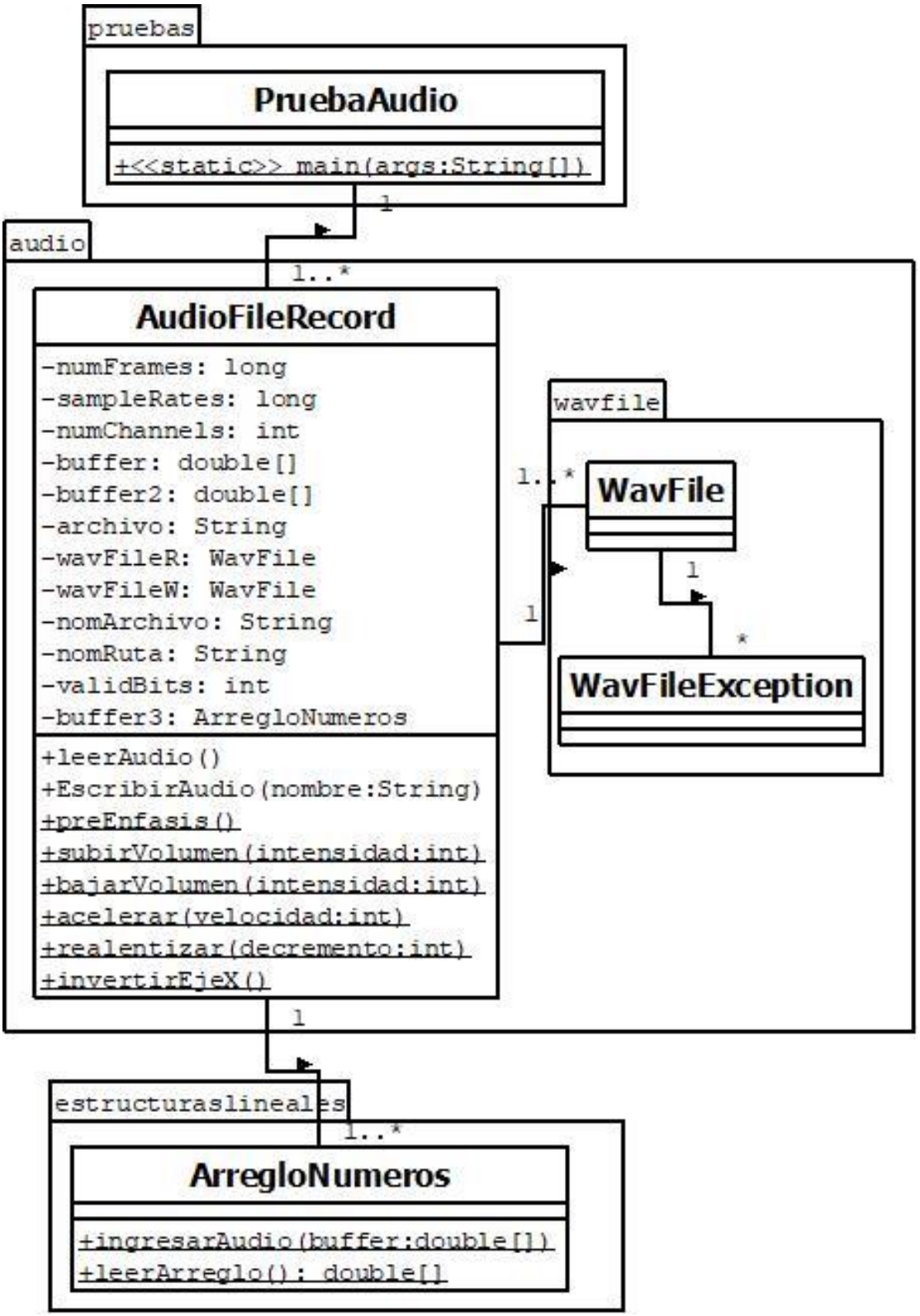
Los audios generados se encuentran en la carpeta del proyecto dentro de src/audios (edylab_2021_7/src/audios/”). Los audios almacenados se les dio un nombre desde el main donde se invoca el método “EscribirAudio(nombre)”, este método se modifico para que se le pase como parámetro el nombre que tendrá el archivo a almacenar.

Todos los audios fueron guardados con un nombre distintivo de lo que tienen modificado, por ejemplo, los archivos con aceleración están en “x2.wav” indicando la velocidad que tienen.

```
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
```

```
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
File: src\audios\audio.wav
Channels: 2, Frames: 597000
IO State: READING
Sample Rate: 48000, Block Align: 4
Valid Bits: 16, Bytes per sample: 2
```

Código agregado:



Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	No
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	Sí
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	Sí
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	Sí
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	Sí
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	75%
Observaciones:	

Conclusión:

Los arreglos son indispensables para acceder a cualquier elemento con mucha facilidad ya que cada elemento esta ligado a un índice con el cual es muy fácil acceder. Como los arreglos numéricos solo almacenan algún tipo de dato que herede de Number, podemos almacenar los datos de un audio y con cada elemento que se almacena en el arreglo hacerle alguna operación como sumar, restar, multiplicar, etc. y con esto obtener un audio modificado muy parecido al original ya que sobre este fue que se hicieron las operaciones.