



ugr

Universidad
de Granada

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA

Seminario: Manejo del sonido con R

Autora: Cristina María Crespo Arco

Correo: cmcrespo@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Curso 2021 - 2022

Índice

1. Primer ejercicio: Leer dos ficheros de sonido (WAV o MP3) de unos pocos segundos de duración cada uno.	2
1.1. Código:	2
1.2. Ejecución del programa:	2
2. Segundo ejercicio: Dibujar la forma de onda de ambos sonidos	3
2.1. Código:	3
2.2. Ejecución del programa:	3
3. Tercer ejercicio: Obtener la información de las cabeceras de ambos sonidos	5
3.1. Código:	5
3.2. Ejecución del programa:	5
4. Cuarto ejercicio: Unir ambos sonidos en uno nuevo	6
4.1. Código:	6
4.2. Ejecución del programa:	6
5. Quinto ejercicio: Reproducir la señal obtenida y almacenarla como un nuevo fichero WAV, denominado “mezcla.wav”	7
5.1. Código:	7
5.2. Ejecución del programa:	7

1. Primer ejercicio: Leer dos ficheros de sonido (WAV o MP3) de unos pocos segundos de duración cada uno.

Para leer los ficheros *gato.mp3* y *oveja.wav* es necesario usar las funciones “readMP3(‘gato.mp3’)” y “readWave(‘oveja.wav’)”, respectivamente.

1.1. Código:

```
# ***** librerías *****  
library(tuneR)  
library(seewave)  
library(audio)  
  
# ***** Ejercicio 1: Leer dos ficheros de sonido (WAV o MP3) de  
# unos pocos segundos de duración cada uno. *****  
gato <- readMP3('gato.mp3')  
oveja <- readWave('oveja.wav')
```

1.2. Ejecución del programa:

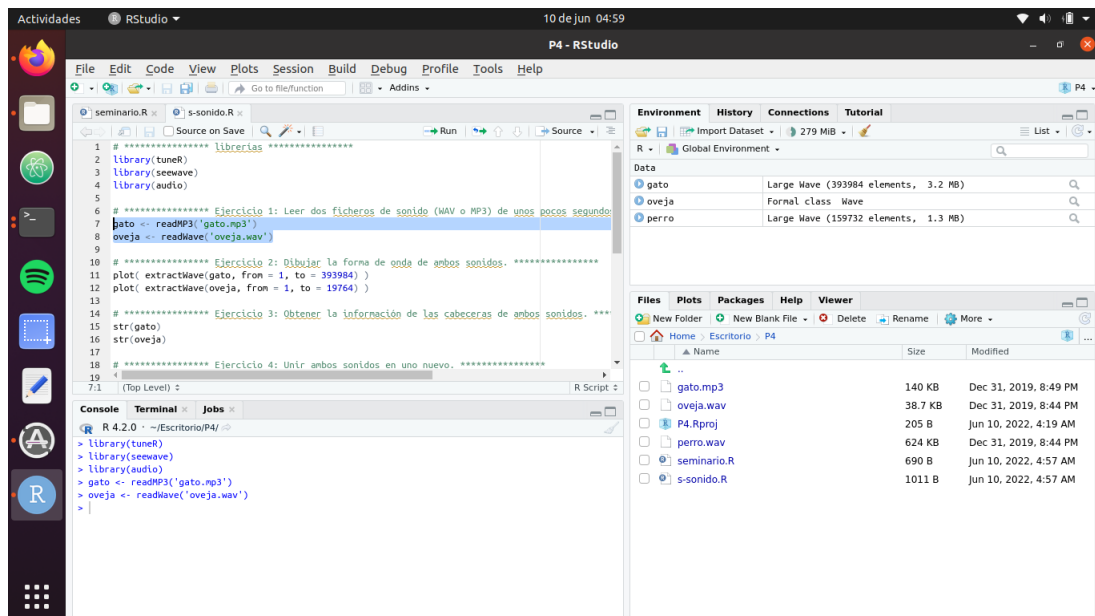


Figura 1: Ejecución del primer ejercicio

2. Segundo ejercicio: Dibujar la forma de onda de onda de ambos sonidos

Para dibujar la forma de onda de ambos sonidos he usado la función `plot(extractWave('sonido', from = 1, to = 'ultimaMuestra'))`.

2.1. Código:

```
# ***** Ejercicio 2: Dibujar la forma de onda de ambos sonidos.
# *****
plot( extractWave(gato, from = 1, to = 393984) )
plot( extractWave(oveja, from = 1, to = 19764) )
```

2.2. Ejecución del programa:

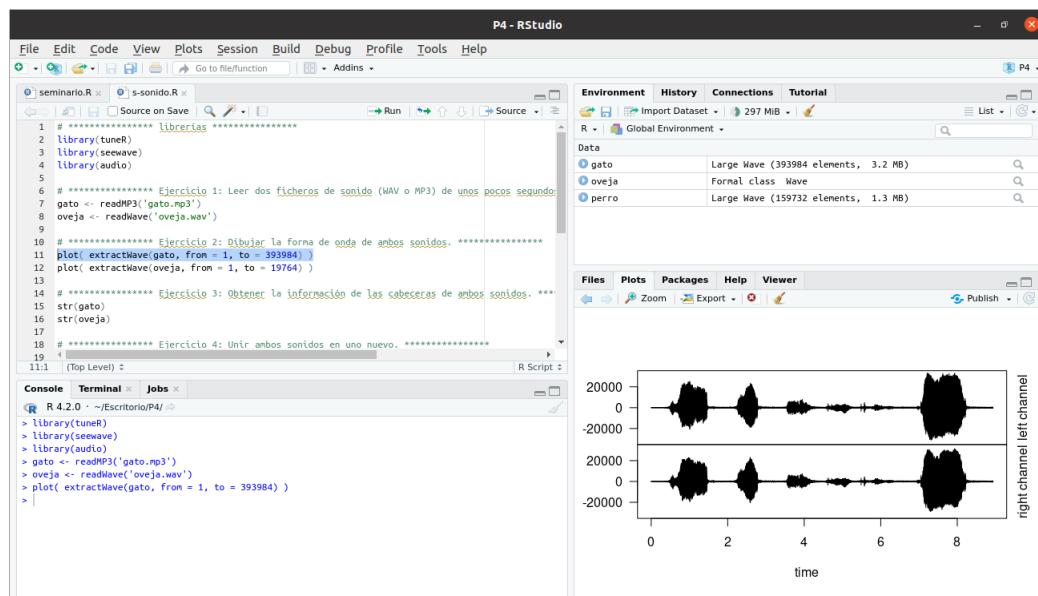


Figura 2: Ejecutar la primera gráfica: *gato.mp3*

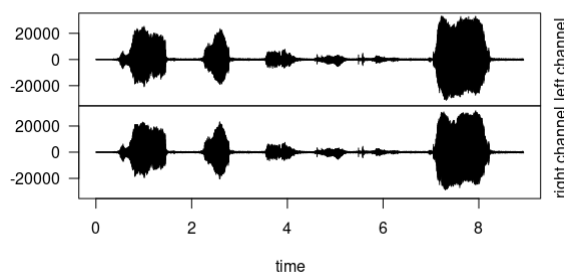


Figura 3: Gráfica del sonido *gato*

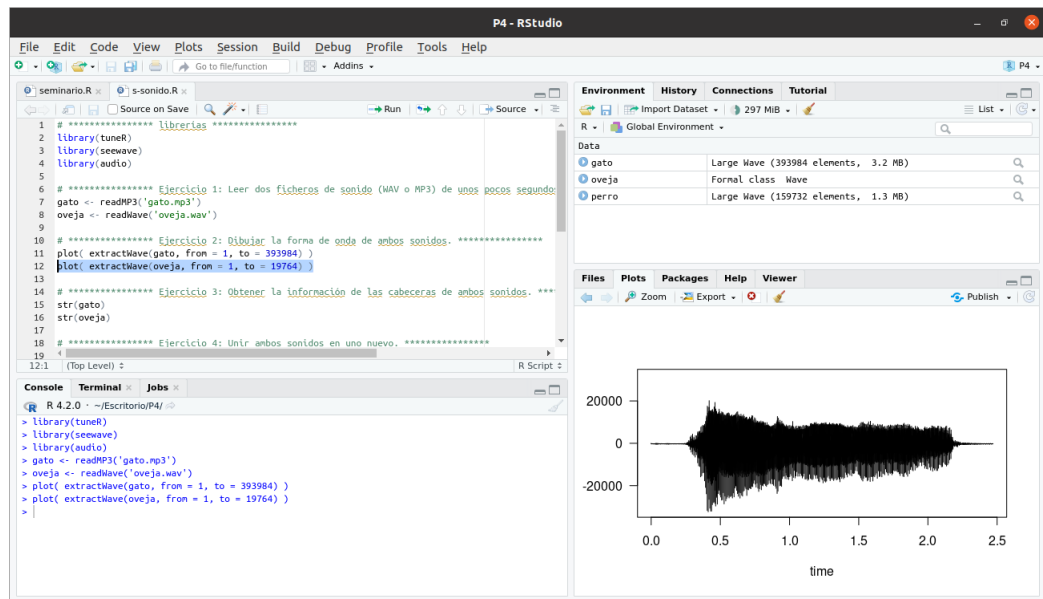


Figura 4: Ejecutar la primera gráfica: *oveja.wav*

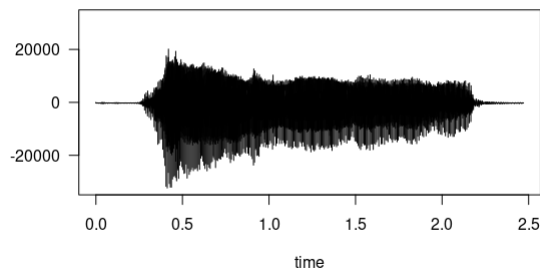


Figura 5: Gráfica del sonido *oveja*

3. Tercer ejercicio: Obtener la información de las cabeceras de ambos sonidos

Para obtener la información de las cabeceras de cada archivos he usado la función `str('sonido')`.

3.1. Código:

```
# ***** Ejercicio 3: Obtener la información de las cabeceras de  
# ambos sonidos. *****  
str(gato)  
str(oveja)
```

3.2. Ejecución del programa:

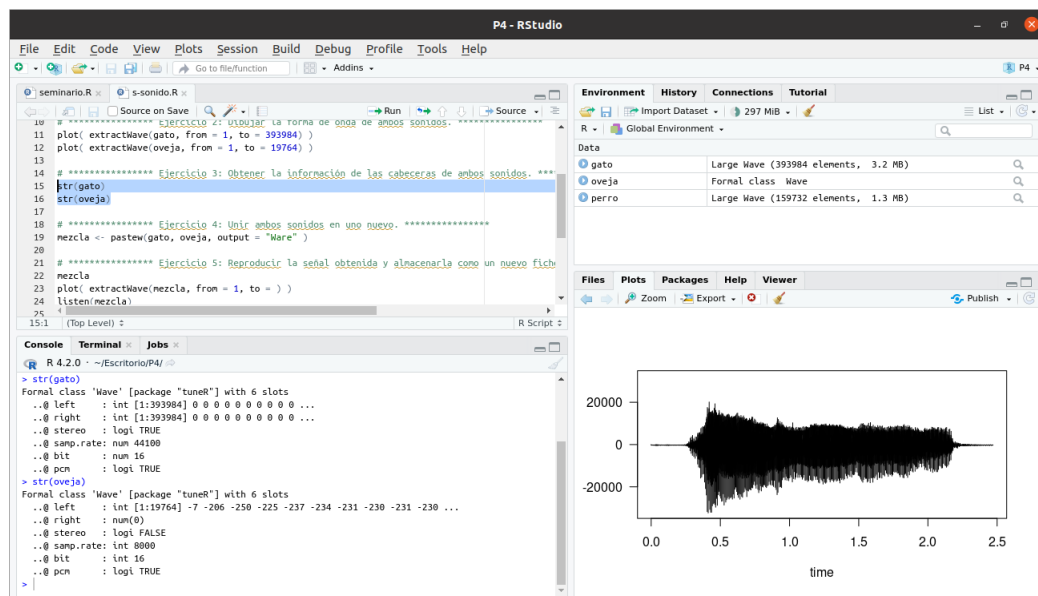


Figura 6: Ejecución del tercer ejercicio

4. Cuarto ejercicio: Unir ambos sonidos en uno nuevo

La función que he ejecutado para unir los dos sonidos ha sido **pastew(gato, oveja, output = "Wave")** y el resultado lo he almacenado en una variable denominada "mezcla".

4.1. Código:

```
# ***** Ejercicio 4: Unir ambos sonidos en uno nuevo. *****
mezcla <- pastew(gato, oveja, output = "Wave" )
```

4.2. Ejecución del programa:

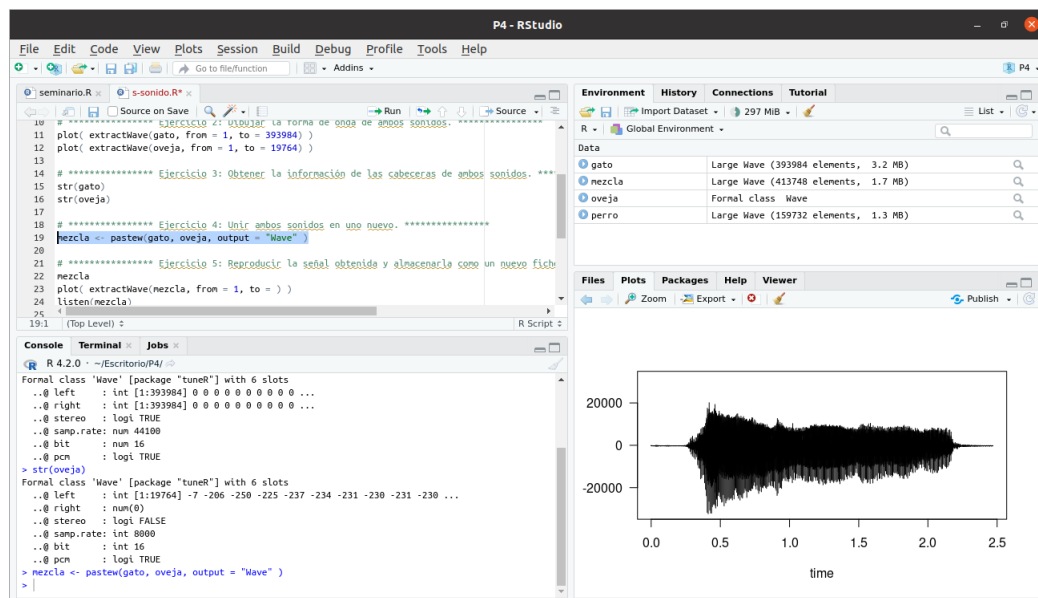


Figura 7: Ejecución del cuarto ejercicio.

5. Quinto ejercicio: Reproducir la señal obtenida y almacenarla como un nuevo fichero WAV, denominado “mezcla.wav”

Primero se muestra el contenido de la variable muestra, a continuación ejecutamos la función `plot(extractWave(mezcla, from = 1, to = 412748))` para mostrar el gráfico obtenido (Figura 9). Posteriormente, ejecutamos la función `listen (mezcla)` para reproducir el sonido. Y por último, almacenamos el sonido en el fichero *mezcla.wav*.

5.1. Código:

```
# ***** Ejercicio 5: Reproducir la señal obtenida y almacenarla
# como un nuevo fichero WAV, denominado \"mezcla.wav\". *****
mezcla
plot( extractWave(mezcla, from = 1, to = 412748) )
listen(mezcla)

writeWave(mezcla, file.path(\"mezcla.wav\"))
```

5.2. Ejecución del programa:

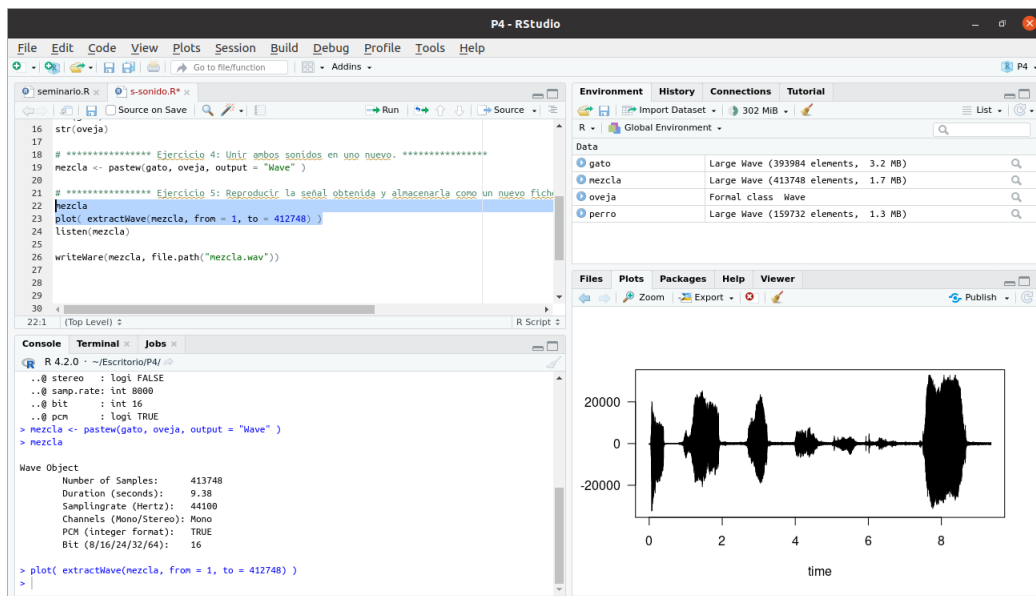


Figura 8: Ejecución del quinto ejercicio.

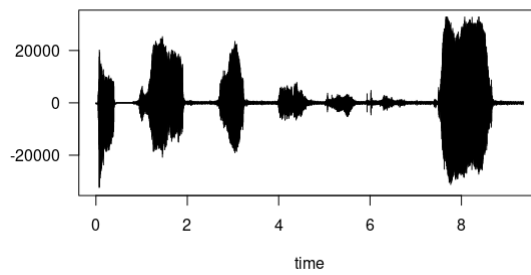


Figura 9: Gráfica del sonido *mezcla*

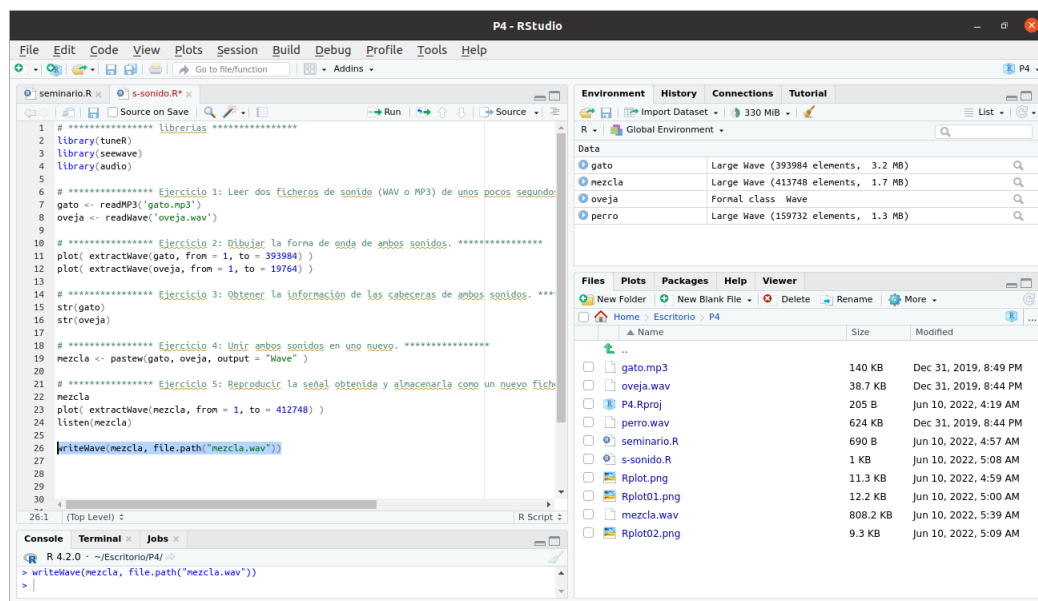


Figura 10: Guardar sonido en el fichero *mezcla.wav*