



ugr

Universidad
de Granada

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA

Práctica 5: Experimentación con el sistema de salida de sonido

Autora: Cristina María Crespo Arco

Correo: cmcrespo@correo.ugr.es

Autor: Andrés Piqueras Brück

Correo: andrespiqueras@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso



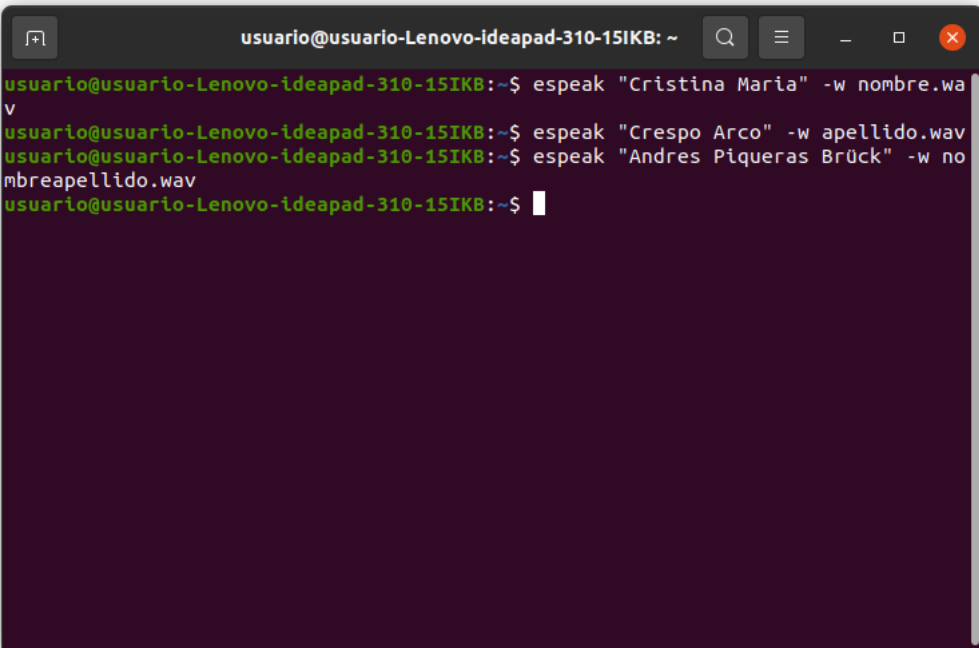
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Curso 2021 - 2022

Índice

1. Primer apartado: Leer dos ficheros de sonido (WAV o MP3) de unos pocos segundos de duración cada uno. En el primero debe escucharse el nombre de la persona que realiza la práctica. En el segundo debe escucharse el apellido.	3
1.1. Código:	3
1.2. Ejecución del programa:	3
2. Segundo apartado: Dibujar la forma de onda de ambos sonidos.	4
2.1. Código:	4
2.2. Ejecución del programa:	4
3. Tercer apartado: Obtener la información de las cabeceras de ambos sonidos.	6
3.1. Código:	6
3.2. Ejecución del programa:	6
4. Cuarto apartado: Unir ambos sonidos en uno nuevo.	7
4.1. Código:	7
4.2. Ejecución del programa:	7
5. Quinto apartado: Dibujar la forma de onda de la señal resultante.	8
5.1. Código:	8
5.2. Ejecución del programa:	8
6. Sexto apartado: Pasarle un filtro de frecuencia para eliminar las frecuencias entre 10000Hz y 20000Hz	9
6.1. Código:	9
6.2. Ejecución del programa:	9
7. Séptimo apartado: Almacenar la señal obtenida como un fichero WAV denominado “mezcla.wav”.	10
7.1. Código:	10
7.2. Ejecución del programa:	10
8. Octavo apartado: Cargar un nuevo archivo de sonido, aplicarle eco y a continuación darle la vuelta al sonido. Almacenar la señal obtenida como un fichero WAV denominado “alreves.wav”.	11
8.1. Código:	11
8.2. Ejecución del programa:	12

Para crear los ficheros *nombre.wav*, *apellido.wav* y *nombreapellido.wav* hemos instalado el sintetizador de voz de software *espeak* y, a continuación, hemos ejecutado los comandos
\$ `espeak "Cristina Maria" -w nombre.wav`,
\$ `espeak "Crespo Arco" -w apellido.wav` y
\$ `espeak "Andres Piqueras Brück" -w nombreapellido.wav`.



```
usuario@usuario-Lenovo-ideapad-310-15IKB: ~  
usuario@usuario-Lenovo-ideapad-310-15IKB:~$ espeak "Cristina Maria" -w nombre.wa  
v  
usuario@usuario-Lenovo-ideapad-310-15IKB:~$ espeak "Crespo Arco" -w apellido.wav  
usuario@usuario-Lenovo-ideapad-310-15IKB:~$ espeak "Andres Piqueras Brück" -w no  
mbreapellido.wav  
usuario@usuario-Lenovo-ideapad-310-15IKB:~$
```

Figura 1: Muestra creación ficheros

1. Primer apartado: Leer dos ficheros de sonido (WAV o MP3) de unos pocos segundos de duración cada uno. En el primero debe escucharse el nombre de la persona que realiza la práctica. En el segundo debe escucharse el apellido.

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

1.1. Código:

```
# ***** librerías *****  
library(tuneR)  
library(seewave)  
library(audio)  
  
# ***** Apartado 1: Leer dos ficheros de sonido (WAV o MP3) de  
# unos pocos segundos de duración cada uno. En el primero debe escucharse  
# el nombre de la persona que realiza la práctica. En el segundo debe  
# escucharse el apellido. *****  
nombre <- readWave('nombre.wav')  
apellido <- readWave('apellido.wav')
```

1.2. Ejecución del programa:

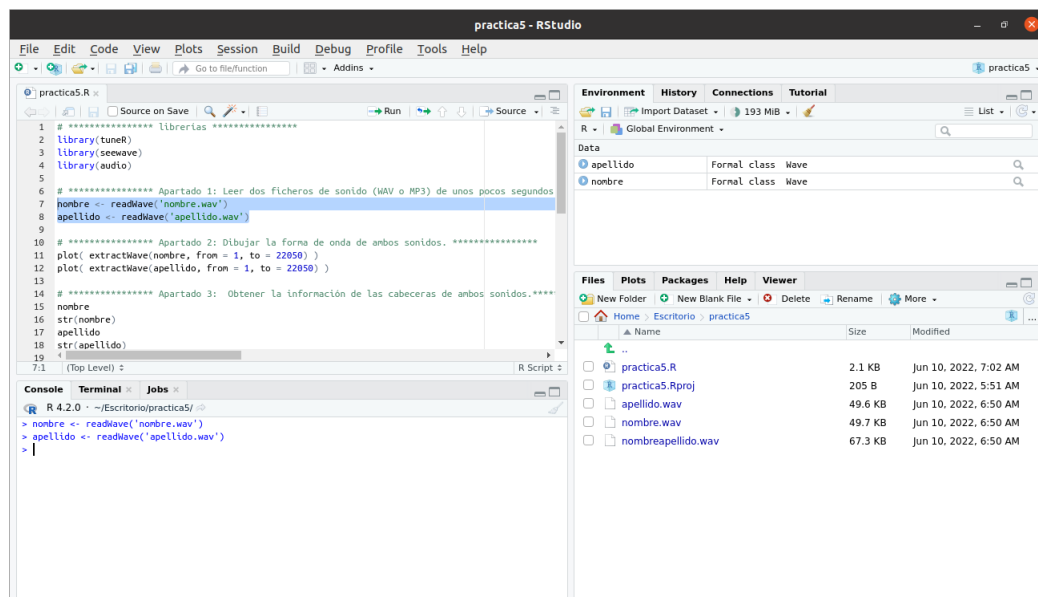


Figura 2: Ejecución del primer apartado

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

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

2.1. Código:

```
# ***** Apartado 2: Dibujar la forma de onda de ambos sonidos.
# *****
plot( extractWave(nombre, from = 1, to = 22050) )
plot( extractWave(apellido, from = 1, to = 22050) )
```

2.2. Ejecución del programa:

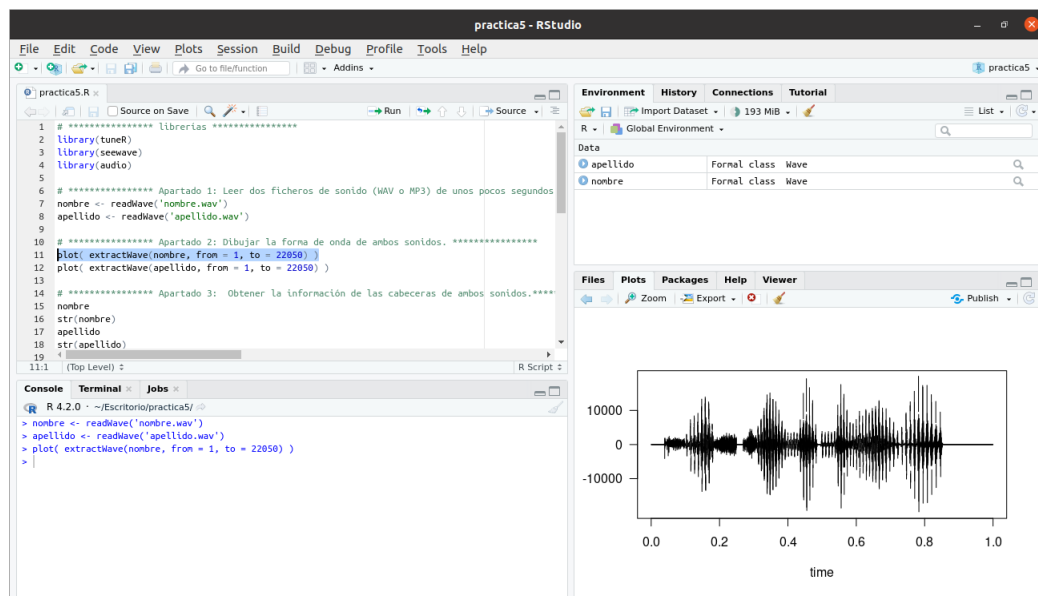


Figura 3: Ejecutar la primera gráfica: *nombre.wav*

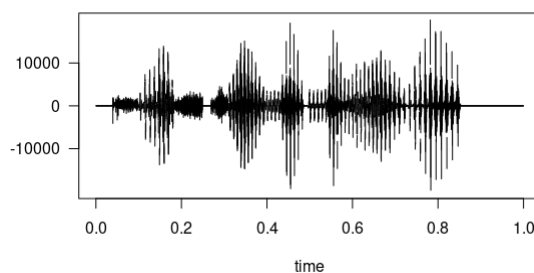


Figura 4: Gráfica del sonido **nombre**

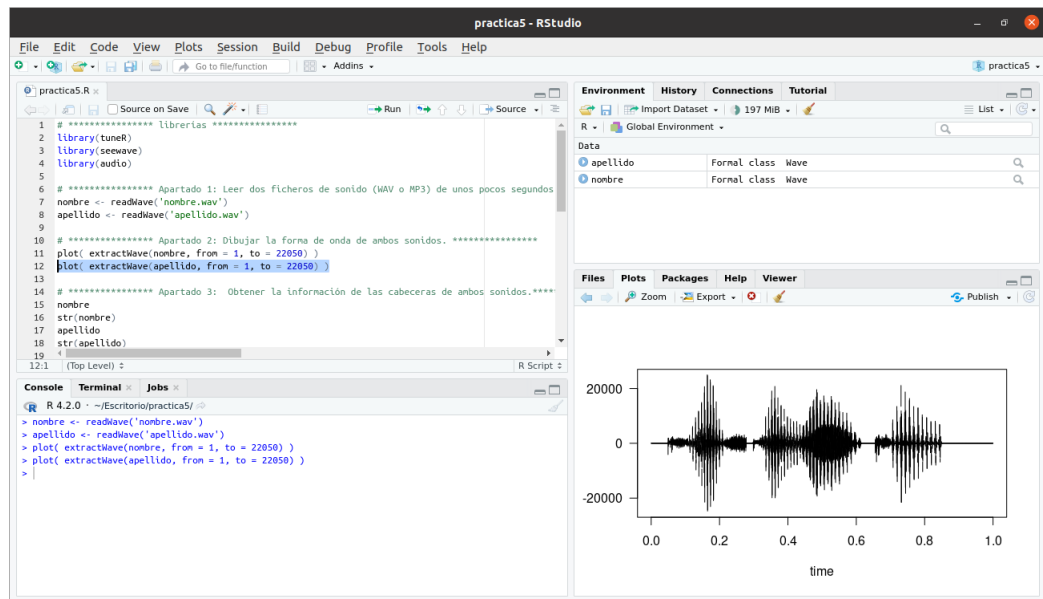


Figura 5: Ejecutar la segunda gráfica: *apellido.wav*

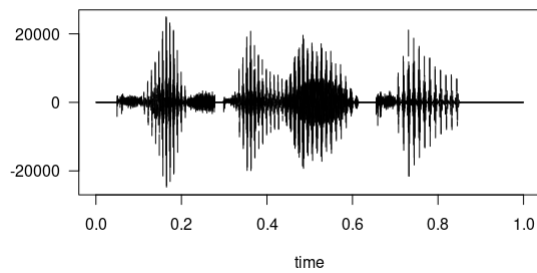


Figura 6: Gráfica del sonido **apellido**

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

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

3.1. Código:

```
# ***** Apartado 3: Obtener la información de las cabeceras de  
# ambos sonidos.*****  
nombre  
str(nombre)  
apellido  
str(apellido)
```

3.2. Ejecución del programa:

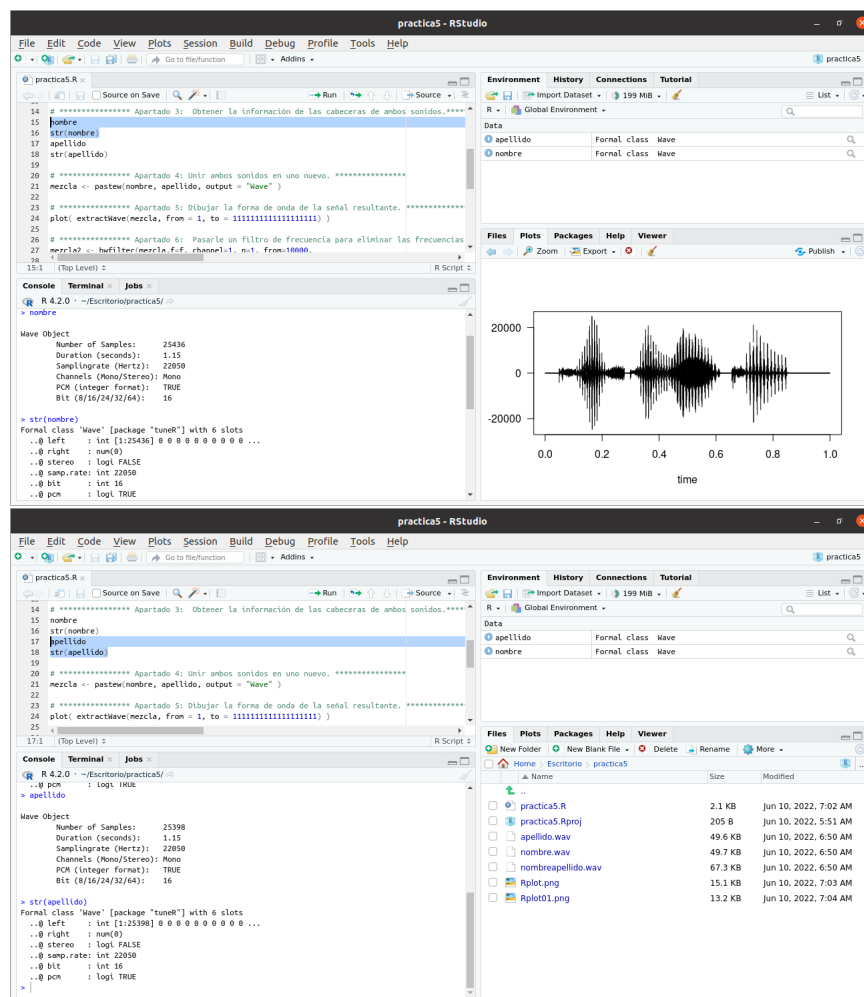


Figura 7: Ejecución del tercer apartado

4. Cuarto apartado: Unir ambos sonidos en uno nuevo.

La función que hemos ejecutado para unir ambos sonidos ha sido `pastew(nombre, apellido, output = "Wave")` y el resultado lo hemos almacenado en la variable “mezcla”.

4.1. Código:

```
# ***** Apartado 4: Unir ambos sonidos en uno nuevo. *****
mezcla <- pastew(nombre, apellido, output = "Wave" )
```

4.2. Ejecución del programa:

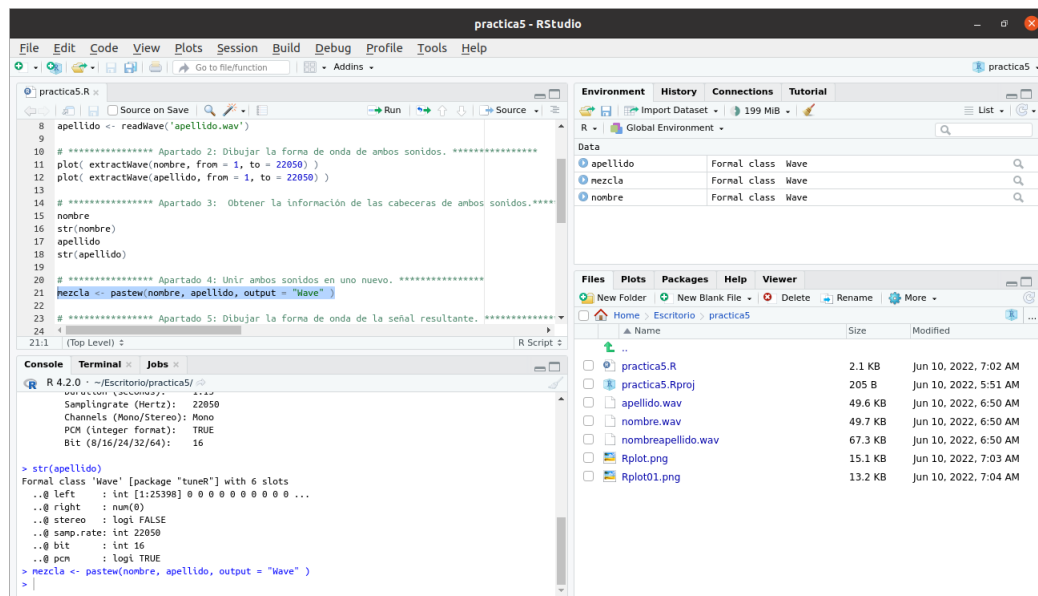


Figura 8: Ejecución del cuarto apartado.

5. Quinto apartado: Dibujar la forma de onda de la señal resultante.

Para dibujar la forma de onda del sonido hemos usado, al igual que en el apartado 2, la función `plot` (`extractWave` (`mezcla`, `from` = 1, `to` = 22050)).

5.1. Código:

```
# ***** Apartado 5: Dibujar la forma de onda de la señal  
# resultante. *****  
plot( extractWave(mezcla, from = 1, to = 22050) )
```

5.2. Ejecución del programa:

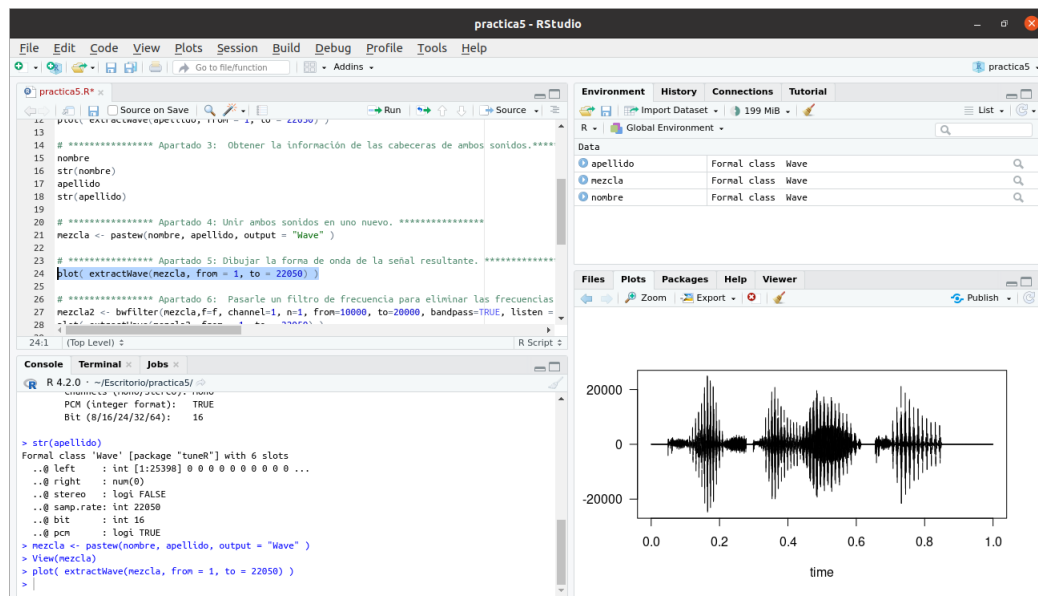


Figura 9: Ejecutar gráfica

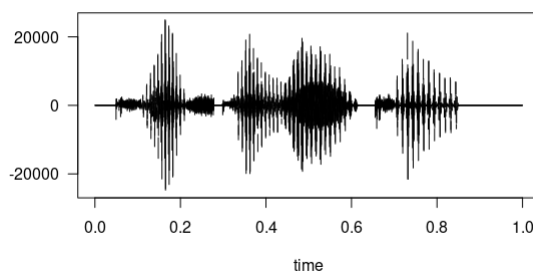


Figura 10: Gráfica del sonido

6. Sexto apartado: Pasarle un filtro de frecuencia para eliminar las frecuencias entre 10000Hz y 20000Hz

Para eliminar las frecuencias entre 10000Hz y 20000Hz hemos implementado la función `mezcla2 <- bwfilter (mezcla,f=f, channel=1, n=1, from=10000, to=20000, bandpass=TRUE, listen = FALSE, output = "Wave")`.

6.1. Código:

```
# ***** Apartado 6: Pasarle un filtro de frecuencia para
# eliminar las frecuencias entre 10000Hz y 20000Hz*****
mezcla2 <- bwfilter(mezcla, f=f, channel=1, n=1, from=10000, to=20000,
                    bandpass=TRUE, listen = FALSE, output = "Wave")
plot( extractWave(mezcla2, from = 1, to = 22050) )
```

6.2. Ejecución del programa:

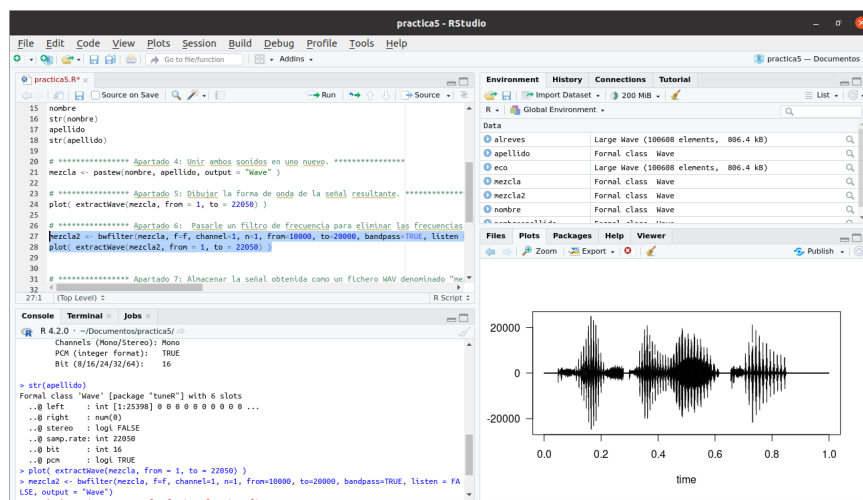


Figura 11: Ejecución del sexto apartado

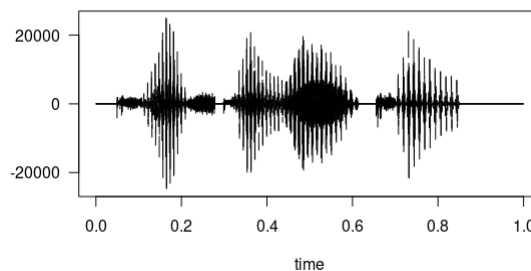


Figura 12: Gráfica del sonido `mezcla2`

7. Séptimo apartado: Almacenar la señal obtenida como un fichero WAV denominado “mezcla.wav”.

Una vez hemos unido los dos sonidos vamos a almacenar el sonido en el fichero *mezcla.wav* mediante la función `writeWave` (`mezcla2`, `file.path("mezcla.wav")`).

7.1. Código:

```
# ***** Apartado 7: Almacenar la señal obtenida como un fichero
# WAV denominado "mezcla.wav". *****
writeWave(mezcla2, file.path("mezcla.wav"))
```

7.2. Ejecución del programa:

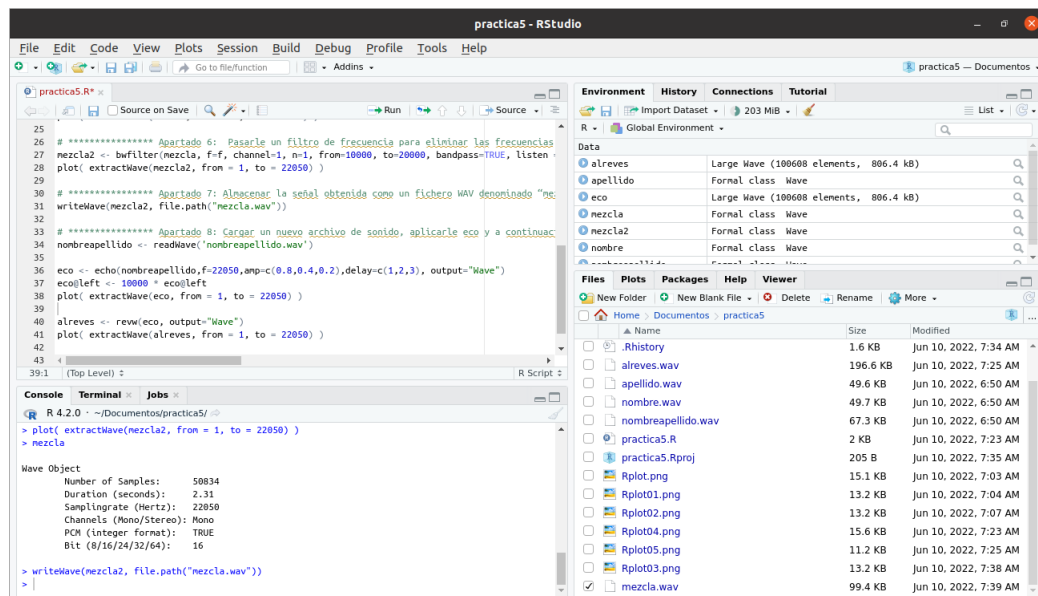


Figura 13: Almacenar sonido en el fichero *mezcla.wav*

8. Octavo apartado: Cargar un nuevo archivo de sonido, aplicarle eco y a continuación darle la vuelta al sonido. Almacenar la señal obtenida como un fichero WAV denominado “alreves.wav”.

Para leer el fichero *nombreapellido.wav* hemos vuelto a usar la función “readMP3 (‘nombreapellido.wav’)”.

A continuación, para generar el eco en el sonido hemos usado la función `eco <- echo (nombreapellido, f=22050, amp=c(0.8,0.4,0.2), delay=c(1,2,3), output=“Wave”)` y hemos ejecutado `eco@left <- 10000 * eco@left`.

Por otro lado, al sonido *eco* que hemos creado le aplicamos la función `alreves <- revw(eco, output=“Wave”)` para invertir el sonido.

Una vez finalizado lo anterior vamos a almacenar el sonido en el fichero *alreves.wav* mediante la función `writeWave (alreves, file.path(“alreves.wav”))`.

8.1. Código:

```
# ***** Apartado 8: Cargar un nuevo archivo de sonido, aplicarle
# eco y a continuación darle la vuelta al sonido. Almacenar la señal
# obtenida como un fichero WAV denominado \alreves.wav". *****
nombreapellido <- readWave('nombreapellido.wav')

eco <- echo(nombreapellido,f=22050,amp=c(0.8,0.4,0.2),delay=c(1,2,3),
            output="Wave")
eco@left <- 10000 * eco@left
plot( extractWave(eco, from = 1, to = 22050) )

alreves <- revw(eco, output="Wave")
plot( extractWave(alreves, from = 1, to = 22050) )

writeWave(alreves, file.path("alreves.wav"))
```

8.2. Ejecución del programa:

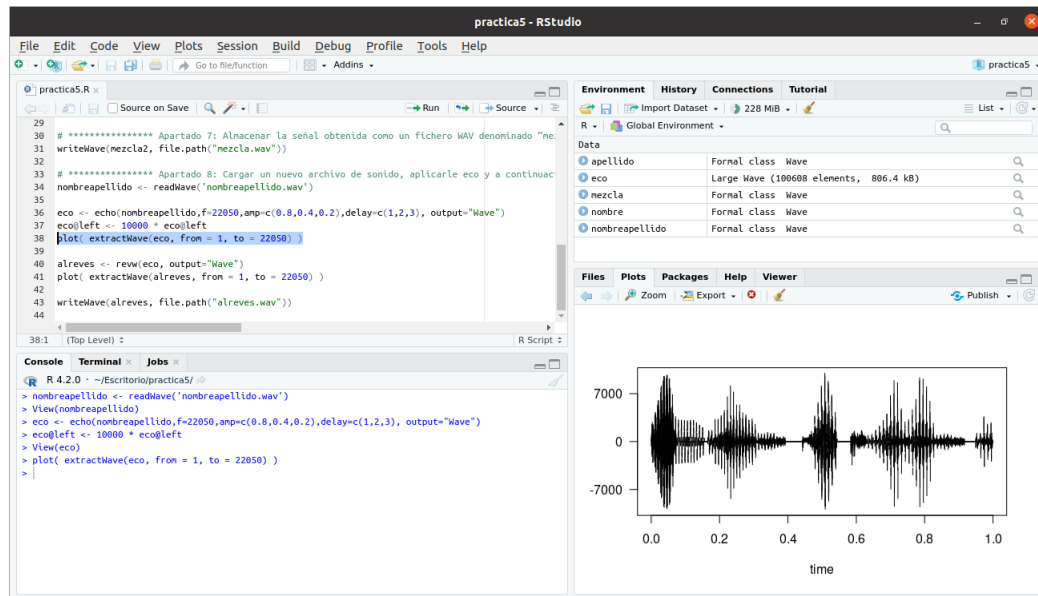


Figura 14: Crear el sonido *eco*.

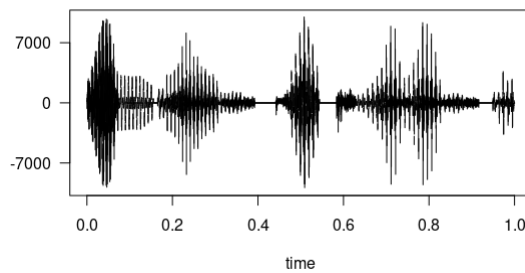


Figura 15: Gráfica del sonido *eco*

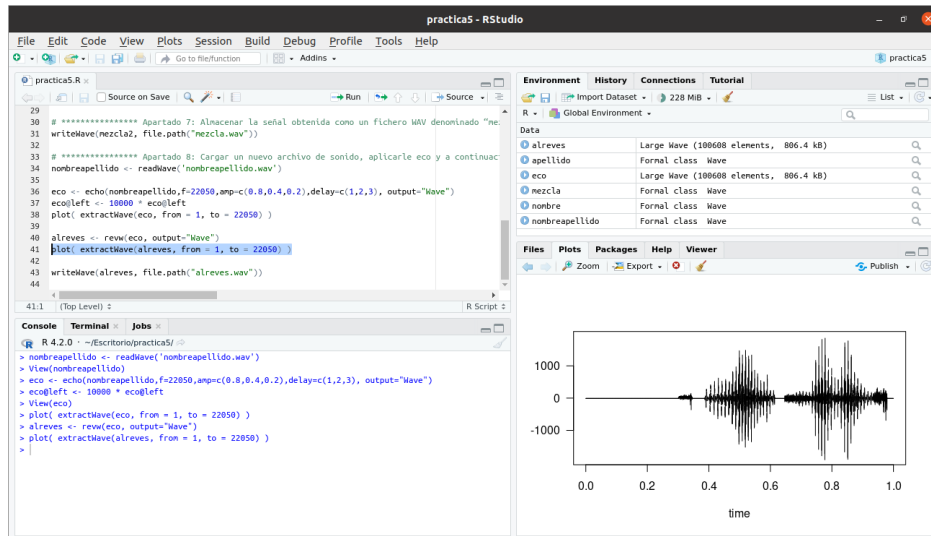


Figura 16: Crear el sonido *alreves*.

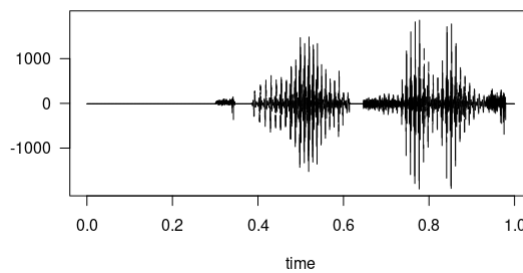


Figura 17: Gráfica del sonido *alreves*

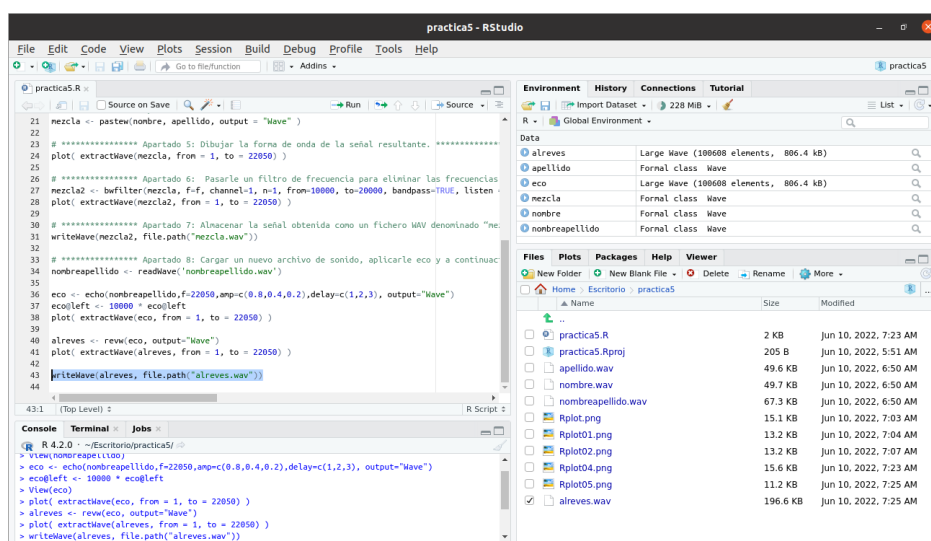


Figura 18: Almacenar sonido en el fichero *alreves.wav*