

# Periféricos y dispositivos de interfaz humana

## Práctica 5



**UNIVERSIDAD  
DE GRANADA**

En esta práctica lo primero que he hecho ha sido crear 2 audios, el primero con mi nombre y el segundo con mi apellido, al hacer esta práctica en windows lo he hecho grabando mi propia voz en y transformar el archivo en .wav.

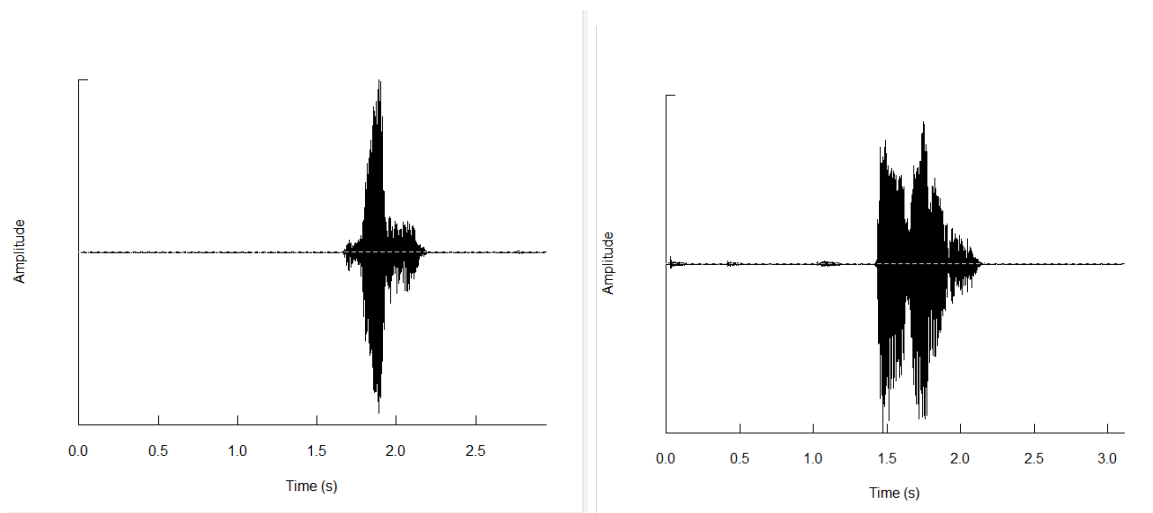
Una vez tenemos el audio debemos abrirlo, para ello debemos establecer el directorio de trabajo con el método `setwd()`, una vez realizado este paso cargamos los archivos *nombre.wav* y *apellidos.wav* en el entorno con la función `readWave()`:

```
# cargar archivos de sonido (wav/mp3):  
nombre <- readwave('nombre.wav')  
nombre  
apellido <- readwave('apellidos.wav')  
apellido
```

Para dibujar la onda de los sonidos he hecho uso de la función `oscillo` de la siguiente manera:

```
# mostrar la onda del sonido:  
oscillo(nombre)  
oscillo(apellido)
```

Esto nos da como resultado las siguientes formas de onda (nombre y apellidos respectivamente):



A continuación debemos obtener información de las cabeceras de con el método `summary`:

```
> summary(nombre)
wave Object
  Number of Samples: 141312
  Duration (seconds): 2.94
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Stereo
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

summary statistics for channel(s):
      Min. 1st Qu. Median      Mean 3rd Qu.  Max.
left -22570    -11      0 -0.0376755    11 24113
right -22570    -11      0 -0.0376755    11 24113

> summary(apellido)
wave Object
  Number of Samples: 149504
  Duration (seconds): 3.11
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Stereo
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

summary statistics for channel(s):
      Min. 1st Qu. Median      Mean 3rd Qu.  Max.
left -18192    -13      0 -0.08623850    13 15338
right -18192    -13      0 -0.08625187    13 15338
```

Lo siguiente que debemos hacer es unir ambos sonidos, pero como podemos apreciar en las ondas de cada audio hay mucho silencio para esto recortamos ambos sonidos con el método *cutw()*, el cual nos permite introducir como parámetro el momento inicial y el final del nuevo audio:

```
nombre_corto <- cutw(nombre, from=1.5,to=2.5, output="wave")
nombre_corto
oscillo(nombre_corto)
listen(nombre_corto)

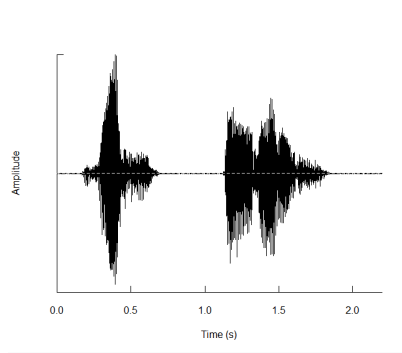
apellido_corto <- cutw(apellido, from=1.3,to=2.5, output="wave")
apellido_corto
oscillo(apellido_corto)
listen(apellido_corto)
```

Para comprobar el correcto funcionamiento del corte también se ha representado la gráfica de cada onda y se ha escuchado el audio con el método *listen()* para obtener el resultado deseado.

Una vez tenemos los audios cortados debemos unirlos para ello hacemos uso de *pastew()* el cual permite juntar dos audios en uno nuevo:

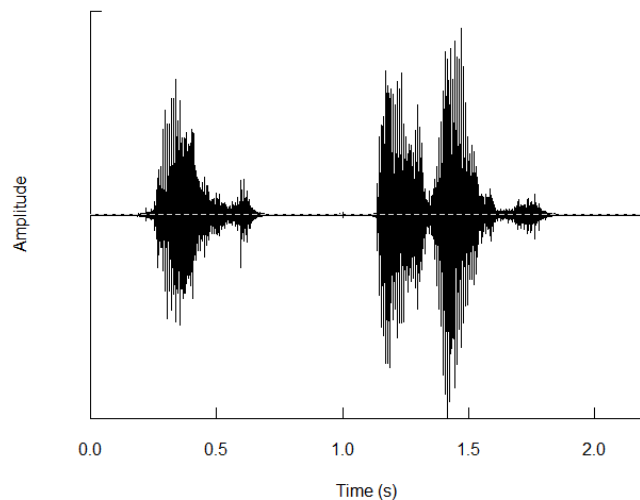
```
union <- pastew(apellido_corto, nombre_corto, output="wave")
oscillo(union)
listen(union)
```

Dando como resultado la siguiente onda:



Ahora le añadiremos un filtro de frecuencia en el que eliminaremos las que estén entre 10000 y 20000 Hz, para esto haremos uso de *bwfilter* el cual nos permitirá eliminar las muestras de dichas frecuencias:

```
#aplicamos filtro de frecuencia
filtro <- bwfilter(union,f=f, channel=1, n=1, from=10000,
                  to=20000, bandpass=TRUE, listen = FALSE, output = "wave")
oscillo(filtro)
listen(filtro)
```



A continuación guardaremos la señal obtenida en un archivo de audio wav. Lo primero que debemos hacer es normalizar el filtro en caso contrario al guardarlo no se escuchará nada, para ello haremos uso de *normalize()*. Una vez normalizado ya podremos almacenarlo de la siguiente forma:

```
#Normalizamos el filtro
nfiltro <- normalize(filtro, unit = "16")

#Guardar sonido
writewave(nfiltro, file.path("mezcla.wav"))
```

Acto seguido abrimos el archivo creado y le añadimos eco con el método `echo()`:

```
#cargamos el archivo  
nom_ap <- readwave('mezcla.wav')
```

Lo  
que

```
nom_ap <- echo(nom_ap, f=f, amp=c(0.8, 0.4, 0.2), delay=c(1, 2, 3),  
               output="wave")
```

último

haremos será darle la vuelta al audio y guardarlo, esto es muy sencillo, solo debemos hacer uso de `revw()`:

```
alreves <- revw(nom_ap, output="wave")  
listen(alreves)  
|  
#Normalizamos el filtro  
alreves <- normalize(alreves, unit = "16")  
  
# Convertir los datos a enteros  
alreves@left <- as.integer(alreves@left)  
  
writewave(alreves, file.path("alreves.wav"))
```