



by **Agustín de Otazua** · Private
Remix of Trabajo Práctico Especial by sys-fiuba

Trabajo Práctico Especial

Demodulación de tonos de discado de teléfono

Nombre: de Otazua, Agustín Fabián

Padrón: 102393

Requisitos para la aprobación

Este trabajo práctico debe ser entregado subiendo al campus en la fecha indicada un archivo comprimido que incluya todo el material relevante. Debe incluir al menos

- Un link al documento de NextJournal publicado. Debe incluir los resultados pedidos con las explicaciones, imágenes, código y gráficos que considere pertinentes.
- Un PDF del informe (puede imprimir la página en pdf)

Verifique que al ejecutar el notebook completo (opción `Run All`), el código corre correctamente y genera las salidas del notebook publicado y entregado.

(Si no lo hace en NextJournal, el archivo comprimido debe contener al menos las respuestas y gráficos solicitados junto el código usado.)

Todos los gráficos deberán tener título, comentarios en ambos ejes sobre la unidad a representar y el eje de abscisas debe estar en unidades de tiempo o frecuencia según corresponda.

Introducción

El sistema de discado por tonos de nuestros teléfonos utiliza los principios de la codificación DTMF, o Dual Tone Multi Frequency. Este sistema de codificación convierte los códigos de información (los 10 dígitos decimales en el caso del discado) en otras tantas señales analógicas, cuya frecuencia debe estar contenida en el rango de las frecuencias de la voz humana, o más bien en el rango de frecuencias del canal telefónico. De este modo se crea un sistema de codificación que puede ser interpretado por todos los sistemas que se conectan a la red telefónica, como (obviamente) teléfonos, pero también, modems, máquinas de fax, centrales telefónicas, hubs, switches, y hasta las distribuidoras de televisión por cable.

La versión de DTMF utilizada en el discado telefónico se conoce con la marca registrada "Touch-Tone" y está estandarizada por la recomendación Q.23 de la ITU-T. El código es muy sencillo y consiste en combinar dos tonos de distintas frecuencias (Dual Tone), de un número de frecuencias distintas (Multi Frequency) pero conocidas y fijas elegidas según se indica en la tabla siguiente:

Código auxiliar

Desarrollo

Será evaluado el día de la fecha de entrega. Esta evaluación será mediante preguntas y problemas con la misma modalidad que las evaluaciones parciales. Se deberá aprobar tanto el trabajo entregado como la evaluación. Esta última puede incluir preguntas sobre:

- Items particulares sobre los ejercicios de esta guía
- Conceptos teóricos necesarios para realizar los ejercicios
- DFT
- Transformada Z

	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

La duración de cada señal para ser considerada un dígito es variable. Cada dígito debe tener una longitud de 70 ms como mínimo, aunque algunos equipos pueden aceptar menor duración de pulso

Utilice la señal `dtmf3.wav`. La señal contiene un código DTMF (convertido a señal discreta mediante un A/D).

0:00 / 0:01

```
x, sr = wavread_mono(dtmf3.wav);
length(x), sr
```

✓ 0.3s

Julia [julia 1.4.2 with pkgs](#)



(13153, 8000.0)

1. Caracterización temporal

Caracterizar y graficar la señal en el tiempo. Escuchar la señal e identificar dentro de la misma la sección donde se realizó el discado. Distinguir las zonas de la señal en donde se están transmitiendo los símbolos y los silencios.

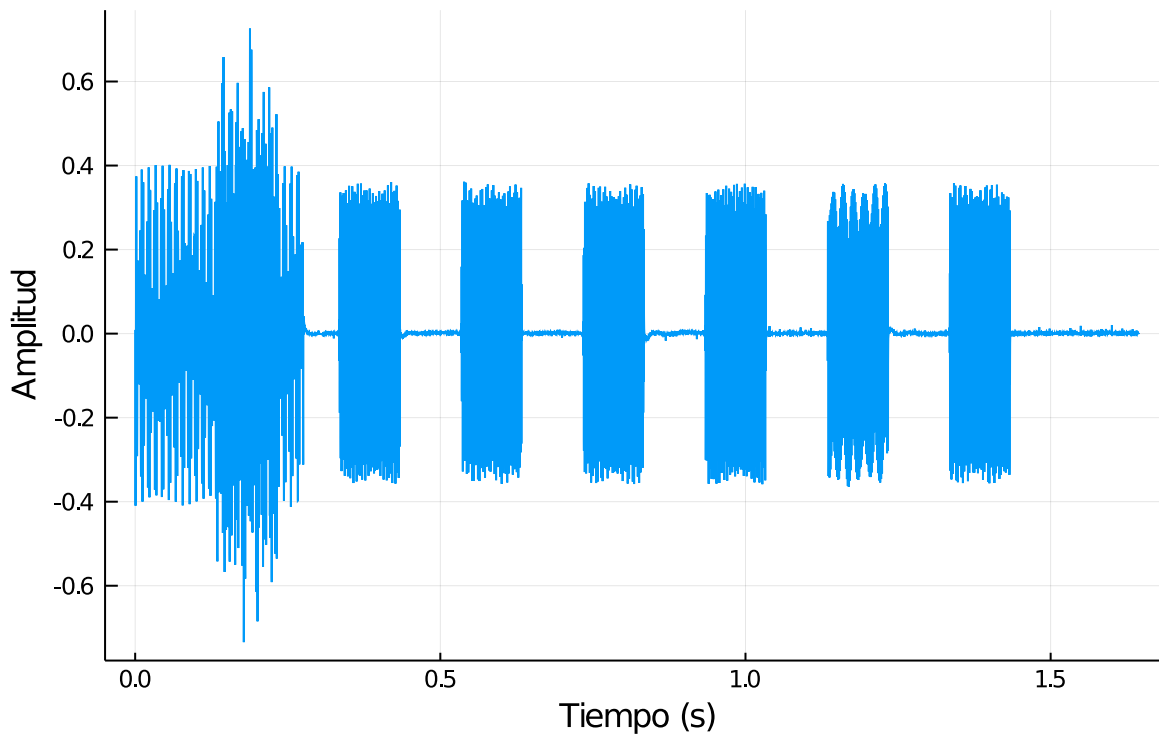
```
t = range(0; step = 1 / sr, length = length(x));
plot(t, x; title="Señal completa", xlabel="Tiempo (s)", ylabel="Amplitud")
```

✓ 0.8s

Julia [julia 1.4.2 with pkgs](#)



Señal completa



La señal consta de siete símbolos, los cuales, al igual que los silencios, tienen una duración de 0,1 s cada uno. En particular, el primer símbolo comienza aproximadamente a los 0,335 s. Por otra parte, desde el comienzo hasta un poco más del primer símbolo, hay un ruido audible pero de frecuencia menor al resto de los tonos.

¿Podría considerarse que las señales del código DTMF corresponden a una sección temporal de una señal periódica infinita? Si es así, identifique la frecuencia fundamental de dicha señal.

Cada símbolo de la señal está caracterizado por la suma de dos armónicos de distinta frecuencia. En principio, cada símbolo podría durar infinito tiempo, dando entonces que sus componentes también tendrán duración infinita. El período de dichos armónicos, en este caso, es un número racional (por ser el inverso de un entero). Asumiendo que la fase de ambos armónicos es cero, al sumarlos darán una nueva señal periódica cuyo período será el mínimo común múltiplo de los períodos de ambas.

2. Caracterización espectral

Analizar y graficar el espectro de los distintos símbolos de la señal en forma individual. Comparar los espectros entre sí.

```
# Grafico todos los espectros de los símbolos
ni = [1080, 2680, 4280, 5880, 7480, 9080, 10680] #índices del comienzo de
cada tono
nf = [1880, 3480, 5080, 6680, 8280, 9880, 11480] #índices del final de cada
tono
l = nf[1] - ni[1] + 1 # longitud de cada tono (igual para todos)
l_mitad = round(Int, l/2) # mitad de longitud
f = range(0; step = π/l_mitad, length = l_mitad)
x_dft = [fft(x[ni[i]:nf[i]]) for i in 1:7];

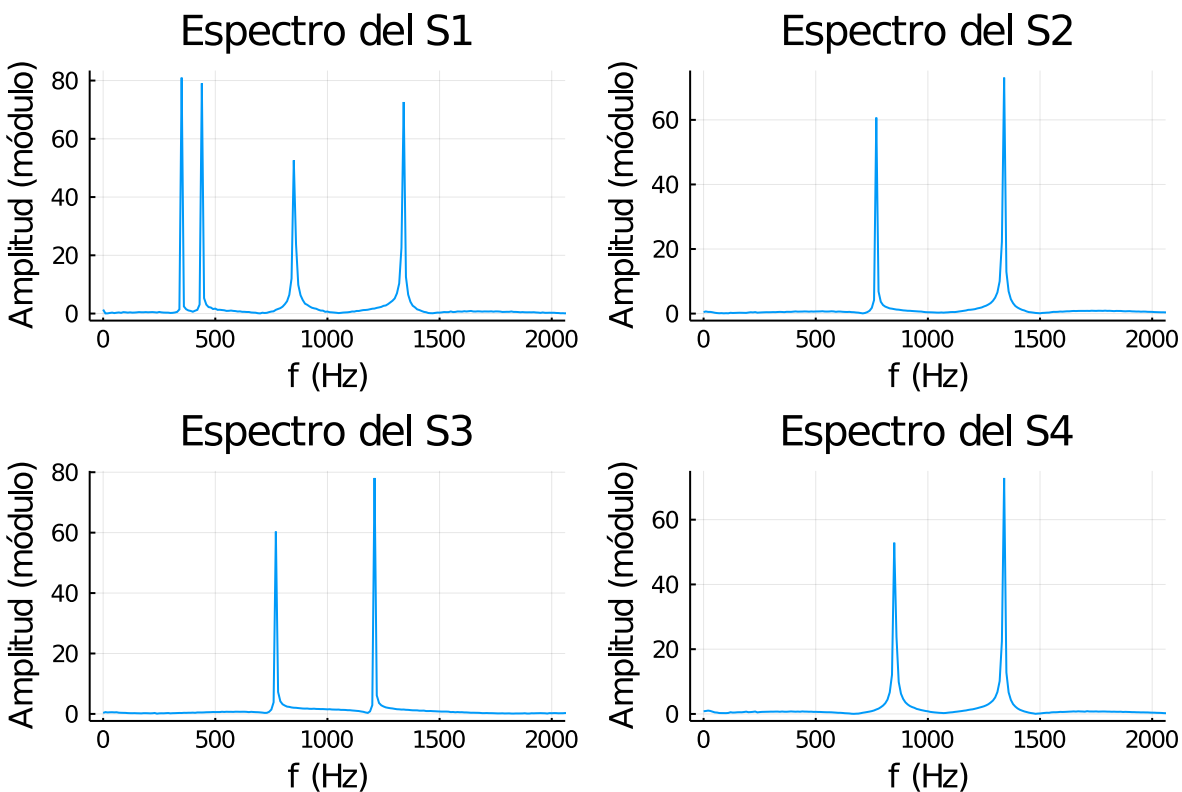
pls = [
    plot(f.*(sr/2/π), abs.(x_dft[i])[1:l_mitad])
    for i in 1:4 ]
```

```

plot(pls...;
  layout = 4,
  title = ["Espectro del S1" "Espectro del S2" "Espectro del S3" "Espectro
del S4"],
  xlab="f (Hz)", ylab="Amplitud (módulo)", xlims = [0, 2000],
)

```

✓ 1.0s

Julia [julia 1.4.2 with pkgs](#)

```

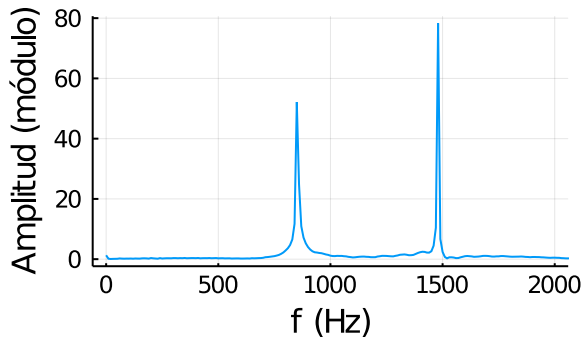
pls = [
  plot(f.*(sr/2/π), abs.(x_dft[i])[1:l_mitad])
  for i in 5:7 ]
plot(pls...;
  layout = 3,
  title = ["Espectro del S5" "Espectro del S6" "Espectro del S7"],
  xlab="f (Hz)", ylab="Amplitud (módulo)", xlims = [0, 2000]
)

```

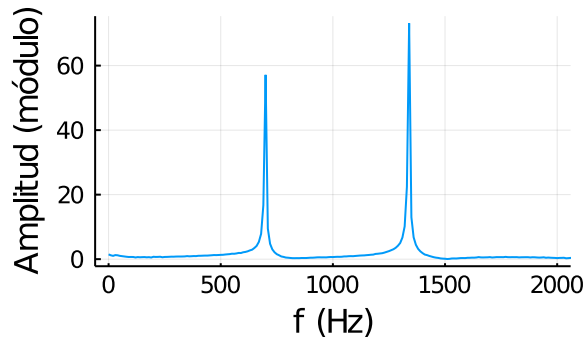
✓ 0.9s

Julia [julia 1.4.2 with pkgs](#)

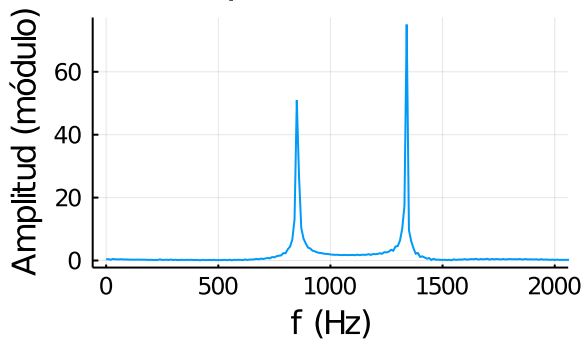
Espectro del S5



Espectro del S6



Espectro del S7



En todos los casos se puede ver que aparecen picos en las frecuencias que componen cada símbolo (S_n , $n=1,\dots,7$). Estos vendrían a ser la aproximación de unas deltas de una señal con armónicos ideales. Además, también se ve que estas frecuencias están en el rango esperado del discado: 697 a 1633 Hz.

En el primer símbolo aparecen dos frecuencias muy cercanas que son las que generan el ruido inicial. Como se dijo anteriormente, estas dos frecuencias no perjudican al análisis del discado ya que están por debajo del ancho de banda del mismo.

Más adelante, en el punto 5, se identificará cada uno de los dígitos.

Realizar la transformada de Fourier del discado completo. ¿Se puede identificar la secuencia a partir de este espectro?

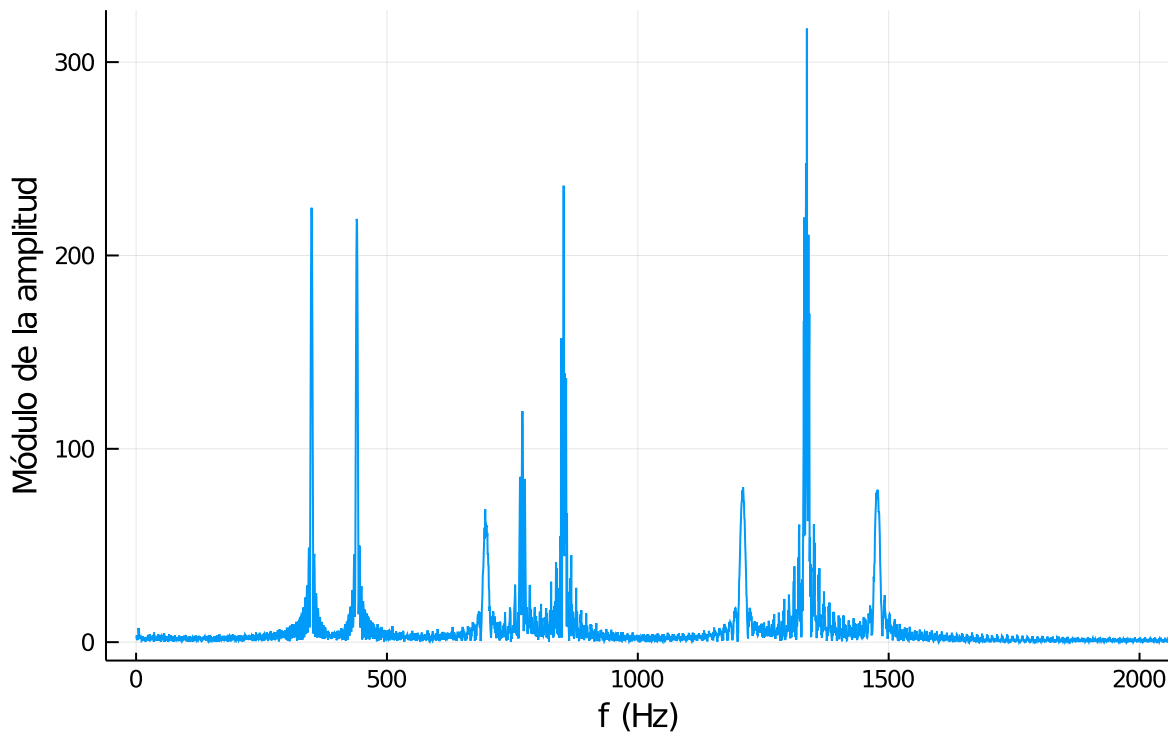
```
l_all = length(x)
l_mit_all = round(Int, l_all/2)
f_all = range(0; step = π/l_mit_all, length = l_mit_all)
x_dft_all = fft(x);
plot(f_all.*(sr/2/π), abs.(x_dft_all)[1:l_mit_all]; title="Espectro de la
señal completa", xlabel="f (Hz)", ylabel="Módulo de la amplitud", xlims = [0,
2000])
```

✓ 0.7s

Julia [julia 1.4.2 with pkgs](#)



Espectro de la señal completa



Teniendo el espectro de la señal completa, se puede saber qué tonos estuvieron presentes, pero no en qué orden. Por lo tanto, no se puede determinar la secuencia de símbolos.

3. Espectrograma

Realizar la transformada de corto tiempo (espectrograma) de la señal y caracterizarla a partir de la misma. Comprobar y mostrar en gráficos el efecto de distintas ventanas (tipo de ventana y duración) en la visualización del espectrograma.

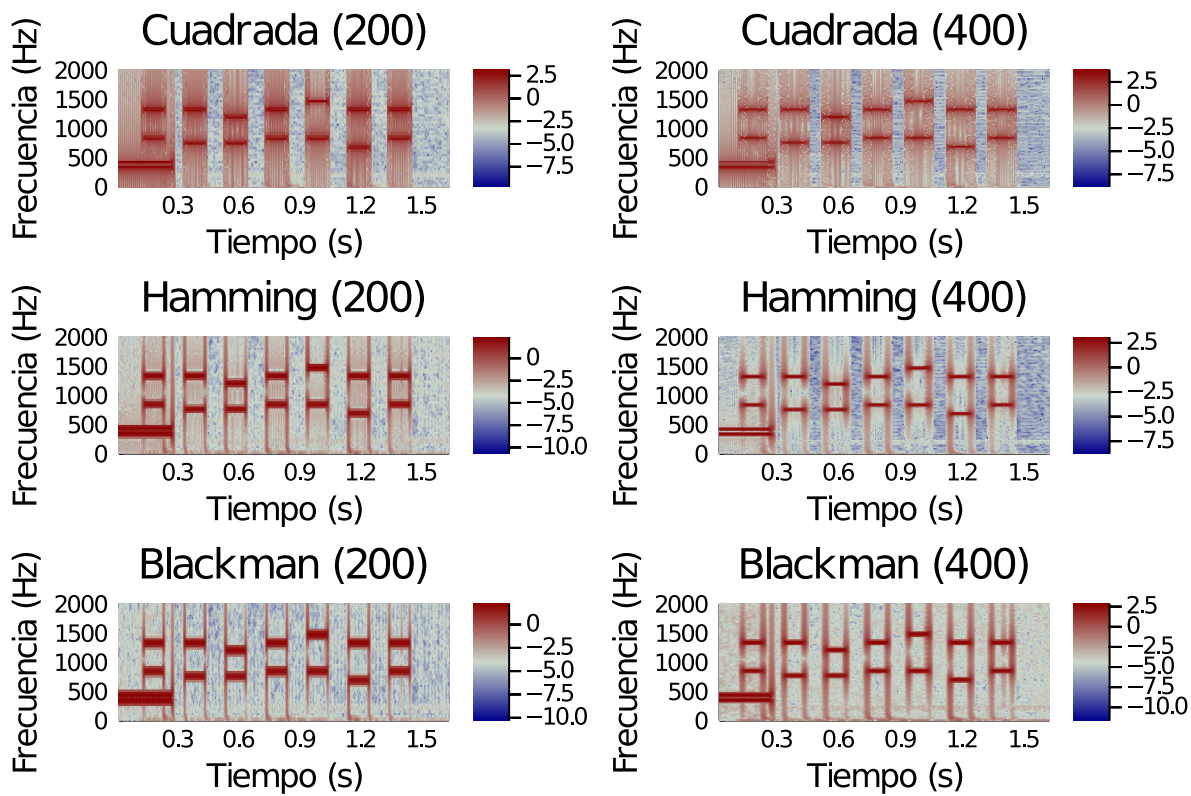
Realizo el espectrograma de la señal mediante tres tipos de ventanas: cuadrada, Hamming y Blackman. A su vez, hago cada una con una duración de 200 y 400 muestras:

```
plot(
  specplot(x; fs=sr, window=rect(200), overlap=0.9),
  specplot(x; fs=sr, window=rect(400), overlap=0.9),
  specplot(x; fs=sr, window=hamming(200), overlap=0.9),
  specplot(x; fs=sr, window=hamming(400), overlap=0.9),
  specplot(x; fs=sr, window=blackman(200), overlap=0.9),
  specplot(x; fs=sr, window=blackman(400), overlap=0.9);
  title = ["Cuadrada (200)" "Cuadrada (400)" "Hamming (200)" "Hamming (400)"
"Blackman (200)" "Blackman (400)"],
  ylims = [0, 2000],
  layout = (3,2)
)
```

✓ 10.1s

Julia [julia 1.4.2 with pkgs](#)





Aquí vuelve a comprobarse que los tonos están compuestos por pares armónicos, simbolizados por pares de barras rojas contrados verticalmente en la frecuencia correspondiente.

Como puede verse, la precisión aumenta con las ventanas de 400 muestras y es aún mayor en la ventana de Hamming. Esto se refleja en el hecho de que, para esta última, las barras rojas tienen la menor altura de todas.

4. Muestreo

¿Cuál es la frecuencia mínima de muestreo requerida para poder trabajar digitalmente con la señal de discado?

Sabiendo que puede haber ruido no deseado de fondo y asumiendo que este tiene un ancho de banda menor al de discado, lo más seguro para mostrar es aplicar el teorema de Nyquist. En este caso, el tono de mayor frecuencia es 1633 Hz, por lo que la frecuencia de muestreo tendrá que ser mayor al doble de esta. Esto equivale a una frecuencia mayor a 3266 Hz.

Sabiendo que las señales discretas se obtuvieron muestreando las originales con $F_s = 8000 \text{ Hz}$, calcular los valores para las frecuencias DTMF presentes en el campo discreto.

Las frecuencias DTMF (f_i) pueden hallarse mediante la siguiente relación:

$$\Omega_i = 2\pi \text{ rad} \cdot f_i / 8000 \text{ Hz}$$

```
f = [697, 770, 852, 941, 1209, 1336, 1477, 1633]
Ω = 2*pi/8000 .*f
```

✓ 0.8s

Julia [julia 1.4.2 with pkgs](#)

8-element Array{Float64,1}:

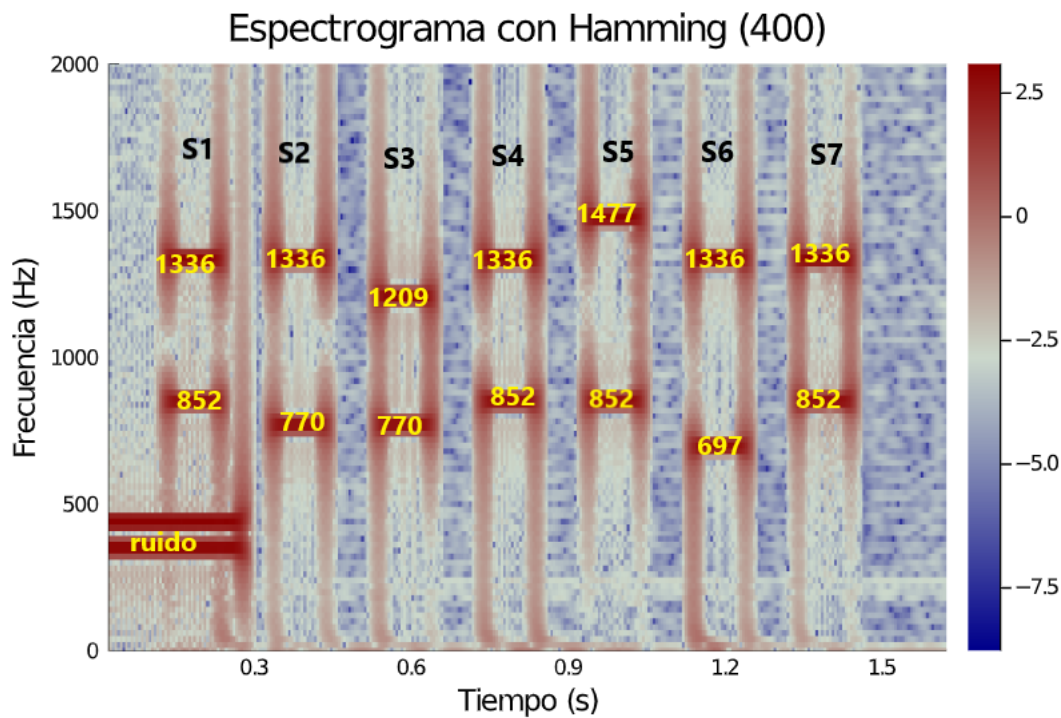
```
0.547423
0.604757
0.669159
0.73906
0.949546
```

1.04929
1.16003
1.28256

5. Detección de dígitos

Utilizar el espectrograma y la transformada de Fourier para encontrar la secuencia de dígitos marcada por inspección visual. Mostrar en los gráficos.

Grafico el espectrograma con una ventana de Hamming de 400 muestras, ya que es el que mayor precisión me daba de los analizados previamente, y marco encima los valores en Hz de las frecuencias graficadas:



Teniendo los pares de frecuencias de cada símbolo, obtengo la siguiente secuencia de dígitos: **8548928**.

Mostrar la ventana de menor N con la que se puede todavía discriminar las 2 frecuencias del primer dígito. ¿Cuál es la ventana de mayor N que se puede utilizar? Mostrar los espectrogramas o las DFT para justificar.

Para este análisis, consideraré una ventana cuadrada de longitud N ya que me da criterios sencillos para analizar.

En primer lugar, busco el menor N (N_{\min}) con el que se puede todavía discriminar las 2 frecuencias del primer dígito. Para ello, el ancho del lóbulo principal ($\Delta\Omega$) tiene que ser comparable a la separación de las dos frecuencias que componen el primer dígito, que es igual a $\Delta\Omega_1 = 0,38$ rad ($\Delta f_1 = 485$ Hz). Considerando que $\Delta\Omega = \Delta\Omega_1 / 2 = 0,19$ rad) y considerando que el ancho del lóbulo principal se aproxima como $\Delta\Omega = 4\pi / N_{\min}$, puedo hallar que $N_{\min} = 4\pi / \Delta\Omega \cong 66$.

Graficando el espectrograma con un overlap de 0.7 para que se vea menos difuso, queda:

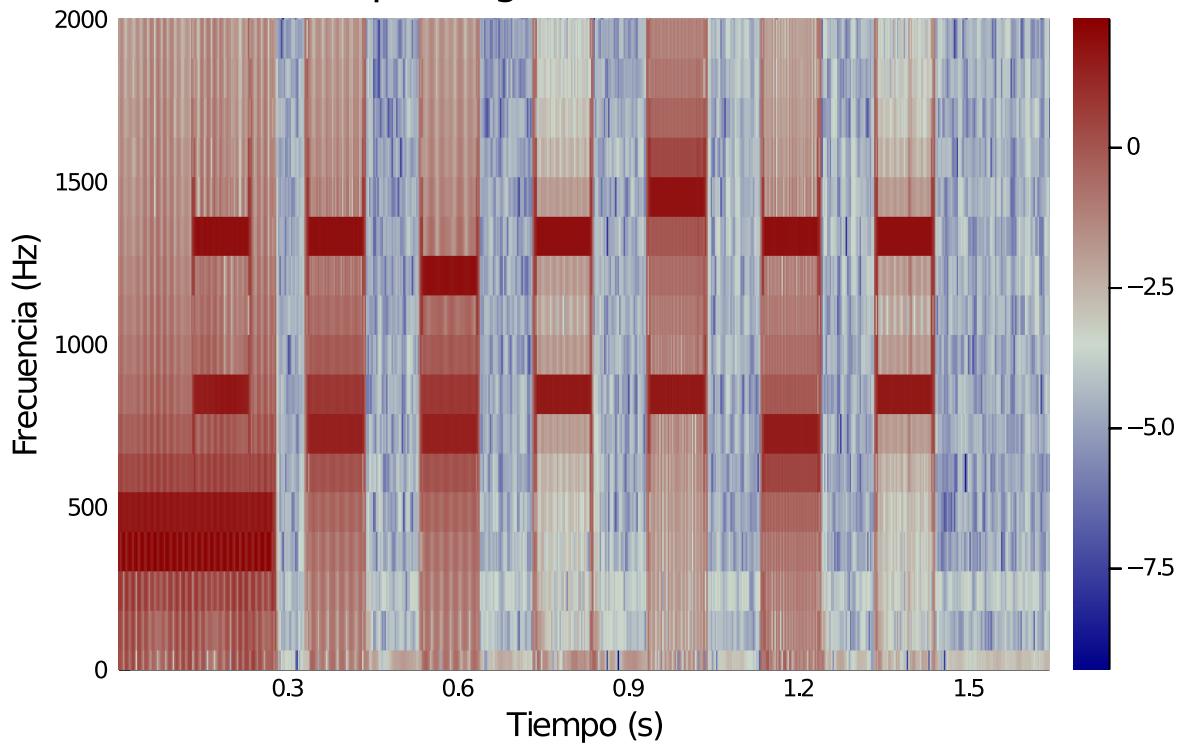
```
specplot(x; fs=sr, window=rect(66), overlap=0.7 , ylims = [0, 2000], title =
"Espectrograma con N = 66")
```

✓ 2.0s

Julia [julia 1.4.2 with pkgs](#)



Espectrograma con N = 66



En segundo lugar, busco el N máximo (N_{\max}) que se pueda utilizar. El problema ocurre cuando la ventana se vuelve tan grande que empieza a encerrar a dos dígitos distintos que, dicho de otro modo, es cuando su longitud es mayor a la longitud del silencio. Esto genera que dos dígitos consecutivos se superpongan y se complique diferenciarlos, a menos que se tenga referencia de la duración de cada uno.

Dado que todos los silencios tienen una duración de 0,1 s y la frecuencia de muestreo es de 8000 Hz, su longitud será de 800 muestras. Tomando ese valor como el ancho máximo límite N_{\max} , puedo graficar el espectrograma correspondiente:

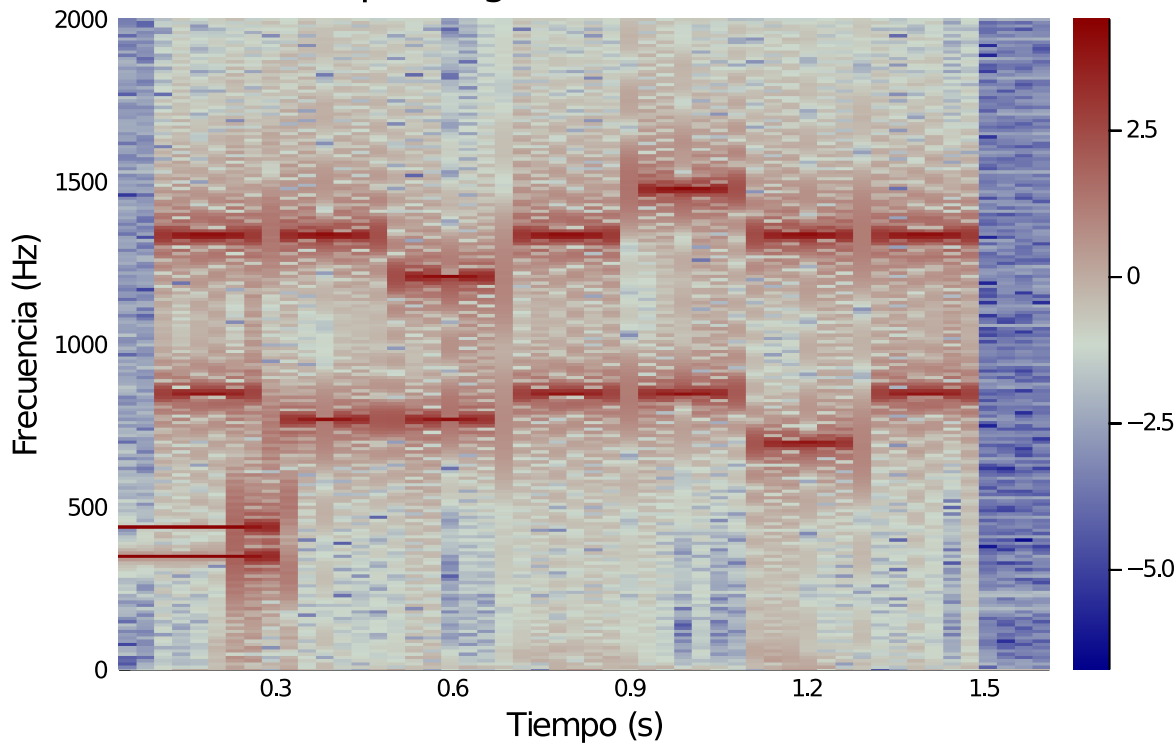
```
specplot(x; fs=sr, window=rect(800), overlap=0.7 , ylims = [0, 2000], title =
"Espectrograma con N = 800")
```

✓ 1.8s

Julia [julia 1.4.2 with pkgs](#)



Espectrograma con $N = 800$



Uno de los métodos que podría utilizarse para decodificar automáticamente los tonos consiste en un banco de filtros pasabanda centrados en cada una de las frecuencias del sistema DTMF. Diseñe el filtro que corresponde a una de las frecuencias del primer tono de la señal por el método de ventaneo. Justificar la elección del ancho de banda, la longitud, el tipo de ventana.

En este punto busco diseñar un filtro pasabanda que corresponda a una de las frecuencias del primer tono de la señal por método de ventaneo. Para ello, centro el filtro en $f_0 = 852 \text{ Hz}$ ($\Omega_0 = 0,213 \pi \text{ rad}$) ya que allí está la señal a recuperar.

Como ancho de banda, busco que el filtro no me recupere también otras frecuencias presentes en la señal, donde las más cercanas serían de 770 Hz y 941 Hz. Entonces propongo un ancho de banda de 42 Hz ($0,0105 \pi \text{ rad}$) que está dentro del rango aceptado para que lo anterior no ocurra.

Dado que el filtro ideal sería anticausal por tener infinitas componentes distintas de cero para $n < 0$, se utiliza una ventana que lo acote. En este caso utilizo una ventana de Hamming, en contraposición a una cuadrada, ya que la relación entre el lóbulo secundario con respecto al primario es mucho menor que en el caso de una cuadrada. A su vez, cuanto mayor sea su longitud menor será esta relación, por lo que termino eligiendo una longitud de 301 muestras, que es menor a la longitud de cada tono (800).

```
h_ideal(n) = 2 * 0.0105 * sinc(0.0105 * n) * cos(pi*0.213*n)
order = 301
ns = range(-(order - 1) / 2; length=order)
winds = hamming(order)
hs = h_ideal.(ns) .* winds

plot(0:order-1, hs; xlabel="Muestras", ylabel="Amplitud", title="Filtro de f = 852 Hz")
```

✓ 0.7s

Julia julia 1.4.2 with pkgs



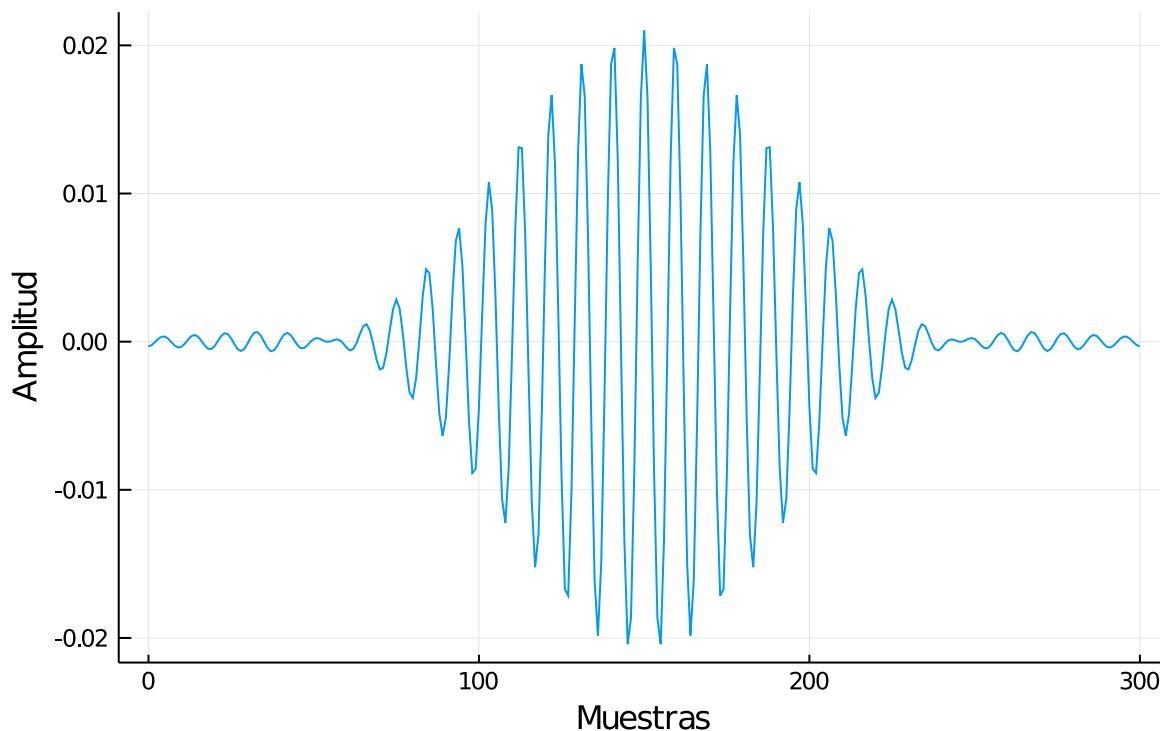
Filtro de $f = 852$ Hz

Grafico el módulo y fase de la transformada del filtro ($H(\Omega)$):

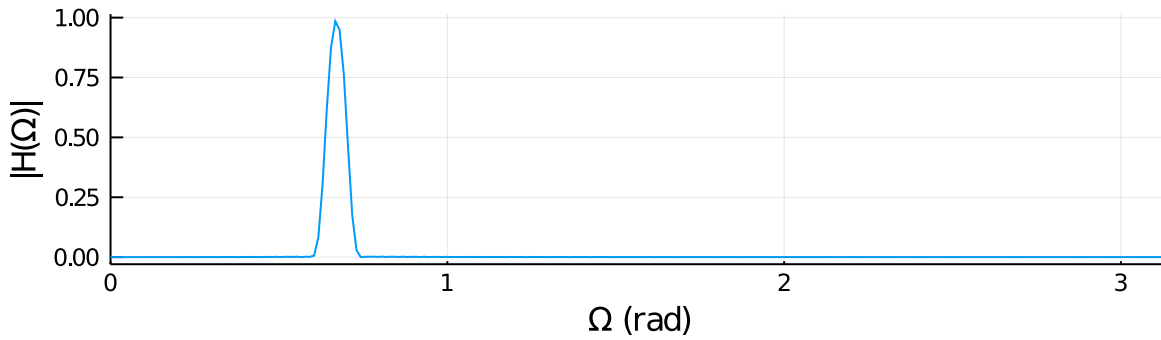
```
plot(
  plot(range(0; stop=2pi, length=500),
    abs.(fft(padright(hs, 500)))); xlims=(0, pi), title="Módulo de H",
  xlab="Ω (rad)", ylab="|H(Ω)|"),
  plot(range(0; stop=2pi, length=500),
    unwrap(angle.(fft(padright(hs, 500)))); xlims=(0, pi), title="Fase de
H", xlab="Ω (rad)", ylab="arg(H(Ω))");
  layout=(2, 1)
)
```

✓ 1.6s

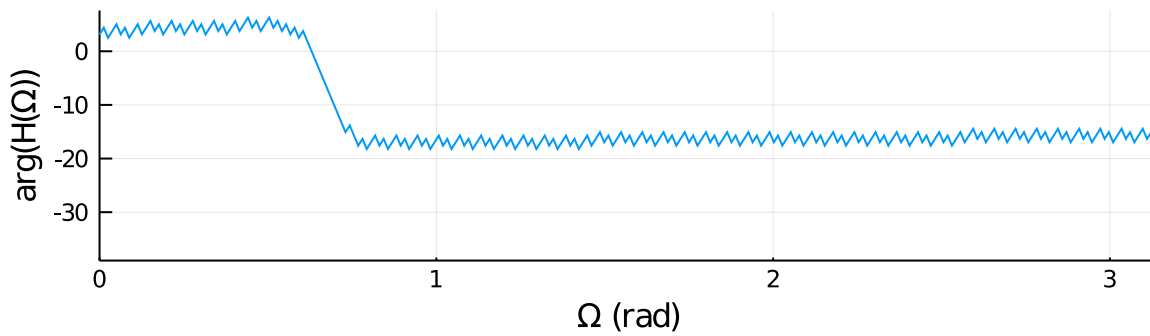
Julia [julia 1.4.2 with pkgs](#)



Módulo de H



Fase de H



En el diagrama de módulo de H se ve claramente el pasabandas centrado en la frecuencia a filtrar. Su anchura angosta se debe a las restricciones de que no filtre las frecuencias DTMF más cercanas. Por otro lado, del diagrama de fase se concluye que existe un corrimiento de fase entre las frecuencias menores y mayores a la de corte. Esto no importa ya que la señal que se quiere obtener está compuesta, idealmente, por una sola frecuencia.

Realizar el diagrama de polos y ceros. Y mostrar cuales son los coeficientes del mismo para filtrar con la función `filt`.

Dado que se trata de un filtro FIR, los coeficientes del numerador serán los elementos de la respuesta al impulso `hs` y no habrán polos.

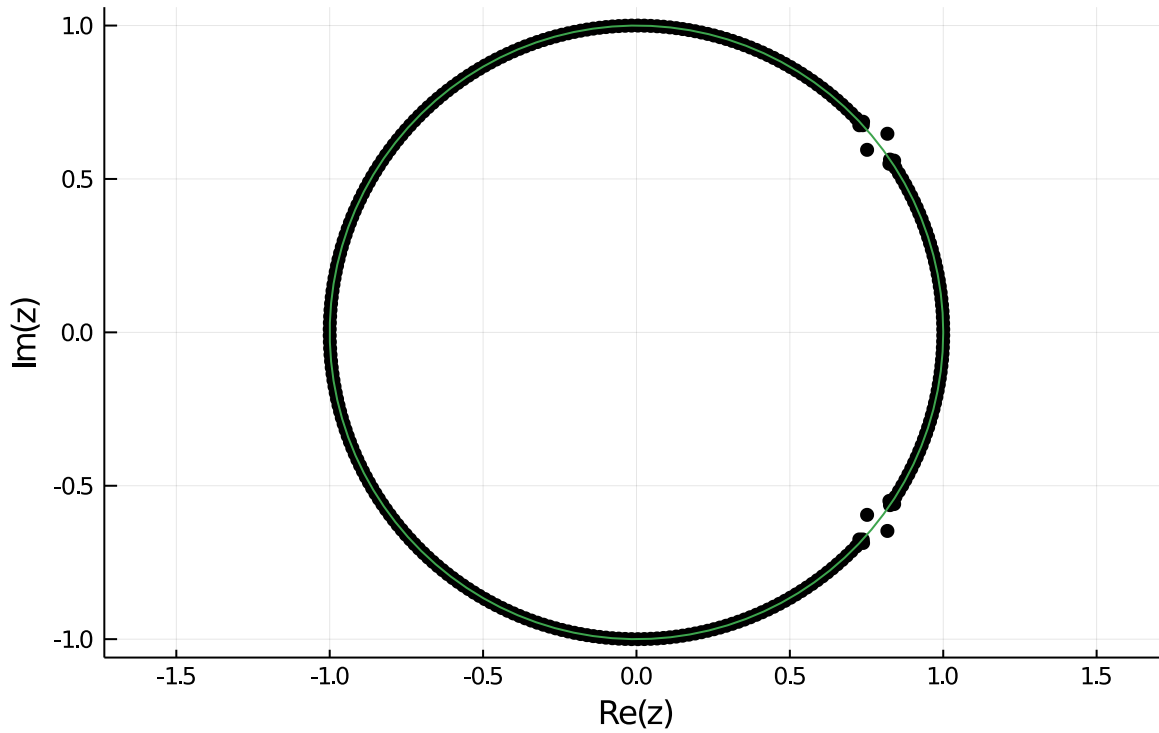
```
pr = polynomialratio(hs,[1])
zplane(getzeros(pr), []; title="Diagrama de polos y ceros", xlabel="Re(z)",
ylab="Im(z)")
```

✓ 0.9s

Julia [julia 1.4.2 with pkgs](#)



Diagrama de polos y ceros



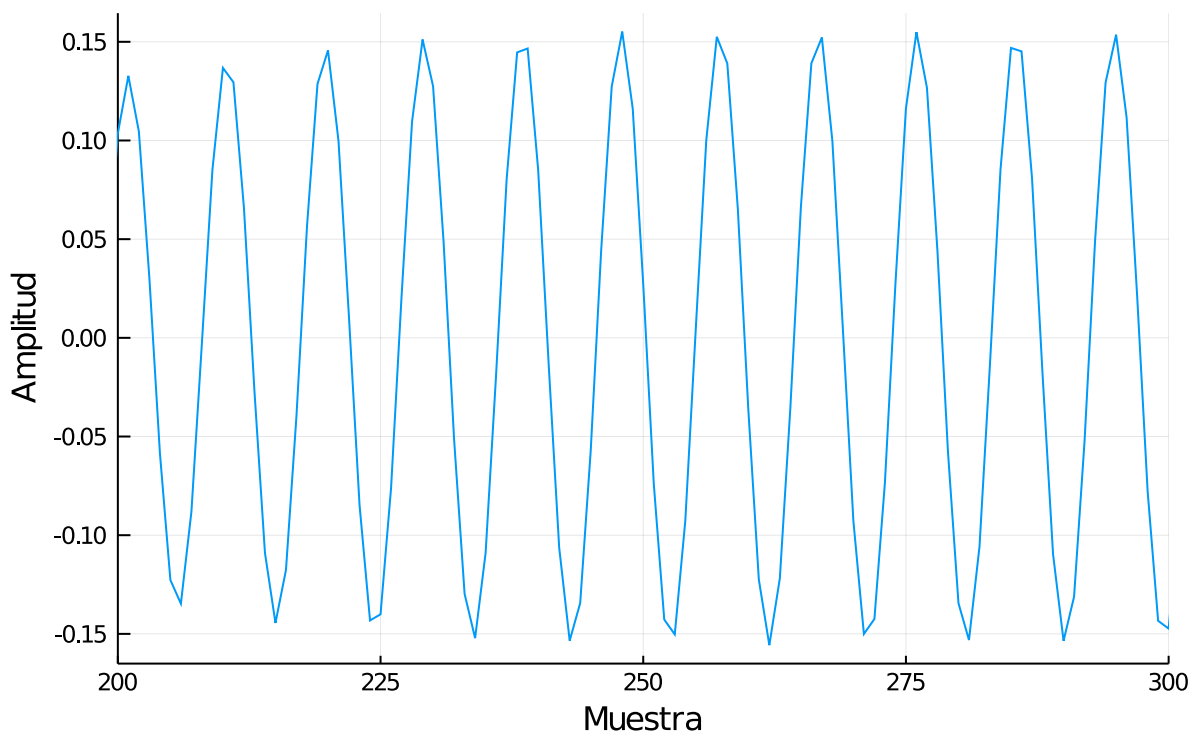
Pongo a prueba el filtro usando la función `filt` y pasándole como señal a filtrar el primer símbolo. Graficando una parte de la respuesta para que se vea bien el armónico:

```
plot(filt(pr, x[ ni[1] : (nf[1] ) ]]); xlims=(200, 300), xlab="Muestra",
ylab="Amplitud", title="Sección de S1 filtrado")
#no supe cómo representar en tiempo en vez de muestras, al usar la función
filt
```

1.1s Shift+Enter to run

Julia [julia 1.4.2 with pkgs](#)

Sección de S1 filtrado



Ahora se aprecia cómo desaparecieron las demás componente de frecuencia (la frecuencia DTMF y el ruido) y quedó únicamente el armónico de $f = 852$ Hz.

¿Tiene fase lineal?

Dado que el filtro es FIR y se puede expresar como un polinomio de grado finito (ya que hs tiene longitud finita), entonces se puede afirmar que el filtro será de fase lineal.

[+ ADD NEW CONTENT](#)

► Runtimes (1)