

[Área personal](#) / [Mis cursos](#) / [SMC 21028](#) / [Práctica 5: audio digital: lectura, descripción y generación](#)
/ [Cuestionario de la primera sesión P5 \(P\)](#).

Comenzado el	miércoles, 4 de marzo de 2020, 11:11
Estado	Finalizado
Finalizado en	miércoles, 4 de marzo de 2020, 12:54
Tiempo empleado	1 hora 43 minutos
Puntos	9,0/12,0
Calificación	7,5 de 10,0 (75%)

Información

Sonido digital: representación, descripción y generación

Vamos a trabajar con los sonidos digitales, como los vamos a crear y manipular el resto del curso. Trabajaremos el formato de audio no comprimido, las muestras y sus valores, el análisis de sonidos existentes y la creación de señales sonoras nuevas. Todo ello mediante programas en Csound.

Información

Parámetros del bloque de formato de un fichero WAV

Las siguientes preguntas están basadas en un programa en Csound que vas a programar para analizar el **bloque de formato** de un fichero WAV, para conocer los valores de los parámetros que caracterizan la señal almacenada.



En el [material previo](#) de la práctica están los operadores Csound de lectura del bloque de formato para hacer el instrumento (de nombre **format**) que lo lea. Sigue los siguientes pasos para su diseño:

- Descarga el fichero que contiene la [PLANTILLA](#) para crear el proyecto de este ejercicio.
- Se proporciona la estructura del instrumento **format**, a falta de que completes lo que falta en el código para que funcione correctamente.
- Este programa en Csound ya trae programada la lectura de los siguientes parámetros de formato:
 - La frecuencia de muestreo del fichero que se indique, con **filesr**
 - El número de canales del fichero, con **filenchnls**
 - La resolución en bits de las muestras fichero, con **filebit**
 - La duración en segundos del fichero, con **filelen**
- Con estos valores, podemos calcular los que faltan:
 - Número de Bytes por muestra = número de canales × resolución en bits / 8
 - Flujo = Frecuencia de muestreo × Número de Bytes por muestra
 - Número de muestras que tiene la señal = Duración × Flujo / Número de Bytes por muestra
- El programa imprimirá en consola, con dos operadores **print**, primero los valores leídos y luego los calculados.
- Programa el instrumento.
- **Completa la partitura** añadiendo a la orden de activación el nombre del fichero que descargues:

```
i "format" 0 0.1 "nombre del fichero"
```

y usa el programa para rellenar los valores de los parámetros de formato de los ficheros que se piden en las preguntas siguientes (todos los valores son enteros salvo la duración en segundos, con coma y 3 decimales).

Pregunta **1**

Finalizado

Puntúa 1,0 sobre 1,0

- Descarga el siguiente fichero WAV de prueba:

0:00 / 0:00

. Escribe los valores de sus parámetros de formato:

Núm. canales	fs (muestras/s)	Resolución (bits)	Bytes/muestra	Flujo (Bytes/s)	Duración (s ± 0,001)	Tamaño (muestras)
1	11025	16	2	22050	1,500	16538

Pregunta **2**

Finalizado

Puntúa 1,0 sobre 1,0

- Descarga el siguiente fichero WAV de prueba:

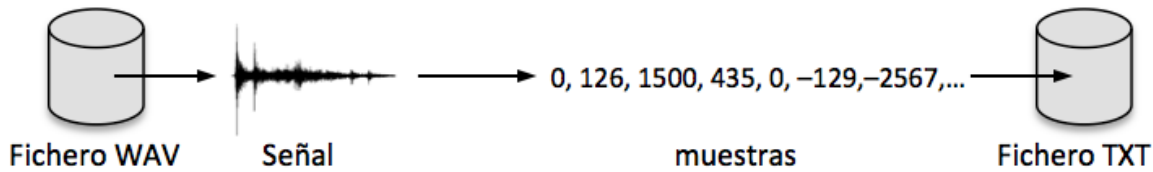
0:00 / 0:00

. Escribe los valores de sus parámetros de formato:


Núm. canales	fs (muestras/s)	Resolución (bits)	Bytes/muestra	Flujo (Bytes/s)	Duración (s ± 0,001)	Tamaño (muestras)
1	48000	24	3	144000	2,499	119972

Información a partir de los valores de las muestras

Se implementará un programa en Csound para leer las muestras del bloque de datos de un fichero WAV y escribir sus valores en un fichero de texto.

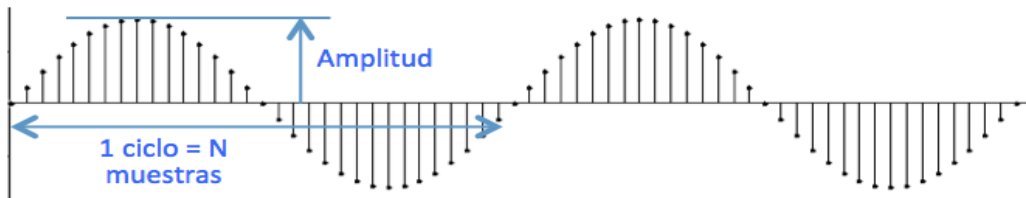


Para ello, sigue las siguientes instrucciones.

- Descarga la [PLANTILLA](#) para este ejercicio. Incluye instrucciones para el instrumento y la partitura para usarlo.
- **PRIMER PASO:** Cuando te bajes el fichero WAV de la pregunta, debes averiguar su frecuencia de muestro (para la variable **sr** de la cabecera) y su duración (para el tiempo de activación del instrumento en la partitura). Para ello, abre el fichero con Audacity:
 - En la esquina inferior izquierda encontrarás el valor de la frecuencia de muestreo en una ventana.
 - Pulsa el botón "ir al final"  Abajo, donde se indica "Posición de audio", te dirá la duración del fichero.
- **CUANDO TERMINES ESTO:** Cierra **Audacity** y abre la plantilla con **CsoundQt**.
 - En la cabecera, añade el valor de frecuencia de muestreo a la declaración de **sr**.
 - En la partitura, añade el valor de duración en la activación del instrumento.
- El instrumento constará de 3 instrucciones:
 - Leer la señal con **soundin** (salida a variable de tipo a-)
 - Pasar cada muestra de la señal anterior a una variable (de tipo k-).
 - Escribir el valor de cada una de estas muestras en el fichero de texto con **fprintks**, siguiendo esta estructura:
 - **fprintks** Variable_con_el_nombre_del_fichero_de_salida, "%d\n", Variable_con_el_valor_de_la_muestra
- **Cuando tu programa esté listo**, ejecútalo para contestar a las preguntas del cuestionario sobre la frecuencia y amplitud del contenido del fichero descargado (debes renombrarlo a "ej5.2-in.wav").

Procedimiento para la medición

- Para calcular la frecuencia de la onda abre el fichero de texto generado con un editor de texto, para ver los valores de las muestras. Esos valores representan lo que se puede ver en la siguiente figura:



- Busca el segundo cruce por cero (será el principio de un segundo ciclo de la onda). Cuenta el número de muestras *N* que forma el ciclo (sin contar la que ya ha cambiado de signo por segunda vez) y usa ese dato, junto con la frecuencia de muestreo, para calcular la frecuencia.

Recuerda: si 1 periodo (1 ciclo) es $T_0 = N$ muestras $\rightarrow T_0 = N T_s = N / f_s \rightarrow F_0 = f_s / N$

- Con los valores de ese primer ciclo, calcula la amplitud de pico de la onda en dBFS (ver figura).

Recuerda: dBFS de pico = $20 \log_{10} (A_{\text{pico}} / 32768)$ en el caso de 16 bits/muestra.

(puedes usar el propio Csound para ver este cálculo con un **print** y la fórmula anterior; en Csound el logaritmo decimal es la función **log10(.)**)

Pregunta **3**

Finalizado

Puntúa 2,0
sobre 2,0

Aplica el programa al siguiente fichero:

0:00 / 0:00

Frecuencia del sonido digitalizado (con 1 decimal): Hz

Amplitud de la onda (con 1 decimal): dBFS

Amplitud de pico

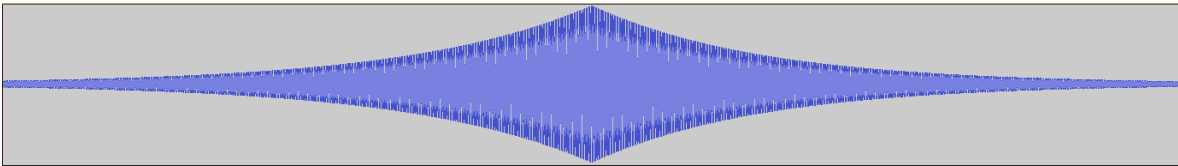
En la práctica de psicoacústica vimos las dificultades de caracterizar la envolvente sobre la propia onda. Un valor muy sencillo de calcular es el valor máximo de amplitud que toma una señal a lo largo de su envolvente. Vamos a hacer un programa que localice este pico en el tiempo y nos dé su valor.

Ejercicio 5.3:

- Descarga la [PLANTILLA](#) de este nuevo proyecto (botón derecho del ratón → guardar como).
- Completa el instrumento **Apico** según las instrucciones en **procedimiento para la medición**.
- Usa el siguiente fichero de audio para desarrollar y probar tu programa:

0:00 / 0:00

(es una onda de 2 segundos con envolvente creciente la primera mitad y decreciente la segunda mitad)



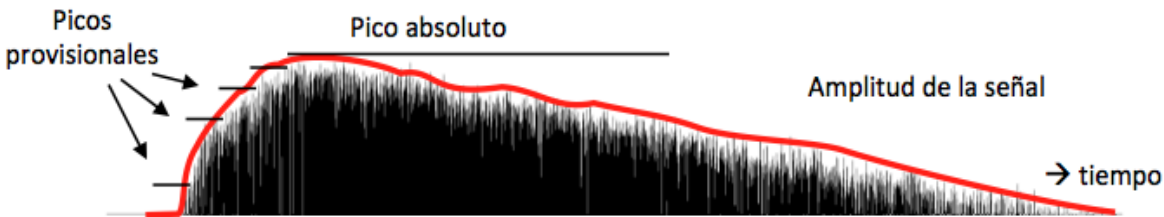
- Cuando creas que el programa está listo, contesta a las preguntas del cuestionario.

Explicación de la medición

- La amplitud de pico se medirá mediante el operador **peak**, que almacena en una variable de salida (tipo **k-**) el valor máximo del valor absoluto de la entrada:

kvariable **peak** asonido ; los nombres son sólo ilustrativos; tú usa los que quieras.

- Recuerda que el sonido irá "pasando" a través de este operador y, cada vez que detecte una amplitud que es un nuevo máximo, actualizará el valor de la variable de salida. Por tanto, una vez se alcanza el pico absoluto, ya no se producen nuevas actualizaciones (ver figura).



Procedimiento para la medición

El instrumento **Apico** debe hacer los siguientes pasos (uno en cada línea de código):

- Cargar el sonido que contiene el fichero en una variable de tipo señal (esto lo hace **soundin**).
- Medir la evolución de la amplitud de pico (con **peak**).
- Verificar si ha cambiado la amplitud de pico respecto al pico hasta ahora (esto se hace con **changed** aplicado a la variable donde almacenes la amplitud de pico). Si el valor analizado no ha cambiado, **changed** emite un cero. Cuando el valor cambia, emite un 1:
 - kCambio **changed** variable_para_la_amplitud_de_pico
- **Si** el valor de la amplitud de pico ha cambiado (cada vez que **changed** haya emitido un 1) **entonces**:
 - Se toma nota del tiempo actual con el operador **times** (sin entradas; lo devuelve en segundos)
 - Se imprime en la consola ese tiempo y el nuevo valor de amplitud de pico (en PCM y dBFS). Esta impresión ya está implementada en la plantilla del ejercicio.
- También se incluye un operador **out** para que oigas el sonido analizado.

La última línea de valores que imprima tu programa es el último valor de pico y, por tanto, lo que debes utilizar para responder a la pregunta del ejercicio.

Pregunta **4**

Finalizado

Puntúa 3,0 sobre 3,0

Descarga el fichero wav

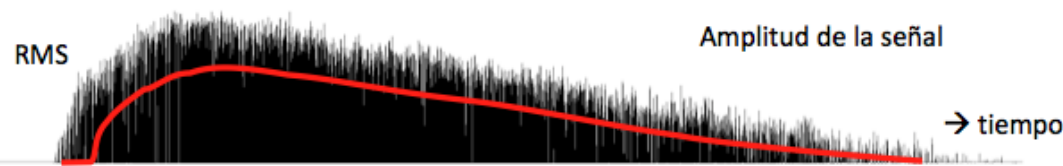
0:00 / 0:00

y aplícale el programa del ejercicio: (usa duración = 10.0 s)

- Amplitud de pico PCM:
- Amplitud de pico en dBFS: (usa coma y 2 decimales)
- Tiempo en el que se ha alcanzado en segundos: (usa coma y 3 decimales)

Envolvente RMS

Ahora vamos a hacer un programa que calcule la envolvente RMS (ver figura) y evaluaremos sus características.



Ejercicio 5.4:

- Descarga la [PLANTILLA](#) del proyecto (botón derecho del ratón → guardar como).
- Escribe el instrumento **A_RMS** y la partitura para controlarlo según las siguientes instrucciones:

Opciones y cabecera

La cabecera ya está preparada.

La frecuencia de control (**kr**, indica cuántas veces se repite el bucle de control) = 10 veces por segundo. De esta manera, el cálculo de la RMS se hará cada décima de segundo (usando, por tanto, 4410 muestras).

Orquesta: instrumento

- Primero se leerá el sonido del fichero wav en una variable de tipo señal (con **soundin**).
- Luego se mide la evolución de la amplitud RMS (con el operador **rms**). Tiene un segundo parámetro para suavizar su salida que debe valer 5.
- Convertir el valor PCM leído a dBFS con la función dbfsamp(.)
- Tomar nota del tiempo actual en segundos con **times**
- Los valores de la envolvente y los tiempos en los que se miden se escribirán en un fichero de texto, de nombre `rms.txt`. La orden que hace esto (**fprintks**) ya viene programada en la plantilla.
- También se incluye un operador **out** para que oigas el sonido analizado.

Partitura

- Ya viene programada. Usa el siguiente fichero de audio para desarrollar y probar el programa (es el mismo de antes; si lo tienes ya descargado no es necesario que lo vuelvas a hacer):

0:00 / 0:00

- Cuando creas que el programa está listo, úsalo para contestar las preguntas del cuestionario. Lee antes las siguientes instrucciones.

Explicación de las mediciones

- Las mediciones necesitan unos criterios para unificarlas. Establecemos una amplitud umbral de -60 dBFS para considerar que, por debajo de él, la envolvente no tiene un valor apreciable.
 - t_A : desde justo antes de que la envolvente supere el umbral hasta que llega a su primer máximo. Si la envolvente comienza ya por encima del umbral se medirá desde $t = 0$.
 - A_S : considera un punto que esté en un valor intermedio de la amplitud durante la fase estable. Si el sonido no tiene fase estable $\rightarrow A_S = 1$
 - t_R : considera como inicio el punto en el que la envolvente empieza a caer hasta que baje del umbral o se termine la señal.
- La resolución para el cálculo de estos tiempos es la duración de una ventana (0,1 s).

Procedimiento para la medición

Para resolver la pregunta siguiente, descargarás un fichero WAV y le aplicarás tu programa. Así obtendrás el fichero `rms.txt` que contendrá la evolución temporal de la envolvente. Abre este fichero con la hoja de cálculo **Excel**. Para observar la forma de la envolvente de amplitud debes representar los valores de la segunda columna (seleccionalos) con un gráfico creado con:

Menú 'insertar' --> 'Linea' --> Líneas



Si en la versión que uses de Excel cambia la estructura de menús, busca la opción de insertar gráfico o consulta con el profesor de prácticas.

Pregunta 5

Finalizado

Puntúa 1,0 sobre 4,0

Descarga el fichero wav

0:00 / 0:00

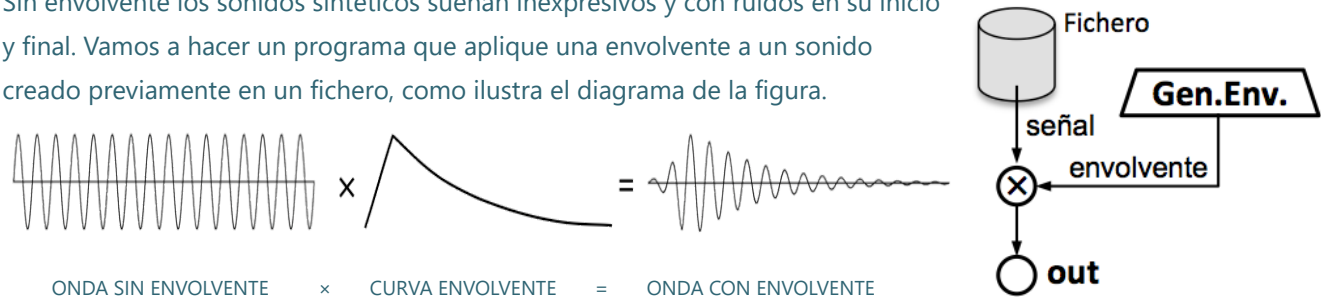
y aplícale el programa del ejercicio: (usa duración = 1.8 s)

- ¿De qué tipo de envolvente se trata? ASR
- Tiempo de ataque t_A (si no hay ataque pon -1): 4 s
- Amplitud de sostenimiento A_S (si no hay fase estable pon 1) : -10 dBFS
- Tiempo de relajación t_R (si no hay relajación pon -1) : 6 s

Información

Aplicar envolvente a fichero

Sin envolvente los sonidos sintéticos suenan inexpresivos y con ruidos en su inicio y final. Vamos a hacer un programa que aplique una envolvente a un sonido creado previamente en un fichero, como ilustra el diagrama de la figura.



Ejercicio 5.5:

- Descarga la [PLANTILLA](#) de este nuevo proyecto (botón derecho → guardar como).
- Programa el instrumento **env_ampli** y la partitura para controlarlo según las instrucciones.
- En el instrumento se han creado las siguientes variables de tipo i- con los parámetros que se pasen desde la partitura:
 - $iDur$: es la duración de la activación (de la nota completa).
 - itA : la duración en segundos del tiempo de ataque.
 - itD : la duración en segundos del tiempo de decaimiento.
 - iAs : la amplitud de sostenimiento (como proporción de la amplitud máxima).
 - itR : la duración en segundos del tiempo de relajación.
- Este fichero contiene la onda a la que aplicarás tu programa (onda seno de 2 segundos sin envolvente, de amplitud máxima (44.100 Hz y 16 bits/muestra):

0:00 / 0:00

El instrumento debe seguir los siguientes pasos:

- Cargar el sonido que contiene el fichero WAV con **soundin**
- Crear la envolvente (en una variable de tipo **k-**) según las instrucciones de la pregunta siguiente, usando el operador adecuado, como se indica en el material previo de la práctica.
- Aplicarla sobre el sonido, multiplicando la envolvente por la señal.
- Enviar el sonido así modificado a la salida con **out**.

En la **partitura**, se debe incluir una única activación del instrumento "env_ampli" que se inicie en tiempo 0, dure 2 segundos y se le pasen, en el orden indicado, los parámetros correspondientes, según la siguiente pregunta. Aquellos parámetros que no se utilicen en tu caso deben ser pasados con valor 0.

Pregunta 6

Finalizado

Puntúa 1,0 sobre 1,0

Aplica una envolvente ASR con:

- Tiempo de ataque = 0.5 s,
- Amplitud de sostenimiento = 0.90, y
- Tiempo de relajación = 0.1 s.

¿Qué operador has usado para implementarla? linen

