

Procesamiento de señal digital 2019-1
Profesor: Felipe A Vallejo, PhD.
Universidad de San Buenaventura, Bogotá
Tarea 2 JUCE

El **objetivo de esta tarea** es generar un aplicativo nativo (Mac) que genere ruido blanco y después lo filtre usando **dos** implementaciones de un filtro tipo Ecualizador Paramétrico y comparar cuál de estas dos implementaciones es mejor. Para la primera, debe implementar el filtro IIR de la sección 5.6.3 del Kuo que usa algoritmo basado en tiempo. Para la segunda, debe implementar el mismo filtro pero ahora usando un algoritmos en frecuencia con el método de convolución rápida, ver Sección 7.4.1 DAFX.

Cada una de estas implementaciones debe hacerse en dos clases independientes y usando programación orientada a objetos como se mostró en clase. Para esto use como plantilla la clase **Filter** que se usa en el [tutorial de Martin Finke](#) y modifíquela para implementar los dos algoritmos en tiempo y frecuencia respectivamente. Primero, cree una nueva clase **ParametricFilter** que implemente el algoritmo para un filtro tipo Ecualizador Paramétrico de la sección 5.6.3 del Kuo y adáptelo a la plantilla de la clase Filter. Después cree una segunda clase **ParametricFilterFFT** y en ella implemente un algoritmo que implemente un algoritmo similar al de la rutina M-file7.7, (VX_Filter.m) en la página 244 del cap 7 del DAFX, pero donde en vez de usar el filtro FIR que se implementó en esa rutina se use el filtro paramétrico de la sección 5.6.3 del Kuo. Para implementar el filtro paramétrico en frecuencia deben computar primero la respuesta en frecuencia o la respuesta al impulso del filtro IIR sección 5.6.3. Este cálculo deberán entregarlo un archivo Word que deben poner en la carpeta principal en la entrega del JUCE. Para implementar la FFT y las ventanas pueden usar [la clase de FFT de JUCE](#).

Adicionalmente, diseñe una interfaz gráfica con tres sliders que controlen el nivel del audio, la frecuencia de corte y la resonancia del filtro. Su aplicativo debe permitir controlar la frecuencia central del filtro y la ganancia de este con dos o tres sliders. Es importante que se pueda lograr efectos de Boost o Cut con estos knobs. También, deben añadir un segundo [combo box](#) a su interfaz para permitir al usuario controlar que el filtro que se va a usar. Por último, añada a su aplicativo el medidor de uso CPU que se implementa en el [tutorial de WaveTable Synthesis](#) para comparar el rendimiento de su aplicativo. La interfaz debe ser estética, original y debe ser intuitiva. Para generar el ruido blanco revise el tutorial [Control audio levels using decibels](#) de JUCE.

Lineamientos para la entrega: Para una entrega exitosa su entrega debe tener los siguientes requerimientos:

- La entrega debe tener la carpeta del proyecto con el código fuente que uso (carpeta Source), el **.jucer** del proyecto, el **.xcodeproj**, e **importantemente** el **.app** generado para el entorno OSX generado por XCode, si su máquina es Windows o Linux, debe usar los computadores de la sala de preproducción para generar el **.app**. **No calificaré ninguna entrega en donde no se someta el .app** pues el **.app** es el producto final de esta tarea.
- El nombre del proyecto debe ser de la forma **Autor1Autor2RuidoFiltrado.jucer**. Para esto es crucial que generen un proyecto nuevo con el aplicativo de JUCE usando la plantilla de *Audio Application* mencionada en el [tutorial para generar proyectos en JUCE](#).
- Archivo Word con cálculo de la respuesta en frecuencia o respuesta al impulso que usó en la implementación de frecuencia del filtro paramétrico.

- El código fuente debe estar comentado de manera correcta como se explicó en clase, sobretodo comentando el código propio y la integración de código externo. En el comentario inicial de cada archivo debe indicar los nombres de los autores, fecha, etc, la funcionalidad del código, y debe referenciar las fuentes de código externo usado en su proyecto.
- Para obtener un 100% en su calificación implemente todos los requerimientos pedidos en la descripción de esta actividad y el DSP debe estar implementado usando programación orientada a objetos.
- La entrega es de a **grupos de máximo 2 personas** y se debe subir al link de la Entrega 2 Juce en aula virtual **antes de la clase** de este **Lunes 1 de Abril**.

Como referencia use: [tutorial de Martin Finke](#), [combo box](#) , [la clase de FFT de JUCE](#), [tutorial de WaveTable Synthesis](#) y [Control audio levels using decibels](#)