

# Proyecto Final de Laboratorio

## Unidad de Efectos para Audio

### Objetivos

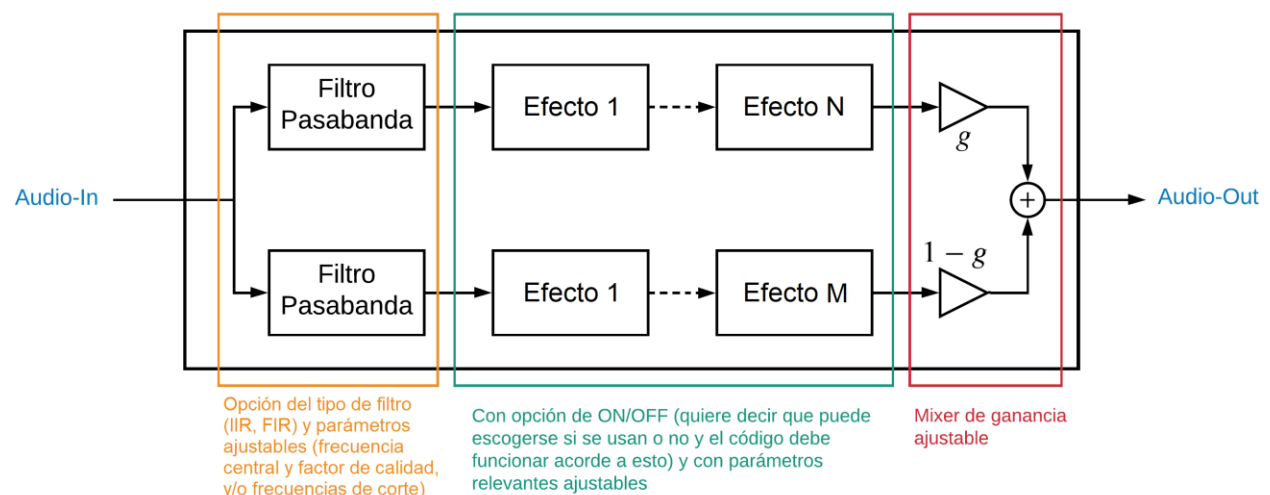
- Aplicar lo aprendido en el curso de Procesamiento de Señales para implementar una unidad de efectos para audio.
- Desarrollar una interfaz gráfica que permita variar los parámetros de la unidad de efectos y generar los resultados deseados.

**Duración:** 3 sesiones.

Para el proyecto final del laboratorio de Procesamiento de Señales, los estudiantes diseñarán una unidad de efectos para audio con interfaz gráfica, en Matlab. **Trabjarán en parejas (si el número de estudiantes es impar, habrá un grupo de tres).**

### Instrucciones

Se tiene como objetivo implementar una unidad de efectos para audio en Matlab, basándose en los algoritmos descritos en los libros *DAFX Digital Audio Effects* y *Designing Audio Effect Plug-Ins in C++ With Digital Audio Signal Processing Theory*. La unidad de efectos debe tener dos canales de procesamiento y debe cumplir con la topología mostrada en la Figura 1.



**Figura 1.** Topología de la unidad de efectos.

La idea es emplear los filtros pasa-banda para poder aislar bandas de frecuencia específicas, usualmente asociadas al contenido espectral de algún instrumento musical particular. A cada banda se le aplicarán uno o varios de los efectos específicos siguientes: **a) distorsión, b) vibrato, c) eco o delay, d) reverberación, y e) phaser o flanger**. Tanto los filtros como los efectos deberán presentar parámetros ajustables por medio de una interfaz gráfica.

La interfaz gráfica deberá permitir al usuario realizar lo siguiente:

- Cargar un archivo de audio, el cual puede ser de uno o dos canales.
- Reproducir el audio original.
- Seleccionar el tipo de filtro (IIR, FIR) y modificar sus parámetros (frecuencia central y factor de calidad, o frecuencias de corte), por banda.
- Seleccionar los efectos deseados, por banda.
- Ajustar la ganancia del *mixer* ( $0 \leq g \leq 1$ ).
- Aplicar los efectos.
- Reproducir el audio procesado más reciente. Si se cambian los filtros y/o efectos, se debe poder volver a procesar el audio original, y reproducir el nuevo audio procesado.
- Generar un archivo de audio, con un nombre especificado por el usuario. El archivo corresponderá al audio procesado más reciente.
- Mostrar gráficas de respuesta en frecuencia de los filtros.
- Mostrar gráficas de los espectros de las señales de entrada (audio original) y salida (audio procesado). Considere el uso de **plot** en lugar de **stem**, para una mejor visualización.
- Exportar filtros y otras variables de interés a un archivo **.mat**, con un nombre especificado por el usuario.

Puede encontrar información sobre el desarrollo de interfaces gráficas en Matlab en el siguiente enlace: <https://la.mathworks.com/discovery/matlab-gui.html>

También pueden buscar otros tutoriales y ejemplos en línea para familiarizarse con la creación de interfaces gráficas.

## **Detalles de Implementación**

Pueden usar herramientas y funciones que hayan usado o visto en laboratorios y teoría, y pueden reciclar código que hayan desarrollado anteriormente. Para el diseño de los filtros, pueden usar cualquier método visto en clase, y pueden usar herramientas y funciones de Matlab, incluyendo el *Filter Designer*. Sin embargo, **deberán implementar (aplicar) los efectos y los filtros IIR por medio de sus ecuaciones de diferencias**. Para aplicar los filtros FIR, sí pueden usar funciones nativas de Matlab como **filter**.

Deberán asegurarse de que sus señales de audio procesadas estén en el rango de amplitud estándar (consideren ganancias unitarias y normalizaciones).

## Validación

Para validar su unidad de efectos, deberán buscar o generar archivos de audio adecuados para usarlos como entradas. Pueden usar canciones, audios de vídeos, etc. La unidad deberá ser capaz de aislar dos bandas de frecuencia, basándose en una tabla de contenido espectral como la que se muestra en la Figura 2. Luego, deberá ser capaz de aplicar distintos efectos a cada banda, con los parámetros especificados en la interfaz. Por ejemplo, se puede pedir aplicar distorsión y vibrato a una banda, y *flanger* a otra.

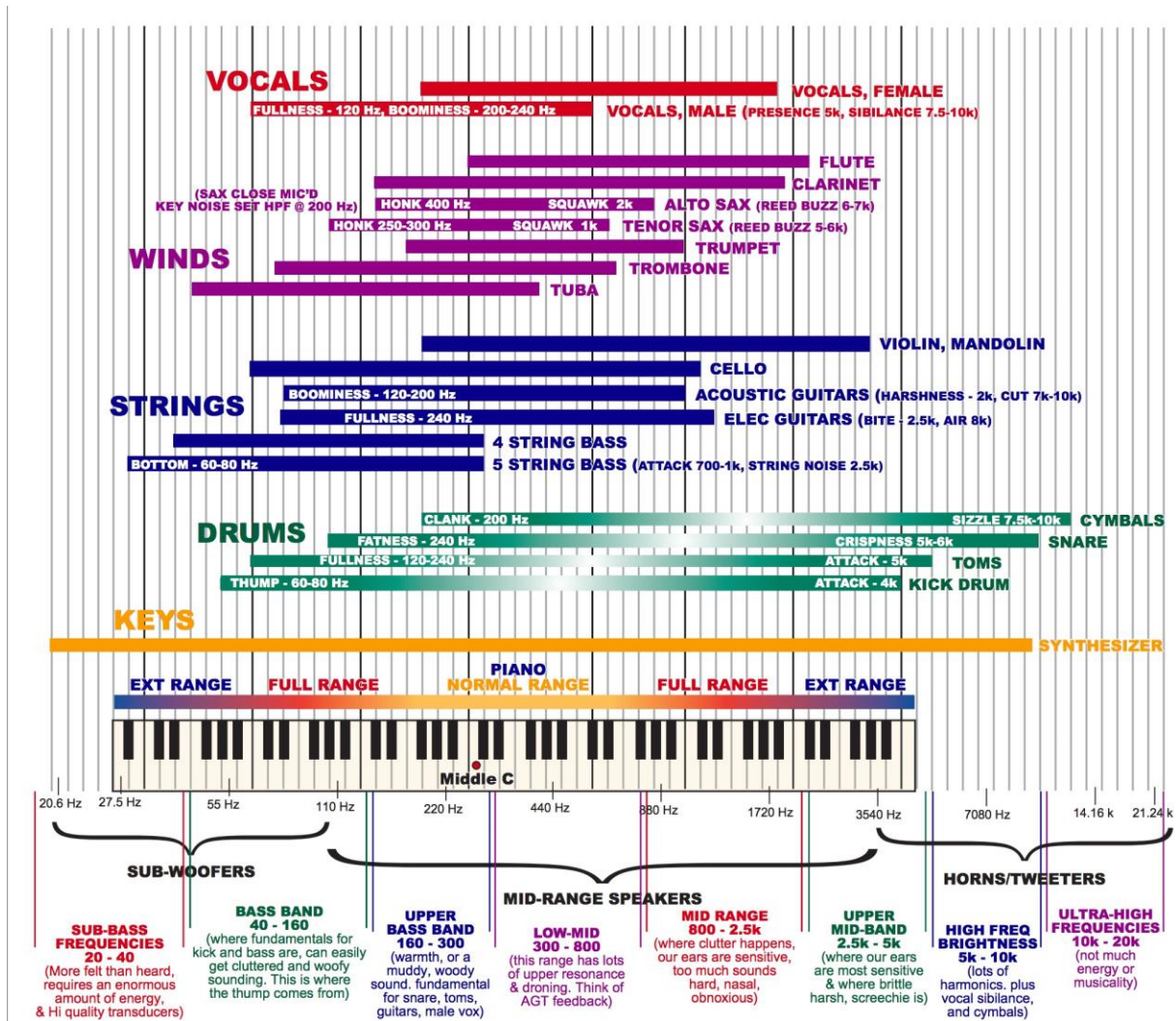


Figura 2. Contenido espectral de diversos instrumentos musicales.

Busquen audios que sean relativamente fáciles de separar (por ejemplo, un audio con pocos instrumentos, o con un instrumento y una persona cantando). **Se recomienda hacer pruebas con señales artificiales creadas con bandas de frecuencia específicas, para evaluar cada parte de su unidad de efectos.**

## **Evaluación**

### **Interfaz gráfica y archivo *ReadMe*: 30%**

Se verificará que la interfaz ofrezca todas las opciones indicadas anteriormente. Deberán crear un archivo *ReadMe*, que explique brevemente cómo se usa la interfaz. Pueden agregar un botón u opción en la interfaz para desplegar las instrucciones de uso.

### **Correcto funcionamiento: 30%**

Se verificará que el audio de salida contenga lo solicitado, según los parámetros y efectos seleccionados. Esto incluye el correcto aislamiento de las bandas y la aplicación de los efectos con el *mixer*.

### **Vídeo con demostración: 30%**

Cada grupo deberá generar un vídeo demostrando la funcionalidad de su unidad de efectos. El vídeo debe durar un máximo de 10 minutos, y el archivo no debe ser mayor a 100 MB. Use formatos estándar (.mp4, .wmv).

### **Código: 10%**

Deberán entregar su(s) archivo(s) de Matlab, debidamente comentados e identificados. Deben incluir todo lo necesario para correr su unidad de efectos.

**Importante:** cualquier evidencia de plagio detectada en el código implicará una nota de **cero** para todos los involucrados.

### **Asistencia a las sesiones:**

Todos deben conectarse a las sesiones de Zoom. En las sesiones se darán indicaciones y sugerencias para el proyecto, y se resolverán dudas. **Por cada ausencia no justificada se deducirán puntos del proyecto (de forma individual).**

**Entrega:** tendrán hasta las **23:59 hrs. del sábado 29 de mayo** para subir un **archivo .zip o .rar** a Canvas. El archivo comprimido deberá incluir sus archivos de Matlab, el *ReadMe*, y su vídeo.

Nombren a su archivo así: IE3032\_Proyecto\_Apellido1\_Apellido2.zip (o .rar). Los apellidos deben estar en orden alfabético. Si el grupo es de tres personas, incluyan los tres apellidos, por supuesto. Ejemplo: IE3032\_Proyecto\_Rivera\_Zea.zip.

**Sólo un miembro del grupo deberá subir el archivo comprimido a Canvas.**

Si tienen problemas al subir su archivo, debido al tamaño del vídeo, pueden incluir un enlace a Google Drive para el vídeo en los comentarios de la tarea de Canvas. En este caso, el .zip/.rar únicamente contendrá los archivos de Matlab y el *ReadMe*. Esto debe ser el último recurso.

**Se bajarán puntos si ambos estudiantes suben archivos, si el vídeo dura más de los 10 minutos y/o si excede los 100 MB indicados.**