
MANUAL DE USUARIO DE PROJUCER

Héctor Jair Morales Gómez ^{*}
Facultad de Ciencias, UNAM
Dr. Eduardo Nahmad Achar ^{**}
Instituto de Ciencias Nucleares, UNAM
Projucer — Versión 6

26 de agosto de 2020

Resumen

Este documento describe las capacidades del software JUCE, un marco de aplicación disponible en múltiples plataformas y de código semi-abierto. El software está escrito en C++ y se recomienda familiaridad con dicho lenguaje antes de comenzar a trabajar con JUCE. En este documento se nos guía a través de la instalación del software y se exploran algunos ejemplos que nos permiten verificar que se pueda ejecutar correctamente en nuestro sistema. Como apéndice, se explora el uso de JUCE para la creación de aplicaciones con interfaz gráfica y se discute qué uso se le dio durante el curso *Introducción a la Acústica Contemporánea* impartido en la Facultad de Ciencias de la Universidad Nacional Autónoma de México.

Este manual fue elaborado como parte de las actividades del servicio social del alumno **Héctor Jair Morales Gómez** bajo la supervisión del **Dr. Eduardo Nahmad Achar**, investigador del ICN, UNAM. El servicio social fue registrado bajo el título de **Optimización de uso de Software de Manipulación Algebraica**. El objetivo fue estudiar el uso de la librería JUCE para evaluar su implementación en cursos del área de acústica impartidos en la Facultad de Ciencias de la UNAM. Este manual hace referencia a contenido disponible en el repositorio de [GitHub](#) del autor. Para cualquier duda o aclaración referente a este manual, por favor referirse a esta dirección de [correo electrónico](#). Agradecimientos especiales al Profesor Andrés Valentín Porta Zepeda, académico de la Facultad de Ciencias y a los alumnos que tomaron el curso de Introducción a la Acústica Contemporánea de la carrera de Física de la UNAM.

Nota aclaratoria: los ejemplos con los que se trabaja en este manual se basan fuertemente en el libro *Designing Audio Effect Plug-Ins in C++: With Digital Audio Signal Processing Theory* del autor *Will Pirkle*. Los programas que se discuten en la sección de efectos de audio se beneficiaron del trabajo de Juan Gil disponible en su [página personal](#).

^{*} hectorjair@ciencias.unam.mx

^{**} nahmad@nucleares.unam.mx

Índice

| | |
|---|-----------|
| 1. Introducción | 3 |
| 1.1. ¿Qué es Projucer? | 3 |
| 1.2. Requerimientos del sistema | 3 |
| 1.3. Registro en my.roli | 3 |
| 1.4. Elección de licencia y descarga | 4 |
| 1.5. Verificar la descarga | 4 |
| 1.6. Accediendo a la interfaz de demostraciones | 5 |
| 1.7. Creando un nuevo proyecto | 5 |
| 1.8. Configuración de la variable PATH | 7 |
| 2. Ventana Principal | 7 |
| 2.1. Secciones de la Ventana Principal | 7 |
| 2.2. Controles de la Ventana Principal | 8 |
| 2.3. Controles del panel lateral | 8 |
| 3. Configuraciones de proyecto | 9 |
| 4. Editor de código | 9 |
| 4.1. Cambiar el color de nuestro editor | 9 |
| 5. Explorador de archivos | 10 |
| 5.1. Panel lateral del explorador de archivos | 10 |
| 5.2. Configuración del explorador de archivos | 10 |
| 5.3. Añadiendo nuevos archivos | 10 |
| 6. Configuraciones de Módulos | 11 |
| 7. Configuraciones de Exportación | 11 |
| 7.1. Panel lateral de exportadores | 11 |
| 8. Añadiendo nuevos exportadores | 12 |
| 9. Conclusión y aspectos prácticos del uso de Projucer | 12 |
| A. Localización del repositorio | 13 |
| B. Descarga del repositorio | 13 |
| C. Interfaz de usuario de los programas de efecto de audio | 14 |

1. Introducción

1.1. ¿Qué es Projucer?

Projucer es una aplicación que nos permitirá desarrollar y montar aplicaciones multiplataforma en conjunto con nuestro ambiente de desarrollo nativo (IDE). Viene precargado con la librería JUCE, que es un marco de aplicación de código abierto escrito en C++.

Es una herramienta sumamente útil de control de configuraciones para nuestro proyecto JUCE que nos permite, entre otras cosas, añadir y quitar archivos fuente y administrar las librerías de nuestro proyecto de forma eficiente.

1.2. Requerimientos del sistema

Projucer está disponible para los tres sistemas operativos más populares y tiene los siguientes requisitos:

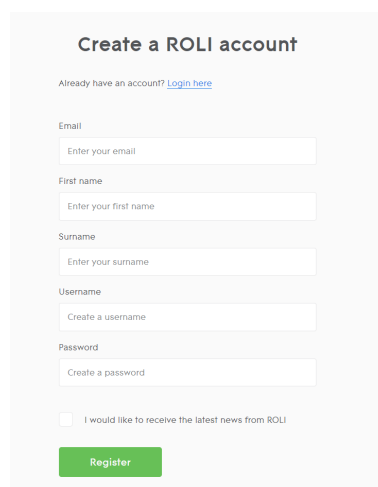
| Sistemas Operativos | | |
|----------------------------|--|----------------------------------|
| MAC | Windows | Linux |
| OS 10.9 Xcode 7.3 NA | OS 8.1 Visual Studio 2015 64 Bit | Ubuntu 16.04.3 Makefile NA |

Es necesario instalar el ambiente de desarrollo apropiado para su sistema antes de usar la totalidad de funciones de JUCE con Projucer. Es posible que en algunas plataformas, haya que instalar dependencias extra; por ejemplo, Android SDK para el desarrollo para plataformas Android. Cualquier caso no previsto, referirse a la documentación disponible en <https://juce.com/discover/stories/projucer-manual>.

1.3. Registro en my.roli

Antes de utilizar Projucer, será necesario crear una cuenta my.roli. La cuenta se puede crear en la siguiente página: <https://auth.roli.com/register>. Será necesario proporcionar nuestra fecha de nacimiento, país y aceptar los términos y condiciones.

Posteriormente, es necesario llenar los campos con la información que se solicita, tal como aparece en la siguiente imagen.



The image shows a web form titled "Create a ROLI account". At the top, there is a link: "Already have an account? [Login here](#)". Below this, there are five input fields, each with a label and a placeholder text: "Email" (placeholder: "Enter your email"), "First name" (placeholder: "Enter your first name"), "Surname" (placeholder: "Enter your surname"), "Username" (placeholder: "Create a username"), and "Password" (placeholder: "Create a password"). Below the password field, there is a checkbox with the text "I would like to receive the latest news from ROLI". At the bottom of the form is a green button labeled "Register".

Después de haber creado nuestra cuenta y de haber verificado la dirección de correo electrónico que proporcionamos, podemos ingresar al sitio de la comunidad Projucer. El acceso al sitio se hace a través de la pantalla que se muestra en la figura 2.

1.4. Elección de licencia y descarga

Ya que tenemos una cuenta my.roli, podemos proceder a la descarga. En la página <https://juce.com/> debemos ir a la pestaña GET JUCE. Se nos mostrarán las opciones para elegir nuestro plan de pago y sistema operativo, tal como se observa en la figura 3. Para las aplicaciones estudiantiles, el plan Personal es suficiente.

| | Personal Free | Indie \$35 per month | Pro \$65 per month | Education Free |
|--------------------------|------------------|-------------------------|-----------------------|-------------------|
| Splashscreen | Made with JUCE | None | None | Made with JUCE |
| Analytics collection | Mandatory | None | None | Mandatory |
| Revenue or funding limit | \$50k | \$200k | No limit | No limit |
| Minimum commitment | None | 12 months | 12 months | None |
| One-off perpetual price | None | \$700 paid once | \$1,300 paid once | None |
| | Download | Purchase Plan | Purchase Plan | Download |

Upgrade to JUCE 5

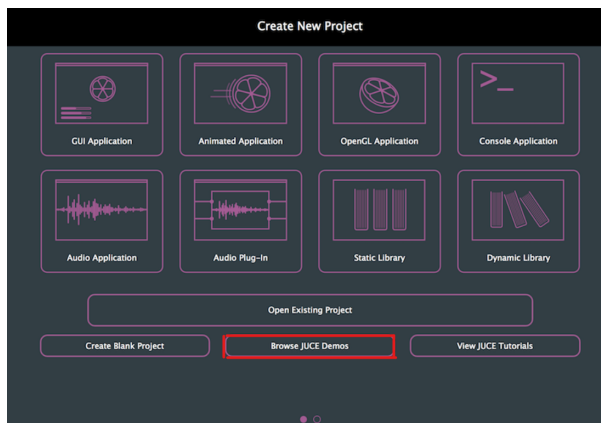
Al presionar el botón de **Download**, se nos da a elegir para qué sistema operativo queremos la descarga. Se recomienda que se descomprima el archivo descargado y se coloquen los documentos que contiene en la carpeta HOME de nuestra computadora. Una vez que esté colocado el archivo, hay que acceder a la carpeta JUCE; ahí podremos encontrar la aplicación Projucer. Para acceder a ella sólo hay que ejecutarla.

1.5. Verificar la descarga

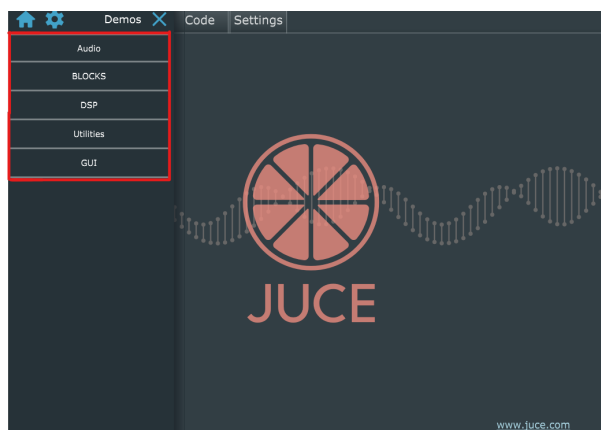
Una forma simple de verificar que la descarga se realizó correctamente es acceder a la aplicación DemoRunner que se encuentra en la carpeta JUCE y correr el primero de los proyectos muestra. Si la aplicación se ejecuta correctamente, tenemos la certeza que no hay incompatibilidades entre el programa y nuestro sistema.

1.6. Accediendo a la interfaz de demostraciones

Ya que verificamos la descarga, podemos visualizar las demostraciones que vienen integradas con Projucer; esto nos permitirá darnos una idea de las capacidades de la aplicación. Para ver las demostraciones, hay que ejecutar la aplicación y acceder a la sección de "Browse JUCE demos". La posición del botón se muestra en la siguiente imagen:



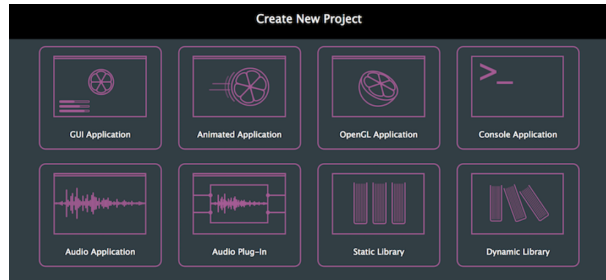
Se abrirá una nueva ventana donde estará la interfaz para correr las demostraciones. Estos ejemplos están dividido en las siguientes cinco categorías.



Si se presiona la pestaña con la leyenda **Code** podemos ver el código fuente de estas demostraciones. No tenemos la opción de compilarlo, pero este código es bastante útil para el desarrollo de nuestras propias aplicaciones. En la pestaña **Settings**, podemos configurar parámetros como la entrada y salida de audio y los canales que queremos utilizar.

1.7. Creando un nuevo proyecto

Para la creación de un proyecto desde cero, tenemos que lanzar la ventana de auxilio de Projucer. Esta ventana es la primera que aparece al ejecutar la aplicación. Para crear un nuevo proyecto, hay que seleccionar el tipo de proyecto que queremos. Tenemos las opciones que se muestran en la imagen:



Para cada tipo de proyecto, Projucer generará automáticamente todos los archivos necesarios y añadirá la cantidad mínima de código para su configuración. Así, después de crear nuestro nuevo proyecto, podemos empezar inmediatamente a desarrollar nuestra aplicación.

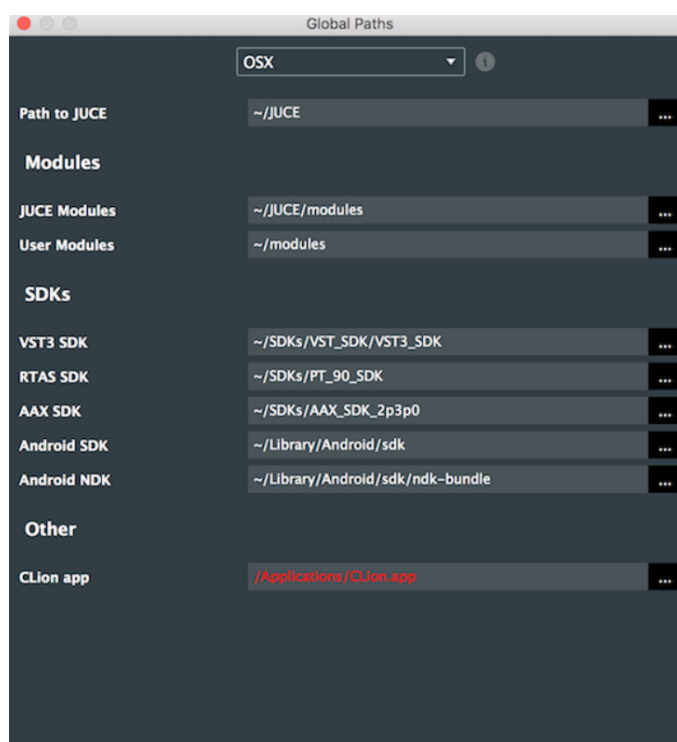
Un buen inicio para empezar a desarrollar aplicaciones es la opción "GUI application". Esta opción nos permitirá desarrollar una aplicación con interfaz gráfica, ya sea para una plataforma de escritorio o una aplicación móvil. La siguiente tabla ofrece un resumen de los diferentes proyectos que podemos crear con Projucer.

| Tipo de proyecto | Descripción |
|-----------------------|---|
| Aplicación GUI | Se crea una aplicación mínima JUCE con una ventana vacía. A partir de aquí se pueden agregar funciones utilizando las diferentes clases del lenguaje. |
| Aplicación animada | Esto crea una aplicación que dibuja una interfaz gráfica animada. Es un buen inicio si buscamos crear una aplicación móvil, por ejemplo. |
| Aplicación OpenGL | Esta opción crea una aplicación JUCE en blanco, justo como una aplicación GUI, pero añade soportes OpenGL que son especialmente útiles para importar modelos 3D y shaders GLSL. |
| Aplicación de consola | JUCE es muy útil para desarrollar aplicaciones ejecutables en consola sin necesidad de una interfaz de usuario. Esta opción es la ideal para ese tipo de proyecto. |
| Aplicación de audio | Esta opción genera un aplicación JUCE en blanco, pero automáticamente añade el código necesario para configurar el audio de entrada y de salida. Es muy útil para juegos y aplicaciones multimedia. |
| Librería estática | Este tipo de proyecto es útil para crear librerías de software reusable. Esta modalidad de proyecto es la mejor para proyectos de vinculación. |
| Librería dinámica | Este tipo de proyecto es útil para crear librerías de software reusable. Esta modalidad de proyecto es la mejor para proyectos de vinculación dinámica. |

Para abrir un proyecto existente, sólo es necesario presionar el botón `.open Existing Project`; automáticamente se abrirá el tipo de aplicación que coincida con el tipo de proyecto que tenemos. Un buen punto de partida para empezar nuestro proyecto es utilizar uno de los proyectos de ejemplo. Estos proyectos pueden abrirse al presionar el botón para abrir un proyecto existente y buscándolos en la carpeta de ejemplos, dentro de la carpeta JUCER.

1.8. Configuración de la variable PATH

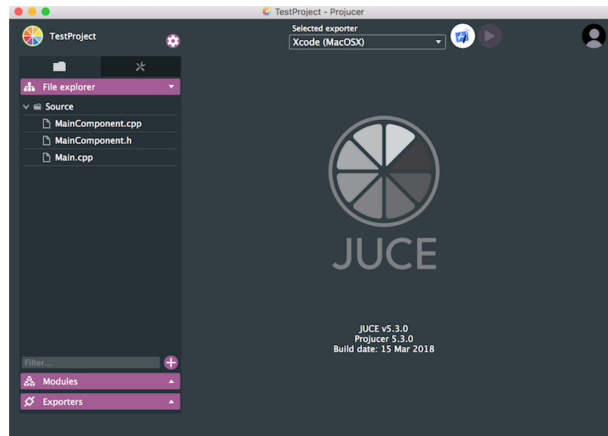
En lugar de especificar una ruta particular en la computadora para cada uno de los proyectos JUCE, es posible establecer rutas relativas para guardar nuestras librerías. De este modo se puede abrir el mismo proyecto Projucer en otra computadora y automáticamente buscará las rutas establecidas en el sistema original. Se puede acceder a esta ventana de configuración en las opciones del menú, en la pestaña Global Paths; tal como se muestra en la siguiente figura.



2. Ventana Principal

2.1. Secciones de la Ventana Principal

La ventana principal de la sección de proyectos de Projucer está dividida en tres secciones diferentes: la barra de herramientas en la parte de arriba; la cual cuenta con cuatro diferentes secciones que se detallarán más adelante, el panel lateral a la izquierda y el editor en la derecha, tal como se muestra en la siguiente figura.



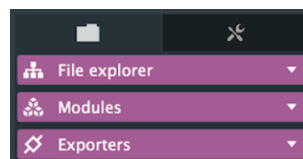
2.2. Controles de la Ventana Principal

La barra de herramientas en la parte de arriba de la ventana provee varias funcionalidades. La siguiente tabla ofrece un resumen y una breve descripción de cada una de ellas.

| Acción | Descripción |
|------------------------------|--|
| Configuraciones del proyecto | Esta función, señalada por el ícono de una tuerca, abre el panel de configuraciones. |
| Guardar y abrir en IDE | Guarda un proyecto y lo abre en el IDE instalado en nuestro sistema. El ícono es el ícono del IDE que tenemos instalado por default. |
| Abrir aplicación | Esta opción marcada por un ícono de Reproducir corre la aplicación y la abre en Projucer. Es necesario compilar primero. |
| Configuraciones de usuario | En esta opción podemos cambiar las configuraciones de Projucer. Es especialmente útil si estamos compartiendo nuestro sistema con algún otro usuario |

2.3. Controles del panel lateral

El panel lateral está compuesto por las tres pestañas que se muestran en la siguiente imagen. Esta sección es especialmente útil en el caso de que nuestra aplicación esté configurada para utilizar una IDE determinada; en ese caso, en la sección de 'exporters' será necesario cambiar la IDE y compilarla nuevamente. Esto será de especial utilidad cuando trabajemos con las aplicaciones de efectos de audio que se proponen en la sección 6.



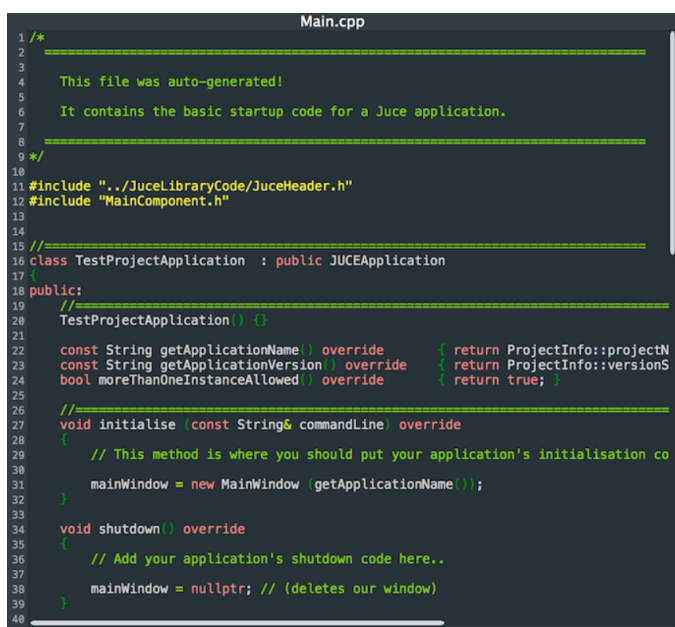
3. Configuraciones de proyecto

La sección de configuraciones de Proyecto nos va a permitir administrar los metadatos de nuestra aplicación. En esta sección podemos configurar cosas como el título del proyecto, su versión y añadir los datos de los autores o la compañía que desarrolla la aplicación.

También se pueden configurar cosas como los colores y tamaños predeterminados de las ventanas y el tamaño máximo que queremos que tenga la aplicación. Cuando el proyecto esté configurado como un plug-in de otro software, será necesario especificarlo en esta sección de acuerdo a los parámetros definidos en la [documentación](#).

4. Editor de código

Esta es la ventana principal siempre que estamos trabajando con Projucer. La aplicación nos ofrece una ventana para editar código en C++. Para que la sintaxis aparezca coloreada resaltando las funciones, variables y clases de nuestro código, es necesario guardar el archivo con el que estamos trabajando con extensión `.cpp`. La siguiente imagen muestra un ejemplo de un archivo `.cpp` abierto en el editor de código.



```
1 /*
2
3
4 This file was auto-generated!
5
6 It contains the basic startup code for a JUCE application.
7
8
9 */
10
11 #include "../JuceLibraryCode/JuceHeader.h"
12 #include "MainComponent.h"
13
14 //
15
16 class TestProjectApplication : public JUCEApplication
17 {
18 public:
19     TestProjectApplication() {}
20
21     const String getApplicationName() override { return ProjectInfo::projectName; }
22     const String getApplicationVersion() override { return ProjectInfo::versionString; }
23     bool moreThanOneInstanceAllowed() override { return true; }
24
25     //
26     void initialise(const String& commandLine) override
27     {
28         // This method is where you should put your application's initialisation code
29         mainWindow = new MainWindow(getApplicationName());
30     }
31
32     void shutdown() override
33     {
34         // Add your application's shutdown code here..
35         mainWindow = nullptr; // (deletes our window)
36     }
37
38 }
39
40
```

El código de nuestras aplicaciones JUCE lo podemos editar directamente en esta ventana o editarlo en la IDE de nuestra preferencia.

4.1. Cambiar el color de nuestro editor

Para finalizar esta breve sección, mencionaremos una característica interesante que nos ofrece Projucer. Si preferimos trabajar con otros colores para el editor de código podemos cambiar la paleta de colores en *View > EditorColourScheme > Create...*

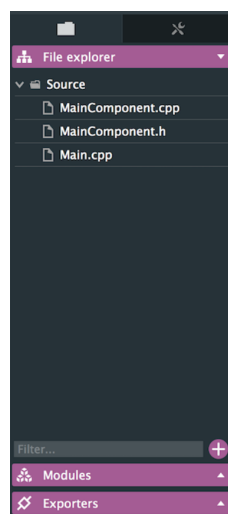
Los colores están en código hexagésimal y se puede encontrar una buena base de datos en esta [página](#). Para aplicar los cambios sólo será necesario presionar el botón de 'save scheme'

5. Explorador de archivos

Nuestros proyectos estarán compuestos la mayoría de veces por varios archivos. La mayoría de estos archivos del tipo `.cpp`. Dependiendo de qué tan complejo o extenso sea nuestro proyecto, tendremos más o menos archivos. Para administrar nuestros archivos, en la parte lateral de la ventana de Projucer tenemos al explorador. Dando click sobre cada una de las pestañas, podemos abrirlas en editor de código. También desde aquí podemos añadir o eliminar archivos de nuestro proyecto.

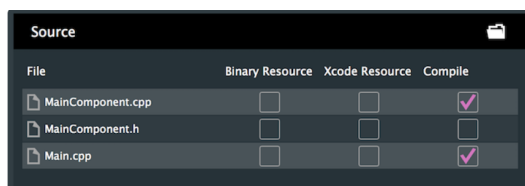
5.1. Panel lateral del explorador de archivos

El panel lateral tiene una vista jerárquica de los archivos de nuestro proyecto. En esta sección se pueden crear, editar o eliminar archivos y los cambios se aplicarán a todo el directorio del proyecto.



5.2. Configuración del explorador de archivos

Si se selecciona un folder del panel lateral, se puede acceder a la sección de configuración desde donde se pueden configurar las opciones de configuración y los binarios.



5.3. Añadiendo nuevos archivos

Si le damos click al símbolo + se abre una pestaña donde se presentan las opciones para añadir archivos a la estructura del proyecto. Cuando se tiene que añadir un archivo, es necesario que se haga desde aquí y no en la IDE nativa, pues los cambios se pueden sobrecribir y esto puede resultar en errores al momento de compilar el proyecto. Esta pestaña también tiene la opción de ordenar los archivos que componen al proyecto según el criterio que seleccionamos. Ordenar alfabéticamente puede resultar muy útiles cuando se trabaja con un proyecto con muchos archivos.

6. Configuraciones de Módulos

El código de JUCE se organiza en diferentes módulos. Sin necesidad de configurar nada. Todos los módulos necesarios para correr el tipo de proyecto que se haya escogido, ya están en el código. Cada librería de JUCE está organizada en diferentes módulos con múltiples funcionalidades. Se puede escoger cierto conjunto de módulos dependiendo de las necesidades de nuestro proyecto en la sección de módulos.

La mayoría de los módulos que contiene JUCE dependen de otros módulos dentro de la misma plataforma para trabajar correctamente. Si se remueve un módulo de la sección, Projucer nos marcará los módulos que están 'rotos' poniéndolos en rojo. Si este error ocurre, se recomienda eliminar también los módulos rotos antes de intentar compilar el código.

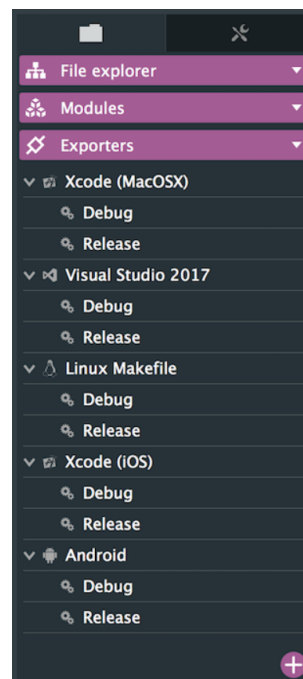
Una lista completa de los módulos disponibles en la versión 6 de projucer puede encontrarse en la [documentación](#) referente a esta sección.

7. Configuraciones de Exportación

Los diferentes exportadores (Xcode, Visual Studio, etc) pueden ser encontrados en la pestaña 'Exporters' en el panel lateral izquierdo. En esta plataforma es posible especificar los compiladores adicionales y las opciones compilación de las diferentes plataformas en las que queramos compilar nuestro proyecto.

7.1. Panel lateral de exportadores

El panel lateral con las opciones de exportación muestra todas las diferentes configuraciones para los exportadores seleccionados para el proyecto. La pestaña tiene la siguiente estructura:



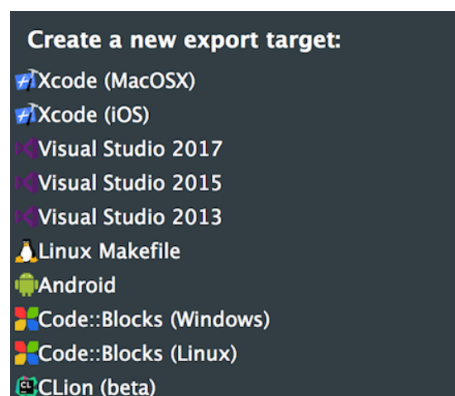
Los proyectos con los que se trabaja el apéndice de este manual, están preconfigurados para tener como exportadores a los compiladores más comunes de los sistemas operativos más populares.

En esta sección no es necesario ir más allá de ver cuáles son los compiladores que tenemos por default en nuestro proyecto. Es posible que queramos compilar con en un sistema que no esté en esta pestaña; en ese caso será necesario agregar el compilador.

8. Añadiendo nuevos exportadores

Finalmente, antes de empezar a trabajar con los ejemplo de efecto de audio es importante discutir un posible error que puede surgir cuando compilemos nuestro proyecto: es posible que el compilador de nuestro sistema no se encuentre configurado automática en Projucer. La solución en ese caso es muy sencilla: agregaremos nuestro exportador en la ventana de exportadores.

Al presionar el símbolo de + en el panel de exportadores, se pueden añadir exportadores nuevos, justo como se muestra en la imagen:



Añadir un nuevo exportador se puede hacer en cualquier fase del proyecto. Por ejemplo, si originalmente se creó una aplicación JUCE para Windows y macOS, pero se tomó la decisión de añadir soporte para Linux, se puede hacer sin problemas.

También se pueden borrar exportadores presionando click derecho sobre el exportador que queremos eliminar y presionando la opción 'delete'.

9. Conclusión y aspectos prácticos del uso de Projucer

Hasta ahora se han discutido superficialmente las funciones principales de Projucer. Con lo que se ha hablado hasta ahora, el usuario está listo para empezar a trabajar en sus propias aplicaciones.

Como puede ser bastante intimidante empezar un proyecto desde cero, se recomienda al lector de este manual hacer modificaciones pequeñas sobre los ejemplos propuestos en el apéndice de este manual (cuyo código fuente puede ser encontrado en el repositorio al que se hizo mención en la introducción) y compilarlas en nuestro equipo.

Después de hacer eso, se estará más familiarizado con el proceso de ejecución de las aplicaciones. Además, los ejemplos que se discuten en el apéndice son de especial importancia para el lector interesado en el tema de efectos de audio; vistos desde la parte acústica y física. Lo discutido en este manual ha sido probado correctamente en Windows 10 y en ubuntu 18.04 y 20.04. Cualquier sistema operativo anterior a estas versiones, podría presentar problemas no previstos en este manual.

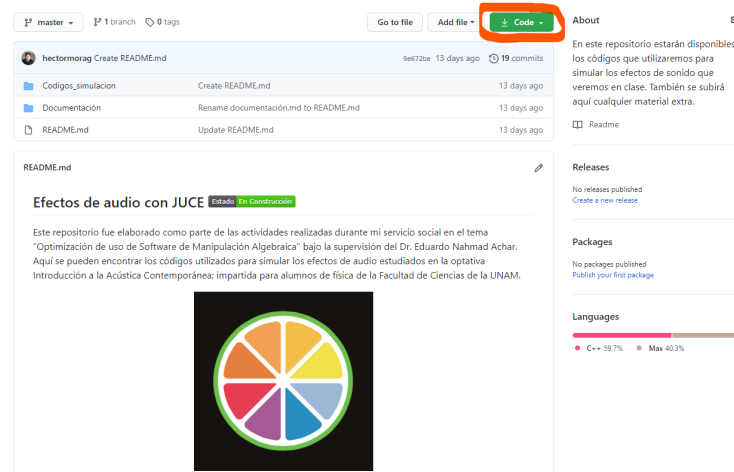
Apéndice: Simulación de efectos de audio

A. Localización del repositorio

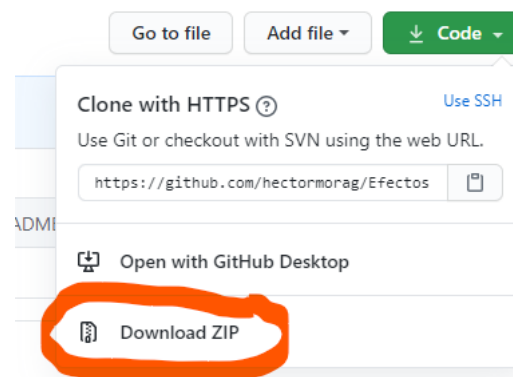
En el [repositorio](#) se encuentran los códigos que discutiremos a continuación. Estos códigos abarcan la mayor parte del temario de la materia *Introducción a la Acústica Contemporánea* impartida en la Facultad de Ciencias de la UNAM. Los códigos están basados en los temas presentados en el libro *Audio Effects: Theory, Implementation and Application* de Joshua D. Reiss y Andrew P. McPherson, disponible en [el siguiente liga](#).

B. Descarga del repositorio

Antes de compilar los códigos, es necesario descargarlos. Esto se puede hacer en un sólo click presionando el botón de descarga que se muestra a continuación,



De las opciones que se presentan, seleccionamos 'Download Zip' o si se tiene familiaridad con el uso de git, podemos descargar el repositorio con el comando `git clone [link de descarga]`.

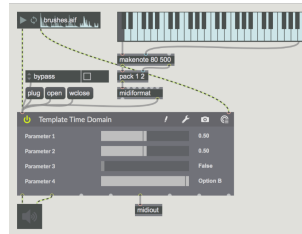


Como destino de nuestra descarga, seleccionamos una carpeta que hayamos creado para nuestros proyectos y descomprimos el archivo zip. Al hacer esto, ya podremos abrir los ejecutables con Projucer. Si se le quiere modificar algo a alguno de los programas, es necesario abrir el código fuente con el editor de código Projucer y compilarlo nuevamente.

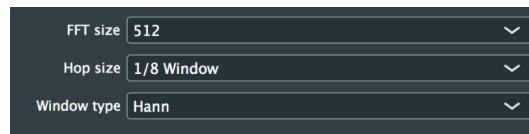
C. Interfaz de usuario de los programas de efecto de audio

Los códigos para la simulación de efectos de audio disponibles en el repositorio son los siguientes:

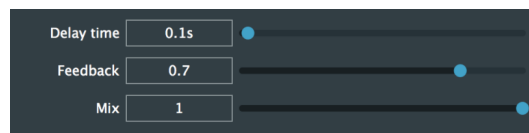
- **Template Time Domain:** este código implementa una GUI genérica con deslizadores lineales y logarítmicos, alternadores y cuadros combinados. Este proyecto introduce una clase particular de parámetros de audio que encierra mucha de la complejidad disponible para añadir, configurar y automatizar plug-ins de audio. Esta aplicación no le aplica ningún procesamiento especial a nuestro audio, pero constituye un modelo muy útil para aplicar efectos de procesamiento de dominio temporal.



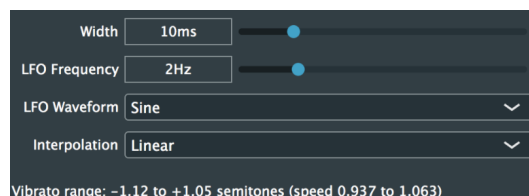
- **Template Frequency Domain:** este código implementa una transformada de fourier de tiempo reducido. Este plug-in no aplica ningún tipo de procesamiento a la señal de entrada, sólo convierte el bloque de entrada al dominio de frecuencia utilizando el método de superposición.



- **Delay:** este código implementa un delay con interpolación lineal simple para lograr tiempos fraccionarios de delay.



- **Vibrato:** este código implementa un oscilador de baja frecuencia (LFO) que modula el delay del input de entrada y simula variaciones periódicas en el tono.



- **Flanger:** este código simula un efecto de audio basado en el código de delay, en el cual una copia de la señal de entrada es atrasada con un tiempo de delay variable.

Flanger effect control interface showing the following parameters:

- Delay: 2.5ms
- Width: 10ms
- Depth: 1
- Feedback: 0
- Inverted mode: ☐
- LFO Frequency: 0.2Hz
- LFO Waveform: Sine
- Interpolation: Cubic
- Stereo: ☒

- **Coro:** simula el efecto que se produce cuando varios músicos tocan la misma pieza al mismo tiempo (i.e. se crean varias copias de la señal de entrada con variaciones minúsculas de tono y tiempo).

Coro effect control interface showing the following parameters:

- Delay: 30ms
- Width: 20ms
- Depth: 1
- Number of voices: 2
- LFO Frequency: 0.2Hz
- LFO Waveform: Sine
- Interpolation: Linear
- Stereo: ☒

- **Ping-Pong Delay:** este código genera una versión estéreo del código de delay. Aquí, la señal retrasada rebota entre el canal izquierdo y el canal derecho.

Ping-Pong Delay effect control interface showing the following parameters:

- Balance input: 0.25
- Delay time: 0.1s
- Feedback: 0.7
- Mix: 1

- **Wah-Wah:** este código simula el efecto de audio que añade un sonido similar al habla humana cuando se pronuncia la onomatopeya "Wa-wa". Se puede usar en modo manual, donde la frecuencia de corte es modificada con el deslizador.

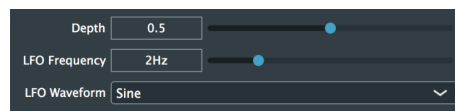
Wah-Wah effect control interface showing the following parameters:

- Mode: Automatic
- Mix: 0.5
- Frequency: 657.9962Hz
- Q Factor: 10
- Gain: 20dB
- Filter type: Resonant Low-pass
- LFO Frequency: 2Hz
- LFO/Env: 0.8
- Env. Attack: 2ms
- Env. Release: 300ms

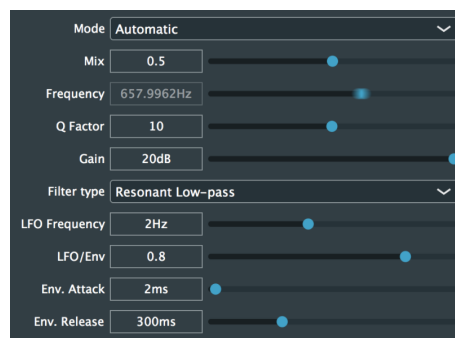
- **Phaser:** este código usa filtros en configuración de cascada para introducir cambios de fase a una señal de entrada. El fasor produce un efecto similar al flanger, pero tiene más control en la localización de los cortes.



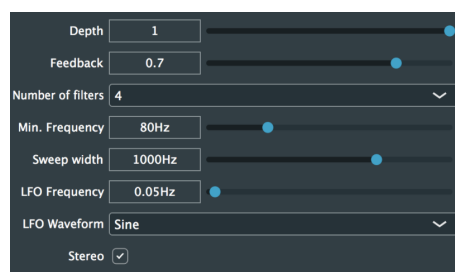
- **Trémolo:** este código usa un LFO para modular la amplitud de la señal de entrada. Esto simula variaciones pequeñas en el nivel de la señal ocasionando una serie rápida de repeticiones.



- **Distortion:** este código aplica una transformación no lineal a la señal de entrada, que incrementa su ganancia. Diferentes transformaciones puede ser seleccionadas y la señal de salida puede ser controlada de forma individual para restaurar los niveles originales de ruido.



- **Panning:** este código cambia la posición aparente de las fuentes de sonido entre dos canales.



Índice alfabético

Coro, 15

Delay, 14

Distortion, 16

Flanger, 15

Github, 4

GUI, 6

JUCE, 3

Linux, 4

Panning, 16

Phaser, 16

Ping-Pong Delay, 15

Projucer, 3

rules, 9

Template Frequency Domain, 14

Template Time Domain, 14

Trémolo, 16

Ventana Principal, 7, 9

Vibrato, 14

Wah-Wah, 15

Windows, 5

Bibliografía

Lecturas adicionales

- [1] Will C. Pirkle. Designing audio effect plugins in c++, 2013.
- [2] Boulanger R. The audio programming book, 2002.
- [3] Farnell A. Designing sound, 2001.