

## **GENERADOR DE SONIDOS BINAURALES POR MEDIO DE LABVIEW Y PURE DATA**

### **GENERATOR OF BINAURAL SOUNDS THROUGH LABVIEW AND PURE DATA**

**Andrés Felipe Cuevas Quiroga**

**Resumen:** Este artículo presenta el diseño e implementación de un generador de sonidos binaurales por medio de la plataforma gráfica LABVIEW y el procesador de audio PURE DATA de modo que por medio de una interfaz, se pueda manipular estos determinados sonidos, es decir, se pueda variar sus características según lo desee el usuario, generando los diferentes tipos de ondas cerebrales gracias a la diferencia de frecuencias que se produce entre el audio izquierdo y el derecho. El generador no solo produce tonos binaurales puros o también denominados pulsos binaurales, sino que también se aplica en la realidad virtual, es decir, se desarrolló en la interfaz una opción que le permite al usuario interactuar con dichos sonidos por medio del movimiento de su cabeza, creando una sensación de realidad por medio de audio.

**Palabras clave:** binaural, generador, labview, tonos, interfaz, ondas cerebrales, realidad virtual.

**Abstract:** This paper presents the design and implementation of a binaural sound generator by means of the LABVIEW graphic platform and the PURE DATA audio

processor so that through an interface, these certain sounds can be manipulated, that is, their features as desired by the user, generating different types of brain waves thanks to the difference in frequencies that occurs between the left and right audio. The generator not only produces pure binaural tones or also called binaural pulses, but it is also applied in virtual reality, that is, an interface that allows the user to interact with said sounds through the movement of his head was developed in the interface. , creating a sense of reality through audio.

**Key Words:** binaural, generator, LabVIEW, tones, interface, brain waves, virtual reality.

## 1. Introducción

Los sonidos binaurales son una ilusión del procesamiento auditivo, que ocurre al escuchar con auriculares una señal que se emite con una frecuencia acústica distinta en cada oído. Por ejemplo si en el auricular derecho se emite una frecuencia de 300 Hz y en el izquierdo una de 310 Hz, el ritmo binaural será de 10 Hz, y nuestro cerebro percibirá una pulsación de baja frecuencia resultante de la composición de los dos sonidos. [1]

En la actualidad se han estudiado en gran manera los efectos de los sonidos binaurales en el área de la medicina comportamental y psicopatología, ya que sus características permiten afectar diferentes estados de la mente humana. [1][2]

En cuanto a la tecnología, se aplican los sonidos binaurales en los campos de la realidad virtual y el sonido 3D, dado que las características de dichos sonidos causan diferentes tipos de sensaciones en el cerebro, entre ellas la sensación del espacio en 3D.[3][4][5]

Observando las diferentes aplicaciones y utilidades que se le pueden dar a los sonidos binaurales, se desarrolló por medio del software de programación gráfico LABVIEW y la plataforma de procesamiento de audio PUREDATA, un generador de este tipo de sonidos,

produciendo tanto pulsos binaurales, como sonido de realidad virtual; permitiendo al usuario una interacción e inmersión en estos determinados sonidos. Para generar los pulsos binaurales, el usuario podrá por medio de la interfaz variar las frecuencias en un rango determinado; En cuanto a la aplicación en la realidad virtual, es importante mencionar que el generador permite cumplir con los tres principios esenciales de esta, los cuales son: inmersión, presencia e interactividad. Ya que el generador tiene una fuente de sonido fija con la cual el usuario podrá interactuar por medio del movimiento de su cabeza.

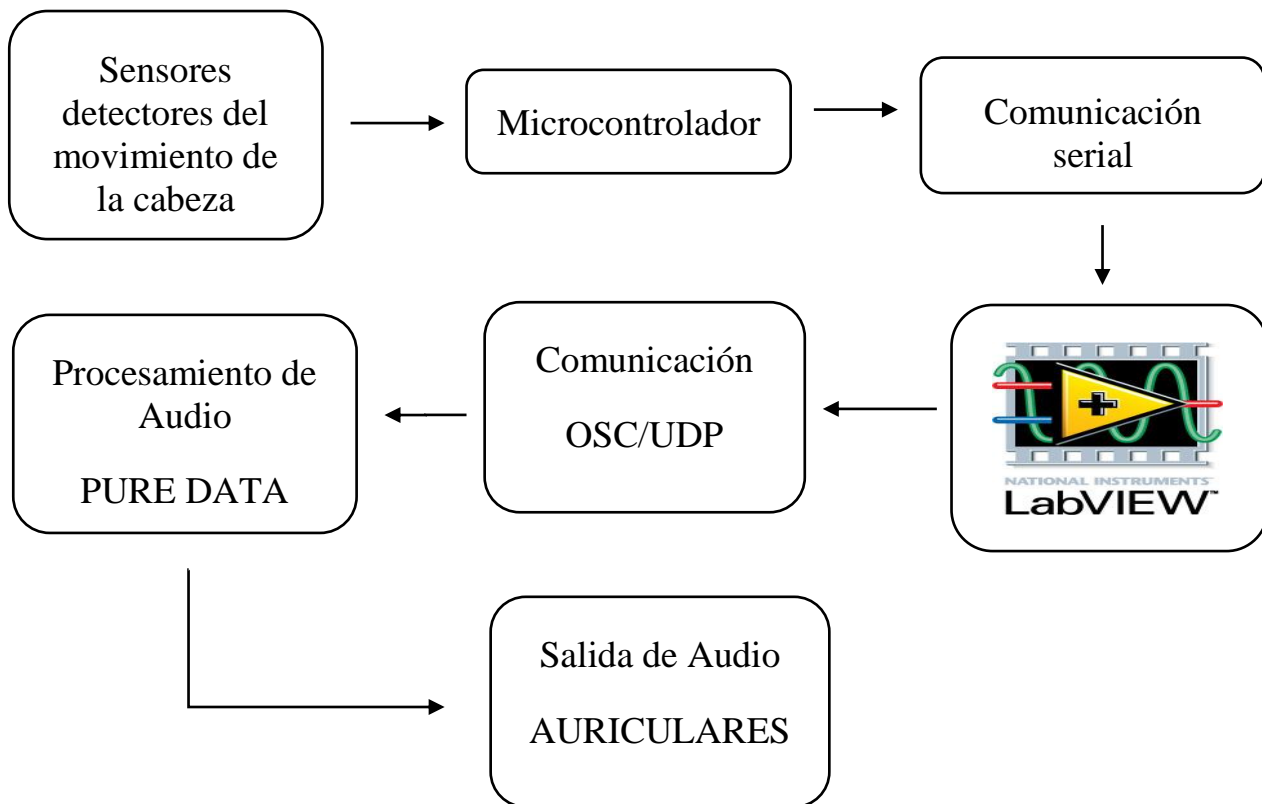
## **2. Desarrollo del proyecto**

### **2.1 Modelamiento**

Para el desarrollo del generador binaural se realizó el diagrama de bloques observado en la figura 1 donde se evidencia cada una de las partes del proyecto y los subsistemas que lo componen. También se puede observar en este por medio de las flechas, el flujo de datos y el modo de operación del proyecto. De forma que cada uno de estos bloques cumple una función específica para el debido funcionamiento del generador.

### **2.2 Desarrollo experimental**

A continuación se explicara el funcionamiento de cada uno de estos subsistemas. Y los materiales utilizados o requeridos dentro de ellos, También se encontraran los códigos y algoritmos elaborados en los diferentes lenguajes de programación usados para el desarrollo del proyecto.

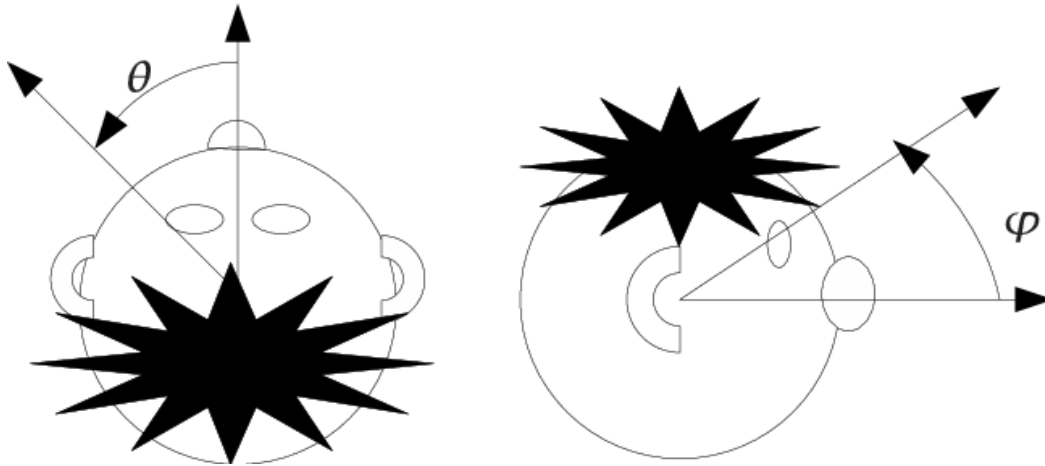


**Figura 1.** Diagrama de bloques del generador binaural

Fuente: Elaboración propia

### 2.2.1 Sensores detectores del movimiento de la cabeza:

Para detectar el posicionamiento y los movimientos de la cabeza fue necesario el uso del sensor inercial IMU MPU6050 y la brújula digital HMC5883L, estos nos dan continuamente el dato del ángulo azimut, es decir la dirección en la cual está mirando el usuario; y el dato de la elevación e inclinación, ver figura 2.



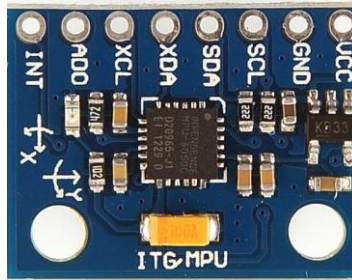
**Figura 2.** Angulo azimut ( $\Theta$ ) y ángulo de elevación ( $\phi$ ). [6]

### **Sensor MPU6050:**

EL sensor MPU6050 es una pequeña pieza tecnológica de procesamiento de movimiento. El cual mediante la combinación de un MEMS (Sistemas Micro electromecánicos) giroscopio de 3 ejes y un MEMS acelerómetro de 3 ejes con un DMP™ (Movimiento Digital Processor™), es capaz de procesar los algoritmos de movimientos complejos de 9 ejes (MotionFusion™) en una placa. [7]

El MPU-6050 es un 6 DOF (grados de libertad = Degrees of Freedom) o un sensor IMU de seis ejes, lo que significa que da seis valores de salida.

El sensor MPU-6050 es muy preciso, ya que contiene una conversión hardware de 16 bits de A/D por cada canal, para la digitalización de las salidas del acelerómetro. Para ello capta los canales x, y y z al mismo tiempo. El sensor utiliza el I2C-bus para interconectar con el microcontrolador. [7]



**Figura 3.** Sensor mpu6050 [7]

En la figura 7 se observan los pines de la mpu6050, los cuales no fueron usados en su totalidad ya que solo fueron necesarios los descritos en la tabla 1.

PINES	FUNCION
VCC	Alimentación del sensor(5v)
GND	GND
SCL	BUS_I2C
SDA	BUS_I2C
ADD0	GND- 0X68

**Tabla1.** Pines mpu6050 usados en el generador binaural

Fuente: Elaboración propia

***Brújula digital HMC5883L:***

EL HMC5883L es un magnetómetro de 3 ejes, con este podemos leer las componentes del campo magnético presente, de esta forma conociendo la dirección del campo magnético terrestre podemos calcular la orientación con respecto al norte magnético de la tierra, esto siempre y cuando nuestro sensor no este expuesto a algún campo magnético externo u algún objeto metálico que altere el campo magnético terrestre. [8]



**Figura 4.** Brújula digital HMC5883L [9]

Los pines usados del HMC5883L fueron los del bus i2c y los de alimentación.

Toda información captada por los sensores es enviada por el bus I2C al microcontrolador para la adquisición de los datos. Tanto el mpu6050 como la brújula digital estarán incorporados en los auriculares para que cumpla su funcionalidad.

### 2.2.2 Microcontrolador:

La función del microcontrolador en este proyecto fue adquirir los datos obtenidos por los sensores, por medio del bus I2C, y enviarlos por medio de comunicación serial a labView.



**Figura 5.** FRDM-KL25Z [10]

Para esta funcionalidad se usó la tarjeta FRDM-KL25Z ya que tiene las cualidades básicas para el desarrollo del proyecto.

Entre las características de esta tarjeta podemos encontrar las siguientes: núcleo ARM Cortex-M0 + de 32 bits que funciona a 48MHz. Incluye 128KB FLASH, 16KB RAM, USB Host, dispositivo USB, SPI, I2C, ADC, DAC, PWM, sensor táctil y otras interfaces de E / S.

El FRDM-KL25Z es totalmente compatible con la plataforma mbed, por lo que tiene acceso a las herramientas gratuitas y al SDK que proporciona a los desarrolladores integrados experimentados herramientas potentes y productivas para crear programas. [10]

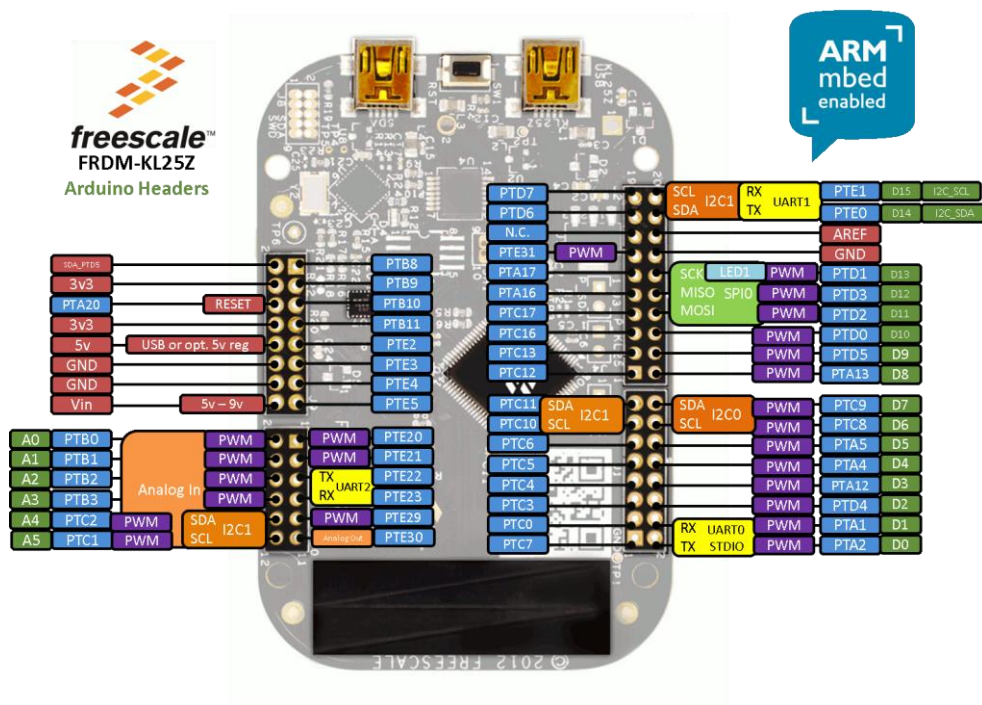


Figura 5. pines de la FRDM-KL25Z [10]

Los pines usados en el generador binaural de la FRDM-KL25Z se muestran en la tabla 2 con su respectiva función.



pin	funcionalidad
<b>Ptc9</b>	SDA conectado al sensor HMC5883L
<b>Ptc8</b>	SCL conectado al sensor HMC5883L
<b>Ptc11</b>	SDA conectado al sensor MPU6050
<b>Ptc10</b>	SCL conectado al sensor MPU6050
<b>3v3</b>	VCC
<b>GND</b>	GND
<b>USB</b>	Conexión serial con el PC

Tabla 2. pines de la FRDM-KL25Z usados en el generador binaural y su funcionalidad

Fuente: Elaboración propia

Para la lectura de los sensores por el bus I2C y el envío de datos por el puerto serial, se elaboró el siguiente código de programación.

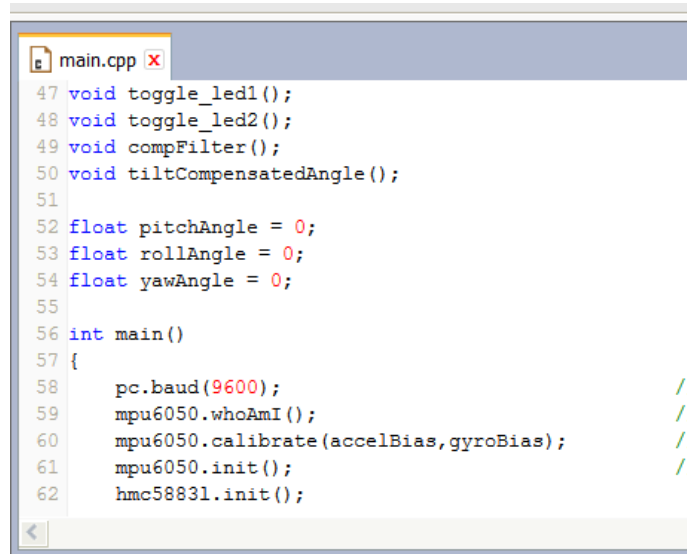
```

main.cpp
35 #include "mbed.h"
36 #include "HMC5883L.h"
37 #include "MPU6050.h"
38 #include "ledControl.h"
39
40 Serial pc(USBTX, USBRX);
41 MPU6050 mpu6050;
42 HMC5883L hmc5883l;
43 Ticker toggler1;
44 Ticker filter;
45 Ticker compass;
46
47 void toggle_led1();
48 void toggle_led2();
49 void compFilter();
50 void tiltCompensatedAngle();

```

Figura 6. programación en la plataforma mbed; librerías y variables

En la figura 6 se observa la primer parte del código en la plataforma mbed donde se muestran las librerías utilizadas en el proyecto, entre las cuales se encuentran la MPU6050 y la HMC5883L que son de vital importancia ya que son las encargadas de obtener los valores de los sensores. También se observa la creación de algunos métodos utilizados más adelante en el código ver figura 8.



```
main.cpp x
47 void toggle_led1();
48 void toggle_led2();
49 void compFilter();
50 void tiltCompensatedAngle();
51
52 float pitchAngle = 0;
53 float rollAngle = 0;
54 float yawAngle = 0;
55
56 int main()
57 {
58     pc.baud(9600); //
59     mpu6050.whoAmI(); //
60     mpu6050.calibrate(accelBias, gyroBias); //
61     mpu6050.init(); //
62     hmc5883l.init();
```

**Figura 7.** programación en la plataforma mbed, creación de variables y método main

Fuente: Elaboración propia en plataforma mbed

En la figura 7 se observamos el método main en el cual se inicializan los sensores y se ajusta el baud rate de la comunicación serial para este caso 9600.

Para adquirir el valor de los ángulos se realiza el procedimiento observado en la figura 8 y posteriormente es enviado por el puerto serial en el método while como se observa en la figura 9.

```
yawAngle = hmc58831.getHeading();

/* Integrate the gyro data(deg/s) over time to get angle */
*pitch += gx * dt; // Angle around the X-axis
*roll -= gy * dt; // Angle around the Y-axis

/* Turning around the X-axis results in a vector on the Y-axis
whereas turning around the Y-axis results in a vector on the X
pitchAcc = atan2f(accelData[1], accelData[2])*180/PI;
rollAcc = atan2f(accelData[0], accelData[2])*180/PI;

/* Apply Complementary Filter */
*pitch = *pitch * 0.98 + pitchAcc * 0.02;
*roll = *roll * 0.98 + rollAcc * 0.02;
```

**Figura 8.**programacion en la plataforma mbed, obtención de los ángulos

Fuente: Librería mpu6050 y Hmc5883l de mbed

```
5 while(1)
6 {
7     pc.printf("%.1f,%.1f,%.1f\r\n",rollAngle,pitchAngle,yawAngle); //
8     wait_ms(40);
9 }
10 }
```

**Figura 9.**programacion en la plataforma mbed, método while principal

Fuente: Elaboración propia en plataforma mbed

### 2.2.3 Interfaz en Labview:

La interfaz en el programa Labview permite interactuar con los sonidos binaurales de modo que el usuario pueda variar sus características según lo desee. Comunica los datos al software pure data para el procesamiento de audio y recibe continuamente los datos obtenidos por el microcontrolador de los sensores.

El proyecto tiene dos funcionalidades en específico, la primera de ellas es generar tonos binaurales puros los cuales funcionan como se ve en la figura 10. De modo que la primer funcionalidad de la interfaz en LabVIEW, es poder variar las frecuencias de cada hemisferio, teniendo en cuenta que los sonidos binaurales no pueden tener una diferencia entre el lado derecho y el izquierdo mayor a 30 hz. Para esto se diseñó la parte de la interfaz visualizada en la figura 11.

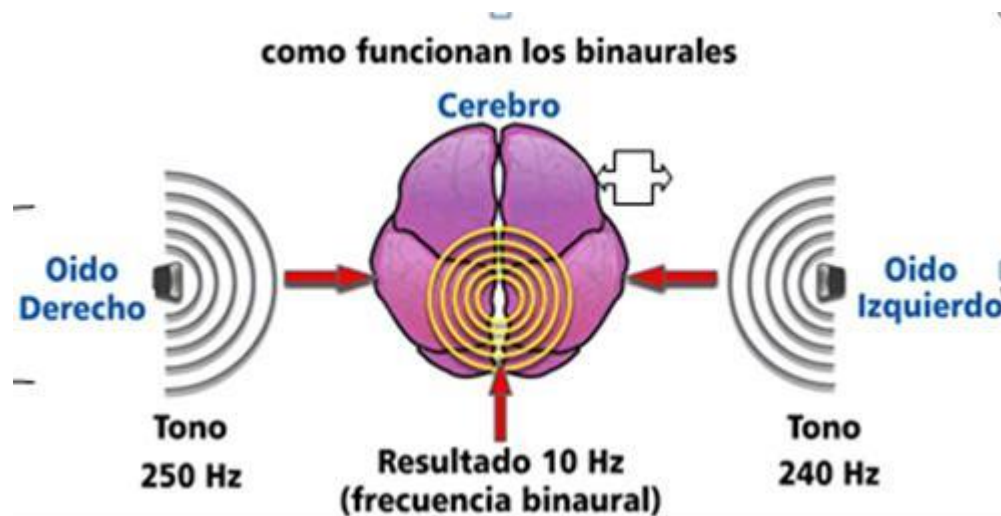


Figura 10. Funcionamiento de los pulsos binaurales [2]



Figura 11. parte de la interfaz en Labview dedicada a los pulsos binaurales

Fuente: Elaboración propia

En la figura 11 se pueden observar varios elementos de la interfaz cuya funcionalidad es poder generar los pulsos binaurales puros entre ellos se visualizan dos controles los cuales son los encargados de variar las frecuencias y las variaciones entre ellas.



**Figura 12.**control de frecuencias en Labview

Fuente: Elaboración propia en Labview



**Figura 13.**control de la variación de frecuencias entre el lado izquierdo y derecho en Labview

Fuente: Elaboración propia en Labview

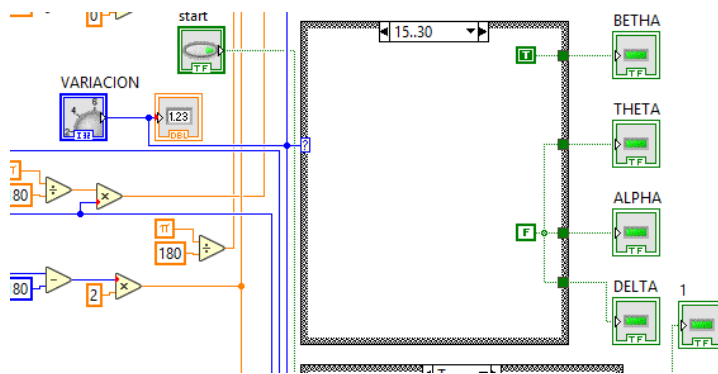
También se puede observar en la figura 11 en la parte izquierda unos leds indicadores los cuales muestran que tipo de ondas se están produciendo.

Los rangos de las frecuencias se visualizan en la tabla 3.

Tipo de ondas	frecuencias
delta	0-4 hz
theta	4-7hz
alpha	7-14hz
beta	14-30hz

**Tabla 3.**tipos de ondas y frecuencias

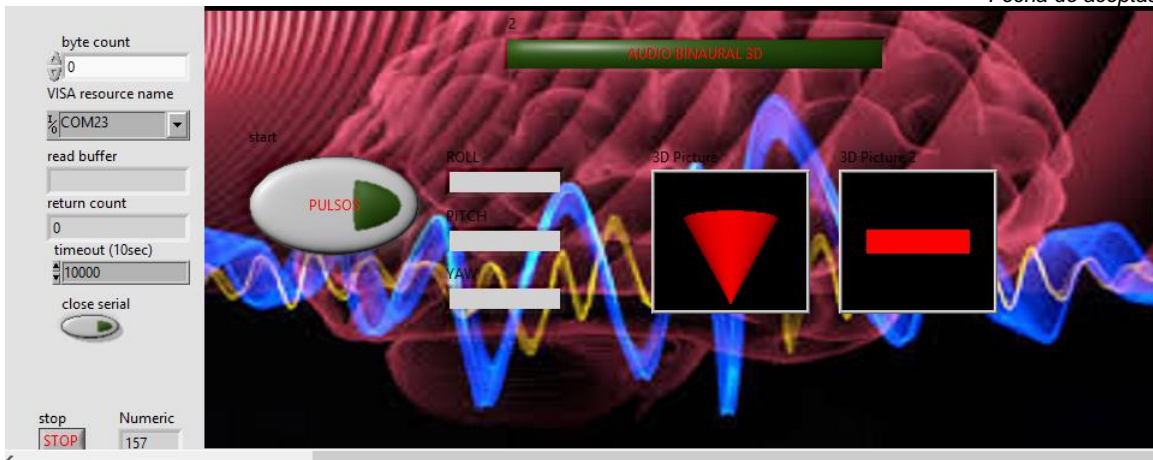
En la figura 14 se puede ver parte del código en la plataforma labview para hacer las variaciones y observarlas en los indicadores



**Figura 14.**codigo en Labview de los pulsos binaurales

Fuente: Elaboración propia en labview

La segunda parte de la interfaz se enfoca en la generación de audio binaural por medio del movimiento de la cabeza, de modo que se obtenga una realidad virtual. Para este fin se realizo la siguiente parte de la interfaz visualizada en la figura 15



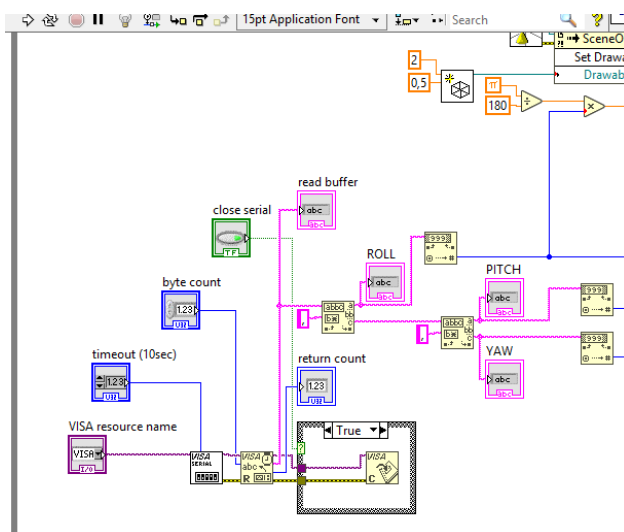
**Figura 15.** parte de la interfaz cuya función es el audio binaural 3d

Fuente: Elaboración propia en labview

En esta parte del generador mostrado en la figura 15 se puede evidenciar el modo como va a interactuar el usuario con la interfaz, ya que el cono de rojo visualizado muestra la dirección en la que el oyente está, es decir el ángulo azimut o también conocido como yaw; Y el cubo de la parte derecha nos muestra en tiempo real la inclinación o elevación de la cabeza, a estos ángulos se le denominan pitch y roll.

Estos ángulos son obtenidos del microcontrolador por medio de la comunicación serial y visualizados en los indicadores de texto como se muestra en la figura 16, en la cual se observa el código usado para adquirir los datos por medio de los bloques visa serial.

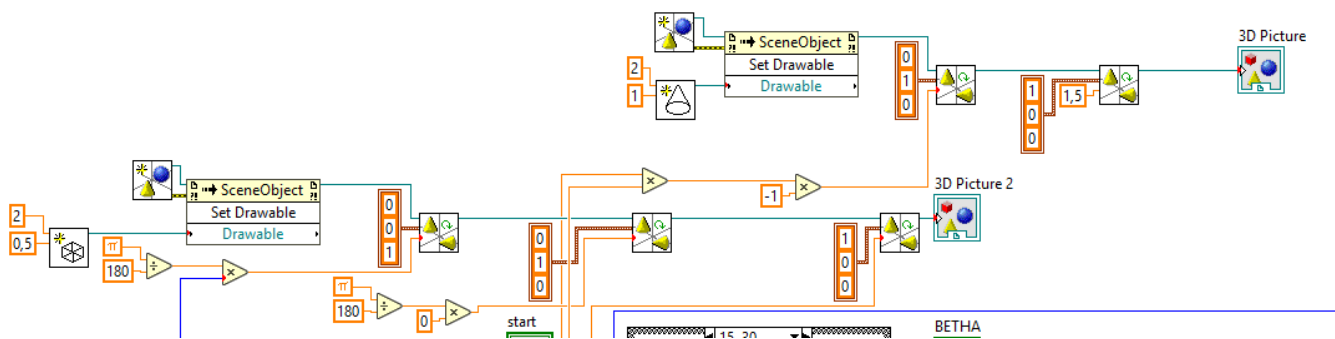
Para establecer la conexión por medio del puerto serial es necesario establecer el puerto COM de enlace con la tarjeta FRDM-KL25Z.



**Figura 16.**Codigo en Labview para la conexión serial

Fuente: Elaboración propia en labview

Para poder graficar y mover el cono y el cubo que nos indican la posición de la cabeza en tiempo real. Se desarrolló el código evidenciado en la figura 17.

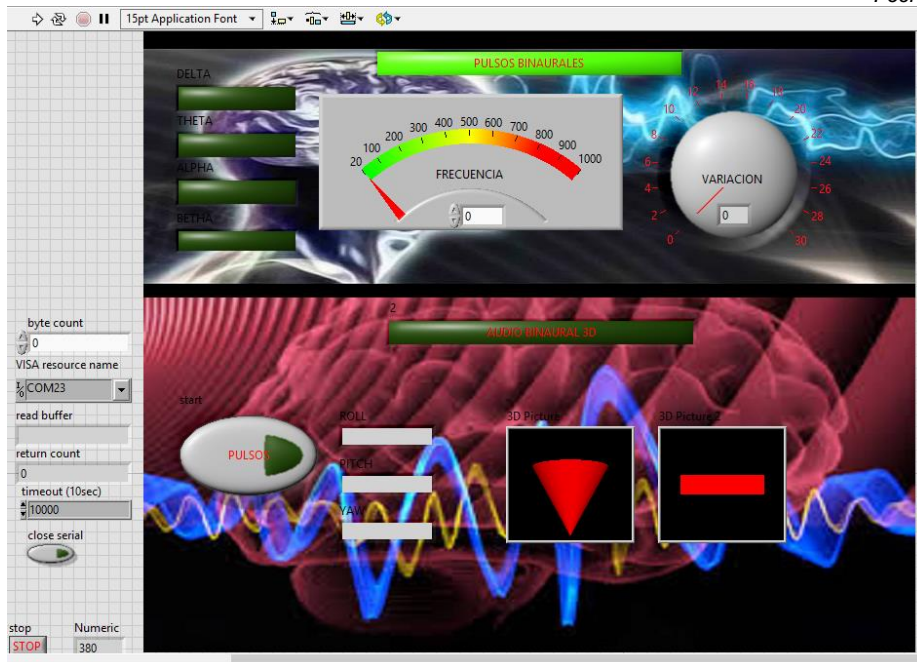


**Figura 17.**Codigo en Labview para la graficación de las figuras

Fuente: Elaboración propia en labview

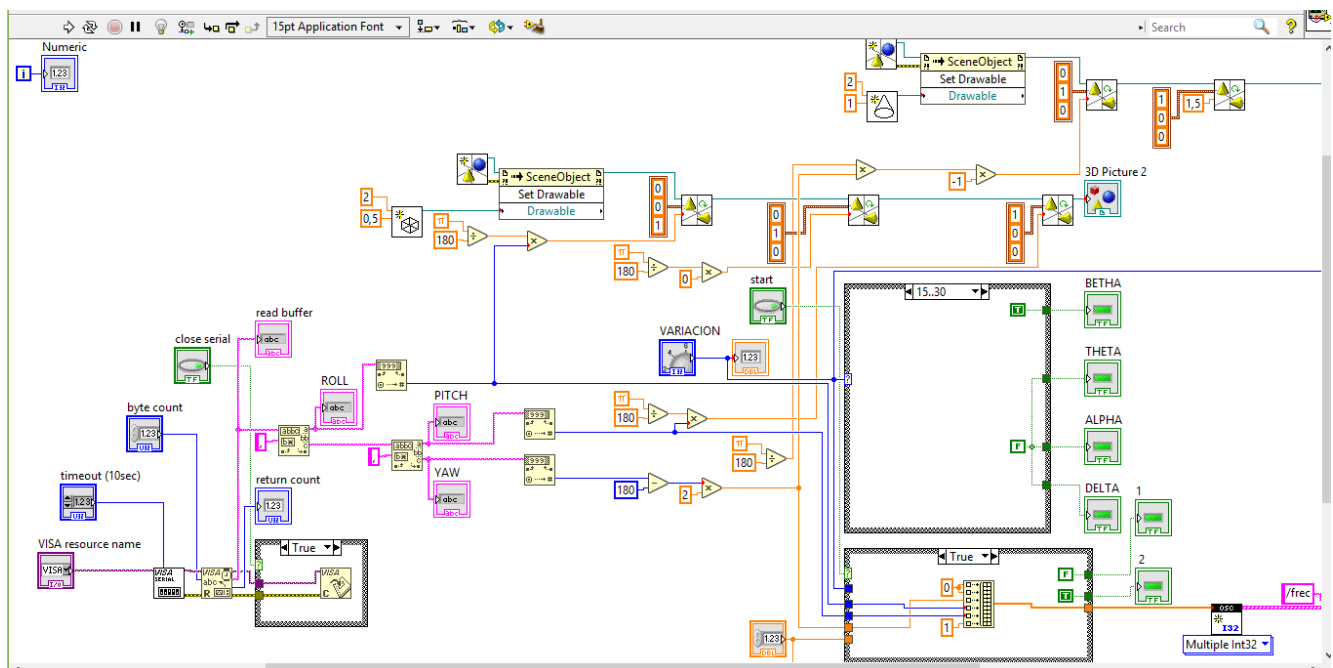
Posteriormente los datos son enviados al procesador de audio pure data, para generar el sonido. La parte grafica del generador se ve completamente en la figura 18.





**Figura 17.**Interfaz gráfica en Labview

Fuente: Elaboración propia en labview



**Figura 18.**Codigo en Labview

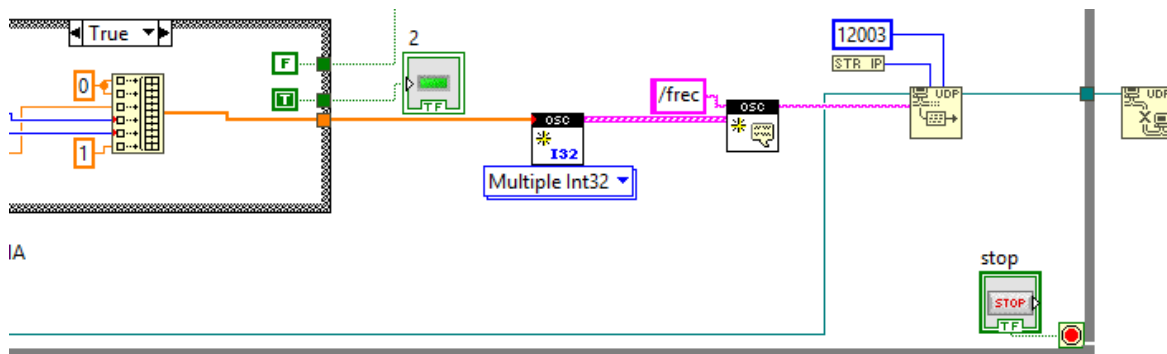
Fuente: Elaboración propia en labview

## 2.2.4 Comunicación UDP/OSC:

Para realizar la comunicación de Labview con Pure Data para procesar el audio se hizo uso del protocolo UDP y el empaquetamiento de datos OSC.

(OSC) Open Sound Control es un protocolo de comunicaciones que permite comunicar instrumentos de música, computadoras y otros dispositivos multimedia (por ejemplo móviles o PDA's equipados con bluetooth) pensado para compartir información musical en tiempo real sobre una red. Aparece como reemplazo del MIDI, siendo muy superior en características y capacidades. [11]

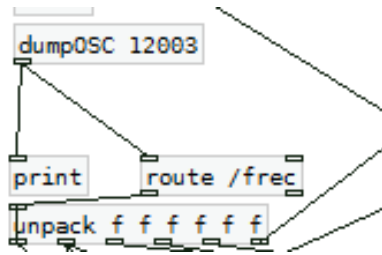
Labview tiene módulos que permiten el empaquetamiento por vectores y por el protocolo OSC como se observa en la figura 19 para posteriormente ser transmitido por un puerto creado UDP en este caso el puerto receptor es el 12003.



**Figura 19.** protocolo osc y udp en Labview

Fuente: Elaboración propia en labview

Todos los datos son transmitidos de Labview a Pure Data por el puerto creado UDP 12003, de modo que en la plataforma de Pure Data es necesario crear este puerto como se ve en la figura 20, y esto se logra con el objeto “dumpOSC” el cual recibe los datos empaquetados como OSC.



**Figura 20.**protocolo osc y udp en Pure Data

Fuente: Elaboración propia en Pure Data

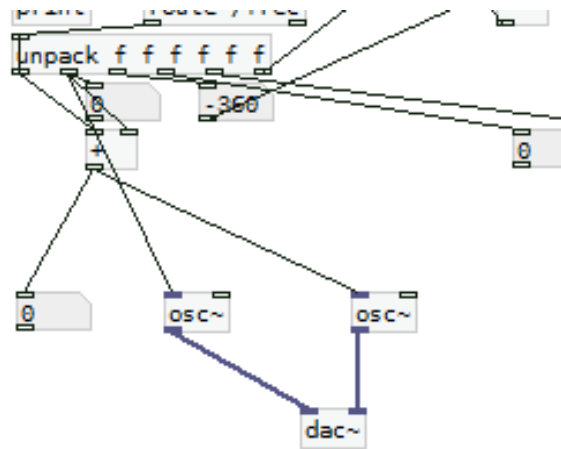
Al lograr transmitir los datos a Pure Data se necesita desempaquetarlos y esta función la cumple el objeto unpack como se ve en la figura 20.

#### **2.2.4 Procesamiento de audio:**

Para generar y procesar el audio se uso pure data la cual es una plataforma y lenguaje de programación visual de código abierto en tiempo real, para audio, video y procesamiento gráfico. Normalmente se utiliza en síntesis de audio y video, interfaz de sensores, análisis de audio, uso de cámaras, control de robots y demás aplicaciones.

La programación con Pure Data se convierte en una experiencia muy cercana a la de manipular cosas en el mundo físico, debido a que su filosofía de programación se basa en la interconexión de cajas a las cuales el usuario define su función, creando las aplicaciones mediante un flujo de datos resultante del proceso realizado[13]

Para generar los pulsos binaurales puros se usaron los objetos (osc~) el cual permite hacer ondas sinusoidales de determinadas frecuencias; para este caso las frecuencias recibida de LabView. Y con el objeto (dac~) se genera el sonido por medio de la tarjeta de audio del pc.



**Figura 20.** Generación de las oscilaciones y salida por el dac en Pure Data

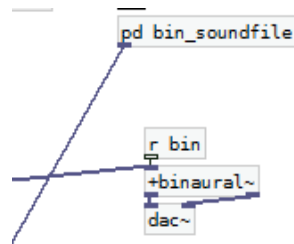
Fuente: Elaboración propia en Pure Data

En cuanto al audio binaural de realidad virtual fue necesario usar dos objetos específicos para generarlos entre los que se encuentra el objeto (+binaural~) encargado de variar el sonido según varia el ángulo azimut.

+ binaural ~ es un filtro que coloca un sonido en una posición específica alrededor de la cabeza del oyente. Hace esto usando filtros que simulan el efecto de filtrado de la cabeza y el oído externo para sonidos en todos los ángulos. Los filtros en + binaural ~ están optimizados para auriculares y solo para una elevación de 0 grados a una frecuencia de muestreo de 44100.

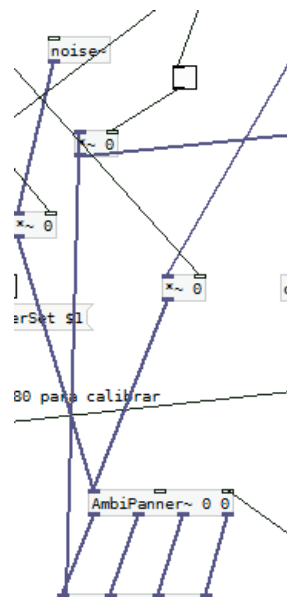
Otro de los objetos importantes es (AmbiPanner~) encargado de dar las elevaciones e inclinaciones al sonido según varíen los ángulos.

Estos dos objetos son de vital importancia para poder procesar cualquier tipo de audio .wav y transformarlo en audio binaural logrando la sensación de realidad.



**Figura 21.** filtro +binaural~ en Pure Data

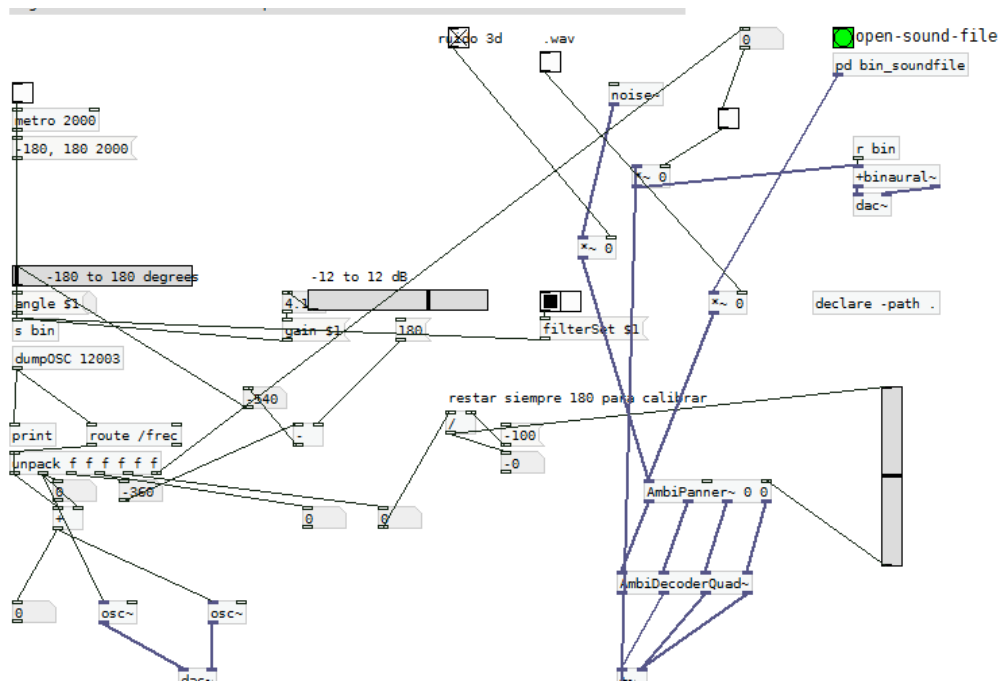
Fuente: Elaboración propia en Pure Data



**Figura 22.** Filtro AmbiPanner en Pure Data

Fuente: Elaboración propia en Pure Data

Los ángulos dados a los filtros que procesan el audio son obtenidos desde LabView. Y Pure Data se encarga de emitir el audio por el dac, es importante tener en cuenta que la salida es estero.



**Figura 23.**Codigo completo en Pure Data

Fuente: Elaboración propia en Pure Data

Algunos de los objetos usados en el código de Pure Data visualizado en la figura 23, se utilizan como control o para mejorar la calidad del audio. Por ejemplo el objeto (filterset ) se utiliza para eliminar un poco el ruido; objetos como los indicadores numéricos solo se usan como método de control para verificar el buen funcionamiento del programa.

Los objetos (\*~0) se usan como métodos de apagar el sonido por lo cual cuando se presionan mutean el dac.

Posteriormente al procesamiento de audio obtenemos la salida, la cual se da por el dac del pc a una frecuencia de muestreo de 44100. Y allí conectamos los auriculares para obtener en ellos el sonido binaural.

### **3 Conclusiones**

Al finalizar la implementación del generador binaural se pueden concluir varias cosas entre ellas la eficiencia de la plataforma Pure Data para procesar y generar audio en tiempo real, sus objetos y librerías permiten que el realizar procesos con audios se vuelva muy simple.

En cuanto a los pulsos binaurales se pudo observar que el efecto binaural después de determinadas frecuencias no era notable. Al realizar varias pruebas, se evidencia que con frecuencias mayores a 1 KHz no es tan notable la diferencia en los hemisferios izquierdo y derecho. Por lo cual se limita el proyecto a frecuencias de 1KHz en el caso de los tonos binaurales puros.

La sensación obtenida en la generación de los sonidos binaurales de 3d es muy aproximada a la del mundo real, por lo cual se cumplen las tres condiciones esenciales de la realidad virtual la inmersión, la presencia y la interacción.

En este proyecto el oyente paso de simplemente escuchar un audio a interactuar con el, gracias a los sensores y dispositivos usados para la comunicación entre lo real y lo virtual.

Ademas se logro enlazar un programador grafico como lo es Labview con un procesador de audio como lo es Pure Data

## Referencias

- [1] E. Ramdinmawii and V. K. Mittal, "The Effect of Music on The Human Mind : A Study Using Brainwaves and Binaural Beats," 2017.
- [2] (en línea) Disponible en: <https://www.rvd-psychologue.com/es/neuroestimulacion-investigaciones-aplicaciones-sonidos-binaurales.html>
- [3] (en línea) Disponible en: <http://www.sonidosbinaurales.com/la-tecnologia-de-realidad-virtual-requiere-sonido-binaural/>
- [4] (en línea) Disponible en: <https://www.futura.space/la-importancia-del-sonido-en-la-realidad-virtual/>.
- [5] J. Benesty, C. Jingdong, Yiteng Huang, "Immersive Audio Schemes," IEEE Trans. Signal Processing, vol. 28, no. 1, pp. 20-32. Enero, 2011
- [6] (en línea) Disponible en: [https://www.google.com/search?q=cw+binaural&client=firefox-bab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjB6rSrte\\_cAhWN7FMKHfJyAwQQ\\_AUICygC&biw=1366&bih=654#imgrc=mq4sn8qjNGBJ8M:](https://www.google.com/search?q=cw+binaural&client=firefox-bab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjB6rSrte_cAhWN7FMKHfJyAwQQ_AUICygC&biw=1366&bih=654#imgrc=mq4sn8qjNGBJ8M:)
- [7] C. B. Ave, D. Number, and R. Date, "MPU-6000 and MPU-6050 Product Specification," vol. 1, no. 408, 2013.
- [8] "Three-Axis Digital Compass IC."
- [9] (en línea) Disponible en: [https://www.google.com/search?q=hmc5883l&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwja07jVw-\\_cAhURzFMKHVIqAu8Q\\_AUICigB&biw=1366&bih=654#imgrc=wlaBnxulyaWQyM](https://www.google.com/search?q=hmc5883l&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwja07jVw-_cAhURzFMKHVIqAu8Q_AUICigB&biw=1366&bih=654#imgrc=wlaBnxulyaWQyM)
- [10] (en línea) Disponible en: <https://os.mbed.com/handbook/mbed-FRDM-KL25Z>
- [11] A. Freed, A. Schmeder, and M. Zbyszynski, "Open Sound Control -- A flexible protocol for sensor networking," Computer (Long. Beach. Calif).
- [12] H. Roncancio and H. Cifuentes, "Tutorial de LABVIEW," Labview Tutor., vol. 1, p. 20, 2000.
- [13] J. F. C. Valencia, "Implementación del seguimiento del movimiento de la cabeza de un individuo, en un sistema de reproducción binaural.," J. Chem. Inf. Model., vol. 53, pp. 1689–1699, 2013.





*Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido*

*Fecha de envío:*

*Fecha de recepción:*

*Fecha de aceptación:*

