

De Sonic Pi a Instrumentos Musicales Precolombinos: Explorando Nuevas Formas de Desarrollar Experiencias Sonoras Creativas

Jean Carlos Correa González
Investigador Independiente
jean_car1993@hotmail.com

Resumen

Esta investigación nace de una inquietud personal, de indagar en dos campos a primera vista dispares: arte y programación. Con el fin de crear una obra sonora interdisciplinar que permita interactuar código de programación con instrumentos precolombinos, dicho de otra manera, investigar otras formas de expresión, otras vías de comunicación, mecanismos y herramientas que se presten al servicio del artista.

Así como en la época pre-colombina nuestros antepasados hacían música con objetos que tenían a su alrededor; plumas, huesos, cerámica, madera. hoy, en nuestra contemporaneidad estamos rodeados por ordenadores, agentes inteligentes (inteligencia artificial) sensores y sin número de gadgets tecnológicos. este trabajo sugiere adoptar este pensamiento ancestral de hacer música con lo que se tiene a disposición, a la mano. Y fusionar el poder de las nuevas tecnologías con instrumentos latinoamericanos precolombinos. Usando como medio de conducción la plataforma de programación en vivo (livecoding) Sonic Pi.

Keywords Improvisation, live coding, software, open source, Arte, Precolombino, Música

Introducción

la tecnología ha avanzado a pasos agigantados en el siglo XX y en lo que va del XXI que durante toda la historia de la humanidad. En la actualidad, el desarrollo de las nuevas tecnologías se ha extendido a todos los campos de la sociedad, “El arte está inmerso en la sociedad, el arte es una forma de expresión, de comunicación, que por tanto está sumida y no es ajena a un contexto. Las nuevas tecnologías han invadido la sociedad en este corto periodo de tiempo y han revolucionado toda manifestación artística” (Almahano, 2011, pág. 42) Las nuevas tecnologías están revolucionando los procesos creativos de manera sorprendente. El computador está tomando el rol de escenario, lienzo, pincel e instrumento musical.

El presente trabajo estudia, explora y juega con la herramienta de programación en vivo Sonic Pi. Y como este software de código abierto (open source) puede aportar en el proceso creativo para el desarrollo de una obra sonora. que tendrá como columna vertebral el uso de instrumentos precolombinos (ocarinas, silbatos, quena).

Gracias a las bondades del software libre, Sonic Pi permite tener una batería, un piano, una línea de bajo y una gran variedad de instrumentos musicales, a partir de comandos de texto. Ahorrando recursos que de otra manera tendrían un costo exorbitante, democratizando la música y las tecnologías digitales. Además, se adapta fácilmente a la manera tradicional de hacer música; respetando las figuras musicales, los tonos, los tiempos, el tempo, la dinámica, la melodía, la armonía e incluso el timbre y transponiéndolos a valores numéricos que se expresan fácilmente en términos de funciones matemáticas abstractas (Aaron & Blackwell, 2013). Lo que significa que una partitura musical puede migrar de manera bastante sencilla a código de programación, demostrando su amplia aplicabilidad y facilidad de adopción.

Lo que se busca es simplemente añadir más posibilidades, que son más amplias en algunos aspectos y más limitadas en otros. (Judge, 2015) Y complementar las carencias con los instrumentos precolombinos y viceversa. la idea es ampliar las posibilidades dentro de un entorno de artes escénicas.

Trabajos Relacionados

Codificación en Vivo (Live Coding)

El medio digital Proyecto Idis (2004) plantea una definición precisa respecto al termino codificación en vivo.

Live coding se refiere a una forma de arte performático y técnica creativa que se basa en el uso de programación interactiva de manera improvisada. Presentaciones en vivo donde uno o varios Live Coders (programadores en vivo) generan visuales y música a medida que avanza la presentación. Tanto las visuales como la escritura del código deben ser proyectadas, la idea es la de improvisar mediante el uso de la escritura de código, para generar un espectáculo audiovisual interactivo para el espectador.

La codificación en vivo es un fenómeno internacional en crecimiento que reúne las habilidades creativas de la creación musical y la programación de computadoras; es un modo de creatividad digital que considera la codificación como una performance. La codificación en vivo amplía las posibilidades del quehacer artístico.

Fundación Raspberry (Raspberry Pi Foundation)

La Fundación Raspberry Pi es una organización benéfica educativa con sede en el Reino Unido, más conocida por la creación de la computadora Raspberry Pi. La Raspberry Pi es una computadora de placa única de bajo costo (\$ 25) que puede ejecutar una versión completa del

sistema operativo Linux. Ha demostrado ser extremadamente popular como plataforma de bajo costo para servidores web, aplicaciones de control, automatización del hogar y otras aplicaciones de pasatiempos. Sin embargo, también puede funcionar como un computador personal (Aaron, Blackwell, & Burnard, 2016).

El desarrollo de Sonic Pi surgió de las actividades de la Raspberry Pi Foundation, que ha estado a la vanguardia del cambio educativo en el Reino Unido. Así lo ha demostrado con el apoyo al proyecto Sonic Pi a través de donaciones al Computer Laboratory de la Universidad de Cambridge.

Sonic Pi

Sonic Pi es un lenguaje de codificación en vivo y un entorno de interpretación con dos objetivos específicos y distintos: en primer lugar, facilitar la enseñanza de la programación en las escuelas y, en segundo lugar, permitir a los artistas ofrecer virtuosas interpretaciones musicales codificadas en vivo (Aaron S. , 2016). Este artículo considera priorizar las implicaciones de los elementos performáticos. Es decir, aprovechar los beneficios que proporciona el uso de Sonic pi en la creación de una obra sonora.

Sonic pi actuará como piso que va a sostener el performance; es donde se desarrolla la base rítmica, la armonía o el ambiente, que estarán en función de los instrumentos precolombinos, con objetivo de resaltarlos y darle protagonismo a estos.

Software de Código Abierto (Software Open Source)

Según la Free Software Foundation, (2021):

el software libre se refiere a la Libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el Software; de modo más preciso, se refiere a cuatro libertades de

los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

Sonic Pi cuenta con una “licencia MIT” de la Open Source Initiative. Una licencia permisiva y gratuita para utilizar software, con la condición principal de que las condiciones de la licencia (entre las que se incluye una atribución del derecho de autor al programador original) se transmitan siempre con el código. (Judge, 2015) Significa que básicamente puedes hacer lo que quieras con el software.

Instrumentos Precolombinos

esta investigación no indagara en el origen, clasificación y detalles arqueológicos de los instrumentos musicales precolombino, más bien será una presentación general de los instrumentos que se usaran en la performance. La descripción de los instrumentos fue tomada del Centro de Documentación de Bienes Patrimoniales de Chile (2021).

Ocarinas y Silbatos

En la figura 1, podemos observar dos ocarinas en la parte superior, una de menor tamaño en la parte inferior izquierda y tres silbatos en la parte inferior derecha.

Figura 1

Instrumentos Precolombinos: Ocarinas y Silbatos



La ocarina es un tipo de flauta usada hasta la actualidad en el mundo andino, normalmente hecha de cerámica, tiene salida de aire y agujeros de digitación y destaca por ser de forma globular, ovalada o elipsoide.

A diferencia de las ocarinas, los silbatos son de menor extensión. Tienen uno o varios agujeros de digitación.

Quena

En la figura 2, se muestran tres quenás, cada una de ellas tiene una afinación diferente.

Figura 2

Instrumentos Precolombinos: Quenas



La quena es una flauta de madera o de caña con agujero de salida y de digitación, usada hasta la actualidad en la música andina.

Zampoña

En la figura 3, podemos observar una zampoña de 8 tubos.

Figura 3

Instrumentos Precolombinos: Zampoña



Este aerófono es la unión de varios tubos que varían de tamaño a fin de matizar el sonido de acuerdo a la extensión por la cual el viento debe pasar y salir. Se fabricaron en madera, piedra y hueso, y fueron de uso extensivo en toda Sudamérica.

Presentación del Entorno Sonic Pi

SonicPi es un lenguaje que se basa en el lenguaje de programación de computadoras Ruby, pero se ha simplificado para los principiantes con el objetivo de ser un sistema de codificación en vivo fácil de usar. También es trivial de instalar (no requiere una configuración compleja).

Tiene una interfaz amigable e intuitiva. permite escribir el código directamente en una ventana de texto, Un botón "run" interpreta el código. cuenta con un menú de ayuda donde está la documentación oficial que detalla todas sus funciones y métodos además de incluir un gran repositorio de ejemplos. A continuación, se muestra una imagen (figura 4.) del entorno de desarrollo de Sonic Pi.

Figura 4*Entorno de Desarrollo de Sonic Pi*

Sonic Pi permite hacer sonidos increíblemente rápido sin tener que tener ningún conocimiento previo de síntesis de sonido, invocados por un conjunto de comandos. A continuación, en la tabla 1 se describe de manera general algunos de los comandos que se usaron para desarrollo del performance.

Tabla 1*Comandos Básicos que se Usaron para el Desarrollo del Performance*

Comandos Sonic Pi	Descripción
play	Toca una nota con el sintetizador actual. se pueden pasar argumentos opcionales para controlar aún más el sintetizador; Estos argumentos opcionales son únicos para cada sintetizador y se definen como parte de la descripción del sintetizador.
sleep	Permite elegir cuánto tiempo esperar antes de ejecutar la siguiente instrucción.

Comandos Sonic Pi	Descripción
<code>use_synth</code>	Cambia el sintetizador de sonido que estamos usando.
<code>play_pattern_timed</code>	Recibe dos listas como parámetro, uno con las notas y otro con los tiempos. Reproduce las notas MIDI especificadas con los tiempos de separación especificados; la lista de tiempos es una lista de duraciones para el sleep entre cada nota.
<code>sample</code>	Reproduce un archivo de sonido grabado (Sample). Sonic Pi viene con muchas muestras geniales incluidas, pero también se puede cargar y reproducir archivos .wav, .wave, .aif, .aiff, .ogg, .oga o .flac desde cualquier lugar del computador.
<code>play_chord</code>	Reproduce una lista de notas al mismo tiempo (acordes). Acepta argumentos opcionales para modificar el sintetizador que se está reproduciendo.
<code>use_bpm</code>	Permite alterar el tempo.
<code>stop</code>	Detiene todos los sonidos.

Nota. Tabla adaptada de la documentación oficial de Sonic Pi. (Raspberry Pi Foundation, 2021)

Para familiarizarnos con las Figuras musicales y su equivalencia con los tiempos en Sonic Pi. Se muestra la figura 5. donde un “sleep” con valor 4 representa un silencio de redonda y uno con valor 0.25 representa una semicorchea y así proporcionalmente con las demás figuras musicales.

Figura 5

Entorno de Desarrollo de Sonic Pi



Nota. Imagen reproducida de notas musicales en Sonic Pi, Mehackit, 2021

(www.mehackit.org)

Implementación

El desafío principal para esta investigación fue migrar una obra musical icónica del folklore Latinoamericano “ojos azules” (figura 6.) a código de programación y a su vez permitir la interacción con los instrumentos musicales físicos (quena, tambor, silbatos).

En este apartado veremos cómo se correlacionan los conceptos de programación y su equivalencia con la música occidental tradicional. las melodías ahora son secuencias; los ritmos repetidos, son iteración; las líneas de bajo o riffs, ahora serán listas. Esto es posible porque todas y cada una de las ideas musicales de Sonic Pi están inmediatamente representadas por una o más técnicas básicas de informática (Aaron S. , 2018).

Lo que es emocionante aquí desde una perspectiva musical es que al secuenciar el **play** y el **sleep**, se tiene acceso a la mayor parte de la música occidental. En otras palabras, con solo dos comandos se puede recrear cualquier melodía o ritmo conocido.

Figura 6

Partitura de la canción Ojos Azules

OJOS AZULES

MANUEL CASAZOLA HUANCOCO

TRANSCRIPCIÓN: EDINSO JOSÉ CORONADO HUAMÁN.

MI MI MI SOL MI re MI DO. DO re MI SOL re si DO LA.

C D C D E7 Am

5 DO DO DO LA sol DO re MI DO re MI SOL re si DO LA.

C G C D E7 Am

1 - Ojos azules no llores,
no llores ni te enamores.
Llorarás cuando me vaya,
cuando remedio no haya.

2 - Tu me juraste quererme
quererme toda la vida.
No han pasado ni tres días,
y ya me estas engañando.

3 - En un potito de chicha
quisiera tener veneno
Veneno para olvidarte
veneno para olvidarte.

Nota. Imagen adaptada de ojos azules & poco a poco, José Coronado, 2016

(www.partiturasjosecorh.blogspot.com)

Como ejemplo de la simplicidad del lenguaje, el primer compás de ojos azules se puede codificar de la siguiente manera:

```
play :e4
sleep 0.5
play :e4
sleep 0.5
play :e4
sleep 0.5
play :g4
sleep 0.5
```

Todo lo que está sucediendo arriba es que el código especifica qué notas tocar, en qué orden tocarlas y los tiempos de pausa entre cada nota. el comando **play** indica que nota va a sonar y el comando de **sleep** le dice a la computadora que espere 0.5 segundo (una corchea) entre la

ejecución de la siguiente instrucción de **play**. La figura 7. Expresa una forma más tradicional de notación para la misma secuencia de notas.

Figura 7

Partitura de los Cuatro Primeros Compases de la Canción Ojos Azules



Nota. Imagen adaptada de ojos azules & poco a poco, José Coronado, 2016

(www.partiturasjosecorh.blogspot.com)

A continuación, se detallará la codificación de los elementos fundamentales de la música: la melodía, armonía y el ritmo, para la creación de la presente performance.

Melodía

Para no escribir una gran cantidad comandos **play** y **sleep** una y otra vez, como se vio anteriormente, Sonic Pi nos permite usar un atajo: **play_pattern_timed**.

En vez de escribir el código anteriormente detallado para el primer compás. se escribió:

```
play_pattern_timed [:e4, :e4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
```

La primera lista `[:e4, :e4, :e4, :g4]` es el grupo de notas y la segunda lista `[0.5, 0.5, 0.5, 0.5]` es el grupo de silencios entre las notas.

Entonces, la melodía de los 4 primeros compases se expresa de la siguiente manera:

```
play_pattern_timed [:e4, :e4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
play_pattern_timed [:e4, :d4, :e4, :c4], [1, 0.5, 0.5, 1]
play_pattern_timed [:c4, :d4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
play_pattern_timed [:d4, :b3, :c4, :a3], [1, 0.5, 0.5, 2]
```

En la figura 8. Se observa la notación tradicional para los cuatro primeros compases.

Figura 8

Partitura de los Cuatro Últimos Compases de la Canción Ojos Azules



Nota. Imagen adaptada de ojos azules & poco a poco, José Coronado, 2016

(www.partiturasjosecorh.blogspot.com)

Las líneas de código que representa los 4 compases finales de la melodía son las siguientes:

```
play_pattern_timed [:c4, :c4, :c4, :a3], [0.5, 0.5, 0.5, 0.5]
play_pattern_timed [:g3, :c4, :d4, :e4], [1, 0.5, 0.5, 1]
play_pattern_timed [:c4, :d4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
play_pattern_timed [:d4, :b3, :c4, :a3], [1, 0.5, 0.5, 2]
```

en la figura 9. Se observa la notación tradicional para los cuatro últimos compases.

Figura 9

Partitura de los Cuatro Primeros Compases de la Canción Ojos Azules



Nota. Imagen adaptada de ojos azules & poco a poco, José Coronado, 2016

(www.partiturasjosecorh.blogspot.com)

Ahora para finalizar con la melodía. Se quiere que los cuatro primeros compases se repitan dos veces y los cuatro últimos también se repitieran dos veces. El término concreto en informática

para esto es iteración, el cual sólo significa hacer cosas más de una vez. Sonic pi nos ofrece la solución a esto con el comando **times**. Y es tan fácil como escribir **2.times do**. Y escribir **end** al final del código que queremos repetir.

Las líneas de código para la melodía completa se expresan así:

```
2.times do
  play_pattern_timed [:e4, :e4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
  play_pattern_timed [:e4, :d4, :e4, :c4], [1, 0.5, 0.5, 1]
  play_pattern_timed [:c4, :d4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
  play_pattern_timed [:d4, :b3, :c4, :a3], [1, 0.5, 0.5, 2]
end
2.times do
  play_pattern_timed [:c4, :c4, :c4, :a3], [0.5, 0.5, 0.5, 0.5]
  play_pattern_timed [:g3, :c4, :d4, :e4], [1, 0.5, 0.5, 1]
  play_pattern_timed [:c4, :d4, :e4, :g4], [0.5, 0.5, 0.5, 0.5]
  play_pattern_timed [:d4, :b3, :c4, :a3], [1, 0.5, 0.5, 2]
end
```

Armonía

La armonía para este arreglo musical de ojos azules consta de tres acordes: do mayor, mi mayor y la menor. Para tocar un acorde ya no usaremos el comando **play**, que solo reproduce el sonido de una nota. Para reproducir un acorde tenemos que usar el comando **play chord**.

```
play chord(:c3, :major)
sleep 4
play chord(:c3, :major)
sleep 3
play chord(:E3, :major)
sleep 2
play chord(:A3, :minor)
sleep 2
```

Para que esta secuencia de acordes ser repitan a manera de loop o bucle se hizo uso del comando **live_loop**. Que permite iterar el conjunto de código que esta entre el **do** y el **end**, además se permite cambiar el comportamiento de un **live_loop** mientras aún se está ejecutando

sin necesidad de detenerlo. Y a la vez se puede sincronizar con otros **live_loops**. Por lo tanto, los **live_loops** son el secreto de la codificación en vivo con Sonic Pi.

Las líneas de código para reproducir el ciclo armónico de ojos azules quedo así:

```
live_loop :armonia do
  use_synth :fm
  play chord(:c3, :major),release: 3
  sleep 4
  play chord(:c3, :major),release: 2
  sleep 3
  play chord(:E3, :major)
  sleep 2
  play chord(:A3, :minor)
  sleep 2
end
```

Base rítmica

Una vez que se dominó el **play** y el **sleep**, el siguiente comando con el que se jugo fue: **sample**. que brinda la capacidad de reproducir cualquier sonido pregrabado. Sonic Pi incluye una gran cantidad de grabaciones integradas para usar, como baterías, guitarras y ruidos atmosféricos. Este comando se lo uso para el patrón rítmico de ojos azules (Huayno). El patrón del Huayno es el siguiente: Tres pulsos, un remate, cuatro pulsos, un remate.

Para simular un bombo se usó el comando: **sample :drum_bass_hard**. Y por último para que este ritmo se repita, se usó el comando **live_loop** que a su vez estará sincronizado con el **live_loop** de la armonía y melodía. Dando como resultado una obra sonora completa.

```
live_loop :bombo do
  3.times do
    sample :drum_bass_hard, amp: rand(0.9,1.5)
    sleep 1
  end
  2.times do
    sample :drum_bass_hard, amp: rand(0.9,1.5)
    sleep 0.5
  end
  4.times do
```

```

    sample :drum_bass_hard, amp: rrand(0.9,1)
    sleep 1
end
2.times do
    sample :drum_bass_hard, amp: rrand(0.9,1)
    sleep 0.5
end
sample :drum_cymbal_closed, amp: rrand(0.9,1)
sleep 0.5
sample :drum_cymbal_pedal, amp: rrand(0.9,1)
sleep 1.5
end

```

La función **play** presenta algunas opciones para retocar el sonido, una de ellas es **amp**: que permite modificar la amplitud o volumen. Y con la función **rrand()** devuelve un valor aleatorio entre un número mínimo y un número máximo. Este proceso se realizó para darle un sonido más orgánico a la base rítmica.

Performances Virtuosas

Con la armonía, la melodía y el ritmo desarrollados se puede empezar a jugar con estos elementos. Por ejemplo: se puede detener con un **stop** el **live_loop** de la melodía, y ahora ejecutar la melodía con la Quena (instrumento musical físico); también podemos aplicar efectos como un **reverb** a la base rítmica; se puede jugar con las opciones del sintetizador como **release** que permite que una nota dure más o menos tiempo; Cambiar el sintetizador que se está usando para generar diferentes matices y texturas, pasar la melodía de un **piano** a un sintetizador **fm**, o a un **dark_ambience**, o a un **beep**; jugar con los cambios de amplitud con la opción **amp**; acelerar el tempo con **use_bpm**; se puede detener la armonía y la melodía para dejar solo la base rítmica sonando y que entren en juego los sonidos de las ocarinas y silbatos, o reproducir solo la armonía y jugar con los instrumentos sobre esta. Y todo esto en tiempo real. Sin duda Las posibilidades para el performance son infinitas.

Sonic Pi viene con aproximadamente 164 samples que puedes libremente usar y jugar, pero también tiene soporte para usar samples externos. Para sacar provecho a esto se recopiló una serie de samples de ambiente como: el fluir de un río, el croar de ranas, la selva, el fuego. También se recopiló samples de voces, relatos y discursos.

Las siguientes líneas de código reproducen un sonido ambiente de una fogata, y encima de este ambiente se entonarían melodías con los instrumentos musicales precolombinos.

```
ambientes = "C:/Users/jean_/Desktop/samples"
use_bpm 30
live_loop :ambiente1 do
  sample ambientes, "fire", beat_stretch: 4, amp: 2
  sleep 4
end
```

Cabe recalcar que, en el transcurso del performance, este código ira evolucionando y puede adoptar perfectamente las funcionalidades antes descrita. Esto en informática se conoce como abstracción o dicho de una manera más sencilla “reutilizar el código”.

Conclusión

Se puede concluir que es completamente posible hibridar estos dos campos (arte y programación) y, además se aportan mucho mutuamente. Como dato anecdótico e importante; Toda la documentación, proyectos y ejemplos que se revisaron para el desarrollo de esta investigación están relacionados a la música anglosajona (Electronic Dance Music, Dubstep, Slow Techno Grime, Acid House y otros). Y sería bueno empezar a deconstruir y decolonializar la manera de abordar el quehacer investigativo de la actualidad. Y que no se mal interprete, en lo personal disfruto sobremanera de la música anglo. Pero hay que empezar explorar otras posibilidades más cercanas a nuestra idiosincrasia latinoamericana.

El presente trabajo demostró que se puede hacer perfectamente folklore latinoamericano acompañado de las nuevas tecnologías, la intención también es motivar a que más investigadores indaguen en ritmos y sonoridades más cercanas a nuestra identidad.

Bibliografía

- Aaron, S. (2016). Sonic Pi – performance in education, technology and art. *International Journal of Performance Arts and Digital Media*, 171-178.
- Aaron, S. (2018). Live Coding Education. *The MagPi Educators Edition*, 44-47.
- Aaron, S., & Blackwell, A. F. (2013). From Sonic Pi to Overtone: Creative Musical Experiences with Domain-Specific and Functional Languages. *Proceedings of the ACM SIGPLAN International Conference on Functional Programming, ICFP*, 35-46.
- Aaron, S., Blackwell, A. F., & Burnard, P. (2016). The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology & Education*, 75-94.
- Almahano, M. L. (2011). DANZA CONTEMPORÁNEA Y NUEVAS TECNOLOGÍAS. *Danzaratte*, 42.
- Centro de Documentación de Bienes Patrimoniales . (2021). *cdbp.patrimoniocultural.gob.cl*.
Obtenido de *cdbp.patrimoniocultural.gob.cl*:
https://www.cdbp.patrimoniocultural.gob.cl/652/w3-article-55226.html?_noredirect=1
- Free Software Foundation. (2021). *fsf.org*. Obtenido de *fsf.org*: www.fsf.org
- Judge, J. (2015). Sonic Pi: la programación informática y la creatividad se dan la mano. *OMPI Revista*. Obtenido de https://www.wipo.int/wipo_magazine/es/2015/03/article_0007.html
- Raspberry Pi Foundation. (2021). *raspberrypi.org*. Obtenido de *raspberrypi.org*:
<https://projects.raspberrypi.org/en/projects/getting-started-with-sonic-pi>

www.mehackit.org. (2021). *www.mehackit.org*. Obtenido de www.mehackit.org: <http://sonic-pi.mehackit.org/exercises/es/01-introduction/02-play-a-melody.html>

www.partiturasjosecorh.blogspot.com. (2016). *partiturasjosecorh*. Obtenido de partiturasjosecorh: <https://partiturasjosecorh.blogspot.com/2016/09/ojos-azules-poco-poco.html>

www.proyectoidis.org. (2004). *proyectoidis.org*. Obtenido de proyectoidis.org: <https://proyectoidis.org/live-coding/>