



---

Memoria para optar al título de Ingeniero en Sonido con  
mención en Diseño y Producción Sonora

# **Octopus Mapper: una interfaz para la exploración creativa de timbres en instrumentos musicales electrónicos**

---

Ingeniería en Sonido  
Facultad de Artes  
Universidad de Chile

**Estudiante:** Pablo Madariaga  
**Profesor Guía:** Javier Jaimovich  
**Profesor Informante:** Leonardo Cendoyya  
Santiago, junio de 2023

## Resumen

En el siguiente documento se expone el proceso de diseño e implementación de *Octopus*, una interfaz de control que permite al usuario manejar de forma simultánea hasta ocho parámetros de un instrumento electrónico compatible. La mecánica de *Octopus* busca facilitar la exploración de timbres, la aparición de sonidos interesantes, la automatización de parámetros, fomentar la comprensión del instrumento y ofrecer una nueva manera de interactuar con instrumentos electrónicos.

El desarrollo de *Octopus* comprendió tres etapas: una fase inicial de investigación, seguida de una fase de diseño e implementación y finalmente una fase de prueba con sujetos de experimentación. La investigación abarcó áreas como *mapping*, diseño de instrumentos electrónicos, de interfaces y de interacciones, además de la exploración y estudio de otras herramientas utilizadas como referentes. Por su parte, la fase de diseño e implementación consistió principalmente en una dinámica reiterativa de programación y evaluación del código que daría forma al primer prototipo. Por último, la fase de prueba consistió en presentar individualmente el prototipo a un total de cuatro personas, seleccionadas debido a su experiencia previa con instrumentos electrónicos. Después de probar el prototipo, las personas respondieron una entrevista semiestructurada diseñada para obtener información relevante. Los resultados de esta última fase fueron bastante consistentes, lo que resultó útil para esclarecer la percepción general de *Octopus*, así como para identificar las fortalezas y debilidades del prototipo y explorar posibles vías para profundizar en el desarrollo de esta herramienta.

Concretamente, los resultados evidencian que *Octopus* permite un notable ahorro de tiempo en tareas relacionadas con la exploración sonora, promueve la generación de timbres “inesperados” o “sorprendentes”, posibilita a los usuarios la obtención de

sonidos que no serían accesibles mediante su método tradicional de trabajo y fomenta el pensamiento divergente en lugar del pensamiento convergente. Por otra parte, los entrevistados tendieron a considerar a *Octopus* como un complemento de su forma habitual de trabajar en lugar de una oportunidad para adoptar un enfoque completamente novedoso. Asimismo, la mayoría de los usuarios criticó la incapacidad del prototipo para guardar cambios realizados o permitir un trabajo continuo a través de múltiples sesiones. Por último, los resultados confirman en cierta medida que las estrategias de *mapping pocos-a-varios* se acercan más y son más apropiadas para la fase de exploración creativa de ideas, tal como se ha propuesto en la literatura especializada.

## Índice

1. Introducción .....	8
1.1. Objetivos.....	10
1.1.1. Objetivo General .....	10
1.1.2. Objetivos específicos .....	10
2. Marco Teórico .....	11
2.1. Introducción.....	11
2.2. Instrumentos Musicales Electrónicos.....	11
2.2.1. Construcción del concepto .....	11
2.2.2. Estructura de un instrumento musical electrónico .....	14
2.3. Mapping .....	16
2.3.1. La definición de <i>mapping</i> .....	16
2.3.2. Formas de clasificar estrategias de <i>mapping</i> .....	17
2.3.3. El diseño de la estrategia de <i>mapping</i> y su relación con el objetivo perseguido.....	19
2.4. Los sintetizadores comerciales.....	23
2.4.1. La interfaz genérica de los sintetizadores comerciales .....	23
2.4.2. El problema del <i>mapping</i> en los sintetizadores comerciales .....	24
2.5. Interpolación de parámetros .....	26
2.6. Conclusiones y reflexiones del capítulo de marco teórico.....	29
3. Referentes .....	31
3.1. Herramientas para el control de instrumentos electrónicos.....	31
3.1.1. PHY-Series .....	31
3.1.2. Gradient .....	33
3.1.3. Shaker .....	34
3.1.4. Kaossilator .....	35
3.1.5. Kaoss Pad.....	36
3.1.6. Touché.....	37
3.1.7. Vector Map .....	38

3.2.	Conclusiones y reflexiones del capítulo de referentes.....	39
4.	Método.....	41
4.1.	Características generales de la investigación.....	41
4.2.	Metodología, técnica e instrumentos. ....	42
4.3.	Análisis de los datos.....	44
5.	Principios de diseño.....	45
5.1.	Hacia la formulación de una propuesta de diseño.....	45
5.1.1.	El espacio combinatorio.....	45
5.1.2.	El uso de la interpolación lineal.....	50
5.1.3.	Interpolando entre presets.....	52
5.1.4.	Las “combinaciones vértice” y la creación de sub-espacios combinatorios.....	54
5.2.	La propuesta de diseño.....	58
5.2.1.	Características deseables de la herramienta.....	60
6.	Desarrollo e implementación del primer prototipo.....	62
6.1.	Herramientas a utilizar.....	62
6.2.	Implementación de Octopus.....	63
6.2.1.	Comunicación herramienta-instrumento:.....	63
6.2.2.	Selección y manipulación de rangos:.....	65
6.2.3.	Interpolación:.....	68
6.2.4.	Módulo de distribuciones.....	71
6.2.5.	Módulo de trayectorias.....	87
7.	Resultados.....	94
7.1.	Perfiles de los entrevistados.....	94
7.2.	Respuestas de los entrevistados.....	95
7.3.	Análisis de los resultados.....	99
7.3.1.	Impresiones generales de Octopus.....	99
7.3.2.	Sobre la incorporación de Octopus en la práctica musical.....	104
7.3.3.	El enfoque divergente en Octopus.....	108

8. Discusión y conclusiones .....	111
8.1. Discusión .....	111
8.1.1. Sobre el desarrollo y logro de los objetivos .....	111
8.1.2. Sobre aspectos generales de Octopus y de la investigación .....	115
8.2. Conclusiones .....	119
8.3. Proyecciones .....	121
8.3.1. Lo técnico .....	121
8.3.2. Sobre la metodología e investigación .....	124
8.3.3. Más allá de los instrumentos musicales electrónicos .....	125
9. Bibliografía .....	126
ANEXOS .....	135
Anexo A: Guía de entrevista .....	135
Objetivos .....	135
Perfil del usuario .....	135
Descripción de la experiencia .....	135
Preguntas .....	136
Consentimiento informado .....	137
Anexo B: Links a videos demostrativos .....	138

## **Agradecimiento**

A mi profesor guía, Javier Jaimovich, por su enorme sentido de la responsabilidad y su compromiso con mi trabajo; a los participantes del estudio, por su buena disposición para desarrollar las pruebas y por su ojo crítico aplicado a mi prototipo; a los profesores Leonardo Cendoyya y Sean Moscoso, por ampliar mi mirada sobre el tema de investigación; a Claudia, por estar siempre presente, y a todas las personas que desde distintos lugares hicieron posible la existencia de esta memoria.

# 1. Introducción

Desde su aparición hasta la actualidad los instrumentos musicales electrónicos (IME) se han extendido y abierto paso desde los más avanzados laboratorios hasta las habitaciones e incluso los computadores de los artistas de hoy, y por lo mismo, no es ajeno para nadie el gran impacto que han tenido en el desarrollo de la música. Su importancia ha sido tal que incluso han nacido campos de estudio completos alrededor de ellos, siendo la conferencia NIME<sup>1</sup> un ejemplo insigne.

Uno de los principales objetos de estudio en el campo del diseño de instrumentos musicales electrónicos (IME) es el *mapping*, es decir, la forma en la que los controles del instrumento se relacionan con los parámetros sonoros del sistema. Dado su naturaleza arbitraria, diversas líneas de investigación se han encargado de estudiar y clasificar las diferentes estrategias de *mapping* que han sido utilizadas a lo largo del tiempo, y por lo mismo, se han ido construyendo diferentes posturas respecto de la manera en la que estas deberían estar diseñadas. Dentro de esta discusión, una de las ideas más frecuentes es que las estrategias de *mapping* del tipo *uno-a-uno*, si bien son las más comunes, traen consigo una serie de problemas relacionados con la expresividad musical, la utilización de interfaces complejas y con ser poco “amigables” con usuarios inexpertos.

Ahora bien, el problema identificado en esta investigación radica en que, a pesar del amplio abanico de posibilidades, las compañías fabricantes de IME han tendido a unificar el formato de sus instrumentos, llegando a lo que se ha identificado como la **interfaz genérica**, la cual entre otras cosas incluye el uso de mapeos exclusivamente *uno-a-uno* (Seago, Holland, Mulholland 2004; Gibson, Polfreman 2019). El problema se agudiza cuando se considera que se trata de instrumentos

---

<sup>1</sup> [www.nime.org](http://www.nime.org) (The International Conference on New Interfaces for Musical Expression | NIME sin fecha)

producidos en masa que tienen un gran alcance, por lo que una considerable cantidad de usuarios se enfrentan a instrumentos cuya estrategia de *mapping* presenta una serie de problemas ya descritos.

El objetivo de este trabajo fue implementar una herramienta que permite aplicar una estrategia de *mapping* del tipo *dos-a-varios* (o divergente) a un IME. Se decidió tal estrategia debido a que se ha demostrado que facilita la exploración creativa de timbres y porque además impone una forma diferente de pensar y relacionarse con el instrumento (Dixon, Tubb 2014; Le Vaillant, Dutoit, Giot 2020). Para ello se utilizó el protocolo MIDI con el cual se pudo acceder de manera directa al generador de sonido del instrumento eludiendo así la estrategia de *mapping* implementada de fábrica. Finalmente, se presentó un prototipo a un grupo de personas quienes entregaron una retroalimentación que fue de utilidad para determinar los caminos a seguir.

El siguiente documento consta de un capítulo de marco teórico en el que se profundiza principalmente conceptos asociados a los instrumentos musicales electrónicos, el *mapping* y los interpoladores gráficos. Posteriormente se incluye un capítulo de referentes en el que se exponen un conjunto de herramientas que cumplen con características similares a las que se pretende llegar con el prototipo. Se incluye además un breve capítulo de metodología en el que se busca explicitar las técnicas a utilizar para llevar a cabo la investigación. Luego de esto último se presenta un capítulo de fundamentos de diseño, en el que se expone la lógica que opera detrás de *Octopus*, para posteriormente llegar al capítulo de desarrollo en el que se documenta la implementación de la herramienta. Debido a que parte de la investigación incluyó una etapa de experimentación con sujetos de prueba, se incluyó un capítulo de resultados en el que se exponen las respuestas de las personas

a una entrevista semiestructurada que les fue aplicada. Finalmente, el documento termina con una sección de discusión, proyecciones y conclusiones.

## **1.1. Objetivos**

### **1.1.1. Objetivo General**

- Generar una herramienta de software que posibilite la exploración creativa de las capacidades tímbricas de un instrumento electrónico mediante la reducción de sus parámetros de control a una interfaz bidimensional.

### **1.1.2. Objetivos específicos**

- Proponer estrategias de mapeo dos-a-varios utilizando técnicas de reducción de dimensiones para poder condensar el espacio de parámetros de un instrumento electrónico en un plano bidimensional.
- Implementar la estrategia de mapeo por medio de una herramienta digital que sea capaz de controlar de forma externa los parámetros del generador de sonido del instrumento electrónico.
- Evaluar las capacidades creativas, exploratorias y renovadoras de la herramienta mediante una encuesta realizada a sujetos de prueba que estén familiarizados con el uso de instrumentos musicales electrónicos.

## 2. Marco Teórico

### 2.1. Introducción

El capítulo presentado a continuación aborda cuestiones fundamentales para la comprensión y construcción del problema de investigación. Se profundiza en el concepto de instrumento musical electrónico, en la arquitectura de estos y en aspectos de diseño relacionados principalmente a sus estrategias de *mapping*. Luego, se pone especial atención a aquellos instrumentos electrónicos denominados ‘sintetizadores comerciales’, y se problematiza el hecho de que su interfaz, su diseño de *mapping*, su instrumentalidad<sup>2</sup> y sus posibilidades sónicas son elementos que se encuentran en disonancia. Finalmente, se ahonda en algunas técnicas de reducción de parámetros utilizadas para el control de sintetizadores y que podrían resultar útiles para soslayar el problema antes descrito. Si bien dichas técnicas no son recientes, estas han recobrado relevancia durante los últimos años debido al renovado interés por parte de los investigadores, quienes se han encargado de darles una mirada más actual.

### 2.2. Instrumentos Musicales Electrónicos

#### 2.2.1. Construcción del concepto

Uno de los esfuerzos más tempranos para definir y clasificar a los instrumentos musicales electrónicos (IME) fue presentado por Curt Sachs en 1940, en su libro llamado ‘*The history of musical instruments*’ (Magnusson 2017). Sachs, quien ya había co-creado uno de los sistemas más extendidos para la clasificación de instrumentos (Bakan et al. 1990; Kartomi 2001; Stéphanie Weisser, Maarten Quanten 2011), conocido como el sistema Hornbostel-Sachs (H-S), introdujo a su modelo una quinta

---

<sup>2</sup> Concepto utilizado por Hardjowirogo en su artículo ‘*Instrumentality. On the Construction of Instrumental Identity*’ (Hardjowirogo 2017).

categoría llamada **electrófonos**, en la que intentaba dar cabida a las nuevas tecnologías musicales que estaban apareciendo hacia la primera mitad del siglo XX (Kartomi 2001; Magnusson 2017). Sachs se refiere a los IME como aquellos que dependen de los descubrimientos físicos propios del siglo XX, entre ellos osciladores, filtros, bobinas, micrófonos y amplificadores (Sachs 2006)<sup>3</sup>.

Si bien los electrófonos de Sachs resultaron ser una categorización útil para la época, el acelerado avance de las tecnologías musicales durante la segunda mitad del siglo XX puso en crisis su modelo (Bakan et al. 1990; Paine 2010; Stéphanie Weisser, Maarten Quanten 2011; Magnusson 2017). En el artículo '*Demystifying and classifying electronic music instruments*' (Bakan et al. 1990), los autores sostienen que los electrófonos del modelo H-S son una categoría miscelánea en la que se agrupa todo tipo de instrumento que tenga alguna relación con la electricidad, por más mínima que sea. Por lo mismo, los autores proponen repensar la quinta categoría del modelo H-S, renombrándola como *electronophones* y estableciendo que en ella se incluirán sólo aquellos **instrumentos que producen sonido por medios exclusivamente electrónicos** (Bakan et al. 1990)<sup>4</sup>.

Nociones más recientes en el campo de los IME incluyen la aparición del concepto *Digital Musical Instrument (DMI)*. En el libro '*New Digital Music Instruments: Control And Interaction Beyond the Keyboard*' (Miranda, Wanderley 2006), los autores

---

<sup>3</sup> Invenciones como el Theremin (1924), el Spharophon (1926), el Superpiano (1927), el Hellertion (1928), las Ondas Martenot (1928), el Trautonium (1930), el Emicon (1930), el Neo-Bechstein (1931), el Radio Organ (1932), el órgano Hammond (1939) y el Novachord (1939) son instrumentos explícitamente mencionados por Sachs como ejemplos de electrófonos (Sachs 2006).

<sup>4</sup> Los autores además consideraron importante la categorización de dispositivos musicales tales como secuenciadores y procesadores de sonido que, si bien no producen sonido por sí mismos, tienen un rol determinante en el resultado sonoro de la máquina que estén controlando. Para ello propusieron una nueva categoría que llamaron *Modifiers*. Los *electronophones* en conjunto con los *modifiers* son llamados por los autores como el sistema de clasificación GAMES (Generators And Modifiers of Electronic Sound).

definen a los DMI como aquellos instrumentos que utilizan la síntesis digital de sonido (generada por computador) como medio para la producción sonora, además de caracterizarlos como instrumentos que constan de una **superficie de control** (o *gestural controller*) que maneja los parámetros del sintetizador de sonido en tiempo real. Miranda y Wanderley profundizan en este último aspecto estableciendo que los *gestural controllers* pueden tomar cualquier forma y estar implementados con cualquier tipo de sensor, por lo que las posibilidades a la hora de diseñar un DMI son muy amplias. De lo anterior se puede concluir que para pensar un DMI no basta solo con considerar la forma en la que se produce el sonido, sino que también se debe tener en cuenta la etapa de control, es decir, pensar la forma en la que se interactúa con él, pensar en qué gestos se quiere utilizar sobre el controlador, qué sensores van a leer dichos gestos, cómo estos gestos van a afectar a los parámetros sonoros, qué tipo de *feedback* va a entregar la interfaz, entre otros aspectos.

La relevancia que han tenido los DMI a lo largo del siglo XXI puede verse reflejada en el hecho de que muchos de ellos son presentados año a año en la conferencia NIME (McPherson et al. 2016), en la cual se abordan diversos tópicos que tienen que ver con tecnologías musicales tales como el diseño de interfaces, la interacción humano-computador, instrumentos aumentados, música generativa, instalaciones sonoras, aplicaciones de *machine learning*, performance, etcétera (Fasciani, Goode 2021). Algunos ejemplos notables de DMI son la *reactTable* (Kaltenbrunner et al. 2006), el *T-Stick* (Malloch, Wanderley 2007), el *Illusio* (Barbosa et al. 2013), el *CD-Synth* (Chwalek, Paradiso 2019) y más recientemente el *Electronic Khipu* (Cadavid 2020). Todos estos instrumentos, si bien tienen en común la producción de sonido mediante síntesis digital, cuentan con interfaces de usuario que son completamente diferentes entre sí.

### 2.2.2. Estructura de un instrumento musical electrónico

Cuando se habla de su estructura, diferentes autores coinciden en que los IME constan de al menos dos unidades independientes: un controlador y un generador de sonido (Diliberto 1983; Chadabe 2002; Hunt, Wanderley, Paradis 2002; Paradiso, O'Modhrain 2003; Jordà 2004; van Nort, Castagné 2007; Hardjowirogo 2017; de las Pozas 2020).

El controlador es el componente del IME que funciona como la interfaz de usuario o *set* de comandos que está disponible para ser utilizada por el intérprete, quien a través de gestos específicos es capaz de controlar la forma en la que el sonido es emitido (Goudeseune 2002; Jordà 2004). En (Bakan et al. 1990) se refieren al controlador como el dispositivo en el que **efectivamente** se interpreta/toca la música electrónica. Por otro lado, en (Miranda, Wanderley 2006) los autores establecen que los controladores pueden tomar cualquier forma, desde una figura que se asemeje a un instrumento acústico (teclados, guitarras, flautas, etc.) hasta lo que ellos llaman *alternate controllers*, los cuales no tienen semejanza alguna con otros instrumentos musicales. Además, los autores indican que un controlador puede estar implementado con cualquier tipo de sensor, cuya señal puede ser utilizada para controlar los parámetros del sintetizador de sonido. Tecnologías como botones, sliders, pantallas táctiles, acelerómetros, giroscopios, sensores infra-rojos, videocámaras, radares, sensores de proximidad o sensores biométricos son solo algunos ejemplos de los dispositivos que se pueden utilizar como sensores de entrada para controlar un IME (Chadabe 2002; Miranda, Wanderley 2006; Magnusson 2017).

Por otra parte, como su nombre lo indica, el generador de sonido (GS) es el componente de naturaleza electrónica (*hardware o software*) que funciona como la fuente sonora del IME (Chadabe 2002). En trabajos como (Wanderley, Depalle 2004)

los autores establecen que el GS puede ser visto como **el algoritmo de síntesis del instrumento y sus parámetros de entrada**, aunque esto último podría ser una definición limitada ya que no necesariamente se debe utilizar la síntesis de sonido como técnica de base para la producción sonora. Respecto a esto último, si bien en la clasificación GAMES (Bakan et al. 1990) se menciona la síntesis analógica y digital como los procesos fundamentales de algunos IME, también se incluye la utilización de trozos de audio (*samples*) grabados analógica o digitalmente como la técnica sobre la cual se puede construir un generador de sonido. Otros métodos con los que se puede implementar un generador de sonido incluyen el modelado físico, la síntesis concatenativa y la síntesis granular (Magnusson 2017). Por otro lado, en el proyecto de investigación 3DMIN han desarrollado el concepto de instrumentos Post-DMI, estableciendo entre otras cosas que un instrumento de dichas características combina en su funcionamiento señales acústicas con síntesis y procesamientos digitales (Hinrichsen, Bovermann 2016), expandiendo así el rango de técnicas bajo las cuales puede operar el generador de sonido de un IME.

A diferencia de un instrumento acústico tradicional, en el que existe una relación indisoluble entre el cuerpo del instrumento y la fuente emisora del sonido (Hunt, Wanderley, Paradis 2002), en un IME el sonido no es un resultado de las características acústicas del material con el que está construido, ni proviene directamente de la interacción con alguno de los componentes físicos del controlador (Hardjowirogo 2017; Magnusson 2017). Es debido a lo anterior que la relación entre el controlador y el módulo generador de sonido tiene que ser definida (Hunt, Wanderley, Paradis 2002). Dicha relación es conocida como *mapping*.

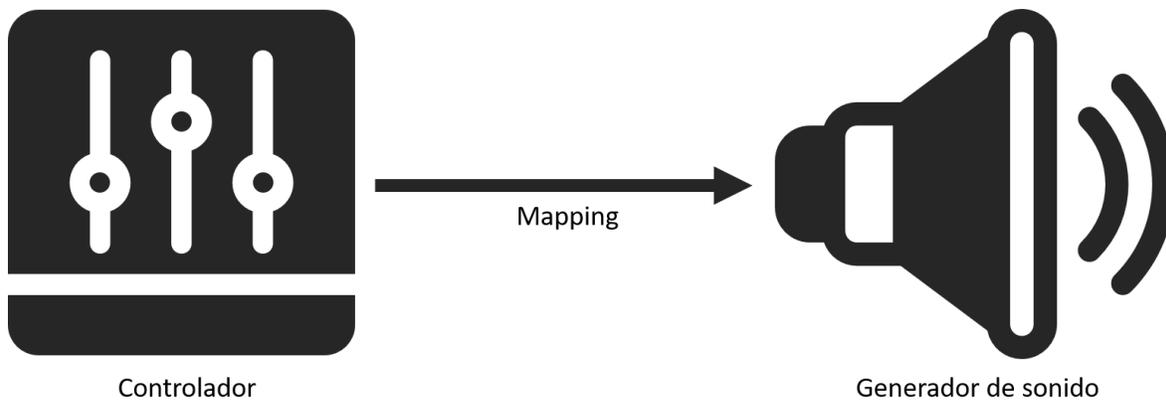


Figura 1: Estructura de un instrumento musical electrónico

## 2.3. Mapping

### 2.3.1. La definición de *mapping*

En la literatura referente a instrumentos electrónicos, cuando se busca definir el concepto de *mapping* por lo general se hace a través de metáforas que lo asocian con la idea de vínculo, de unión, de conexión o de correspondencia entre los elementos de un IME. Comúnmente el *mapping* es pensado a nivel de la arquitectura del IME, es decir, como la conexión entre el *output* del controlador y el *input* del generador de sonido (Van Nort, Wanderley, Depalle 2014). En esta línea existen nociones tempranas como la de Don Buchla, quien en una entrevista para la revista *Polyphony* esboza una definición de *mapping* cuando dice que un instrumento electrónico consta de tres partes: una estructura de entrada, una estructura de salida y **una conexión entre las dos** (Diliberto 1983). En este sentido, en (Hunt, Wanderley, Paradis 2002) los autores se refieren al *mapping* como **el arte de conectar el controlador con el generador de sonido**. Así mismo, existen definiciones más acotadas como la de Chadabe, quien sostiene que el *mapping* describe **la manera en la que un control está conectado a una variable** (Chadabe 2002). McGlynn, por su parte, defiende una postura similar pero más amplia, estableciendo que el *mapping* describe **la forma en la que los datos recopilados por el dispositivo de entrada se**

**relacionan con los parámetros musicales del sistema** (Mcglynn 2011). En (Wang et al. 2019) se habla del *mapping* como **la traducción de las señales de entrada en el sonido resultante**. Otras posturas en torno al *mapping* consideran no solo la arquitectura del IME, sino que también el sistema que se forma entre la gestualidad del usuario y el resultado sonoro. En esta línea, en (Rovan et al. 1997) los autores remarcan la importancia que tiene para la expresividad de un instrumento la forma en la que la información gestual es **mapeada** a los parámetros de síntesis. Por su parte, Goudeseune establece que el *mapping* es **el mundo que se encuentra entre los gestos y los sonidos**, y que en torno a él se pueden hacer tres preguntas: ¿Desde qué? ¿Hacia qué? ¿Y de qué manera? (Goudeseune 2002). Así mismo, en (Paradiso, O'Modhrain 2003) los autores se refieren al *mapping* como **la cuestión en torno a asociar las capacidades del sistema sensomotor humano con el espacio de parámetros del instrumento que está siendo tocado**. En resumen, si bien la definición de *mapping* no es única, siempre que se habla al respecto subyace la idea de un puente que conecta una cosa con otra.

### **2.3.2. Formas de clasificar estrategias de *mapping***

Al no existir ninguna relación inherente entre el controlador y el generador de sonido se dispone de total libertad a la hora diseñar la forma en la que ambos elementos se conectan (Bevilacqua, Müller, Schnell 2005), es decir, el *mapping* es un proceso esencialmente arbitrario (Malloch, Wanderley 2007; Magnusson, Hurtado 2017), y por lo mismo, no es de extrañar que haya sido llamado el **arte** de conectar el controlador con el generador de sonido (Hunt, Wanderley, Paradis 2002).

Dada la amplia gama de posibilidades se han hecho esfuerzos por categorizar y describir las distintas estrategias de *mapping* bajo diversos criterios de clasificación. Hunt y Wanderley establecen una división general entre dos formas de aproximarse al problema del *mapping*: *mapping* implícito y *mapping* explícito (Hunt, Wanderley

2002). El primero tiene que ver con la utilización de redes neuronales que funcionan como una especie de caja negra auto-organizada a la que no es posible acceder, y el segundo tiene que ver con utilizar estrategias en las que se puede elegir exactamente qué parámetros de entrada están asociados a qué parámetros de salida. Por otro lado, Rován et al. proponen una clasificación basada en la forma en la que se conectan los elementos del controlador con los del GS, la cual se divide en tres grupos: *mapping* uno-a-uno, *mapping* divergente<sup>5</sup> y *mapping* convergente<sup>6</sup> (Rován et al. 1997). El primero consiste en la forma más simple de *mapping*, en la que un gesto específico sobre el controlador está asociado a un solo parámetro del generador de sonido<sup>7</sup>. En el *mapping* divergente, un gesto específico es utilizado para controlar dos o más parámetros sonoros. Finalmente, en el *mapping* convergente un parámetro del sonido es controlado por dos o más gestos sobre el controlador. Cualquier combinación entre las categorías anteriores se podría considerar un *mapping* varios-a-varios (Hunt, Wanderley 2002). Por su parte, Kvifte se interesa en encontrar formas adecuadas de describir los modelos de *mapping* denominados complejos, tales como el varios-a-varios (Kvifte 2008). Para ello el autor estudia el funcionamiento de las interfaces de algunos instrumentos acústicos, argumentando que de esa forma se puede tener claridad del grado y tipo de complejidad deseado. Kvifte desarrolla los conceptos de *value selectors* y *set selectors*, que son dos maneras de clasificar la forma en la que operan los gestos de entrada sobre el sonido resultante, concluyendo que tal forma de pensar el *mapping* puede establecer similitudes entre instrumentos que bajo otros sistemas de clasificación no tendrían mayor relación. Por otro lado, Mcglynn ha propuesto pensar el *mapping* desde un

---

<sup>5</sup> También conocido como uno-a-varios.

<sup>6</sup> También conocido como varios-a-uno.

<sup>7</sup> Por ejemplo, rotar un potenciómetro que esté asociado a la frecuencia de un oscilador, o variar con los movimientos de una mano la señal emitida por un sensor de proximidad que esté asociado a la amplitud de un oscilador.

enfoque que clasifica la información gestual ingresada al controlador (*performance data*), ordenándola en tres niveles ascendentes de complejidad: *Raw Data*, *Symbolic/Semiotic Data* y *Gestural Data* (Mcglynn 2011). El autor sostiene que clasificar la información del controlador de acuerdo a **cómo** se utilizará y no de acuerdo a su procedencia permite evitar replicar antiguos paradigmas para así avanzar en dirección al desarrollo de sistemas interactivos más complejos, elegantes y satisfactorios. Más recientemente, en (Van Nort, Wanderley, Depalle 2014) los autores identifican tres puntos de vista sobre los cuales se puede pensar el *mapping*: una visión a nivel de sistema (*system point of view*) que está enfocada en la interconexión entre los parámetros del controlador y los parámetros sonoros, una visión a nivel funcional (*functional point of view*) que tiene un enfoque holístico para la construcción de espacios de parámetros, y una visión perceptual (*perceptual point of view*) que está asociada a la percepción de la intencionalidad y de la expresividad en la interpretación. **Para el desarrollo de esta investigación se optó por utilizar el sistema de clasificación propuesto por Rován et. al**, esto debido a su carácter descriptivo y lo extendido que está en la literatura.

### **2.3.3. El diseño de la estrategia de *mapping* y su relación con el objetivo perseguido**

Los criterios bajo los cuales se diseña una estrategia de *mapping* dependen del diseñador del instrumento electrónico y del objetivo al que se quiera llegar con él. A continuación se expondrán tres corrientes principales que se han identificado en el transcurso de esta investigación, nombradas por el autor como enfoque expresivo, enfoque paramétrico y enfoque exploratorio.

Uno de los principales –y primeros– esfuerzos en el diseño de IME será referido en esta investigación como **enfoque expresivo**. Este enfoque busca alcanzar con los IME un nivel de interacción similar al que se tiene con instrumentos acústicos

tradicionales (Wanderley, Depalle 2004), principio por el cual se han realizado diversos experimentos y modelos teóricos que buscan estudiar la influencia del *mapping* sobre la relación entre el intérprete y el instrumento. Entre los autores que defienden esta postura se encuentra Kvifte, quien sostiene que uno de los principales problemas en el diseño de instrumentos electrónicos radica en el uso de interfaces con numerosos controles que dificultan alcanzar la expresividad en la interpretación, y que una forma de solucionar este problema es estudiar el diseño de instrumentos acústicos tradicionales (Kvifte 2008). En la misma línea, en (Hunt, Wanderley, Paradis 2002) los autores presentan los resultados de tres experimentos que, si bien tienen diferencias metodológicas, todos concluyen en que el *mapping* es un factor determinante en la interacción (o *feeling*) ofrecida por el instrumento. Se concluye además que utilizar un diseño de *mapping* complejo<sup>8</sup>, semejante al ofrecido por los instrumentos tradicionales, resulta más atractivo para el usuario, posibilita el desarrollo de cierto virtuosismo, permite una mayor expresividad y podría hacer la diferencia entre un simple 'juguete' y un instrumento musical. Sumado a lo anterior, en (Fels, Gadd, Mulder 2003) se afirma que al igual que sucede en los instrumentos acústicos tradicionales los instrumentos electrónicos deberían aspirar a tener un *mapping* 'transparente', es decir, que exista una relación evidente entre el gesto del intérprete y el sonido resultante, cosa que posibilitaría alcanzar la expresividad musical con el instrumento. El enfoque subyacente en los trabajos anteriormente expuestos busca utilizar el *mapping* con la finalidad de que el IME se comporte como lo haría un instrumento acústico tradicional, y para ello los diseñadores tienden a utilizar estrategias de *mapping* complejas del tipo varios-a-

---

<sup>8</sup> Vale decir la utilización de mapeos del tipo varios-a-varios por sobre la utilización de mapeos exclusivamente uno-a-uno.

varios, ya sea programándolas de forma explícita o ayudándose de herramientas de *machine learning* tales como *Wekinator* (Fiebrink, Trueman, Cook 2009).

Un segundo enfoque, referido en este trabajo como **enfoque paramétrico**, tiene que ver con otorgar al usuario un control fino y detallado sobre el generador de sonido, es decir, permitirle manejar de forma individual tantos parámetros como el diseñador estime necesario. En favor de lo anterior, es común que este tipo de implementaciones cuenten con interfaces de elevado número de controles (Rasmussen 2018; Esling et al. 2020) mapeados a los parámetros del sintetizador mediante estrategias de tipo uno-a-uno (Hunt, Kirk 2000; Gibson, Polfreman 2019). La simpleza de este tipo de estrategias podría ser la razón por la que la mayoría de los sintetizadores modernos están diseñados con ellas (Rovan et al. 1997; Gibson, Polfreman 2019).

Por último, el acá denominado **enfoque exploratorio** utiliza las estrategias de *mapping* con el objetivo de explorar el espacio de parámetros del IME, es decir, para poder navegar por sus posibilidades tímbricas. En esta línea existen trabajos como (Gibson, Polfreman 2019), donde se afirma que una buena alternativa para facilitar la creatividad y el diseño de nuevos sonidos es la utilización de interpoladores gráficos, los cuales utilizan estrategias del tipo pocos-a-varios y permiten al usuario explorar diferentes combinaciones de los parámetros del sintetizador. Otro ejemplo de este enfoque está en (Dixon, Tubb 2014), donde interesados por saber de qué forma se relaciona el *mapping* con las etapas del proceso creativo, los autores desarrollan un experimento para poner a prueba la hipótesis de que la estrategia *uno-a-varios* y la estrategia *uno-a-uno* son particularmente adecuadas para el pensamiento divergente y el pensamiento convergente respectivamente. Los resultados obtenidos sugieren que la hipótesis propuesta es correcta, siendo la

estrategia *uno-a-varios*<sup>9</sup> preferida para la exploración de ideas y la estrategia *uno-a-uno*<sup>10</sup> preferida para ajustes finos del sonido. De manera similar al trabajo anterior, en (Le Vaillant, Dutoit, Giot 2020) se realiza un experimento en el que los individuos deben escuchar y memorizar el timbre de diferentes secuencias tocadas en un sintetizador de cuatro parámetros, para luego intentar recrear el sonido por medio de dos interfaces: una ‘holística’, que permite controlar simultáneamente los cuatro parámetros por medio de un *mapping uno-a-varios*, y una ‘analítica’, que cuenta con cuatro sliders que controlan de forma independiente cada parámetro gracias a un *mapping uno-a-uno*. Los resultados demuestran que la interfaz holística es más amigable, atractiva y permite a los usuarios un mejor desempeño a la hora de explorar los diferentes timbres de un sintetizador. En resumen, el enfoque exploratorio busca facilitar la exploración de timbres, el diseño de sonido y el *sound morphing*, por lo general a través del uso de estrategias *uno-a-varios* o *pocos-a-varios*.

En los párrafos anteriores se han descrito tres formas diferentes de utilizar el *mapping*: una que busca lograr una similitud entre instrumentos electrónicos e instrumentos acústicos, nombrada ‘**enfoque expresivo**’; una que tiene como objetivo otorgar al usuario un control detallado e individualizado sobre los parámetros del generador de sonido, nombrada ‘**enfoque paramétrico**’; y una que tiene como finalidad fomentar la exploración creativa de timbres tomando en cuenta aspectos cognitivos, nombrada ‘**enfoque exploratorio**’. Cabe destacar que las categorías descritas anteriormente no son absolutas, es decir, existen puntos de encuentro entre ellas y el hecho de que un instrumento esté implementado con alguna no quiere decir que esté completamente imposibilitado de realizar tareas asociadas a las otras, sino que simplemente será más dificultoso de realizar (ej. Un IME con enfoque

---

<sup>9</sup> Controlada a través de un plano bidimensional que puede ser recorrido

<sup>10</sup> Controlada a través de un grupo de *sliders*.

paramétrico permite explorar timbres pero controlando un parámetro a la vez, a diferencia de un IME con enfoque exploratorio que con solo mover un cursor se producen grandes cambios en el sonido).

## 2.4. Los sintetizadores comerciales

### 2.4.1. La interfaz genérica de los sintetizadores comerciales

El impacto de los sintetizadores comerciales<sup>11</sup> desde su aparición hasta la actualidad ha sido notable tanto por el hecho haber definido nuevos géneros musicales como por toda la industria que se ha generado a su alrededor (Pinch, Trocco 2002). Sin embargo, y a pesar de la amplia gama de posibilidades a la hora de diseñar un IME, los fabricantes han tendido a unificar el formato de estos sintetizadores dotándolos de controladores con forma de teclado<sup>12</sup> (Wessel, Wright 2002; Miranda, Wanderley 2006) y de interfaces con un elevado número de controles (Gibson, Polfreman 2019; Esling et al. 2020). La convergencia que ha tenido el diseño de los sintetizadores comerciales permite hablar de la existencia de una **interfaz genérica**, la cual no sólo se limita al mundo físico sino que también es replicada en sintetizadores virtuales (Seago, Holland, Mulholland 2004). A partir de lo anterior es que cabe la pregunta en torno al enfoque de *mapping* que está presente en la interfaz genérica y cómo se relaciona esto con la usabilidad de los sintetizadores comerciales. Tomando en cuenta que diversos autores han notado la elevada cantidad de controles presentes en estos instrumentos (Kvifte 2008; Rasmussen 2018; Esling et al. 2020), y sumando el hecho de que estos controles suelen estar mapeados por medio de estrategias *uno-a-uno* (Hunt, Kirk 2000; Gibson, Polfreman 2019), es razonable sostener que estos sintetizadores se encuentran bajo un **enfoque paramétrico** en lo que respecta a su

---

<sup>11</sup> Por sintetizador comercial se entiende aquel sintetizador (de *hardware* o *software*) que es producido en masa con la intención de ser vendido (Pinch, Trocco 1998).

<sup>12</sup> Producto conocido como *keyboard synthesizer* (Bakan et al. 1990)

estrategia de *mapping*, es decir, privilegiando el control detallado de los parámetros sonoros por sobre aspectos como la gestualidad, la expresividad, la facilidad para explorar sonidos o el control simultáneo de parámetros.

#### **2.4.2. El problema del *mapping* en los sintetizadores comerciales**

Dada la consolidación que ha tenido a lo largo de los años, la interfaz genérica de los sintetizadores comerciales pareciera cumplir con las expectativas de los compradores y fabricantes, sin embargo, su diseño no está libre de inconvenientes, existiendo críticas tanto del enfoque 'expresivo' como del enfoque 'exploratorio'. En primer lugar, existen autores que han explicitado la necesidad de desarrollar mejores estrategias que simplemente las relaciones *uno-a-uno* (Hunt, Wanderley, Paradis 2002). En esta línea se argumenta que, si bien son más fáciles de entender, son menos atractivas, no se asemejan a las existentes en instrumentos acústicos, no incentivan el desarrollo de virtuosismo y tienden a empobrecer la experiencia musical tanto del usuario como de la audiencia (Hunt, Kirk 2000), llegando incluso a afirmar que es la estrategia menos expresiva de todas (Rovan et al. 1997). Sumado a lo anterior, se dice que un elevado número de controles dificulta que el sintetizador tenga un comportamiento musical expresivo (Kvifte 2008) y además obstaculiza la tarea de diseñar nuevos sonidos (Gibson, Polfreman 2019), situación que empeora cuando se trata de usuarios inexpertos en la materia (Esling et al. 2020). Goudeseune expande esta discusión considerando además los *recursos* del usuario, es decir, sus limitaciones físicas y cognitivas, concluyendo que reducir la cantidad de controles vuelve el instrumento más amigable con el intérprete y favorecería una interpretación más rica musicalmente (Goudeseune 2002).

Como es descrito en (Hunt, Wanderley, Paradis 2002), la estrategia de *mapping* empleada en un IME puede determinar por completo el comportamiento del instrumento, la forma en la que se interactúa con él y finalmente la sensación que el

intérprete tendrá de este. Sin embargo, el hecho de que un instrumento cuente con una estrategia de *mapping* asociada a un objetivo en particular no quiere decir que aquello sea lo único que se espere del instrumento, ni lo único que este pueda llegar a hacer. En el caso de los sintetizadores comerciales, el contar con una estrategia de *mapping* considerada poco expresiva no quita el hecho de que estos artilugios sean definidos –y utilizados– como instrumentos musicales, con todas las implicaciones que esto conlleva, es decir, que entre otras cosas funcionen como un medio de expresión musical y que permitan una performance expresiva (Hardjowirogo 2017). No en vano muchos de ellos incorporan una interfaz con forma de teclado cuyas teclas son sensibles a la fuerza con la que se tocan, hecho que entrega atisbos de la dimensión músico-expresiva presente en estos instrumentos. Así mismo, a pesar de tener una estrategia de *mapping* que lo dificulta, los sintetizadores han sido publicitados desde sus inicios como instrumentos capaces de generar nuevos sonidos y que permiten explorar nuevas posibilidades tímbricas (Anthony, Nelson, Tripsas 2016), volviéndose esto último parte de su identidad como instrumento.

El asunto, entonces, no es que un enfoque de *mapping* con un objetivo en particular elimine toda chance de acercarse a otro tipo de interacciones, sino que simplemente lo dificulta. En los sintetizadores habitan simultáneamente el potencial de controlar detalladamente el sonido, el potencial de ser utilizados en un contexto músico-expresivo y el potencial de explorar un amplio espacio de parámetros sonoros, la cuestión es con qué facilidad se pueden hacer cada una de estas cosas, siendo el enfoque de *mapping* la respuesta a esta interrogante. Al contar con un enfoque paramétrico sólo se facilita lo primero, surgiendo la pregunta por aquellos usuarios que quisieran explotar las otras posibilidades latentes en estos sintetizadores ¿Qué alternativas tienen los usuarios? ¿están destinados a solo conformarse con el diseño ofrecido por los fabricantes? ¿es inevitable encontrarse con las limitaciones del

enfoque paramétrico y de la interfaz genérica? ¿existe la posibilidad de alterar la interacción ofrecida por el instrumento? ¿se puede implementar una nueva estrategia de *mapping* sobre estos instrumentos? Si así fuese ¿supondría esto que los sintetizadores se comporten de manera tal que faciliten nuevos tipos de interacciones? Y finalmente ¿significaría esto un beneficio para los usuarios? Estas preguntas son especialmente relevantes considerando que la literatura especializada tiende a estar escrita en función de instrumentos que van a ser construidos en un futuro, y no de instrumentos como los sintetizadores comerciales, que ya vienen acabados de fábrica y cuyos usuarios no tienen acceso a la etapa de diseño.

A partir de los cuestionamientos anteriores, y aprovechando que los componentes de un IME son separables e independientes entre sí, es que surge la inquietud por encontrar maneras alternativas de aproximarse al control de los sintetizadores comerciales.

## **2.5. Interpolación de parámetros**

A grandes rasgos, la interpolación es un procedimiento matemático a través del cual se busca encontrar/construir el comportamiento de una función tomando como base un grupo de puntos conocidos (Steffensen 2006). No existe una única manera de hacerlo y las técnicas varían según los objetivos que se tengan, sin embargo, los métodos de interpolación más empleados son *Distance Weighting*, *Inverse Distance Weighting*, *Kriging*, *Interpolating Polynomials*, *Natural Neighbour Interpolation*, *Moving Average*, entre otros (Lam 1983, p. 19; Yang et al. 2004).

Ampliamente utilizada en diversas disciplinas, la interpolación no ha pasado desapercibida para el área de la tecnología musical. Goudeseune establece que controlar el espacio de parámetros del generador de sonido a través de un controlador cuyo espacio de parámetros sea de menores dimensiones promueve una

serie de ventajas para el usuario, y que una buena forma de generar un mapeo entre ambos espacios es por medio de la interpolación (Goudeseune 2002). En línea con lo anterior, la interpolación se ha utilizado para la creación de interfaces interactivas que, generalmente a través de un *mapping* dos-a-muchos, permiten la modificación, exploración y mutación de *presets*. Un *preset* es el conjunto de valores que tienen los parámetros del generador de sonido en un instante determinado (Gibson, Polfreman 2019) cuya acción conjunta tiene como resultado un sonido en particular. Cuando los usuarios de los sintetizadores diseñan un sonido que les agrada tienen la posibilidad de guardarlo en forma de *preset* para poder acceder a él de forma rápida en un futuro. La interpolación, entonces, permitiría al usuario transformar gradualmente un *preset X* en un *preset Y* por medio del cálculo de los puntos que forman el recorrido entre los valores de ambos *presets*.

A lo largo de los años han surgido diversas propuestas que utilizan la interpolación como técnica para controlar instrumentos musicales electrónicos. Un ejemplo temprano de ellos es SYTER, una estación de procesamiento digital de señales desarrollada a principio de los ochenta que contaba con una pestaña llamada INTERPOL (Todoroff 2002). En ella, el usuario podía controlar el tamaño y la posición de un grupo de círculos sobre un plano bidimensional, en el que cada uno estaba asignado a un *preset*. Por medio de un cursor el usuario podía explorar el plano y dependiendo del tamaño y posición de cada círculo es que se generaba una interpolación entre los valores de cada *preset*. En el año 2001 se presenta *Interpolator* (Spain, Polfreman 2001), una interfaz que propone una metáfora basada en las lámparas. En ella el usuario recorre un plano bidimensional donde cada *preset* está representado por un foco que emana un haz de luz desde cierta posición, con cierto grado de apertura y cierta dirección. La interpolación ocurre en aquellas zonas en las que los focos se intersectan, lo que a diferencia de SYTER otorga un *feedback* visual

de la zona de influencia de cada *preset*. Otro trabajo similar es *The Metasurface* (Bencina 2005), una interfaz gráfica en la que los *presets* están representados en un plano bidimensional por polígonos contiguos. El usuario puede explorar el plano por medio de un cursor y dependiendo de su posición ocurre una interpolación calculada a través del método *Natural Neighbour*. Por otro lado, el ambiente de programación Max/MSP ofrece un objeto llamado *nodes* que en conjunto con el objeto *patrrstorage* permite la interpolación de *presets* (Gibson, Polfreman 2019). Por su parte, Martin Marier desarrolla un nuevo método de interpolación denominado *Intersecting N-spheres*, a través del cual construye una interfaz que permite la interpolación de *presets* a través de un *mapping* varios-a-varios (Marier 2012).

Recientemente se ha renovado el interés por este tipo de interfaces, esta vez motivado por preguntas en torno al tipo de interacción que ofrecen al usuario y la forma en la que esta afecta a la usabilidad de los instrumentos electrónicos. En esta línea, Gibson y Polfreman analizaron las características de algunos interpoladores gráficos notables y posteriormente desarrollaron un listado de los diez puntos más relevantes para su evaluación e implementación (Gibson, Polfreman 2019). Consecutivamente, los autores diseñaron un experimento para determinar el impacto de diferentes visualizaciones sobre la forma en la que los usuarios exploran el plano bidimensional (Gibson, Polfreman 2020a), para finalmente presentar *Star Interpolator* (Gibson, Polfreman 2020b), una interfaz que no busca ofrecer un nuevo método de interpolación sino que una nueva forma de visualización en la que los *presets*, además de estar distribuidos en un plano bidimensional, entregan de manera sencilla información sobre los valores de cada uno de sus parámetros. De esta forma, el usuario puede monitorear lo que sucede con cada parámetro sin perder la capacidad de controlarlos de manera sistémica. Paralelamente, en (Le Vaillant, Dutoit, Giot 2020) se retoman los conceptos de ‘enfoque analítico’ y ‘enfoque

holístico' propuestos por Hunt y Kirk, y se aplican a un experimento que compara el desempeño de los usuarios al utilizar una interfaz de *mapping* uno-a-uno y al utilizar una interfaz de interpolación gráfica (*mapping* dos-a-varios). Los resultados muestran que esta última facilita la búsqueda de sonidos y acorta positivamente la brecha entre usuarios con distintos niveles de experticia.

Dado lo anterior es que los interpoladores gráficos parecieran ser una herramienta viable para el control alternativo de sintetizadores comerciales, ofreciendo tanto una nueva interfaz de control como una nueva estrategia de *mapping*.

## **2.6. Conclusiones y reflexiones del capítulo de marco teórico**

En las páginas anteriores se ha profundizado en tópicos que resultan fundamentales para situar esta investigación. El definir el concepto de instrumento musical electrónico, el entender su estructura, el haber ahondado en el estudio del *mapping* y el haberse detenido en el caso de los sintetizadores comerciales fue clave para poder encontrar un espacio que no está del todo cubierto y en el que una investigación podría ser pertinente.

En cuanto a la construcción del problema como tal, en primer lugar se encuentra que fruto del proceso de investigación se lograron identificar las estrategias de *mapping* más utilizadas y posteriormente se propuso una clasificación basada en el objetivo al que se apunta con cada una de ellas. La clasificación en enfoque expresivo, paramétrico y exploratorio, si bien claramente no es del todo rigurosa, resultó útil para comenzar a vislumbrar el hecho de que los sintetizadores comerciales tienen el potencial de habitar un espacio que se encuentra en las intersecciones de estas categorías. Dicho potencial podría llegar a ser desbloqueado por medio de herramientas que permitan controlar al sintetizador de formas alternativas,

complementando a su interfaz original sin que eso implique necesariamente una remodelación destructiva del instrumento.

Se suma a lo anterior que en la literatura especializada en *mapping* pareciera existir un vacío a la hora de abordar a los sintetizadores comerciales, no por el hecho que no se hayan estudiado en algún momento, sino que por el hecho de que una vez identificados y descritos los problemas asociados a su implementación la tendencia de los investigadores ha sido a utilizarlos como ejemplo de cómo **no** hacer las cosas y nada más. Adicionalmente, muchos autores han manifestado su descontento con aquellos controladores con forma de teclado (Wessel, Wright 2002; Hardjowirogo 2017; Miranda, Wanderley 2006), los cuales son típicos de los sintetizadores comerciales, lo que de alguna forma u otra le quita respaldo académico a este tipo de instrumentos. Así, la mayoría de los esfuerzos en el área están puestos en la creación de nuevos instrumentos con nuevas interfaces, y se ha abandonado la idea de mejorar y proponer herramientas que complementen a aquellos instrumentos que ya están acabados y en los que no se tiene acceso a la etapa de diseño, como es el caso de los sintetizadores comerciales. Es en este espacio desatendido en el que se sitúa esta propuesta.

### 3. Referentes

A continuación se presenta una recopilación de algunos trabajos realizados por terceros que de una forma u otra tienen relación con el control alternativo de sintetizadores comerciales. El objetivo de esta sección es describir aquellos trabajos, detallar sus características y elaborar en torno a los elementos que puedan resultar más relevantes para el desarrollo de esta investigación.

#### 3.1. Herramientas para el control de instrumentos electrónicos

##### 3.1.1. PHY-Series

PHY-Series (Bazzoni 2019) es un grupo de parches para *Max4Live* en el que se presenta una simulación física de un espacio tridimensional llamado PHY. Dentro de PHY existen esferas móviles que pueden interactuar entre ellas de distintas maneras. Dependiendo del parche los datos de posición, velocidad y ángulo de algunas de ellas pueden ser mapeados a los parámetros disponibles en *Ableton Live* para su control en tiempo real. El usuario puede interactuar con PHY ejerciendo perturbaciones con el movimiento del mouse, por lo que es correcto afirmar que se trataría de un sistema con una estrategia de *mapping* dos-a-varios.

El comportamiento de los sintetizadores al ser controlados por estas herramientas podría describirse como 'caótico' al mismo tiempo que 'natural'. 'Caótico' ya que a pesar de que el usuario ingrese una perturbación con cierta dirección y velocidad intencionadas, es muy difícil que al repetir esta acción los valores ingresados sean los mismos; se suma a lo anterior el hecho de que los valores de posición, velocidad y ángulo de las esferas dependen del historial de perturbaciones anteriores, lo que significa que un mismo gesto sobre el espacio PHY puede llevar a diferentes resultados sonoros cada vez que se realice. 'Natural' porque la forma en la que

varían los parámetros a lo largo del tiempo está determinada por el modelado físico de la herramienta y no necesariamente se ajusta a un comportamiento lineal. Ejemplo de lo anterior es lo que sucede con *PHY-Elastic*, donde las esferas están unidas por un elástico que de alguna manera imprime su ‘elasticidad’ en la forma en la que varían los parámetros, otorgando al timbre resultante la sensación de estiramiento/expansión y acortamiento/compresión.

Resulta interesante pensar en que esta herramienta logra unir los conceptos de ‘natural’ y ‘caótico’ con un espacio que podría ser considerado más bien ajeno a ellos, como lo es el mundo de los sintetizadores donde prima el control, la precisión y lo artificial (que no dista mucho de lo sintético). Así mismo, otro aspecto relevante es la dimensión interactiva de estas herramientas, la cual permite al usuario involucrarse con el proceso de la exploración tímbrica de una forma novedosa e incluso lúdica.



Figura 2: Las dos pestañas del dispositivo ‘Phy-Elastic’

### 3.1.2. Gradient

Gradient (pATCHES 2020) es un interpolador gráfico programado en *Max4Live* capaz de emparejarse con cualquier dispositivo/VST/*plug-in* abierto en Live. Para utilizar esta herramienta el usuario debe asignar un *preset* a cada una de las cuatro esquinas de un plano bidimensional para luego, mediante un cursor, explorar las más de tres mil combinaciones que se pueden generar al interpolarlos. Si es que el usuario encontrase un sonido de su agrado dentro de la zona de interpolación la herramienta le permite seleccionarlo como uno de los *presets* principales para afinar aún más la búsqueda de sonoridades. Si bien *Gradient* utiliza todos los parámetros disponibles en el dispositivo, también ofrece la posibilidad de ignorar aquellos que puedan estar generando una respuesta indeseada (ej. problemas de desafinación).

Bajo la premisa de que de la mezcla de buenos elementos deberíamos obtener otros buenos elementos, es que *Gradient* presenta una aproximación atractiva y sencilla para la creación de nuevas sonoridades. La simpleza de la interfaz, sumada al hecho de que el usuario debe ingresar poca información para que la herramienta funcione, vuelven de *Gradient* una buena alternativa para aquellas personas con poco conocimiento técnico y baja experticia en el uso de sintetizadores. Por lo mismo, es que su diseño podría ser atendido en caso de que se requiera implementar una herramienta con las características ya mencionadas.

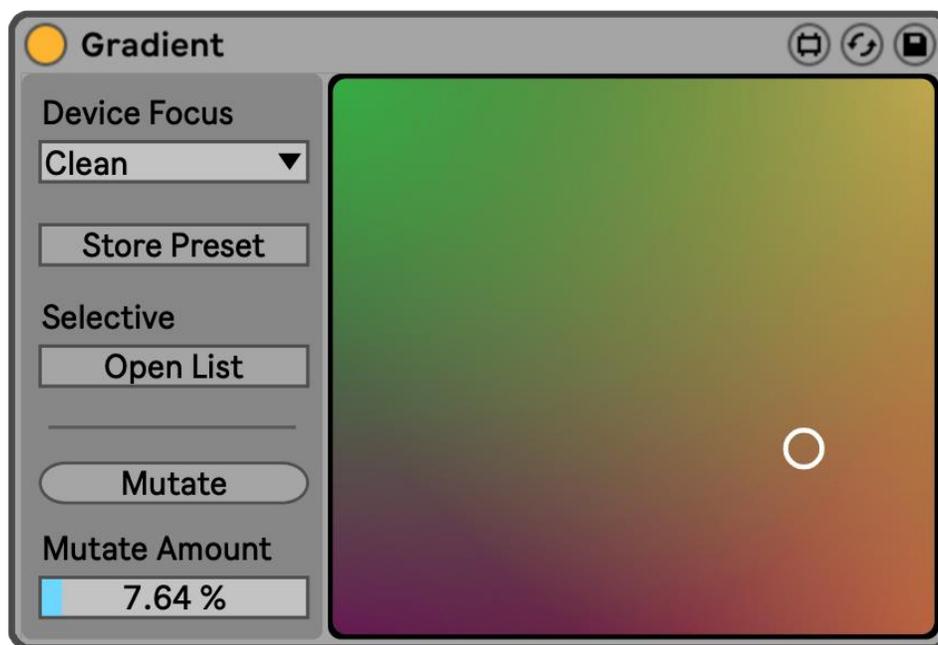


Figura 3: La interfaz del interpolador gráfico Gradient

### 3.1.3. Shaker

*Shaker* (FloSzk 2015) es un dispositivo de *Max4Live* que funciona metafóricamente como una coctelera a través de la cual se puede ‘revolver’ de forma aleatoria los valores hasta cinco parámetros establecidos por el usuario. Para cada parámetro se puede asignar de forma independiente un rango de valores posibles como también una velocidad de variación. Para accionarla el usuario sólo debe pulsar un botón llamado *Shake!* y la herramienta ejercerá un control automático y caótico sobre los parámetros escogidos.

Una de las cosas interesantes de esta herramienta es que si bien con ella los parámetros pueden comportarse de forma muy caótica con sólo apretar un botón, al mismo tiempo la herramienta permite al usuario establecer rangos detallados para delimitar el comportamiento de cada uno de ellos, es decir, en esta herramienta coexiste y se complementa la aleatoriedad con la precisión. Otro elemento a destacar

es la posibilidad de que cada parámetro tenga su propia frecuencia de cambio, lo que puede ser utilizado para diseñar patrones interesantes en el timbre.



Figura 4: La interfaz de 'Shaker'

### 3.1.4. Kaossilator

El *Kaossilator* (Korg sin fecha) es un instrumento electrónico de la compañía Korg que funciona como sintetizador y como *sampler*. Su interfaz cuenta con un panel táctil con el que el usuario debe interactuar para reproducir notas y manipular el sonido en tiempo real, eliminando así la necesidad de utilizar teclas y perillas para controlarlo. Además de producir sonidos propios, esta herramienta ha sido utilizada para controlar instrumentos electrónicos externos a través de protocolo MIDI (PsyOps Project 2013), lo que lo vuelve una opción atendible a la hora de buscar métodos alternativos para el control de sintetizadores. De realizarse esto último, resulta particularmente interesante que al utilizar un panel táctil se puede evadir la interacción que impone la tecla, por lo que se facilita la ejecución de fraseos más rápidos, con mayor rango de notas y el uso de recursos musicales como el *glissando* o el *vibrato*.



Figura 5: Kaossilator de Korg

### 3.1.5. Kaoss Pad

El *Kaoss Pad* (Korg sin fecha) es una unidad de efectos de la compañía Korg que, al igual que *Kaossilator*, tiene una interfaz implementada con un panel táctil. *Kaoss Pad* permite aplicar procesamientos en tiempo real a las señales de audio con las que se alimente, sin embargo, también puede ser utilizado como un controlador MIDI. Para esta última aplicación, además de las características que comparte con *Kaossilator*, resultan particularmente interesantes los diferentes patrones en los que se puede organizar la pantalla táctil, es decir, cómo se pueden asociar distintas regiones de la pantalla a cierto parámetro en particular. *Kaoss Pad* ofrece hasta ocho patrones diferentes dentro de los cuales destacan un plano XY para controlar dos parámetros simultáneamente; un grupo de ocho *sliders* para controlar los parámetros de forma individual; un patrón en espiral para manejar un solo parámetro con alta resolución; entre otros. El tener acceso a varios parámetros dispuestos en una pantalla táctil

permite evadir la típica interacción impuesta por los potenciómetros, lo que finalmente expande el horizonte de posibilidades sonoras.



Figura 6: Kaoss Pad de Korg

### 3.1.6. Touché

Touché (Expressive E sin fecha) es una superficie de control que se puede emparejar con instrumentos electrónicos externo vía protocolo MIDI. Esta herramienta ofrece al usuario hasta cuatro grados de libertad para el control simultáneo y en tiempo real de los parámetros del instrumento que esté emparejado. Según los fabricantes, Touché ofrece una respuesta sensible a todo tipo de gestos sobre su superficie, por lo que podría ser útil para aumentar la expresividad en la interpretación y explorar otras formas de controlar instrumentos electrónicos. Una de las grandes ventajas que tiene este controlador es que su interfaz es un objeto físico y no un programa de computador, lo que permite al usuario tener un *feedback* háptico. Además, al ofrecer una interacción alternativa a la que impone el *mouse* el usuario podría ser capaz de efectuar gestos más naturales y precisos.



Figura 7: Touché

### 3.1.7. Vector Map

Vector Map (Bastan 2022) es dispositivo de *Max4Live* que permite controlar distintos parámetros dentro de Live por medio de una simulación bidimensional de un sistema de partículas. Esta herramienta permite mapear los atributos de las partículas (posición en x, posición en y, velocidad, tamaño, distancia del centro, etc.) a los parámetros del sintetizador que el usuario estime convenientes. De esta manera, al afectar las propiedades físicas del sistema las partículas cambian su comportamiento y por ende también cambia el sonido resultante.

Una de las características interesantes de esta herramienta es que, a diferencia de otras simulaciones físicas, Vector Map propone una forma controlada de afectar los parámetros del sintetizador. Esto es debido a que las partículas del sistema operan bajo ciertas reglas que son definidas por el usuario, lo que le confiere al sistema una suerte de periodicidad similar a la de un secuenciador. La capacidad de manejar los parámetros mediante patrones controlados por una interfaz interactiva es ciertamente útil para variadas aplicaciones musicales, y es una posibilidad que sólo está dada por esta herramienta de la lista.

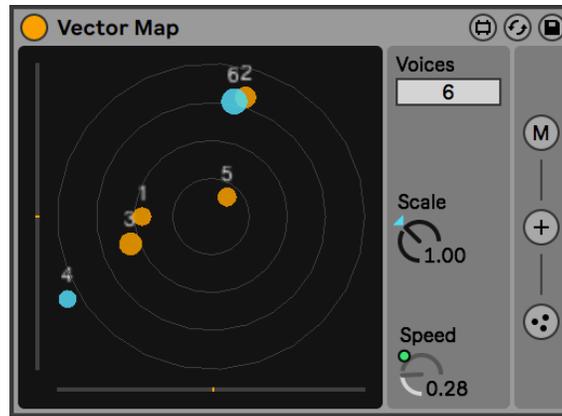


Figura 8: Interfaz de Vector Map

### 3.2. Conclusiones y reflexiones del capítulo de referentes

Se han presentado varias herramientas que debido a sus particularidades se relacionan de alguna u otra manera con la esencia de esta investigación. A continuación se profundizará en algunos aspectos interesantes de ellas.

Una de las primeras conclusiones es que a pesar de que las herramientas que componen el listado de referentes difieran entre sí en términos de aspecto, usabilidad o interacción, todas tienen algo en común: otorgan al usuario la capacidad de manejar al instrumento electrónico de formas que no serían posibles de no ser por ellas. Ya sea porque la interfaz del IME apunta a un uso más genérico, porque la interfaz es muy compleja o porque el cuerpo humano es limitado en términos de precisión, movimiento y cantidad de extremidades, existe una amplia gama de posibilidades sónicas que no pueden ser exploradas sin recurrir a herramientas que traduzcan los gestos del usuario en nuevas formas de controlar el espacio de parámetros del instrumento. Más concretamente, el hecho de que *phy-elastic* sea capaz de imprimir la sensación de 'elasticidad' en el timbre del instrumento es producto de estar implementada con ecuaciones que modelan el comportamiento de un elástico, cosa que difícilmente podría ser emulada con una mano y un potenciómetro. Así mismo, a menos que el usuario cuente con una

elevada cantidad de manos, el poder mutar todos los parámetros de un sintetizador desde un *preset* a otro no sería posible de no ser por *Gradient*. La misma reflexión puede ser llevada a cabo sucesivamente con todos los referentes de la lista.

Ahora bien, el que cada herramienta permita interactuar con el espacio de parámetro de formas tan específicas hace surgir otra conclusión: ninguno de los referentes expuestos en este capítulo pretende reemplazar por completo la interfaz original de los instrumentos que controlan, y ejemplo de ello es que solamente uno permite enviar mensajes para producir notas. Esto resulta interesante puesto que invita a pensar en este tipo de herramientas más bien como complementos a la interfaz y no como una solución absoluta a todos los problemas de diseño que puedan aparecer. El contar con un instrumento electrónico al que se pueda añadir o quitar complementos cuando se persiga un objetivo en particular puede ser una forma simple, versátil y efectiva de expandir sus capacidades sin necesidad de cambiarlo o intervenirlo de forma destructiva.

Por último, otro aspecto interesante de estas herramientas es que al estar orientadas a la exploración y creatividad se les ha puesto especial cuidado al diseño de sus interacciones. Lo anterior genera un precedente que invita a los diseñadores a pensar en herramientas que, además de ser efectivas en su cometido, sean capaces de ofrecer una interacción atractiva, creativa y que invite al usuario a seguir experimentando con ellas en diferentes contextos musicales.

## 4. Método

En el siguiente capítulo se presenta el marco metodológico de la investigación. En él se declaran entre otras cosas las características de la investigación, las técnicas a utilizadas, los métodos de análisis y los enfoques ocupados tanto para el diseño de la herramienta como para la etapa de pruebas con terceros. La formulación de este marco está diseñada de manera tal que busca ser congruente con los objetivos de investigación para así asegurar que su aplicación culmine en un resultado adecuado.

### 4.1. Características generales de la investigación

En primer lugar, es necesario señalar que esta investigación consiste en el diseño y desarrollo de una herramienta para la creación sonora/musical/artística. La naturaleza subjetiva de estas prácticas vuelve de las herramientas disponibles objetos útiles (o no) dependiendo de cómo puedan llegar a ser integrados en el proceso creativo. Por lo mismo, esta investigación no pretende presentar una herramienta que signifique una solución absoluta y definitiva para el problema planteado, sino que más bien se trate de una alternativa viable y útil que esté disponible para aquellas personas cuya práctica artística involucra instrumentos electrónicos.

A partir de lo declarado anteriormente es que se pueden perfilar las características de esta investigación y así proponer los métodos más adecuados para llevarla a cabo. A modo general, se puede establecer que en este trabajo se está realizando una investigación desde la práctica (Vargas Cordero 2009; Jensenius 2011), es decir, se busca aplicar conocimientos teóricos relacionados al *mapping* y al diseño de IME con la finalidad de producir una herramienta que responda a una problemática ya expuesta. La aplicación de dichos conocimientos debiera dar origen a la producción de otros nuevos que aparecen a medida que se prueba y desarrolla la herramienta.

Al mismo tiempo, dado que el desarrollo de un prototipo implica un proceso de constante prueba, error y depuración, esta investigación también se caracteriza por tener un fuerte componente experimental e iterativo. Lo anterior se evidencia principalmente en el hecho de que al comenzar a construir la herramienta no se tiene una total claridad de cómo será la configuración final de sus variables, por lo que es de esperar que se restructure el código numerosas veces, se prueben diferentes algoritmos para la realización de una misma tarea, se realicen pruebas rápidas para monitorear el correcto funcionamiento del código, se desechen partes del código, se añadan funcionalidades que no se tenían contempladas en un inicio, entre otras cosas. Por otro lado, es innegable la influencia que va a tener la sensibilidad del autor en la toma de decisiones referidas a la implementación de la herramienta, es decir, la dirección que se tome en cuanto al diseño estará mediada por la intuición, expectativas y bagaje de quién esté realizando la tarea. Es por esto que la experiencia personal va a tener un rol clave en los resultados finales de la investigación, y por lo mismo, para que ello no signifique un problema metodológico, es necesario explicitar que en esta investigación existe también un enfoque autoetnográfico (Ellis, Adams, Bochner 2015; López-Cano, Opazo 2014). A pesar de lo anterior, también se debe considerar que la relevancia que tiene la visión del autor en la fase de implementación será contrastada con la relevancia que cobra la experiencia de terceras personas en la fase de prueba del prototipo. Por último, corresponde declarar que todo lo que ha sido expuesto hasta el momento puede ser englobado dentro de una investigación con enfoque **cualitativo**.

## **4.2. Metodología, técnica e instrumentos.**

El desarrollo de esta investigación puede descomponerse en dos procesos claramente identificables: la fase de programación y la fase de pruebas. En la primera fase predomina la autoetnografía como método para el desarrollo de la

investigación, ya que la intuición y experiencia personal del autor son los criterios con los que se evalúa la correcta implementación de la herramienta. En la segunda fase el enfoque muta de la autoetnografía a la etnografía como método para la investigación, ya que en ese momento cobra relevancia la opinión y experiencia de las personas que prueban la herramienta.

En cuanto a la técnica de investigación, para la fase de prueba se optó por la utilización de una entrevista semiestructurada con los sujetos luego de que estos hayan probado la herramienta. Esta elección se fundamenta en el hecho de que este tipo de entrevista permite conocer en mayor profundidad la experiencia de los entrevistado ya que se permite cierto grado de flexibilidad a la hora de guiar las preguntas y se crea el espacio para reducir formalismos, para aclarar conceptos o para profundizar en aristas que no se habían considerado previamente (Díaz-Bravo et al. 2013; Stowell, Plumbey, Bryan-Kinns 2008). A pesar de la flexibilidad que presenta, esta técnica conserva la suficiente uniformidad como para que no se desvíe el foco de la conversación y se puedan recabar los datos necesarios para la investigación. Por último, cabe destacar que las entrevistas a los sujetos de prueba serán registradas en audio y video.

Las personas a entrevistar deben cumplir con: tener un instrumento musical electrónico de *hardware* o *software*, utilizar ese instrumento desde hace al menos un año e incluirlo de forma habitual en su práctica musical. Estos requisitos buscan que las personas entrevistadas tengan un conocimiento lo suficientemente alto sobre su instrumento como para estar familiarizados con su comportamiento y así poder comparar cómo cambia al ser utilizado con Octopus. La guía de entrevista puede ser revisada en el Anexo A.

### **4.3. Análisis de los datos**

Para poder hacer uso de los datos recabados con la entrevista se realizará una transcripción de ellas, sin embargo, no se traspasará la totalidad del contenido sino sólo aquel que sea relevante para la investigación. Más concretamente, se analizará por separado la respuesta para cada una de las preguntas de la entrevista y se ingresarán en una tabla las ideas principales mencionadas por el entrevistado. El paso anterior será realizado con cada uno de los sujetos de prueba. La idea es terminar con una tabla en la que cada fila esté representado un entrevistado y en cuyas columnas esté la(s) respuesta(s) a cada pregunta. Además de lo anterior, se dejará un apartado para anotar las observaciones realizadas por el investigador al tomar la prueba.

Los aspectos que serán relevantes para el análisis de las entrevistas tienen que ver con aquello que permita establecer la forma en la que el habitualmente el entrevistado se relaciona con su instrumento, aquello que permita determinar si Octopus altera la manera de relacionarse con el instrumento, aquello relacionado con la usabilidad de Octopus y aquello relacionado con los aspectos negativos de la herramienta (Jensenius 2011). A partir del análisis de los datos obtenidos es que se dará origen al capítulo de discusión en las secciones finales del documento.

## 5. Principios de diseño

En este capítulo se expone la serie de conceptos y razonamientos a través de los cuales se diseñó la estrategia de *mapping* que rige en Octopus.

### 5.1. Hacia la formulación de una propuesta de diseño

#### 5.1.1. El espacio combinatorio

Los sintetizadores son sistemas cuyo sonido resultante depende del valor que tiene cada uno de sus controles en un momento determinado, es decir, cada sonido emitido por el instrumento puede ser representado por un conjunto de valores. Esto último es de hecho la forma en la que funciona un *preset*. Ahora bien, dado que los controles del sintetizador están limitados a un rango de valores determinado por los fabricantes, es posible afirmar que estos instrumentos tienen un número **finito** de posibles sonidos resultantes. La suma de todas las combinaciones posibles que se pueden establecer entre los controles del instrumento es conocida como el espacio combinatorio (Tubb, Dixon 2014).

A partir de lo anterior, cuando se mira al instrumento electrónico desde la óptica del espacio combinatorio se puede expandir la forma en la que se piensa el concepto de *preset*. Este ya no es simplemente un trozo de memoria que sirve como atajo para llegar a un sonido, sino que se trata de una combinación única de los controles del instrumento. En cierta forma se puede afirmar que un *preset* es la unidad mínima del espacio combinatorio, y por lo tanto funciona como la unidad mínima de timbre. En consecuencia de lo anterior es que el *preset* del IME es equiparable al concepto de punto de la geometría euclidiana, y es que efectivamente, si se genera un sistema de coordenadas en el que cada eje representa a un control del sintetizador, resulta ser que cada sonido del instrumento puede ser ubicado como un punto en el espacio. A modo de ejemplo, considere un IME constituido de tres controles cuyo rango sean

los números enteros desde el 0 al 127 (con 7-bits de resolución). La representación gráfica del espacio combinatorio de este instrumento es un cubo de 128x128x128 (ver Figura 9). A partir de aquí, cada uno de los sonidos que pueden ser emitidos por el instrumento es representado por cada uno de los puntos contenidos dentro del cubo de la figura. La lógica es la misma cuando se trata de instrumentos con un mayor número de controles, y a pesar de que se dificulte la tarea de graficar el espacio combinatorio, los sonidos aún pueden ser representados por tuplas de  $n$  dimensiones.

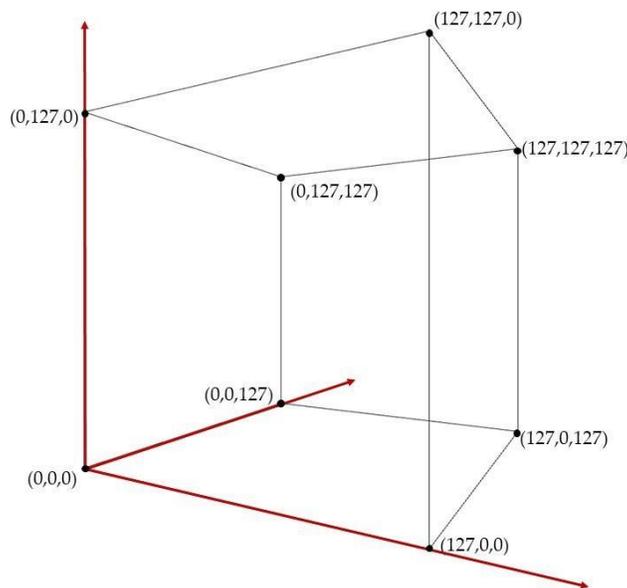


Figura 9: El espacio combinatorio de un sintetizador de tres controles. Nótese que cada control es representado por un eje.

Como fue mencionado anteriormente, el espacio combinatorio es finito, esto quiere decir que existe un número determinado de combinaciones que se pueden hacer entre los controles del instrumento. Para conocer dicho valor es necesario considerar el número de controles y la resolución de estos, es decir, la cantidad de valores que puede tomar cada uno. La fórmula para obtener el número total de combinaciones es conocida como la fórmula de “variaciones con repetición”, y está definida como  $VR_m^n = m^n$ , siendo  $m$  la resolución y  $n$  el número de controles. Resulta evidente

entonces que a mayor resolución y mayor número de controles el total de combinaciones aumenta. Para el sintetizador hipotético de tres controles el número de combinaciones es de 2.048.383. Si bien es un número alto, sigue siendo una cifra conservadora al lado de otros IME que tienen una cantidad mucho más elevada de controles. Tómese como ejemplo el Korg Volca Sample, un IME que ofrece al usuario hasta once parámetros con 7-bits de resolución para modificar una muestra de sonido previamente cargada. En este caso  $m = 128$  y  $n = 11$ , por lo que para este instrumento existen  $1,5 \cdot 10^{23}$  posibles combinaciones de parámetros. Para el caso del sintetizador Korg Minilogue ( $m = 1024$  y  $n = 25$ ) el número crece a  $1,8 \cdot 10^{75}$  posibles combinaciones. En vista de lo anterior queda claro que en ambos casos se trata de un número extremadamente alto de combinaciones.

Para dimensionar de mejor manera la cantidad de sonidos a los que se podría enfrentar el usuario en un IME cualquiera, a continuación se presenta una tabla que muestra la relación entre resolución interna y número de combinaciones para un caso general de un sistema con ocho controles.

<b>N° controles</b>	<b>Resolución (n° de valores)</b>	<b>N° de combinaciones</b>
8	10-bit (1024)	$1,2 \cdot 10^{24}$
8	9-bit (512)	$4,7 \cdot 10^{21}$
8	8-bit (256)	$1,8 \cdot 10^{19}$
8	7-bit (128)	$7,2 \cdot 10^{16}$
8	6-bit (64)	$2,8 \cdot 10^{14}$

N° controles	Resolución (n° de valores)	N° de combinaciones
8	10-bit (1024)	$1,2 \cdot 10^{24}$
8	5-bit (32)	$1,1 \cdot 10^{12}$
8	4-bit (16)	$4,3 \cdot 10^9$
8	3-bit (8)	16.777.216
8	2-bit (4)	65.536
8	1-bit (2)	256

*Tabla 1: Relación entre la resolución de cada control y el número de combinaciones que se pueden establecer.*

Como se puede ver en la Tabla 1, el número de combinaciones es del orden de los millones incluso en resoluciones tan bajas como 3-bits, bajando a decenas de miles para una resolución de 2-bit y recién llegando a un número moderado cuando se trata de una resolución de 1-bit. Considerando este comportamiento, resulta razonable buscar algún tipo de equilibrio entre las variables para así lograr proponer una herramienta que mantenga la riqueza tímbrica del IME a la vez que reduzca su espacio combinatorio a una cantidad que se adapte mejor a los recursos del usuario<sup>13</sup>.

Ya que se ha expuesto con anterioridad que el hecho de que simplificar las interfaces de los IME puede llevar a mejorar la experiencia del usuario (Goudeseune 2002), en esta investigación se busca formular una herramienta que permita al usuario relacionarse con la vastedad del espacio combinatorio, pero desde una interfaz que

---

<sup>13</sup> Concepto propuesto por Goudeseune que hace referencia al conjunto de aptitudes físicas y cognitivas de las que dispone el usuario al utilizar un IME (Ej: capacidad de atención, número de extremidades, rango y velocidad de movimiento, entre otras.).

simplifique la tarea de pensarlo, dimensionarlo y navegarlo. El desafío entonces consiste en **encontrar la manera de comprimir el espacio combinatorio en una interfaz de baja dimensionalidad.**

Como primer antecedente para la formulación de una propuesta, resulta llamativo de la tabla anterior que recién cuando se trata de una resolución binaria (1-bit) el número de combinaciones desciende a una cantidad que podría ser sencilla de dimensionar, siendo una resolución idónea desde la perspectiva de los recursos del usuario. El problema de utilizar esa resolución es que cada control solamente podría tomar dos valores, con lo que pasarían a comportarse prácticamente como botones o interruptores y ya no como *sliders*, lo que tendría como consecuencia que el instrumento pierda una gran cantidad de posibilidades tímbricas. Ahora bien, si se encontrase la manera de utilizar 1 bit de resolución a nivel de la interfaz y al mismo tiempo mantener la resolución original del instrumento a nivel de su generador de sonido, entonces se habría encontrado la solución al problema: una herramienta que simplifica al usuario la tarea de pensar y explorar el espacio combinatorio (ver Figura 10).

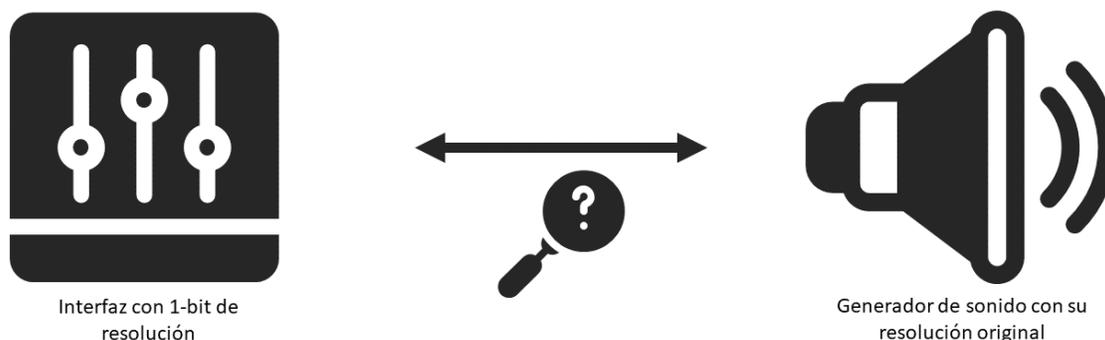


Figura 10: La interfaz de 1-bit unida al generador de sonido por medio de un método a desarrollar.

### 5.1.2. El uso de la interpolación lineal

Una forma de lograr condensar el espacio combinatorio en una interfaz que trabaje con una resolución de 1-bit es utilizando interpolación lineal<sup>14</sup>. Para comprender este razonamiento es necesario abandonar la idea de que utilizar una resolución de 1-bit convierte los controles en un simple interruptor. En lugar de eso, es conveniente pensar en que los dos estados que puede tomar cada control son en realidad un valor mínimo y un valor máximo de un rango de funcionamiento (ver Figura 11). Este cambio de perspectiva no es una decisión arbitraria y tiene que ver con la forma en la que opera la interpolación: si dicho procedimiento permite construir un recorrido entre dos puntos conocidos, perfectamente esos dos puntos podrían ser los valores extremos del rango y el recorrido construido sería el conjunto de valores que se encuentran entremedio. En otras palabras, utilizar interpolación permitiría mantener una interfaz que trabaje con sólo dos valores por control (1-bit de resolución) y al mismo tiempo permitiría que dichos controles operen con su resolución de fábrica a nivel de su generador de sonido (ver Figura 12).

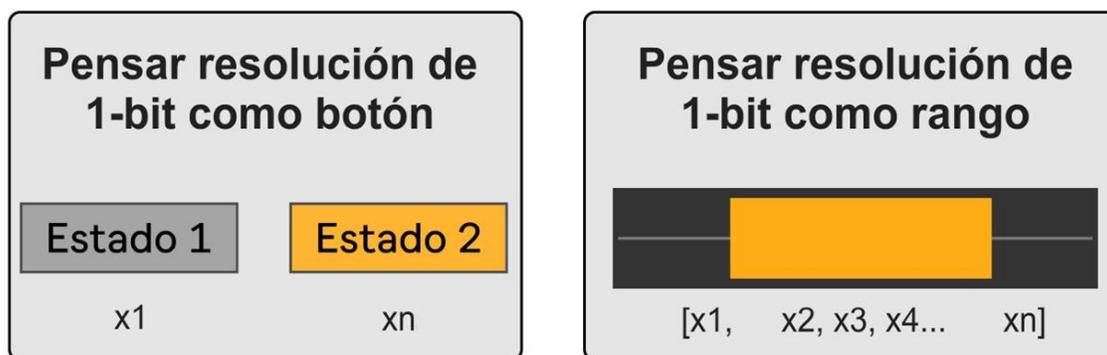


Figura 11: Las dos maneras de pensar la resolución de 1-bit. Nótese que los dos valores que puede tomar cada control son  $x_1$  y  $x_n$ .

<sup>14</sup> Recordar que este procedimiento permite la 'reconstrucción' del recorrido que existe entre dos puntos conocidos. En otras palabras, genera una recta que pasa por todos los puntos que se encuentran entre un punto A y un punto B, sólo conociendo A y B.

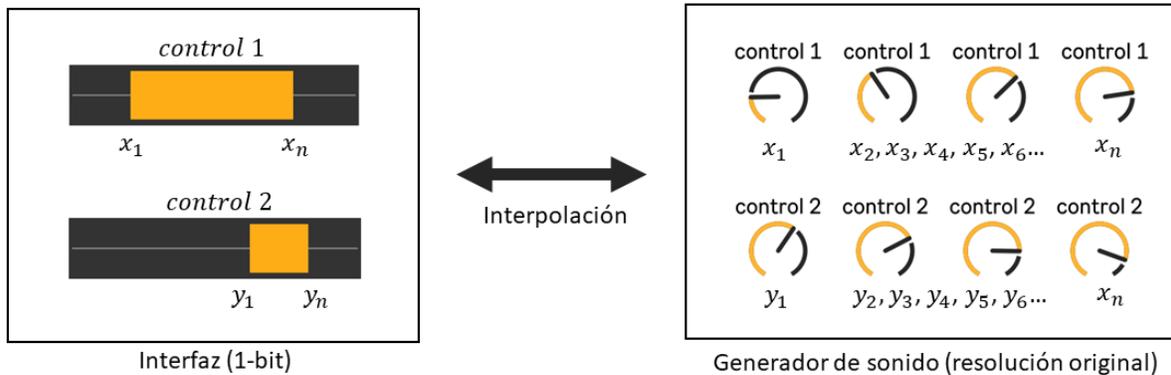


Figura 12: La interpolación en acción

Llegado este punto resulta pertinente agregar que los dos valores que puede tomar cada control a nivel de interfaz no tienen que corresponder necesariamente a los valores mínimo y máximo del rango de operación establecido por el fabricante para cada control. Como ejemplo de lo anterior, considere un IME implementado con MIDI, protocolo que establece que los controles del instrumento pueden tomar valores enteros que van desde el 0 al 127. Al reducir la resolución a 1-bit, los dos valores asociados a cada control en la interfaz no tienen que ser necesariamente 0 y 127, sino que podrán ser fácilmente escalados a cualquier rango de valores escogido por el usuario<sup>15</sup> (ver Figura 13). De esta manera, la interfaz otorga al usuario control sobre cuáles son los dos puntos entre los que se está aplicando la interpolación, lo que a nivel de generador de sonido significa que el usuario tiene libertad de elegir el rango de acción de los controles y en consecuencia tiene control sobre el timbre del instrumento. Con motivo de esta investigación, los valores mínimo y máximo establecidos por el fabricante para cada control serán llamados Valores Límite de Fábrica (VLF), y aquellos valores mínimo y máximo establecidos por el usuario serán

<sup>15</sup> Siempre y cuando dichos valores se encuentren en el rango de funcionamiento establecido por el fabricante, es decir, que sean números enteros entre el 0 y el 127.

referidos como Valores Límite de Usuario (VLU). Estos conceptos serán retomados en la sección contigua y en otras venideras.

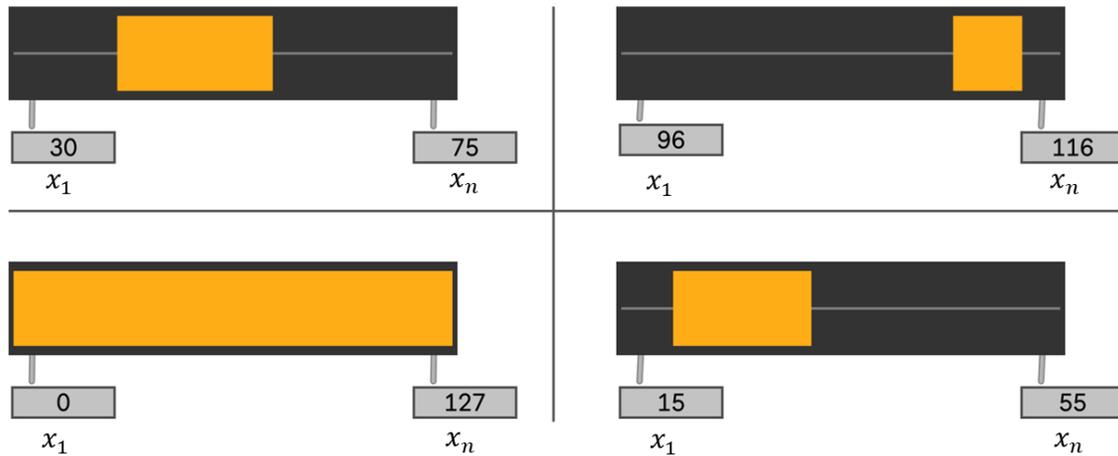


Figura 13: Los valores mínimo y máximo ( $x_1$  y  $x_n$ ) pueden ser escalados a cualquier valor numérico.

### 5.1.3. Interpolando entre presets

En la sección anterior se desarrolló la idea de que la interpolación lineal podría ser el mecanismo para encontrar el equilibrio entre una interfaz con resolución de 1-bit y un espacio combinatorio de altas dimensiones. Además de lo anterior, se expuso la forma en la que funciona la interpolación aplicada a un control **individual** de un IME. Por lo mismo, en esta sección se profundizará en la manera en la que la interpolación puede ser aplicada a nivel de *preset*, es decir, afectando simultáneamente a **múltiples** controles del instrumento.

En primer lugar, y como fue mencionado anteriormente, un *preset* es la unidad mínima del espacio combinatorio, y consiste en un listado de elementos que representan un valor específico para cada uno de los controles del instrumento. Cuando sucede una interpolación entre un *preset* A y un *preset* B, lo que ocurre es que se genera una interpolación entre el valor que tiene cada uno de los elementos contenidos en el *preset* A y su correspondiente del *preset* B (ver Figura 14). Esto significa que si se aplica interpolación para encontrar el punto medio entre los *preset*

A y B, el valor que tendrá cada uno de los elementos dependerá de su valor en A y de su valor en B. El hecho de que este procedimiento permite trabajar con un rango de valores independiente para cada elemento del listado vuelve de la interpolación una herramienta idónea para el establecimiento de VLU.

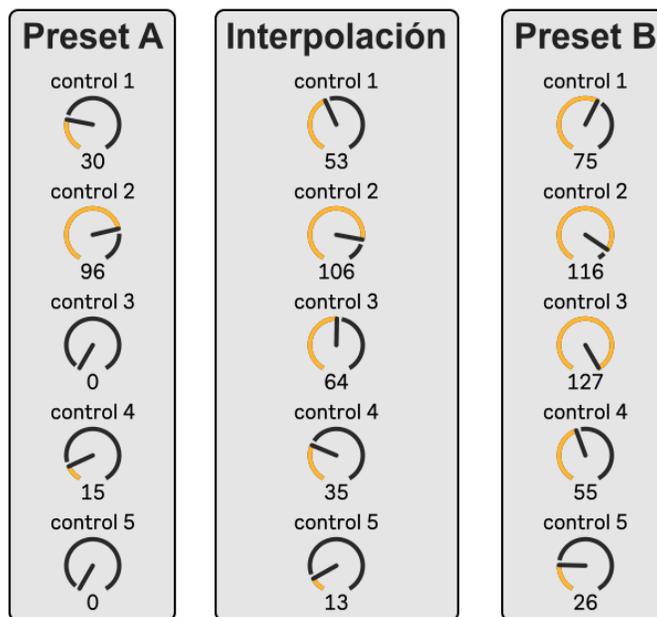


Figura 14: El punto medio entre un preset A y un preset B

También es importante señalar que si el valor de un control se repite tanto en A como en B, no hay ningún recorrido que construir entre esos dos puntos -ya que se trata del mismo punto-, por lo que no se efectuaría interpolación alguna y su valor permanece constante. A partir de lo anterior, interpolar entre un *preset* A y un *preset* B puede afectar a uno, a algunos o a todos los controles del instrumento. La interpolación afectaría a un solo control si es que la única diferencia entre ambos *preset* es el valor de un solo elemento; afectaría a algunos controles si ambos *preset* presentan diferencias entre varios -más no todos- los elementos; y afectaría a todos los controles en el caso de que todos los elementos del listado tengan diferentes valores en A y en B.

Finalmente, señalar que al interpolar entre dos *presets* se produce una modificación continua del valor de cada elemento del listado, es decir, no existen discontinuidades entre el punto de partida y el punto de llegada. Esto se asemeja a lo que sucede cuando se operan manualmente los potenciómetros de un IME, los cuales no permiten llegar de un valor a otro sin pasar por aquellos que se encuentran entremedio. Por lo mismo, se podría decir que cuando el usuario opera los controles del IME con sus manos está ‘interpolando’ entre un *preset* y otro. La diferencia entre la interpolación realizada por el usuario y aquella realizada por el computador es que, debido a sus recursos limitados, el primero sólo puede modificar hasta dos controles simultáneamente<sup>16</sup>, mientras que el segundo puede manejar sin problemas múltiples controles al mismo tiempo, como si se tratase de un ser con **múltiples extremidades**.

A partir de todo lo mencionado anteriormente, es que la interpolación de *preset* parece ser el mecanismo de funcionamiento adecuado para la herramienta a diseñar. Ahora bien, la pregunta es ¿cómo seleccionar y gestionar el conjunto de combinaciones entre las cuales sería interesante aplicar la interpolación? considerando sobre todo que el objetivo principal de la herramienta es la exploración de la vastedad del espacio combinatorio.

#### 5.1.4. Las “combinaciones vértice” y la creación de sub-espacios combinatorios.

Dentro del espacio combinatorio existen algunas combinaciones especiales que podrían ser la respuesta a la pregunta con la que finaliza el apartado anterior. Dichas combinaciones serán denominadas **combinaciones vértice**, y son todas aquellas que se pueden formar utilizando los valores límite de fábrica (VLF). Para aclarar lo

---

<sup>16</sup> Asumiendo que deja de utilizar sus dos manos para tocar notas.

anterior, tómesese como ejemplo un sintetizador de tres controles<sup>17</sup> cuyos VLF son 0 y 127. Al tratarse de tres controles se pueden formar hasta ocho diferentes combinaciones utilizando dichos valores (Tabla 2). Al graficar dichas combinaciones se vuelve evidente porqué resulta adecuado llamarlas “vértices” (ver Figura 15).

<b>Nº de combinación</b>	<b>Combinaciones vértice a nivel conceptual</b>	<b>Combinaciones vértice escaladas a los VLF</b>
1	[min, min, min]	[0, 0, 0]
2	[min, min, max]	[0, 0, 127]
3	[min, max, min]	[0, 127, 0]
4	[min, max, max]	[0, 127, 127]
5	[max, min, min]	[127, 0, 0]
6	[max, min, max]	[127, 0, 127]
7	[max, max, min]	[127, 127, 0]
8	[max, max, max]	[127, 127, 127]

*Tabla 2: las combinaciones vértice del sintetizador de tres controles.*

---

<sup>17</sup> Cabe destacar que la esencia de las combinaciones vértice se mantiene igual, aun cuando se trate de instrumentos con un mayor número de controles.

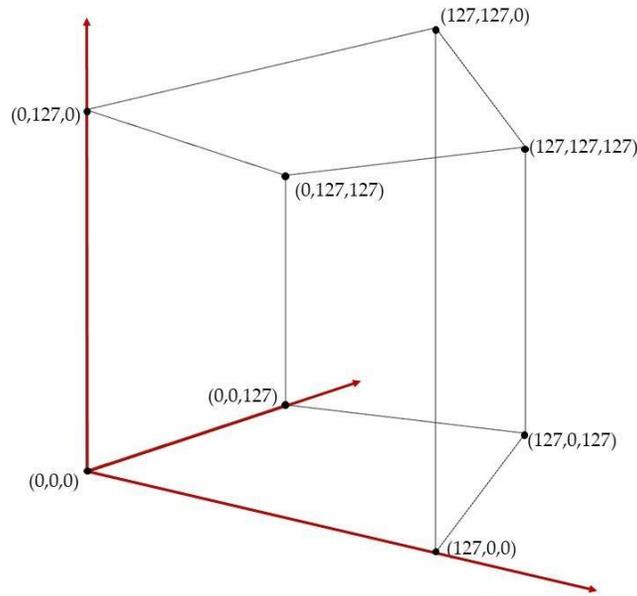


Figura 15: Las combinaciones vértice del espacio combinatorio del sintetizador de tres controles. Nótese que cada control es representado por un eje y que en cada eje siempre hay dos valores: el mínimo y el máximo del rango de funcionamiento del control.

Es importante destacar de ese tipo de combinaciones que, cuando los elementos que las constituyen son reducidos a nivel de concepto, es decir, cuando se asume que pueden ser o el mínimo o el máximo de los VLF (ver segunda columna de la Tabla 2), entonces sucede que las combinaciones vértice de un IME son siempre las mismas, independiente del valor numérico que pueda tomar cada elemento en su estado mínimo o en su estado máximo. Pensar en las combinaciones vértice de esta manera resulta útil puesto que se pueden trabajar desde un punto de vista más general y no dependen del valor numérico que tengan asociado, por lo mismo, esta será la óptica que se va a ocupar en esta investigación.

Por otro lado, una característica que vuelve de las combinaciones vértice una buena alternativa para implementar en la herramienta es que, debido a que son las que demarcan los límites del espacio combinatorio, éstas son **distantes** entre sí. Es por lo anterior que si se aplicase interpolación entre dos combinaciones vértice cualquiera

se podría recorrer un extenso grupo de combinaciones, y por ende, un extenso conjunto de timbres, lo que resulta ideal para una herramienta que persigue ese propósito. Ahora bien, un aspecto negativo de esto es que al tratarse de un conjunto tan grande de combinaciones, se podría dificultar la tarea de encontrar timbres significativos para el usuario. Por otro lado, otra característica problemática que se debe considerar al trabajar con estas combinaciones es que, al tratarse de configuraciones en las que los controles están o completamente al mínimo o completamente al máximo, su sonido puede no ser del todo “útil” o del todo “musical”<sup>18</sup>.

Ya que trabajar directamente con estas combinaciones podría generar ciertas dificultades, en este trabajo se utilizará la esencia del concepto de combinación vértice y se modificará para poder formular una propuesta novedosa de exploración tímbrica. Para ello, se partirá de la premisa de que el espacio combinatorio es extremadamente vasto como para que sea conveniente intentar trabajar con toda su extensión. Por lo mismo, lo que aquí se plantea consiste en permitir al usuario la creación de un sub-espacio combinatorio, el cual esté contenido dentro del original, que sea más reducido que él y que permita acotar la búsqueda de timbres a un rango de preferencia. La forma de crear dicho sub-espacio es con el establecimiento de nuevas combinaciones vértice, esta vez formadas utilizando los VLU en vez de los VLF. Estos nuevos vértices ingresados por el usuario funcionan como los límites del

---

<sup>18</sup> Imagine las consecuencias de esto cuando se trata de controles como la amplitud de un oscilador, la intensidad de un LFO, la resonancia de un filtro o su frecuencia de corte. Habría combinaciones vértice en las que el resultado sonoro podría ser aceptable, mientras que habría otras en las que directamente podría no haber sonido (con la frecuencia de corte al mínimo, por ejemplo). Es pertinente mencionar que el sonido asociado a estas combinaciones depende por completo de la forma en la que está diseñado el instrumento.

sub-espacio y restringen la exploración de timbres a una cantidad “más manejable” de combinaciones (ver Figura 16).

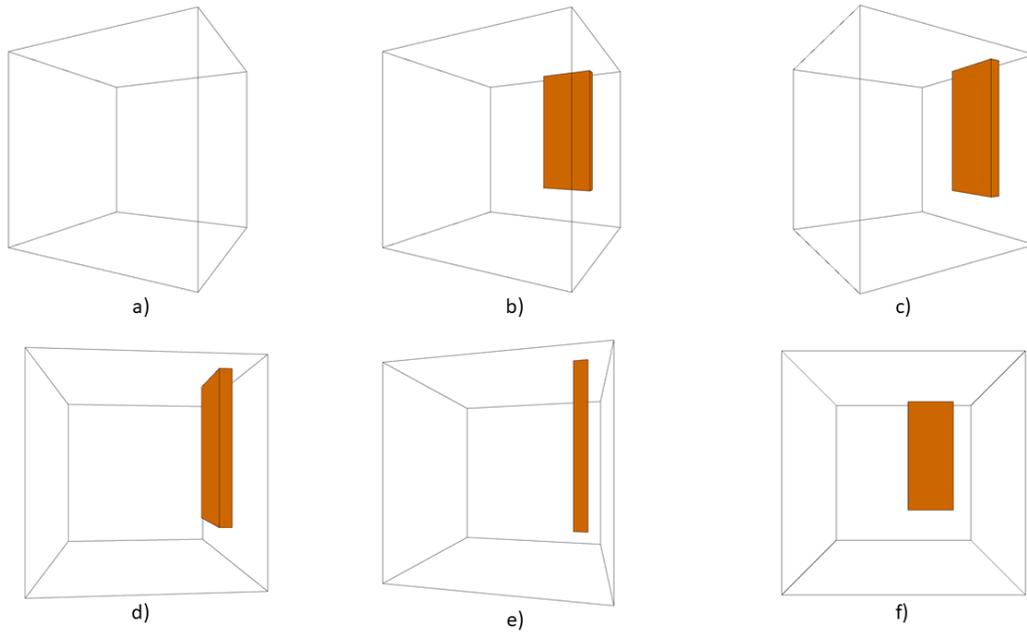


Figura 16: Un sub-espacio combinatorio (en naranja) contenido dentro del espacio combinatorio de un IME con 3 controles.

## 5.2. La propuesta de diseño

Esta subsección tiene como objetivo exponer la manera en la que los conceptos e ideas presentados anteriormente pueden relacionarse entre sí para converger en una herramienta que permita la exploración creativa de timbres. Además, se explicitan algunas determinaciones necesarias para el desarrollo del prototipo.

En primer lugar, la esencia de esta herramienta consiste en permitir al usuario explorar porciones reducidas del espacio combinatorio de su instrumento electrónico, es decir, **no** se pretende que el usuario tenga acceso despejado y simultáneo a todos y cada uno de los sonidos de su instrumento. Junto con lo anterior, uno de los objetivos perseguidos en el diseño de esta herramienta es ofrecer al usuario la posibilidad de relacionarse con la complejidad del espacio

combinatorio pero desde una interfaz que le simplifique la tarea de pensarlo, dimensionarlo y recorrerlo.

En vista de lo anterior, se tomó la determinación de que la herramienta pueda manejar hasta ocho controles del IME al mismo tiempo<sup>19</sup>. Así mismo, se estableció que Octopus permitirá al usuario relacionarse con el espacio combinatorio desde la creación de un sub-espacio de combinaciones. Más concretamente, la interfaz de la herramienta requiere que el usuario establezca un rango de operación para cada uno de los ocho controles, es decir, un valor mínimo y un valor máximo (VLU). De esta forma, más allá de estar decidiendo un par de números, lo que está haciendo el usuario es definir exactamente cuáles son las combinaciones vértice que definen los límites del sub-espacio combinatorio a explorar. Así, a la hora de aplicar la interpolación entre las CVU, la herramienta permitirá explorar aquellos sonidos que se encuentren dentro de los límites que el mismo usuario estableció. Dado que la interfaz de la herramienta trabaja virtualmente con ocho controles con 1-bit de resolución, el número total de combinaciones vértice de usuario que se pueden establecer es de 256<sup>20</sup>.

Por otro lado, el uso de la interpolación tiene como finalidad poder construir distintos recorridos entre los diferentes pares de CVU. De esta manera, la herramienta posibilita al usuario explorar (a nivel de generador de sonido) aquellos timbres contenidos dentro del sub-espacio combinatorio como si nunca se hubiese reducido la resolución del IME, pero desde una interfaz que se adapta a sus recursos

---

<sup>19</sup> Esto significa que los usuarios de instrumentos que tengan mayor cantidad de controles deberán priorizar aquellos con los que deseen trabajar. Cabe destacar que ello no inhabilita ni desactiva los controles que se dejen fuera del dominio de la herramienta, sino que simplemente éstos deberán ser operados desde la interfaz original del instrumento y no desde Octopus.

<sup>20</sup> Recordar que a nivel conceptual las combinaciones vértice son siempre las mismas, independiente del valor numérico de los VLF o de los VLU.

al sólo necesitar que éste se relacione con las combinaciones vértice y sus valores numéricos.

### **5.2.1. Características deseables de la herramienta**

En esta sección se enlistan y resumen las principales características y prestaciones que debería tener la herramienta en base a los conceptos discutidos anteriormente.

#### **5.2.1.1. Control de instrumentos electrónicos**

- La herramienta es capaz de controlar instrumentos electrónicos de forma **externa**.
- La herramienta utiliza el protocolo MIDI para la comunicación con el instrumento (es necesario que este cuente con entrada y salida MIDI).
- La herramienta puede operar tanto con instrumentos electrónicos de hardware como de software.
- La herramienta puede funcionar independientemente del algoritmo de producción de sonido del IME (síntesis FM, síntesis granular, síntesis sustractiva, etcétera).

#### **5.2.1.2. Manejo y gestión de controles**

- La herramienta es capaz de manejar hasta 8 controles del instrumento de forma simultánea.
- Se le permite al usuario especificar cuáles son los controles que desea manejar con la herramienta. Además, se le permite al usuario cambiar los controles mapeados si así lo desea.
- El usuario puede seleccionar los valores mínimo y máximo para cada control de forma independiente. Al trabajar con protocolo MIDI estos valores son números enteros entre el 0 y el 127. Por último, los valores pueden reajustarse en cualquier momento conforme el usuario estime conveniente.

### 5.2.1.3. Gestión de las combinaciones.

- La herramienta maneja un listado con cada una de las 256 combinaciones vértice normalizadas.
- Cada combinación es un listado compuesto de ocho elementos que representan un valor específico para cada control. Los valores son un 0 para representar el estado “mínimo” y un 1 para representar el estado “máximo”.
- El valor de cada elemento de la combinación es escalado de forma lineal a los valores mínimo y máximo ingresados por el usuario para cada control.

### 5.2.1.4. Interpolación y trayectorias

- La herramienta permite aplicar interpolación lineal por medio de un interpolador gráfico.
- El interpolador gráfico dispone de un plano bidimensional en el que las CVU son representadas a través de círculos de diferente tamaño y posición. La interpolación ocurre en aquellas zonas en las que los dos o más círculos se intersectan.
- La herramienta ofrece al usuario diferentes maneras de distribuir las CVU en el plano. Esto incluye variaciones en el número de círculos, variaciones en su tamaño y posición o variaciones en la manera en la que se intersectan.
- La herramienta ofrece al usuario diferentes maneras de asignar (o reasignar) las combinaciones a los círculos distribuidos en el plano.
- La herramienta permite al usuario guardar sonidos que encuentre al usar el interpolador.
- La herramienta ofrece al usuario diferentes opciones para recorrer el plano bidimensional: de forma manual o automatizada de diferentes maneras.

## 6. Desarrollo e implementación del primer prototipo

### 6.1. Herramientas a utilizar

Para el desarrollo del primer prototipo se decidió utilizar el ambiente de programación Max<sup>21</sup> debido a la facilidad con la que se puede trabajar audio y MIDI. Por motivos prácticos, a la hora de trabajar con sintetizadores de *software* también se utilizó la versión de Max incorporada dentro de Live<sup>22</sup>, más conocida como *Max for live*<sup>23</sup>. Junto con lo anterior, durante el desarrollo de la herramienta se contó con el sintetizador Korg Minilogue<sup>24</sup> como instrumento de prueba para todas las etapas de programación y testeó. Además, se contó con el sampler Korg Volca Sample<sup>25</sup> como instrumento de respaldo para pruebas ocasionales. Para trabajar con los instrumentos de *hardware* se utilizó en todo momento el protocolo MIDI, y para trabajar con los instrumentos de *software* se utilizó -además de MIDI- el protocolo OSC para comunicar Octopus con *Max for live*.

---

<sup>21</sup> <https://cycling74.com/products/max> (What is Max? | Cycling '74 sin fecha)

<sup>22</sup> <https://www.ableton.com/en/live/what-is-live/> (Ableton Live | Ableton sin fecha)

<sup>23</sup> <https://cycling74.com/products/maxforlive> (Max for Live | Cycling '74 sin fecha)

<sup>24</sup> <https://www.korg.com/cl/products/synthesizers/minilogue/> (minilogue - POLYPHONIC ANALOGUE SYNTHESIZER | KORG (Chile) sin fecha)

<sup>25</sup> [https://www.korg.com/cl/products/dj/volca\\_sample/index.php](https://www.korg.com/cl/products/dj/volca_sample/index.php) (volca sample - DIGITAL SAMPLE SEQUENCER | KORG (Chile) sin fecha)

## 6.2. Implementación de Octopus

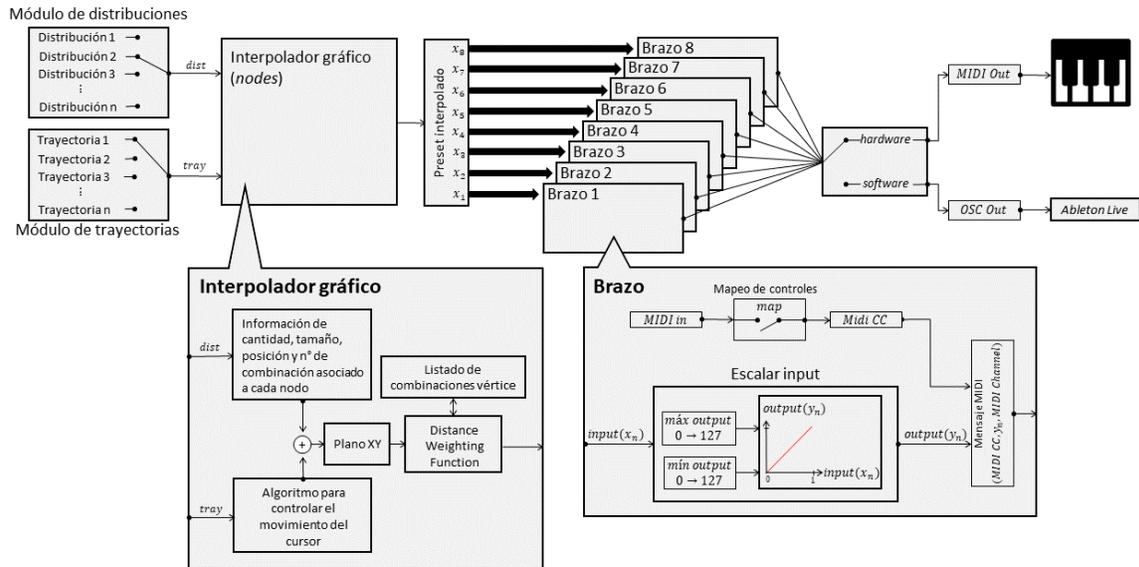


Figura 17: Diagrama en bloque del primer prototipo de Octopus (acabado).

### 6.2.1. Comunicación herramienta-instrumento:

Como primera tarea para el desarrollo del prototipo se volvió necesario establecer la forma de comunicar Octopus con el instrumento. Para ello se optó por la utilización de mensajes MIDI, particularmente de aquellos conocidos como Control Change Messages, los cuales permiten el traspaso de información relacionada con los controles del IME. Así, la herramienta es capaz de controlar de forma externa e individualizada el valor de cada uno de los controles del instrumento, lo que resulta ideal dado el fin último de esta investigación. Esta primera parte de la implementación es fundamental para etapas posteriores, ya que es acá donde se establece el canal por el cual la herramienta ejercerá su dominio sobre el IME.

A partir de lo anterior se implementó un algoritmo que permite al usuario indicar a la herramienta cuáles son exactamente los controles del instrumento con los que

desea trabajar, además de poder elegir a cuál de los ocho brazos<sup>26</sup> de la herramienta queda asignado cada control. Para ello se tomaron en consideración dos escenarios probables: que el usuario cuente con un instrumento de *hardware* o que el usuario cuente con un instrumento de *software*. Para el primer caso, cada brazo contará con un botón llamado *Map*, el cual una vez clicado estará “esperando” a que el usuario mueva el control que desee poner bajo el dominio de la herramienta<sup>27</sup>. Una vez detectado el movimiento, dicho control quedará asignado al brazo seleccionado en primer lugar. Para el segundo caso, se decidió utilizar Octopus en conjunto de un parche de *Max for live* que funciona como una ‘antena’ dentro de Live, y cuya comunicación está mediada por el protocolo OSC. Entonces, para lograr conectar Octopus con el sintetizador virtual es necesario que este último sea cargado dentro de una sesión de Live en la que previamente exista una instancia de la antena. Dicha antena cuenta con un botón *Map* para cada brazo, el cual una vez pulsado espera a que el usuario cliquee sobre el control del instrumento virtual que desea asignar. Tanto para *hardware* como para *software* los mapeos pueden ser revertidos y el usuario podrá reasignar a cualquiera de los brazos cualquier otro control si así lo desea.

En el siguiente [video demostrativo](#) se puede observar más concretamente la forma en la que operan los algoritmos descritos con anterioridad. Además, se provee de una demostración para un instrumento de *hardware* y uno de *software*.

---

<sup>26</sup> Se llamó *brazo* al módulo encargado de manejar un control cualquiera del instrumento. Ya que la herramienta es capaz de trabajar hasta con ocho controles, esta cuenta con ocho brazos.

<sup>27</sup> Es importante señalar que para que esto funcione el instrumento de *hardware* debe contar con entrada y con salida MIDI. De faltar esta última la herramienta no tiene cómo percibir el movimiento de los controles.

## 6.2.2. Selección y manipulación de rangos:

Una vez lista la etapa anterior, corresponde implementar el mecanismo que permite al usuario establecer los límites del espacio combinatorio que desea explorar. Más concretamente, se debe implementar un algoritmo para definir el rango de valores bajo el cual va a operar cada brazo de la herramienta. Para ello se utilizó el objeto *rslider*<sup>28</sup> presente *Max for Live*, el cual facilita al usuario la selección/modificación rápida e interactiva de un rango de valores. Cada brazo de la herramienta cuenta con su propio *rslider*, por lo que es posible trabajar con un rango diferente para cada uno. Dicho objeto es accionado vía mouse y cuenta con una serie de comandos que permiten manipular el rango de valores de diferentes formas.

En este [video demostrativo](#) se muestra el funcionamiento de *rslider* y del algoritmo descrito anteriormente.

### 6.2.2.1. Creación y escalado de las combinaciones vértice:

La esencia de esta herramienta se sostiene en la idea de las combinaciones vértice y en la posibilidad de crearlas y controlarlas, sin embargo, hasta acá la implementación del prototipo sólo permite al usuario seleccionar los controles que desea utilizar y elegir un rango de funcionamiento para cada uno, por lo que todavía hace falta desarrollar el mecanismo para que los datos ya ingresados den origen efectivamente a las combinaciones vértice. Para ello, es necesario recordar que dado un IME con un número determinado de controles, el conjunto de combinaciones vértice (en su forma abstracta) será siempre el mismo, independiente del valor

---

<sup>28</sup> Más detalles de su funcionamiento en <https://docs.cycling74.com/max7/refpages/rslider> (*rslider Object Reference - Max 7 Documentation* sin fecha).

numérico que pueda tomar cada control. Para esta herramienta de ocho controles existen 256 de estas combinaciones.

Ahora bien, para construir las combinaciones vértice que delimitan el sub-espacio a explorar se deben tomar en cuenta los ocho rangos ingresados por el usuario<sup>29</sup> y las 256 combinaciones vértice en su forma abstracta. Si se toma una combinación cualquiera del listado, se puede observar que se trata de una tupla compuesta por ocho elementos cuyos dos estados posibles son “mínimo” o “máximo”, y que cada elemento de la tupla está representando el estado del control que le corresponde<sup>30</sup>. A partir de acá, para construir las combinaciones es necesario escalar los elementos de la tupla a los valores mínimo o máximo del rango ingresado por el usuario, según corresponda. Este procedimiento se debe realizar para cada una de las 256 combinaciones y se actualiza cada vez que el usuario modifica alguno de los rangos. Como resultado de lo anterior se debería obtener un conjunto de tuplas cuyos ocho elementos representan un valor MIDI para el control que tengan asociado.

Para implementar el procedimiento ya descrito se optó por normalizar el listado de combinaciones vértice en su forma abstracta. Para ello, el estado “mínimo” pasó a ser representado por un 0 y el estado “máximo” pasó a ser representado por un 1. Se creó un archivo de texto que contiene el listado de todas las combinaciones normalizadas y se desarrolló un algoritmo que permite acceder a él desde Max. Para cruzar las combinaciones normalizadas con los rangos ingresados por el usuario se

---

<sup>29</sup> Es decir, ocho pares de números. Cada par está asociado a un brazo de la herramienta y está compuesto por un número que representa el límite inferior y otro que representa el límite superior del rango de operación de cada control.

<sup>30</sup> Recordar que al primer elemento de la tupla le corresponde el control asociado al primer brazo de la herramienta, y así sucesivamente.

utilizó el objeto *scale*<sup>31</sup>, el cual permite escalar un rango de entrada a un rango de salida. En este caso el rango de entrada es [0,1] y el rango de salida es aquel ingresado por el usuario. Por último, cabe destacar que al igual que con el objeto *rslider*, cada brazo de la herramienta cuenta con su propia instancia del objeto *scale*.

En el siguiente [video demostrativo](#) se puede observar el algoritmo que almacena y gestiona el listado de las combinaciones ya normalizadas. Además, se puede ver la manera en la que opera el objeto *scale* a la hora de escalar las combinaciones normalizadas a los rangos establecidos por el usuario.

#### **6.2.2.2. El sonido de las combinaciones vértice:**

Llegado este punto, el usuario ya es capaz de indicar los controles que desea manejar, el rango de operación para cada uno y además la herramienta transforma esos datos en paquetes de información MIDI, por lo que ya es posible enviar mensajes de control hacia el instrumento. Dado que las combinaciones vértices son representaciones numéricas de un sonido, es posible entonces escuchar cómo suenan los puntos límite del sub-espacio combinatorio escogido por el usuario.

El siguiente [video demostrativo](#) funciona como un resumen en el que se puede observar todo el proceso necesario para poner en marcha la herramienta, desde la selección del sonido, la selección y mapeo de controles, el proceso de selección de rangos hasta finalmente llegar a escuchar el sonido de cada una de las 256 combinaciones vértice. Cabe recordar que hasta este momento aún no se aplica ningún tipo de interpolación.

---

<sup>31</sup> Más información sobre su funcionamiento en <https://docs.cycling74.com/max8/refpages/scale> (*scale Reference - Max 8 Documentation* sin fecha).

### 6.2.3. Interpolación:

En esta etapa se desarrolló un método para interpolar entre dos o más combinaciones vértice para así poder explorar los timbres que se encuentran entremedio. Para ello se utilizó el objeto *nodes*<sup>32</sup>, el cual es un sistema gráfico de interpolación incluido dentro de Max y cuyo *output* está determinado por la posición de un cursor dentro de un plano bidimensional. Dicho objeto permite posicionar sobre el plano un conjunto de hasta 64 círculos (o nodos) de diferente tamaño y posición, en el que cada uno representa una combinación vértice en particular. En aquellas zonas en las que dos o más círculos se intersectan es donde sucede la interpolación, la cual se calcula utilizando una función basada en la distancia existente entre la posición del cursor y el centro de cada círculo. La influencia que tendrá cada círculo en el resultado final es determinada por un número decimal dentro del rango [0,1]. Para ver el funcionamiento del objeto *nodes* en el contexto de Octopus dirigirse al [video demostrativo](#).

En cuanto a la relación existente entre *nodes* y las combinaciones vértice, cabe destacar que para poder conocer la combinación vértice asociada a cada nodo basta con observar el número identificador que tiene cada uno en el centro (ver Figura 18). Por otro lado, es necesario hacer notar que dado que este objeto solamente permite un máximo de 64 círculos en el plano, no pueden coexistir las 256 combinaciones de forma simultánea. Para sortear esta dificultad se optó por el uso de la aleatorización, es decir, cada vez que se carga una instancia de *nodes* la combinación vértice asociada a cada nodo sería automáticamente actualizada por un valor aleatorio entre 1 y 256, esto con el cuidado de que no haya repeticiones. La finalidad de volver aleatorio el listado de combinaciones vértice presentes en el plano es poder presentar al usuario

---

<sup>32</sup> Más detalles de este objeto en <https://docs.cycling74.com/max7/refpages/nodes> (*nodes Object Reference - Max 7 Documentation* sin fecha).

intersecciones novedosas de forma constante. En la sección 6.2.4.4 se detallan algunas funciones de Octopus relacionadas a la administración y modificación del listado de combinaciones vértice presentes en el plano.

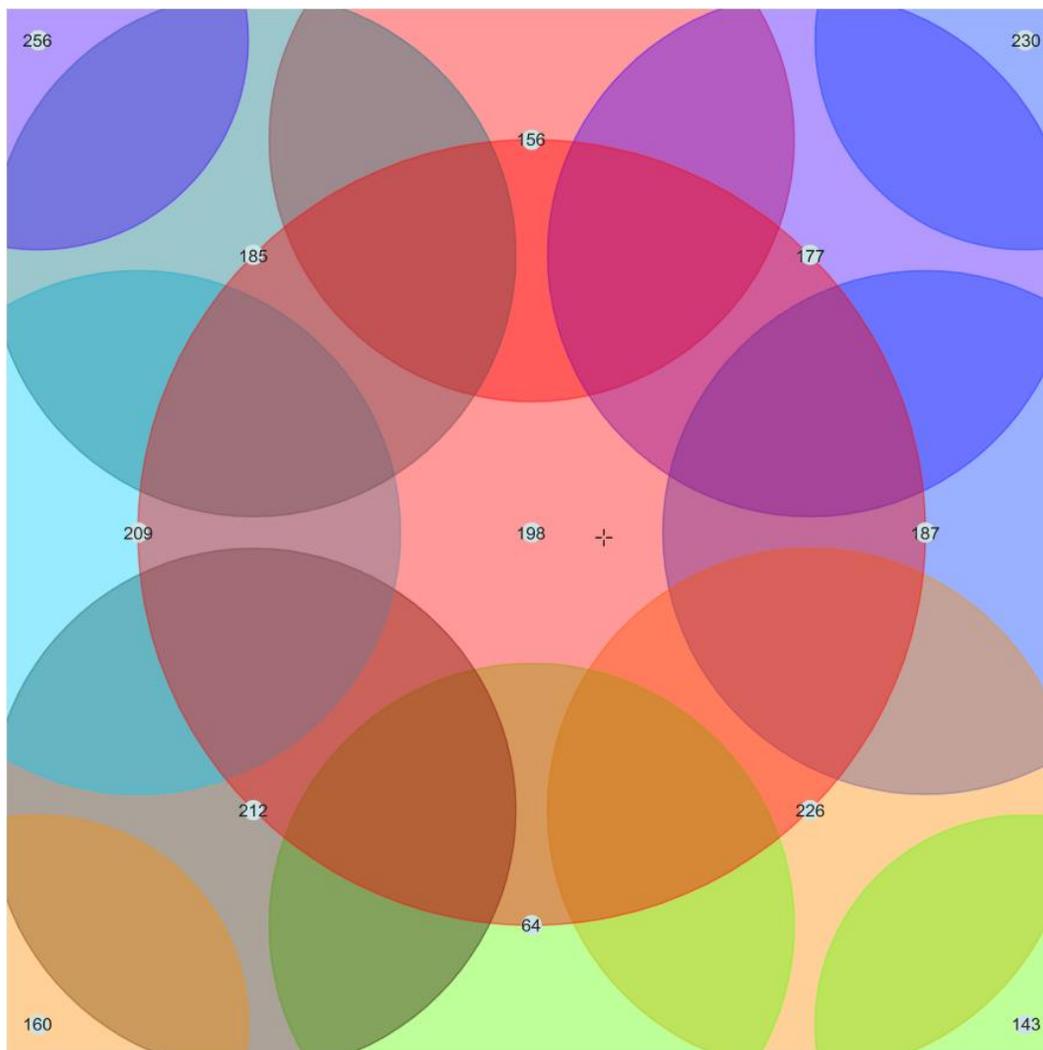


Figura 18: Vista del objeto nodes dentro de Octopus. Nótese que están siendo representadas las combinaciones 256, 156, 230, 185, 177, 209, 198, 187, 212, 64, 226, 160 y 143.

### 6.2.3.1. Almacenamiento de combinaciones de usuario

Dado que la esencia de esta herramienta reside en la idea de la exploración y descubrimiento de timbres, es altamente probable que el usuario encuentre sonidos de su interés al posicionar el cursor en las áreas de intersección entre nodos. Por lo

mismo, se decidió incorporar en Octopus una funcionalidad que permite al usuario guardar hasta 16 de esos sonidos. Dichos sonidos quedan almacenados en la memoria de la herramienta en forma de una tupla de ocho elementos cuyos valores corresponden a los calculados por el algoritmo de interpolación (ver Figura 19).

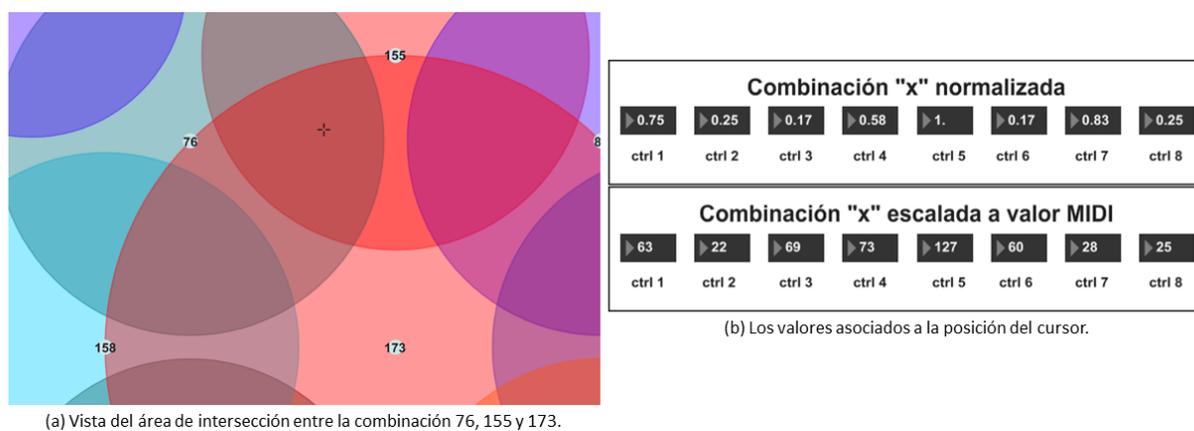


Figura 19: En (a) se puede ver el cursor posicionado en el área de intersección de tres nodos. En (b) se muestran los valores asociados a cada uno de los ocho controles del instrumento luego de haber aplicado el algoritmo de interpolación entre los tres nodos. Al guardar un sonido de usuario este quedará almacenado en su forma normalizada.

Para poder guardar un sonido en la biblioteca de usuario es necesario utilizar los comandos implementados en el módulo de trayectorias (sección 6.2.5). Dicho módulo debe configurarse en modo manual (ver sección 6.2.5.1), luego el usuario deberá seleccionar un punto del plano, presionar el botón “¿guardar sonido actual?” y posteriormente seleccionar una de las 16 ranuras de memoria en la que desea guardar el sonido (ver Figura 20). Las combinaciones almacenadas por el usuario están numeradas del -1 al -16, utilizándose valores negativos principalmente para diferenciarlos de las combinaciones vértice 1 a la 16.

Una demostración del algoritmo para guardar sonidos de usuario puede ser vista en este [video demostrativo](#).

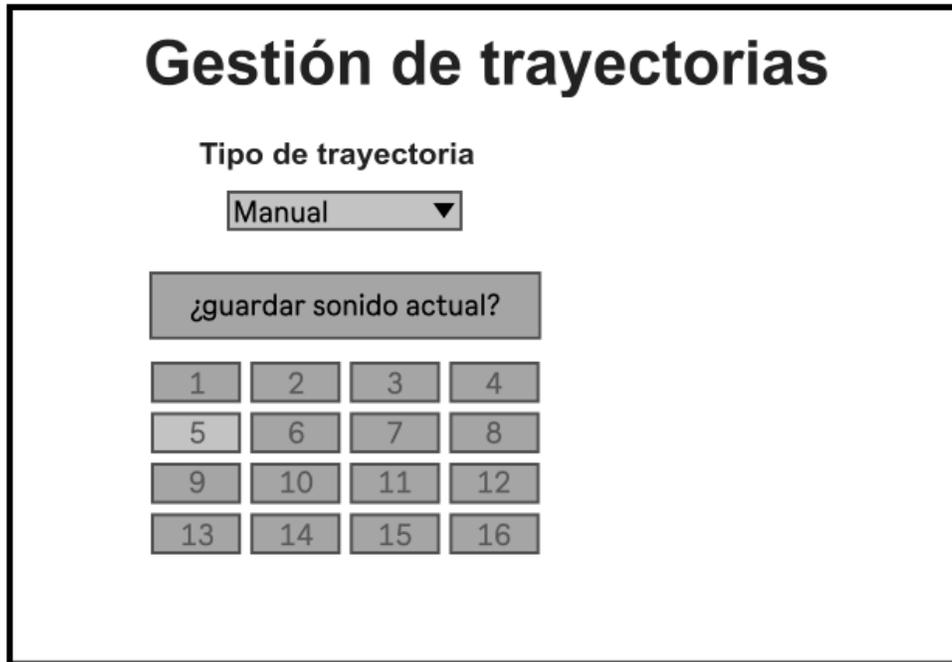


Figura 20: Vista del módulo de trayectorias en modo manual. Los comandos que aparecen en pantalla son los necesarios para guardar sonidos de usuario.

#### 6.2.4. Módulo de distribuciones

En este trabajo se llamará **distribución** a una forma cualquiera de configurar los elementos visuales del objeto *nodes*, es decir, todo aquello relacionado con la cantidad, la posición y el tamaño de los círculos presentes en el plano bidimensional. Dado el alto grado de libertad que ofrece este objeto a la hora de configurar las variables ya descritas, Octopus cuenta con una sección llamada módulo de distribuciones (ver Figura 21) y cuya función es administrar un grupo de distribuciones pre-configuradas junto con el listado de combinaciones vértice que son representadas por los círculos. El diseño de estas distribuciones está adaptado a las limitaciones del objeto *nodes* y además está basado en diferentes criterios que dependen del objetivo que se persigue. Dada los objetivos de esta investigación los criterios son: reconocimiento, exploración y automatización.

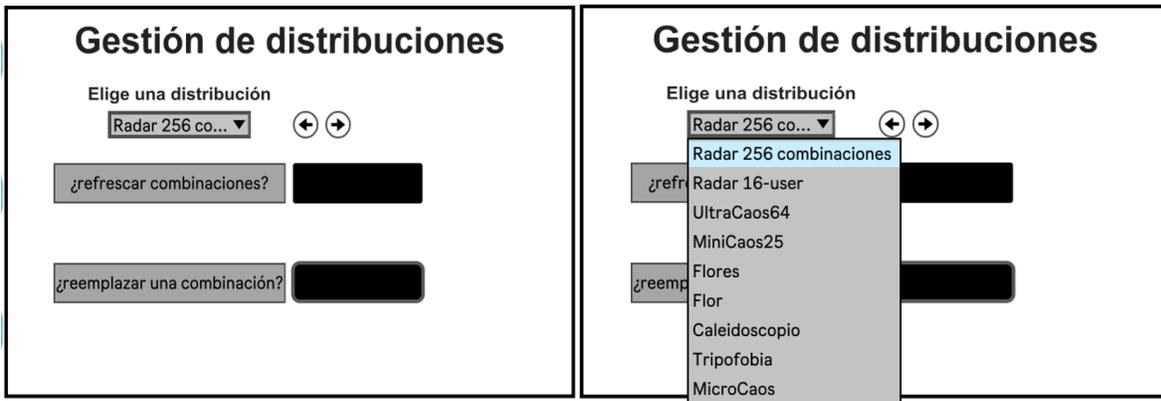


Figura 21: Vista del módulo de distribuciones (izq) y vista del módulo de distribuciones con el listado de distribuciones desplegado (der).

#### 6.2.4.1. Criterio 1: Reconocimiento

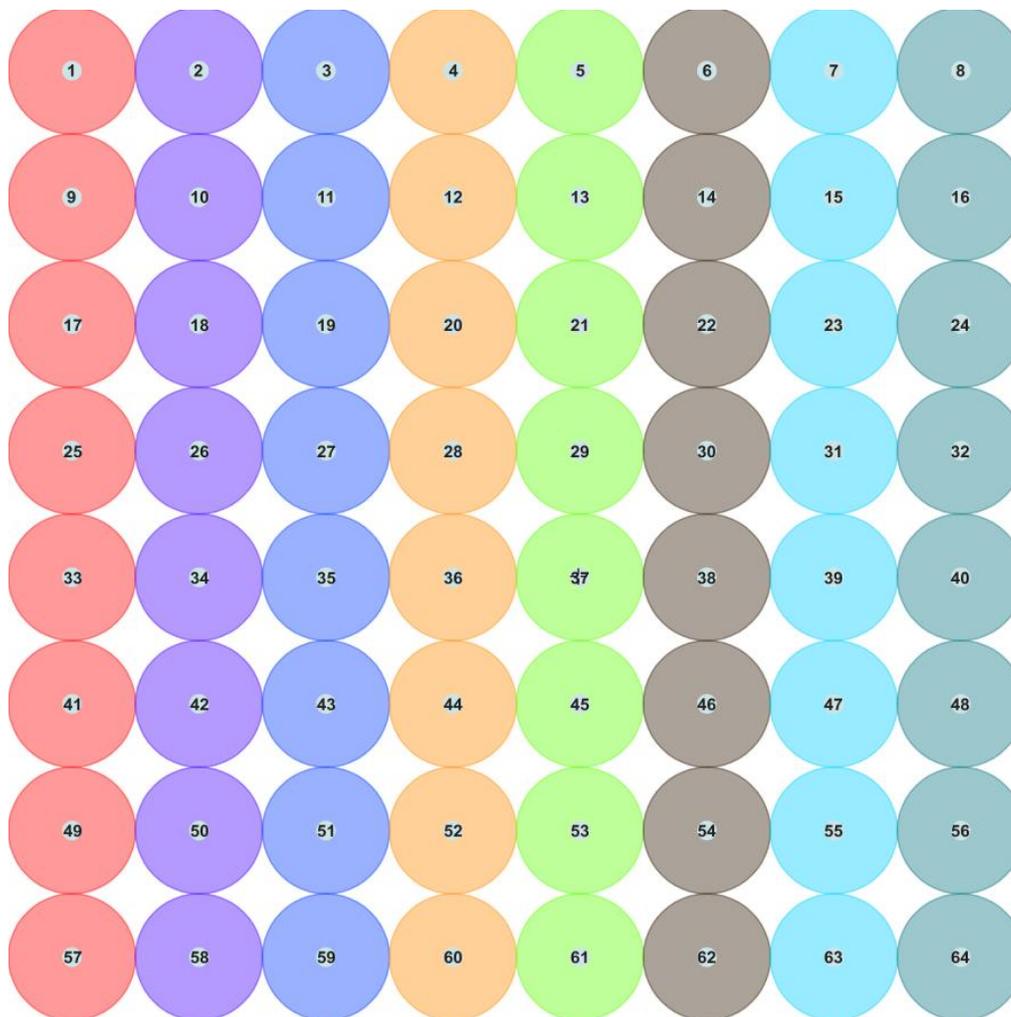
El objetivo de las distribuciones diseñadas bajo este criterio es permitir al usuario monitorear el sonido de las combinaciones en su estado puro, es decir, sin ningún tipo de interpolación de por medio. Lo anterior permite al usuario orientarse y poder determinar si es que debe realizar cambios en etapas anteriores o si es que está conforme con las combinaciones generadas. Además, el hecho de poder monitorear las combinaciones le permite al usuario distinguir las que más le gustan para luego poder aplicar interpolación entre ellas. A continuación se presentan las dos distribuciones diseñadas en base a este criterio.

##### 6.2.4.1.1. Distribución 1: Radar-256

Esta distribución permite al usuario acceder rápida y ordenadamente a cada una de las 256 combinaciones en su forma pura, es decir, sin intersecciones ni interpolación entre ellas (ver Figura 22). Dado las limitaciones del objeto *nodes*<sup>33</sup>, las combinaciones se deben presentar en grupos de 64, por lo que el usuario cuenta con un comando

<sup>33</sup> Este objeto permite un máximo de 64 círculos en el plano.

que le permite navegar entre los cuatro conjuntos de combinaciones ([1,64]; [65,128]; [129,192]; [193,256]). Para observar cómo se comporta esta distribución al controlar un instrumento electrónico dirigirse al siguiente [video](#).



*Figura 22: Vista de la distribución Radar-256 presentando el conjunto [1,64]. Nótese que no existen áreas de intersección entre los círculos.*

#### 6.2.4.1.2. Distribución 2: Radar-16u

Como fue mencionado anteriormente, al interpolar entre dos o más combinaciones el usuario se podrá encontrar con una diversidad de sonidos, de los cuales podrá seleccionar hasta 16 para guardarlos en forma de combinación. Por lo mismo, esta

distribución está diseñada para que el usuario pueda monitorear los sonidos que guardó con anterioridad.

Por defecto Octopus no viene cargado con ningún sonido de usuario, por lo que al inicializar la herramienta esta distribución tendrá todos sus nodos 'inactivos', estado que puede identificarse visualmente por una evidente opacidad su color (ver Figura 23). Al guardar un sonido usuario en alguna de las 16 ranuras, dicho nodo se activa viéndose notablemente más nítido que aquellos que no.

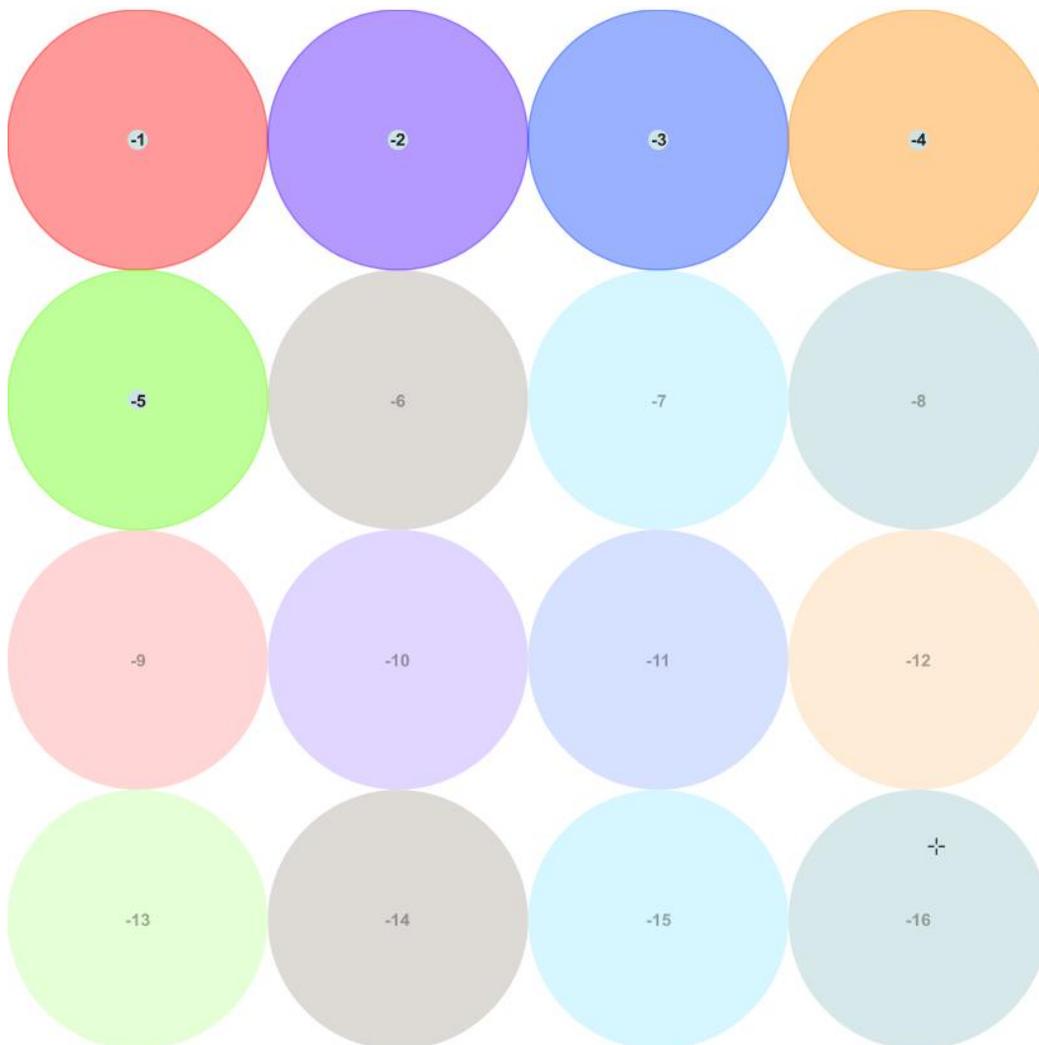


Figura 23: Vista de la distribución Radar-16u. Nótese que los nodos del -1 al -5 están activos, es decir, tienen una combinación de usuario asignada, mientras que el resto de nodos están vacíos por lo que permanecen inactivos.

#### 6.2.4.2. Criterio 2: Exploración

Las distribuciones diseñadas bajo este criterio tienen como objetivo proveer al usuario un acceso rápido a una variada cantidad de timbres. Para ello, estas distribuciones cuentan con un elevado número de nodos ordenados de manera tal que se intersequen unos a otros de diversas formas. Es por lo anterior que estos diseños suelen estar sobrecargados de información visual, y el movimiento de cursor, por pequeño que sea, tiende a generar grandes cambios en el sonido. En términos de las expectativas de uso, la idea es que el usuario utilice estas distribuciones para encontrar timbres de su agrado, tanto para guardarlos en su instrumento en forma de *preset*, como para guardarlos dentro de Octopus en forma de una combinación de usuario. Por otro lado, dado que dentro del plano puede haber máximo 64 círculos no hay manera de representar todas las CVU de forma simultánea, por lo que se vuelve necesario priorizar algunas combinaciones por sobre otras. Por lo mismo, se debe implementar un algoritmo que permita al usuario elegir la combinación asociada a cada círculo así como la posibilidad de poder modificar esa decisión (ver sección 6.2.4.4). A continuación se presentan el conjunto de distribuciones diseñadas bajo el criterio de exploración.

##### 6.2.4.2.1. Distribución 3: Ultra Caos 64

Esta distribución está formada por 64 nodos ordenados en una matriz de 8x8 (ver Figura 24). Dada la forma en la que están organizados se producen tres diferentes zonas de interpolación: una que conecta un par de nodos, una que conecta un trío de nodos y una que conecta un cuarteto de nodos (ver Figura 25). Una demostración del comportamiento de esta distribución puede verse en el siguiente [video](#).

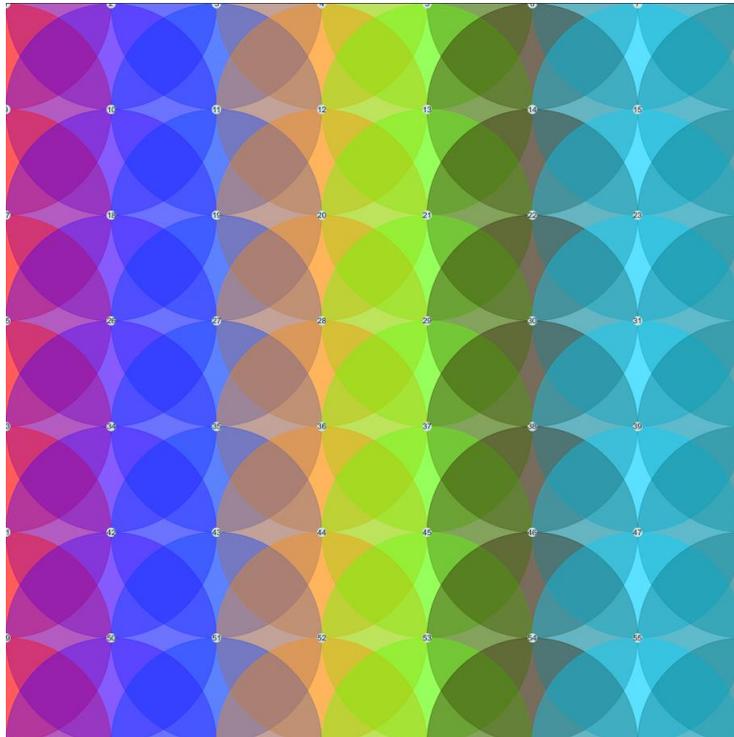


Figura 24: Vista de la distribución Ultra-Caos64.

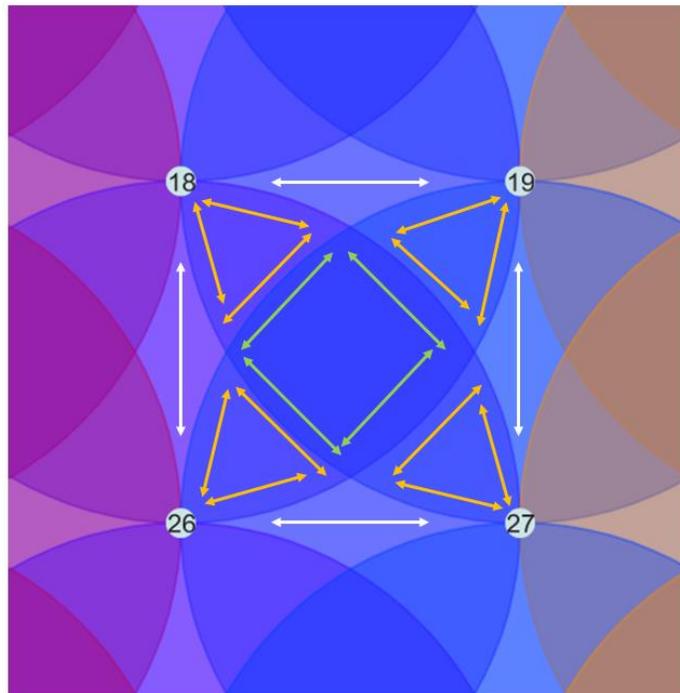


Figura 25: Las tres diferentes áreas de interpolación que se producen con el patrón de Ultra-Caos64. Las flechas blancas muestran las zonas en las que se intersecta un par de nodos, las flechas naranjas las zonas en las que se intersecta un trío de nodos y las flechas verdes las zonas en las que se intersecta un cuarteto de nodos.

#### 6.2.4.2.2. Distribución 4: Mini caos 25

Esta distribución es una versión reducida de la distribución *Ultra Caos 64*. Conformada por una matriz de 5x5, esta distribución presenta un total de 25 nodos entre los cuales se producen las mismas intersecciones que en su versión de 64 nodos (ver Figura 26). Una demostración del comportamiento de esta distribución puede verse en el siguiente [video](#).

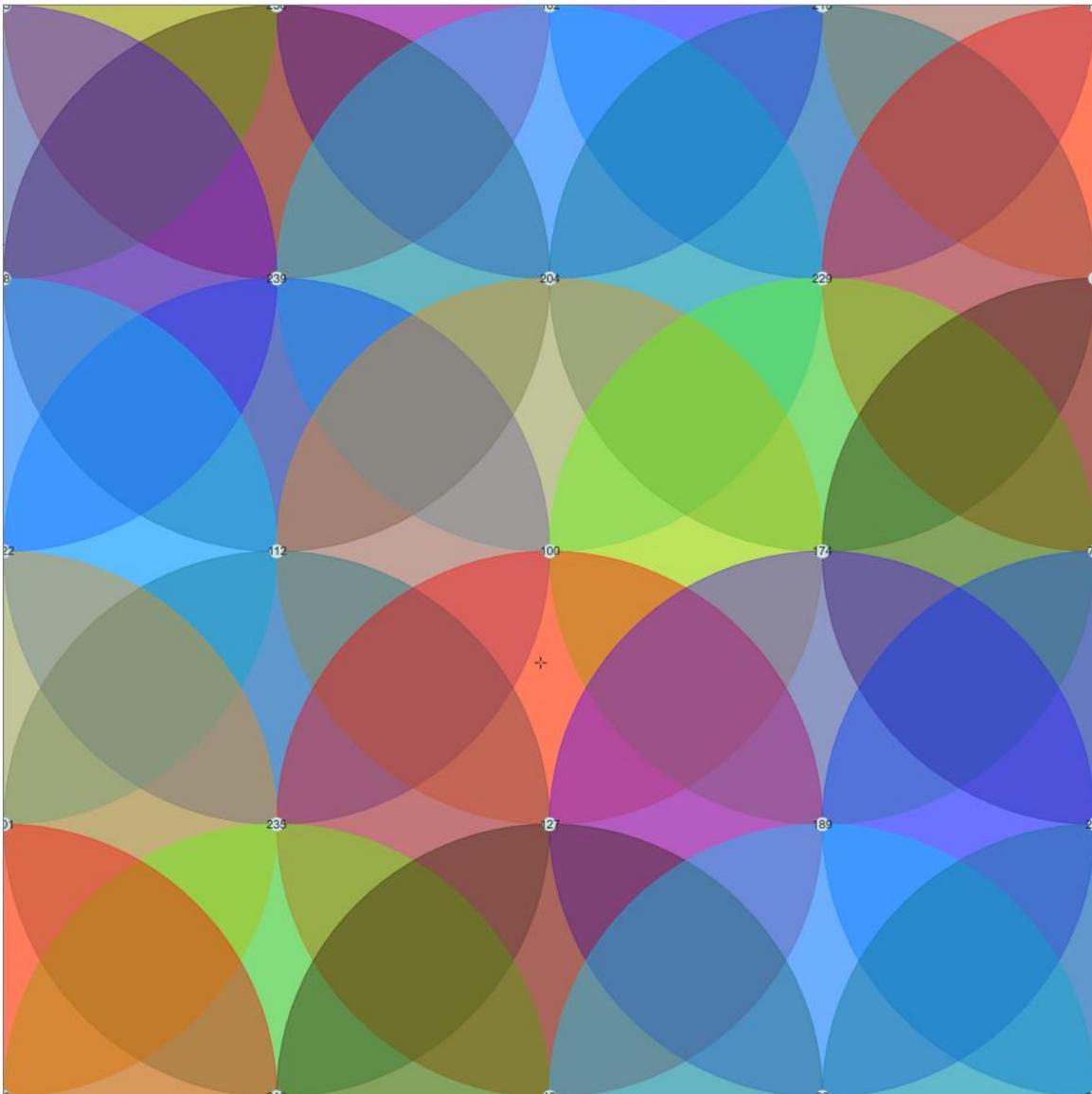


Figura 26: Vista de la distribución Mini-Caos25.

### 6.2.4.2.3. Distribución 5: Flores

Esta distribución está constituida por 39 nodos dispuestos de manera tal que cada uno es un vértice de un triángulo equilátero (ver Figura 27). Debido al patrón en el que están distribuidos los nodos se producen dos zonas de interpolación: una que une un trío de nodos y otra que conecta un cuarteto de nodos (ver Figura 28). Una demostración del comportamiento de esta distribución puede verse en el siguiente [video](#).

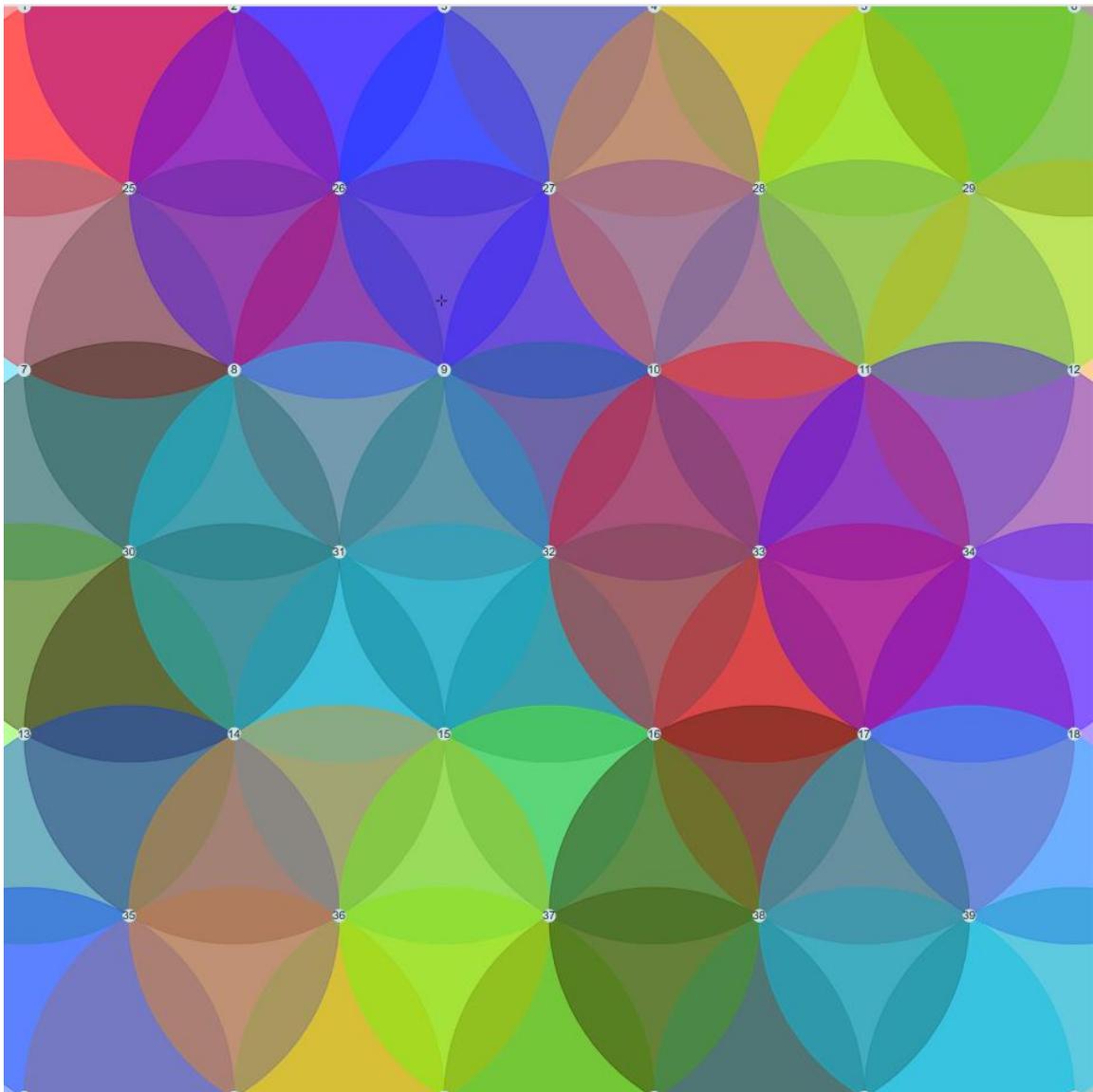


Figura 27: Vista de la distribución Flores.

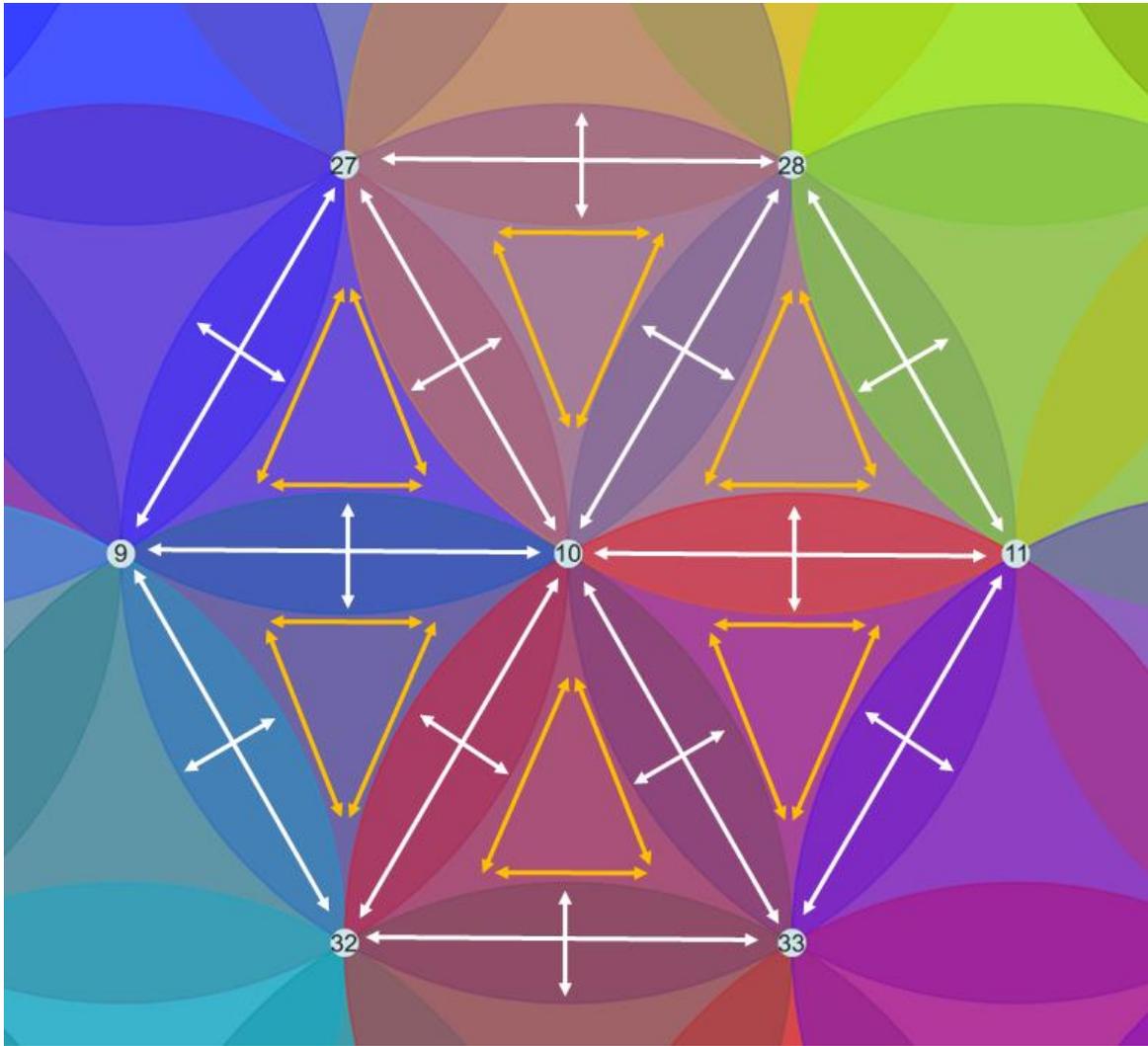
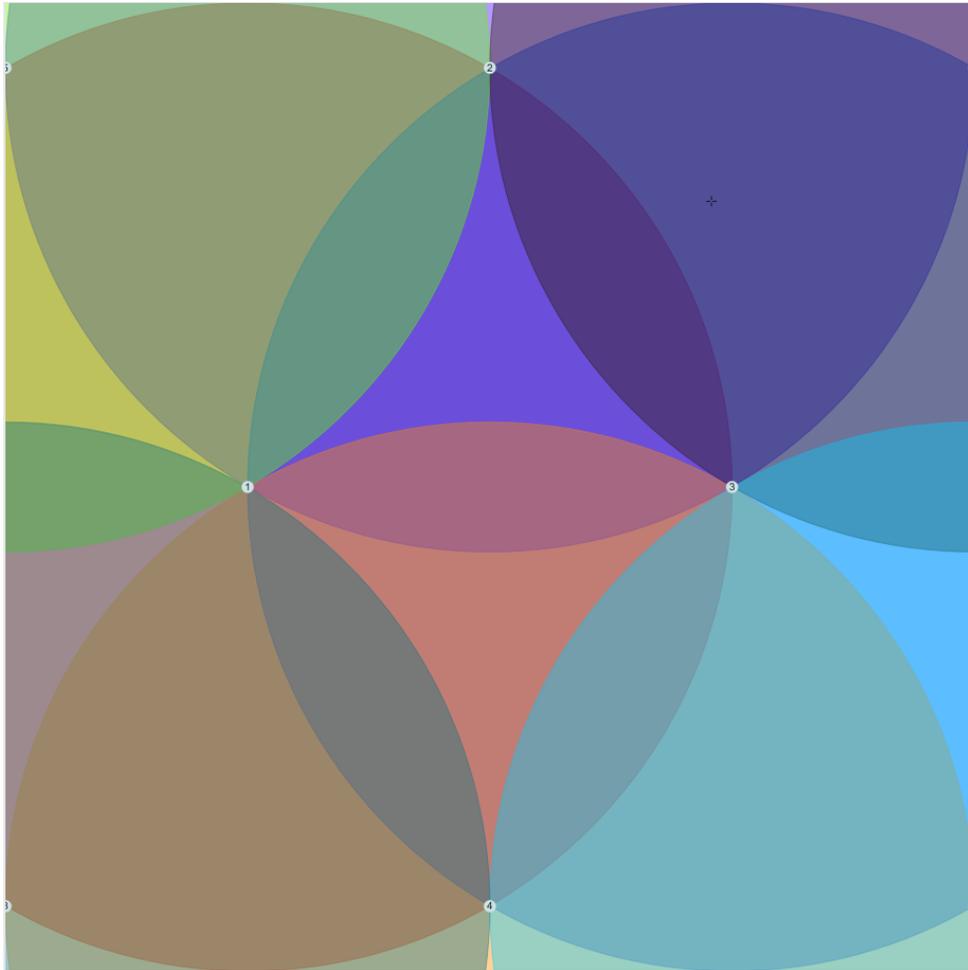


Figura 28: Las dos áreas de intersección que se producen con la distribución Flores. Las flechas blancas muestran las áreas en las que se intersectan cuatro nodos y las flechas naranjas muestran las áreas en las que se intersectan tres nodos.

#### 6.2.4.2.4. Distribución 6: Flor

Constituida por 8 combinaciones, esta distribución es una versión reducida de la distribución *Flores* (ver Figura 29). Cuenta con las mismas intersecciones pero otorga mayor resolución. Una demostración del comportamiento de esta distribución puede verse en el siguiente [video](#).

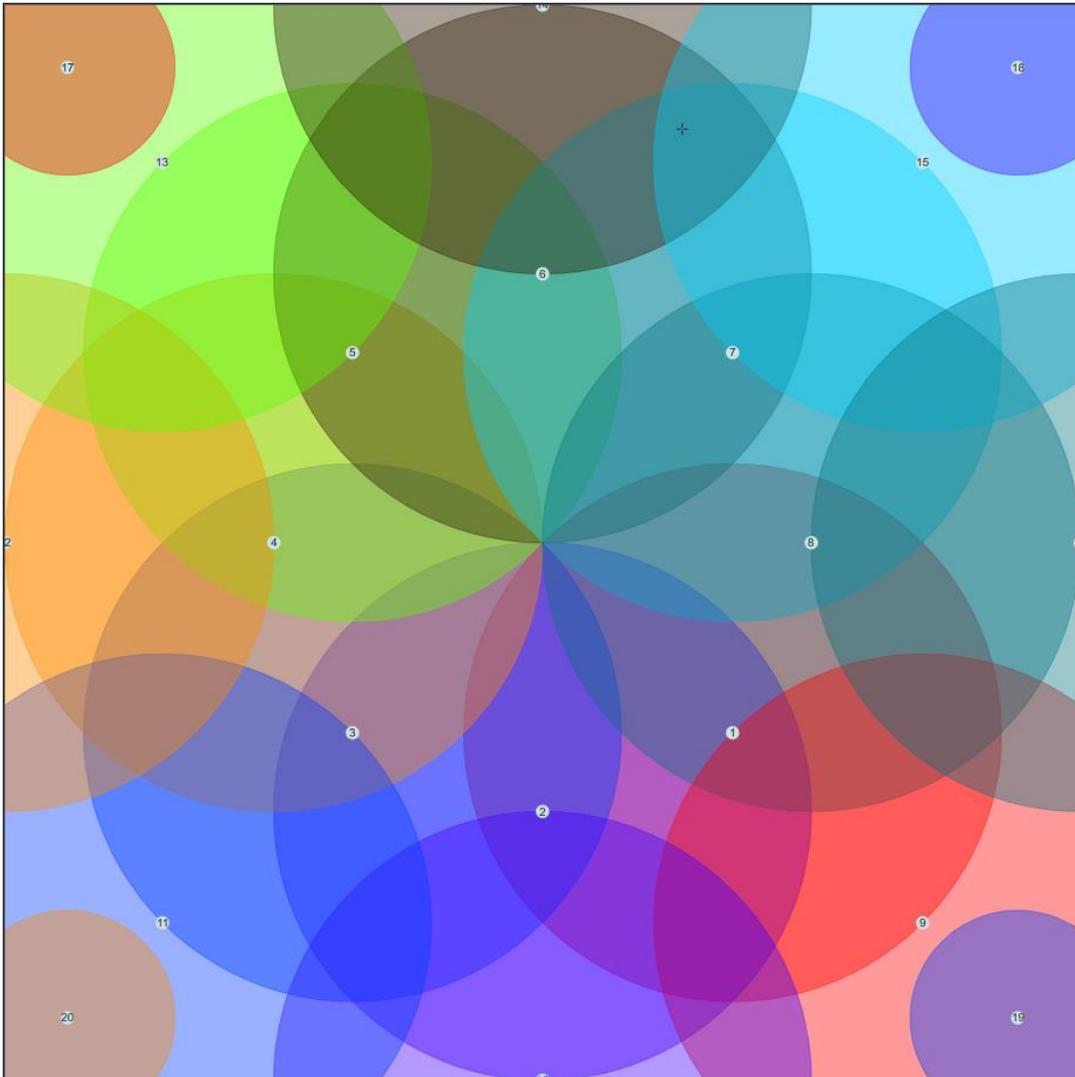


*Figura 29: Vista de la distribución Flor.*

#### 6.2.4.2.5. Distribución 7: Caleidoscopio

Conformada por 20 nodos, esta distribución se construye dibujando una circunferencia con los primeros ocho nodos, luego dibujando otra de mayor radio con los próximos ocho y finalizando con un nodo en cada esquina (ver Figura 30).

Por la forma en la que se distribuyen las combinaciones se generan zonas de intersección de una, dos, tres y cuatro combinaciones. Una demostración del comportamiento de esta distribución puede verse en este [video](#).



*Figura 30: Vista de la distribución Caleidoscopio.*

#### **6.2.4.3. Criterio 3: Automatización**

Las distribuciones diseñadas bajo este criterio están pensadas para ser utilizadas una vez que el usuario ya exploró los resultados de interpolar diferentes pares de combinaciones. La idea es que llegado este punto el usuario tenga claro cuáles son las combinaciones de su preferencia para que pueda informarlas a la herramienta y

que esta proponga una distribución lo más adecuada posible. Por lo mismo, y dado que se trata de un proceso mucho más intencionado, estas distribuciones cuentan con una menor cantidad de nodos y con zonas de intersección más grandes para ayudar a la precisión. A continuación se presentan las distribuciones diseñadas bajo este criterio.

#### 6.2.4.3.1. Distribución 8: Tripofobia

Esta distribución está conformada por un total de 13 combinaciones, y permite interpolar entre múltiples pares de nodos, interpolar entre múltiples tríos de nodos y además tener acceso a las 13 combinaciones en su estado puro (ver Figura 31). Nótese que las zonas en las que se intersecan los círculos son de mayores dimensiones que en otras distribuciones. Una demostración del comportamiento de esta distribución puede verse en este [video](#).

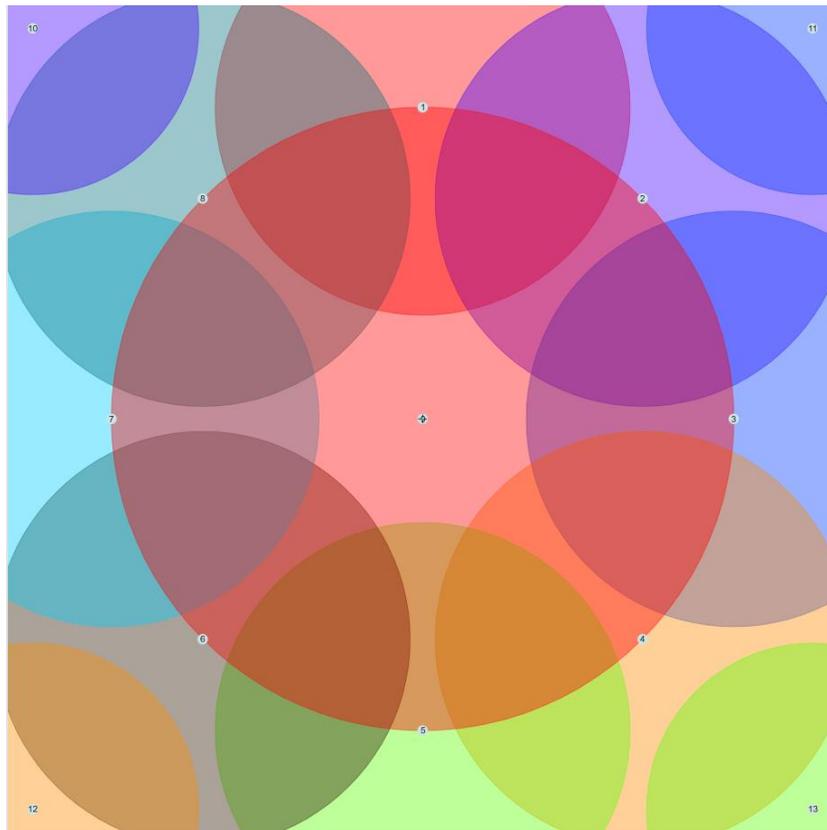


Figura 31: Vista de la distribución Tripofobia.

#### 6.2.4.3.2. Distribución 9: MicroCaos.

Esta distribución es una versión aún más reducida de *Ultra-Caos64*. Se trata de una configuración en la que solamente hay cuatro nodos y en la que maximizan las tres áreas de intersección que se generan con el patrón de *Ultra-Caos64* (ver Figura 32). Una demostración del comportamiento de esta distribución puede verse en este [video](#).

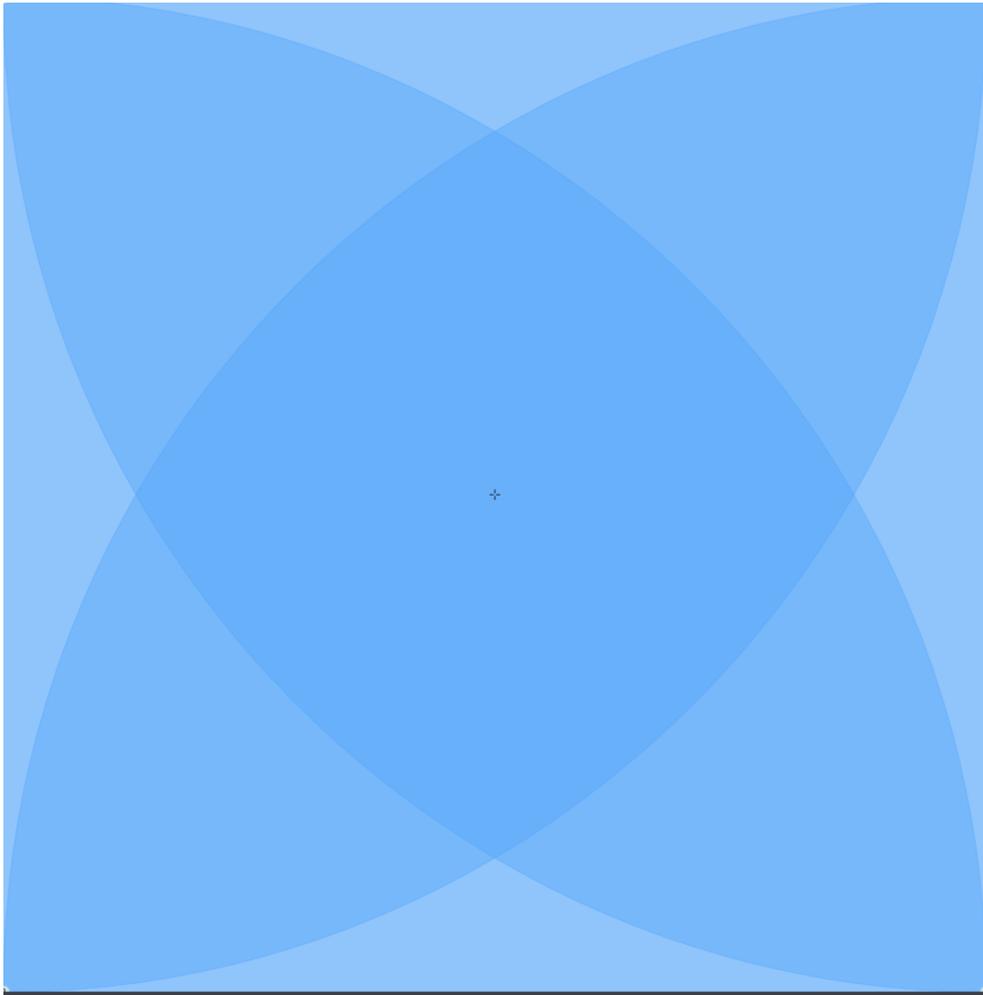


Figura 32: Vista de la distribución MicroCaos

#### 6.2.4.4. Administración de combinaciones vértice en el plano

Además de ofrecer diferentes maneras de ordenar los círculos, el módulo de distribuciones permite al usuario administrar cuáles son las combinaciones vértice representadas por los nodos dispuestos en el plano. Para explicar esto de mejor manera, es necesario comprender que cada vez que se carga una distribución, además de aparecer un grupo de círculos ordenados de una manera en particular, sucede que cada uno de ellos aparece rotulado con un número en el centro cuyo valor corresponde a la combinación vértice que está siendo representada, por lo mismo, dicho número sólo puede ser un valor entre 1 y 256. Además de lo anterior, es importante mencionar que el número con el que aparece rotulado cada nodo es un valor **aleatorio**, por lo mismo, el módulo de distribuciones está implementado con dos algoritmos que permiten al usuario modificar las combinaciones vértice dispuestas en el plano.

##### 6.2.4.4.1. Algoritmo 1: Refrescar combinaciones

Este algoritmo permite al usuario mantener la distribución de los círculos en el plano pero renovando la combinación vértice que tiene asignada cada uno. En caso de que el usuario quisiera refrescar las combinaciones pero manteniendo alguna(s) de las que ya están en el plano, este algoritmo cuenta con una ventana de comandos en la que el usuario puede especificar aquellas combinaciones que desea mantener.

Para accionar este algoritmo basta con pulsar el botón “¿refrescar combinaciones?” y luego ingresar en la ventana de comandos aquellas que se desean mantener. Una vez ingresadas se debe pulsar la tecla *enter* y automáticamente se le asignará un valor aleatorio a todas aquellas que no fueron especificadas (ver Figura 33). Si es que no se quiere conservar ninguna combinación se debe ingresar un cero y pulsar *enter*.



(a)

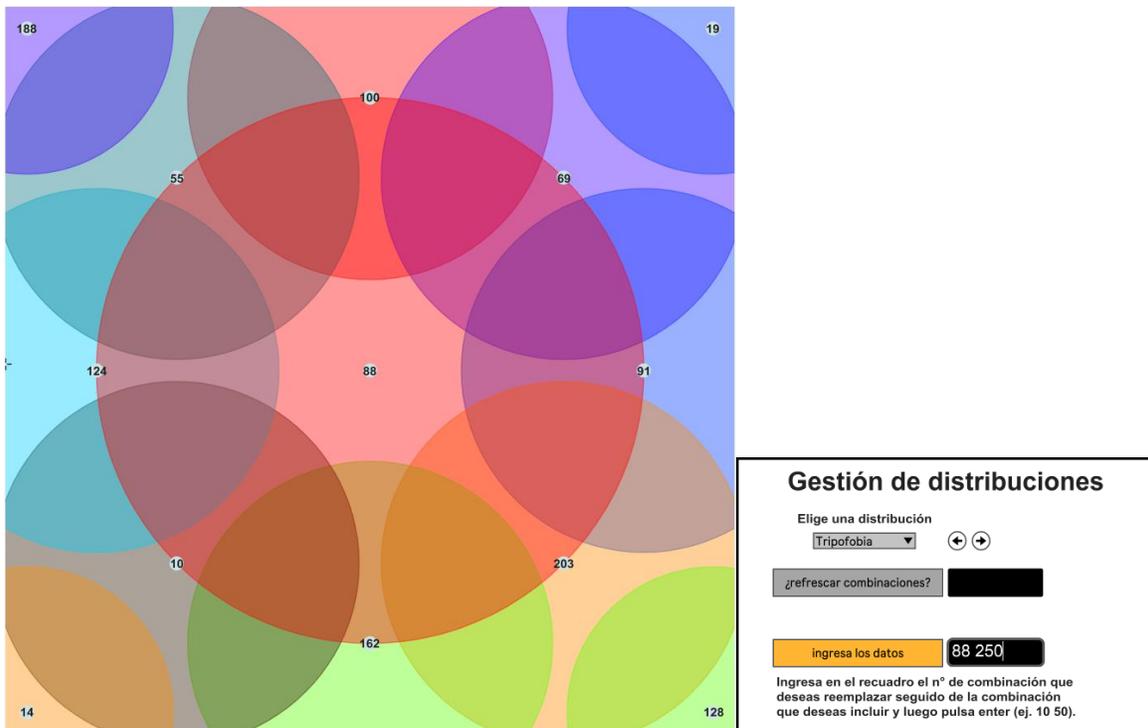


(b)

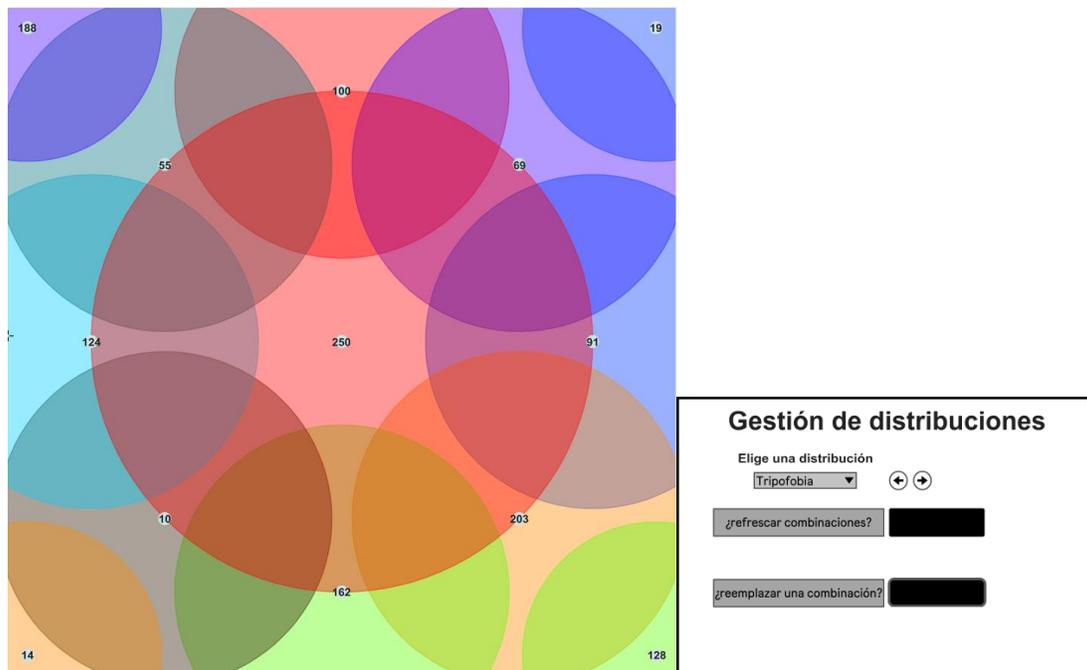
Figura 33: Algoritmo para refrescar las combinaciones en acción. En (a) se puede ver que se está especificando que se mantenga la combinación 245, 32 y 6. En (b) se puede ver que una vez pulsada la tecla 'enter' el botón "¿refrescar combinaciones?" se desactiva y se actualiza el valor de todas las demás combinaciones.

#### 6.2.4.4.2. Algoritmo 2: Reemplazar combinación

Este algoritmo permite al usuario reemplazar una combinación presente en el plano por otra cualquiera. A diferencia del algoritmo anterior, este solamente afecta a un círculo a la vez, y además ofrece mucho más control al usuario ya que no depende en ningún momento de un proceso aleatorio. Para accionarlo sólo basta con pulsar el botón “¿reemplazar una combinación?” y luego ingresar en la ventana de comandos el número de la combinación a reemplazar seguido de la combinación que la reemplazará (ver Figura 34).



(a)



(b)

Figura 34: Algoritmo para reemplazar combinaciones en acción. En (a) se muestra que se busca reemplazar la combinación 88 por la combinación 250. En (b) se puede ver el resultado una vez pulsada la tecla enter.

### 6.2.5. Módulo de trayectorias

Uno de los últimos puntos a trabajar en la herramienta tiene que ver con diseñar diversas maneras de recorrer el plano bidimensional. Por defecto *nodes* permite al usuario recorrer el plano de forma manual con el *mouse*, lo que evidentemente resulta bastante intuitivo dado que el ratón del computador es un objeto cotidiano. Sin embargo, utilizar el mouse significa que el usuario debe usar una de sus extremidades para poder controlar la herramienta, por lo que se dificulta la tarea de generar cambios en el timbre al mismo tiempo que se tocan notas. Por lo mismo, y con la finalidad de ahorrarle recursos al usuario, se implementaron en la herramienta distintas modalidades para recorrer el plano de forma automatizada, las cuales se suman al recorrido manual como una alternativa para interactuar con Octopus.

Para el correcto funcionamiento de las modalidades no-manuales la herramienta pedirá al usuario proveer ciertos datos generales, tales como un valor de BPM y una división de tiempo para definir la velocidad a la que se desplazará el cursor. Estos datos serán utilizados por la mayoría de las modalidades y tienen como finalidad facilitar el uso de la herramienta en contextos musicales. Por otro lado, habrá dos maneras de accionar las trayectorias no-manuales: a través de un botón de inicio/pausa y a través de la información MIDI relacionada con los mensajes *note-on* y *note-off*, es decir, al pulsar las notas de un teclado. Por último, cada una de las modalidades implementadas cuenta con un grupo de comandos dedicados que no se repiten en las otras, por lo que cada una tiene una manera característica de funcionar. A continuación se detallan las diferentes configuraciones.

#### **6.2.5.1. Recorrido Manual:**

Como su nombre lo dice, esta modalidad no cuenta con ningún tipo de automatización y es la modalidad que tiene *nodes* por defecto. Ofrece al usuario un alto nivel de control sobre el resultado sonoro, y es ideal si se está explorando cómo se comportan las distintas zonas de interpolación, o si se están buscando sonidos interesantes para guardar. Al mismo tiempo, esta modalidad resulta poco útil si lo que se busca es generar cambios sostenidos en el timbre a lo largo del tiempo.

En cuanto al grupo de comandos que la acompañan, esta se trata de la única modalidad en la que se permite al usuario guardar sonidos en la memoria interna de la herramienta (ver Figura 35). Una demostración del comportamiento de esta trayectoria puede verse en este [video demostrativo](#).

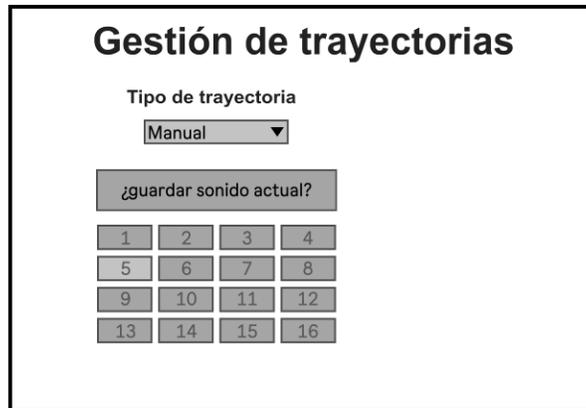


Figura 35: Vista del módulo de trayectorias al seleccionar el recorrido manual. Nótese los comandos para guardar sonidos de usuario.

### 6.2.5.2. Gesto de usuario:

Esta modalidad, de naturaleza más bien híbrida entre lo manual y lo automatizado, permite al usuario grabar un gesto que luego puede ser reproducido. Para esto, lo único que se debe hacer es posicionar el *mouse* sobre el plano, clicar el botón izquierdo y mantenerlo pulsado mientras se dibuja la trayectoria. Una vez soltado el botón, todos los movimientos que se hayan realizado mientras estuvo pulsado quedarán grabados en la memoria interna de la herramienta. Si se desea modificar el gesto simplemente se deben repetir los pasos anteriores.

Al igual que en el resto de modalidades no-manuales, para reproducir el gesto el usuario deberá seleccionar uno de los dos métodos para accionar la trayectoria (ver Figura 36). Además, cabe destacar que esta modalidad es la única del tipo no-manual que **no** depende del BPM ingresado por el usuario, por lo que la velocidad de la trayectoria será exactamente igual a lo que grabó el usuario y no se podrá modificar. Por último, el comando asociado a esta modalidad permite seleccionar que el gesto sea reproducido en bucle o que sea reproducido sólo una vez. Una demostración del comportamiento de esta trayectoria puede verse en este [video](#).

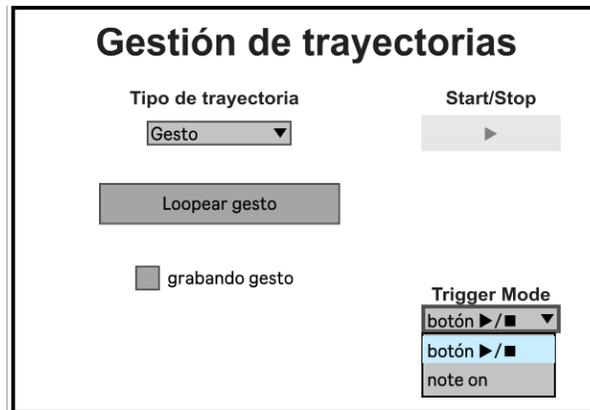


Figura 36: Vista del módulo de trayectorias al seleccionar la opción de Gesto. Nótese que a pesar de ser una trayectoria no-manual no se le puede ingresar información de BPM ni datos para controlar la velocidad de reproducción.

### 6.2.5.3. Secuencia de nodos:

Esta modalidad permite al usuario generar una trayectoria automatizada que pasa por el centro de los nodos y que se repite en bucle. Para ello, esta modalidad cuenta con una ventana de comandos en la que el usuario debe ingresar la secuencia de nodos por los que desea que pase la trayectoria (ver Figura 37). Para especificar a la herramienta cuáles son los nodos simplemente se debe ocupar el número que tiene rotulado cada uno en el centro. Una secuencia está correctamente ingresada cuando los números están presentes en el plano, cuando están separados por un espacio, y cuando están dispuestos en el orden que se desea la secuencia. Cabe destacar que no hay ningún inconveniente con repetir los valores más de una vez. Una vez ingresada la secuencia basta con pulsar el botón *enter* y luego accionarla. La velocidad de reproducción de la secuencia dependerá del BPM y la división de tiempo ingresada por el usuario. Una demostración del comportamiento de esta trayectoria puede verse en este [link](#).

The screenshot shows a software interface titled "Gestión de trayectorias". It contains several control elements:

- Tipo de trayectoria:** A dropdown menu currently showing "Secuencia d...".
- Start/Stop:** A button with a right-pointing arrow.
- BPM:** A text input field containing the value "120".
- Rate:** A dropdown menu showing "1/4".
- Trigger Mode:** A dropdown menu showing "botón ▶/■".
- Ingresar tu secuencia:** A large black rectangular area, likely a placeholder for a drawing or a list of points.
- Secuencia actual:** A text input field containing the sequence of numbers "115 121 50 50 225".

Figura 37: Vista del módulo de trayectorias al seleccionar la opción secuencia de nodos.

#### 6.2.5.4. Secuencia de puntos:

De manera similar a la anterior, esta modalidad permite generar una trayectoria que pasa por una secuencia de hasta ocho puntos ingresados por el usuario. Para ingresar la secuencia basta con pulsar el botón "¿ingresar puntos?", seleccionar en el plano los puntos de la secuencia en el orden deseado, y luego volver a pulsar el botón (ver Figura 38). Cabe destacar que como se puede seleccionar un máximo de ocho puntos, al noveno *click* se sobre-escribirá el primero, y así sucesivamente. Con esta modalidad, a diferencia de la anterior, ya no es requisito que la trayectoria pase por el centro de los nodos, por lo que la secuencia puede estar más enfocada en las zonas de intersección. Una demostración del comportamiento de esta trayectoria puede verse [acá](#).

### Gestión de trayectorias

Tipo de trayectoria

Note Off

**ingrese puntos (máx 8)**

**BPM**

**Puntos ingresados:**

**Rate**

**Trigger Mode**

Figura 38: Vista del módulo de trayectorias al seleccionar la opción secuencia de puntos.

#### 6.2.5.5. Circunferencia:

Como su nombre lo indica, esta modalidad permite que el movimiento del cursor sea completamente circular. Los comandos asociados permiten al usuario seleccionar un radio para la circunferencia, así como un centro y la posibilidad de que la trayectoria se repita en bucle (ver Figura 39). El tiempo que demora el cursor en completar la circunferencia está determinado por el BPM y la división de tiempo ingresada por el usuario. Además, al igual que en el resto de modalidades la trayectoria podrá ser accionada por un botón o por las teclas del instrumento.

Esta configuración resulta ideal para situaciones en las que el usuario desea que el sonido varíe rítmicamente a lo largo del tiempo, dado que la trayectoria circular se comporta de forma muy similar a un LFO. Una demostración del comportamiento de esta trayectoria puede verse en este [video](#).

### Gestión de trayectorias

Tipo de trayectoria  
Circunferencia ▼

radio círculo:  
0.07

Loopear trayectoria

Note Off

BPM  
95

Rate  
1/2 ▼

Trigger Mode  
note on ▼

Figura 39: Vista del módulo de trayectorias al seleccionar la opción circunferencia.

#### 6.2.5.6. Espiral:

De manera muy similar a la anterior, esta modalidad genera un movimiento en espiral por parte del cursor. Los comandos asociados permiten que el usuario elija un radio para el espiral así como la posibilidad de que se repita en bucle (ver Figura 40). Una demostración del comportamiento de esta trayectoria puede verse [acá](#).

### Gestión de trayectorias

Tipo de trayectoria  
Espiral ▼

radio espiral:  
0.25

Loopear trayectoria

Start/Stop  
▶

BPM  
76

Rate  
1/8 ▼

Trigger Mode  
botón ▶/■ ▼

Figura 40: Vista del módulo de trayectorias al seleccionar la opción espiral.

## 7. Resultados

En el presente capítulo se exponen los resultados obtenidos luego de haber presentado el prototipo de forma individual a un grupo de cuatro personas, quienes luego de haber probado la herramienta debieron contestar una serie de preguntas. La metodología empleada para diseñar la experiencia de usuario fue presentada en el capítulo de método, así mismo, la guía de entrevista empleada para la recopilación de datos está adjunta en el anexo A (ver página 135). Se incluye además en este capítulo una sección de análisis de los resultados obtenidos.

### 7.1. Perfiles de los entrevistados.

Sujeto	Edad	Ocupación	Instrumento	Tiempo de uso
Sujeto 1	24 años	Estudiante de Composición Musical	Operator <sup>34</sup> (software)	4 años
Sujeto 2	25 años	Ingeniero en Sonido	Roland GAIA SH-01 <sup>35</sup> (hardware)	4 años
Sujeto 3	24 años	Músico Autodidacta	Element 2.0 <sup>36</sup> (software)	1 año
Sujeto 4	25 años	Estudiante de Teoría de la Música y Profesor de Música	MicroFreak <sup>37</sup> (hardware)	1,5 años

Tabla 3: Datos generales sobre los entrevistados

---

<sup>34</sup> <https://www.ableton.com/en/manual/live-instrument-reference/#26-6-operator> (Live Instrument Reference — Ableton Reference Manual Version 11 | Ableton sin fecha)

<sup>35</sup> [https://www.roland.com/global/products/gaia\\_sh-01/](https://www.roland.com/global/products/gaia_sh-01/) (Roland - GAIA SH-01 | Synthesizer sin fecha)

<sup>36</sup> <https://www.waves.com/plugins/element#tab-tech-specs> (Virtual Analog Synth — Element 2.0 | Waves sin fecha)

<sup>37</sup> <https://www.arturia.com/products/hardware-synths/microfreak/overview> (Arturia - MicroFreak - MicroFreak sin fecha)

## 7.2. Respuestas de los entrevistados

Preguntas previas a la experimentación con el prototipo				
	Instrumento y tipo	Uso habitual del instrumento	Proceso para diseñar sonidos en el instrumento	Dificultades y rapidez para obtener resultados satisfactorios
Sujeto 1	Operator (software)	<ul style="list-style-type: none"> <li>- Producción Musical</li> <li>- Sonidos tonales como pads y leads</li> <li>- Sonidos atmosféricos</li> <li>- Música Generativa</li> </ul>	<ul style="list-style-type: none"> <li>- Comienza diseñando el sonido desde cero.</li> <li>- Inicia experimentando con sección de osciladores</li> <li>- Atención especial a envolventes y filtros</li> <li>- Prueba diferentes algoritmos de síntesis FM</li> <li>- Gusta de agregar "movimiento" a las voces controlando sus parámetros con LFOs</li> </ul>	<ul style="list-style-type: none"> <li>- 30 minutos para armar un sonido base</li> <li>- En condiciones óptimas tarda entre 1 y 2 horas para llegar a texturas más complejas</li> <li>- Una dificultad que se encuentra habitualmente es que no conoce todas las funciones del instrumento</li> <li>- Dice que no todos los intentos son exitosos y que puede tomarle varias sesiones en llegar a un resultado satisfactorio</li> </ul>
Sujeto 2	Roland GAIA SH-01 (hardware)	<ul style="list-style-type: none"> <li>- Principalmente para performances en vivo</li> <li>- Recrear sonidos de sintetizadores grabados</li> <li>- Uso como controlador MIDI y sintetizador</li> <li>- Diseño de pads y sonidos "muy melódicos"</li> <li>- Uso ocasional en estudio para componer arreglos muy específicos</li> </ul>	<ul style="list-style-type: none"> <li>- Suele diseñar sonidos desde cero, en contadas ocasiones parte desde un preset</li> <li>- Exploración enfocada en los osciladores y el filtro</li> <li>- Para recrear sonidos escucha la referencia a medida que lo diseña</li> </ul>	<ul style="list-style-type: none"> <li>- Para recrear sonidos tarda entre 0,5 y 1 hora en obtener un buen resultado</li> <li>- Para hacer sonidos "raros" tipo FX tarda entre 1 y 1,5 hrs.</li> <li>- Una vez llegado a un buen resultado, sigue experimentando con el sonido por diversión.</li> <li>- Como dificultad se encuentra que, a diferencia de sus sintetizadores de software, el SH-01 es muy poco visual por lo que le cuesta trabajo configurar la envolvente de los filtros</li> </ul>
Sujeto 3	Element 2.0 (software)	<ul style="list-style-type: none"> <li>- Producción Musical</li> <li>- Búsqueda tonal</li> <li>- Diseño de leads melódicos</li> <li>- Diseño de bajos</li> </ul>	<ul style="list-style-type: none"> <li>- Suele explorar los presets hasta llegar a uno que considere buen punto de partida</li> <li>- Se enfoca en la modificación de aquellos parámetros cuyo funcionamiento conoce y NO utiliza aquellos que no</li> <li>- A priori descarta configuraciones porque cree que puede predecir el resultado sonoro que darán</li> </ul>	<ul style="list-style-type: none"> <li>- Para sonidos electrónicos tarda poco tiempo ya que considera que es fácil obtenerlos con su instrumento.</li> <li>- Para sonidos más realistas demora más tiempo</li> </ul>
Sujeto 4	Arturia MicroFreak (hardware)	<ul style="list-style-type: none"> <li>- Producción Musical en casa</li> <li>- Diseño de pads y colchones armónicos que mutan lentamente</li> <li>- Diseño de secuencias aleatorias para melodías o leads</li> <li>- Aunque el instrumento lo permite, descarta su uso para percusiones y bajos</li> <li>- Eventualmente uso para performances en vivo</li> </ul>	<ul style="list-style-type: none"> <li>- Solía diseñar los sonidos desde cero</li> <li>- Actualmente explora uno a uno los presets para encontrar un punto de partida adecuado</li> <li>- Tiene un par de presets favoritos a los que acude a menudo para modificarlos</li> <li>- Experimenta con distintos motores de síntesis (FM, <i>Wavetable</i>, etc)</li> <li>- Pone especial atención a las envolventes y a los filtros.</li> <li>- Busca sincronías entre las envolventes y un LFO lento</li> </ul>	<ul style="list-style-type: none"> <li>- Cuando modifica los presets que conoce llega a buenos resultados en minutos</li> <li>- Cuando sólo busca explorar puede pasar horas buscando controles y "fórmulas", no busca necesariamente un sonido en específico.</li> <li>- Como dificultad considera que siempre llega a resultados parecidos, ya que ocupa los mismos métodos y controles.</li> <li>- Considera que necesita conocer el sintetizador más a fondo</li> <li>- Considera que le falta disposición personal para enfrentar algo nuevo.</li> </ul>

Tabla 4: Respuestas de los entrevistados

Preguntas posteriores a la experimentación con el prototipo			
	Impresiones de Octopus	Opinión sobre el diseño de Octopus	Facilidad de uso
Sujeto 1	<ul style="list-style-type: none"> <li>- Permite llegar lejos con la exploración tímbrica de manera <b>"muy rápida"</b></li> <li>- Genera movimientos y automatizaciones muy complejas que él normalmente no consigue dado la "simpleza" de los LFO que utiliza.</li> <li>- Se obtienen <b>resultados no esperados</b></li> <li>- Afirma que hay maneras de predisponerse a lo que va a sonar, pero que igual el caos gana</li> <li>- Valora positivamente la posibilidad de ingresar un gesto para recorrer el plano, considera que añade un componente interpretativo y performático</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Considera fácil el proceso de mapear los controles del instrumento</b></li> <li>- Dice que es bueno que Octopus no tenga demasiadas ventanas, a diferencia de otros <i>softwares</i> que ha probado G5</li> <li>- <b>Encuentra positivo que la interfaz esté dividida por módulos</b> y que estos sean pocos</li> <li>- <b>Valora positivamente que el módulo de trayectorias incluya un <i>pop-up</i> con instrucciones</b></li> <li>- Piensa que el módulo de interpolación es visualmente atractivo</li> <li>- Encuentra <b>incómodo tener que cambiar de ventana con frecuencia</b> entre Octopus y Live</li> </ul>	<ul style="list-style-type: none"> <li>- Señala que una vez se le explicó qué es lo que hacía cada módulo le pareció mucho más sencilla.</li> </ul>
Sujeto 2	<ul style="list-style-type: none"> <li>- Es una herramienta <b>"muy fuerte"</b> y <b>complementaria</b></li> <li>- Tiene una <b>curva de aprendizaje larga</b></li> <li>- Considera que permite una exploración sonora de diferentes características dependiendo de la configuración que se use (él probó con pads y leads)</li> <li>- Permite encontrar <b>opciones que no tenías contempladas</b></li> <li>- Piensa que <b>ayuda "mucho" en términos de expresión</b> para performances en vivo si es que se configura de la manera adecuada</li> <li>- Dice que le gustaría probarla con Serum, otro instrumento que él considera más versátil</li> </ul>	<ul style="list-style-type: none"> <li>- Dice que a primera vista la interfaz le pareció un poco intimidante</li> <li>- Considera que <b>el proceso para comenzar a utilizar la herramienta es un poco lento</b></li> <li>- Cambiaría el tamaño de los módulos para que quepan todos en la pantalla</li> <li>- Reconoce que se debe ingresar mucha información a la herramienta pero que dicha cantidad es adecuada y proporcional</li> </ul>	<ul style="list-style-type: none"> <li>- Considera que <b>la curva de aprendizaje no es trivial y que definitivamente se tiene que entregar un manual</b> o tutorial a usuarios nuevos</li> <li>- Afirma que una vez se tiene claridad de la función de cada módulo todo se vuelve muy claro</li> </ul>
Sujeto 3	<ul style="list-style-type: none"> <li>- <b>Aparecen combinaciones de parámetros que él no habría decidido juntar de forma consciente</b></li> <li>- Te permite encontrar combinaciones de forma <b>"mucho más rápida"</b> que si lo hicieras manualmente</li> <li>- <b>Afirma que llegó a rincones que a través de su método habitual no habría conseguido</b></li> <li>- Dice que al tener una esencia "tan matemática" Octopus no tiene criterio y simplemente arroja todas las posibilidades. Por lo mismo, ayuda a explorar y a encontrar <b>sonidos que pueden sorprender</b>.</li> <li>- <b>Le gusta que la máquina le ofrece alternativas y que él sólo debe escuchar.</b></li> <li>- Al principio <b>pensó que ocho parámetros era poco pero luego de probar la herramienta consideró que era un número adecuado.</b></li> </ul>	<ul style="list-style-type: none"> <li>- Señala que <b>el proceso de mapear los parámetros es incómodo</b></li> <li>- Considera que el proceso de configurar la herramienta toma demasiado tiempo</li> <li>- Dice que <b>el diseño de la interfaz le parece tosco</b> y que podría ser mejorado</li> <li>- <b>Le gusta que sólo sean ocho parámetros</b>, siente que simplifica las cosas</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Afirma que le parece intuitiva</b> y que lo más extraño es el interpolador gráfico</li> </ul>
Sujeto 4	<ul style="list-style-type: none"> <li>- Comenta que hasta que empezó a sonar no había comprendido el poder que tiene la herramienta</li> <li>- Herramienta creativa con <b>"muchísimo potencial"</b></li> <li>- <b>Ahorra tiempo</b> en la exploración tímbrica</li> <li>- Tiene <b>potencial de herramienta compositiva</b> dado la posibilidad de secuenciar cambios en el timbre</li> <li>- <b>Le gusta la limitante de solo trabajar con ocho parámetros</b></li> <li>- El prototipo tiene <b>demasiada aleatoriedad</b> para su gusto</li> <li>- Le gustaría que la herramienta estuviera <b>pensada para un uso más continuado</b>, ve como una buena opción poder guardar sesiones.</li> <li>- <b>Valora como algo positivo (y a reforzar) aquellos elementos que otorgan control al usuario.</b> Ej. guardar sonidos de usuario, poder decidir qué combinación está presente en el plano</li> <li>- Considera (con recelo) que la herramienta <b>suspende/minimiza el componente racional</b> a la hora de diseñar el sonido, y en cambio pone al usuario en un estado pasivo en el que sólo debe decidir si el sonido le gusta o no</li> </ul>	<ul style="list-style-type: none"> <li>- Considera que <b>comenzar a utilizar la herramienta es un proceso un poco tosco</b>, pero que cinco minutos después se transforma en una sala de juegos increíble.</li> <li>- La interfaz está un "punto cómodo"</li> <li>- Piensa que <b>puede ser un tanto complicado el lenguaje matemático</b> que maneja la herramienta</li> <li>- <b>Valora positivamente el <i>feedback</i> que entrega Octopus de la posición de los controles al ser interpolados</b>, considera que ello sirve para comprender mejor lo que está sucediendo dentro del instrumento</li> </ul>	<ul style="list-style-type: none"> <li>- Considera que es <b>"peligrosamente fácil de usar"</b>. Mira con recelo el hecho de que según él hay una disminución del proceso racional a la hora de diseñar los sonidos.</li> </ul>

Tabla 5: Respuestas de los entrevistados

Preguntas posteriores a la experimentación con el prototipo		
Relación entre Octopus y el flujo habitual de trabajo	¿Tiene cabida en tu forma de trabajar? ¿Cómo?	Opinión sobre timbres conseguidos
<p><b>Sujeto 1</b></p> <p>- Comenta que su forma de trabajar el diseño de sonidos en su instrumento siempre incluye agregar "movimiento" a las voces por medio de secuencias y LFO. Por lo mismo, dice que todas las funciones de automatización del recorrido le parecen útiles.</p>	<p>- <b>Afirma que Octopus sí tiene cabida en su forma de trabajar</b> y que pareciera una herramienta que podría usar siempre</p> <p>- Lo utilizaría para hacer texturas más raras y para encontrar sonidos complejos de manera sencilla</p> <p>- Dice que <b>sería interesante ocuparla como "forma final"</b>. Se imagina grabar una sesión en la que puede generar una composición libre o recortar algunos trozos de la interpretación para manipularlos después.</p>	<p>- Señala que de <b>los timbres que aparecieron hubo algunos que jamás había escuchado de Operator</b>. Sostiene que eso se debe a que normalmente no mueve ocho parámetros de forma simultánea.</p> <p>- Dice que muchos parecían una especie de baile coordinado entre los diferentes parámetros</p> <p>- <b>Comenta que hubo algunas configuraciones en las que le costó mucho establecer los rangos adecuados para obtener buenos sonidos</b>. Se encontró con combinaciones que no producían sonidos.</p>
<p><b>Sujeto 2</b></p> <p>- Comenta que la herramienta representa una mejora leve a alguno de los usos que ya le da a su instrumento</p> <p>- Cuenta que al tocar en vivo le gusta incorporar sonidos raros y transiciones en las que mueve aleatoriamente algunos parámetros, por lo mismo, afirma que Octopus le ayudaría a encontrar nuevas opciones.</p> <p>- Dice que le gusta jugar con los pads modificándolos a través del tiempo en cuanto armonía, y sostiene que Octopus le sería útil para complementar esa modificación con una dimensión tímbrica.</p>	<p>- <b>Afirma que Octopus sí tiene cabida en su forma de trabajar</b></p> <p>- Señala que debe aprender a manejarla mejor para así poder determinar cuál es el uso específico que podría darle</p> <p>- Dice que para este instrumento en particular <b>le llama la atención la idea de utilizarla en performances en vivo</b></p> <p>- Comenta que le gustaría probarla con más instrumentos para poder encontrarle un uso en producción musical</p>	<p>- Señala que los timbres conseguidos eran locos y extraños, aunque afirma también que eso era lo que estaba buscando</p> <p>- <b>Comenta que aparecieron sonidos a los que no esperaba llegar y a sonidos en los que no había pensado</b></p>
<p><b>Sujeto 3</b></p> <p>- Comenta que normalmente le incomoda no poder prever los resultados sonoros, sin embargo, considera que esa es la esencia de esta herramienta y es justamente lo que le parece atractivo de ella.</p> <p>- Dice que tiene el problema de siempre llegar a los mismos sonidos, por lo que esta herramienta podría servirle de solución.</p>	<p>- <b>Afirma que Octopus sí tiene cabida en su forma de trabajar.</b></p> <p>- Comenta que su búsqueda de sonidos suele ser muy tonal y que abandona bastante lo relacionado a la textura del sonido. Por lo mismo, dice que Octopus "definitivamente" le podría ayudar.</p>	<p>- Dice que se sorprendió bastante con el hecho de que <b>mapeó los parámetros que siempre utiliza y que aún así aparecieron sonidos que no se imaginaba que se podrían conseguir.</b></p>
<p><b>Sujeto 4</b></p> <p>- Comenta que la interacción que ofrece Octopus no se parece en nada a su forma habitual de trabajar</p> <p>- Agrega que al diseñar sonidos manipula conscientemente cada parámetro para llegar a un objetivo, y siente que con Octopus eso se le va de las manos</p> <p>- Dice que Octopus le propone nuevas formas de flujo de trabajo y que va a quedar a a su criterio hasta qué punto ocuparía la herramienta</p> <p>- Señala que normalmente le toma trabajo diseñar un sonido, por lo que una vez alcanzado suele conservarlo por largo tiempo. Agrega que con Octopus le cuesta tan poco esfuerzo obtener un sonido que al mismo tiempo le es muy fácil desecharlo.</p>	<p>- <b>Afirma que la herramienta sí tiene cabida en su forma de trabajar.</b></p> <p>- Cree que hay que aprender a contextualizarla y saber en qué situación ocuparla</p> <p>- <b>Considera que la podría ocupar en la parte más primitiva y exploratoria de la creación musical</b>, y dado su exceso de aleatoriedad no se sentiría tan cómodo dándole espacio en otras etapas del proceso creativo.</p> <p>- Le parecería útil utilizarla en la gestación de uno o varios sonidos que podrían formar parte de una pieza en su conjunto</p> <p>- <b>Advierte que, si bien la ocuparía para explorar sonidos, una vez encontrados los recrearía de forma manual en el sintetizador para tener más control sobre ellos.</b></p>	<p>- Comenta que le gusta la variedad de sonidos que se pueden conseguir, y <b>le parece interesante que de todas formas estos comparten una esencia similar.</b></p> <p>- Insiste en el hecho de que si se encontraba con timbres que no le gustaban, simplemente movía el cursor en búsqueda de aquellos que sí.</p>

Tabla 6: Respuestas de los entrevistados

Preguntas posteriores a la experimentación con el prototipo	
Timbres conseguidos con Octopus vs timbres conseguidos habitualmente	Aspectos negativos y a mejorar
<p><b>Sujeto 1</b></p> <ul style="list-style-type: none"> <li>- Afirma que sin Octopus difícilmente podría haber conseguido los timbres a los que llegó. Lo atribuye al hecho de que jamás trabaja con la automatización de ocho parámetros de forma simultánea, dice que ese hecho lo cambia todo.</li> </ul>	<ul style="list-style-type: none"> <li>- Le pareció problemático que Octopus está puesto en lenguaje MIDI (valores del 0 a 127) mientras que muchos parámetros de Operator utilizan distintas unidades de medida, tales como porcentaje, hertz, dB, factor Q, etcétera. <b>Sugiere buscar la forma de traducir las unidades de medida.</b></li> <li>- Sumado a lo anterior, hizo notar que algunos parámetros de Operator no responden bien al ser controlados con Octopus. Se encontró que el origen del problema es que Operator recibe valores entre 0 y 1 para algunos parámetros mientras que para otros espera valores entre 0 y 127.</li> <li>- <b>Se encontró con combinaciones que por algún motivo no resultaban en ningún sonido y no pudo encontrar el origen del problema.</b></li> <li>- Recomienda incluir más variedad de tiempos en los <i>rates</i> del módulo de trayectorias. Le gustaría que hubiese sincopa y tiempos mucho más lentos que los que ofrece Octopus.</li> <li>- Recomienda añadir más posibilidades para controlar el <i>playback</i> de los gestos (reproducirlos en reversa, cambiar la velocidad, recortarlos).</li> <li>- Recomienda que los gestos no deban ser necesariamente movimientos continuos.</li> </ul>
<p><b>Sujeto 2</b></p> <ul style="list-style-type: none"> <li>- Dice que definitivamente la herramienta produce sonidos a los que él no podría haber llegado por su cuenta.</li> <li>- Afirma que necesitaría de mucho tiempo para poder llegar a los sonidos que consiguió con Octopus.</li> <li>- Añade que los movimientos extremadamente rápidos o lentos que permite Octopus no pueden ser replicados al operar físicamente el instrumento.</li> </ul>	<ul style="list-style-type: none"> <li>- No encontró mayores aspectos negativos pero sí tuvo numerosas recomendaciones.</li> <li>- <b>Recomienda añadir una función que permita guardar un <i>preset</i> de la configuración establecida</b>, cosa de poder volver abrir la sesión y que esté todo en el mismo lugar.</li> <li>- A modo de complemento con la distribución Radar-256, recomienda añadir una especie de libreta en la cual registrar las combinaciones que considera interesantes.</li> <li>- A partir de lo anterior recomienda añadir una función para poder cargar las distribuciones con un conjunto de combinaciones previamente ingresadas por el usuario.</li> <li>- <b>Le gustaría que la herramienta permitiese controlar más de ocho parámetros</b>, argumenta que en algún momento quiso mapear la envolvente del filtro y que con eso ya ocupaba la mitad de los brazos.</li> <li>- <b>Le parecería interesante que Octopus permitiese al usuario crear sus propias distribuciones.</b></li> </ul>
<p><b>Sujeto 3</b></p> <ul style="list-style-type: none"> <li>- Considera que quizás sí podría haber llegado a los timbres que obtuvo con Octopus, sólo que con <b>muchas horas de trabajo</b>.</li> <li>- Señala que lo bueno de la herramienta es que justamente todas esas horas de trabajo pueden ser exploradas con el movimiento del mouse</li> </ul>	<ul style="list-style-type: none"> <li>- Encuentra engorroso el proceso de mapear los controles, sugiere encontrar maneras de simplificarlos.</li> <li>- Así mismo, <b>considera que es demasiado el tiempo que toma la configuración inicial de la herramienta.</b></li> <li>- <b>Se encontró con combinaciones que no emitían sonido, considera que es un problema asociado a los rangos que escogió para ciertos parámetros.</b></li> <li>- Sugiere buscar maneras de que la herramienta reconozca los parámetros que están mapeados y que limite el rango de operación de aquellos que pueden provocar redundancias y/o combinaciones que no suenan.</li> <li>- Sugiere mejorar el diseño gráfico de la interfaz.</li> <li>- <b>Sugiere agregar una función para que el usuario pueda crear sus propias distribuciones.</b></li> <li>- Sugiere que la herramienta <b>proponga plantillas de parámetros</b> para mapear, es decir, que te invite a mapear un conjunto de parámetros en específico.</li> </ul>
<p><b>Sujeto 4</b></p> <ul style="list-style-type: none"> <li>- Afirma que al haber partido de un <i>preset</i> que él conoce, algunos de los timbres que resultaron le parecían familiares.</li> <li>- También señala que habían muchos otros timbres que le parecían completamente lejanos al <i>preset</i> del que partió, y que incluso le sonaban parecidos a otros de los <i>presets</i> que vienen en su instrumento.</li> <li>- A partir de lo anterior, siente que el ejercicio de utilizar Octopus le sirvió para darse cuenta del potencial que tiene el motor de síntesis de su instrumento.</li> </ul>	<ul style="list-style-type: none"> <li>- Considera que el prototipo está diseñado de manera tal que no permite un trabajo continuado. <b>Recomienda implementar una función para guardar la sesión de trabajo y así poder retomarla después.</b> Añade que de esa manera Octopus se vuelva más una herramienta.</li> <li>- En un principio no le gustó el exceso de aleatoriedad que presenta el prototipo</li> <li>- Dado lo anterior, <b>recomienda potenciar aquellas funcionalidades que permiten tener más control sobre los resultados</b>, tales como la función de guardar combinaciones de usuario y los algoritmos para administrar las combinaciones dispuestas en el plano.</li> <li>- Afirma que puede llegar a ser sobrecogedora la inmensidad de posibilidades que se le presentan.</li> <li>- Dice que el lenguaje excesivamente matemático de la herramienta podría ser un problema para algunas personas.</li> <li>- Recomienda <b>encontrar la manera de que la herramienta pueda potenciar el entendimiento del usuario sobre su instrumento</b> y que no sea simplemente una caja negra que hace todo el trabajo por él.</li> </ul>

Tabla 7: Respuestas de los entrevistados

## **7.3. Análisis de los resultados**

De las entrevistas realizadas a los sujetos de prueba surgieron una serie de ideas y reflexiones que resultan de mucha utilidad a la hora de evaluar el desempeño de la herramienta. La experiencia de los usuarios permite además contrastar su percepción de la herramienta con las características que el autor pretendía implementar en ella. A continuación se presentan las ideas y reflexiones más relevantes para esta investigación.

### **7.3.1. Impresiones generales de Octopus**

Si se analizan detalladamente las respuestas de los sujetos de prueba, se puede observar que existen algunas ideas comunes entre ellos. La primera de ellas, mencionada por los cuatro entrevistados, es que en comparación con el método tradicional Octopus es percibida como una herramienta que proporciona un significativo ahorro de tiempo a la hora de trabajar con el instrumento electrónico. Ahora bien, cabe precisar que el ahorro de tiempo del que hablan los participantes, más que estar relacionado con demorarse menos en alcanzar una meta o sonido satisfactorio, está relacionado con la idea de que la exploración de sonidos se vuelve más fluida, es decir, que se pueden oír más timbres en menos tiempo. El hecho de poder cambiar el timbre simplemente con el movimiento del mouse fue en la mayoría de los casos percibido como algo positivo, sin embargo, el sujeto n° 4 señaló en la Tabla 6 que, a diferencia de cuando utiliza el método tradicional, con Octopus le cuesta tan poco trabajo llegar a un sonido que al mismo tiempo le es muy fácil desecharlo, relacionando esta característica de la herramienta con la “cultura de la inmediatez”.

El ahorro de tiempo que significa Octopus a la hora de explorar timbres se ve contrarrestado por los tiempos que toman otros procesos propios de la herramienta.

La mayoría de los entrevistados señaló en la Tabla 5 que, a pesar de haber sido asistidos por el autor durante la prueba, Octopus presenta una curva de aprendizaje lenta, es decir, que se requiere de un intervalo de tiempo no menor para poder entender de qué se trata la herramienta, qué hace cada uno de sus módulos y cómo se configura. Tal es la situación que el sujeto n° 2 comentó que “definitivamente se debe tener un manual para iniciarse”. Esta percepción de los usuarios podría haber sido incluso peor de no haberse prestado ayuda y consejos a los participantes durante la prueba, por lo que definitivamente es un aspecto a mejorar. Otro punto que podría estar sumando dificultad a la hora de introducirse a la herramienta es que, como señala el participante n°4 en la Tabla 5, existe una predominancia del lenguaje matemático por sobre un lenguaje musical/creativo, cosa que puede generar cierta distancia con el usuario. A pesar de lo anteriormente expuesto, los participantes también fueron bastante explícitos al señalar que una vez tuvieron una noción general de lo que hace cada elemento, Octopus se vuelve bastante menos intimidante y comienza a percibirse como una herramienta atractiva.

Por otro lado, y como se puede ver en la Tabla 5, los dos participantes que trabajaron con *software* sintieron que el proceso de mapeo de controles y de selección de rangos es engorroso y toma demasiado tiempo, lo que puede deberse al hecho de que al trabajar con *software* necesariamente se tiene que tener abierto Max, donde corre Octopus; Ableton Live, donde corre el instrumento; y un parche de *Max for Live* para comunicar Octopus con el instrumento. A raíz de esto último es que el sujeto n° 1 declaró que le resultó incómodo tener que cambiar de ventana constantemente para poder mapear los controles de su instrumento. Los problemas de utilizar Octopus con instrumentos de *software* no sólo fueron percibidos por los entrevistados, sino que también por el autor. A la hora de alistar todo lo necesario para tomar las pruebas, se hizo evidente que trabajar con *hardware* era notablemente más fácil que

configurar los computadores de los entrevistados que usaron *software*, principalmente debido a que en el primer caso se reduce la posibilidad de error al prescindir por completo del uso de Ableton Live y Max for Live. A raíz de lo anterior es que en etapas futuras del desarrollo de esta herramienta deben solucionarse los problemas ya descritos.

En cuanto a la cantidad de parámetros que puede manejar la herramienta, los sujetos n° 3 y n°4 fueron explícitos al señalar que ocho controles era una buena limitante y un número adecuado (ver Tabla 5). Por su parte, el sujeto n° 2 propuso como recomendación la posibilidad de aumentar este número (ver Tabla 7), argumentando que en un punto de la prueba quiso controlar la envolvente de un filtro y que sólo eso ocupaba la mitad de los brazos disponibles. Por otro lado, el sujeto n° 1 no emitió comentarios al respecto, sin embargo, se observó durante la prueba que tuvo dificultades para terminar de ocupar los ocho brazos con algún parámetro, por lo que se puede especular que quizás fueron muchos para él. Al haber tanta disparidad en la percepción de cada participante se vuelve difícil tomar alguna determinación, y por lo mismo, es conveniente recordar porqué se decidió que ocho brazos era un número adecuado. En el capítulo “Principios de diseño” se expone que la cantidad de brazos es un factor clave en la cantidad de combinaciones que se generan, por lo que cualquiera sea el número de brazos que se escoja debe producir una cantidad de combinaciones que sea manejable desde el punto de vista de los recursos del usuario (Goudeseune 2002). Más sobre esto último será discutido en la sección de proyecciones.

Otra cosa que fue mencionada por los cuatro participantes es que, sin importar que hayan mapeado los controles y configurado los rangos de forma completamente consciente, Octopus les mostró timbres que describieron como inesperados (ver Tabla 5 y Tabla 6). Así mismo, cuando se les preguntó si es que esos sonidos se

parecen a los que llegan cuando no utilizan la herramienta las respuestas tendieron a ser negativas. A pesar de lo anterior, los participantes no sintieron que ello se debía a un desconocimiento técnico de su parte y en general todos intentaron formular una explicación para el fenómeno. El participante n° 1 señaló (Tabla 6) que él nunca había trabajado con la automatización de ocho controles de forma simultánea, y que justamente esa era la explicación principal. Por su parte, los participantes n° 2 y n° 3 lo atribuyen a una cuestión de tiempo, ya que aseguran que sin Octopus tendrían que haber invertido una gran cantidad de horas para poder haber llegado a resultados similares (ver Tabla 7). El sujeto n° 2 agrega además que la precisión que maneja la máquina a la hora de mover los parámetros a velocidades muy rápidas o muy lentas es irrealizable con el cuerpo humano, lo que está muy relacionado con el concepto de los recursos del usuario (Goudeseune 2002). Por último, el participante n° 4 considera que la razón por la cual existe una distancia entre los sonidos que habitualmente diseña y los que alcanzó con Octopus se debe a que esta herramienta reduce el componente analítico y añade demasiada aleatoriedad (Tabla 5). Un análisis en profundidad respecto de la cualidad sorpresiva de los sonidos que genera Octopus y su relación con los enfoques analíticos y holísticos del *mapping* va a ser discutida más adelante.

Durante las pruebas, tres de las cuatro personas entrevistadas obtuvo en algún momento combinaciones vértice que no emitían sonido alguno, situación que experimentaron particularmente los sujetos n° 1(*software*), n° 3 (*software*) y n° 4 (*hardware*). Cuando se tomó conocimiento de esta situación se recomendó a los sujetos revisar los rangos establecidos para descartar que haya habido algún parámetro funcionando como compuerta para el sonido, sin embargo, en general no se pudo encontrar el origen del problema. Cabe destacar que la situación fue notablemente más problemática para los usuarios que utilizaron instrumentos de

*software*. A raíz de lo anterior es que se vuelve necesario encontrar una o varias maneras de que Octopus pueda dar luces al usuario de las posibles causas del problema. Esto podría ser a través de alguna función en la que el usuario pueda indicar a la herramienta una muestra de combinaciones que no suenan y un grupo de combinaciones que sí, para que luego Octopus pueda encontrar el o los parámetros que están generando problemas.

Un punto que fue mencionado por algunos de los entrevistados es la necesidad de que Octopus permita llevar a cabo un trabajo continuado (ver Tabla 7). Más concretamente, dos participantes expresaron la necesidad de poder guardar sesiones de trabajo en Octopus, es decir, que se pueda conservar la información de controles mapeados, rangos seleccionados, la distribución escogida, las combinaciones presentes en el interpolador e incluso la configuración general del sintetizador que se estaba ocupando. Con esto, los entrevistados sostienen que podrían trabajar distintas sesiones de Octopus a lo largo de diferentes jornadas de trabajo, lo que les ayudaría a obtener configuraciones más pulidas y que les serían útiles en el largo plazo. Actualmente el prototipo no permite realizar ninguna de las cosas mencionadas anteriormente y cada vez que se cierra se pierde toda la configuración establecida. Esto último es definitivamente una función que se debe implementar para asegurar que Octopus pueda suplir las necesidades de los usuarios.

En un punto de la entrevista, el sujeto n° 4 recomendó encontrar la manera de que Octopus pueda potenciar el entendimiento del usuario sobre su propio instrumento (Tabla 7), ya que en el estado actual del prototipo él considera que la herramienta funciona como una especie de caja negra. Sin embargo, a pesar de lo señalado por esta persona, Octopus cuenta con ciertos elementos de monitoreo que sí dan luces de lo que está ocurriendo a nivel interno del instrumento. Ejemplo de lo anterior es que la herramienta muestra en tiempo real el valor instantáneo de cada uno de los

parámetros controlados, cosa que fue valorada positivamente por la mayoría de los entrevistados, incluyendo al sujeto n° 4. Además, durante la realización de las pruebas se pudo observar que en general los entrevistados sí vieron enriquecido su conocimiento del instrumento en distintos aspectos. El mismo sujeto n° 4 señaló que el ejercicio de haber utilizado Octopus le permitió tener una idea del potencial que tiene el motor de síntesis de su instrumento (Tabla 7). A un nivel más técnico, durante la configuración de los brazos el sujeto n° 2 se dio cuenta que muchos *sliders* de su instrumento estaban defectuosos ya que no estaban enviando mensajes MIDI. Así mismo, el sujeto n° 1 se enteró que no todas los parámetros de Operator se pueden controlar con señales MIDI, ya que algunos de ellos están esperando recibir valores flotantes entre cero y uno. De todas formas las observaciones recién expuestas no constituyen motivos suficientes para creer que Octopus mejora el entendimiento de los usuarios sobre su instrumento, lo que además se ve atravesado por el hecho de que las personas sólo tuvieron contacto con la herramienta durante una sesión de aproximadamente una hora.

A modo de cierre de esta subsección del análisis, cabe destacar que en general todas las pruebas de usuario fueron realizadas con éxito, los entrevistados pudieron comprender rápidamente la idea detrás de Octopus y los alcances que tiene la herramienta. Así mismo, las respuestas de todos los participantes contenían información valiosa que resultó útil para poder evaluar el desempeño de la herramienta así como para poder trazar los próximos pasos a seguir en el desarrollo de este proyecto.

### **7.3.2. Sobre la incorporación de Octopus en la práctica musical**

Uno de los primeros resultados interesantes respecto de la incorporación de Octopus en la práctica musical es que todos los participantes consideraron que la herramienta sí tiene cabida en su forma de trabajar (Tabla 6). Ahora bien, este hallazgo debe ser

tratado con cautela puesto que no todas las personas señalaron que incluirían la herramienta de la misma manera, en la misma medida o en su estado actual (refiriéndose al prototipo). Algunas de las ideas que surgieron fueron utilizar Octopus exclusivamente para etapas de experimentación previas a la gestación de una obra, usarlo como herramienta compositiva, sacar provecho de sus mecanismos de secuenciación, utilizarlo para improvisaciones y shows en vivo, usarlo para diseñar texturas o sonidos complejos, utilizarlo en conjunto con otros instrumentos y poder utilizarlo de forma continuada en diferentes sesiones. La variedad de enfoques que aparecieron tiene directa relación con que cada participante declaró tener una forma particular de operar con su instrumento, cosa que se debe tener siempre en consideración para poder realizar una correcta contextualización y análisis de sus respuestas.

Por otro lado, cuando se le preguntó a los usuarios por la similitud entre la interacción que ofrece Octopus y su flujo normal de trabajo las respuestas fueron dispares (ver Tabla 6). Mientras que los usuarios 1 y 2 identificaron en Octopus ciertos atributos que están presentes en su forma habitual de trabajar, los usuarios 3 y 4 apuntaron a que existe cierta distancia entre la herramienta y sus propios procesos. Más concretamente, cuando se le preguntó al sujeto 1 por Octopus, a pesar de que comentó que él “jamás trabaja automatizando ocho parámetros al mismo tiempo”, sí reconoció que la idea de secuenciar y automatizar está tan presente en Octopus como en su método habitual de trabajo. Cuando se consultó al sujeto 2, este declaró que aprovecha las prestaciones de su instrumento para mover múltiples controles y así improvisar efectos y “sonidos raros” para situaciones en vivo, e identificó que con Octopus se puede realizar esa misma tarea. Por otro lado, el sujeto 3 declaró que la distancia entre Octopus y su flujo de trabajo radica en que habitualmente le incomoda no poder predecir el resultado al que va a llegar. Así

mismo, el sujeto 4 fue categórico en decir que Octopus “se parece en nada” a su forma habitual de diseñar sonidos, añadiendo que él tiene un método analítico y se esfuerza por conscientemente manipular los parámetros de su instrumento para llegar a un objetivo claro, y que la herramienta “le arrebató eso de las manos”; agrega también que se vuelve imposible llevar un trabajo continuado con el prototipo ya que este no permite guardar sesiones. Ahora bien, resulta interesante que, a pesar de la distancia que perciben con Octopus, los sujetos 3 y 4 hayan respondido afirmativamente a la idea de incorporar la herramienta en su práctica musical. Más detalles sobre esto último serán discutidos más adelante.

En cuanto al uso que darían a Octopus, evidentemente las respuestas fueron variadas y están fuertemente influenciadas por el uso habitual que da cada sujeto a su instrumento (ver Tabla 6). Más concretamente, el sujeto 1, quien ya había señalado la importancia que tiene para él el uso de secuencias, automatizaciones y LFO, declaró que le gustaría aprovechar la posibilidad de hacer automatizaciones “muy complejas” y que le parecen útiles todas las funciones asociadas a la secuenciación de movimientos. El sujeto 2 quien declaró que en general utiliza su instrumento para tocar en vivo e improvisar con el diseño de sonido, señaló que podría ocupar Octopus para tener más y nuevas alternativas a mano a la hora de improvisar en vivo. Por otro lado, el sujeto 3 señaló que debido a ciertas limitaciones de conocimiento no suele explotar la dimensión tímbrica de su instrumento, y que por lo mismo Octopus le podría servir para romper esa barrera y expandir su método de trabajo al enfocarse también en añadir nuevas texturas a sus composiciones. El sujeto 4, quien declaró en numerosas ocasiones tener un enfoque analítico y racional a la hora de diseñar sonidos, advirtió que necesita un tiempo para aprender a contextualizarla, pero que con el estado actual del prototipo

pensaría únicamente en utilizarla con fines exploratorios en las etapas “primitivas” de la creación de una pieza. Sin embargo, el sujeto 4 añade que si la herramienta toma una dirección en la que se entregue más control analítico al usuario, y en la que se reduzca el exceso de aleatoriedad, sí pensaría en utilizarla en etapas más composicionales de la creación de una pieza.

Las respuestas de los sujetos a la pregunta por el uso que darían a Octopus arrojan resultados muy interesantes ya que, al contrastarlas con el flujo de trabajo que cada uno describió como propio, se puede entrever la relación y compromiso que tiene cada sujeto con su propio proceso creativo. Para desarrollar esta idea, considere las respuestas de los sujetos 1 y 2. Ambos declararon que la forma en la que utilizarían Octopus es justamente como un **complemento** a una práctica que ellos ya venían realizando con su instrumento. En el caso del sujeto 1 ocuparía la herramienta para poder automatizar de otras maneras y en el caso del sujeto 2 la ocuparía para tener más alternativas para improvisar en vivo, es decir, ambos deciden utilizar Octopus de manera tal que la relación que tiene cada uno con su instrumento no se ve alterada sustancialmente. Algo similar sucede con el sujeto 4, quien sostiene que si la herramienta fuera más compatible con el método racional/analítico que él dice tener, muy probablemente le daría más espacio en su práctica musical. En otras palabras, la manera en la que está diseñado el prototipo obliga al sujeto 4 a moverse de su forma habitual de trabajar con el instrumento, y es justamente eso lo que le incomoda. Por otro lado, el sujeto 3 fue la única persona que expresó que Octopus podría significar una modificación en su proceso creativo y que eso le podría servir para explorar ciertas aristas que hasta el momento no había atendido. Todo lo anterior resulta información útil a la hora de reflexionar en torno a qué es lo que valora una persona de una herramienta creativa, incluso más allá de Octopus. Mientras que la idea de una herramienta que transforme por completo la forma de

trabajar pareciera *a priori* ser adecuada, los resultados obtenidos en esta investigación muestran que la mayoría de los sujetos prefieren una herramienta que funcione como un complemento a su forma habitual de operar. Esto podría deberse en parte al hecho de que se entrevistó a personas que llevan un largo tiempo trabajando con sus instrumentos por lo que es probable que se hayan generado una costumbre.

### **7.3.3. El enfoque divergente en Octopus**

En el capítulo de marco teórico se expuso que hay autores que establecen una relación entre algunos enfoques de *mapping* y las distintas etapas del proceso creativo. En ese sentido, la estrategia *uno-a-uno* ha sido ampliamente relacionada con el pensamiento de tipo convergente o analítico, y estrategias como *pocos-a-varios* o *varios-a-varios* han sido relacionadas al pensamiento divergente u holístico (Dixon, Tubb 2014; Le Vaillant, Dutoit, Giot 2020). A grandes rasgos, la diferencia entre estos dos tipos de pensamiento radica en que el primero busca llegar a un resultado determinado por medio de pasos lógicos mientras que el segundo tiene más que ver con la idea de la exploración y la experimentación. Las ideas expuestas anteriormente tienen una importante relación con los resultados obtenidos de las pruebas de usuario, particularmente con el lenguaje utilizado por los entrevistados para describir los timbres alcanzados con Octopus.

Cuando se preguntó a los usuarios por los timbres a los que llegaron durante la experimentación aparecieron entre otros adjetivos las palabras “inesperados”, “no contemplados”, “sorpresivos” e “impensados” (ver Tabla 6). Todas ellas hacen referencia a la idea de que se rompieron expectativas, en el sentido de que dado un instrumento conocido, unos controles y unos rangos seleccionados de forma consciente, los usuarios se hicieron una idea de cómo podrían ser los sonidos resultantes, la cual terminó siendo refutada por los hechos. Esto último a simple

vista puede no sugerir nada, sin embargo, cuando esa información se cruza con que un entrevistado respondió que “me gusta que la máquina toma decisiones por mí y yo sólo tengo que escuchar” (Tabla 5) o con que otro sujeto señaló que la herramienta “se salta el proceso racional de preguntarse qué perilla estoy moviendo o en qué valor está, y pasa a ser sólo me gusta o no me gusta”(Tabla 5), empieza a aparecer algo interesante. Las palabras con las que los usuarios describieron los timbres alcanzados encierran la idea de que sus expectativas, sostenidas por el pensamiento racional/analítico, fueron quebradas por la forma en la que opera la herramienta. ¿Y cuál es esa forma? pues con una estrategia de *mapping dos-a-varios*. Esto último es muy interesante ya que los resultados obtenidos de alguna manera reafirman lo que ha sido planteado en la literatura, esto es, la idea de que las estrategias *dos-a-varios* son mucho más cercanas a un tipo de pensamiento divergente/holístico/exploratorio que a un pensamiento del tipo analítico.

Por otra parte, si se considera el caso del sujeto n° 3 aparece la pregunta por las ventajas que pueden tener los enfoques divergentes por sobre los analíticos a la hora de diseñar herramientas creativas. El entrevistado mencionó en un inicio que al trabajar con su instrumento siempre tiende a utilizar los mismos parámetros debido a que conoce su funcionamiento, añadiendo que descarta utilizar ciertas configuraciones con dichos parámetros ya que cree conocer el resultado tímbrico *a priori* (Tabla 4). Cuando tuvo que configurar los brazos de Octopus el sujeto señaló explícitamente que iba a mapear los parámetros que siempre utiliza, sin embargo, a la hora de preguntarle por los sonidos obtenidos señaló estar sorprendido. Agregó que la herramienta le presentó combinaciones entre parámetros que él no sabía que podían sonar bien, y que de hecho formaban parte de aquellas configuraciones que él normalmente descarta utilizar (Tabla 5 y Tabla 6). A raíz de lo anterior es que cabe la pregunta respecto de qué tanto puede estar limitando el pensamiento

analítico/racional a la exploración creativa de sonidos, y cómo ello puede ser evitado diseñando herramientas con otros enfoques. Evidentemente la respuesta a esta interrogante va a variar dependiendo de cuál es la finalidad última del uso de un instrumento, sin embargo, es a lo menos atendible el hecho de que esta persona se haya estado perdiendo de ciertos sonidos sólo por un “exceso de pensamiento racional”.

## 8. Discusión y conclusiones

En este capítulo se busca relacionar los resultados obtenidos del proceso de investigación tanto con los objetivos de investigación como con las ideas expuestas en el capítulo de marco teórico, para así finalmente poder dar paso a una discusión en torno a los alcances que tuvo la investigación y la dirección que puede tomar en el futuro.

### 8.1. Discusión

#### 8.1.1. Sobre el desarrollo y logro de los objetivos

Como fue esbozado en el capítulo de método, el desarrollo de esta investigación puede dividirse en dos grandes momentos: la implementación de Octopus, que incluye tanto la programación del código como el proceso reflexivo que dio origen a su diseño; y la fase de prueba con usuarios, la cual tuvo lugar una vez estuvo listo el prototipo. Ambos acontecimientos en su conjunto debieran haber permitido que el Objetivo General (ver sección 1.1.1) de esta investigación pudiese alcanzarse con éxito. En cuanto a los Objetivos específicos (ver sección 1.1.2), los primeros dos están claramente relacionados con la primera parte del desarrollo mientras que el último está exclusivamente relacionado con la fase de pruebas. Cómo y hasta qué punto fueron alcanzados cada uno de los objetivos será tema de discusión en los párrafos venideros.

Para cumplir con el primer objetivo se llevó a cabo un proceso de investigación en el que se profundizó en áreas como el diseño de instrumentos electrónicos y *mapping*, orientándose principalmente a las diferentes estrategias de *mapping* y su impacto sobre el comportamiento de los IME. Fruto de lo anterior es que se identificaron y describieron tres enfoques distintos a la hora de abordar el problema del *mapping* (ver sección 2.3.3). A partir de esto último es que se levantó el problema de

investigación y se tomó la decisión de abrir un camino en dirección a utilizar mapeos *pocos-a-varios* para controlar IME. Más concretamente, para el diseño de la estrategia de *mapping* se revisaron en profundidad trabajos recientes que estudian la utilización de mapeos del tipo *pocos-a-varios* implementados a través de interfaces bidimensionales (Gibson, Polfreman 2019; Le Vaillant, Dutoit, Giot 2020; Dixon, Tubb 2014). Así mismo, se indagó sobre las distintas interfaces bidimensionales que pudieran servir para el propósito de la investigación, y se optó por utilizar aquellas conocidas como interpoladores gráficos (Gibson, Polfreman 2019). Paralelamente, en el capítulo de referentes (ver capítulo 3) se recopilaron diferentes herramientas que incluyen mapeos del tipo *pocos-a-varios* y que sirvieron de inspiración para el diseño de la estrategia de mapeo de esta investigación. Cabe destacar que un hito clave para el diseño de esta estrategia fue la aparición del concepto de ‘espacio combinatorio’ (Dixon, Tubb 2014), idea sobre la cual se construyó la lógica que operaría en Octopus y que resultaría ser su característica más distintiva.

En cuanto al cumplimiento del primer objetivo específico, en el capítulo 5 se expone detalladamente el proceso de diseño de la estrategia de *mapping*, la cual incluye efectivamente la utilización de un mapeo del tipo *dos-a-varios* y el uso de técnicas de reducción de dimensiones, sin embargo, el objetivo plantea que se pueda condensar **todo** el espacio de parámetros del instrumento y Octopus sólo permite trabajar con ocho parámetros, por lo que se podría decir que el objetivo está medianamente cumplido. A pesar de esto, en etapas de prueba los usuarios consideraron positivo que la herramienta estuviera limitada a ocho parámetros.

Una vez listo lo anterior se dio inicio a la etapa con la que se alcanzaría el segundo objetivo de investigación, es decir, comenzó la fase de implementación de la estrategia de mapeo. Para ello se utilizó principalmente el ambiente de programación Max/MSP y el sintetizador Korg Minilogue como el instrumento

electrónico sobre el cual se realizarían las pruebas del código, siendo el protocolo MIDI el canal de comunicación entre el computador y el instrumento. Cabe destacar que en un inicio se comenzó a programar Octopus en Max for Live, sin embargo, luego de una serie de problemas técnicos y de la aparición de limitaciones se optó por utilizar directamente Max/MSP.

Para facilitar la programación, una de las primeras tareas que se realizaron fue el diseño de un diagrama de flujo que representa la estrategia de mapeo de Octopus (ver Figura 17: Diagrama en bloque del primer prototipo de Octopus (acabado)). Con el diagrama listo, la programación se realizó parte por parte con pequeñas sesiones de prueba del código. Lo primero que se implementó fue el módulo de brazos en conjunto con una versión primitiva del interpolador gráfico. Una vez probado, se prosiguió con el mejoramiento del módulo del interpolador gráfico y finalmente se implementó el módulo de distribuciones seguido del módulo de trayectorias. Cabe destacar que la programación del módulo de distribuciones fue la etapa en la que más tiempo se invirtió, debido a que cada una de las distribuciones fue diseñada desde cero y además debían ser testeadas para conocer su comportamiento y evaluar si es que era adecuado incluirlas en el prototipo. Además de lo anterior, algunas de las pruebas realizadas se documentaron en video y están incluidas en el Anexo B. Por último, y considerando lo expuesto anteriormente, es posible afirmar que el nivel de logro alcanzado para el segundo objetivo específico fue total.

Por último, para cumplir con el tercer objetivo específico se debió realizar una serie de tareas más numerosas y variopintas que en las etapas anteriores. Una vez estuvo listo el prototipo se debía encontrar la manera de evaluar las capacidades de la herramienta. Para ello se realizó una pequeña investigación relacionada a los métodos de evaluación de instrumentos musicales electrónicos, revisándose principalmente trabajos publicados en la conferencia NIME. Además de lo anterior,

en conversaciones con el profesor guía se tomó la determinación de que la manera adecuada de evaluar el prototipo era con una entrevista semiestructurada. Paralelamente, fue necesario determinar de qué manera se iban a filtrar y seleccionar las personas que iban a participar del testeo de Octopus, ya que, como se establece en el objetivo, debe tratarse de personas que cuenten con experiencia previa en el uso de IME. Se tomó la decisión de incluir personas que tengan **al menos** un año de experiencia utilizando el mismo instrumento, independiente de su formación musical. Respecto de la entrevista en sí, se visitó a cada participante de forma individual y se aplicó la pauta de preguntas incluidas en el Anexo A. Cada experiencia fue registrada en audio y luego se transcribieron en una tabla para su posterior análisis.

En cuanto al nivel de logro del tercer objetivo, ciertamente existe una discordancia entre la manera en la que este está planteado y la información que se pudo obtener con la entrevista semiestructurada. El problema de haber pretendido “**evaluar las capacidades creativas, exploratorias y renovadoras de la herramienta (...)**” radica en que esas capacidades no están contenidas necesariamente dentro de la herramienta y dependen más bien de la manera en la que los usuarios la utilizan. Ahora bien, dejando de lado el problema de redacción, los resultados efectivamente muestran la percepción que tuvieron los participantes de la mayoría de los aspectos mencionados en el objetivo. Se logró determinar que efectivamente la herramienta sugiere y proporciona una interacción que promueve la exploración creativa de timbres. Así mismo, se pudo determinar que en general Octopus **no** renueva la forma que tienen los usuarios de pensar y operar con su instrumento, sino que más bien tiende a ser visto como un complemento al método habitual de trabajo. Por último, en cuanto a las capacidades creativas, la información presente en las respuestas de los usuarios no fue lo suficientemente contundente como para poder

determinar algo al respecto. Es por todo lo anterior que el tercer objetivo fue medianamente alcanzado.

### **8.1.2. Sobre aspectos generales de Octopus y de la investigación**

Además de los objetivos de investigación, es necesario señalar algunos detalles sobre la implementación de este primer prototipo. En primer lugar, aunque inicialmente se pretendía que Octopus fuera una herramienta capaz de controlar **cualquier** instrumento electrónico, las limitaciones técnicas surgidas durante su desarrollo han llevado a que el prototipo actualmente sea compatible únicamente con:

- Instrumentos electrónicos de *hardware* que dispongan de puertos de entrada y salida MIDI (USB o conector DIN de 5 pines) y que además permitan ajustar sus parámetros a través de mensajes MIDI de tipo *Control Change*.
- Instrumentos electrónicos de *software* que se puedan ejecutar en *Ableton Live* y cuyos parámetros de entrada operen dentro del rango del protocolo MIDI, es decir, valores enteros entre 0 y 127.

En ambos casos, la naturaleza del instrumento no debería representar un obstáculo para la aplicación de Octopus. Sin embargo, durante las pruebas se observó que utilizar la herramienta con instrumentos de *software* resultaba más difícil en comparación con los instrumentos de *hardware*. Los entrevistados señalaron una complicación principal: al utilizar Octopus con un IME de *software* abierto en Live, debían cambiar constantemente de ventana entre aplicaciones para ajustar el sistema, lo cual resultaba incómodo al no estar todo unificado en una interfaz central. Por lo tanto, en futuras versiones de Octopus se debe mejorar la integración con Ableton Live. Además, sería conveniente añadir una tercera alternativa aparte de la opción de trabajar con *hardware* y Ableton. Esta alternativa implicaría el uso

del objeto *vst~*<sup>38</sup>, que permitiría ejecutar un instrumento de software dentro de la interfaz de Octopus, eliminando la necesidad de utilizar un programa adicional como Live. Esto ahorraría recursos computacionales y eliminaría la necesidad de cambiar entre ventanas.

Otro problema identificado durante la etapa de pruebas fue que la mayoría de los usuarios se encontraron con un gran número de combinaciones vértice que no producían ningún sonido. Esto fue sorprendente para todos a quienes les sucedió, y en general no se le encontró explicación. Es posible que se trate de combinaciones que asignen valores extremos a parámetros que tienen una gran influencia en el sonido, como asignar un valor de 0 al filtro pasa-bajos. Sin embargo, este fenómeno ocurrió sin que los usuarios tuvieran la intención, lo que refuerza el carácter sorprendente de esta herramienta. En futuras versiones de Octopus, es necesario abordar este problema y proporcionar soluciones. Una posible solución sería implementar un algoritmo que, con la capacitación previa del usuario, pueda comparar el contenido numérico de las combinaciones que no producen sonido con aquellas que sí lo hacen, y determinar así qué parámetros están generando el problema. De esta manera, una vez que el usuario reciba esta información, podrá reemplazar dichos parámetros o restablecer sus rangos a valores adecuados.

Sumado a lo anterior, según la información recopilada durante la evaluación del prototipo, se observa que Octopus presenta una curva de aprendizaje pronunciada. Esto se debe a que los usuarios deben pasar por un proceso que consiste inicialmente en comprender la lógica de la herramienta para luego aprender a utilizarla. Si bien

---

<sup>38</sup> Más detalles de este objeto en <https://docs.cycling74.com/max7/refpages/vst~> (*vst~ Object Reference - Max 7 Documentation* sin fecha)

se podría argumentar que este proceso es similar al que se encuentra en otras herramientas de audio, como ecualizadores, sintetizadores o mezcladoras, la diferencia radica en que estas últimas son herramientas ampliamente conocidas y con una presencia notable a lo largo de los años. Octopus, por otro lado, es un dispositivo nuevo y sin antecedentes demasiado conocidos tanto en términos de lógica como de interfaz. Por esta razón, es necesario que los usuarios tengan acceso a algún tipo de guía o instructivo que les permita iniciarse en Octopus y profundizar más en su utilización. A partir de todo lo anterior, es que al evaluar la usabilidad de Octopus la respuesta es mixta. Si bien resulta una herramienta efectiva para la exploración sonora, también sucede que la experiencia del usuario que se inicia en ella podría mejorar considerablemente.

Por otro lado, al compararlo con el estado del arte, se identifica un aspecto a mejorar en la implementación de Octopus relacionado con la incorporación de algoritmos de optimización y técnicas de *machine learning*. Estos elementos no están presentes en el primer prototipo, pero podrían resultar altamente beneficiosos para perfeccionar los diferentes procesos de la herramienta. Por ejemplo, el módulo de distribuciones podría beneficiarse de la aplicación de un algoritmo que optimice la posición, tamaño y cantidad de nodos, teniendo en cuenta criterios como maximizar las combinaciones disponibles en el plano, disponer combinaciones similares o diferentes, entre otros. Además, la aplicación de técnicas de *machine learning* podría permitir el reconocimiento de gestos del usuario, el aprendizaje de los parámetros que el usuario mapea con regularidad y generar recomendaciones personalizadas, reforzar la aparición de ciertas combinaciones o distribuciones, así como recomendar combinaciones que podrían eventualmente gustarle al usuario, entre otras posibilidades.

En cuanto al proceso de investigación, al analizarlo en retrospectiva se vuelve claro que existe una serie de temas que, de haber sido replanteados o incluidos, podrían haber enriquecido los resultados, por lo que es adecuado tomarlos en cuenta para plantear los caminos que puede tomar esta investigación a futuro. Una de las primeras posibilidades es profundizar en el estudio de la dimensión cognitiva del proceso creativo y su posible relación con las interfaces de usuario. Dado que esta investigación se basó en literatura que entiende el proceso creativo como una simple dicotomía entre el pensamiento convergente y el divergente, resultaría interesante explorar modelos cognitivos más complejos y actualizados. Así, se podría evaluar su viabilidad para la construcción de nuevas estrategias de *mapping* que sean compatibles con ellos. Al enfocar la investigación en la dimensión cognitiva se abre la posibilidad de abarcar temas como la capacidad de atención, el *flow state* e incluso la neurodiversidad, resultando así en una versión de Octopus más adaptada al funcionamiento estructural del cerebro humano.

Sumado a lo anterior, la investigación que dio origen a Octopus se centró principalmente en aspectos técnicos del diseño de instrumentos electrónicos, dejando de lado enfoques provenientes de otras disciplinas pertinentes al tema. La consideración de estos enfoques habría enriquecido tanto el proceso como los resultados. En el futuro, sería particularmente interesante incluir estudios del campo de la filosofía de la tecnología, como la tecnoestética o la política de la interfaz (*interface politics*). Teniendo en cuenta que Octopus es una interfaz y, en última instancia, un dispositivo tecnológico, su proceso de diseño es en sí mismo una actividad que puede ser objeto de numerosas interrogantes (Badía et al. 2016). Por ejemplo, ¿qué estamos haciendo cuando diseñamos interfaces? ¿Cuál es el papel de las interfaces en nuestra relación con la tecnología? ¿Cuáles son las implicaciones éticas o políticas de las interfaces que estamos creando? ¿Cómo afectan las interfaces

a nuestra autonomía como usuarios? ¿Cuáles son los límites de las interfaces en términos de creatividad o pensamiento crítico? Estas son solo algunas de las cuestiones que podrían abordarse.

## 8.2. Conclusiones

En este proyecto de investigación se implementó una herramienta de *software* que permite al usuario aplicar una estrategia de *mapping* del tipo dos-a-varios directamente sobre el generador de sonido de un instrumento electrónico. Los resultados obtenidos a partir del *testeo* de Octopus arrojaron luces sobre la percepción que podrían futuros usuarios así como también de la manera en la que el diseño de la herramienta se condice (o no) con lo expuesto en la literatura.

Uno de los hallazgos más interesantes es que, tal como se ha propuesto en (Dixon, Tubb 2014; Le Vaillant, Dutoit, Giot 2020), se pudo corroborar que las estrategias de *mapping* del tipo pocos-a-varios son efectivamente más adecuadas para las etapas iniciales del proceso creativo, siendo de particular ayuda para la exploración creativa de timbres musicales. Los usuarios entrevistados reportaron que Octopus les permitió llegar a timbres a los que no habrían podido llegar de otra manera. En la misma línea, todos los usuarios coincidieron en que la herramienta les permitía un significativo ahorro de tiempo ya que con ella pudieron escuchar más timbres en menor tiempo, aumentando así las posibilidades de encontrar resultados satisfactorios.

Por otro lado y contrariamente a lo que se esperaba en un principio, el hecho de que Octopus haya añadido una nueva capa de *mapping* sobre el IME de cada participante **no** modificó sustancialmente la manera en la que los usuarios se relacionaron con su instrumento. Es más, cuando se le preguntó a los usuarios por la forma en la que incluirían Octopus en práctica musical muchos de ellos dijeron que lo proyectan

como un **complemento** a su método habitual de trabajo, surgiendo así la pregunta por qué tan dispuestos están los usuarios a modificar su manera de operar con el instrumento. Lo anterior de alguna manera tensiona la idea de que cambiar la capa de *mapping* necesariamente cambia la manera que tienen los usuarios de relacionarse con su instrumento (Hunt, Wanderley, Paradis 2002). Ahora bien, esta disonancia entre los resultados y la literatura también se puede deber al hecho de que los entrevistados estuvieron en contacto con Octopus por no más de una hora, estando la posibilidad de que un trabajo continuado con la herramienta vaya acercando al usuario a otras maneras de pensar su instrumento. Además de lo anterior, es pertinente que se revisen las técnicas y metodologías empleadas para la recolección de datos relacionados con la percepción que tienen los usuarios de sus instrumentos.

Además de lo anterior, las respuestas de los entrevistados muestran que muchos de ellos, si bien valoran el enfoque exploratorio presente en Octopus, preferirían que la herramienta permitiera al mismo tiempo un control un poco más analítico del timbre. En la misma línea, muchos de ellos expresaron que les gustaría que Octopus pudiera guardar sesiones de trabajo para así desarrollar parches más personalizados y menos aleatorios. Lo anterior se condice plenamente con la idea de que, más allá de elegir un enfoque analítico o un enfoque divergente en la estrategia de *mapping*, las interfaces musicales se verían beneficiadas si es que ambos enfoques estuvieran presentes de forma complementaria (Dixon, Tubb 2014; Le Vaillant, Dutoit, Giot 2020; Gibson, Polfreman 2020b). Esto último es ciertamente la dirección en la que debería seguir el desarrollo de esta herramienta.

Otro punto importante es que las pruebas de usuario fueron realizadas casi en su totalidad con sintetizadores sustractivos (a excepción de Operator), por lo que en una segunda etapa de pruebas sería interesante incluir a más usuarios y preocuparse de que los IME utilizados sean más variados en cuanto a su algoritmo de producción

sonora, es decir, incluir sintetizadores granulares, sintetizadores FM, sintetizadores *wavetable*, *samplers*, etcétera.

Por último, mencionar que este trabajo se enmarca dentro de una reciente línea de investigación preocupada de relacionar las diferentes estrategias de *mapping* con los aspectos cognitivos del usuario. Trabajos como *divergent interface* (Dixon, Tubb 2014) o los estudios relacionados con los interpoladores gráficos de Gibson y Polfreman (Gibson, Polfreman 2019; 2020a; 2020b) fueron esenciales para la construcción de Octopus. Se espera que esta herramienta pueda servir como referente para otros trabajos y que en el futuro existan más opciones para el control alternativo de instrumentos electrónicos.

### **8.3. Proyecciones**

A continuación se presentan algunas ideas respecto de los caminos que puede seguir el desarrollo de esta herramienta.

#### **8.3.1. Lo técnico**

##### **8.3.1.1. Mejoras simples de baja complejidad**

En primer lugar, existe una serie de mejoras menores que pueden ser implementadas en el corto plazo. Entre ellas se encuentran:

- Mejorar el diseño gráfico de la interfaz tanto en aspectos estéticos como funcionales, buscando con ello facilitar la comprensión de la herramienta y de sus distintos componentes por parte del usuario.
- Añadir la posibilidad de que el usuario pueda crear y guardar sus propias distribuciones, lo que fue propuesto por dos de los cuatro entrevistados.
- Actualmente la interpolación que aplica Octopus es exclusivamente lineal, por lo que incluir en cada brazo la posibilidad de cambiar el tipo de

interpolación entre una lineal, una exponencial o una logarítmica podría enriquecer las posibilidades interpretativas de la herramienta.

- Aumentar la paleta de *rates* en el módulo de trayectorias, incluyendo así tiempos lentos, síncopas y tresillos.
- Diversificar las interacciones con la trayectoria llamada Gesto, posibilitando así modificar su velocidad de reproducción y otros aspectos del *playback*.

Además de lo anterior, otro aspecto que debe ser mejorado y cuya implementación no representa gran complejidad es aquello relacionado a la retroalimentación que entrega Octopus al usuario. En primer lugar, podría incluirse en cada brazo un pequeño cuadro de texto en el que se exponga el nombre del parámetro asignado, esto debido a que se observó durante las pruebas que los usuarios tuvieron problemas para recordar la configuración que establecieron. Por otro lado, también sería conveniente agrandar y mejorar la distribución de los potenciómetros que muestran el estado instantáneo de los controles. Además de lo anterior, se podrían buscar otras maneras de que Octopus mantenga informado al usuario de los procesos que están sucediendo dentro de su instrumento, esto con la finalidad de que la herramienta también pueda expandir el conocimiento técnico que se maneja de este último.

#### **8.3.1.2. Mejoras y añadiduras de mediana y alta complejidad**

De los resultados obtenidos es posible desprender una serie de mejoras y nuevas funciones que podrían hacer de Octopus una herramienta más completa. A continuación se presentan las que resultaron más relevantes.

Dentro de las ideas contenidas en esta sección, la primera y más simple es hacer caso a las recomendaciones de los entrevistados e implementar la posibilidad de aumentar o disminuir el número de brazos. Hasta qué punto disminuirlos o hasta qué punto aumentarlos es materia de reflexión puesto que la cantidad de

combinaciones vértice que se generen debe estar siempre en función de los recursos del usuario (Goudeseune 2002). Este añadido podría significar que Octopus se vuelva más versátil ya que podría configurarse según el contexto para cubrir distintas necesidades.

Así mismo, a futuro también es conveniente mejorar la integración con Ableton Live, y añadir la posibilidad de abrir los instrumentos virtuales directamente dentro de Octopus, con la ayuda del objeto *vst~*. Esto podría mejorar el flujo de trabajo de los usuarios y reducir la utilización de recursos computacionales.

Como fue mencionado previamente en este capítulo, otra función que definitivamente debe ser implementada en Octopus es la posibilidad de guardar sesiones de trabajo en las que se pueda conservar toda la información relacionada a mapeos, rangos y distribuciones. Esto permitiría a los usuarios poder realizar un trabajo continuado con la herramienta y poder pulir sus configuraciones hasta llegar a resultados que puedan ocupar en el largo plazo. Por otro lado, aprovechando que gran parte de los instrumentos electrónicos comerciales cuentan con más o menos los mismo parámetros, podría ser interesante que Octopus tenga plantillas predeterminadas que sugieran al usuario mapear ciertos controles con ciertos rangos. De esta manera se podría asegurar que los usuarios principiantes tengan una buena experiencia de entrada con la herramienta y que no la abandonen producto de su complejidad.

En línea con lo anterior, esta herramienta se podría ver beneficiada al incluir en su diseño una etapa de análisis de contenido de audio. Dicha etapa tendría lugar inmediatamente después de la configuración de los brazos y su objetivo sería caracterizar el sonido de cada una de las 256 combinaciones por medio de diferentes descriptores psicoacústicos. Los datos asociados a cada una de las combinaciones podrían ser ingresados a un algoritmo de reducción de dimensiones para así poder

generar *clusters* de sonidos que estarían clasificados según los descriptores de timbre. De esta manera Octopus podría ahorrarse la necesidad de incluir la distribución *Radar-256*, que justamente está pensada para poder conocer más o menos el sonido de las combinaciones, y el usuario no se enfrentaría al hecho de explorar el sonido de las combinaciones ‘a ciegas’.

Por otro lado, un añadido que podría significar una gran mejora para la esencia misma de la herramienta es encontrar métodos más rigurosos para diseñar las distribuciones. En este prototipo cada una de las distribuciones fue diseñada por el autor en base a su intuición e intentando buscar ciertos resultados, sin embargo, en ningún momento se realizó algún estudio en profundidad de la efectividad y eficiencia de estos diseños. Lo más cercano a esto último fue haber probado las distribuciones con el instrumento del autor, desechando o modificando las que tenían bajo desempeño, pero nada más. En el futuro sería interesante aplicar técnicas de aprendizaje de máquinas para poder maximizar la eficiencia y calidad de los diseños de las distribuciones.

### **8.3.2. Sobre la metodología e investigación**

Como fue discutido en secciones anteriores, la investigación aquí presente tuvo un enfoque mayoritariamente técnico con la finalidad de diseñar e implementar una herramienta para el control de IME. Sin embargo, ahora que ya están establecidos los fundamentos de diseño de Octopus, se libera espacio para que la investigación a futuro pueda hacerse de cargo de otras cuestiones tales como la dimensión cognitiva del proceso creativo o las implicaciones filosóficas del acto mismo de crear interfaces. Además, es necesario realizar un estudio más exhaustivo de la literatura relacionada con la evaluación de herramientas digitales, especialmente en el campo de los NIME, con el objetivo de diseñar una estrategia más sólida para evaluar el impacto de Octopus.

### **8.3.3. Más allá de los instrumentos musicales electrónicos**

Si bien este prototipo está diseñado para trabajar con instrumentos electrónicos, no hay motivo alguno para impedir que Octopus sea utilizado en cualquier otra área. Cualquier sistema electrónico cuyas variables puedan ser manejadas por señales de control externas es candidato a ser manipulado con Octopus. Disciplinas creativas del entorno digital tales como la síntesis de video, la programación de visuales, la animación, el diseño de videojuegos o la mezcla de audio podrían encontrar en Octopus una herramienta complementaria, útil e inspiradora.

## 9. Bibliografía.

Ableton Live | Ableton, sin fecha en línea. Recuperado a partir de : <https://www.ableton.com/en/live/what-is-live/> [accedido 25 enero 2023].

ANTHONY, Callen, NELSON, Andrew J. y TRIPSAS, Mary, 2016. “Who Are You?...I Really Wanna Know”: Product Meaning and Competitive Positioning in the Nascent Synthesizer Industry. *Strategy Science*. Vol. 1, número 3, pp. 163-183. DOI 10.1287/stsc.2016.0015.

Arturia - MicroFreak - MicroFreak, sin fecha en línea. Recuperado a partir de : <https://www.arturia.com/products/hardware-synths/microfreak/overview> [accedido 25 enero 2023].

BADÍA, Tere et al., 2016. Manifiesto for a critical approach to the user interface. Context, theoretical framework and further actions. En : *Interface Politics 1st International Conference*, pp. 217-225. . Barcelona, Spain. 2016. ISBN 978-84-617-5132-7.

BAKAN, Michael B. et al., 1990. Demystifying and classifying electronic music instruments. *Selected reports in ethnomusicology*. Vol. 8, pp. 37-63.

BARBOSA, Jerônimo et al., 2013. Illusio: A Drawing-Based Digital Music Instrument. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp.499-502. School of Culture Technology, KAIST. 2013. DOI <http://doi.org/10.5281/zenodo.1178566>.

BASTAN, Dillon, 2022. Vector Map version 1.0 by ndivuyo on maxforlive.com. *maxforlive.com* en línea. febrero 2022. Recuperado a partir de : <https://maxforlive.com/library/device/7879/vector-map> [accedido 12 julio 2022].

BAZZONI, Riccardo, 2019. fuse\* | Phy-Series. *fuseworks* en línea. septiembre 2019. Recuperado a partir de : <https://www.fuseworks.it/research/phy-series/> [accedido 10 julio 2022].

BENCINA, Ross, 2005. The Metasurface – Applying Natural Neighbour Interpolation to Two-to-Many Mapping. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp.101-104. 2005. DOI <http://doi.org/10.5281/zenodo.1176701>.

BEVILACQUA, Frédéric, MÜLLER, Rémy y SCHNELL, Norbert, 2005. MnM: a Max/MSP mapping toolbox. En : *Proceedings of the International Conference on New*

*Interfaces for Musical Expression*, pp. 85-88. 2005.  
DOI <http://doi.org/10.5281/zenodo.1176703>.

CADAVID, Laddy P, 2020. Knotting the memory//Encoding the Khipu\_: Reuse of an ancient Andean device as a NIME. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 495-498. Birmingham City University. 2020. DOI <http://doi.org/10.5281/zenodo.4813495>.

CHADABE, Joel, 2002. The Limitations of Mapping as a Structural Descriptive in Electronic Instruments. En : *Proceedings of the 2002 conference on New interfaces for musical expression*, pp. 38-42. Dublin, Ireland. 2002. DOI [10.5281/zenodo.1176402](http://doi.org/10.5281/zenodo.1176402).

CHWALEK, Patrick y PARADISO, Joseph A, 2019. CD-Synth: a Rotating, Untethered, Digital Synthesizer. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 371-374. UFRGS. 2019. DOI <http://doi.org/10.5281/zenodo.3672998>.

DE LAS POZAS, Virginia, 2020. Semi-Automated Mappings for Object-Manipulating Gestural Control of Electronic Music. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 631-634. Birmingham City University. 2020. DOI <http://doi.org/10.5281/zenodo.4813232>.

DÍAZ-BRAVO, Laura et al., 2013. La entrevista, recurso flexible y dinámico. *Investigación en Educación Médica*. Vol. 2, número 7, pp. 162-167. DOI [10.1016/S2007-5057\(13\)72706-6](http://doi.org/10.1016/S2007-5057(13)72706-6).

DILIBERTO, John K, 1983. An Interview with Donald Buchla. *Polyphony*. Vol. 8, número 5, pp. 14-17.

DIXON, Simon y TUBB, Robert, 2014. The Divergent Interface: Supporting Creative Exploration of Parameter Spaces. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 227-232. Goldsmiths, University of London. 2014. DOI <http://doi.org/10.5281/zenodo.1178967>.

ELLIS, Carolyn, ADAMS, Tony E y BOCHNER, Arthur P, 2015. AUTOETNOGRAFÍA: UN PANORAMA. *Astrolabio*. Número 14, pp. 249-273. DOI <https://doi.org/10.55441/1668.7515.n14.11626>.

ESLING, Philippe et al., 2020. Flow Synthesizer: Universal Audio Synthesizer Control with Normalizing Flows. *Applied Sciences*. Vol. 10, número 1, p. 302. DOI [10.3390/app10010302](http://doi.org/10.3390/app10010302).

EXPRESSIVE E, sin fecha. Touché. *Expressive E* en línea. Recuperado a partir de : <https://www.expressivee.com/1-touche> [accedido 12 julio 2022].

FASCIANI, Stefano y GOODE, Jackson, 2021. 20 NIMES: Twenty Years of New Interfaces for Musical Expression. En : *NIME 2021*. Shanghai, China : PubPub. 1 junio 2021. DOI 10.21428/92fbeb44.b368bcd5.

FELS, Sidney, GADD, Ashley y MULDER, Axel, 2003. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*. Vol. 7, número 2, pp. 109-126. DOI 10.1017/S1355771802002042.

FIEBRINK, Rebecca, TRUEMAN, Dan y COOK, Perry R, 2009. A Meta-Instrument for Interactive, On-the-fly Machine Learning. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 280-285. Pittsburgh, United States. 2009. DOI <http://doi.org/10.5281/zenodo.1177513>.

FLOSZK, 2015. Shaker version 1.0 by floszk on maxforlive.com. *maxforlive.com* en línea. julio 2015. Recuperado a partir de : <https://maxforlive.com/library/device/3118/shaker> [accedido 11 julio 2022].

GIBSON, Darrell y POLFREMÁN, Dr Richard, 2019. A Framework for the Development and Evaluation of Graphical Interpolation for Synthesizer Parameter Mappings. En : *Proceedings of the 16th Sound & Music Computing Conference*, pp. 302-309. Málaga, Spain. mayo 2019. ISBN 978-84-09-08518-7. DOI <https://doi.org/10.5281/zenodo.3249366>.

GIBSON, Darrell y POLFREMÁN, Richard, 2020a. Analyzing journeys in sound: usability of graphical interpolators for sound design. *Personal and Ubiquitous Computing*. Vol. 25, número 4, pp. 663-676. DOI 10.1007/s00779-020-01398-z.

GIBSON, Darrell y POLFREMÁN, Richard, 2020b. Star Interpolator – A Novel Visualization Paradigm for Graphical Interpolators. En : *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME-20)*, pp. 49-54. Birmingham, UK. junio 2020. ISBN 2220-4806. DOI 10.5281/zenodo.4813168.

GOUDESEUNE, Camille, 2002. Interpolated mappings for musical instruments. *Organised Sound*. Vol. 7, número 2, pp. 85-96. DOI 10.1017/S1355771802002029.

HARDJOWIROGO, Sarah-Indriyati, 2017. Instrumentality. On the Construction of Instrumental Identity. En : BOVERMANN, Till et al. (eds.), *Musical Instruments in the 21st Century*, pp. 9-24. Singapore : Springer Singapore. ISBN 978-981-10-2950-9. DOI 10.1007/978-981-10-2951-6\_2.

HINRICHSEN, Amelie y BOVERMANN, Till, 2016. Post-DMI musical Instruments. En : *Proceedings of the Audio Mostly 2016*, pp. 124-131. Norrköping Sweden : ACM. 4 octubre 2016. ISBN 978-1-4503-4822-5. DOI 10.1145/2986416.2986440.

HUNT, A. y KIRK, Ross, 2000. Mapping Strategies for Musical Performance. En : *Trends in Gestural Control of Music* en línea. París, France, IRCAM : Centre Georges Pompidou. ISBN 2-84426-039-X. Recuperado a partir de : <https://www.semanticscholar.org/paper/Mapping-Strategies-for-Musical-Performance-Hunt-Kirk/24bae2761b3985db7b43b92553b5cf909c0c5f12> [accedido 22 octubre 2021].

HUNT, Andy D., WANDERLEY, Marcelo M. y PARADIS, Matthew, 2002. The importance of Parameter Mapping in Electronic Instrument Design. En : *Proceedings of the 2002 conference on New interfaces for musical expression*, pp. 88-93. Dublin, Ireland : Zenodo. 1 junio 2002. DOI 10.5281/zenodo.1176424. event: International Conference on New Interfaces for Musical Expression DOI: 10.5281/zenodo.1176424page: 88-93container-title: Proceedings of the International Conference on New Interfaces for Musical Expression

HUNT, Andy y WANDERLEY, Marcelo, 2002. Mapping performer parameters to synthesis engines. *Organised Sound*. Vol. 7, número 2, pp. 97-108. DOI 10.1017/S1355771802002030.

JENSENIUS, Alexander Refsum, 2011. Beyond Evaluation : Linking Practice and Theory in New Musical Interface Design. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 280-283. Oslo, Norway. 2011. DOI <http://doi.org/10.5281/zenodo.1178053>.

JORDÀ, Sergi, 2004. Instruments and Players: Some Thoughts on Digital Lutherie. *Journal of New Music Research*. Vol. 33, número 3, pp. 321-341. DOI 10.1080/0929821042000317886.

KALTENBRUNNER, Martin et al., 2006. The reacTable\*: A Collaborative Musical Instrument. En : *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*, pp. 406-411. Manchester, UK : IEEE. 2006. ISBN 978-0-7695-2623-2. DOI 10.1109/WETICE.2006.68.

KARTOMI, Margaret, 2001. The Classification of Musical Instruments: Changing Trends in Research from the Late Nineteenth Century, with Special Reference to the 1990s. *Ethnomusicology*. Vol. 45, número 2, pp. 283-314. DOI 10.2307/852676.

KORG, sin fecha. KAOSILATOR PRO+ - DYNAMIC PHRASE SYNTHESIZER/LOOP RECORDER | KORG (Chile). *KORG Global* en línea. Recuperado a partir de : [https://www.korg.com/cl/products/dj/kaossilator\\_pro\\_plus/](https://www.korg.com/cl/products/dj/kaossilator_pro_plus/) [accedido 11 julio 2022 a].

KORG, sin fecha. KAOS PAD KP3+ - DYNAMIC EFFECT/SAMPLER | KORG (Chile). *KORG Global* en línea. Recuperado a partir de : [https://www.korg.com/cl/products/dj/kaoss\\_pad\\_kp3\\_plus/](https://www.korg.com/cl/products/dj/kaoss_pad_kp3_plus/) [accedido 12 julio 2022 b].

KVIFTE, Tellef, 2008. On the Description of Mapping Structures. *Journal of New Music Research*. Vol. 37, número 4, pp. 353-362. DOI 10.1080/09298210902731394.

LAM, Nina Siu-Ngan, 1983. Spatial Interpolation Methods: A Review. *The American Cartographer*. Vol. 10, número 2, pp. 129-150. DOI 10.1559/152304083783914958.

LE VAILLANT, Gwendal, DUTOIT, Thierry y GIOT, Rudi, 2020. Analytic vs. holistic approaches for the live search of sound presets using graphical interpolation. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 227-232. Birmingham City University. 2020. DOI <http://doi.org/10.5281/zenodo.4813330>.

Live Instrument Reference — Ableton Reference Manual Version 11 | Ableton, sin fecha en línea. Recuperado a partir de : <https://www.ableton.com/en/manual/live-instrument-reference/#26-6-operator> [accedido 25 enero 2023].

LÓPEZ-CANO, Rubén y OPAZO, Úrsula San Cristóbal, 2014. *Investigación artística en música*. Barcelona, Spain. ISBN 978-84-697-1948-0.

MAGNUSSON, Thor, 2017. Contextualizing Musical Organics: An Ad-hoc Organological Classification Approach. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 470-475. Aalborg University Copenhagen. 2017. DOI <http://doi.org/10.5281/zenodo.1176320>.

MAGNUSSON, Thor y HURTADO, Enrike, 2017. The Acoustic, the Digital and the Body: A Survey on Musical Instruments. En : JENSENIUS, Alexander Refsum y LYONS, Michael J. (eds.), *A NIME Reader*, pp. 317-333. Cham : Springer International Publishing. 2017. Current Research in Systematic Musicology. ISBN 978-3-319-47213-3. DOI 10.1007/978-3-319-47214-0\_21.

MALLOCH, Joseph y WANDERLEY, Marcelo M., 2007. The T-Stick: from musical interface to musical instrument. En : *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, pp. 66-69. New York, New York : ACM Press. 2007. DOI 10.1145/1279740.1279751.

MARIER, Martin, 2012. Designing Mappings for Musical Interfaces Using Preset Interpolation. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*. University of Michigan. 2012. DOI <http://doi.org/10.5281/zenodo.1178343>.

Max for Live | Cycling '74, sin fecha en línea. Recuperado a partir de : <https://cycling74.com/products/maxforlive> [accedido 25 enero 2023].

MCGLYNN, Patrick, 2011. Towards more effective mapping strategies for digital musical instruments. En : *Proceedings of the 9th Annual Linux Audio Conference*. . Maynooth, Ireland. 2011.

MCPHERSON, Andrew P et al., 2016. NIMEhub: Toward a Repository for Sharing and Archiving Instrument Designs. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 4. . Queensland Conservatorium, Griffith University. 2016.

minilogue - POLYPHONIC ANALOGUE SYNTHESIZER | KORG (Chile), sin fecha en línea. Recuperado a partir de : <https://www.korg.com/cl/products/synthesizers/minilogue/> [accedido 25 enero 2023].

MIRANDA, Eduardo y WANDERLEY, Marcelo, 2006. *New Digital Music Instruments: Control And Interaction Beyond the Keyboard*. Middleton, Wisconsin : A-R Editions. Computer Music and Digital Audio Series.

nodes Object Reference - Max 7 Documentation, sin fecha en línea. Recuperado a partir de : <https://docs.cycling74.com/max7/refpages/nodes> [accedido 25 enero 2023].

PAINE, Garth, 2010. Towards a Taxonomy of Realtime Interfaces for Electronic Music Performance. En : *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression*, pp. 436-439. Sydney, Australia. 2010. DOI <http://doi.org/10.5281/zenodo.1177873>.

PARADISO, Joseph A. y O'MODHRAIN, Sile, 2003. Current Trends in Electronic Music Interfaces. Guest Editors? Introduction. *Journal of New Music Research*. Vol. 32, número 4, pp. 345-349. DOI 10.1076/jnmr.32.4.345.18855.

PATCHES, 2020. Gradient version 1.0.1 by pATCHES on maxforlive.com. *maxforlive.com* en línea. mayo 2020. Recuperado a partir de : <https://maxforlive.com/library/device/6241/gradient> [accedido 11 julio 2022].

PINCH, Trevor y TROCCO, Frank, 2002. *Analog days: the invention and impact of the Moog synthesizer*. Cambridge, MA : Harvard University Press. ISBN 0-674-00889-8.

PSYOPS PROJECT, 2013. *Tutorial Kaossilator Pro as MIDI Controller playing Gaia SH-01* en línea. 22 octubre 2013. Recuperado a partir de : <https://www.youtube.com/watch?v=bpdcpc86OQQ> [accedido 19 julio 2022].

RASMUSSEN, Curtis E, 2018. *Evaluating the Usability of Software Synthesizers: An Analysis and First Approach*. . Guelph, Ontario, Canada : The University of Guelph.

Roland - GAIA SH-01 | Synthesizer, sin fecha en línea. Recuperado a partir de : [https://www.roland.com/global/products/gaia\\_sh-01/](https://www.roland.com/global/products/gaia_sh-01/) [accedido 25 enero 2023].

ROVAN, Joseph Butch et al., 1997. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. En : *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop*, pp. 68-73 en línea. Genes, Italy : Associazione di Informatica Musicale Italiana. octubre 1997. Recuperado a partir de : <https://hal.archives-ouvertes.fr/hal-01105514> [accedido 17 octubre 2021].

rslider Object Reference - Max 7 Documentation, sin fecha en línea. Recuperado a partir de : <https://docs.cycling74.com/max7/refpages/rslider> [accedido 25 enero 2023].

SACHS, Curt, 2006. *The History of Musical Instruments*. New York : Courier Corporation. ISBN 0 393 02068 1.

scale Reference - Max 8 Documentation, sin fecha en línea. Recuperado a partir de : <https://docs.cycling74.com/max8/refpages/scale> [accedido 25 enero 2023].

SEAGO, Allan, HOLLAND, Simon y MULHOLLAND, Paul, 2004. A Critical Analysis of Synthesizer User Interfaces for Timbre. En : DEARDEN, Andy y WATT, Leon (eds.), *Proceedings of the XVIII British HCI Group Annual Conference HCI 2004*, pp. 105-108 en línea. Bristol, UK : Research Press International. 2004. 4 p. ISBN 978-

1-897851-13-5. Recuperado a partir de : <http://oro.open.ac.uk/5688/> [accedido 18 octubre 2021].

SPAIN, Martin y POLFREMAN, Richard, 2001. Interpolator: a two-dimensional graphical interpolation system for the simultaneous control of digital signal processing parameters. *Organised Sound*. Vol. 6, número 2, pp.147-151. DOI 10.1017/S1355771801002114.

STEFFENSEN, Jogan Frederik, 2006. *Interpolation: Second Edition*. Mineola, New York : Dover Publications. ISBN 0-486-45009-0.

STÉPHANIE WEISSER y MAARTEN QUANTEN, 2011. RETHINKING MUSICAL INSTRUMENT CLASSIFICATION: TOWARDS A MODULAR APPROACH TO THE HORNBOSTEL-SACHS SYSTEM. *Yearbook for Traditional Music*. Vol. 43, pp. 122-146. DOI 10.5921/yeartradmusi.43.0122.

STOWELL, Dan, PLUMBIEY, Mark D. y BRYAN-KINNS, Nick, 2008. Discourse Analysis Evaluation Method for Expressive Musical Interfaces. En : *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 81-86. Genova, Italy. 2008. DOI <http://doi.org/10.5281/zenodo.1179631>.

The International Conference on New Interfaces for Musical Expression | NIME, sin fecha en línea. Recuperado a partir de : <https://www.nime.org/> [accedido 31 enero 2023].

TODOROFF, T., 2002. Control of Digital Audio Effects. En : ZÖLZER, Udo (ed.), *DAFX: Digital Audio Effects*, pp. 465-497. Chichester, UK : John Wiley & Sons, Ltd. ISBN 978-0-471-49078-4. DOI 10.1002/047085863X.ch12.

VAN NORT, Doug y CASTAGNÉ, Nicolas, 2007. Mapping, in digital musical instruments. En : *Enaction and Enactive Interfaces: A Handbook of Terms*, pp. 191-192 en línea. ACROE. Enactive Systems Books. ISBN 978-2-9530856-0-0. Recuperado a partir de : <https://hal.archives-ouvertes.fr/hal-00289119>

VAN NORT, Doug, WANDERLEY, Marcelo M. y DEPALLE, Philippe, 2014. Mapping Control Structures for Sound Synthesis: Functional and Topological Perspectives. *Computer Music Journal*. Vol. 38, número 3, pp. 6-22. DOI 10.1162/COMJ\_a\_00253.

VARGAS CORDERO, Zoila Rosa, 2009. La Investigación aplicada: Una forma de conocer las realidades con evidencia científica. *Revista Educación*. Vol. 33, número 1, p. 155. DOI 10.15517/revedu.v33i1.538.

Virtual Analog Synth – Element 2.0 | Waves, sin fecha *waves.com* en línea. Recuperado a partir de : <https://www.waves.com/plugins/element> [accedido 31 enero 2023].

volca sample - DIGITAL SAMPLE SEQUENCER | KORG (Chile), sin fecha en línea. Recuperado a partir de : [https://www.korg.com/cl/products/dj/volca\\_sample/index.php](https://www.korg.com/cl/products/dj/volca_sample/index.php) [accedido 25 enero 2023].

vst~ Object Reference - Max 7 Documentation, sin fecha en línea. Recuperado a partir de : <https://docs.cycling74.com/max7/refpages/vst~> [accedido 25 enero 2023].

WANDERLEY, M.M. y DEPALLE, P., 2004. Gestural Control of Sound Synthesis. *Proceedings of the IEEE*. Vol. 92, número 4, pp. 632-644. DOI 10.1109/JPROC.2004.825882.

WANG, Johnty et al., 2019. Webmapper: A Tool for Visualizing and Manipulating Mappings in Digital Musical Instruments. En : *Proceedings of the 14th International Symposium on Computer Music Multidisciplinary Research*, pp. 823-834. . Marseille, France. 2019.

WESSEL, David y WRIGHT, Matthew, 2002. Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*. Vol. 26, número 3, pp. 11-22. DOI <https://doi.org/10.1162/014892602320582945>.

What is Max? | Cycling '74, sin fecha en línea. Recuperado a partir de : <https://cycling74.com/products/max> [accedido 25 enero 2023].

YANG, Chin-Shung et al., 2004. TWELVE DIFFERENT INTERPOLATION METHODS: A CASE STUDY OF SURFER 8.0. En : *Proceedings of the XXth ISPRS congress*, pp. 778-785. . Istanbul, Turkey. 2004.

# ANEXOS

## Anexo A: Guía de entrevista

### Objetivos

- a. Conocer la impresión que genera *Octopus* en personas que utilizan por primera vez la herramienta, esto debido al sesgo que presenta el autor a la hora de mirar su creación desde el punto de vista de usuario.
- b. Determinar si la herramienta puede modificar la manera en la que el usuario se relaciona con su instrumento electrónico. Esto incluye los resultados a los que puede llegar, la velocidad del flujo de trabajo, la noción que se tiene de las capacidades instrumento, etcétera.
- c. Obtener retroalimentación de los usuarios con la finalidad de identificar las posibles mejoras que pueden añadirse al prototipo, poniendo especial atención a aquellos aspectos que NO les hayan gustado.

### Perfil del usuario

Se trabajará con personas que estén familiarizadas con la práctica musical desde el mundo de los instrumentos musicales electrónicos. Además, será requisito excluyente que las personas utilicen *Octopus* sobre un instrumento con el que tengan experiencia de al menos un año, esto con la finalidad de que puedan determinar si es que efectivamente la herramienta modifica el comportamiento de su IME.

### Descripción de la experiencia

Experiencia de carácter presencial, duración aproximada de 1 hora en total.

#### Primera parte

Se realizará una serie de preguntas de control con tal de conocer la relación existente entre el usuario y su instrumento electrónico. Luego de eso, se presentará la herramienta al usuario introduciendo los fundamentos que operan en ella.

## **Segunda parte**

Esta parte de la experiencia consiste en que el usuario pueda manejar su instrumento electrónico con *Octopus*. Para ello se realizará una pequeña inducción de aproximadamente 10 minutos a la interfaz y funcionalidades de la herramienta. Luego, el usuario podrá experimentar con la herramienta durante no menos de 10 minutos ni más de 40 minutos. La experiencia será acompañada constantemente por el autor y se le proveerá de asistencia técnica al usuario cada vez que lo requiera. Además, la sesión de experimentación será grabada tanto en audio como en video.

## **Tercera parte**

Se realizará una segunda ronda de preguntas al usuario, esta vez orientadas a conocer su experiencia con la herramienta así como las posibles comparaciones que se pueden establecer entre este y su flujo normal de trabajo. Además, se le preguntará por los aspectos negativos de la experiencia.

## **Preguntas**

### **Previas al *testeo* de la herramienta:**

1. ¿Qué uso le das a este instrumento cuando lo incluyes en tu práctica musical?  
Ej: creación de sonidos tonales como melodías o acordes, creación de sonidos atonales como texturas o ritmos, entre otros.
2. ¿Cómo es habitualmente tu proceso para diseñar sonidos en este instrumento? ¿Cuáles son los pasos que sigues?
3. ¿Qué tan rápido es para ti llegar a resultados satisfactorios con este instrumento? ¿Con qué dificultades te encuentras?

### **Posteriores al *testeo* de la herramienta:**

4. ¿Cuáles son tus primeras impresiones luego de utilizar esta herramienta?

5. ¿Qué te parece el diseño de la herramienta? (considere cosas como flujo de trabajo, la cantidad de datos que te pide ingresar, la disposición de los elementos en la interfaz, la velocidad de aprendizaje, etcétera)
6. ¿Qué tan fácil o difícil te pareció utilizar la herramienta?
7. ¿Cómo se relaciona la interacción que ofrece esta herramienta con tu forma habitual de trabajar con el instrumento?
8. Esta herramienta ¿Tiene cabida en tu forma de trabajar? ¿Dónde?
9. Luego de estos minutos de uso ¿Qué piensas de los timbres a los que pudiste llegar?
10. En relación a tu experiencia previa con el instrumento, los resultados que conseguiste en la sesión de hoy ¿se parecen a los que podrías haber obtenido sin utilizar la herramienta?
11. ¿Qué cosas no te gustaron de la experiencia? ¿Qué cosas podrían ser mejoradas?

### **Consentimiento informado**

**Consentimiento:**

Se le solicita a \_\_\_\_\_, rut: \_\_\_\_\_ el consentimiento de la grabación en formato audio y video de la entrevista y la autorización para incorporar la información que él o ella entregue para ser parte de la investigación.

La información recolectada será empleada en la memoria de título “Desarrollo de una herramienta interactiva para el control creativo de instrumentos musicales electrónicos”, la cual será realizada por Pablo Madariaga Lepe, estudiante de Ingeniería en Sonido de la Universidad de Chile, para fines estrictamente académicos.

\_\_\_\_\_  
Firma entrevistado y fecha

\_\_\_\_\_  
Firma entrevistador

## **Anexo B: Links a videos demostrativos**

A continuación se presenta una lista en la que se adjuntan los link que dirigen a los videos demostrativos de la herramienta.

**Demostración de algoritmo de brazos:**

<https://youtu.be/VA3KCbZL1eY>

**Demostración del objeto *rslider*:**

<https://youtu.be/VA3KCbZL1eY?t=409>

**Demostración algoritmo de almacenamiento de combinaciones vértice:**

<https://youtu.be/40QzyN3qVAU>

**Demostración del sonido de las combinaciones vértice:**

<https://youtu.be/40QzyN3qVAU?t=144>

**Demostración del funcionamiento del objeto *nodes*:**

<https://youtu.be/RYdUTYsurUI>

**Demostración del algoritmo para guardar sonidos de usuario:**

<https://youtu.be/RYdUTYsurUI?t=874>

**Demostración de la distribución Radar-256:**

<https://youtu.be/RYdUTYsurUI?t=281>

**Demostración de la distribución Ultra Caos 64:**

<https://youtu.be/RYdUTYsurUI?t=350>

**Demostración de la distribución Mini Caos 25:**

<https://youtu.be/RYdUTYsurUI?t=427>

**Demostración de la distribución Flores:**

<https://youtu.be/RYdUTYsurUI?t=500>

**Demostración de la distribución Flor:**

<https://youtu.be/RYdUTYsurUI?t=582>

**Demostración de la distribución Caleidoscopio:**

<https://youtu.be/RYdUTYsurUI?t=661>

**Demostración de la distribución Tripofobia:**

<https://youtu.be/RYdUTYsurUI?t=713>

**Demostración de la distribución MicroCaos:**

<https://youtu.be/RYdUTYsurUI?t=792>

**Demostración de la trayectoria manual:**

<https://youtu.be/RYdUTYsurUI?t=876>

**Demostración de la trayectoria Gesto:**

<https://youtu.be/RYdUTYsurUI?t=944>

**Demostración de la trayectoria Secuencia de nodos:**

<https://youtu.be/RYdUTYsurUI?t=1108>

**Demostración de la trayectoria Secuencia de puntos:**

<https://youtu.be/RYdUTYsurUI?t=1333>

**Demostración de la trayectoria Circunferencia:**

<https://youtu.be/RYdUTYsurUI?t=1390>

**Demostración de la trayectoria Espiral:**

<https://youtu.be/RYdUTYsurUI?t=1467>