

Propuesta inicial de la formación del cap Performatividad  
de la(s) máquina(s) + Código + IA + Live Coding +  
Agencialidad + Composición algorítmica + Música  
generativa

Emilio, Aaron, Marianne, Hernani, Iván, Pablo, Dorian

25 de marzo de 2021

# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. La capacidad investigativa del algoritmo y el código: trazos sobre el ciclo práctica, tecnología e investigación</b>	<b>3</b>
2.1. Introducción . . . . .	3
2.2. Código y algortimo . . . . .	4
2.3. La condición del artista investigador/programador . . . . .	5
2.4. Los agentes que se agregan a la fórmula . . . . .	6
2.5. Referencias . . . . .	7
<b>3. SubCap2</b>	<b>9</b>
3.1. si necesitamos subsecciones . . . . .	9
<b>4. SubCap3</b>	<b>10</b>
4.1. si necesitamos subsecciones . . . . .	10
<b>5. SubCap4</b>	<b>11</b>
5.1. si necesitamos subsecciones . . . . .	11
<b>6. Conclusiones</b>	<b>12</b>
6.1. si necesitamos subsecciones . . . . .	12
<b>7. Anexos</b>	<b>13</b>
7.1. si necesitamos subsecciones . . . . .	13

# Capítulo 1

## Introducción

Aquí va la introducción.

## Capítulo 2

# La capacidad investigativa del algoritmo y el código: trazos sobre el ciclo práctica, tecnología e investigación

Aaron Escobar Castañeda y Hernani Villaseñor Ramírez

### Abstract

Este texto parte de nuestras investigaciones doctorales bajo el reto de construirlo con temáticas que están en proceso activo de problematización. Los términos y preguntas que arroja quizá son imposibles de resolver en un escrito de las dimensiones propuestas. Sin embargo, este ejercicio nos lleva a detectar puntos en común y diferencias acerca de la materialidad de nuestras investigaciones: el algoritmo y el código. A partir de lo anterior, problematizamos la ‘capacidad investigativa’ del algoritmo y el código así como el ciclo de retroalimentación de la práctica artística, el desarrollo tecnológico y la investigación dentro de nuestros trabajos. En este texto trataremos de describir la metodología que nos lleva a problematizar el ensamblaje entre humanos, máquinas, prácticas y tecnologías en la creación de música con computadoras en el ámbito de la improvisación libre con aprendizaje y escucha de máquinas y el live coding. El término que encontramos como punto de transformación de nuestras prácticas es el de algoritmicidad o agencia algorítmica desde donde nos preguntamos ¿cómo transforman los algoritmos nuestras prácticas musicales? y ¿cómo se define el artista-programador-investigador dentro del ciclo de retroalimentación mencionado?

## 2.1. Introducción

Escribir un texto colaborativo implica negociar perspectivas, ideas, términos y conceptos para llegar a un acuerdo sobre el tema o problema a desarrollar. Como estrategia hemos escrito párrafos intercalados de los cuales sustrajimos términos que nos parecen relevantes. Pese a que el resultado de este procedimiento es extenso y poco acotado, nos ayudó a detectar algunas temáticas transversales a las dos investigaciones. Estas recaen en las prácticas y técnicas musicales que se desprenden del cruce entre la música y las ciencias de la computación. Por un lado, la creación de un sistema de escucha automática para la libre improvisación y por el otro, el estudio de una práctica artística de live coding basada en la captura de sonido al momento.

Como primer punto encontramos que nos interesa la ‘capacidad investigativa’ del algoritmo y el código; y como segundo punto el ciclo de retroalimentación por el que pasa el artista-investigador. Dicho ciclo se compone por la práctica artística, el desarrollo tecnológico y la investigación, el cual es puesto a prueba en el laboratorio de experimentación que vemos en el performance. A partir de estas aproximaciones trataremos de desarrollar nuestro texto con los siguientes conceptos: agencia, algoritmicidad, creatividad computacional y programación creativa. Así pues, se estudiarán las relaciones entre manos y mentes, técnicas e ideas, materias y formas reforzando el componente experimental para conceptualizar tanto errores como aciertos que promuevan la construcción de un pensamiento artístico-tecno-científico comprometido con la generación de nuevos conceptos, teorías e hipótesis.

## 2.2. Código y algoritmo

Esta sección vincula los términos código fuente y algoritmo a partir de las siguientes preguntas: ¿qué relación tiene un algoritmo con su código? y ¿cómo se expresan a través de una práctica musical computacional? Para abordar estas preguntas partimos de que el código computacional es una serie de símbolos y caracteres organizados con un lenguaje de programación legibles por el humano y, posteriormente, al ser traducidos son ejecutables por la máquina. Por algoritmo entendemos un objeto técnico integrado por procesos lógicos y de control que articula un medio/espacio de algoritmicidad o agencialidad algorítmica que incluye los principios que animan los algoritmos.<sup>1</sup> Es decir, la capacidad de interactuar y experimentar con ellos y entenderlos como vehículos de pensamiento además de su propia relación con los procesos de intercambio afectivo con su entorno (sea digital o físico). En la algoritmicidad se insertan las prácticas de los

---

<sup>1</sup>Kowalski (1979): “Algoritmo = Lógica + Control”. La parte lógica se compone de definiciones de procedimientos abstractos relacionados con el conocimiento sobre el dominio del problema y de las estructuras de datos sobre las que operan estos procedimientos, mientras que la parte de control se ocupa de estrategias para convertir el componente lógico en una máquina eficiente, estrategias para desenrollar el conocimiento en el tiempo y el espacio.

científicos de la computación pero también de artistas, programadores independientes y usuarios. El algoritmo es entonces una abstracción que tiene una materialidad autónoma al código fuente y a los lenguajes de programación que harían posible su ejecución. Con lo anterior no proponemos el análisis del código fuente de un algoritmo, aunque esta sea una forma de aproximarnos a las implicaciones del algoritmo, sino aquello que posibilita en el contexto de una práctica musical, es decir, los espacios de agencialidad de los algoritmos/códigos o algortimicidad. Aquí, entendida como la implicación que tienen tanto dentro como fuera de sus procesos de ejecución, así como su capacidad para transformar la práctica y generar un resultado específico o no. En este contexto podemos pensar, por un lado, un panorama que ve al humano al centro de la creatividad con computadoras, y por otro lado, a la máquina como agente que participa en la creación musical. Estos escenarios nos dejan ver diferentes aproximaciones de la relación humano-máquina en la creación musical con lenguajes de programación, desde la escritura de algoritmos en vivo, pasando por la colaboración entre humanos y agentes artificiales hasta la creación/interpretación autónoma de las máquinas.

Retomando estas relaciones sería interesante preguntarnos ¿qué significa lo experimental en el contexto de los algoritmos? y ¿cuáles son los límites entre humano y máquina? Estas preguntas buscan prevenir una estabilización y oscilaciones dirigidas a la simple demostración del algoritmo a través de una prueba autocontenida que lo llevaría a perder su capacidad investigativa. Se parte de la idea de que los algoritmos (incluido también el hardware) no son entidades estructuradas fijas en el tiempo-espacio-materia sino que aparecen impermanentes a través de la posibilidad intrínseca a su ejecución, manipulación transformación y actualización ad infinitum. En vez de apostar por una estética probabilística y predictiva se parte de una “tendencia especulativa intrínseca de la computación, que produce una novedad genuina incapaz de explicarse por fuerzas externas o condiciones iniciales” (Parisi, 2013). Por otro lado, la “agencialidad crítica”, es decir, una aproximación que analiza permanentemente cómo, cuándo, dónde y porqué se produce la agencia, es una interfaz entre el investigador y su aparato, una herramienta epistémica que serviría para desdibujar cada vez más las fronteras entre estos agentes y reforzar la injerencia/proximidad/entrelazamiento de la máquina en el artista-investigador y viceversa (extimacy Rheinberger 2013, intra-action Barad 2012).

### 2.3. La condición del artista investigador/programador

En esta sección vamos a definir nuestra condición de artistas investigadores/programadores para aclarar el contexto en el que escribimos. Al momento de escribir este texto ambos autores nos encontramos realizando el doctorado en música en el área de tecnología musical. Nuestras investigaciones, aunque tratan temas distintos dentro del ámbito de la música computacional, están conectadas por la aproximación a la investigación desde la práctica artística y el desarrollo

de tecnología. En este sentido, escribimos sobre casos que se transforman conforme avanza cada investigación y cuya materialidad está constituida por algoritmos escritos con código computacional. Por un lado, estos algoritmos están aplicados a la escucha y al aprendizaje de máquinas; por otro lado, a la escritura y modificación en vivo.

La forma de aproximarnos a la investigación, desde el arte, nos lleva a pensar cómo estructuramos nuestros trabajos de investigación. Desde la condición de artista programador a veces no se parte desde método científico riguroso que resuelve algo específico o que trata de demostrar algo. Más bien de un proceso heurístico que muestra un fenómeno intrincado de los que se tienen claros los problemas pero requiere entender la utilidad que tendrán en la música al ser resueltos. Esta aparente disociación de campos especializados nos lleva a preguntarnos por nuestro rol dentro del proceso que implica cada investigación. ¿Soy programador, compositor, improvisador, investigador? o ¿cómo operan estos roles en el ciclo de retroalimentación de la investigación artística y tecnológica? Quizá la respuesta es que no se pueden separar sino que actúan en múltiples hilos, en lo que tal vez vale la metáfora del los múltiples proceso que realizan un procesador computacional al mismo tiempo.

Otra aproximación del artista investigador en el contexto de la investigación estudia la creatividad humana con lenguajes de programación sonora al centro de esta relación. Desde esta perspectiva, que tiene que ver más con la práctica del live coding, el artista investigador genera sus algoritmos en el momento de una presentación, los cuales se caracterizan por ser pequeños y modificables. Esta práctica implica probar y memorizar estos algoritmos en varias circunstancias, es decir, estos no emergen de la nada sino provienen del bagaje de conocimientos de quien práctica el live coding. Marije Baalman (2015) se refiere a este tipo de escritura internalizada en la memoria y en la habilidad de teclear como la in-corporación del código. El humano memoriza el código, el algoritmo y lo que posiblemente obtendrá a su salida y se familiariza con el proceso físico de teclear. En este sentido, la habilidad de lecto escritura del código, que algunos autores refieren como literacidad de la programación (Soon y Cox, 2020; y Marino, 2020) y la capacidad estética de la programación (Soon y Cox, 2020) conforman un campo de reflexión en el que el código y el algoritmo, en sus procesos de iteración, introducen los referentes teóricos del artista investigador, ampliando el bucle de la práctica artística al incluir prácticas de investigación.

## 2.4. Los agentes que se agregan a la fórmula

Vayamos pues a problematizar sobre cada una de estas agencias, intentar ver qué hay más allá del reduccionismo para visibilizar el sistema complejo al que nos enfrentamos al hablar de agencia en términos de co-creación donde varios agentes humanos y no humanos a distintos

niveles y en distintos espacios-tiempos están implicados en la generación de una obra o una improvisación.

Desde el punto de vista de la creación de la obra por el artista-programador hay que considerar qué implica el desarrollo e implementación de un algoritmo para generar una obra artística. Lo primero es afirmar que ningún conocimiento parte de cero, sería ingenuo pensar que el artista-programador apartado de todo contexto, es capaz de generar un desarrollo tecnológico que no requiera de la línea evolutiva de la algoritmicidad. En este punto, entendida como la actividad humana que gira en torno al proceso de estudio, modificación, transformación, reapropiación y distribución al que están sujetos los algoritmos y su implementación en código al ser empleados para intentar resolver un problema específico o no. Segundo, todos los agentes humanos y no humanos interconectados sufren un proceso de afectación mutua que los lleva a transformar sus objetivos iniciales. Bruno Latour plantea el concepto de rodeo para referirse a los procesos de transformación de los objetivos iniciales que surgen del proceso de afectación de dos o más agentes implicados en un problema. Resulta interesante analizar este concepto para referirnos a cómo el algoritmo, el artista-programador y el contexto, están en un flujo de constante intercambio energético, de ideas, conceptos, propuestas, donde “las técnicas actúan como modificaciones de las formas”. (Latour, 2001)

Partir de esta perspectiva donde cada uno de los agentes implicados juega un papel de afectación directa sobre los otros, deja ver que sería imposible pensar en un desvinculamiento entre ellos para la generación de un producto artístico. En este sentido, el caso del artista-programador que se posiciona como el creador absoluto de la obra resulta un tanto ficticio, aunque en contextos artísticos, por ejemplo, los que se sitúan en la producción museística es común que esto siga ocurriendo. En este ejemplo solo vemos en la ficha de la obra el nombre del artista, la técnica empleada y el año de producción, más no el nombre de los técnicos y programadores que hicieron posible el desarrollo del proyecto. Si vamos más a fondo, podemos ir destapando cada una de las cajas negras invisibilizadas que a su vez contienen más cajas negras implicadas en la concreción de una obra artística. Desde las librerías usadas, las librerías que usan esas librerías, los lenguajes de programación superpuestos unos sobre otros, los desarrolladores que crearon las librerías, y todo el ensamblaje de agentes intermediarios que se adhieren al proceso de producción.

## 2.5. Referencias

- Baalman, M. (2015). Embodiment of code.
- Kleiman, A (2012). Interactions: an interview with Karen Barad, Mousse Magazine, (34) 76-81.



Kowalski, R. (1979). Algorithm = logic + control. *Communications of the ACM*, 22(7), 424–436.

Rheinberger, H.-J. (1998) Experimental systems, graphematic spaces. In T Lenoir (Ed.), *Inscribing Science: scientific texts and the materiality of communication* (pp 285-303). Palo Alto: Stanford University Press.

Manovich, L. (2013). *Software takes command*. New York: Bloomsbury Academic. Parisi, L. (2013). *Contagious architecture: computation, aesthetics, and space*. Cambridge, MA:MIT Press.

Rheinberger, H.-J. (2013). Forming and Being Informed: Hans-Jörg Rheinberger in conversation with Michael Schwab. In M. Schwab (Ed.), *Experimental systems. future knowledge in artistic research* (pp. 198–219). Leuven: Leuven University Press.

Schwab, M. (Ed.). (2013). *Experimental systems. future knowledge in artistic research*. Leuven: Leuven: Leuven University Press. Xambó, A., Lerch, A. y Freeman, J. (2019). Music Information Retrieval in Live Coding: A Theoretical Framework. *Computer Music Journal*, 42(4), 9-25.

Soon, W. y Cox, G. (2020). *Aesthetic Programming: A handbook of software studies*. Open Humanities Press.

# Capítulo 3

## SubCap2

bla bla bla...

### 3.1. si necesitamos subsecciones

bla bla bla...

# Capítulo 4

## SubCap3

bla bla bla...

### 4.1. si necesitamos subsecciones

bla bla bla...

# Capítulo 5

## SubCap4

bla bla bla...

### 5.1. si necesitamos subsecciones

bla bla bla...

# Capítulo 6

## Conclusiones

Blablabla...

### 6.1. si necesitamos subsecciones

bla bla bla...

# Capítulo 7

## Anexos

Por si necesitamos anexos...

### 7.1. si necesitamos subsecciones

bla bla bla...

./bib/bib1