

**1**

# **SUBJETIVIDADES TECNOARTÍSTICAS: INTERVENCIÓN IMAGINATIVA**

*Carolina Di Prospero*

**diproser@gmail.com /**

Doctora en Antropología Social.

Investigadora Posdoctoral (CONICET) Instituto de Altos Estudios Sociales.

Universidad Nacional de San Martín (IDAES-UNSAM). San Martín. Buenos Aires. Argentina.

## RESUMEN

El objeto de estudio en este trabajo es el *Live Coding*, expresión musical digital desarrollada por un colectivo artistas computacionales, programando música en vivo. El *live coder* como programador artista, dialoga en un ida y vuelta con el lenguaje de programación que utiliza en sus performances, a través de la manipulación de los códigos en vivo (*on the fly*). En ese diálogo se produce la intervención imaginativa que los constituye como *live coders*: híbrido lenguaje-artista constituido en el momento de la improvisación.

La intención exploratoria y experiencial caracteriza esta práctica tecnoartística. El artista y el software se hacen uno, se confunden, y los códigos tipeados en ese aquí y ahora se proyectan sobre el cuerpo del *live coder*, quien los porta como una máscara que los cubre durante toda la performance.

En este artículo se aborda la subjetividad *live coder*: como artista computacional, como agente activo que subsume varios roles en una misma persona buscando eliminar las barreras entre quienes crean los lenguajes de programación y quienes ejecutan la música. Asimismo se analizará cómo a partir de la improvisación de la música algorítmica, los *live coders* construyen tanto su expresión artística como su autocomprensión como programadores artistas.

## PALABRAS CLAVE

- > música algorítmica
- > improvisación
- > nuevas subjetividades
- > intervención imaginativa
- > hibridación

**ABSTRACT**

The object of study in this work is the musical expression called *Live Coding*, developed by a group of computer artists, programming live music. The live coder establishes a dialogue in feedback with the programming language he uses in his performances, through the manipulation of live codes (on the fly). In this dialogue, an imaginative intervention that constitutes them as live coders takes place: hybrid language-artist are constituted at the moment of improvisation.

The exploratory and experiential intention characterizes this techno-artistic practice. The artist and the software become one, they are confused, and the codes typed in that here and now are projected onto the body of the live coder, who carries them as a mask that covers them throughout the performance. In the present paper the live coder subjectivity is approached: as a computational artist, as an active agent that subsumes several roles in the same person, seeking to eliminate the barriers between those who create programming languages and those who execute music. We will analyze how, from the improvisation of algorithmic music, live coders build both their artistic expression and their self-understanding as programmers artists.

**KEYWORDS**

- > algorithmic music
- > improvisation
- > new subjectivities
- > imaginative intervention
- > hybridization

## INTRODUCCIÓN. EL CONTEXTO

*Un lenguaje de programación bien escrito es el lugar de descanso ideal para su autor. Los programas se originan en la imaginación humana, así es que los programadores conocen sus creaciones extremadamente bien. La imaginación del programador se basa en algo intangible. Los pensamientos toman forma de código, fluyen y se convierten en fluido una vez más durante la ejecución<sup>1</sup>*

**(Alex, live coder)**

Este artículo surge a partir de un estudio etnográfico sobre la construcción de subjetividades y sociabilidades en colectivos tecnoartísticos. El trabajo de campo fue realizado en Inglaterra entre los meses de febrero a septiembre de 2013, junio y julio de 2014, y de junio a agosto de 2015, ha sido desarrollado principalmente a partir de la técnica de observación participante en performances en vivo, muestras/trabajos interdisciplinarios, workshops académicos con participación de *live coders*; jornadas laborales de los sujetos de investigación y otros espacios, tanto presenciales como en forma *online*. Los sujetos de la investigación fueron veinte *live coders* en su mayoría residentes en Inglaterra, pero provenientes de distintos países, entre ellos México, Colombia, Alemania, Eslovenia, EEUU, etc. En este artículo se aborda la subjetividad del *live coder*: como artista computacional (*computational artist*) y como agente activo que subsume varios roles en una misma persona buscando eliminar las barreras entre quienes crean los lenguajes de programación y quienes ejecutan la música. Asimismo se analizará cómo a partir de la improvisación de la música algorítmica, los *live coders* construyen tanto su expresión artística como su autocomprensión como programadores artistas.

El campo artístico, en términos de Bourdieu, posee las características de todo campo social:

Tiende a conseguir de quienes entran en él que tengan esta relación con el campo que llamo *illusio*. Pueden querer trastocar las relaciones de fuerza en ese campo, pero precisamente por ello, conceden reconocimiento a los envites. [...] Entre las personas que ocupan posiciones opuestas en un campo y que parecen radicalmente opuestas

<sup>1</sup> Traducción propia.

en todo, existe un acuerdo oculto y tácito sobre el hecho de que vale la pena luchar por cosas que están en juego en el campo. (Bourdieu, 1997: 142-143)

El campo artístico y sus diferentes formas de exploración suele ser un prisma interesante para analizar cambios culturales y subjetividades en emergencia. El colectivo conocido como *live coding*, en adelante LC, es parte del campo de los defensores y entusiastas del software libre. Es por ello que al abordar el estudio de un colectivo artístico que desde principios de los años 2000 explora el cruce de la tecnología y la música a partir del *live coding*, es importante realizar una referencia general al software libre como antecedente a nivel tecnológico y como contexto de experiencias compartidas por los miembros de este grupo.

Tempranamente, los actuales *live coders* se han convertido en programadores de software libre y, más tardíamente, en su época universitaria y de posgrado, en artistas que exploran tanto el mundo de la tecnología como el de la música algorítmica, incursionando en ambos mundos desde la improvisación. La incansable experimentación y aprendizaje forma parte de la misma música que se produce durante la actividad del *live coding*, deviene y es parte de las prácticas de programación propagadas en colectivos que estudian la apertura de sistemas operativos (*free open source software*). En ese sentido el LC es vivido como una experiencia de aprendizaje y exploración musical tecnológica, como una expresión de programadores que se convierten en artistas.

Como señala Adrian Mackenzie en *The performativity of the code: software and cultures of circulation*, el software había sido fuertemente mercantilizado en los años '80:

Las computadoras pasaron de ser dispositivos de aficionados a computadoras de oficina, sin embargo, el software se ha resistido a la mercantilización de varias maneras. (Mackenzie, 2005: 72)

Entre los programadores, la producción y consumo de software ha sido producto de una intensa identificación y estilos personales: divididos en subgrupos que corresponden a plataformas específicas (programadores de Unix versus programadores de Windows, por ejemplo). El mundo del software está atravesado por estándares globales y al mismo tiempo es anárquicamente polimórfico y

mutable (Mackenzie, 2005: 72). El mundo del software libre ha girado en torno al desarrollo de proyectos independientes y Linux es frecuentemente mencionado como el principal ejemplo de software libre o software de código abierto.

El sistema operativo Linux también puede ser analizado como una forma de agencia colectiva:

En este sentido los desarrolladores que utilizan Linux están enmarcados en un proceso que los constituye como parte de un tipo específico de programadores que comparten prácticas mediadas de interacción colectiva que incluyen: codificar, programar, distribuir o configurar un sistema operativo. (Mackenzie, 2005: 73-78)

Por ejemplo ser un *hacker* que trabaja con Linux significa, en cierta forma, ver al sistema operativo como más que un simple objeto a ser aprehendido cognitivamente, se trata más de una forma de encarar el trabajo desde normas que desafían la propiedad y de una organización cooperativa del trabajo. (Mackenzie, 2005: 81)

Karim Lakhani y Robert Wolf (2005) diferencian entre «motivaciones externas» (conseguir mejores trabajos, avanzar en el desarrollo profesional, por ejemplo); «motivaciones internas»: el disfrute, el sentirse creativo; y «motivaciones basadas en las normas de la comunidad» (Lakhani y Wolf, 2005: 2), refiriéndose al grupo de personas dedicadas a desarrollar proyectos de software libre *free open source software (F/OSS)*:

En los proyectos F/OSS vemos un fuerte sentido de identificación con la comunidad y adhesión a las normas de comportamiento. Los participantes del movimiento F/OSS exhiben identidades colectivas fuertes. Poseen textos canónicos como *The Jargon File*, *The New Hacker Dictionary* (Raymond, 1996), *The Cathedral and the Bazaar* (Raymond, 1999), y *General Public License (GPL)* (Stallman, 1999) han creado significados compartidos acerca de las identidades colectivas y las responsabilidades de la membresía. De hecho, el término *hacker* es una insignia de honor entre la comunidad F/OSS, opuestamente a lo que sucede en los medios de comunicación masiva. La identidad *hacker* incluye resolver problemas, divertirse y compartir código, todo al mismo tiempo. (Lakhani y Wolf, 2005: 5-6, traducción propia).

Una de las motivaciones principales tiene que ver con contribuir en la creación de un bien público, y señalan que el esfuerzo en estos proyectos constituye una parte sustancial: en promedio contribuyen 14 horas por semana, y el 60% de quienes contribuyen lo hace *ad honorem* (Lakhani y Wolf, 2005: 16).

Muchas de las características y prácticas mencionadas como la defensa de los principios del software libre, la diversión, asumir determinadas responsabilidades individuales y colectivas, son comunes entre los *live coders*. A menudo utilizan el término «*hack*» para describir sus proyectos o «*to hack*» en referencia a su forma de actuar, y esto es porque el LC es considerado por la gran mayoría de sus miembros como una actividad que rompe con reglas tradicionales, tanto de hacer música como de programar.

Los sentidos y prácticas compartidas alrededor del sistema operativo Linux hacen que siga siendo un ícono entre diversos grupos de práctica del software libre, como los *live coders*, quienes a partir de allí dieron un paso hacia el campo artístico, transformando la programación en expresión artística y el código en música.

Aquellos primeros años del sistema operativo Linux fueron contemporáneos al uso del término *cibersespacio*, como sinónimo de Internet. El ciberespacio surge en la literatura, William Gibson combinó los términos «*cibernética*» (término acuñado en la teoría del control de Norbert Wiener en 1948) y espacio. *Burning Chrome* (1982) es el nombre del cuento en el que Gibson utilizó la palabra ciberespacio por primera vez, para describir una red de dispositivos situada en el año 2030, pero el término se hizo famoso en su novela *Neuromante*, de 1984. En esta novela se describía el ciberespacio como una «realidad virtual, descorporizada, (...) formada por conglomerados y constelaciones de datos» (Jones, 2014: 27). El término mismo trascendió la literatura y fue adoptado en el lenguaje coloquial y cotidiano, pero también por la academia hacia la década de 1990, para describir un nuevo mundo a partir de la masificación del uso de Internet. A través de una vasta producción en la que podemos destacar el libro *Life on the Screen*, de Sherry Turkle, (1995) el ciberespacio fue descrito como un mundo paralelo, del otro lado de la pantalla en el cual uno se podía sumergir y navegar, desvinculándose del mundo real. La década del 2000 con la Internet 2.0 trajo otro tipo de fenómenos, como las redes sociales o dispositivos como los *Smartphones* para los cuales la metáfora de la inmersión (inmersión en la pantalla) ya no fue adecuada, entonces comenzó a hablarse, desde un sentido

opuesto, de las formas en las que emergía Internet a partir de la presencia de las pantallas en la vida cotidiana. Si la metáfora de la inmersión remitía a entrar en el mundo virtual de Internet situado adentro de computadoras con conexión, en la metáfora del emerger, es el mundo virtual el que irrumpe en forma de redes sociales, de sistemas de mensajería instantánea, y todo tipo de aplicaciones en dispositivos que no solo portamos durante todo el día, sino que siempre permanecen encendidos y cercanos, incluso cuando dormimos.

El término también surge en la literatura: «en 2007, William Gibson articula por primera vez la idea del emerger «*evertion*» del ciberespacio en la novela *Spook Country*. Situada en el año 2006, la historia se basa en el crecimiento de las redes de teléfonos móviles, marketing viral, vigilancia penetrante y un estado de total seguridad» (Jones, 2014: 22).

En su libro *The emergence of Digital Humanities*, Steven Jones rescata el término de Gibson, *evertion*, ya que, como metáfora, llama la atención sobre un proceso desordenado y desigual: «la red tiene una fuga y se derrama en el mundo», dice el autor (2014: 29). Jones argumenta que la metáfora del emerger es particularmente útil porque articula un amplio movimiento en nuestra comprensión de las redes a partir de un movimiento contrario a la *inmersión*, de la que se hablaba en *Neuromante*:

«De un mundo aparte a formar parte del mundo, de una realidad virtual a la experiencia mundana, de un mundo misterioso, abstracto e invisible, a una experiencia aún mayormente invisible (pero real). Una red de datos en la que nos movemos cotidianamente en el mundo físico. Si el ciberespacio alguna vez pareció un *otro* lugar trascendente, algún lugar diferente al que normalmente habitamos, esa relación se ha invertido conforme la red ha emergido» (Jones, 2014: 19, traducción propia).

Según Jones, si antes teníamos que sumergirnos y navegar en el ciberespacio, ahora esas redes forman parte del mundo físico cotidiano. Este cambio de época ha suscitado inquietudes desde numerosos campos conformando una «Nueva Estética»<sup>2</sup>, que para Jones tiene que ver con la emergencia:

<sup>2</sup> «New Aesthetic», con mayúscula es como está expresado en el texto de Jones.



«La irrupción de otra dimensión, ya sea en los diseños pixelados en la ropa, en la creciente presencia de drones, en el arte, en la vigilancia, en la guerra, en la composición musical usando algoritmos, o trabajos realizados en *8-bit street art*» (Jones, 2014: 47).

En cuanto a la música algorítmica el *live coding*, viene explorando en torno a la composición colectiva a distancia, a partir de un proyecto llamado *Extramuros*. Básicamente, quienes participan en la sesión *Extramuros* de *live coding* se conectan en un momento determinado del día con el fin de realizar una composición colectiva. De esta forma, cada uno se encuentra en una situación cotidiana diferente: en el trabajo, en la casa, en la biblioteca o en un bar recibiendo esa «irrupción de otra dimensión» a la que hace referencia Jones, el mundo virtual dentro del real, del mundo digital en el físico» (Jones, 2014: 47).

En la siguiente sección observaremos a los programadores que han crecido en este contexto compartiendo experiencias y sueños que llevaron a la realidad, constituyéndose en *live coders*.

### «EL LIVE CODING EXISTE»

En general, romper con las convenciones existentes y sus manifestaciones en la estructura social y artefactos materiales incrementan los problemas de los artistas y hace decrecer la circulación de su trabajo, pero, al mismo tiempo se incrementa su libertad para elegir alternativas no convencionales y para despegarse de las prácticas habituales. Si esto es cierto, podemos entender cualquier actividad como el producto de una elección entre el éxito de la facilidad convencional y los problemas no convencionales y la falta de reconocimiento (Becker, 2002: 183).

Al mismo tiempo que los *live coders* iban construyendo su «*art world*» (Becker, 1974) se iban constituyendo en programadores especiales, que se dedican a crear lenguajes de programación para hacer música en lugar de desarrollos para empresas. No ha sido un camino corto o fácil, además de abrirse paso en el campo artístico, también querían ser reivindicados como programadores artistas en el campo académico. Los medios de comunicación, por ejemplo, los descri-

bían como *geeks*<sup>3</sup>, una especie de usuarios fanáticos de las nuevas tecnologías. Cuando estas apreciaciones aparecen, los *live coders* suelen disentir muchas veces con humor, otras reflexionando (sobre cómo los ven y cómo se ven ellos como grupo) y, la mayoría de las veces de ambas formas: con humor y reflexión.

Para ilustrar tanto las dificultades en su autoconstrucción como programadores artistas, como la reflexividad que los caracteriza, veamos un ejemplo extraído de interacciones en una lista de correo que comparten cotidianamente. Un miembro de la lista compartió un link a una nota en la web del medio de comunicación británico, BBC, en la sección tecnología llamada: «*Tech Know: Programming meets music*». Se trata de una cobertura de un evento de *live coding* que la sección de tecnología de la BBC realizó en un pub de Londres. La periodista que los entrevistaba se mostraba muy interesada y curiosa ante la novedosa actividad, pero en un momento, hace un chiste un tanto irónico y compara la presentación de un dúo de *live coders* con Paul McCartney y Ringo Starr, lo cual suscitó bromas y comentarios en la lista de correo. Al final del video se ve a dos de los *live coders* allí presentes, jugando y bailando con unas caretas improvisadas con papel, al estilo de las que usan los integrantes *Anonymous*<sup>4</sup> para ocultar su identidad. Este momento final fue relatado en la crónica periodística de la siguiente forma: «en lugar de un código que resulta en sonido, resultó en movimientos de baile, fue menos alta tecnología y más una performance de arte conceptual», este comentario propició correos con quejas que dejan entrever la propia *autocomprensión live coder*.

<sup>3</sup> De acuerdo a CIO.com el origen del término *geek* viene del circo, en donde era el personaje que arrancaba a mordidas cabezas de pollos, entre otros actos igual de extraños. Resulta bastante congruente esta definición, pues es una deformación de *freak*, palabra que en inglés significa extraño o raro. El término evolucionaría con la llegada de la era digital y los *geeks* ya no arrancarían cabezas de animales sino errores de programas informáticos y videojuegos, lo cual requiere un grado de inteligencia considerable y una obsesión peculiar.

<sup>4</sup> *Anonymous* es una masa de sub-comunidades autónomas que tienen diferentes programas y proyectos de activismo que ejecutan a través de Internet. Llevan sus objetivos a la práctica mediante actividades denominadas «operaciones». En general luchan por los derechos de libertad de expresión y los programas de código abierto y su socialización en internet. Poseen una estructura de mando descentralizado. Su identidad es anónima y colectiva, y se puede distinguir en público por el uso de máscaras de Guy Fawkes, un personaje que era una persona real que luchó por sus ideales, a partir de la conspiración. Anon desarrolló un tipo de acciones conspirativas en grandes redes, a través de Internet (Coleman, 2014).

Elijo este término ya que, como explican Brubaker y Cooper:

«autocomprensión» carece de las connotaciones reificantes de «identidad». Sin embargo no está restringido a situaciones de flujo e inestabilidad. Las autocomprensiones pueden ser variables a través del tiempo y las personas, pero pueden ser estables. Semánticamente, «identidad» implica igualdad a través del tiempo y las personas; de aquí la torpeza de seguir hablando de «identidad» al mismo tiempo que se repudia la implicación de igualdad. «Autocomprensión», en cambio, no tiene conexiones semánticas privilegiadas con igualdad o diferencia» (Brubaker y Cooper, 2001: 30–67).

Observemos algunos correos relacionados a la nota periodística:

> Algún comentario sobre el artículo / video? Estoy bastante conforme por cómo fuimos retratados, aunque estaba un poco preocupado de hasta dónde podían llevarlo en dirección geeky (*live coder A*)

> Hubo una atmósfera de acusación, lo curioso es que nunca se sabe con exactitud de qué se nos acusa. Para mí las acusaciones de elitismo, falta de musicalidad, etc., son como una cortina de humo. [...] Lo siento chicos, pero es evidente que hay una necesidad de hacer posters gigantes para adornar las habitaciones de los adolescentes, luego tendremos que tener disputas artísticas, seguido por intentar carreras solos y, finalmente vendrán los rumores de adicción a las drogas. (...) Ah, y no olviden mencionar en las entrevistas que su calendario de giras no les deja tiempo para la vida en pareja, eso es siempre muy importante. (*live coder B*)

> Hicieron un gran trabajo —lo cual debe ser difícil de conseguir alrededor de ese ambiente acusatorio. Sencillamente debemos decirlo: el *live coding* existe. (Y no es tan inusual como parece (*live coder C*))

> No creo que haya necesidad de hacer una defensa. Dejen que las acusaciones soplen, en las ráfagas de viento de *live coding* real, sucediendo realmente en los pubs reales, con gente real mirando. Y también con cerveza (*live coder D*)

Los comentarios graciosos o irónicos (como el de *live coder* B sobre los grupos pop) se fueron enlazando a los más serios, (como el de *live coder* C), pero todos, más graciosos o más serios planteaban un posicionamiento ante lo que se pensaba de ellos, una reivindicación de su actividad y un pronunciamiento como colectivo: «el *live coding* existe». Aunque *live coder* D afirme que no hay una necesidad de una defensa, ellos internamente están protegiéndose de las «acusaciones» de la periodista, es decir de las opiniones externas, y defendiendo su performance en el pub, y su actividad en general.

*live coder* C fue más allá con su reflexión a partir de lo sucedido en el pub:

> Mi especulación sería la siguiente: estamos viviendo en una época donde se considera que los conocimientos y el elitismo vienen juntos. Cualquier cosa que parece tener que ver con lo formal / la matemática (a la cual mucha gente le teme desde la escuela primaria) se asocia con la élite. Por lo tanto, la programación está típicamente enmarcada en el sentido de lo necesario, pero nunca interesante, o algo a ser expuesto. Una vez que está expuesto, todas estas acusaciones: —son nerds, alardean, son elitistas—, surgen. Personalmente, me parece que la proyección del código es algo bueno, porque es un acto de resistencia contra, exactamente, la conexión entre el conocimiento (formal) y el elitismo, al mostrar que simplemente se puede hacer y disfrutar [...] (*live coder* C)

La nota periodística generó bromas y quejas que fueron más allá de la misma publicación, es decir, ésta fue un disparador de reflexiones sobre lo sucedido en el pub y lo que se publicó en el artículo periodístico. Este fragmento de conversaciones caracteriza un pasado (reciente) en el cual la subjetividad disruptiva (o *hacker*) estaba relacionada a una búsqueda de respeto en los campos de la música y la tecnología, ambos campos en donde estaban conformando la actividad del *live coding*.

Por otro lado, en la segunda intervención de *live coder* C, vemos que habla de la proyección del código, también como reivindicación del sujeto *live coder*, en el acto de resistencia contra una conexión del sentido común entre el conocimiento formal y el elitismo. Una forma de contestar a las «acusaciones» es exponer, mostrar, visibilizar el código: «*show your screens*», como reza el *ManifiestoDraft* del *live coding*. Mostrar que lo que hacen es arte y no matemática formal, o, mejor dicho, se trata de matemática formal artística, algo demasiado nuevo

que debía seguir siendo construido y defendido, con el fin de agregar al sentido formal de la matemática, el sentido artístico. De esta forma, tempranamente en su conformación como grupo y como *live coders*, sintieron que su camino era explorar esa relación con la tecnología, en el diálogo que ellos podían establecer con los lenguajes de programación para música, y constituirse en los propulsores de un cambio, asumiendo ese desafío.

## RESPONSABILIDAD AVANZADA

Como propulsores de un cambio del cual se sentían responsables, desde los inicios de su actividad los *live coders* se sintieron confiados en conformar una suerte de vanguardia, y hoy en día siguen remarcando ese rol. Así lo expresó Alan Blackwell en su presentación durante la primera Conferencia Internacional de *Live Coding* (ICLC 2015<sup>5</sup>): «los códigos están por todas partes, los *live coders* son el grupo de avanzada».

Alan Blackwell (presentado por sus pares como el primer profesor de *live coding*) remite con esta frase al papel de los *live coders* como vanguardia, en lo que respecta a manipular el código artísticamente, lo cual los coloca en un lugar de privilegio pero también de responsabilidad. Los códigos están por todas partes, en los alimentos que compramos (códigos de barras), en el cine y también en las cámaras digitales que nos vigilan, en las transacciones que realizamos con nuestras tarjetas de débito, en cada escaneo y en cada objeto del hogar que se va digitalizando en el mundo de Internet de las cosas («*internet of things*») en diferentes «grados de invisibilidad de los datos y las conexiones y objetos que nos rodean» (Jones, 2014: 69). La confianza que tienen los *live coders* en su actividad deriva de la responsabilidad que su capacidad de manipulación del código les otorga (o se han auto-otorgado). Confianza en el rol que han construido como responsables de un cambio en la percepción de la figura del programador. El giro desde una subjetividad de programador como un mero ejecutor de comandos hacia un agente que interviene responsablemente en su tarea. La responsabilidad

<sup>5</sup><http://iclc.livecodenetwork.org/2015/>

refiere a la capacidad de actuar, de intervenir activamente en la programación, desde la improvisación: «el programador interactúa con el sistema de tiempo real distribuido procesalmente mediante la modificación de código en la marcha», explica Andrew. Para otro *live coder*, Dave, esa responsabilidad está ligada a la oportunidad de realizar cambios en la realidad:

El *Live coding* presenta oportunidades maravillosas. La mitificación del código (...) en un mundo donde los algoritmos impactan cada rincón de nuestras vidas —de nuestras economías a través de las transacciones virtuales, en nuestras amistades a través de la clasificación y filtrado de las redes sociales, cuando las políticas gubernamentales se basan en las fallas profundas para captar posibilidades de las redes, etc.— es evidente que tenemos algunos problemas graves que creo que el *live coding* ayuda a abordar, en una pequeña escala (Entrevista a Dave, agosto de 2013).

Los *live coders* ven su actividad más allá del aspecto musical, como explica Andrew en una charla TEDx: «el énfasis de la performance no está sólo en la música sino en cómo ésta está siendo construida (...) se trata de presenciar el proceso compositivo, viéndolo en la pantalla. Un proceso que por lo general se mantiene oculto detrás de escena».

Expertise en el uso y maestría / destreza de ese uso, junto a la responsabilidad y confianza en el rol encarnado por los *live coders*:

Tengo confianza en mi papel, soy una música que tiene una función, aunque sea pequeña, pero quien dice en el futuro tal vez la gente esté preparada para escuchar otra música. (Alexandra, compositora y *live coder*, entrevistada por Audition Records<sup>6</sup> en julio de 2013)

Esa confianza constituye una suerte de capacidad a través de la cual para los *live coders* es posible concretar la imaginación (por medio de la improvisación) y al mismo tiempo dar cuenta de ello haciendo visible el proceso. Al hacer visible el proceso, «el *Live coding* brinda acceso a la mente del performer», es una frase bastante recurrente en el colectivo.

<sup>6</sup>Entrevista completa a Alexandra disponible en el siguiente enlace: <https://vimeo.com/73420344>

### Reflexión e imaginación, van de la mano:

Todavía pienso que mi técnica es muy básica, pero aun así, siento que todo lo que imagine lo puedo hacer. No hay límites. (Entrevista que realicé a Alexandra, abril de 2013)

El rol de mediadores activos con la tecnología es manifestado en la proyección del código, ya que es la prueba más clara de su destreza, interviniendo en los lenguajes de programación en vivo, práctica que los constituye en programadores artistas y los aleja de una subjetividad ligada a la programación convencional.

Los *live coders* insisten en este punto de compartir las ideas y pensamientos composicionales, concretamente en la improvisación y en la actuación en vivo: en la performance. Thor Magnusson en un breve artículo escrito como ponencia para una conferencia<sup>7</sup>, explica qué significa compartir esas ideas:

El *live coding* es emocionante. Fuerza al compositor a revelar su patrón de pensamiento composicional para hacer público un proceso íntimo, que puede resultar en un profundo éxito o en errores graves. [...] Si bien la velocidad en la actuación es fundamental, la mayor parte de los lenguajes apuntan a la facilidad de lectura, centrándose en cómo el diseño del lenguaje puede permitir formas eficaces de pensamiento, en términos de organización y manipulación de datos. (Magnusson, 2011: 2)

Magnusson llama a esa sesión del artículo «Pensando a través de las herramientas» (*Thinking through tools*). El objetivo del texto es mostrar, desde su experiencia, cómo el aprendizaje de un nuevo lenguaje de programación produjo un cambio en su forma de pensar, al punto que introdujo modificaciones en el lenguaje que él mismo había creado:

«Trabajar con *Impromptu* por algunos meses ha cambiado mi forma de pensar en programación y cómo resolver problemas computacionales o musicales. Gradualmente, se presentó un nuevo paisaje metafórico. Un ejemplo de esta influencia en mi pensamiento puede ser encontrado en la reciente adición que realicé a *ixi lang: the matrix*, directamente inspirada por la programación funcional.» (Magnusson, 2011: 6)

<sup>7</sup>International Computer Music Conference, University of Huddersfield, UK, 31 July – 5 August, 2011

Thor habla de dos lenguajes y dos tipos de programación (*ixi lang*: programación orientada a objetos / *Impromptu*: programación funcional) y de cómo el acercamiento y aprendizaje de un lenguaje afectó al otro. En el texto son más frecuentes las referencias a elementos computacionales que musicales, en efecto, el autor señala que:

«Un lenguaje de programación específico define el pensamiento musical del compositor, y cambia las formas de pensar a través del habituarse a aprenderlo». (Magnusson, 2011: 7)

Uno de los aspectos más importantes para la creación artística que el uso de computadoras permite, es la revelación de principios estructurales en las obras de arte.

El uso de ciertos lenguajes de programación brinda un incremento significativo en la habilidad de manejar y considerar las estructuras subyacentes en obras y sistemas de producción de arte. (Cádiz, 2012: 451)

Candy (2007) profundiza en esta idea, mencionando que una característica muy importante de la tecnología digital es que, para ser utilizada en todo su potencial, es necesario estar preparado para explicitar cualquier presunción implícita en la mente del creador. Es esta imperiosa necesidad de explicitación es la que hace al trabajo creativo desafiante y gratificante para muchos artistas (Cádiz, 2012: 451)

Thor reflexiona respecto a las dificultades de incorporar un nuevo lenguaje (el cual él nunca había utilizado), tiene que ver con incorporar una nueva forma de pensar. El hecho de lograr esa experiencia, documentarla y compartirla, remite a las inquietudes creativas, el aprendizaje en base a nuevos desafíos, la confianza en sí mismos y en su rol en el crecimiento del LC. Si bien todas estas intenciones derivadas de la actividad exploratoria que constituye al *live coding* no están expresadas en los objetivos del texto de Thor, son inferibles, como señala Bourdieu: «los agentes sociales tienen estrategias que pocas veces se manifiestan en una verdadera intención estratégica» (Bourdieu, 1997: 147).

El *live coder* experto necesita demostrarlo permanentemente, exponer la habilidad en el manejo de la abstracción: la destreza. Magnusson afirma que en *live coding*, el componente visual (la exposición del código) es un elemento



importante en la conexión con la audiencia, aunque la demostración de la destreza parecería ser otro objetivo subyacente. A veces esa conexión resulta muy difícil: la otra cara de explorar y crear a partir del trabajo artístico experimental e imaginativo es la posibilidad concreta de que al público no le guste el resultado.

Sé que tengo que hacer feliz a la gente y sé que muchas veces eso no sucede pero tengo que hacerlo. Aunque a veces la gente se vaya del lugar o luego me pregunten «¿esto es música?», en este momento no me preocupa si son o no felices (risas). Es algo que tengo que hacer aunque sea extraño para la mayoría. Tal vez confrontar sea parte de esto y de continuar rompiendo esta forma de ver las cosas. Pienso que es necesario confrontar, necesitamos continuar rompiendo paradigmas todo el tiempo. Creo que confrontar es una forma de buscar nuevos territorios. Podes pensar que no es hermoso, yo pienso que es necesario. (Alexandra, entrevistada por Audition Records<sup>8</sup> en julio de 2013).

Confrontar, romper, *hackear*, asumir nuevos retos, son prácticas, derivadas de la constante actividad exploratoria, que constituyen al *live coder* experto, el que encarna el papel de agente de cambio y lo ejecuta con responsabilidad. En ese sentido se constituyen en una especie de vanguardia, reflejada en una autocomprensión de sí mismos como agentes de un cambio radical en las formas de pensar, componer, ejecutar y disfrutar la música.

## LA IMAGINACIÓN EN EL CENTRO: SUBJETIVIDAD IMAGOTÉCNICA

Los *live coders* se suelen autodefinir como programadores y artistas o artistas computacionales. Ambos roles tienen características particulares, como vimos, no son músicos electrónicos convencionales, tampoco son solo programadores de software. Por un lado hay una diferencia de intención que tiene que ver con el desarrollo de una actividad constitutivamente experimental, les interesa

<sup>8</sup> Entrevista completa a Alexandra disponible en el siguiente enlace: <https://vimeo.com/73420344>

explorar y aprender de esa exploración más que ejecutar, crear más que reproducir y experimentar el proceso más que alcanzar el producto terminado. Por otro lado, y derivada de la manipulación de los lenguajes de programación en vivo, la destreza, es una característica altamente valorada entre los *live coders* ya que durante las performances que realizan programan instrumentos al mismo tiempo que los ejecutan, improvisando patrones condicionales de códigos que dan como resultado sonidos musicales. Desde la práctica de la improvisación, el LC se acerca a otras expresiones musicales como el jazz o el blues, sobre todo en cuanto al carácter inacabado que propicia la práctica asidua y, como consecuencia, un mayor *expertise* en la ejecución. El tiempo tiene un rol fundamental en la improvisación y sus lenguajes de programación tienen en cuenta el tiempo de reacción y acción, esto les permite improvisar interviniendo en la programación. Sin tiempo, no habría música LC. La intervención imaginativa que practican durante las actuaciones juega un papel fundamental en la construcción de la subjetividad de los *live coders*, ya que los constituye como tales.

### **IMAGINACIÓN: DE LA SOCIOTÉCNICA A LA IMAGOTÉCNICA**

La noción de «sociotécnica» nos permite referirnos a la compleja relación de ida y vuelta entre lo social y lo técnico:

«Lo técnico está construido socialmente, tanto como lo social está construido técnicamente. [...] lo sociotécnico no debe ser tratado simplemente como una combinación de factores sociales y técnicos. Es algo *sui generis*. En lugar de artefactos, nuestra nueva unidad de análisis es el «ensamblaje sociotécnico»». (Bijker, 1995: 273-274)

Desde el aspecto comunicacional una red sociotécnica evidencia una multiplicidad de canales de comunicación que propician conexiones, y una gran red de socialización que mantiene en contacto cotidiano al colectivo disperso en el mundo geográfico, en el caso del *live coding*. El ensamblaje también nos permite pensar en este tipo de expresiones musicales en un nuevo tipo de prácticas: tecnoartísticas.

En el aspecto vivencial, ya que el término nos propone pensar en distintos tipos de ensamblajes, redes, relaciones que tengan en cuenta las formas en que las sociedades experimentan los fenómenos tecnológicos y los modifican, no hay una determinación sino una constitución conjunta: socio-técnica (o tecno-artística, en el ensamblaje de las expresiones artísticas y técnicas). En ese sentido, cuando nos referimos a los sujetos y su relación con la tecnología, la lógica de constitución conjunta es la misma, las experiencias de los sujetos se dan en una lógica socio-técnica. El *live coder* Thor Magnusson en su artículo *Confesiones de un live coder*, texto «autoetnográfico» como él lo llama, ya que se auto analizó en situación de aprendizaje de un nuevo lenguaje de programación, describe lo que sucede en una improvisación de LC, casi como una fusión entre humano y máquina:

«Los artistas eligen trabajar en lenguajes que puedan ofrecer lo que ellos se proponen pero, a la inversa, también es cierto que durante el aprendizaje de un nuevo lenguaje, éste comienza a condicionar la forma de pensar del artista. Como dice Bruno Latour, no hay disparos sin una persona que apriete el gatillo, tampoco hay disparo sin un revólver. Lo que emerge es un híbrido, una persona-revólver, que deviene el actor». (Magnusson, 2011: 3, traducción propia)

«La composición con computadora requiere que el compositor tenga un claro entendimiento de técnicas de síntesis y procesamiento que utiliza (Burt, 1996; Castagne y Cadoz, 2002; Wolek, 2005). A medida que nuevas formas de procesamiento y síntesis son propuestas, el modo de componer cada obra en particular se ve afectado». (Cádiz, 2012: 454)

La tecnología en general ocupa un lugar protagónico en la actividad LC: como herramienta, instrumento, canal de comunicación entre pares, como material de expresión. Ahora bien, cuando nos referimos a la relación sujeto-máquina, en el contexto de la actuación en vivo ¿Deberíamos hablar de subjetividades *sociotécnicas* para referirnos a ese híbrido que plantea Thor?

En el *live coding*, «Para muchos de los profesionales es insoslayable enmarcar el proceso de composición de una improvisación desde la revelación al público no sólo sus habilidades musicales, de programación y de escritura, sino también de sus patrones de pensamiento» (Magnusson, 2011: 6). Los *live coders* afirman que logran «revelar» sus patrones de pensamiento en el diálogo con la máqui-

na, ese diálogo es el que el público puede ver proyectado (es decir, revelado) en la pared. El híbrido «persona–revólver» se constituye en el momento de la improvisación, desde una subjetividad que se amalgama a la técnica en una experiencia que llamaré, más que sociotécnica, *imago*técnica. El término «imago» es un término acuñado por el psiquiatra suizo Carl Gustav Jung para describir la manera en que las personas forman su personalidad mediante su individuación (Jung, 2000:118). No realizaré abordajes relacionados a la psicología, pero el término me resulta propicio en este caso por la situación de comunión descrita por los *live coders* cuando refieren a su experiencia en las performances y en la composición algorítmica.

Durante la performance se da una suerte de identificación entre el live coder y la máquina (software y hardware), el híbrido «persona–revólver», que deviene en actor (Magnusson, 2011: 3). Además del diálogo con la programación que resulta en improvisación, la proyección del código en el cuerpo del live coder es otro signo de esta autocomprensión:

**Imagen 1.**  
Live coder  
durante una  
performance,  
se observan las  
líneas de código  
proyectadas en  
el cuerpo.



**Imágenes 2 y 3º.**

*Live coders  
en diferentes  
performances  
durante la  
I Conferencia  
Internacional  
de Live Coding  
(2015)<sup>10</sup>*



<sup>9</sup> Imágenes tomadas por la autora del artículo.

<sup>10</sup> Sitio web de la I Conferencia Internacional de Live Coding (ILC 2015) <http://iclc.livecodenetwork.org/2015/>

«Cuando producimos una imagen en y con nuestro cuerpo no se trata de una imagen de este cuerpo. Más bien, el cuerpo es el portador de la imagen, o sea un medio portador. La máscara proporciona al respecto la idea más concreta. Se la coloca en el cuerpo, ocultándolo en la imagen que de él muestra. Intercambia al cuerpo por una imagen en la que lo invisible (el cuerpo portador) y lo visible (el cuerpo de la manifestación) conforman una unidad medial». (Belting, 2007: 44, traducción propia)

Si bien en el caso del LC más que hablar de estructuras de identidad, podemos afirmar que conforman una forma de ser y pensar en el aquí y ahora de la actuación, una identidad *momentánea* de performer, funciona como una unidad medial, en términos de Belting. En este sentido, retomo el concepto de autocomprensión, término que permite analizar situaciones de flujo e inestabilidad. «Las autocomprensiones pueden ser variables a través del tiempo» (Brubaker y Cooper, 2001: 30-67).

*Imago*, desde el planteo propuesto, designa la imagen interna que el live coder se forma de sí mismo durante su autocomprensión de *performer*. En la psicología de Jung, esta imagen interna es del orden del inconsciente. En el caso del *live coding* esa imagen es interna e íntima, es el *self momentáneo* de live coder o su autocomprensión de performer, como explica Thor, el LC «fuerza al compositor a revelar su patrón de pensamiento compositivo para hacer público un proceso íntimo» (Magnusson, 2011: 2). Ese *self momentáneo* también puede pensarse como un rol sobre el escenario, en términos de Goffman:

«Hay una relación entre la persona y el rol. Pero la relación responde al sistema interactivo (al marco) en el cual el rol es ejecutado y el *self* del intérprete es vislumbrado. El sí mismo (*self*) no es entonces una entidad medio oculta detrás de los eventos, sino una fórmula intercambiable para manejarse uno mismo durante esos eventos». (Goffman, 1986: 573)

En este sentido es una imagen íntima, ya que el *live coder* expone al público sus pensamientos compositivos íntimos, mientras ejecuta el rol de performer. Una imagen íntima que se exterioriza: la de los patrones de algorítmicos que se convierten en códigos que, imaginados en ese momento van conformando la improvisación y las imágenes de los códigos proyectadas en la pared. Esta subjetividad imagotécnica del live coder es híbrida: pensamientos y algoritmos,

humano y máquina («persona-revólver»). Y, dada la contingencia del momento, estamos hablando de imaginación, no de reflexiones o pensamientos pulidos, sino de improvisación:

Los programas se originan en la imaginación humana, así que los programadores conocen sus creaciones extremadamente bien. [...] Los pensamientos toman forma de código, fluyen y se convierten en fluido una vez más en la ejecución. (Alex, live coder: traducción propia)

En este sentido, la fluidez de respuesta requiere de un cierto grado de precisión en la coordinación de la percepción y la acción que sólo puede lograrse a través de la práctica. «Más que conocimiento de las reglas, es la práctica es lo que distingue al profesional del novato, es la esencia de la improvisación» (Ingold y Hallam, 2007: 12). El término que en conclusión propongo para definir la subjetividad live coder, imagotécnica, es una combinación entre imaginación y *expertise* técnico. Se trata de una imaginación en sentido amplio: relacionada a la creatividad y la destreza con los dedos, al virtuosismo, a asumir desafíos, explorar, a lograr un alto *expertise* en la ejecución de música algorítmica en vivo, etc. «Todo lo que imagine lo puedo hacer. No hay límites» (Alexandra).

## REFERENCIAS BIBLIOGRÁFICAS

- Bijker, W. y Law, J. (1995).** *Shaping Technology / Building Society: Studies in Socio-technical Change*. Massachusetts: MIT Press.
- Becker, H.S. (1974).** Art as a Collective Action. *American Sociological Review*, 39(6): 767–776.
- (2002). Art Worlds. *Cultural Sociology*, 2002: 178–189.
- Belting, H. (2007).** *Antropología de la imagen*. Buenos Aires: Katz editores.
- Bourdieu, P. (1997).** *Razones prácticas. Sobre la teoría de la acción*. Barcelona: Anagrama.
- Brubaker, R. y Cooper, F. (2001).** Más allá de «identidad». *Apuntes de Investigación del CECYP*, 5(7). Buenos Aires: Fundación del Sur.
- Cádiz, R. (2012).** Creación musical en la era postdigital. *AISTHESIS*, (52):449–475. Santiago de Chile: Pontificia Universidad Católica de Chile.
- Gibson, W. (1989).** *Neuromante*. Barcelona: Minotauro.
- Goffman, E. (1986).** *Frame Analysis. An Essay on the Organization of Experience*. Boston: Northeastern University Press.
- Ingold, T. y Hallam, E. (2007).** *Creativity and Cultural Improvisation*. New York: Berg.
- Jones, S. (2014).** *The emergence of the digital humanities*. London: Routledge.
- Jung, C.G. (2000).** *Ensayo de exposición de la teoría psicoanalítica. 4. Neurosis y factores etiológicos en la infancia. El complejo parental*. Obra completa. Volumen 4. Freud y el psicoanálisis: 132–305. Madrid: Editorial Trotta.
- Lakhani, K. y Wolf, R. (2005).** Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *Perspectives on Free and Open Source Software*. Editado por J. Feller, B. Fitzgerald, S. Hissam, and K.R. Lakhani. Boston: MIT Press.
- Mackenzie, A. (2005).** The performativity of the code: software and cultures of circulation. *Theory, Culture & Society* 22(1): 71–92. SAGE.
- Magnusson, T. (2011).** *Confessions of a live coder*. Brighton: University of Brighton.
- Turkle, S. (1997).** *Life on the screen. Identity in the age of the Internet*. New York: Touchstone Book.