

DOSIS WEB: Expansión Humano-Maquina mediante el Live Coding Análogo

Camilo Nemocón

Maestría en Ing. Sistemas y Computación
Universidad Jorge Tadeo Lozano
Bogotá, Colombia
canfcero@gmail.com

Álvaro O'Byrne

Pregrado en física
Independiente
Bogotá, Colombia
alvarobyrne@gmail.com

Laura Zapata

Maestría en Enseñanza Artística de Composición Musical en Electroacústica
Independiente
Bogotá, Colombia
laura.zapata.composer@gmail.com

Resumen

Dosis Web es una plataforma de software libre para live coding, que integra la programación algorítmica en tiempo real con el hardware hacking, ampliando las posibilidades de creación en las artes digitales contemporáneas. La plataforma permite ejecutar y modificar código durante la performance para generar audio y visuales en tiempo real, incorporando además una integración nativa con Arduino que posibilita controlar dispositivos físicos, motores, sensores, luces y actuadores directamente desde el entorno web. Esta característica convierte al código en un puente entre el mundo digital y el material, permitiendo la construcción y manipulación de esculturas cinéticas e instrumentos electroacústicos como parte del proceso creativo.

La propuesta de Dosis Web trasciende el desarrollo de una herramienta tecnológica al plantear un modelo de creación donde el live coding y el hardware hacking convergen en una metodología que integra pensamiento, código y materia. En este contexto, el código deja de ser únicamente un lenguaje de programación para convertirse en un gesto performático capaz de materializar acciones físicas, mientras que los dispositivos electrónicos funcionan como extensiones del proceso creativo. De esta manera, los movimientos y decisiones del artista se

traducen en comportamientos de máquinas y objetos que participan activamente en la construcción de la obra.

Dosis Web propone así un entorno abierto para la experimentación, la improvisación y la creación interdisciplinar, donde convergen arte, tecnología y diseño para explorar nuevas relaciones entre cuerpo, algoritmo, sonido y materia, consolidando un modelo innovador de creación audiovisual expandida y colaborativa.

Palabras Clave

Live Coding, Hardware Hacking, Arduino, Análogo, Esculturas Cinéticas.

Introducción

En las últimas décadas, las prácticas artísticas contemporáneas han experimentado profundas transformaciones derivadas de la incorporación de tecnologías digitales a los procesos de creación. La programación, la electrónica abierta, la fabricación digital y las culturas colaborativas han ampliado las posibilidades expresivas del arte, desplazando el interés desde la producción de objetos terminados hacia la construcción de procesos abiertos, experimentales y participativos (Paul, 2015; Manovich, 2001). En este contexto, la convergencia entre arte, y tecnología ha configurado nuevos escenarios donde el código, los dispositivos electrónicos y la interacción humano-máquina adquieren un papel central en la producción de experiencias estéticas. Como plantea Hansen (2006), la tecnología digital no constituye únicamente un medio de representación, sino un mecanismo que transforma la percepción, el cuerpo y la experiencia del sujeto.

Entre estas prácticas emerge el Live Coding, un movimiento artístico internacional que propone la escritura y modificación de código informático como acto performático. A diferencia de la programación convencional, cuyo propósito es desarrollar sistemas estables y funcionales, el Live Coding entiende el código como un material creativo susceptible de escribirse, transformarse e improvisarse durante la ejecución de una obra. Alex McLean (2014), uno de sus principales impulsores, define esta práctica como "tener reglas y cambiarlas mientras se siguen las reglas mismas", resaltando su carácter dinámico, abierto e improvisado. Desde esta perspectiva, el código deja de ser un proceso invisible para convertirse en un instrumento de interpretación donde la composición ocurre simultáneamente con la ejecución (Collins & McLean, 2003). La comunidad internacional resume esta filosofía mediante el principio Show us your screens, reivindicando la transparencia del proceso creativo al hacer

visible el código durante la performance y convirtiendo la programación en parte integral de la experiencia estética.

Paralelamente, el desarrollo de plataformas de hardware abierto como Arduino ha favorecido la aparición de prácticas centradas en la exploración material de los dispositivos electrónicos. En este contexto surge el Hardware Hacking, entendido no como una intervención destructiva de la tecnología, sino como un proceso de apropiación, reinterpretación y resignificación creativa de objetos y sistemas electrónicos (Collins, 2006). Estas prácticas se articulan con las culturas DIY (Do It Yourself), DIT (Do It Together) y DIWO (Do It With Others), las cuales promueven formas de aprendizaje colaborativo, experimentación abierta y producción colectiva de conocimiento (Richards, 2013; Lieberman, 2015). Desde esta perspectiva, la tecnología deja de concebirse como un sistema cerrado de consumo para convertirse en un territorio de exploración donde los usuarios construyen, modifican y reinventan sus propios instrumentos de creación.

La convergencia entre Live Coding y Hardware Hacking permite extender la programación más allá de la pantalla para situarla en el espacio físico. Los algoritmos dejan de controlar exclusivamente imágenes y sonidos digitales para accionar motores, sensores, luces, solenoides y otros dispositivos capaces de producir movimiento, sonido y comportamiento. De esta manera, el código se convierte en un puente entre el pensamiento y la materia, mientras que la máquina adquiere una dimensión performática al participar activamente en la construcción de la obra. Como señalan Blackwell y Cocker (2022), el Live Coding no solo transforma la relación entre el artista y el computador, sino que redefine la interacción entre pensamiento, acción y representación durante el acto creativo.

Es precisamente en esta intersección donde se sitúa Dosis Web, una plataforma de software libre, concebida para integrar la práctica del Live Coding con el control de dispositivos físicos mediante Arduino desde una interfaz web.

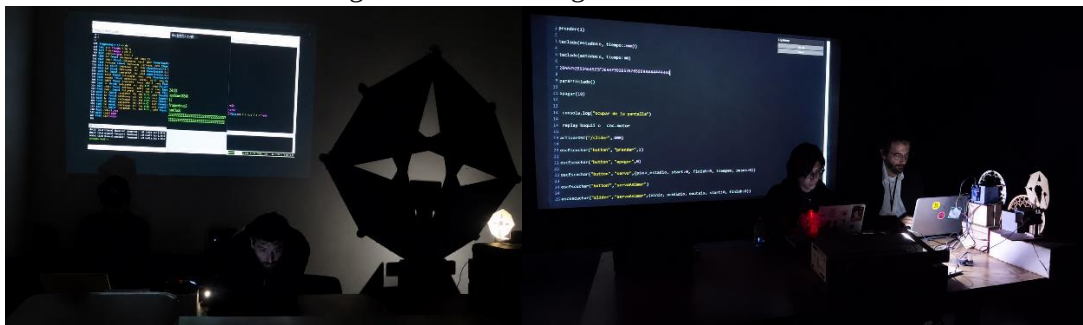
Figura 1. Interfaz de Dosis Web



Fuente: <https://toplapbogota.github.io/dosisweb/>

La plataforma permite ejecutar y modificar algoritmos en tiempo real mientras controla simultáneamente elementos sonoros, visuales y dispositivos electrónicos, ampliando significativamente las posibilidades del performance audiovisual contemporáneo. Gracias a esta integración, motores, sensores, LED, actuadores y esculturas cinéticas pueden responder inmediatamente a las instrucciones escritas por el performer, generando una relación continua entre código, espacio, sonido y movimiento.

Figura 2. Live Coding con Dosis Web



Fuente: Sesión de Live Coding en la Cinemateca de Bogotá, 2025

Más allá de su dimensión tecnológica, Dosis Web propone una metodología que integra programación, construcción de dispositivos físicos e improvisación performática dentro de un mismo proceso creativo. La plataforma invita a comprender el código no únicamente como lenguaje computacional, sino como una forma de pensamiento capaz de dialogar con materiales, objetos y máquinas. Desde esta perspectiva, la construcción de esculturas cinéticas, instrumentos electroacústicos y dispositivos experimentales se convierte en una extensión natural de la escritura algorítmica, favoreciendo procesos donde el conocimiento emerge mediante la práctica, el ensayo, el error y la experimentación colectiva.

Metodología

Esta investigación surge del interés por establecer un puente entre dos prácticas que históricamente han evolucionado de manera paralela dentro de las artes mediales: el Live Coding, centrado en la improvisación algorítmica para la generación de sonido e imagen, y el Hardware Hacking, orientado a la apropiación, modificación y construcción de dispositivos físicos como medio de experimentación artística. Aunque ambos campos comparten una

filosofía basada en el software libre, la experimentación y la cultura Do It Yourself (DIY), sus espacios de desarrollo rara vez convergen en una misma práctica performática.

A partir de esta tensión, la investigación plantea una pregunta central: ¿cómo establecer puntos de encuentro creativos entre lo digital y lo análogo mediante la programación en vivo? Esta pregunta conduce a otras inquietudes que orientan el desarrollo metodológico del proyecto: ¿cómo co-crear con las máquinas? ¿Cómo aprender colaborativamente mediante la experimentación, la práctica y el error? ¿Cómo reformular las relaciones entre pensamiento, código y materia? Estas preguntas no buscan respuestas definitivas, sino abrir un campo de exploración donde el desarrollo tecnológico y la creación artística se alimentan mutuamente.

En consecuencia, la metodología no se plantea como un proceso lineal de diseño y validación tecnológica, sino como un proceso de investigación-creación, en el que la construcción de la herramienta y la práctica performática evolucionan simultáneamente. En este sentido, la investigación se aproxima a los planteamientos de Research through Design, entendiendo que el conocimiento puede producirse mediante la creación de artefactos y la reflexión derivada de su uso (Frayling, 1993; Zimmerman, Forlizzi & Evenson, 2007). Dosis Web no constituye únicamente el resultado del proyecto; es, al mismo tiempo, el laboratorio desde el cual se investiga la relación entre programación, materialidad y performance.

El proyecto Dosis Web, entiende el código como un material de experimentación y no como un conjunto de instrucciones destinadas exclusivamente al computador. Siguiendo a McLean (2011), el Live Coding propone que la composición ocurre durante la ejecución, permitiendo que la escritura algorítmica sea un proceso abierto donde decisiones, improvisaciones, errores y accidentes forman parte de la obra. Sin embargo, esta investigación propone extender dicho principio hacia el mundo físico: si el código puede modificarse continuamente durante una performance, también deberían poder transformarse, en tiempo real, los objetos que ese código controla. Así, la improvisación deja de ocurrir únicamente sobre imágenes y sonidos digitales para expandirse hacia motores, sensores, esculturas cinéticas e instrumentos electroacústicos.

Esta ampliación metodológica encuentra fundamento en las prácticas de Hardware Hacking, entendidas como procesos de resignificación tecnológica donde los dispositivos electrónicos dejan de ser objetos cerrados para convertirse en instrumentos abiertos de exploración. La construcción de máquinas sonoras, esculturas cinéticas y dispositivos experimentales no responde únicamente a una necesidad técnica, sino a una postura crítica frente a la tecnología, donde construir constituye también una forma de pensar, de esta forma el diseño de instrumentos y artefactos análogos experimentales permite producir conocimiento a partir de la interacción directa entre cuerpo, materiales y electrónica.

Figura 3. Construcción de esculturas cinéticas



Fuente: Fotos de Lorena Sanchez, 2026

Bajo esta perspectiva, Dosis Web fue concebida como una plataforma abierta capaz de articular ambos territorios. Su desarrollo se realizó mediante ciclos iterativos donde programación, construcción física y performance se retroalimentaban continuamente. Cada nueva función del software surgía de una necesidad performática identificada durante la experimentación con dispositivos físicos, mientras que cada nuevo objeto construido modificaba las posibilidades expresivas de la plataforma. En consecuencia, el software no evolucionó independientemente del hardware, ni el hardware fue concebido como un simple periférico del software; ambos constituyeron un único sistema de investigación.

Este proceso metodológico estuvo guiado por una lógica de aprender haciendo, característica de las culturas DIY, DIT (Do It Together) y DIWO (Do It With Others). En lugar de transmitir conocimientos cerrados sobre programación o electrónica, la metodología propone escenarios donde el conocimiento emerge mediante la práctica compartida, la construcción colectiva, el ensayo y el error. El error deja de entenderse como una falla para convertirse en una estrategia de descubrimiento, mientras que la colaboración sustituye la lógica tradicional de experto-aprendiz por una comunidad de práctica donde artistas, programadores, diseñadores y makers construyen conjuntamente herramientas, dispositivos y experiencias performáticas.

Desde esta perspectiva, el concepto de Coreografías Humano-Máquina deja de ser únicamente una categoría teórica para convertirse en un principio metodológico y ontológico. La coreografía no describe solamente el movimiento de los dispositivos, sino las relaciones que se establecen entre personas, algoritmos, objetos y materiales durante el proceso creativo. En este sentido, la propuesta dialoga con la Teoría del Actor-Red (Latour, 2005), según la cual tanto los seres humanos como los elementos no humanos participan como actantes dentro de una red dinámica de relaciones. La agencia no reside exclusivamente en el sujeto humano, sino

que emerge de las interacciones entre múltiples entidades que, de acuerdo con su posición y capacidad de mediación dentro de la red, ejercen diferentes niveles de influencia sobre el desarrollo del proceso.

El performer escribe código; el algoritmo interpreta decisiones; Arduino traduce instrucciones digitales; motores, luces y actuadores responden físicamente; y el comportamiento de estos dispositivos modifica nuevamente las decisiones del performer, generando un ciclo continuo de retroalimentación. La creación deja así de localizarse en un único agente para distribuirse entre múltiples entidades que negocian constantemente la forma final de la obra.

Desde la perspectiva de Latour, estas relaciones no obedecen a jerarquías fijas, sino a asociaciones que se configuran, transforman y reconfiguran continuamente. Sensores, algoritmos, datos, dispositivos electrónicos, cuerpos y performers actúan como mediadores dentro de una red de actantes que, leída desde una perspectiva deleuziana, adquiere una configuración rizomática, caracterizada por conexiones múltiples, no jerárquicas y en constante transformación y permanente devenir, donde cada actante contribuye, en distinta medida, a la estabilidad o transformación del ecosistema creativo. Como señala Latour, "la naturaleza de los grupos no es fija, sino que se forman y disuelven constantemente. Las acciones no son solo humanas, sino que emergen de redes de actantes."

Dosis Web se configura así como una plataforma donde código, materia, sonido y movimiento convergen en un mismo proceso performático, permitiendo explorar las Coreografías Humano-Máquina como una práctica artística fundamentada en la co-agencia, la experimentación y la creación colectiva. La investigación de Dosis Web se desarrolló mediante un proceso iterativo de investigación-creación compuesto por cuatro fases: conceptualización, desarrollo tecnológico, experimentación material y experimentación performática. A diferencia de un proceso lineal de desarrollo de software, estas fases evolucionaron de manera interdependiente, permitiendo que las decisiones conceptuales, el diseño de la plataforma y la construcción de dispositivos físicos se retroalimentaran continuamente. En este sentido, el software no fue entendido como un producto final, sino como un artefacto de investigación cuyo desarrollo estuvo estrechamente vinculado a la práctica artística.

La primera fase correspondió a la conceptualización del sistema. Se realizó una revisión de referentes asociados al Live Coding, el Hardware Hacking, las interfaces musicales digitales y las prácticas colaborativas de software libre, con el propósito de identificar las limitaciones de las plataformas existentes para controlar dispositivos físicos durante una performance. Asimismo, se analizaron los resultados obtenidos con Dosis App, primera versión del proyecto desarrollada en Processing en 2018, cuyos resultados fueron presentados en la International

Conference on Live Coding mediante los trabajos DOSIS Processing Live Coding Interface (Nemocón, 2019) y Full-Body Interaction for Live Coding (Olaya-Figueroa, Zapata-Cortés & Nemocón-Farfán, 2019). Este análisis permitió definir como objetivo el desarrollo de una plataforma web abierta que integrara Live Coding y Hardware Hacking dentro de un mismo entorno creativo.

La segunda fase estuvo orientada al desarrollo tecnológico de Dosis Web. La plataforma fue implementada como una aplicación web de software libre utilizando HTML, CSS y JavaScript, privilegiando tecnologías abiertas que favorecen la accesibilidad, la reutilización y la apropiación tecnológica por parte de comunidades creativas (Kelty, 2008; Eglash et al., 2004). Desde su concepción, el proyecto buscó reducir las barreras de acceso propias de muchas plataformas de Live Coding, eliminando la necesidad de instalar software especializado y trasladando la totalidad del proceso creativo al navegador web.

La arquitectura del sistema se estructuró alrededor de diferentes bibliotecas especializadas. El editor de código fue implementado mediante CodeMirror (Haverbeke, 2012), proporcionando resaltado de sintaxis, edición dinámica y una interfaz optimizada para la escritura y modificación de código en tiempo real, aspectos fundamentales para la improvisación performática.

Para el control de dispositivos físicos se integró la plataforma Johnny-Five (Waldron, 2012), una biblioteca de JavaScript orientada a la robótica y al Internet de las Cosas (IoT), la cual utiliza el protocolo Firmata para establecer comunicación con placas Arduino. Sobre esta infraestructura se desarrolló una API propia que abstrae la complejidad técnica de Johnny-Five y simplifica el control de motores, sensores, servomotores, LED, relés y actuadores, permitiendo que dichos dispositivos sean manipulados mediante instrucciones escritas durante la performance.

Una de las principales innovaciones técnicas consistió en incorporar la API Web Serial de los navegadores modernos, posibilitando la comunicación directa entre el navegador y Arduino. Esta decisión elimina la dependencia de aplicaciones nativas basadas en Node.js y convierte al navegador en un entorno completo para la programación en vivo y el control de hardware, facilitando su utilización en contextos educativos, artísticos e investigativos.

El desarrollo de Dosis Web también contempló la integración con otras plataformas de Live Coding. La aplicación utiliza hydra-synth, motor gráfico desarrollado para Hydra (Jack, 2022), permitiendo generar visuales en tiempo real mediante código. Gracias a una arquitectura basada en un monorepositorio, Dosis Web puede ejecutarse tanto como aplicación

independiente como extensión de Hydra, favoreciendo la interoperabilidad entre diferentes entornos creativos.

De manera complementaria, la plataforma incorpora comunicación mediante el protocolo Open Sound Control (OSC) utilizando la biblioteca BrowserGlue (Silvani, 2021). Esta implementación permite intercambiar mensajes entre aplicaciones web y software especializado mediante WebSockets, facilitando la sincronización con herramientas como SuperCollider (McCartney, 2002) y Strudel (Roos, 2017). Esta capacidad amplía significativamente las posibilidades performáticas del sistema, al permitir que diferentes aplicaciones compartan información en tiempo real para coordinar procesos de síntesis sonora, generación visual y control de dispositivos físicos.

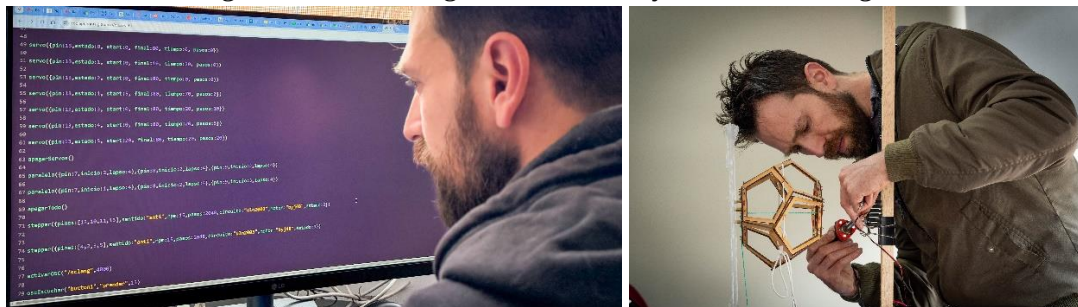
La tercera fase estuvo orientada a la experimentación material mediante la construcción de esculturas cinéticas, instrumentos electroacústicos y dispositivos interactivos. En esta etapa, el Hardware Hacking fue entendido como un proceso de resignificación tecnológica donde motores, mecanismos reciclados, sensores y componentes electrónicos dejaron de cumplir funciones utilitarias para convertirse en interfaces performáticas. Estas exploraciones permitieron comprobar que el comportamiento físico de los objetos constituye también una forma de programación expandida, en la que los algoritmos se manifiestan como movimiento, vibración, luz y sonido.

Finalmente, la cuarta fase correspondió a la experimentación performática, desarrollada mediante sesiones de improvisación donde software, hardware y performer evolucionaban simultáneamente. Durante cada sesión se modificaban algoritmos, configuraciones electrónicas y comportamientos físicos en función de los resultados observados durante la ejecución. Esta dinámica responde a los principios del Live Coding, donde la composición ocurre durante la interpretación y el código permanece abierto a la transformación constante (McLean & Wiggins, 2010). Cada performance funcionó simultáneamente como espacio artístico y como escenario de validación tecnológica, permitiendo identificar nuevas necesidades expresivas que retroalimentaban el desarrollo de la plataforma.

En consecuencia, tanto el software como las esculturas cinéticas, los instrumentos electroacústicos y los algoritmos desarrollados durante la investigación constituyen dispositivos de conocimiento que permitieron explorar nuevas relaciones entre pensamiento, código y materia. Dosis Web se configura así como una plataforma en la que la programación, la electrónica abierta y la performance convergen en un mismo proceso de investigación-creación, posibilitando el desarrollo del Live Coding Análogo como una práctica artística que

integra el Live Coding y el Hardware Hacking en un entorno de experimentación, improvisación y creación colectiva.

Figura 4. Live Coding con Dosis Web y artefactos análogos



Fuente: Fotos de Lorena Sanchez, 2026

Resultados

El desarrollo de Dosis Web permitió demostrar que es posible integrar, dentro de un mismo entorno de creación, dos prácticas que tradicionalmente han permanecido separadas: el Live Coding y el Hardware Hacking. Más allá de constituir una plataforma para la programación en vivo, la investigación evidenció que el código puede extenderse hacia el espacio físico y convertirse en un medio para controlar dispositivos electrónicos, esculturas cinéticas e instrumentos electroacústicos, ampliando significativamente las posibilidades expresivas del performance audiovisual.

Uno de los principales hallazgos consiste en que Dosis Web facilita los procesos de improvisación al integrar en una única interfaz la generación de sonido, visuales y el control de hardware. A diferencia de otros entornos de Live Coding, donde la interacción es principalmente a la síntesis audiovisual, la plataforma incorpora Arduino como una extensión del lenguaje de programación, permitiendo que cada modificación realizada sobre el código produzca simultáneamente transformaciones sonoras, visuales y físicas. Esta integración favorece una improvisación expandida donde el performer no solo compone con algoritmos, sino también con movimiento, luz y artefactos análogos.

La investigación también evidenció una expansión de las posibilidades expresivas del Live Coding. Durante las sesiones de experimentación, el código dejó de operar únicamente como un mecanismo para controlar procesos digitales y pasó a funcionar como un lenguaje capaz de accionar máquinas análogas en tiempo real. Como resultado, las composiciones audiovisuales

adquirieron una dimensión material en la que los algoritmos se manifestaron mediante esculturas cinéticas, objetos sonoros y dispositivos interactivos que respondían continuamente a las decisiones del performer.

Otro resultado significativo fue la posibilidad de construir instrumentos performáticos personalizados. En lugar de utilizar interfaces comerciales estandarizadas, Dosis Web permitió diseñar y desarrollar máquinas sonoras, esculturas cinéticas y dispositivos electroacústicos adaptados a las necesidades expresivas de cada performance. Esta flexibilidad favoreció procesos de apropiación tecnológica donde los participantes a talleres y laboratorios de Dosis Web, resignificaron objetos cotidianos y componentes electrónicos, transformándolos en interfaces de creación artística. En este sentido, el hardware dejó de ser un soporte técnico para convertirse en un componente activo del lenguaje performático.

Los resultados también evidenciaron que la construcción de dispositivos físicos modifica la manera en que los participantes comprenden la programación. El código dejó de percibirse como una abstracción informática para convertirse en una herramienta tangible cuyos efectos podían observarse inmediatamente en el comportamiento de motores, luces, sensores y mecanismos cinéticos. Esta relación directa entre algoritmo y materia favoreció procesos de aprendizaje basados en la experimentación, el ensayo y el error, fortaleciendo la apropiación crítica de la tecnología y la construcción colectiva de conocimiento.

Finalmente, la investigación permitió consolidar el concepto de Coreografías Humano-Máquina como una práctica artística y metodológica. Se observó que la creación no reside exclusivamente en el performer ni en el sistema computacional, sino en la interacción dinámica entre cuerpos, algoritmos, dispositivos electrónicos y objetos materiales. En este contexto, Dosis Web actúa como un mediador que posibilita la emergencia de nuevas relaciones entre pensamiento, código y materia, donde la programación en vivo se expande hacia el espacio físico y las máquinas participan activamente en la construcción de la experiencia performática.

Conclusiones

La presente investigación demuestra que es posible establecer un puente entre el Live Coding y el Hardware Hacking, integrando la programación en vivo con la construcción y control de dispositivos físicos dentro de un mismo proceso creativo. Esta convergencia amplía el campo de acción del Live, donde el código adquiere una dimensión tangible a través del movimiento, el sonido y la interacción con objetos físicos.

La integración de programación, electrónica abierta y hardware hackeado facilita la construcción de instrumentos personalizados, expande las posibilidades expresivas del performance audiovisual y promueve nuevas formas de interacción entre humanos y máquinas.

Asimismo, esta investigación plantea un esfuerzo por aproximar dos prácticas que tradicionalmente han permanecido separadas: el Live Coding y el Hardware Hacking. Los resultados obtenidos muestran que Dosis Web constituye un punto de encuentro entre ambos campos ampliando las posibilidades del performance audiovisual más allá de la pantalla y proponiendo nuevas formas de creación en la convergencia entre lo digital y lo análogo.

El desarrollo de la plataforma permitió evidenciar que la creación artística puede entenderse como un proceso de co-creación entre humanos y máquinas, donde el código, los dispositivos electrónicos y el performer participan conjuntamente en la construcción de la obra. En este contexto, la máquina deja de ser un instrumento pasivo para convertirse en un agente que responde, transforma y retroalimenta las decisiones del artista, configurando una relación de co-agencia que amplía las posibilidades expresivas del Live Coding.

Dosis Web se proyecta como una plataforma de software libre con potencial para futuras investigaciones, procesos pedagógicos y prácticas artísticas colaborativas. Su carácter abierto favorece la apropiación, modificación y expansión por parte de comunidades de artistas, programadores, diseñadores y makers, contribuyendo al desarrollo de nuevos lenguajes performáticos en la intersección entre arte y tecnología.

<https://github.com/toplapbogota/dosisweb>

Finalmente, el principal aporte conceptual de esta investigación consiste en reformular las relaciones entre pensamiento, código y materia. Dosis Web propone comprender estos elementos no como dimensiones independientes, sino como componentes de un mismo proceso creativo en el que las ideas se materializan mediante algoritmos, los algoritmos adquieren presencia física a través de objetos y máquinas, y la interacción con estos dispositivos transforma continuamente el pensamiento del performer. En este sentido, la plataforma no solo amplía el campo de acción del Live Coding, sino que propone una nueva metodología para explorar las Coreografías Humano-Máquina, abriendo posibilidades para futuras investigaciones y prácticas artísticas en la intersección entre arte, ciencia y tecnología.

Referencias

1. Manovich, L. (2001). The language of new media. MIT Press.

2. Paul, C. (2016). Digital art (3rd ed.). Thames & Hudson.
3. Hansen, M. B. N. (2006). Bodies in code: Interfaces with digital media. Routledge.
4. McLean, A. (2014). Making programming languages to dance to: Live coding with Tidal. En Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design (FARM '14) (pp. 63–70). Association for Computing Machinery. <https://doi.org/10.1145/2633638.2633647>
5. Collins, N., McLean, A., Rohrerhuber, J., & Ward, A. (2003). Live coding in laptop performance. Organised Sound, 8(3), 321–330. <https://doi.org/10.1017/S135577180300030X>
6. Collins, N. (2006). Handmade electronic music: The art of hardware hacking. Routledge.
7. Richards, J. (2013). Beyond DIY in electronic music. Organised Sound, 18(3), 274–281. <https://doi.org/10.1017/S1355771813000175>
8. Lieberman, Z. (2015). The future is handmade [Conferencia]. Eyeo Festival. <https://eyeofestival.com/talks/the-future-is-handmade>
9. Blackwell, A. F., Cocker, E., Cox, G., McLean, A., & Ward, A. (Eds.). (2022). Live coding: A user's manual. MIT Press.
10. Frayling, C. (1993). Research in art and design. Royal College of Art Research Papers, 1(1), 1–5.
11. Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. En Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 493–502). Association for Computing Machinery. <https://doi.org/10.1145/1240624.1240704>
12. McLean, A. (2011). Artist-programmers and programming languages for the arts. En C. Sommerer & L. Mignonneau (Eds.), Read_me: Software art and cultures (pp. 205–221). Aarhus University Press.

13. Latour, B. (2005). *Reassembling the social: An introduction to actor-network-theory*. Oxford University Press.
14. Nemocón, C. A. (2019). DOSIS – Processing live coding interface. En *Proceedings of the Fourth International Conference on Live Coding (ICLC 2019)*. Medialab Prado. <https://doi.org/10.5281/zenodo.3946351>
15. Olaya-Figueroa, J. F., Zapata-Cortés, L. V., & Nemocón-Farfán, C. A. (2019). Full-body interaction for live coding. En *Proceedings of the Fourth International Conference on Live Coding (ICLC 2019)*. Medialab Prado. <https://doi.org/10.5281/zenodo.3946303>
16. Kelty, C. M. (2008). *Two bits: The cultural significance of free software*. Duke University Press.
17. Haverbeke, M. (2012). CodeMirror [Software]. <https://codemirror.net/>
18. Waldron, R. (2012). Johnny-Five [Software]. <https://johnny-five.io/>
19. Jack, O. (2022). Hydra [Software]. <https://hydra.ojack.xyz/>
20. Silvani, D. (2021). BrowserGlue [Software]. GitHub. <https://github.com/munshkr/browserglue>
21. McCartney, J. (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, 26(4), 61–68. <https://doi.org/10.1162/014892602320991383>
22. Roos, F., & McLean, A. (2022). Strudel: Algorithmic patterns for the web. En *Proceedings of the Web Audio Conference 2022*. Zenodo. <https://doi.org/10.5281/zenodo.6821023>
23. Eglash, R., Croissant, J., Di Chiro, G., & Fouché, R. (Eds.). (2004). *Appropriating technology: Vernacular science and social power*. University of Minnesota Press.
24. McLean, A., & Wiggins, G. A. (2010). Live coding towards computational creativity. En *Proceedings of the First International Conference on Computational Creativity* (pp. 133–138). <https://doi.org/10.5281/zenodo.7219926>

Bibliografía

- Belting, H. (2007). Antropología de la imagen. Katz Editores.
- Blackwell, A. (2015). Patterns of user experience in performance programming. En Proceedings of the First International Conference on Live Coding (pp. 12–22). <https://doi.org/10.5281/zenodo.19315>
- Blackwell, A. F., Cocker, E., Cox, G., McLean, A., & Ward, A. (Eds.). (2022). Live coding: A user's manual. MIT Press. <https://doi.org/10.7551/mitpress/13770.001.0001>
- Collins, N. (2006). Handmade electronic music: The art of hardware hacking. Routledge.
- Collins, N., & McLean, A. (2014). Algorave: Live performance of algorithmic electronic dance music. En Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2014). http://www.nime.org/proceedings/2014/nime2014_426.pdf
- Collins, N., McLean, A., Rohrerhuber, J., & Ward, A. (2003). Live coding in laptop performance. Organised Sound, 8(3), 321–330. <https://doi.org/10.1017/S135577180300030X>
- Eglash, R., Croissant, J., Di Chiro, G., & Fouché, R. (Eds.). (2004). Appropriating technology: Vernacular science and social power. University of Minnesota Press.
- Frayling, C. (1993). Research in art and design. Royal College of Art Research Papers, 1(1), 1–5.
- Hansen, M. B. N. (2006). Bodies in code: Interfaces with digital media. Routledge.
- Haverbeke, M. (2012). CodeMirror. <https://codemirror.net/>
- Jack, O. (2022). Hydra. <https://github.com/hydra-synth/hydra>
- Kelty, C. M. (2008). Two bits: The cultural significance of free software. Duke University Press.
- Latour, B. (2005). Reassembling the social: An introduction to actor-network-theory. Oxford University Press.

Lieberman, Z. (2015, noviembre 27). Lessons for students. Medium. <https://zachlieberman.medium.com/lessons-for-students-cf1acf200ee>

Lieberman, Z. (s. f.). ofBook – Philosophy. openFrameworks. https://openframeworks.cc/ofBook/chapters/of_philosophy.html

Magnusson, T. (2019). Sonic writing: Technologies of material, symbolic and signal inscriptions. Bloomsbury Academic.

Manovich, L. (2001). The language of new media. MIT Press.

McCartney, J. (2002). SuperCollider. <https://supercollider.github.io/>

McLean, A. (2011). Artist-programmers and programming languages for the arts. En C. Sommerer & L. Mignonneau (Eds.), *Read_me: Software art and cultures* (pp. 205–221). Aarhus University Press.

McLean, A. (2014). Making programming languages to dance to: Live coding with Tidal. En *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design (FARM '14)* (pp. 63–70). Association for Computing Machinery. <https://doi.org/10.1145/2633638.2633647>

McLean, A. (2016, noviembre 18). Friday lunchtime lecture: Live coding, algoraves and opening up music algorithms [Video]. YouTube. Open Data Institute. <https://www.youtube.com/watch?v=wlvHFuzpULc>

McLean, A., & Wiggins, G. A. (2010). Tidal: Pattern language for the live coding of music. En *Proceedings of the 7th Sound and Music Computing Conference*.

McLean, A., & Wiggins, G. A. (2010). Live coding towards computational creativity. En *Proceedings of the First International Conference on Computational Creativity*. <https://doi.org/10.5281/zenodo.7219926>

Nemocón, C. (2018). DOSIS [Software]. GitHub. <https://github.com/camiloNemocon/DOSIS>

Nemocón, C. A. (2019). DOSIS – Processing live coding interface. En Proceedings of the Fourth International Conference on Live Coding (ICLC 2019). Medialab Prado. <https://doi.org/10.5281/zenodo.3946351>

Olaya-Figueroa, J. F., Zapata-Cortés, L. V., & Nemocón-Farfán, C. A. (2019). Full-body interaction for live coding. En Proceedings of the Fourth International Conference on Live Coding (ICLC 2019). Medialab Prado. <https://doi.org/10.5281/zenodo.3946303>

Paul, C. (2016). Digital art (3.^a ed.). Thames & Hudson.

Richards, J. (2013). Beyond DIY in electronic music. *Organised Sound*, 18(3), 274–281. <https://doi.org/10.1017/S1355771813000175>

Roos, F. (2017). Strudel [Software]. <https://codeberg.org/uzu/strudel>

Silvani, D. (2021). BrowserGlue [Software]. <https://github.com/munshkr/browserglue>

Suchman, L. A. (2007). Human–machine reconfigurations: Plans and situated actions (2.^a ed.). Cambridge University Press.

TOPLAP. (2004). TOPLAP Manifesto. <http://toplap.org>

Waldron, R. (2012). Johnny-Five. <https://johnny-five.io/>

Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. En Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 493–502). Association for Computing Machinery. <https://doi.org/10.1145/1240624.1240704>