



Tres Estudios Abiertos

Nuevas prácticas performáticas audiovisuales experimentales para el navegador.

Emilio Ocelotl

7 de abril de 2021

Resumen

Tres Estudios Abiertos es un proyecto de investigación que busca adentrarse en nuevas prácticas experimentales y audiovisuales para el navegador. Parte de la programación de un esqueleto de granulación audiovisual complementado con módulos personalizados de software. Atiende a niveles altos de programación para el control de software por medio de una interfaz de texto.

El proyecto busca demostrar que las lógicas de los lenguajes de programación posibilitan formas de pensamiento específicos con consecuencias estéticas que pueden extenderse y aportar a la investigación artística que involucra tecnología.

Para discutir la premisa anterior, el proyecto toma piezas como casos de estudio con características específicas: ligeras, sin instalaciones, sin referencias a dependencias de terceros, basadas en la web, distribuidas y optimizadas para

el bajo consumo de recursos de una computadora o una red de computadoras.

Hasta el momento considera a *Javascript*¹ como el lenguaje principal de los casos de estudio. De manera secundaria el proyecto se pregunta si es posible deducir ideas estéticas de las arquitecturas de los lenguajes de programación para la escritura de interfaces textuales de control.

Consideramos un tercer hilo, adicionalmente al estético y tecnológico, que corre paralelamente en bucle de investigación-creación: el investigativo. En esta hilo encontramos la relación de la escritura del texto resultante con la escritura de software. El vínculo se establece con el uso de sistemas distribuidos de control de versiones para la escritura de software y de texto. De esta manera, el documento resultante busca desbordarse del papel y de la palabra escrita por medio de una versión web.

Palabras clave— software, navegador, javascript, granulación, cyberespacio, par a par, programación estética, distribución, webAssembly, WebGL

Antecedentes

Los antecedentes de este proyecto describen la transición del desarrollo de software para la realización de sistemas interactivos a la escritura de módulos de software audiovisuales. Estas experiencias parten de la optimización y la ligereza del lado del hardware (por ejemplo, con el uso de computadoras de placa reducida como Raspberry Pi o Jetson Nano) y de sistemas ligeros, con pocas capas de abstracción y accesibles de síntesis y renderización de audio y video en el navegador.

Parte de los antecedentes están directamente relacionados con la experiencia performática de escribir de código al vuelo con fines creativos, audiovisuales y experimentales, de manera similar a como lo describen Villaseñor-Ramírez y Paz (2020) para los casos de Barcelona y Ciudad de México.

De manera directa, los siguientes proyectos están relacionados con la investigación y se convierten en casos de estudio, en algunas otras ocasiones funcionan como antecedentes directos:

- Caso de estudio. THREE.studies (EmilioOcelotl, 2020c) y en específico, las iteraciones threechn (EmilioOcelotl, 2020b) y threeBEASTs (EmilioOcelotl, 2021b)
- Caso de estudio. Anti (EmilioOcelotl, 2020a)
- Antecedente directo. Diálogo IV @ Coloquio Salvador Contreras. Artes Sonoras y Creación Musical en México: Siglo XXI (EmilioOcelotl, 2021a)
- Antecedente directo. Panorama. Escritura de espacios libres e inmersivos para el performance audiovisual (PiranhaLab, 2020)²

Algunos proyectos similares al que describe *Tres Estudios Abiertos* son:

- **Nivel Bajo:** Ruffbox (the-drunk coder, 2019), WebAssembly/Rust Tutorial³ y Flocking (continuing creativity, 2011).
- **Nivel Medio:** (khilnani, 2012), Web Audio API⁴, Tone.js⁵ y supercolliderjs⁶.

- **Nivel Alto:** Estuary (dktr0, 2019), Troop (Qirky, 2016), flock (munshkr, 2019), tilt (munshkr, 2021), LiveLab (ojack, 2020b), Hydra (ojack, 2020a), timeNot (AFrancoB, 2019) y seis8s (luisnavarrodelangel, 2020)

Planteamiento del problema

De entre los proyectos similares destacamos aquellos que son de nivel alto para responder a la pregunta: ¿Cuál es la diferencia entre los proyectos mencionados y *Tres Estudios Abiertos*?

Dos perspectivas podrían aclarar el punto de partida del proyecto. La primera es funcional y hace referencia a la solución de problemas partiendo de una comunidad que ejecuta, retroalimenta y enriquece al proyecto. La segunda, apuesta por la diversidad en el desarrollo de interfaces de control, para este caso destacamos las posibilidades de Estuary (dktr0, 2019).

El presente proyecto busca responder, en un momento anterior a la realización de módulos de software, si hay diferencias estéticas heredadas de notaciones musicales y computacionales, lenguajes de programación, estilos musicales, flujos de voltaje que desembocan en síntesis de audio / imagen (de manera similar a lo sintetizadores modulares) e incluso planteamientos críticos sobre decolonialidad.

Referimos a este conjunto de diferencias como decisiones de diseño en sintaxis de control que tienen consecuencias en la estética que resulta de controlar motores de audio y video. En este sentido, *Tres Estudios Abiertos* es una búsqueda que orbita en estas decisiones y se adscribe a la diversidad en la escritura de y con software.

De manera particular, este proyecto propone una solución para la síntesis granular audiovisual en el navegador. Aprovecha las posibilidades de procesamiento y conexión en red para generar módulos de software. Uno de los objetivos de la investigación consiste en la implementación del entramado de módulos en piezas audiovisuales para el navegador. Cada una de las piezas tendrán módulos adicionales que matizarán el esqueleto granular y que aportarán elementos tecnológicos para la diferenciación entre piezas. Del

resultado entre la escritura de software y su implementación en piezas específicas es que el sistema se afinará y se probará.

La ruta de navegación de este escrito implica tres niveles de exploración que coinciden con los niveles de los proyectos anteriormente citados como antecedentes a la investigación. Estos niveles no responden a una línea temporal.

- **Nivel bajo.-** Como parte de la búsqueda por la ligereza y el bajo consumo de recursos por parte de la computadora, el proyecto busca explorar las posibilidades de la compilación de código fuente directamente en el navegador. El uso de *Web Assembly* será el camino para esta fase del proyecto. En este nivel es posible proponer una librería granular compliada para el navegador.
- **Nivel medio.-** Implementación del esqueleto de módulos con *frameworks* dedicados previamente escritos como Web Audio API⁷, three.js⁸ y Icecast⁹. Actualmente hay dos piezas (y repositorios) que corren en este nivel: *THREE.studies* (threecln y threeBEASTs) y *Anti* (en desarrollo). Este nivel es de utilidad para bosquejar el posible comportamiento del esqueleto granular con marcos robustos y estables.
- **Nivel alto.-** Implementación de funciones de control que puedan interactuar con el esqueleto de módulos para la granulación de audio y video. Idealmente éstas conformarán la sintaxis de una interfaz de texto que será deducida de *Javascript*. La escritura de software y la referencia a otros proyectos de nivel alto permite tener una idea de cómo podría resolverse el control de librerías y de motores de audio y video. Este nivel permite explorar las posibilidades poéticas del texto como interfaz.

Si bien el proyecto puede diferenciarse en estos tres niveles, el objetivo al que apunta es la interacción del software escrito en cada fase.

Para la realización de una librería de síntesis granular audiovisual, el proyecto parte de conceptos propuestos por [Roads \(2001\)](#) y en específico aborda la

noción de escala de tiempo para realizar acercamientos o alejamientos que impliquen obra artística, tecnología y que incluso alcancen a plantear un marco de observación para lo político y lo social. De manera resumida, esta propuesta se desplaza en un continuo que nos permite observar el sonido en una dimensión microscópica, lo político y lo social en una dimensión macroscópica y las piezas artísticas como un punto intermedio entre estas dos escalas.

Los casos de estudio estarán alojados en la web. En este sentido, el proyecto pretende resolver el *backend* del proyecto y en específico, busca explorar las posibilidades técnicas y conceptuales de la distribución del web a partir del concepto par-a-par¹⁰. La aplicación concreta para las piezas de arquitecturas distribuidas apuntan a la comunicación entre nodos para reducir carga, para no depender de la centralidad de un servidor y especialmente para compartir flujos de audio y video a través de la web. El proyecto en este sentido apunta hacia la realización de un sintetizador distribuido alojado en la web (Hugo) y del uso de la noción de nodo y red para la resolución tecnológica de las piezas pero también como una forma de observar la organización social en la web.

Marco Teórico

La presente investigación parte del giro de los nuevos medios y de los estudios del software ([Manovich, 2001](#)). Como una extensión del punto de partida, la investigación se adscribe a la escritura con y sobre software ([Soon y Cox, 2020](#)). *Tres Estudios Abiertos* desemboca en el papel que juega la experiencia subjetiva y las implicaciones políticas y sociales en la programación estética ([Cox y McLean, 2012](#)).

Premisa/Hipótesis

las lógicas de los lenguajes de programación posibilitan formas de pensamiento específicos con consecuencias estéticas que pueden extenderse y aportar a la investigación artística que involucra tecnología.¹¹

Metodología

La presente investigación hace referencia a la investigación artística y en específico retoma la idea de *loop* o bucle para hablar de la relación entre investigación-creación. Esta forma de trabajo retroalimenta la escritura investigativa con la escritura investigativa y la práctica artística y de regreso. Consideramos que esta forma de proceder en la investigación revela aspectos que el distanciamiento convencional en la investigación no toma en cuenta.

En esta investigación hacemos referencia a procesos artísticos que involucran música, software y computadoras. En este sentido, las peculiaridades de la investigación se toman en cuenta para plantear un proyecto con escritura de software. El objetivo primordial del proyecto no busca realizar mediciones para establecer parámetros de ligereza o eficacia el software resultante. En todo caso busca aprovechar la lógica tecnológica de la programación y del procesamiento de información para encontrar soluciones para la investigación y la reflexión.

Uno de los aspectos que más rescatamos en este sentido es el uso del repositorio como una estrategia para trabajar con la escritura de software, un respaldo para el trabajo colaborativo y como una documentación que puede modificarse y consultarse en el tiempo. Para la presente investigación la noción de repositorio de código es central ya que nos permite adelantarnos en *las tripas del software* al mismo tiempo que atiende al código como un recurso de investigación. En este sentido la escritura y la consulta de software puede realizarse con sistemas distribuidos de control de versiones como *Git*¹². De esta manera buscamos lidiar con el carácter efímero del software y de piezas artísticas para el navegador.

Aprovechamos la lógica de los sistemas distribuidos para el control de versiones para construir un entramado “rizomático” que pueda dar cuenta, por un lado, del proceso creativo con los repositorios de las piezas que existen en la web y por el otro, el texto que conforma la investigación y que pretende discutir con la contraparte de programación. De esta manera los procesos quedan lo suficientemente abiertos para interrelacionarse sin perder delimitación y diferencia-

ción. En este sentido buscamos extender la propuesta de un momento anterior de investigación que anuncia la lógica del trabajo con sistemas distribuidos para la investigación artística con tecnología.

En este sentido la ejecución de la investigación coincide con los planteamientos y la delimitación del proceso de reflexión-creación: estos procesos implican múltiples objetivos y programas (Latour) que corren al mismo tiempo y que podemos enunciar de manera general como: tecnológico, estético y de investigación. Consideramos que la explicitación de esto

Objetivo principal

Escribir una serie de módulos interconectados de software para realizar síntesis granular audiovisual en el navegador.

Objetivos Secundarios

- Implementar el sistema en obras. Detonador de la relación entre práctica y reflexión escrita de cara a la tesis.
- Utilizar una serie de instrucciones personalizadas como interfaz textual que puedan mediar entre niveles y que discutan con el contexto cultural en el que se enmarcan.
- Realizar reflexiones a manera de documentación, teniendo como eje de vinculación conceptos tecno-sociales y estéticos.
- Aportar módulos de desarrollo tecnológico que puedan ser de utilidad para otros proyectos.
- Estudiar estrategias de colaboración y creación a distancia, por medio de la computadora y e instrumentos acústicos.
- Comparar críticamente el desarrollo tecnológico y la tecnología existente.
- Documentar los desarrollos y resultados secundarios realizados en colectivo.

Notas

¹¿Todavía es posible usar otros lenguajes de programación?

²Una versión en construcción y en línea del artículo se puede encontrar en <https://piranhalab.github.io/panorama/>

³<https://www.toptal.com/webassembly/webassembly-rust-tutorial-web-audio> (Consultado el 7 de abril de 2021)

⁴https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API

⁵<https://tonejs.github.io/>

⁶<https://github.com/crucialfelix/supercolliderjs/>

⁷“La API de Audio Web provee un sistema poderoso y versátil para controlar audio en la Web, permitiendo a los desarrolladores escoger fuentes de audio, agregar efectos al audio, crear visualizaciones de audios, aplicar efectos espaciales (como paneo) y mucho más.” https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API (Consultado el 7 de abril de 2021)

⁸“El proyecto de three.js apunta a la creación de una librería 3D fácil de usar, ligera, multinavegador, multi-propósito.” <https://threejs.org/> (Consultado el 7 de abril de 2021)

⁹“Icecast es un servidor para transmitir audio y video, actualmente soporta Ogg (Vorbis y Theora), Opus, WebM y MP3. Puede ser usado para crear una estación de radio por Internet o para correr de manera privada una rocola y muchas otras cosas.” <https://icecast.org/> (Consultado el 7 de abril de 2021)

¹⁰P2P (par a par) “La arquitectura de una red distribuida puede ser llamada Par a Par (P-to-P, P2P, ...) si los participantes comparten una parte de los recursos de su propio software (poder de procesamiento, capacidad de almacenamiento, capacidad de conexión a la red, impresoras,...) Estos recursos compartidos son necesarios para proveer el Servicio y el contenido ofrecido por la red... Estos son accedidos por otros pares directamente sin pasar por entidades intermediarias.”(Schollmeier, 2001)

¹¹Hipótesis premisa que traspola la propuesta que Ludwig Wittgenstein expresa en el *Tractatus Logico Philosophicus*: ¿Los límites de mi lenguaje (de programación) son los límites de mi mundo?

¹²“Git es un sistema distribuido de control de versiones libre y abierto diseñado para tratar con todo, desde proyectos pequeños hasta proyectos muy grandes con velocidad y eficiencia.” <https://git-scm.com/> (Consultado el 7 de abril de 2021)

Referencias

AFrancoB (2019). timenot. <https://github.com/AFrancoB/timeNot>.

continuing creativity (2011). Flocking. <https://github.com/continuing-creativity/Flocking/>.

Cox, G. y McLean, A. (2012). *Speaking Code: Coding as Aesthetic and Political Expression*. The MIT Press.

dktr0 (2019). Estuary. <https://github.com/dktr0/estuary>.

EmilioOcelotl (2020a). anti. <https://github.com/EmilioOcelotl/anti>.

EmilioOcelotl (2020b). threecln. <https://github.com/EmilioOcelotl/THREE.studies/tree/main/threecln>.

EmilioOcelotl (2020c). Three.studies. <https://github.com/EmilioOcelotl/THREE.studies>.

EmilioOcelotl (2021a). Dialogoiv. <https://github.com/EmilioOcelotl/dialogoIV>.

EmilioOcelotl (2021b). threebeasts. <https://github.com/EmilioOcelotl/THREE.studies/tree/main/threeBEASTs>.

khilnani (2012). supercollider.web. <https://github.com/khilnani/supercollider.web/>.

luisnavarrodelangel (2020). seis8s. <https://github.com/luisnavarrodelangel/seis8s>.

Manovich, L. (2001). *The Language of New Media*. Leonardo (Series) (Cambridge, Mass.). MIT Press.

munshkr (2019). flok. <https://github.com/munshkr/flok>.

munshkr (2021). tilt. <https://github.com/munshkr/tilt>.

ojack (2020a). hydra. <https://github.com/ojack/hydra>.

- ojack (2020b). Livelab. <https://github.com/ojack/LiveLab>.
- PiranhaLab (2020). Panorama. <https://github.com/pirahalab/panorama>.
- Qirky (2016). Troop. <https://github.com/Qirky/Troop>.
- Roads, C. (2001). *Microsound / Curtis Roads*. MIT Press, Cambridge, Mass.
- Schollmeier, R. (2001). A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. pp. 101 – 102.
- Soon, W. y Cox, G. (2020). *Aesthetic Programming: A Handbook of Software Studies*. Open Humanities Press.
- the-drunk coder (2019). ruffbox. <https://github.com/the-drunk-coder/ruffbox>.
- Villaseñor-Ramírez, H. y Paz, I. (2020). Live Coding From Scratch: The Cases of Practice in Mexico City and Barcelona. En *Proceedings of the 2020 International Conference on Live Coding (ICLC2020)*, pp. 59–68, Limerick, Ireland. University of Limerick.