

Universidad Nacional Autónoma de México

PROGRAMA DE MAESTRÍA Y DOCTORADO EN MÚSICA

FACULTAD DE MÚSICA

INSTITUTO DE CIENCIAS APLICADAS Y TECNOLOGÍA

INSTITUTO DE INVESTIGACIONES ANTROPOLÓGICAS

TRES ESTUDIOS ABIERTOS

ESCRITURAS AUDIOVISUALES PARA EL NAVEGADOR

QUE PARA OPTAR POR EL GRADO DE
DOCTOR EN MÚSICA
(TECNOLOGÍA MUSICAL)

PRESENTA

EMILIO OCELOTL REYES

TUTOR PRINCIPAL: HUGO SOLÍS GARCÍA

COMITÉ TUTOR: IRACEMA E ANDRADE, FERNANDO MONREAL

30 de noviembre de 2022

ÍNDICE GENERAL

1	Introducción	7
1.1	Motivaciones	9
1.2	Preguntas de investigación	11
1.3	Objetivos	12
1.4	Justificación	13
1.5	Metodología, herramientas y organización	13
2	Antecedentes	17
2.1	Antecedentes de Investigación	17
2.2	Niveles, ecosistemas y fuera de línea	18
2.3	Obras cercanas y obras lejanas	24
2.4	PiranhaLab	26
2.5	Seminario Permanente de Tecnología Musical	28
3	Marcos de trabajo	31
3.1	La ofuscación como motivo	33
3.2	Tres Estudios	37
3.3	La investigación compilada	39
4	Expresividad	41
4.1	Gesto, expresividad y performance en vivo	41
4.2	Estática/Dinámica	42
4.3	¿Es posible encontrar una solución?	44
5	Extensión y escritura	45
5.1	Modulos personalizados	45
5.2	Reconfiguración de distinciones	45
5.3	Acción e idea	46
6	Integración	47
6.1	La escritura como rodeo	47
6.2	Segunda sección	48
6.3	Tercera sección	48
7	Conclusiones	49
	Referencias	49
	Glosario	53

4 | ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Figura 1	Presentación colectiva en Coloquio	22
Figura 2	Notas de Ausencia - Marianne Teixido	25
Figura 3	Hydra en la web - Olivia Jack y Ritchse	27
Figura 4	Conversatorio CCD	27
Figura 5	Distopía - Edges 2020	29
Figura 6	Openframeworks 1	32
Figura 7	Anti portada	33
Figura 8	Anti	34
Figura 9	Foto de perfil	35
Figura 10	THREE.studies	38
Figura 11	THREE.studies - Iracema de Andrade	43

6 | ÍNDICE DE FIGURAS

1

INTRODUCCIÓN

La presente investigación es un recorrido que pregunta sobre las posibilidades artísticas de **Javascript**, algunos marcos de trabajo circundantes, y bibliotecas que permiten la renderización de audio y video. En específico, el proyecto aborda la relación entre este lenguaje de programación y las posibilidades del **Navegador Web** y sus consecuencias como piezas audioviduales. Parte de la observación del giro de los nuevos medios y pasa por los estudios del software (**Manovich, 2001**) para preguntarse sobre el papel que juegan las funciones de programación como estructuras socialmente convenientes y las posibilidades del paradigma de la programación como estructura, ejecución y metáfora.

Este trabajo en ciertos momentos se desborda hacia otros lenguajes como **C++**, **Lisp** e incluso hacia interfaces de texto personalizadas como **Tidal Cycles**. De manera específica, el proyecto se vincula con módulos de síntesis granular escritos en **Web Audio API**. Desde el punto de vista de la realización audiovisual, esta investigación parte de los planteamientos de la síntesis granular. Aborda la noción de escala de tiempo (**Roads, 2001**) para rebasar la dimensión de la computación que concierne a las señales de audio, al microsonido y a la obra para realizar observaciones sobre el ecosistema de piezas y conceptos que lo rodean.

Como ejercicio de escritura produce ramas de reflexión que se cuestionan sobre la brecha entre escritura reflexiva y escritura de código. Este proyecto persigue el equilibrio entre investigación, producción artística y escritura de software como un tipo ideal¹ del que pueden emergir puntos de encuentro para la investigación (**Shanken, 2010**) y el reconocimiento de perspectivas que conjuntan investigación, escritura de software y puesta en marcha de piezas artísticas en espacios de discusión que cuenten con diversos perfiles e intereses.

Como una rama adicional, el proyecto se pregunta sobre la narrativa de una investigación que parte de la reflexión y que no considera parámetros de medición como el núcleo del trabajo con software. Como un elemento más del ramal reflexivo, el proyecto aprovecha la discusión sobre los lenguajes de programación y los recursos audiovisuales para preguntarse sobre la reconfiguración de los

¹ La noción de tipo ideal fue propuesta por el sociólogo alemán Max Weber. Esto requiere más explicación.

formatos de investigación y el planteamiento de escrituras que permitan interrelacionar texto, ejecución y salida. El núcleo de esta investigación es un texto plano que relaciona imagen en movimiento, **renderización** en tiempo real, estructuras asistidas por datos con el texto y la imagen fija.

Partimos de la problematización de la programación desde los estudios del software y tomamos la perspectiva de Winnie Soon y Geoff Cox para hablar de la programación como práctica:

“Consideramos la programación como una práctica cultural dinámica y un fenómeno, una forma de estar y hacer en el mundo y un medio para entender algunos de los complejos procedimientos que sustentan y constituyen nuestras realidades vividas, y para actuar sobre esas realidades” (**Soon y Cox, 2020**, p. 14)

Esta investigación está acompañada de tres piezas artísticas para el navegador: THREE.studies², Anti³ y la instancia de este proyecto en el navegador⁴. La reflexión se centra en el ecosistema en el que se inscriben estas piezas como una forma de plantear la observación sobre esta realidad de una diversidad de instancias que relacionan lenguajes de programación y práctica artística. La investigación se adentra en el código como una tecnología que permite establecer un tipo de relación entre los agentes involucrados. La investigación de Di Prospero (2015) ilustra el interés por vincular una práctica técnica con procesos sociales: “La sociabilización en la actividad del live coding, se constituye, como veremos, en consonancia con la técnica, por lo cual se hace necesario un análisis que dé cuenta de la relación de los sujetos de estudio con la tecnología.”(Di Prospero, 2015, p. 48)

La ejecución y existencia de estas piezas supone un ecosistema de pensamientos, prácticas, plataformas y de otras piezas cercanas en términos tecnológicos y artísticos. La observación de estas piezas en contexto nos permitirá considerarlas como instancias de las reflexiones expuestas en esta investigación y que permitan a su vez, observar al paradigma de los lenguajes de programación como tecnología. De esta manera podemos contemplar las agencias que estas instancias

² El repositorio oficial de THREE.studies se encuentra en: <https://github.com/EmilioOcelotl/THREE.studies>. Consultados el 12 de diciembre de 2022. La última versión de la pieza se puede encontrar en: <https://three.ocelotl.cc/>

³ El repositorio de Anti se encuentra en: <https://github.com/EmilioOcelotl/anti> y el sitio con la pieza en: <https://anti.ocelotl.cc/>. La última versión de la pieza se puede encontrar en: <https://anti.ocelotl.cc/>. Consultados el 12 de diciembre de 2022

⁴ El nombre todavía está pendiente

por sí mismas tienen y que se les imputan en términos de relaciones políticas, sociales y económicas.

1.1 MOTIVACIONES

Considero que hay una distinción importante que permite explicar las motivaciones de esta investigación. La reflexión y los estudios que acompañan a esta investigación gira en torno al navegador como un dispositivo de renderización que prácticamente puede compararse como un sistema operativo en miniatura que permite realizar procesos que competen al servidor y también a la interfaz frontal⁵. El núcleo de la investigación reside en la capacidad del navegador para resolver estas necesidades y de manera secundaria o como un desbordamiento de esta posibilidad, está la conectividad a la web y las características que se desprenden. A lo largo de esta investigación estaremos hablando de este proceso primario (el navegador como renderizador y servidor) y del proceso secundario (consecuencias de la conexión a la web).

La perspectiva del **Software Libre** es una de las motivaciones más importantes de este trabajo de investigación. De manera particular, entendemos que las implicaciones del software libre y de código abierto tienen repercusiones en los resultados sonoros y visuales y especialmente, en las formas de organización social, económica y social de las personas que se involucran con lo que García llama sistemas de producción musical y que define como:

“un sistema de producción musical cuyos productos surgen de la interacción de personas que colaboran de manera distribuida, con herramientas, modos de operación y circuitos de distribución que funcionan bajo un principio generalizado de compartición y circulación libre de la cultura.”(García, 2021, p. 65)

Este proyecto parte de motivaciones que expresan formas de practicar e investigar con y sobre tecnología. Algunas de las metodologías de trabajo que parten del hacer pero que desde nuestra perspectiva, se pueden extender al pensamiento y a la reflexión, están presentes en formas de trabajo como DIY o DIWO.

Las circunstancias que permitieron el giro de esta investigación hacia el navegador estuvieron directamente relacionadas con la pandemia de COVID-19 y con el trabajo colaborativo realizado en el marco de eventos que sucedieron en el

⁵ En inglés se utiliza la dupla *front end* y *back end*

ciberespacio. El caso de los espacios virtuales para la realización de conciertos en la web se explica a detalle en la sección de antecedentes.

Una de las motivaciones que promueven esta investigación es el interés por la imagen, independientemente del sonido y acompañada de éste. La realización de piezas audiovisuales colaborativas expandió las posibilidades del trabajo con materiales digitales. Destacó el trabajo realizado con Marianne Teixido, Jessica Rodríguez, Celeste Betancur, Iracema de Andrade y Alejandro Brianza en este rubro. Las búsquedas personales en piezas audiovisuales me han motivado para encontrar soluciones a la integración entre audio e imagen: en primera instancia, recurrió a la conexión a partir de protocolos como Open Sound Control ([OSC](#)).

Considero que esta relación todavía se encuentra en una fase exploratoria, sin embargo, he encontrado en Javascript un marco de trabajo integrado que permite trabajar bajo una misma estructura compartida sin necesidad de preocuparme por el intercambio de información como una pista de realización tecnológica adicional.

El proyecto busca explorar las posibilidades del navegador como paradigma de renderizado de audio y video y en general, como un "sistema operativo" multi-plataforma. Las posibilidades de cero instalación y compatibilidad motivan técnicamente al proyecto.

Otras motivaciones son las reflexiones que resultan de vincular presencia y distancia.

Javascript es un lenguaje de programación ampliamente utilizado por la industria. Encarnar las contradicciones en la escritura de software permite a esta investigación reflexionar sobre las posibilidades de la escritura de software artístico y para el arte, sobre todo en lo que apunta hacia la diversidad tecnológica.

El proyecto busca plantear un aporte a la diversidad de marcos para realizar investigaciones con tecnología. Esta perspectiva plantearía el hacer explícitas ciertas contradicciones detectadas en la escritura investigativa y con software.

Finalmente consideramos que esta investigación puede dialogar con perspectivas que apunten al beneficio común para invertir las prioridades en lo que respecta a la escritura de software y su investigación:

"El buen vivir sumak kawsay, demanda, en esta globalidad de conocimiento, de un sumak yachay, un buen conocer, de los saberes (nuevos y viejos). Es por tanto necesario desarrollar el buen conocer, aquel que beneficia a todos, que crea un entorno rico y fértil para la vida cultural, social, económica, política. En definitiva, crear una

matriz productiva basada en conocimiento común y abierto”(Platohedro *et al.*, 2019, p. 31)

un contexto inclusivo y políticamente activado podría incursionar en reflexiones sobre el buen conocer, como un ámbito del buen vivir que apunten a la escritura de software con lenguajes de programación.

1.2 PREGUNTAS DE INVESTIGACIÓN

El presente proyecto se pregunta por los límites del pensamiento audiovisual en la web, tomando en cuenta los marcos de trabajo, las limitaciones y las posibilidades de expresividad, la extensión y la reconfiguración de los agentes involucrados en el proceso creativo con lenguajes de programación y la posible integración entre escritura como software, investigación y acto creativo.

Como consecuencia de la pregunta principal y en dialogo con la parte práctica de esta investigación, instanciada en piezas audiovisuales, se desprenden cuatro preguntas secundarias de investigación, cada una con dos ejes articuladores. A continuación se enuncian de manera resumida:

1. Marcos de trabajo

- El primer eje aborda la aparente contradicción entre la persecución de la programación eficiente y optimizada y otras búsquedas en la escritura del código que toman el error o la ofuscación como premisas.
- La restricción y el ofrecimiento como categorías de los estudios de la **Interacción Persona-Computadora** (IPC) que pueden guiar el trabajo tecnológico y el resultado artístico en piezas que usan lenguajes de programación.

2. Expresividad

- La contradicción que existe entre un tipo de programas que deben ser detenidos y recompilados para cambiar y la programación que puede cambiar dinámicamente y que puede ser transformada al vuelo.
- El papel que tiene la gestualidad y la interpretación en la dimensión digital de piezas audiovisuales que pueden ser fijas pero también que pueden ser intervenidas en el momento.

3. Extensión

- La escritura con lenguajes de programación como una posibilidad para la reconfiguración de las distinciones entre usuarios y desarrolladores.
- El papel de la diversidad tecnológica en la extensión de los marcos de trabajo y la resistencia a la escalada de recursos computacionales

4. Integración

- Reflexiones en torno al software que es arte y al software que produce arte.
- La posible resolución de la brecha que existe entre la escritura que genera programas de computadora, investigación y acción creativa.

Cada uno de estos aspectos corresponde con un capítulo de la investigación.

1.3 OBJETIVOS

Los objetivos de esta investigación son consecuencia de las preguntas antes mencionadas.

- Reflexionar sobre las posibilidades y limitaciones del lenguaje de programación Javascript para la generación de piezas audiovisuales en la web.
- Indagar sobre las posibilidades gestuales en el navegador, tomando en consideración el pensamiento implícito en Javascript y en otros casos de lenguajes que utilizan motores de audio y video.
- Escribir módulos de síntesis granular con [Web Audio API](#), [AudioWorklets](#) y [Three.js](#) e identificar el ecosistema que rodea a estos módulos.
- A partir de una versión extendida para la web de esta investigación, reflexionar sobre la brecha entre la escritura de investigación y la escritura de software.

1.4 JUSTIFICACIÓN

El proyecto busca explorar las posibilidades del navegador como paradigma de renderizado de audio y video y en general, como un "sistema operativo" multi-plataforma. Las posibilidades de cero instalación y compatibilidad motivan técnicamente al proyecto.

Otras motivaciones son las reflexiones que resultan de vincular presencia y distancia.

Javascript es un lenguaje de programación ampliamente utilizado por la industria. Encarnar las contradicciones en la escritura de software permite a esta investigación reflexionar sobre las posibilidades de la escritura de software artístico y para el arte, sobre todo en lo que apunta hacia la diversidad tecnológica. El proyecto busca plantear un aporte a la diversidad de marcos para realizar investigaciones con tecnología. Esta perspectiva plantearía el hacer explícitas ciertas contradicciones detectadas en la escritura investigativa y con software. Finalmente consideramos que esta investigación puede aportar elementos para invertir las prioridades en lo que respecta a la escritura de software y su investigación: un contexto inclusivo y políticamente activado podría incursionar en reflexiones sobre el buen conocer, como un ámbito del buen vivir.

1.5 METODOLOGÍA, HERRAMIENTAS Y ORGANIZACIÓN

Partimos de la realización de una investigación con y sobre código, esto quiere decir que hay código que complementa el proyecto y a su vez, funge como el punto de partida de reflexiones sobre las posibilidades y restricciones de los lenguajes de programación en la realización de piezas audiovisuales.

La escritura de este proyecto parte del concepto de retroalimentación para explicar el proceso de ida y vuelta entre código y texto de investigación. Este bucle tuvo varias iteraciones que se expresaron en distintas versiones de las piezas que acompañan este código y que pueden ser consultadas en los respectivos repositorios. Adicionalmente, los objetivos que se plantearon estuvieron enfocados a realizar observaciones sobre este proceso, teniendo en consideración al mismo código como interlocutor.

El texto que compone esta investigación fue compilado y adecuado de acuerdo a salidas específicas.

Para considerar esta relación en una dimensión funcional pero también poética, he elegido la palabra escritura para referir a la acción que involucra software. Esta decisión busca distinguirse de otros conceptos como desarrollo o producción de software. En general, esta decisión responde al distanciamiento del software como mercancía. Sin embargo, la investigación no reniega de los aportes que puede hacer la economía política para problematizar la escritura de programas con lenguajes de programación. De hecho retomamos la relación social que puede existir entre la escritura y el escritor como una relación que pone en evidencia las contradicciones sobre el trabajo que ambos realizan y la relación social de los escritos tecnológicos como expresiones del trabajo invertido por otros escritores en otros momentos.

El proyecto toma como punto de partida la propuesta de Geoff Cox y de manera particular, el uso de la dialéctica para abordar las problemáticas del software desde la contradicción irresuelta.

En el espíritu de las prácticas críticas que buscan transformar el aparato técnico, se enfatiza aún más cómo el pensamiento dialéctico permanece productivo para entender cómo la transformación es inherente al software. En consecuencia, se sugiere que una práctica crítica en el software art, busca revelar estas contradicciones, con especial atención al código fuente como expresión de una acción potencial.(Cox, 2010, p.14)

La relación entre usuario y computadora vislumbra un concepto que visibiliza la relación entre escritura y escritor de software.

Clarificar que la aproximación no es cuantitativa, es decir, si el proyecto toma en consideración algun tipo de medición, esto no será lo principal. No persigo la optimización como valor “artístico”. El objetivo del proyecto es explorar las implicaciones alternativas y poéticas del código. Para esto, es necesario tener una aproximación cualitativa en términos de la interpretación intersubjetiva del código y sus instancias en piezas artísticas.

La organización de los materiales de esta investigación es importante. La publicación de una sola fuente es el núcleo de este ordenamiento. El texto quedan organizado en archivos \LaTeX ⁶, lo cual posibilita compilar un documento

⁶ \LaTeX es la instancia más usada del lado de usuarios, sin embargo, vale la pena diferenciarlo de $\text{T}_{\text{E}}\text{X}$: “Para dar una idea de la diferencia entre los dos programas, se podría comparar $\text{T}_{\text{E}}\text{X}$ con un cuerpo y \LaTeX con la ropa” más popular (hecha, sin embargo, de instrucciones en lenguaje $\text{T}_{\text{E}}\text{X}$) que a lo largo de los años fue confeccionada para acercarla al público de manera amigable.”(Pantieri, 2008, p. 14).

con calidad tipográfica con esfuerzo relativamente menor ([Knuth, 1991](#)) y exportarlo a una estructura de datos compatible con una página web.

Hemos elegido [L^AT_EX](#)sobre otros posibles entornos para la escritura académica que dialoga con código por la posibilidad de compilar texto hacia otras salidas como [HTML](#), que permite el montaje del proyecto de investigación en una web. Esta investigación tiene una salida en la web pero no depende de la conexión/desconexión del usuario a internet. En este sentido hemos descartado opciones como pubpub que parece, son estandares más bien de publicaciones arbitradas. Sitios como Overleaf también han sido descartados debido a la dependencia del estado de conexión permanente a la web, lo cual dificulta la escritura del documento en situaciones fuera de línea.

Por otro lado consideramos que [L^AT_EX](#)es un estándar de investigación que puede vincularse con otras herramientas que han sido fundamentales para la escritura de código y de esta investigación. El tránsito entre lenguajes de programación y marcos de trabajo es posible gracias a editores como Emacs y sus modos como org-mode o clang-mode. En general, destacamos el uso de la terminal como recurso para la escritura, la ejecución y la compilación. También destacamos el uso de [Git](#) como un entorno para el trabajo distribuido que permite guardar el historial de cambios del proyecto.

Herramientas que cumplen con las condiciones editoriales de este proyecto.

Por último, considero que hay algunas aclaraciones necesarias para la lectura de esta investigación, tanto para la versión en papel/pdf como para la versión web.

El proyecto escrito va intercalado con imágenes, fotografías y capturas de pantalla. Ésta últimas dan cuenta de la imagen fijada, rendereada en solo cuadro de los eventos que suceden en el navegador. También es posible consultar las piezas en los enlaces que aparecen en el texto. La versión web de esta investigación permite vincular de una manera más estrecha estos recursos sin que tengan que ser consultados con imágenes incrustadas y ligas como los únicos recursos multimedia.

El glosario que se encuentra al final de este texto tiene información referente a conceptos y plataformas que de acuerdo a mi consideración, requieren

La cercanía o la distancia del código con respecto al usuario es algo que en esta investigación se problematiza desde la escritura. La presente investigación toma como referente la combinación de paquetes para el diseño editorial que usa *L'arte di scrivere con L^AT_EX*. También tiene una influencia notable del trabajo y pensamiento de *Perro tuerto* en lo que respecta al uso de software libre y de código abierto para el diseño editorial de trabajos de esta magnitud. Para mayores referencias, consultar: https://perrotuerto.blog/content/html/es/005_hiim-master.html

clarificación adicional. Es el apartado que mejor expresa la diversidad tecnológica que rodea a esta investigación.

2 | ANTECEDENTES

Los antecedentes de esta investigación aluden a una trayectoria que va de la transición de la escritura de software para la realización de sistemas interactivos a la escritura de módulos de software audiovisuales. Estas experiencias toman como premisas la optimización y la ligereza de hardware (por ejemplo, con el uso de computadoras de placa reducida como Raspberry Pi o Jetson Nano) y la elaboración sistemas ligeros, accesibles y portables para la síntesis y renderización audiovisual en el navegador.

A la par de la práctica artística con tecnología, los elementos precedentes de esta investigación se han conducido hacia la problematización del software como una instancia tanto de conocimiento como de ideas que conforman un paradigma que atraviesa todo el entramado de personas, investigaciones, escritos y obras involucrado con esta actividad. Estas reflexiones han tenido eco en el campo de investigación denominado estudios del software.

Los antecedentes a continuación descritos no necesariamente tienen un orden cronológico o de importancia. En todo caso es un compendio de perspectivas, plataformas, obras y realizaciones prácticas que desde la perspectiva de quien suscribe este texto, son relevantes de enunciar para captar las incidencias en los casos de investigación.

2.1 ANTECEDENTES DE INVESTIGACIÓN

Tres Estudios Abiertos es resultado de proyectos de investigación previos.

Objeto, Paisaje y Efecto(Ocelotl, 2015) fue un proyecto de investigación que abordó las nociones de objeto sonoro (Schaeffer, 2003), paisaje sonoro(Schafer, 1994) y efecto sonoro (Augoyard y Torgue, 2006) para considerar a la escucha como un recurso para la investigación sociológica en música y para la investigación social desde el sonido. Este último aspecto marco el salto de la investigación que parte de una observación distanciada a la intención de incorporar conceptos de otras disciplinas para la investigación en ciencias sociales pero también para la reflexión sobre el cruce entre música y tecnología. Rescató esta investigación en lo

que respecta a la postura hacker en tanto que actitud para apertura y aprendizaje con dispositivos que no necesariamente se ciñen a lo tecnológico, sino que también pueden problematizar las relaciones sociales que existen en la triada performática que existe entre compositores, escrituras tecnológicas e instrumentistas.

Un segundo punto de investigación fue *Cuidado con la brecha*. Esta investigación involucró un proceso de investigación-producción artística (*Ocelotl, 2019*). La realización de este proyecto fue un prototipo tecnológico y partió de objetivos que inicialmente estaban propuestos como secundarios pero que más tarde se revelaron como parte del núcleo en la investigación. Estos aspectos son: 1) el proceso de trabajo colaborativo y su implementación con herramientas como git, 2) la reflexión sobre la interacción entre audio e imagen en la composición musical electroacústica y 3) el uso de herramientas libres, personalizadas para la realización de prototipos audiovisuales y para el planteamiento de una observación crítica de procesos creativos donde investigador y artista son el mismo agente. La propuesta de los estudios del software fue incorporada en este momento de investigación.

Desde la óptica de la escritura de software, esta investigación conecta y retoma algunas ideas que han sido plasmadas anteriormente. Estas fueron incorporadas como elementos secundarios que acompañan el objetivo principal de esta investigación: La visibilización de las contradicciones en la escritura de software como pregunta de investigación y como motivo para la realización artística.

2.2 NIVELES, ECOSISTEMAS Y FUERA DE LÍNEA

Los proyectos cercanos tecnológica, conceptual y estéticamente a esta investigación se pueden dividir en cinco rubros: por un lado, nivel bajo, medio y alto dependiendo de la cercanía con el lenguaje de la máquina y por el otro, ecosistemas y proyectos fuera de línea.

Las plataformas y marcos de trabajo utilizados en esta investigación son problematizados a profundidad en capítulos siguientes; el resto, puede ser considerado con un conjunto de proyectos colindantes. Los casos, al ser ordenados de bajo a alto nivel, también describen una trayectoria de realización de prototipos de las piezas que forman parte de esta investigación.

Un cuarto conjunto de proyectos describen esfuerzos por mantener la diversidad sintáctica en los lenguajes de programación de alto nivel, fungiendo como

contenedores que permiten la renderización de audio y en algunos casos, de video, a partir de esta sintaxis.

Consideramos un último rubro de proyectos tecnológicos que permiten este tipo de realización audiovisual: aquel que se desenvuelve fuera de línea y que ha sido un punto de partida para la práctica performática audiovisual que también sirve como antecedente para el presente trabajo.

A continuación, un listado de proyectos que contempla con estas premisas y una descripción que las relaciona con esta investigación.

NIVEL BAJO Destaca por la optimización pero también por la complejidad en la escritura de código. Algunos proyectos que colindan con esta investigación son:

- **WASM** o WebAssembly es una alternativa a la compilación con lenguajes de programación dedicados al navegador. Por ejemplo, es posible realizar una aplicación en OpenFrameworks y compilarla para que pueda funcionar en el navegador.
- **Rust** es un lenguaje de programación que permite compilar a WebAssembly. La indagación realizada con el siguiente proyecto permitió tener en cuenta este lenguaje de programación en la medida que permite acceder a funciones de bajo nivel sin perder flexibilidad.
- **Ruffbox** es un proyecto escrito por Niklas Reppel en el lenguaje de programación Rust. Considero que este proyecto es de suma utilidad para la presente investigación ya que es un caso de realización de bajo nivel para el navegador con una sintaxis y lógica personalizadas.
- **Flocking¹**

EL NIVEL MEDIO supone herramientas que tienen funcionalidades determinadas y que no requieren que las instrucciones tengan que ser escritas con un nivel de abstracción inferior. Cabe destacar que una parte importante de la experiencia inicial con el trabajo con audio y video en la web descansa en este tipo de proyectos. Por lo general, tienen funciones y posibilidades que solucionan problemas comunes como cargar un objeto tridimensional previamente diseñado o reproducir un archivo de audio. Este nivel es abordado a profundidad en el siguiente capítulo, a continuación, una descripción breve y su posible relación con esta investigación y el resto de los antecedentes.

¹ <https://github.com/continuing-creativity/Flocking/>

- **Web Audio API** Es una especificación para realizar audio en el navegador. Hasta el momento, los módulos de código escritos dentro de esta investigación parte de este proyecto ya que permite enlazarlos con módulos de más alto nivel y también permite realizar modificaciones en estructuras de procesamiento de audio que son similares a los Generadores Unitarios.
- **Tone.js** fue una librería que este proyecto implementó para tener un mayor control en el procesamiento de audio. Si bien la librería es limitada, permite resolver problemas de audio con una curva de aprendizaje relativamente corta. Algunos problemas de secuenciación y de distribución de eventos en el tiempo se resuelven de manera eficiente. Por este motivo, las estructuras que permiten generar patrones son utilizadas como estructuras de control temporal en las piezas que se han mencionado anteriormente.
- **Three.js** es una plataforma para generar gráficos en 3d. Cuenta con algunos módulos para reproducir audio y posicionar la escucha en algún lugar de este espacio tridimensional. Esta aproximación fue la primera que el presente proyecto realizó, ya que la integración con la librería de imagen prácticamente estaba resuelta. Estas características son suficientes para reproducir pero son limitadas para procesar audio.
- **SuperCollider.js** y **SuperCollider.web** son casos que se integran con SuperCollider desde el navegador. Con este plataforma no es posible compilar y ejecutar un servidor de SuperCollider en el navegador o en algún servidor conectado a la web, sin embargo, sirve como punto de partida para hablar de los proyectos que compilan SC desde el navegador y que aprovechan las posibilidades de las aplicaciones web.
- maximilian

EL NIVEL ALTO implica que las funciones convencionales para el tratamiento de audio y video están resumidas en una sintaxis que permite llegar a resultados complejos y delimitados con pocas líneas de código. Por lo general, este nivel permite la interpretación al vuelo del código y posibilita el uso del código en presentaciones en vivo. Bajo ciertas circunstancias, podría comparar este nivel alto con el resultado del trabajo de un artesano. Considero que este nivel es el más interesante de analizar ya que permite observar las intenciones estéticas y políticas en decisiones sintácticas de alto nivel. Algunas de ellas pueden perseguir expresiones convencionales que van de la renderización de ruido sin pasar por la

construcción desde cero de una función matemática a la reproducción de patrones rítmicos con muestras de audio que coinciden con los estilos de la música latinoamericana para la pista de baile.

Algunos proyectos que cumplen con estas características son:

- **Hydra** es una inspiración directa para este proyecto. Cumple con la premisa del trabajo con código de manera dinámica, asumiendo los postulados de la programación al vuelo o live coding. También utiliza tecnologías ligeras que corren en el navegador, esto quiere decir que es posible utilizarlas en cualquier sistema operativo en distintos navegadores. Es posible utilizar Hydra como una biblioteca adicional de Javascript, lo cual permite la incorporación de texturas de Hydra en otras librerías como Three.js. Uno de los momentos de la realización de piezas para el navegador implementó Hydra como un generador de texturas.
- **Seis8s** es un proyecto que estuvo relacionado con Estuary y los talleres de la gira Bellacode. Este minilenguaje surgió en este contexto y actualmente tiene una interfaz autónoma que permite probar la interacción independientemente de un contenedor². El ordenamiento de las funciones que detonan los sonidos corresponde a un contexto cultural específico que da como resultado estilos latinos considerados como bailables(*Angel y Ogborn, 2022*). En este sentido, seis8s posibilita el desarrollo de un discurso musical para la música de baile con pocas líneas de código y con consecuencias que se perciben en situaciones de fiesta y celebración³.
- **LiveLab** es una herramienta que permite enviar varias señales de video y de sonido en una especie de videollamada compartida. El objetivo de este proyecto es dotar de soluciones para artistas que requieren tener un flujo compartido de eventos que pueden ser musicales pero también gestuales y performáticos. LiveLab cuenta con una interfaz que permite lidiar con varios problemas de transmisión de señales por medio de la web. la experiencia con esta herramienta está completamente relacionada con el contexto de la pandemia de COVID-19. La retransmisión de señales y la oferta de herramientas para la transmisión de audio y video diferentes a Zoom o Jitsi es una de las intenciones más contundentes de LiveLab.

² El sitio de seis8s se encuentra en:<https://seis8s.org/>

³ Uno de estos eventos es Algorave, que empata con la perspectiva del live coding.

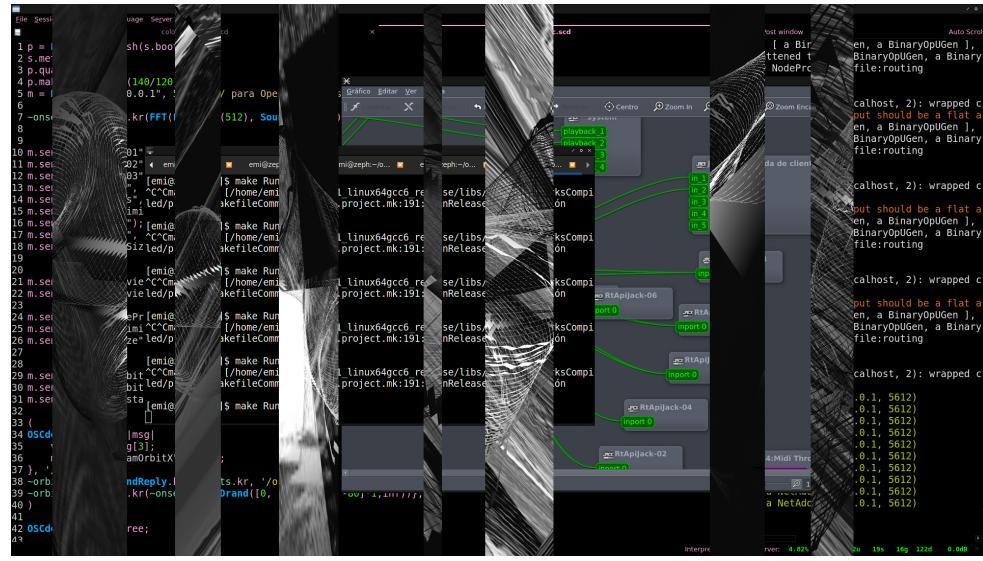


Figura 1: Captura de pantalla de una presentación realizada a distancia durante la pandemia de COVID-19. Audio compartido por medio de Sonobus e imagen intervenida y transmitida.

- Troop⁴
- tilt⁵
- timeNot⁶
- **Camposonico** es un caso que la persente investigación ha seguido de cerca, en la medida que es un desarrollo que conjunta programación al vuelo y despliegue audiovisual en la web. La peculiaridad de este proyecto es que es posible utilizar la API del sitio web Freesound Project para reproducir muestras a partir de una búsqueda con palabras clave.
- **INSTRUMENT** fue uno de los primeros proyectos que motivaron esta investigación. Fue parte del núcleo inicial de proyectos impulsados por PiranhaLab en el ciclo de talleres que se describe más adelante. Una de las características que destaco sobre este proyecto es la posiblidad de construir marcos para la codificación en vivo de audio sobre SuperCollider. De esta librería señalamos en particular la intención de desarrollar un planteamiento musical con pocas instrucciones.

⁴ <https://github.com/Qirky/Troop>

⁵ <https://github.com/munshkr/tilt>

⁶ <https://github.com/luisnavarrodelangel/seis8>

- El proyecto de Hegel
- **SonoTexto** es una de las escrituras que comparte inquietudes con esta investigación. Tiene por objetivo la resolución de problemas personalizados relacionados con la grabación de buffer de audio, con código escrito como un bloque modular de un entorno mucho más extenso(**Villaseñor, 2022**). Para el caso de SonoTexto, el marco contenedor es SuperCollider, esto quiere decir que es posible integrar de una manera mucho más orgánica la intención específica de la clase escrita por Hernani Villaseñor con SC y otros entornos para hacer programación al vuelo como **JitLib**.

LOS ECOSISTEMAS no necesariamente resuelven problemas específicos sino que sirven como contenedores de otros proyectos, lenguajes y sintaxis y que en todo caso permiten la renderización común entre interfaces de distinta índole. Una posible característica de la ejecución en web de estas plataformas es la colaboración espacialmente dislocada. Las escrituras que coinciden con esta descripción son:

- **Flok** puede entrar en la definición de ecosistemas, ya que cumple con la condición de albergar distintos lenguajes de programación al mismo tiempo que permite el uso de motores de audio y gráficos en el navegador y locales. Cuenta con un editor de texto que puede personalizarse para determinar cantidad de espacios de escritura y asociar cada uno de ellos con un marco específico.
- **Sema-engine**⁷

El caso de **Estuary** es bastante significativo para esta investigación, incluso, varios de los proyectos de alto nivel antes mencionados están alojados en este contenedor, se basan parcialmente en él para ejecutar instrucciones de audio en el navegador e incluso heredan la forma y la posibilidad del diseño de interfaces personalizadas para realizar estas instrucciones. Por este motivo, su función como antecedente es descrita de manera extensa a continuación.

La influencia de Estuary se remonta a la gira Bellacode del colectivo RGGTRN y al trabajo de investigación que inicio Luis Navarro en la Universidad McMaster. Además de conciertos y presentaciones, se realizaron talleres de retribución que utilizaron Estuary y CineVivo como herramientas principales. El objetivo de la actividad consistía en explicar la lógica de Estuary como un contenedor de

⁷ <https://github.com/frantico/sema-enginef>

interfaces de texto, la primera que es posible utilizar y que en muchos casos sugiere una familiaridad es una versión reducida y adaptada para la web de Tidal Cycles.

Posteriormente, se formaban grupos para que cada uno pudiera diseñar, bajo la lógica de reproducción de muestras, instrucciones que persiguieran distintos objetivos estéticos: funcionalidad al momento de invocar muestras de audio, desarrollo poético a partir de la metáfora e incluso, reapropiación de la jerga de canciones de reggaeton.(Angel et al., 2019)

Conclusiones generales sobre lo anteriormente expuesto.

2.3 OBRAS CERCANAS Y OBRAS LEJANAS

El siguiente apartado hace referencia a piezas específicas que he identificado como cercanas a mi práctica en lo que respecta al uso de tecnologías web, a la problematización de la escritura y de la palabra, a la emergencia del código escrito como parte de la pieza por sí misma y desde la relación entre imagen y sonido. Si bien ya está anunciado anteriormente en esta investigación, el texto se convierte en un aspecto tan central como la imagen y el sonido.

Notas de Ausencia de Marianne Teixido es un ensayo generativo en la web. Utiliza texto-dato que por medio de la computadora como agente resignificante, deconstruye estructuras discursivas para resemantizar la narrativa sobre las desapariciones de mujeres en México y América Latina.

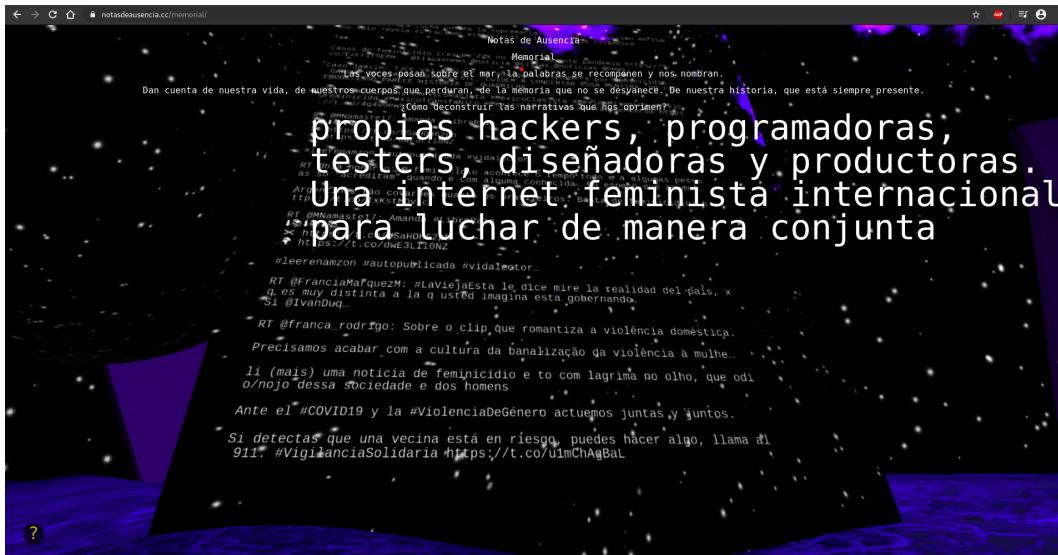


Figura 2: Captura de Notas de Ausencia de Marianne Teixido. Consultado el 12 de diciembre de 2022 en: <https://github.com/MarianneTeixido/notasdeausencia>

El tiempo y espacio virtual conforman una partitura para la memoria y la denuncia. La narrativa, semi-autónoma, argumenta a partir de textos tomados de tweets, poemarios, libros y artículos feministas que explican desde la teoría las desapariciones forzadas, el feminicidio y la violencia de género. Estos elementos están presentes como texto, imagen y sonido en un espacio tridimensional diseñado a manera de memorial.

La narrativa de la pieza está articulada mediante la intervención de dos bots que interactúan en Twitter. El primero realiza una búsqueda de tuits a partir de un filtro de palabras clave escritas como hashtags como: #MéxicoFeminicida, #MadresEnBúsqueda, #ViolenciadeGenero, #NiUnaMenos, entre otros.⁸

Posteriormente, recomparte el tuit que contiene alguno de los hashtags de la base de datos previamente delimitada. El segundo bot retoma los textos de los tuits seleccionados y los remixea para generar un segundo texto automático por medio de cadenas de Markov.

Esta pieza se realizó en el contexto de la exhibición en línea Creaciones con Algoritmos: Visualización y Sonificación de Datos del Centro de Cultura Digital en abril de 2020. Su salida oficial se realizó en video sin embargo, el planeamiento original contempló la creación de un espacio virtual en la web que pudiera visualizar en tiempo real la información viva proveniente de los tweets.

⁸ Al momento de escritura, el bot se encuentra activo en la cuenta: <https://twitter.com/notasausencia> (Consultado el 12 de diciembre de 2022)

Dos características definieron el diseño de la pieza: ubicación de la pieza en un espacio tridimensional y la interacción y visualización de la información proporcionada por los bots. Para la realización, el proyecto retomó módulos iniciales de Panorama y partió del uso de Three.js como un entorno de trabajo que pudiera conectar los dos momentos de la pieza antes descritos.

Uno de los referentes en términos de live coding con resultados visuales es la obra de Olivia Jack. La realización de éstas va de la mano con el software desarrollado por ella misma: Hydra. El proyecto está alojado en la web y la descripción se define como: “[U]na plataforma para live codear visuales, en la que cada ventana del browser puede ser usado como un nodo de un sintetizador de video, modular y distribuido”⁹

Hydra es un proyecto abierto y libre, lo cual permite que una comunidad de usuarios se involucren en el uso, mantenimiento y expansión del repositorio.

Algunos proyectos que utilizan directa o indirectamente hydra.

Destacamos del proyecto la sintaxis personalizada que permite controlar el resultado visual a partir de funciones y parámetros. Estas funciones pueden anidarse y concatenarse, lo cual da como resultado una serie de instrucciones aglomeradas y encadenadas. Este continuo coincide con la analogía del patcheo de los sintetizadores modulares de audio y video.

The stage is (a)Live¹⁰ - Johana Chicau y Renick Bell

2.4 PIRANHALAB

Otro antecedente de este proyecto es la práctica y reflexión planteada en colectivo por *PiranhaLab*¹¹.

El ciclo de talleres realizado en el Centro de Cultura Digital (CCD) en coparticipación con el Laboratorio de Tecnologías Libres¹² permitió plantear dos conclusiones que se heredan a *Tres Estudios Abiertos*: La difuminación de la distinción usuario/desarrollador como una motivación para la escritura de software y la procuración de diversidad en la escritura de software en América Latina.

⁹ Consultado el 12 de diciembre de 2022 en: hydra.ojack.xyz

¹⁰ <https://www.geometries.xyz/theStageIsAlive/>

¹¹ “PiranhaLab es un laboratorio interdisciplinario que trabaja en las tripas del software”. <https://piranhalab.github.io/> (Consultado el 12 de diciembre de 2022)

¹² Actualmente Laboratorio de Tecnologías Compartidas

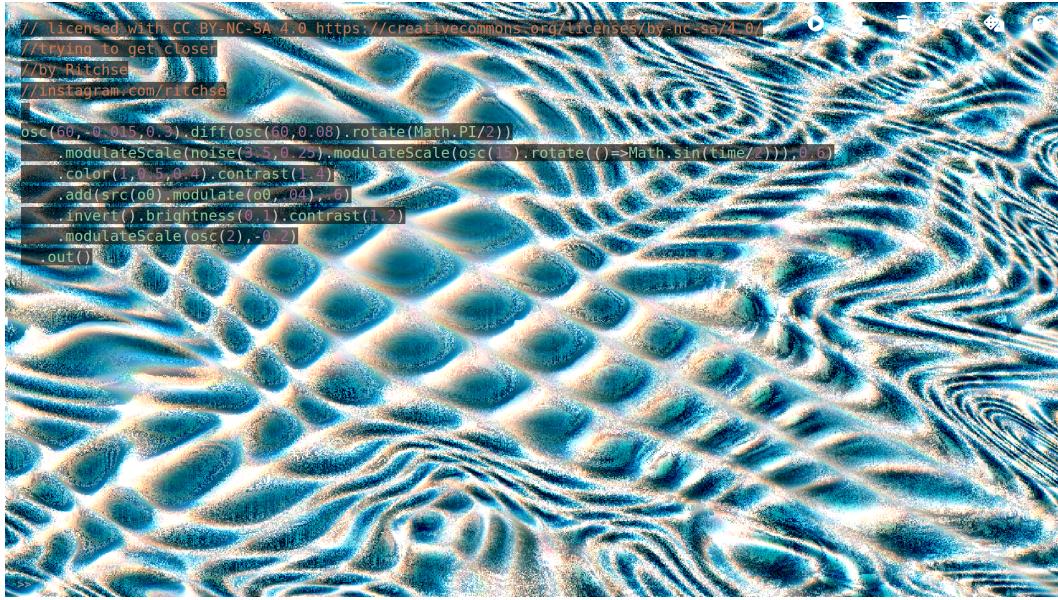


Figura 3: Hydra de Olivia Jack en la web. Código por Ritchse. Consultado en: https://hydra.ojack.xyz/?sketch_id=ritchse_3 el 12 de diciembre de 2022

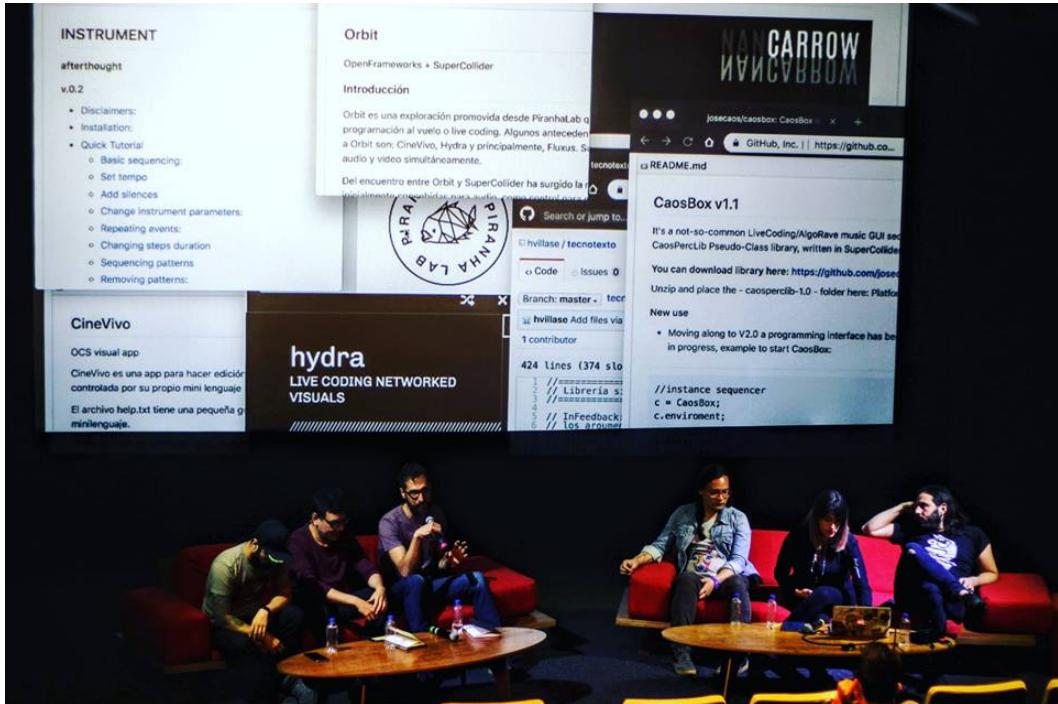


Figura 4: Conversatorio organizado por PiranhaLab y el Laboratorio de Tecnologías Compartidas en el CCD. 2019.

La escritura de espacios para el ciclo de conciertos EDGES 2020 realizado por el Taller de Imágenes en Movimiento del Centro Multimedia (CMM) permitió la exploración de entornos tridimensionales inmersivos en el navegador en el contexto del encierro causado por la pandemia de COVID-19. Técnica y conceptualmente la escritura de estos espacios digitales influye en el presente proyecto. El artículo *Panorama* (*PiranhaLab, 2020*) hace referencia de manera extensa al ecosistema de espacios y propuestas que también inciden en *Tres Estudios Abiertos*.

Espacios inmersivos que estuvieron activos de manera simultánea.

2.5 SEMINARIO PERMANENTE DE TECNOLOGÍA MUSICAL

El Seminario Permanente de Tecnología Musical (SPTM) es un espacio que se enmarca en el Posgrado en Música, específicamente en la línea de tecnología musical. Parte de las reflexiones de este proyecto de investigación se han abordado ahí y definitivamente se relacionan con investigaciones terminadas o en curso. La participación en este seminario permitió vislumbrar el mapa de investigaciones que se realizan en el Posgrado en Música de la UNAM y en específico, las posibles líneas de investigación que también forman parte del contexto en el que se inscribe esta investigación.

Una de las primeras actividades que se realizaron en el marco de este seminario fue el Encuentro Latinoamericano de Música y Tecnología *Transferencias Aurales* que tuvo lugar del 8 al 12 de noviembre de 2021.

El ciclo de conciertos EDGES 2020 abrió posibilidades de reflexión a partir de la escritura de programas. Estas reflexiones fueron puestas a discusión en un espacio que consolidó una serie de artículos que conforman el libro *Algoritmos Arruinados*.



Figura 5: Espacio diseñado para Distopía. ZGAMU (Música), LVSTVCVR(visuales e imágenes) y PiranhaLab (Diseño de espacio 3d)

3

MARCOS DE TRABAJO

Este apartado hace referencias a reflexiones que resultaron del trabajo con marcos en la realización de piezas audiovisuales para el navegador. Cuando hablo de marcos me refiero a lenguajes de programación de alto nivel como SuperCollider, OpenFrameworks, Three.js, Web Audio API entre otros. Si bien el caso de SuperCollider es particular, ya que es motor de audio, lenguaje de programación e intérprete al mismo tiempo, considero que marcos como los antes mencionados, guardan una relación abstracta con los motores que los posibilitan (WebGL, OpenGL e incluso el servidor de SuperCollider como módulo independiente). Esto quiere decir que hay funciones sintetizadas que resuelven problemas específicos que no requieren una reconstrucción de bajo nivel. Dos ejes establecen el flujo de este capítulo:

- Las intenciones expresivas de programas escritos con lenguajes de programación y la contraposición entre la programación eficiente y optimizada y otro tipo de búsquedas como la ofuscación o el error.
- El papel que juega la relación restricción y ofrecimiento como conceptos de la interacción humano-máquina (HCI), en la experiencia de los marcos de trabajo y la extensión a partir de funcionalidades básicas.

A continuación, relacionamos estos dos ejes con algunas descripciones técnicas de las piezas realizadas en conjunto con esta investigación, al mismo tiempo que las vinculamos con algunos de los marcos que se utilizaron en los distintos momentos de cada pieza.

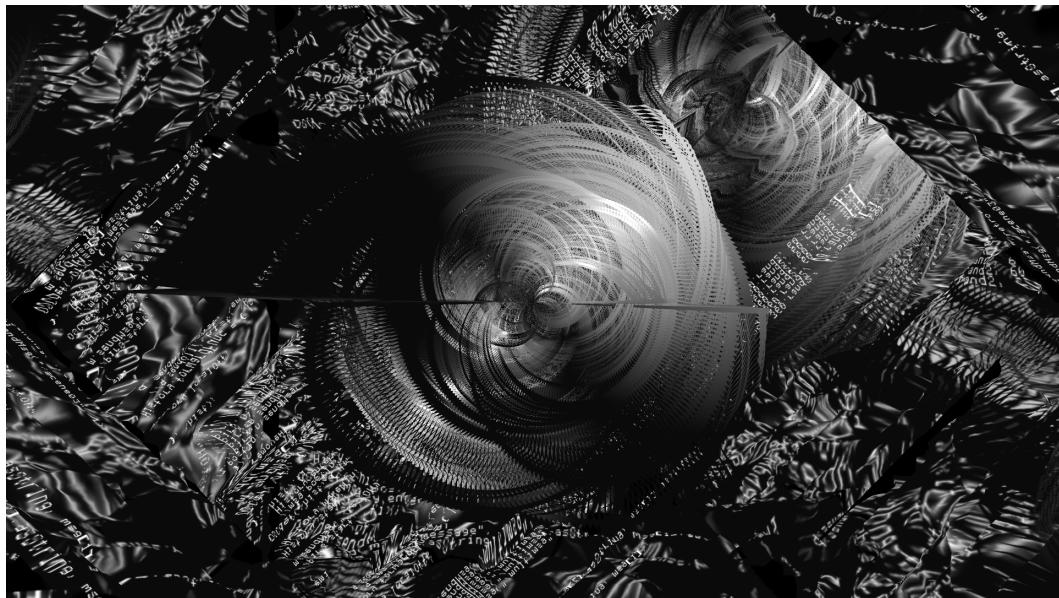


Figura 6: Captura de render realizado con OpenFrameworks

3.1 LA OFUSCACIÓN COMO MOTIVO

Anti es una pieza que surgió como respuesta a tecnologías vinculadas con redes sociales como Instagram o Facebook. En específico, atiende a las tecnologías que permiten el diseño de experiencias, capas y filtros en Realidad Aumentada. En un conjunto de herramientas como la que ofrece Meta Spark (antes llamado Spark AR), la personalización de las capas que se superponen a una imagen fija o en movimiento está determinada por una interfaz que acota las posibilidades de estas tecnologías al mismo tiempo que facilita su uso.



Figura 7: Portada del primer momento de Anti. La versión actual se puede consultar en: <https://anti.ocelotl.cc>.

Para la realización de Anti, tomé una decisión inicial que contemplaba tres caminos: el uso de Meta Spark, [OpenFrameworks](#) o algún tipo de instancia para la web basada en P5.js o [Three.js](#).

Como parte del conjunto de restricciones que formaron parte de la intención inicial de este proyecto, Meta Spark quedó desacartada por ser una herramienta privativa. El uso de herramientas de software libre o código abierto puede ser muy ambiguo si consideramos que las licencias que sustentan estos estatutos tienen variaciones que por un lado pueden reforzar visiones optimistas pero poco efectivas en el campo profesional y que por otro, se adscriben a modelos que apoyan a proyectos o instancias con poca claridad en el manejo de datos como Facebook. Sin embargo, como parte de las decisiones subjetivas de este proyecto,

considero que hay una frontera fundamental que se relaciona con una intención política pero también con una implicación de accesibilidad: experiencias que solamente pueden ser accedidas por medio de una red social, plataformas de creación de capas tipo filtro que solamente funcionan en un sistema operativo e incluso, la restricción que implica el uso de tarjetas gráficas dedicadas o cualquier tipo de escalada en los recursos físicos de la computadora o dispositivo desde el cual se accesa.

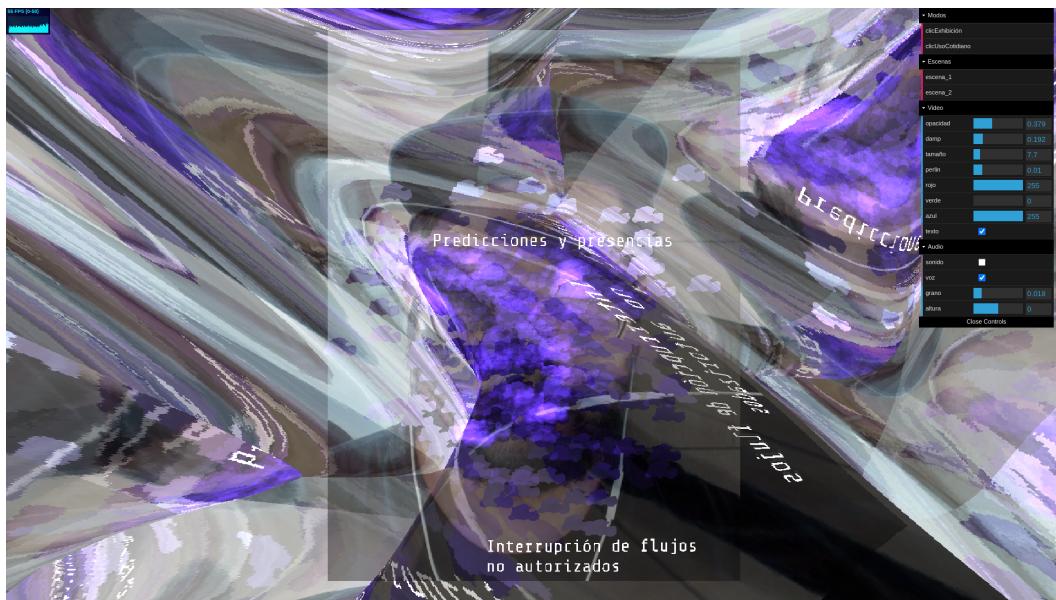


Figura 8: Anti e interfaz gráfica. <https://anti.ocelotl.cc>.

La premisa principal de Anti fue la exploración en términos visuales y sonoros del concepto ofuscación audiovisual de voz y rostro. El objetivo de la premisa tecnológica fue intervenir de alguna manera la cara que es capturada por una cámara y el audio que es registrado por un micrófono. Del lado del audio, no se utilizó algún tipo de análisis para transformar la voz, bastó con un filtro de granulación que permitiera modificar la voz sin que esta pudiera ser reconocible pero que pudiera seguir transmitiendo palabras entendibles por un escucha externo. Del lado de la imagen implementé un modelo de TensorFlow previamente entrenado de reconocimiento de puntos clave del rostro¹. Hay tres ventajas de este modelo sobre de otros: la información no pasa por servidor externo, el modelo no busca detectar y diferenciar rostros y el nivel de precisión del modelo es bastante alto(Kartynnik *et al.*, 2019).

¹ El modelo específico que utilicé fue face-landmarks-detection. La colección de modelos se encuentra en: <https://github.com/tensorflow/tfjs-models>. Consultado el 12 de diciembre de 2022

Selecciona una foto para tu perfil

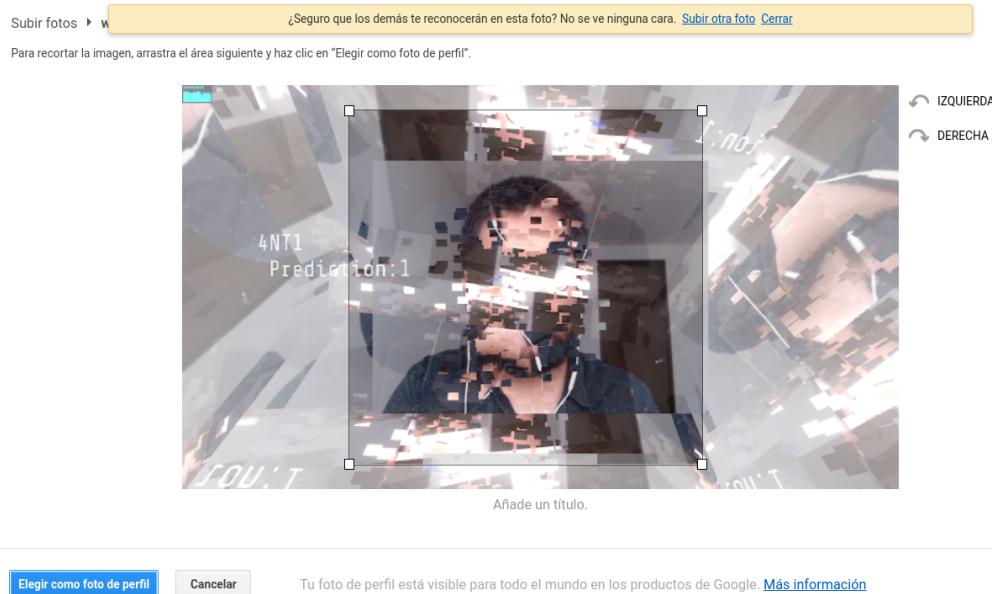


Figura 9: Captura de la primera imagen de Anti subida a Google como foto de perfil en la que no se reconoce el rostro.

Como el modelo se ejecuta en Javascript, fue posible incorporarlo de una manera nativa a las librerías de renderizado tridimensional y de audio. Eso generó un sistema relativamente eficiente e integrado que si bien no explora la ofuscación en términos del código mismo, si busca tener una intención en términos de la configuración de una herramienta tecnológica con un objetivo cotidiano como es ofuscar la información que se puede extraer del rostro y de la voz, sobre términos de la identificación, seguimiento y vigilancia del individuo. La ofuscación como motivo desdibuja la identidad para conservarla en un anonimato que puede ser compartido y que puede ser el principio de otras formas de pensar la relación el humano como agente y el código como una tecnología imputada de un sentido que no solamente persigue la eficiencia.

Precisando en la discusión, la ofuscación puede definirse como el acto deliberado de encubrir el significado de una comunicación. Para el caso de la programación y apuntando ideas hacia los estudios del software, la presente investigación toma la noción de ofuscación de un conjunto de posibilidades para la escritura de software que coinciden, dialogan o se enfrentan a que podríamos definir como la

convención de la *estética del código* y aquellos programas que exploran “otros principios estéticos” además de los convencionales.

Edsger Dijkstra conoincide con la delimitación convencional de esta forma de escribir programas:

“[...] el programador no difiere de algún otro artesano: a menos de que ame sus herramientas, es altamente improbable que pueda crear algo de calidad superior. Al mismo tiempo estas consideraciones nos hablan de las más grandes virtudes que un programa puede mostrar: Elegancia y Belleza”(Dijkstra, nd, p. 10)

Como respuesta a la posición de Dijkstra y en un ámbito de programación que se aproxima lúdicamente a la escritura de programas, la ofuscación:

“arroja luz a la naturaleza del código fuente, que es leído por un humano e interpretado por una máquina, y puede recordar a los críticos la búsqueda por diferentes dimensiones de sentido y múltiples codeos en todo tipo de programas”(Montfort, 2008, p. 198)

El código su lectura, así como las funciones de los programas que ejecuta, son subjetivas y están determinadas por un sentido imputado que socialmente se acuerda de manera tácita y que puede ser visibilizado para interpelarlo en un sentido crítico, lúdico e incluso satírico.² En este punto encontramos conexiones con las posibilidades de la programación en una dimensión artística:

“[La práctica de la programación ofuscada] sugiere que el codeo puede resistir a la claridad y a la elegancia para pugnar en su lugar por la complejidad, puede hacer familiar lo desconocido y puede luchar con el lenguaje en el que está escrito, justo como lo hace la literatura contemporánea.”(Montfort, 2008, p. 198)

La ofuscación puede definirse como el acto deliberado de encubrir el significado de una comunicación. Para el caso de la programación y apuntando ideas hacia los estudios del software, la presente investigación toma la noción de ofuscación de un conjunto de posibilidades para la escritura de software que coinciden, dialogan o se enfrentan a que podríamos definir como la convención de la *estética del código* (Dijkstra, nd) y aquellos programas que exploran “otros principios estéticos” (Montfort, 2008) además de los convencionales.

¿Puede el código fuente “luchar” en contra del marco de uso para el que fue escrito?

² Tal es el caso de Windows 93 del artista Jankenpopp. <https://www.windows93.net/>

3.2 TRES ESTUDIOS

La práctica artística en contextos audiovisuales y performáticos que realicé hasta 2020 estuvo asociada al uso de SuperCollider para el audio y OpenFrameworks para el video. Las primeras aproximaciones a esta relación tecnológica se debió al trabajo realizado con Celeste Betancur, Jessica Rodríguez, Iracema de Andrade y Alejandro Brianza. Altamisa y Leviatán fueron dos piezas que implementaron esta relación y con apoyo de Celeste Betancur, fue posible realizar un sistema que pudiera detonar videos de acuerdo a decisiones que fueron controladas por medio del protocolo OSC. Esta relación fue ampliamente explorada y en particular, las relaciones y la composición centrada ya no únicamente en audio o video sino en flujos compartidos de información. Este eje continúa presente en la actualidad. El tránsito de la relación SC y OF hacia Javascript fue motivado por las exigencias tecnológicas que implicó la transmisión de audio e imagen para espacios virtuales. El dominio tecnológico de una nueva herramienta puede suponer un reto y una curva de aprendizaje que requiere dedicación de tiempo. Consideramos que estas curvas pueden reducirse si en el proceso de escritura de código hay información sobre los antecedentes históricos y las ideas que conducen el funcionamiento fundamental de los marcos de trabajo. En este sentido, podemos realizar una observación comparativa entre la plataforma anteriormente utilizada (OF) y Three.js.

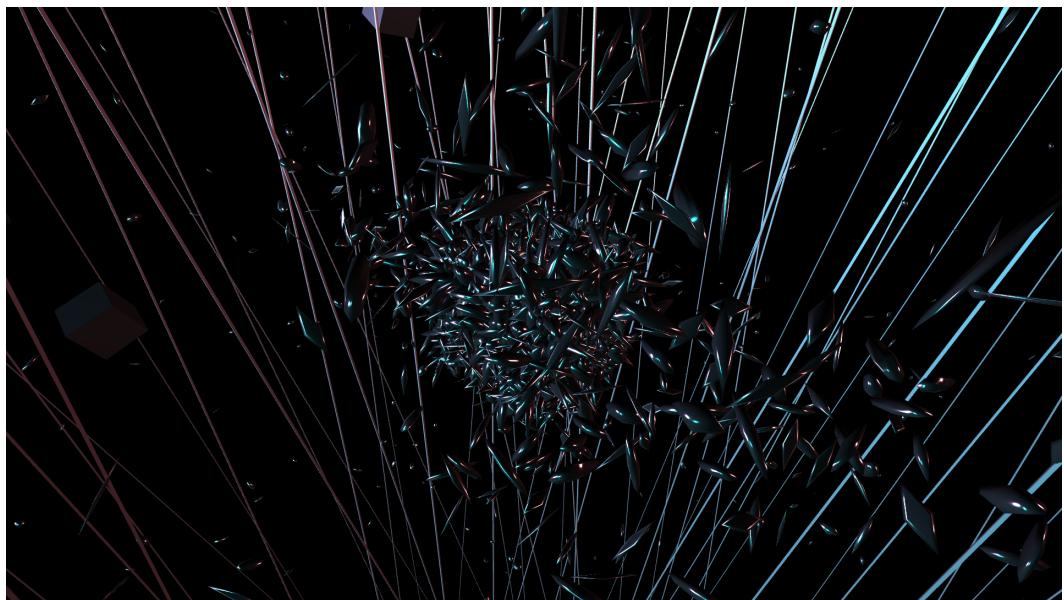


Figura 10: Captura de la primera versión de THREE.studies

OpenFrameworks tiene una forma de concebir el espacio tridimensional, los objetos y las texturas de manera muy similar a Processing en tanto que son objetos que pueden ser invocados y se envuelven entre sí. Para el caso de Three.js, la aproximación se parece mucho más a la forma con la que se conciben los espacios y objetos tridimensionales en programas de diseño y modelado en tres dimensiones. En este sentido es posible trabajar de manera independiente con geometrías, materiales y mallas.

Comparativa entre OpenFrameworks y Three.js. Estándares del trabajo con imagen que se heredan a través de Processing y que vienen de ideas más antiguas. El verdadero reto está en conciliar los tiempos del audio y el video, siendo que cada uno corre en distintas tasas.

3.3 LA INVESTIGACIÓN COMPILADA

Como se mencionó en la introducción de esta investigación, este texto busca tener dos tipos de salidas: la primera es el texto fijado, impreso o en PDF y la segunda, una serie de recursos que giran en torno al texto y que pueden ser compiladas para el navegador web.

4 | EXPRESIVIDAD

- La contradicción que existe entre un tipo de programas que deben ser detenidos y recompilados para cambiar y la programación que puede cambiar dinámicamente y que puede ser transformada al vuelo.
- El papel que tiene la gestualidad y la interpretación en la dimensión digital de piezas audiovisuales que pueden ser fijas pero también que pueden ser intervenidas en el momento.

4.1 GESTO, EXPRESIVIDAD Y PERFORMANCE EN VIVO

Encuentro múltiples formas de abordar el tema. El live coding puede describirse en términos de una comunidad que realiza una práctica y que se enuncia como tal, como un sub-campo creativo que comparte elementos con otras expresiones como la música algorítmica y el arte generativo. Finalmente, el live coding podría acotarse a motivos de investigación académica e independiente y a la realización tecnológica de interfaces de texto que corren sobre motores de audio y video.

De estas múltiples formas de abordar el fenómeno y de acuerdo a los fines de esta investigación, nos preguntamos si el eje que las articula son discusiones sobre el lenguaje y la escritura.

Antecedentes del live coding: la escena mod.

En el manifiesto del live coding se expresan pensamientos que hasta el momento, son vigentes.

Dentro de los antecedentes está la experiencia performática de escribir código al vuelo con fines creativos, audiovisuales y experimentales.

Las primeras expresiones reflexivas y prácticas del live coding no solamente establecen puntos de partida performáticas, sino que al mismo tiempo aportan elementos para el diseño y análisis de sistemas basados en interfaces de código:

“Nuestro argumento nos lleva a través de capas de representación, comenzando con símbolos, luego palabras, lenguaje y notación, para considerar el papel que estas representaciones pueden jugar en la creatividad humana” (McLean, 2011, p. 3)

La noción de capas de representación permite analizar la complejidad del proceso artístico con código en términos de la relación entre humanos y los aspectos cognitivos, sociales y políticos que les atraviesan, y los no-humanos en lo que respecta a las posibilidades de conocimiento instanciado presentes en ellos. Live coding desde cero y la improvisación.

La práctica de la programación al vuelo, delimitada técnica, escénica, musical y visualmente se origina en Inglaterra.

Como lo describen Villaseñor-Ramírez y Paz (2020) para los casos de Barcelona y Ciudad de México.¹

Exploración visual para la integración con el sonido.

El campo del live coding tuvo un giro importante con la llegada de hydra de Olivia Jack. La rápida expansión de esta plataforma y la sintaxis que recuerda la conexión de flujos de energía similar a los sintetizadores analógicos favorecieron la producción de piezas e interpretaciones enfocadas en la imagen y en algunos otros casos a la par del audio. La importancia de esta plataforma ha delimitado una estética basada en la transformación de píxeles, de formas y de transformaciones basadas en funciones matemáticas.

En palabras de Olivia Jack, los procesos que realiza la computadora hay un eje de profundidad, en términos del despliegue gráfico físico de la computadora de espacios bi o tridimensionales, el resultado es bidimensional.

La lógica de Hydra es modular, esto quiere decir que puede incorporarse como un componente adicional a proyectos que no necesariamente se centren en la producción visual con esta plataforma. Esta modularidad juega a favor de lenguajes de programación como Javascript.

4.2 ESTÁTICA/DINÁMICA

Three.studies y el performance que fue de espacios digitales a la sala de concierto.

¹ Un ejemplo reciente se encuentra en: <https://youtu.be/n5kwi4eRAE4>

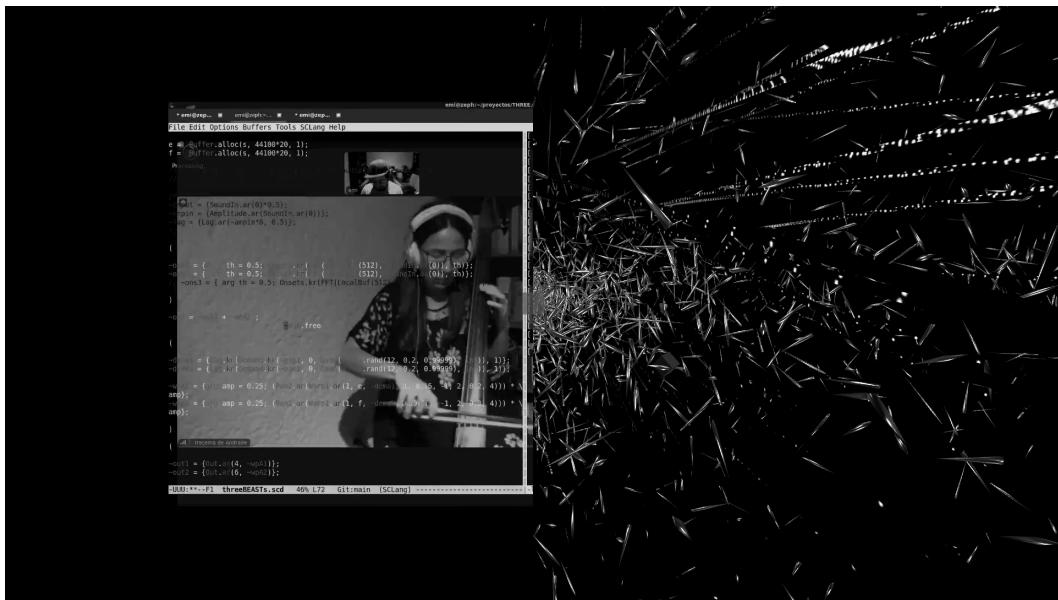


Figura 11: Captura de pantalla de la versión de THREE.studies que incluyó el video con el código y la grabación de audio y video.

Piezas fijas y la composición conducida por (bases) de datos

Sonificación-Visualización La tradición electroacústica-acusmática

Live coding y captura gestual El problema de la expresividad con algo que aparentemente no es gestual

4.3 ¿ES POSIBLE ENCONTRAR UNA SOLUCIÓN?

Esas estructuras pueden ser fijas o dinámicas y dependen de varios niveles de profundidad que las significan y las ejecutan En la práctica, cualquier entorno está en esta mediación. Trabajo muerto acumulado anteriormente. Reflexionar sobre esto

5 | EXTENSIÓN Y ESCRITURA

En este capítulo hablaremos de dos aspectos que detonan reflexión referente a la escritura de módulos de software personalizados.

- La escritura con lenguajes de programación como una posibilidad para la reconfiguración de las distinciones entre usuarios y desarrolladores.
- El papel de la diversidad tecnológica en la extensión de los marcos de trabajo y la resistencia a la escalada de recursos computacionales

En este sentido, destaco el papel que tiene la extensión de las funcionalidades de marcos de trabajo y lenguajes de programación. En particular, es de interés de este trabajo destacar la ampliación de estas posibilidades de cara a las consecuencias poéticas de piezas construidas con lenguajes de programación. Estas librerías adicionales también pueden perseguir, de manera paralela, objetivos funcionales que pueden medirse de alguna manera.

5.1 MODULOS PERSONALIZADOS

Escritura de bibliotecas para alcanzar una expresividad intermedia. ¿Qué implica escribir módulos personalizados? Primero que no existan antecedentes y que la idea en lo general pueda al menos realizarse técnicamente.

5.2 RECONFIGURACIÓN DE DISTINCIONES

Acción: reconfigurar la distinción usuario/desarrollador-escritor PLab como casos. Espacio de encuentro y tecnodiversidad Declaración/acción

5.3 ACCIÓN E IDEA

El código como interpretación y como partitura Valoraciones del código: elegancia y ofuscación Encuentros y brechas entre escritura de código, de productos académicos y de algo más abierto.

6

INTEGRACIÓN

En este apartado estaremos abordando la discusión en dos ejes principales:

- Reflexiones en torno al software que es arte y al software que produce arte y cómo esto tiene consecuencias en el proceso artístico y de investigación que se ha realizado hasta el momento.
- La posible resolución de la brecha que existe entre la escritura que genera programas de computadora, investigación y acción creativa.

En este sentido, este apartado aborda la relación entre la escritura de/con software y la acción que se activa cuando se ejecuta esa escritura.

6.1 LA ESCRITURA COMO RODEO

Hasta el momento hemos abordado tangencialmente la temática de la escritura. Este apartado describe de manera recursiva el proceso por el cuál fue escrita la versión expandida de esta investigación. Si bien hay una versión en formato PDF/impresión, también hay una versión que intercala de una manera integrada los elementos que hemos mencionado hasta el momento: audio, imagen y texto. Para realizar esta labor, retomaremos la noción de síntesis granular para desplazar el ordenamiento de estos elementos a una entidad históricamente referida en la escritura de software para hacer audio y tecnológicamente integradora para esta investigación: el generador unitario. En este sentido no persigo el diseño de un documento con el mismo texto de la investigación embebido y acompañado con audio o imágenes de fondo; busco explorar la posibilidad de escribir módulos para el reordenamiento del material en el espacio y tiempo integrados desde el elemento tecnológico más pequeño que las puede producir. La suma de estos elementos puede generar una densidad que transmita en la dimensión de lo sensible, las reflexiones vertidas en esta investigación.

Recursividad Documentos que se ejecutan

6.2 SEGUNDA SECCIÓN

La realización de este proyecto de investigación y la integración de los elementos que lo componen puede complementarse con una navegación estilo videojuego. La noción de cámara y algunos elementos gestuales que la componen como posición y rotación permite una navegación que toma en cuenta las decisiones espaciales del usuario. Aunado a esto, es posible extender las posibilidades del espacio tridimensional con distintos tipos de controles que de acuerdo a la necesidad, centran su comportamiento dependiendo del objetivo: vuelo, orbitación o exploración como videojuegos en primera persona¹. Juegos que busca la mejor panorámica, por ejemplo pokémon y fotografías.

6.3 TERCERA SECCIÓN

¹ En este caso refiero a la convención de los videojuegos de disparos en primera persona (FPS). Para la enunciación de esta perspectiva he decidido prescindir de la palabra disparos por su carácter bélico. La exploración bajo esta perspectiva no necesariamente se adscribe a la búsqueda o eliminación de algún objetivo.

7 | CONCLUSIONES

BIBLIOGRAFÍA

- Angel, L. N. D. y Ogborn, D. (2022). Seis8s: a web-based localized computer-music language. En *NIME 2022*.
<https://nime.pubpub.org/pub/wolnquiz>.
- Angel, L. N. D., Teixido, M., Ocelotl, E., y Ogborn, D. (2019). Bellacode: localized textual interfaces for live coding music. En *Proceedings of the Fourth International Conference on Live Coding*, p. 353, Madrid, Spain. Medialab Prado / Madrid Destino.
- Augoyard, J.-F. y Torgue, H., editores (2006). *Sonic experience. A guide to everyday sounds*. McHill Queen's University Press, Canada.
- Cox, G. (2010). *Antithesis: The Dialectics of Software Art*. Digital Aesthetics Research Center, Denmark.
- Di Prospero, C. (2015). Live coding. arte computacional en proceso. *Revista Contenido. Cultura y Ciencias Sociales*, 1(5):44–63.
- Dijkstra, E. W. (n.d.). Some meditations on advanced programming. Circulated privately.
- García, J. D. (2021). Tecnología libre para una música libre ¿es posible construir un sistema de producción alternativo? *Revista F-ILIA*, II(4):57–88.
- Kartynnik, Y., Ablavatski, A., Grishchenko, I., y Grundmann, M. (2019). Real-time facial surface geometry from monocular video on mobile gpus.
- Knuth, D. E. (1991). *The TexBook*. Addison –Wesley Publishing Company.
- Manovich, L. (2001). *The Language of New Media*. Leonardo (Series) (Cambridge, Mass.). MIT Press.
- McLean, A. (2011). *Artist-Programmers and Programming Languages for the Arts*. Tesis doctoral, Department of Computing, Goldsmiths, University of London.
- Montfort, N. (2008). Obfuscated code. En Fuller, M., editor, *Software Studies: A Lexicon*. The MIT Press.

- Ocelotl, E. (2015). Objeto, paisaje y efecto. Aportes para la investigación social en música. Tesis de licenciatura en sociología, Facultad de Ciencias Políticas y Sociales. Universidad Nacional Autónoma de México.
- Ocelotl, E. (2019). Cuidado con la brecha autorreferencial. Aportes para la producción-investigación en música de sistemas interactivos. Tesis de maestría en música (tecnología musical), Posgrado en Música. Universidad Nacional Autónoma de México.
- Pantieri, L. (2008). L arte di scrivere con latex. un introduzione a latex.
- PiranhaLab (2020). Panorama. <https://github.com/piranhalab/panorama>.
- Platohedro, Correa, A., Alvarez, L., Fleischmann, L., Rodríguez, Y., Rueda, D., Jaramillo, J. A., Correa, C., y y O. Narváez (2019). *Platohedro. Multiversos*. Cráneo Invertido, Medellín, Colombia.
- Roads, C. (2001). *Microsound / Curtis Roads*. MIT Press, Cambridge, Mass.
- Schaeffer, P. (2003). *Tratado de los objetos musicales*. Alianza Música, México D.F.
- Schafer, M. R. (1994). *The Soundscape. Our Sonic Environment and the Turning of the World*. Destiny Books, Vermont.
- Shanken, E. A. (2010). Historicizing art and technology: Forging a method and firing a canon. En Grau, O., editor, *Media Art Histories*, pp. 43–70. The MIT Press.
- Soon, W. y Cox, G. (2020). *Aesthetic Programming: A Handbook of Software Studies*. Open Humanities Press.
- Villaseñor, H. (2022). *Sonotexto : relación entre práctica artística y desarrollo tecnológico en el live coding*. Tesis doctoral, Posgrado en Música (Tecnología Musical). Universidad Nacional Autónoma de México.
- 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.

GLOSARIO

AudioWorklets La interfaz AudioWorklet de la Web Audio API es usada para cumplir scripts personalizados de procesamiento de audio que se ejecutan en un hilo independiente para proporcionar procesamiento de audio de baja latencia. Consultado en: <https://developer.mozilla.org/en-US/docs/Web/API/AudioWorklet> el 12 de diciembre de 2022. ⁹

C++ es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido. Consultado en: <https://es.wikipedia.org/wiki/C%2B%2B> el 12 de diciembre de 2022. ⁴

Campsonico Camposónico (radio algorítmica) es una aplicación para escuchar e intervenir paisajes sonoros y música experimental, haciendo posible la interacción con el sonido que va de una autorreproducción simple a la experimentación creativa a través del uso de código. Consultado en: <https://github.com/diegovdc/campsonico> el 12 de diciembre de 2022. ¹⁸

Estuary Estuary es una plataforma para la colaboración y el aprendizaje a través de la codificación en vivo. Estuary permite experimentar con sonido, música e imágenes en un navegador web. Así mismo, Estuary reúne una colección curada de lenguajes de codificación en vivo en un solo entorno, sin el requisito de instalar software (que no sea un navegador web) y con soporte para agrupaciones artísticas en red (ya sea con gente participando en la misma sala o distribuida por todo el mundo). Consultado en <https://estuary.mcmaster.ca/> el 12 de diciembre de 2022. ¹⁹

Flok Editor P2P colaborativo basado en web para live codear música y gráficos. Los plugins REPL permiten ejecutar TidalCycles, SuperCollider, FoxDot y Mercury. Los plugins para web están orientados a lenguajes embebidos en el editor como Hydra y P5.js. Consultado en: <https://github.com/munshkr/flok> el 12 de diciembre de 2022. ¹⁸

Git es un sistema distribuido, libre y de código abierto de control de versiones diseñado para manejar todo, desde proyectos muy grandes o pequeños con velocidad y eficiencia. [11](#)

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Consultado en:
<https://developer.mozilla.org/es/docs/Web/HTML> el 12 de diciembre de 2022. [11](#)

Hydra Hydra es un sintetizador de video y entorno de codificación en vivo (livecoding) que funciona sobre el navegador. Es de software libre y está pensado tanto para principiantes como expertos. Consultado en:
<https://hydra.ojack.xyz/> el 12 de diciembre de 2022. [17](#)

Interacción Persona-Computadora La interacción persona-computadora o persona-ordenador (IPO) es la disciplina dedicada a diseñar, evaluar e implementar sistemas informáticos interactivos para el uso humano, y a estudiar los fenómenos relacionados más significativos.. [8](#)

Javascript es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase.. [4](#)

JitLib JitLib consiste en una serie de espacios reservados (proxies del lado del servidor y del lado del cliente) y esquemas de acceso. Estos dos aspectos de espacio corresponden a inclusión y referencia, dependiendo de su contexto - aquí, los espacios reservados son como roles que tienen cierto comportamiento y que pueden ser llenados por ciertos objetos.
Consultado en: <https://doc.sccode.org/Overviews/JITLib.html> el 12 de diciembre de 2022.. [18](#)

Lenguajes de programación de alto nivel Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad con que los ejecutan las máquinas. [26](#)

Lisp es una familia de lenguajes de programación de computadora de tipo multiparadigma con larga historia y una inconfundible y útil sintaxis homoicónica basada en la notación polaca. Consultado en:
<https://es.wikipedia.org/wiki/Lisp> el 12 de diciembre de 2022. [4](#)

LiveLab es una nueva herramienta que empodera a artistas y a presentadores de arte para encontrarse, crear, colaborar, ensayar y por último, producir

performances en múltiples locaciones desde virtualmente cualquier parte del mundo. Este software innovador de colaboración por medio de video expanda el campo actual de ofertas al permitir a los usuarios personalizar los medios en caminos que mejor acomodan a sus necesidades.

Consultado en: <https://github.com/ojack/LiveLab> el 12 de diciembre de 2022. ¹⁷

Navegador Web Un navegador web (en español, web browser) es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.. ⁴

openFrameworks es una conjunto de herramientas de código abierto escritas en C++ diseñadas para asistir el proceso creativo, proveyendo un marco de trabajo simple e intuitivo par la experimentación. Consultado en: <https://openframeworks.cc/about/> el 12 de diciembre de 2022. ²⁸

OSC OpenSoundControl (OSC) es una especificación de transporte de datos (una codificación) para la comunicación en tiempo real a través de mensajes entre aplicaciones y hardware. Consultado en: <https://ccrma.stanford.edu/groups/osc/index.html> el 12 de diciembre de 2022. ⁶

Renderización Anglicismo de representación gráfica. Procedimiento para generar una imagen bidimensional o tridimensional por medio de la computadora. ⁵

Ruffbox es un sintetizador/sampleador sencillo con un sequenciador de pasos basado en texto que corre en la web. Consultado en: <https://github.com/the-drunk-coder/ruffbox> el 12 de diciembre de 2022. ¹⁵

Rust Rust es un lenguaje de programación compilado, de propósito general y multiparadigma. Es un lenguaje de programación multiparadigmático que soporta programación funcional pura, por procedimientos, imperativa y orientada a objetos.. ¹⁵

Seis8s es un lenguaje de programación que permite la interacción en tiempo real con audio digital y conocimiento musical localizado, particularmente de músicas de Latinoamérica. Consultado en: <https://github.com/luisnavarrodelangel/seis8s> el 12 de diciembre de 2022. ¹⁷

Software Libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el “software libre” es una cuestión de libertad, no de precio. Para entender el concepto, piense en “libre” como en “libre expresión”, no como en “barra libre.”. ⁶

SonoTexto es una clase para grabar y reproducir buffers de audio. Esta clase se usa para improvisar con músicos que tocan instrumentos acústicos, así, es posible guardar pequeños buffers de los instrumentos al momento de la improvisación. Consultado en: <https://github.com/hvillase/sonotexto> el 12 de diciembre de 2022.. ¹⁸

SuperCollider.js SuperCollider.js es una librería-cliente con todas las funciones y baterías incluídas para el servidor de síntesis de audio SuperCollider y el intérprete del lenguaje SuperCollider. Consultado en: <https://crucialfelix.github.io/supercolliderjs> el 12 de diciembre de 2022. ¹⁶

SuperCollider.web Aplicación web basada en Node.js para crear archivos de audio para SuperCollider, con integración para SoundCloud. Consultado en: <https://github.com/khilnani/supercollider.web> el 12 de diciembre de 2022. ¹⁶

Three.js El objetivo de este proyecto consiste en crear una librería 3D fácil de usar, ligera, multi-navegador, para múltiples propósitos. Consultado en: <https://github.com/mrdoob/three.js> el 12 de diciembre de 2022. ^{9, 16, 28}

Tidal Cycles (o ‘Tidal’) es un software libre/abierto escrito en Haskell. Tidal usa SuperCollider, otro software de código abierto para síntesis y I/O. Consultado en: <https://tidalcycles.org/> el 12 de diciembre de 2022. ⁴

Tone.js Tone.js es un marco de audio en la web para crear música interactiva en el navegador. La arquitectura de Tone.js busca la familiaridad tanto para músicos como para programadores de audio para crear aplicaciones de audio basadas en la web. Consultado en: <https://tonejs.github.io/> el 12 de diciembre de 2022.. ¹⁶

WASM WebAssembly (abreviado Wasm) está diseñado como un objetivo de compilación portátil para lenguajes de programación, lo que permite la implementación en la web para aplicaciones de cliente y servidor. Consultado en: <https://webassembly.org/> el 12 de diciembre de 2022.. ¹⁵

Web Audio API provee un sistema poderoso y versatil 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. Consultado en: https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API el 12 de diciembre de 2022. 4, 9, 16