

Generación y Control de Contenido Audiovisual Utilizando TouchDesigner, Ableton Live y los Movimientos del Cuerpo a Través del Perception Neuron

Trabajo de grado

por

Matías Macías Gómez y Simón Pineda Agudelo



**GENERACIÓN Y CONTROL DE CONTENIDO AUDIOVISUAL UTILIZANDO
TOUCHDESIGNER, ABLETON LIVE Y LOS MOVIMIENTOS DEL CUERPO A
TRAVÉS DEL PERCEPTION NEURON**

MATÍAS MACÍAS GÓMEZ, SIMÓN PINEDA AGUDELO

UNIVERSIDAD PONTIFICIA BOLIVARIANA

INGENIERÍA

TIC

INGENIERÍA EN DISEÑO DE ENTRETENIMIENTO DIGITAL

MEDELLÍN

2020

**GENERACIÓN Y CONTROL DE CONTENIDO AUDIOVISUAL UTILIZANDO
TOUCHDESIGNER, ABLETON LIVE Y LOS MOVIMIENTOS DEL CUERPO A
TRAVÉS DEL PERCEPTION NEURON**

MATÍAS MACÍAS GÓMEZ, SIMÓN PINEDA AGUDELO

**Trabajo de grado para optar al título de
Ingeniero en Diseño de Entretenimiento Digital**

ASESOR

JUAN FERNANDO FRANCO HIGUITA

Magister en Ingeniería

UNIVERSIDAD PONTIFICIA BOLIVARIANA

INGENIERÍA

TIC

INGENIERÍA EN DISEÑO DE ENTRETENIMIENTO DIGITAL

MEDELLÍN

2020

FECHA: 5 de mayo de 2020

MATÍAS MACÍAS GÓMEZ Y SIMÓN PINEDA AGUDELO

“Declaramos que este trabajo de grado no ha sido presentado con anterioridad para optar a un título, ya sea en igual forma o con variaciones, en ésta o en cualquiera otra universidad”. Art. 92, párrafo, Régimen Estudiantil de Formación Avanzada.

FIRMA

Matías Macías Gómez

A handwritten signature in black ink, appearing to be "Simón Pineda Agudelo", written in a cursive style. The signature is positioned to the right of the name "Matías Macías Gómez" and above a horizontal line.

A la memoria de Jorge Luis Gómez Cano y Marcela Cardona Muñoz

AGRADECIMIENTOS

Un agradecimiento a José Ricardo Zapata por su acompañamiento y guía para la realización y planteamiento de este proyecto y en especial a Juan F. Franco, nuestro asesor, por habernos guiado a lo largo de todo el proceso y toda nuestra carrera universitaria.

Al grupo de docentes de la Facultad de Ingenierías y, en especial, al cuerpo docente de Ingeniería en Diseño de Entretenimiento Digital por su labor y compromiso con nuestra educación.

A nuestros amigos y compañeros, por su apoyo y compañía a lo largo de nuestra vida universitaria.

Y en especial, infinitas gracias a nuestros padres Luis Fernando Macías, Diana Cecilia Gómez, Jesús William Pineda y Adriana María Agudelo por haber hecho esto posible, por su amor incondicional, constante apoyo, acompañamiento y paciencia.

CONTENIDO

| | |
|---|-----------|
| RESUMEN | 14 |
| INTRODUCCIÓN | 16 |
| PLANTEAMIENTO DEL PROBLEMA | 17 |
| OBJETIVOS | 18 |
| GENERAL | 18 |
| ESPECÍFICOS | 18 |
| JUSTIFICACIÓN | 19 |
| METODOLOGÍA | 21 |
| TECNOLOGÍAS | 21 |
| EXPERIMENTACIÓN CON LAS TECNOLOGÍAS | 27 |
| PRUEBAS PRELIMINARES | 33 |
| ENTENDIMIENTO Y PREPARACIÓN DE LOS DATOS | 35 |
| DISEÑO DEL SISTEMA | 37 |
| INTERFAZ GRÁFICA Y MÓDULOS | 42 |
| DESARROLLO DEL SISTEMA | 44 |
| GESTIÓN DEL PROYECTO | 44 |
| ARQUITECTURA Y FUNCIONAMIENTO | 46 |
| RESULTADOS | 53 |

| | |
|------------------------|-----------|
| | 7 |
| CONCLUSIONES | 56 |
| RECOMENDACIONES | 58 |
| BIBLIOGRAFIA | 59 |
| ANEXOS | 61 |

LISTA DE FIGURAS

| | |
|---|-----------|
| FIGURA 1. INTERFAZ DEL AXIS NEURON | 21 |
| FIGURA 2. INTERFAZ INICIAL DE TOUCHDESIGNER | 23 |
| FIGURA 3. DIAGRAMA DE CONEXIÓN DE HERRAMIENTAS. | 24 |
| FIGURA 4. SKETCHED TOUCH DESIGNER | 28 |
| FIGURA 5. TOUCHOSC | 29 |
| FIGURA 6. RACKS DE ABLETON LIVE | 30 |
| FIGURA 7. POSES DE CALIBRACIÓN. | 33 |
| FIGURA 8. USO DE DATOS MOTION | 35 |
| FIGURA 9. MODELO DE UN PARÁMETRO | 38 |
| FIGURA 10. DIAGRAMA DE FLUJO DE SISTEMA DE MAPEO | 40 |
| FIGURA 11. COMPARACIÓN INTERFAZ ABLETON LIVE E INTERFAZ TDAXIS | 41 |
| FIGURA 12. LAYOUT DE LA INTERFAZ | 42 |
| FIGURA 13. ESTRUCTURA DE CARPETAS EN GOOGLE DRIVE | 44 |
| FIGURA 14. ARQUITECTURA FINAL DEL TDAXIS | 45 |
| FIGURA 15. ARQUITECTURA DEL AXIS MAPPING MODULE | 46 |
| FIGURA 16. PANELES INTERFAZ GRÁFICA DEL AXIS MAPPING MODULE | 47 |
| FIGURA 17. MODELO DEL PATCH | 48 |
| FIGURA 18. ARQUITECTURA DEL PATCH INPUT MODULE | 48 |
| FIGURA 19. TOX SELECTOR VIEW Y TOX PARAMETER VIEW. | 49 |
| FIGURA 20. ARQUITECTURA DEL VIDEO OUTPUT MODULE | 50 |
| FIGURA 21. POST PROCESSING RACK VIEW EN EL TDAXIS | 50 |

| | |
|---|-----------|
| FIGURA 22. RESULTADOS POST PROCESSING RACK | 50 |
| FIGURA 23. ARQUITECTURA ABLETON MAPPING MODULE | 51 |
| FIGURA 24. GENERACIÓN VISUAL | 52 |

GLOSARIO

TOUCHDESIGNER: es una plataforma de desarrollo visual que permite crear proyectos que corren en tiempo real. Sirve para controlar y generar video mapping en vivo, visualizaciones reactivas a música o prototipado rápido. [1]

ABLETON LIVE: software para creación musical y puestas en escena en vivo. [2]

NOTION: software de administración de proyectos colaborativo basado en la nube.

GOOGLE DRIVE: servicio de almacenamiento basado en la nube que ofrece Google.

EXPRESIVIDAD PERFORMÁTICA: es la capacidad de manifestar sentimientos o pensamientos en un acto o puesta en escena en vivo.

PHYSICAL COMPUTING: es crear, construir o diseñar sistemas interactivos que utilizan diferentes tipos de software y de hardware para medir y responder a diferentes tipos de estímulos.

MULTIMEDIA: cualquier objeto que haga uso de diferentes medios (ej. texto, video, sonido...), ya sean analógicos o digitales, en sincronía para comunicar un mensaje o un concepto.

CAPTURA DE MOVIMIENTOS CORPORALES: también llamado MoCap es el proceso de grabación de movimientos tanto corporales como de objetos, con el fin de utilizarse para estudios médicos, deportivos, militares o más recientemente, en la industria del arte y tecnología. [3]

ARTE GENERATIVO: o arte por computadores lo entenderemos en este trabajo como toda creación artística realizada de manera parcial o total por un sistema autónomo no-humano (un computador) que genere una pieza visual o sonora basándose en el input y/o los datos generados o introducidos por un usuario al sistema. [4]

INTERFAZ HUMANO-MÁQUINA: es el medio por el cual un humano y una máquina pueden interactuar el uno con el otro.

TOX: formato de archivo de TouchDesigner encapsulado en un nodo que se puede utilizar dentro de otros archivos de TouchDesigner.

RACK: hace referencia a la manera en la cual se organizan los componentes de hardware en un estante dedicado. En el contexto del contenido audiovisual digital es un módulo compuesto de varios efectos para señales de audio o video.

FBX: formato de archivo para distribuir animaciones, modelos 3D, etc.

RESPONSIVIDAD: es un concepto de informática que hace referencia a la capacidad específica de un sistema o unidad funcional para completar las tareas asignadas en un tiempo determinado.

[5]

RESUMEN

La danza contemporánea nació como una búsqueda para aumentar la expresividad corporal de los bailarines. En esta búsqueda se utilizaron todas las tecnologías disponibles en el momento. Así pues, se propone utilizar las tecnologías disponibles actualmente, en este caso concreto se utilizó TouchDesigner, Ableton Live y el Perception Neuron. Esto con el fin de desarrollar un sistema que les permita a los bailarines utilizar sus cuerpos como una interfaz de generación y control de contenido multimedia (IGCCM) en tiempo real para aumentar sus posibilidades de expresión escénica, mapeando sus movimientos a imágenes o sonidos.

PALABRAS CLAVE:

Interfaz Natural De Usuario (NUI); Movimiento Corporal; Multimedia; Performance; TouchDesigner.

ABSTRACT

Contemporary dance was born as a search to increase the body's expressiveness of dancers. In this search, they used all the available technologies at the time. Therefore, we propose the implementation of currently available technologies, in this project TouchDesigner, Ableton Live and Noitom's Perception Neuron were used in order to develop a system that allows dancers to use their bodies as an interface for generation and control of multimedia (IGCCM) content in real time, that allows the increase of their scenic expression by mapping their movements to images or sounds.

KEYWORDS:

Natural User Interface (NUI); Body Movement; Multimedia; Performance; TouchDesigner.

INTRODUCCIÓN

A partir de las investigaciones que se han realizado en el campo de la interacción Humano-Máquina (HCI) se ha descubierto que utilizando computadores se pueden detectar, procesar, interpretar y reaccionar a diferentes estímulos, emociones y acciones humanas de manera muy precisa y en tiempo real.

Basados en esto, vemos la oportunidad de crear un sistema que permita utilizar los movimientos del cuerpo para generar y controlar contenido multimedia [6] brindando la experiencia de sentir el cuerpo como un instrumento de creación vivo, con la finalidad de transformarlo en una interfaz natural de usuario que aumenta la expresividad performática. Esto, mediante el uso de tecnologías de captura de movimiento, técnicas de procesamiento de datos en tiempo real y arte generativo.

Teniendo a la interacción humano-máquina como uno de los pilares del diseño de experiencias interactivas, la temática central del proyecto es aplicar el concepto de physical computing a la generación de experiencias significativas y memorables. Por ello, se busca diseñar un sistema de generación y control de contenido multimedia utilizando tecnologías de captura de movimientos corporales en tiempo real, en este caso concreto utilizaremos el Perception Neuron de Noitom; TouchDesigner como lenguaje integrador entre el mundo performático y el mundo digital y Ableton Live como procesador de audio. Esto en busca de su implementación dentro del contexto del arte performático como una interfaz natural de usuario que aumenta la expresividad escénica.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

En las puestas en escena en vivo las superficies de control y generación de contenido audiovisual disponibles, utilizan como principales medios de interacción con el usuario pantallas, botones, potenciómetros y suiches. Este tipo de interfaces funcionan muy bien siempre y cuando el operario pueda permanecer quieto durante el espectáculo. Sin embargo, en el caso de los artistas performáticos, y específicamente en el de los bailarines contemporáneos, permanecer quieto no es posible y se les dificulta el manejo en tiempo real del contenido audiovisual que los acompaña. Esto los obliga a utilizar contenido audiovisual previamente producidos o depender de equipos de personas para que se encarguen de generar y/o controlar dicho contenido. Relegando todas la imágenes y sonidos a simples acompañantes o entornos en los que se desarrolla el performance, limitando el control que el artista tiene sobre su puesta en escena y afectando su capacidad de expresión performática.

OBJETIVOS

GENERAL

Diseñar un sistema de generación y control de contenido multimedia, basado en el Perception Neuron como herramienta de captura de movimientos corporales en tiempo real, TouchDesigner como lenguaje integrador entre el mundo performático análogo y el mundo digital y motor visual y Ableton Live como procesador de audio; con el fin de convertir el cuerpo humano en una interfaz humano-máquina que aumenta la expresividad performática.

ESPECÍFICOS

1. Determinar las propiedades, atributos clave y las relaciones más importantes de los datos que se obtienen al utilizar el Perception Neuron, y su apropiada asociación con parámetros dentro de TouchDesigner y Ableton Live.
2. Desarrollar en TouchDesigner un sistema modular que permita generar y controlar contenido multimedia utilizando como parámetros de control los atributos determinados a partir de pruebas realizadas a diferentes usuarios con el Perception Neuron.
3. Probar el comportamiento y el funcionamiento de las diferentes parametrizaciones y mapeos dentro de TouchDesigner y Ableton Live para controlar y generar contenido audiovisual utilizando datos capturados con el Perception Neuron con un usuario.

JUSTIFICACIÓN

Desde los años 50, con el nacimiento de la danza contemporánea, que fue una respuesta a la danza clásica, se comenzaron a buscar maneras de expresarse más libremente con el cuerpo humano; la danza dejó de ser un instrumento para contar historias y se convirtió en un medio para expresar emociones y transmitir conceptos más abstractos.

En esta búsqueda por una mayor expresividad se utilizaron todos los medios audiovisuales y tecnológicos disponibles para la época: música, fotografía, y montajes escenográficos que permitieran potenciar el mensaje del bailarín, ya que, para la danza contemporánea, en palabras de Martha Graham: “El instrumento mediante el que se expresa la danza es también el instrumento mediante el que se vive la vida: el cuerpo humano” [7].

Teniendo en cuenta que la danza contemporánea nunca ha sido ajena a la idea de usar la tecnología para potenciar la transmisión de sus mensajes y conceptos, se hace oportuno desarrollar un sistema que permita la colaboración en tiempo real entre un humano y un computador para así aumentar las posibilidades expresivas del intérprete [8] en una colaboración en la cual la música, las luces y las visuales no son sólo el contexto dentro del cual se desarrolla la puesta en escena, sino que se convierten en agentes activos y reactivos que potencian el performance del bailarín y que gracias a la capacidad de los computadores pueden interpretar, reaccionar y responder al intérprete como si fueran otro agente no-corpóreo en escena. Por lo tanto, la creación de este sistema posibilita la profunda colaboración entre humanos y computadores en contextos de creación y expresión. En una búsqueda para liberar al intérprete

de la necesidad de interactuar directamente, durante el performance, con otros sistemas tecnológicos que puedan inhibir o dificultar su proceso creativo y/o performático.

CAPÍTULO 2

METODOLOGÍA

TECNOLOGÍAS

Previo al desarrollo del proyecto se decidieron las tecnologías con las que se desarrollaría el mismo, esto a partir de comparaciones realizadas entre las herramientas disponibles en el momento del planteamiento de este proyecto.

La primera tecnología que se definió fue el Perception Neuron de Noitom como sistema de captura de movimientos en tiempo real. Este se comparó con el Kinect V.1 de Microsoft y con cámaras web utilizando un sistema de visión artificial. Sin embargo, su precisión al momento de reconocer movimientos, su robustez ante diversas condiciones lumínicas, y su rango de operación, ya que no está sujeto a un campo de visión, convirtieron al Perception Neuron en la herramienta ideal para cumplir con la visión y el objetivo del proyecto [3].

Una vez definido el Perception Neuron, se debió definir en qué entorno se desarrollaría el producto principal del proyecto. Este debía recibir los datos provenientes del Axis Neuron (*Software del Perception Neuron, figura 1*), procesarlos, normalizarlos, mapearlos y luego vincularlos a parámetros en un programa de control y generación de visuales y audio, todo esto con una interfaz de usuario que facilitara dicho proceso.

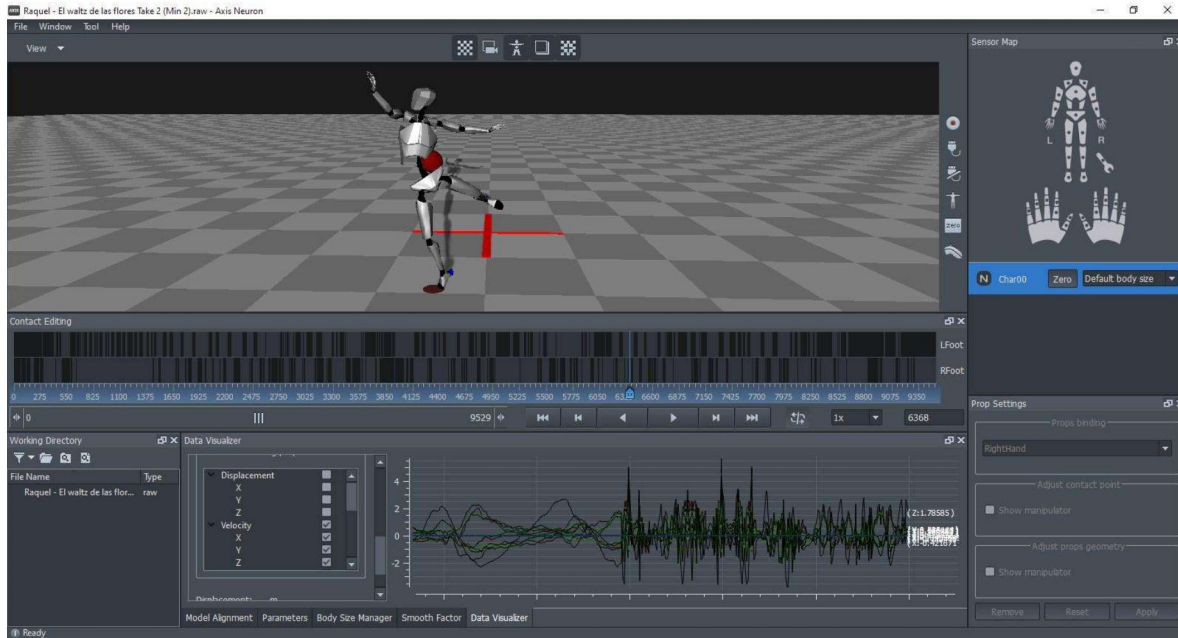


Figura 1. Interfaz del Axis Neuron

En esta búsqueda se consideraron varias opciones: TouchDesigner, OpenFrameworks, Notch, VVVV, Processing, JUCE y Unity. Para decidir cuál era la mejor opción se definieron los siguientes puntos: precio, documentación, facilidad de desarrollo, conectividad integrada (*elementos incluidos en la herramienta listos para ser usados*), potencia gráfica, uso principal (*para que cosas está hecha la herramienta*) y en base a ellos se construyó una matriz comparativa en la que a cada opción se le asignaba un valor entre cero y tres (0 y 3) por cada una de los puntos definidos y luego se sumaron los puntos, esto se puede observar en la tabla 1. La opción con el mayor puntaje sería la escogida para el desarrollo del proyecto.

Tabla 1

Matriz de Comparación de Herramientas

| Nombre | Precio | Docs. | Facilidad | Integración | Potencia | Uso ppal. | Total |
|----------------|--------|-------|-----------|-------------|----------|-----------|-------|
| TouchDesigner | 3 | 2 | 3 | 3 | 3 | 3 | 17 |
| OpenFrameworks | 3 | 2 | 1 | 1 | 3 | 2 | 11 |
| Notch | 0 | 1 | 2 | 3 | 3 | 3 | 12 |
| VVVVVV | 3 | 1 | 1 | 2 | 3 | 3 | 13 |
| Processing | 3 | 3 | 3 | 1 | 1 | 2 | 13 |
| JUCE | 3 | 1 | 1 | 1 | 2 | 1 | 9 |
| Unity | 3 | 3 | 2 | 1 | 2 | 1 | 12 |

En términos generales, los resultados de esta matriz se pueden resumir en que Notch se descartó porque el precio de las licencias es demasiado elevado y la versión de prueba no permite guardar las sesiones; en el caso de JUCE y OpenFrameworks, la facilidad de desarrollo es muy baja, y su conectividad integrada es limitada; Processing tiene una potencia gráfica limitada, y su conectividad integrada también lo es; Unity tiene una conectividad integrada limitada; finalmente, TouchDesigner es el entorno que obtuvo el mayor puntaje, obteniendo puntaje casi perfecto en todos los puntos establecidos. Además de esto, se encontró una gran cantidad de proyectos desarrollados en TouchDesigner (*figura 2*) hechos en específico para performance en tiempo real, dándole una gran credibilidad como herramienta para esto.

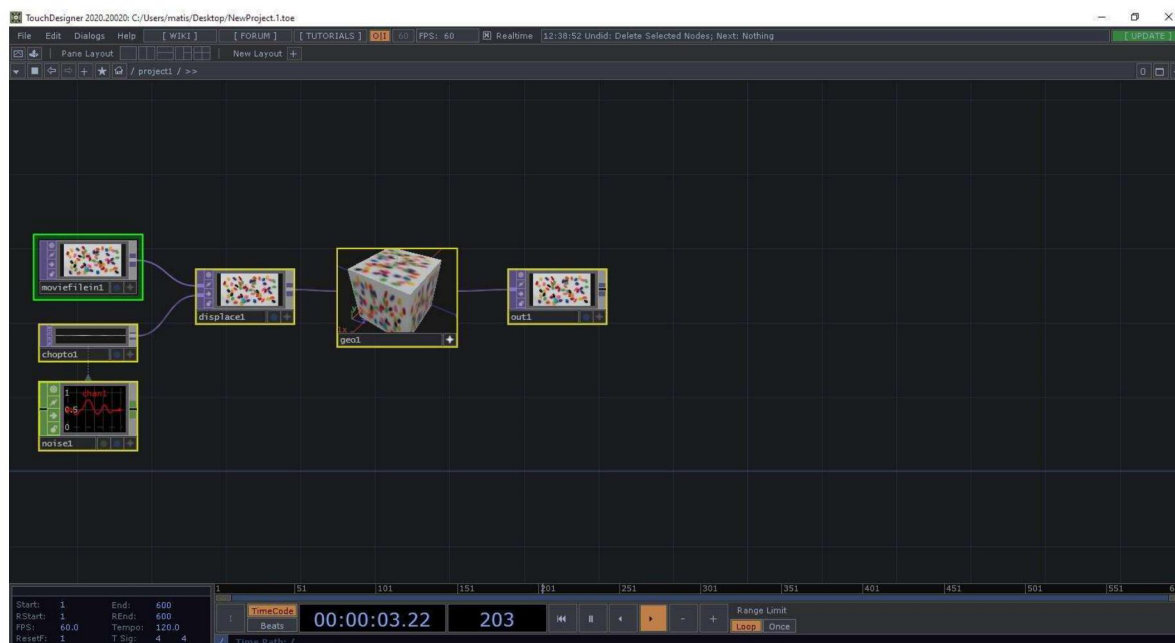


Figura 2. Interfaz inicial de TouchDesigner.

Luego, se necesitaba una herramienta para poder controlar y manejar archivos de audio; sintetizadores; múltiples pistas de audio con controles de panning, volumen y efectos independientes por pista; y que tuviera algún tipo de conectividad con otros programas. Para escoger esta herramienta se basó en el entorno previamente definido, es decir, TouchDesigner, el cual tiene integrado un nodo y un protocolo para comunicarse y controlar de manera sencilla y directa a Ableton Live mediante Max/MSP, un lenguaje de programación visual, que permite comunicación bidireccional (*enviar y recibir información*) con TouchDesigner. Es importante aclarar que el precio de las licencias de Ableton Live es elevado, sin embargo, ya se contaba con una, por lo cual no fue un inconveniente para el desarrollo del proyecto. Por las razones anteriores, otras opciones como PureData se descartaron y se decidió trabajar con Ableton Live. Al finalizar este proceso ya se tenían las herramientas necesarias para cumplir los objetivos previamente definidos: el Perception Neuron para capturar los datos del movimiento corporal en tiempo real, TouchDesigner como entorno principal de desarrollo, centro de conexiones,

principal interfaz de interacción con los usuarios y generador de contenido visual, Ableton Live como procesador de audio y Max/Msp como interfaz de conexión entre TouchDesigner y Ableton Live. A continuación, en la figura 3, se puede ver el diagrama de conexión de las herramientas.

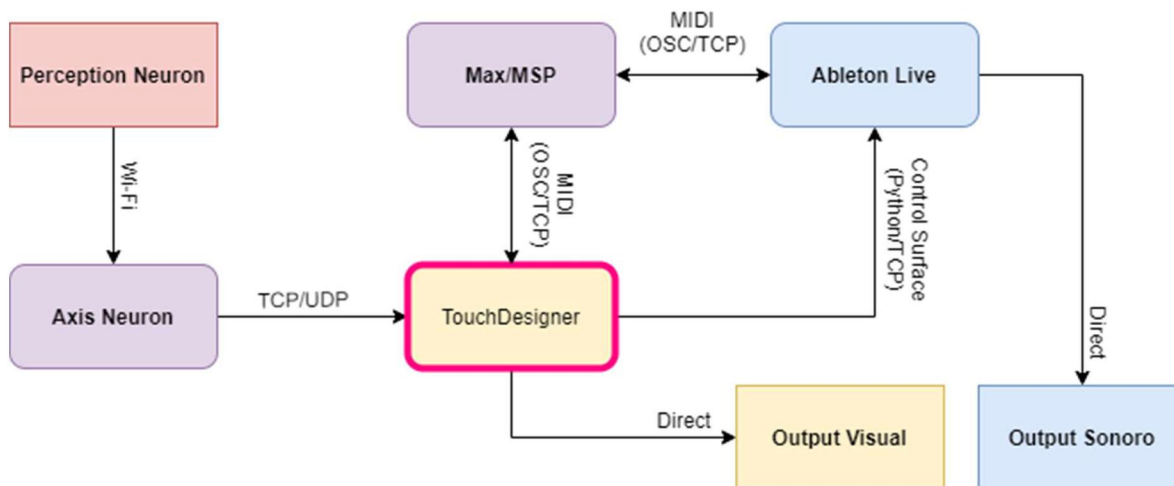


Figura 3. Diagrama de conexión de herramientas. Rectángulos representan hardware, rectángulos curvos representan software. Rojo es el sistema de MoCap, morado es software intermedio, amarillo es visual, y azul sonoro. Bordes rosados es centro de conexiones e interfaz para usuarios.

Para realizar la gestión del proyecto, se buscaron herramientas que permitieran hacer seguimiento al desarrollo del mismo; llevar registro de los avances, problemas y resultados; y que fuera basado en la nube para poder acceder a él y actualizarlo o revisarlo desde cualquier dispositivo. Además de esto, se buscó un servicio de almacenamiento en nube que permitiera colaborar y compartir carpetas sincronizadas con los computadores del equipo de trabajo de manera dinámica y que de esta manera todos los miembros tuvieran acceso constante a los archivos más actualizados del proyecto. Para cumplir estas funciones se utilizaron dos (2) herramientas en conjunto. La primera es Notion, que se utilizó como diario de campo. En este se hizo seguimiento de los avances, los problemas, los resultados y se crearon bases de datos en las

cuales se registraron todas las nuevas fuentes de información que se iban utilizando, los registros audiovisuales que se iban tomando y los prototipos que se fueron desarrollando a lo largo del proyecto. Además, se utilizó Google Drive como almacenamiento en nube que permitiera compartir y sincronizar archivos y documentos entre varios computadores. Ambas herramientas se eligieron asumiendo que eran las más óptimas para realizar el seguimiento del proyecto dada la familiaridad que se tenía trabajando con ellas.

EXPERIMENTACIÓN CON LAS TECNOLOGÍAS

Tras definir las tecnologías que se iban a utilizar para el desarrollo del proyecto, se comenzó un proceso de exploración, en el cual se realizaron diferentes pruebas con cada una ellas.

La primera tecnología en ser explorada fue el Perception Neuron. Para esto, se realizaron pruebas para entender cómo funcionan sus protocolos de comunicación, cómo transmitir los datos en tiempo real y que tan precisas son las capturas de movimiento. Las pruebas se realizaron durante cinco (5) días en cinco (5) sujetos de prueba diferentes, tres (3) no bailarines y dos (2) bailarines profesionales, esto con el propósito de ver que tanto influye la técnica de la persona en la precisión de la captura de los datos de movimiento. Se encontró que la técnica de baile de la persona con el traje influye bastante en la precisión de la toma de datos, incluso más que la velocidad y la potencia de los movimientos.

Mientras menos técnica tenía una persona, la necesidad de calibrar nuevamente y la probabilidad de que el traje falle en la mitad del performance aumentaban.

Los resultados de cada día se pueden leer accediendo al anexo B, accediendo a la página principal y posteriormente accediendo a la página de cada fase.

De seguido, se comenzó a explorar con los datos que se producían al utilizar el traje, cómo transmitirlos y qué protocolos de comunicación utilizar. Se enfocaron los esfuerzos en la comunicación entre Axis Neuron y TouchDesigner ya que era donde se encontraba la mayor dificultad técnica. Luego de entender el funcionamiento del Perception Neuron como tal, transmitir datos entre Axis Neuron y TouchDesigner era el siguiente paso. Inicialmente el objetivo fue transmitir los datos de movimiento ya que eran los más complejos. Para esto se tenían dos (2) opciones: *BVH*, o *Advanced BVH*. La principal diferencia entre estas dos opciones

es que la primera es una comunicación simplex que ofrece la posibilidad de transmitir sus datos en ASCII o en binario, mientras que la opción Advanced BVH es una comunicación dúplex que solo se puede transmitir en binario. Noitom, el fabricante y desarrollador del Axis Neuron recomienda utilizar la opción Advanced BVH [9] ya que al ser una comunicación dúplex brinda la posibilidad de tener un sistema más robusto, que pueda detectar fallos en la transmisión, nuevos actores en escena, momentos de calibración, etc. Sin embargo, implementar este protocolo de comunicación presentaba un reto mayor al ser un protocolo mucho más complejo que el BVH simple. La primera dificultad que se encontró al intentar implementar este protocolo fue que la trama binaria estaba comprimida en formato zip. Para descomprimir la trama se utilizó la librería de python gzip. La siguiente dificultad encontrada fue la necesidad de usar el uso método de serializado de información de Google, *Protobuf*. Sin embargo, leyendo la documentación del Axis Neuron se logró des encriptar la trama sin ningún problema. En este punto ya se tenía todo lo necesario para comunicarse con el servidor de datos del Axis Neuron, sin embargo, para que este comenzara la transmisión de datos era necesario enviar un Handshake al servidor y no se encontró documentación suficiente para lograr hacerlo. Por ello, se descartó este protocolo de comunicación y se pasó a BVH simple por ASCII. Este protocolo, aunque en términos técnicos es más simple, implementarlo tampoco fue fácil ya que no había documentación suficiente. Con este protocolo, desde el momento en el que se realiza la conexión al servidor que transmite los datos, los datos se comienzan a transmitir. Axis Neuron transmite datos 120 veces por segundo, sin embargo, esto se podía reducir a la mitad, a un cuarto, o a un octavo. Los datos se transmitían en una sola trama separada por espacios. El reto en implementar este protocolo estuvo en descifrar el orden de los datos. Habiendo implementado este protocolo exitosamente, se determinó que reducir la velocidad de transmisión a un cuarto de la original era

óptimo, ya que esto aseguraba la estabilidad de la transmisión y no reducía la responsividad del sistema.

Para aumentar la familiaridad y destreza con TouchDesigner, cada mañana se plantearon retos de programación que debían ser resueltos antes de mediodía (*figura 4*). Estos consistían en recrear, utilizando TouchDesigner, piezas de arte generativo inspiradas en imágenes de internet (anexo B en la sección de TouchDesigner y anexo D en el repositorio TD-Sketches). Estos retos, además de servir como herramientas de estudio, serían los insumos para las posteriores pruebas con el sistema de generación y control de contenido visual.

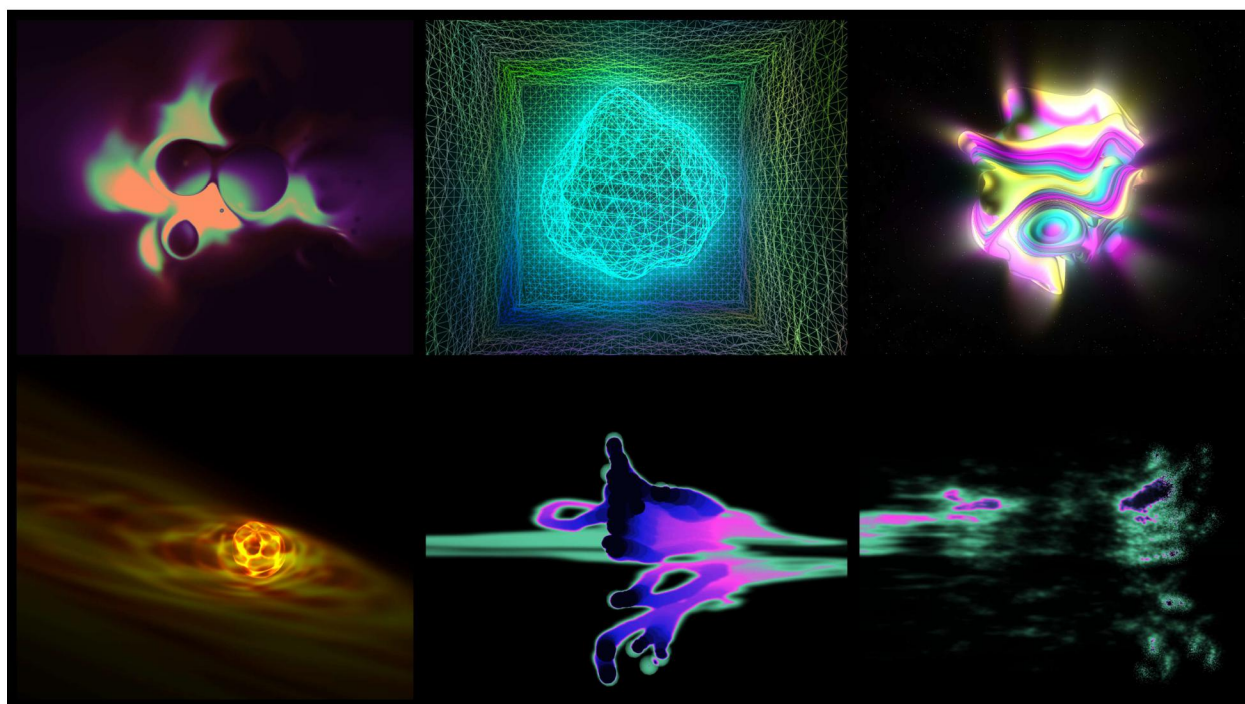


Figura 4. Sketched Touch Designer.

Luego de tener completados varios de estos retos, se utilizaron para realizar una serie de pruebas de interactividad, generación, control y conectividad inalámbrica en TouchDesigner. Para esto se utilizó la aplicación TouchOSC (*figura 5*) en un smartphone, que permite crear

interfaces de control personalizadas con botones, potenciómetros, faders, pads XY e interruptores. Con este montaje se podían controlar diferentes parámetros en TouchDesigner con el celular mediante conexión OSC. Esto sirvió especialmente para entender cómo TouchDesigner administra las conexiones inalámbricas y fue un primer acercamiento a sus capacidades performáticas.

Las pruebas con Max/Msp y Ableton Live en conjunto se realizaron con un enfoque diferente, ya que con estas herramientas se tenía experiencia previa, tanto en ámbitos de programación como en entornos performáticos, pero no se tenía experiencia vinculando estas herramientas a TouchDesigner. Es por esto que los experimentos se enfocaron en entender mejor el comportamiento, la estabilidad y sobre todo las capacidades de generación y control de contenido sonoro.

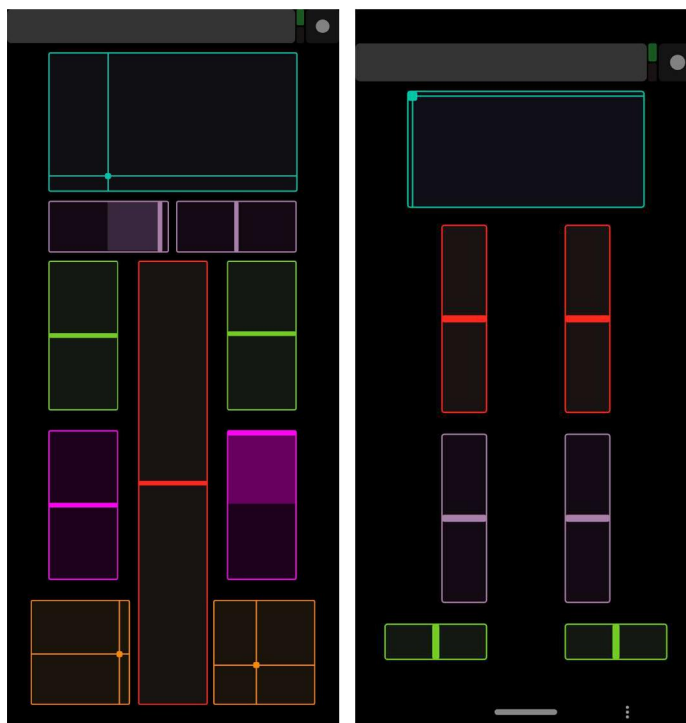


Figura 5. TouchOSC. Interfaces personalizadas para simular el movimiento de los sensores del cuerpo, la imagen de la izquierda representa a la parte superior del cuerpo y la de la derecha la parte inferior.

Para la realización de los experimentos se compusieron y produjeron varias piezas musicales que incluían diferentes tipos de pistas con variedad de sintetizadores, envíos y racks de efectos (figura 6).



Figura 6. Racks de instrumentos, sintetizador y efectos en Ableton Live. junto con los plugins de Max4Live que permiten la conexión con TDableton.

Una vez se habían realizado varias pruebas utilizando TouchDesigner y TDableton como superficie de control de Ableton Live, se comenzaron las pruebas de estabilidad. Estas se realizaron utilizando las sesiones de Ableton Live previamente creadas para los experimentos creativos y los programas de arte generativo programados en TouchDesigner. La primera prueba se realizó en un solo computador conectando TouchDesigner y Ableton Live por la red local del equipo. Este montaje ofreció los mayores beneficios en términos de estabilidad y facilidad de conexión, pero era demasiado exigente para el procesador, por lo cual limitaba la cantidad de efectos que se podían utilizar tanto en TouchDesigner como en Ableton Live.

La siguiente prueba se realizó ejecutando TouchDesigner y Ableton Live en dos equipos diferentes conectados por red a través del hotspot de un smartphone. Este sistema funcionó bien, con esporádicas caídas de frames y saltos en los cambios de los parámetros cuando se realizaban

desde un programa al otro; sin embargo, este montaje presentó problemas cuando también se ejecutaba el Axis Neuron de manera simultánea con los otros dos programas. En estos casos la comunicación entre las herramientas se volvía inestable, intermitente y reducía su responsividad al punto de ser inutilizable. Finalmente, se realizaron pruebas con los programas corriendo en equipos diferentes, conectados por red a través de un router dedicado, dando como resultado la mejor estabilidad, y rendimiento de los equipos.

PRUEBAS PRELIMINARES

Si bien el proyecto, se justifica desde la danza contemporánea y los problemas que se detectaron como espectadores, se decidió de manera premeditada no cerrar la realización de las pruebas a bailarines contemporáneos, para poder extrapolar los resultados a las artes performáticas en general.

En total se realizaron pruebas con cinco (5) personas, incluidas personas ajenas a las artes performáticas, bailarines casuales y bailarines profesionales de ballet (anexo B, sección de personas). Con cada una de estas se realizaron hasta 4 tipos de pruebas diferentes. Las dos pruebas más básicas se nombraron *prueba de precisión* y *prueba de caminata*. El propósito de las pruebas de precisión fue comprobar qué tan precisos eran los datos producidos por el Perception Neuron en relación a la constitución física de la persona con la que se hacía la prueba; con el fin de determinar limitaciones y/o requisitos para el correcto uso del Perception Neuron. Las siguientes pruebas fueron de distancia, y se realizaron con el propósito de determinar el alcance máximo del Perception Neuron según el router que se utilizara, ya fuera el hotspot de Wi-Fi de un celular Xiaomi Mi 9T Pro, o un router TP-Link modelo TL-WR841HP (anexo B, repositorio de hardware y software). Por último, las pruebas de caminata se realizaron con el propósito de determinar si los datos de posicionamiento producidos por el Perception Neuron eran confiables. El tercer tipo de prueba se realizó únicamente con tres de las cinco personas. En esta prueba se solicitaba bailar algo a elección. Esta prueba cumplió varias funciones, la primera fue observar el comportamiento del Perception Neuron ante situaciones reales tales como movimientos fuertes, giros, saltos, arrastrarse, entre otros. La segunda función fue recopilar

grabaciones para poder simular una persona realizando un performance cuando fuera necesario, sin necesidad de realizar el montaje completo del Perception Neuron.

Con cada persona con la que se realizaron pruebas se siguió el mismo procedimiento.

En primera instancia se les colocó el traje y se midió el tiempo necesario para montar el Perception Neuron: sacar el traje del estuche, buscar las piezas necesarias, montar las neuronas y colocarlas en la persona; luego, se comenzaba un ciclo en el que se empezaba por calibrar (*figura 7*) y a continuación se realizaba una prueba de precisión. Este ciclo se repetía modificando los parámetros corporales del Axis Neuron para intentar obtener una mejor calibración. Cuando se sentía que la calibración no mejoraba, se cerraba este ciclo con una prueba de caminata. Después de esto, según los objetivos que se tuvieran planteados para el día y las habilidades de la persona, se realizaba la prueba de alcance o la grabación de un performance.

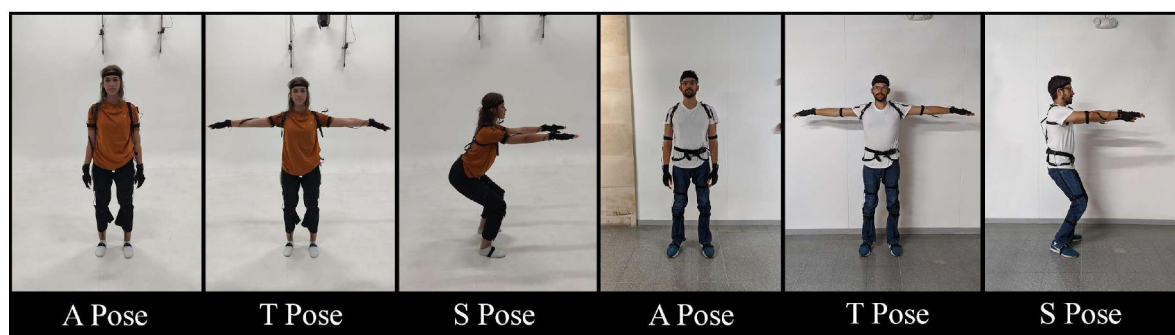


Figura 7. Poses de calibración.

ENTENDIMIENTO Y PREPARACIÓN DE LOS DATOS

Al profundizar en el funcionamiento del Perception Neuron y leer su manual de usuario, se descubre que hay 3 conjuntos de datos que se pueden transmitir en tiempo real mediante el Axis Neuron. Estos tres conjuntos corresponden a datos de cálculo (calculation data), datos de movimiento (motion data) y datos de acción (action data) [9].

Calculation Data hace referencia a los datos provenientes de cada uno de los sensores, preprocesados o no. Estos son: desplazamiento relativo al centro de calibración, velocidad, cuaterniones, aceleración, los datos provenientes de los giroscopios y el estado de contacto de los pies. Estos datos, a excepción de los contactos de los pies, vienen en una matriz de tamaño 14 x 24 [9].

Motion Data hace referencia a un conjunto de datos que representa el movimiento de la persona que usa el traje. Son datos que describen las posiciones de una estructura jerárquica de nodos, teniendo como nodo inicial la cintura. Este se considera el nodo principal y partiendo de este se construye la estructura de un cuerpo utilizando desvíos estáticos que se acumulan a medida que se profundiza en la jerarquía. El movimiento de esta estructura viene dado por la rotación de cada uno de los nodos con pivote en el nodo directamente superior en la jerarquía. Estos datos están basados en el estándar BVH (BioVision Hierarchy) [10].

Action Data hace referencia a un conjunto de datos en el que hay algunos gestos predefinidos por el Axis Neuron. Se decidió no utilizar los datos de acción ya que los gestos que estos datos detectan no son relevantes para el propósito del sistema.

En este punto se tenían dos (2) tipos de datos que se podrían utilizar en el sistema final. Los datos de movimiento se podrían utilizar para crear visualizaciones en las que la postura y la forma del cuerpo del artista tendrán influencia directa sobre la geometría del resultado, mientras que los datos de cálculo se utilizarían para mapearlos a diferentes parámetros en el sistema. A este último conjunto de datos se le aplicaron varias transformaciones y modificaciones ya que su forma original no era apropiada para el uso que se le quería dar. La primera transformación que se realizó fue eliminar los cuaterniones ya que la representación física de estos es muy abstracta para ser utilizada de forma consciente por un artista. La segunda transformación que se hizo fue agregar la capacidad de obtener datos de distancia entre dos nodos. Finalmente, se agregó la capacidad de obtener la magnitud de la distancia, la velocidad, la rotación o la aceleración. En la figura 8 se puede observar un posible uso de los datos de movimiento.

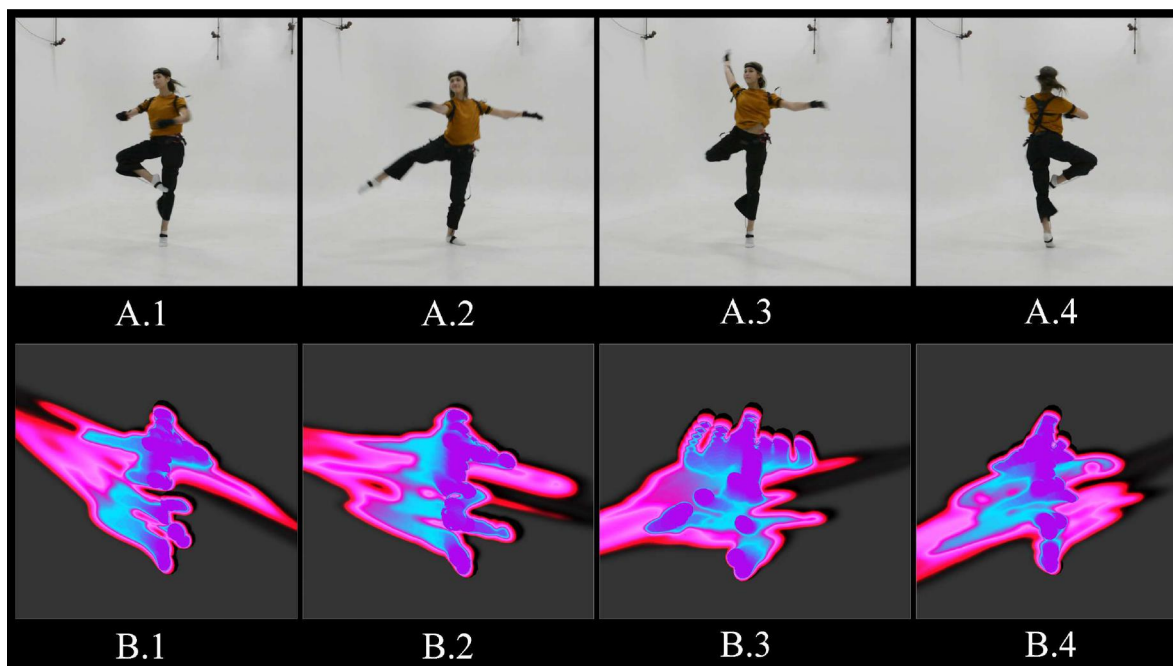


Figura 8. Uso de datos Motion. En la fila A se presenta una secuencia de imágenes obtenidas de un video que se filmó durante una sesión de captura de movimiento. En la fila B se pueden ver los resultados obtenidos con dichos datos. La desincronización se debe a la no sincronización de los FPS de la cámara y la captura.

DISEÑO DEL SISTEMA

Según Kai Tuuri [11], las interfaces y tecnologías con las que interactuamos, especialmente en contextos creativos y performáticos tienen un efecto directo sobre el resultado final. Ya que la influencia, el control y la comunicación con el sistema se dan de manera bidireccional. El usuario manipula e interactúa con el sistema y a su vez, sus decisiones y acciones creativas se ven influenciadas por el sistema mismo. El sistema no solo es el ente que determina los límites de las posibilidades escénicas, sino que también controla el flujo creativo que el artista sigue a lo largo de su creación.

Por esto, el diseño de la interfaz, su flujo de trabajo y las posibilidades técnicas que ofrece a los usuarios fueron factores definitivos en cuanto al resultado artístico final que se podría obtener con el sistema.

Para definir el flujo del sistema se comenzó con la definición de sus funciones principales, el control y la generación.

Control del contenido se entendió como la manipulación de parámetros de salida. Ejemplo: filtros, cambios de color, volumen, panning, o la cantidad enviada a un efecto de un canal de audio. En otras palabras, todos los efectos que se le colocan a contenidos ya producidos.

Por otro lado, generación se entendió como la manipulación de parámetros de entrada. Ejemplo: tipos de geometría, notas MIDI, velocidad de un ruido, etc. Con estos se produce el contenido, que posterior a esta creación podrá ser manipulado con diferentes tipos de efectos [4].

Se planteó esta diferencia porque desde un principio se vio la necesidad de poder interactuar con contenido previamente producido. En este caso el objetivo no es generar una pieza visual o sonora in situ, sino tener más control sobre la salida de piezas previamente creadas y que su presentación al público sea modificada con los movimientos del artista [12].

Una vez definidos estos conceptos en el contexto del proyecto, se pasó al planteamiento de un sistema que permitiera la vinculación de los datos obtenidos del Perception Neuron con parámetros en TouchDesigner y Ableton Live, y así poder generar y controlar contenido audiovisual.

Para esto se diseñó un flujo de trabajo fijo que permitiera vincular los datos de movimiento a cualquier tipo de parámetro. Esto con el objetivo de facilitar el uso de la herramienta a personas sin conocimientos técnicos.

Este flujo está dividido en cuatro (4) partes: parametrización de los movimientos del cuerpo, normalización de los datos, mapeo a una curva de expresión y vinculación a otro parámetro(s). Cualquier dato proveniente del Axis Neuron pasa por este flujo para ser mapeado a un parámetro en TouchDesigner o Ableton Live.

Dada la cantidad de datos que llegaban desde Axis Neuron, fue importante facilitar una manera de escoger qué parte del cuerpo, qué dato(s) se querían utilizar, y cómo se querían interpretar. Para lograr esto se diseñó un sistema que, en primera instancia, agrega una entrada a una lista relacionada a un dato proveniente del Axis Neuron (*Ej. distancia entre manos, rotación cabeza,*

etc.), este nombre serviría como etiqueta para identificar el dato, una vez creada la referencia al dato se le daban una serie de atributos definidos por el usuario para su posterior procesamiento. Estos atributos definían qué tipo de dato se quería obtener (*distancia, rotación, aceleración, velocidad, posición*), si se quería leer la magnitud de los datos, o sus componentes. En caso de que se leyeran los componentes del dato, se podía definir si se quería controlar su salida con una curva para cada uno o una sola curva para todos, a esto se le nombró *Multymap*. Finalmente se definía de qué nodo o nodos se querían leer los datos, es decir: cabeza, mano izquierda, pie derecho, etc.

Este proceso completo se denominó parametrización y se ilustra en la figura 9.

Habiendo establecido este proceso, todos los datos que se querían utilizar del Axis Neuron pasaban por él y quedaban en una lista desde la cual se podía volver a cada uno y ajustarlo a necesidad.

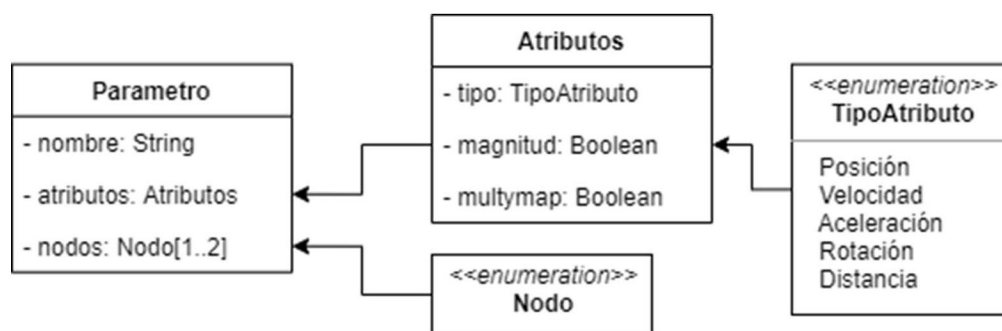


Figura 9. Modelo de un parámetro. Nodo hace referencia a un nodo del Perception Neuron.

Luego de la parametrización, dado que los datos provenientes del Axis Neuron estaban en rangos de valores diferentes dependiendo de su tipo y de la persona que estaba utilizando el traje, para facilitar su posterior mapeo y vinculación fue importante normalizarlos a un rango entre cero (0) y uno (1). Para esto el usuario debía ingresar los límites superiores e inferiores del dato a

normalizar. Se decidió que este proceso fuera controlado por el usuario ya que el rango de los datos varía dependiendo de las dimensiones del bailarín, dificultando la automatización de este proceso. Una vez normalizados, estos se mapeaban a una curva editable, con la cual se podía controlar si los movimientos tenían una salida lineal, cuadrada, exponencial o variaciones de las tres. Permitiendo a los artistas tener mucho más control sobre la interacción de sus movimientos y los parámetros que se desean controlar en TouchDesigner y Ableton Live. Finalmente, los datos resultantes de esta curva se volvían a mapear a otro rango de valores dependiendo de los del rango que el usuario definía como óptimos para el parámetro al cual se querían vincular. Al final de este proceso, se obtuvo un flujo de trabajo claro y repetible que permitía a los usuarios sin muchos conocimientos técnicos, controlar con los movimientos del cuerpo parámetros en TouchDesigner y Ableton Live, posibilitando la generación y control del contenido audiovisual.

La integración con Ableton Live se implementó mediante el módulo TD Ableton de TouchDesigner que funciona como un control surface [13]. Estos últimos son scripts que permiten controlar a Ableton Live mediante un agente externo, sea software o hardware. TD Ableton permitió enviar y recibir datos MIDI, controlar parámetros de efectos y sintetizadores; controlar volúmenes, paneos, envíos de cada canal y disparar clips de audio o MIDI de la vista sesión de Ableton Live.

Para integrar la funcionalidad del TD Ableton dentro del sistema, se utilizó un sistema parecido al que se implementó con los datos del Axis Neuron. Dependiendo de qué se quisiera controlar se creaba una entrada en una de 3 listas. La primera para recibir datos MIDI, la segunda para controlar parámetros de efectos y canales; y la tercera para activar o desactivar clips de audio o

MIDI en Ableton Live. Una vez creada la entrada en la lista desde allí se podía volver a acceder a los datos del parámetro en caso de necesitar modificarlos.

Estas opciones se eligieron luego de realizar pruebas de experimentación con Ableton Live y TouchDesigner; en las cuales se vio que eran las que permitían mayor nivel de control e interacción entre sonido, visuales y movimiento (anexo B, repositorio audiovisual, entrada Sync Waltz de las Flores).

Los datos provenientes de TDableton no necesitaban ser procesados para su correcto funcionamiento. Sin embargo, para su óptima vinculación con los parámetros dentro de TouchDesigner, en algunas ocasiones era importante mapear los valores de los datos al rango del parámetro que se quería controlar.

La comunicación con TDableton, a diferencia de la comunicación con el Axis Neuron, es bidireccional. Esto con el objetivo de permitir una mayor interacción e interrelación entre las visuales y la música. Es decir, tanto TouchDesigner podía modificar parámetros de Ableton Live, como Ableton Live podía modificar parámetros de TouchDesigner, esto se puede ver en la figura 10.

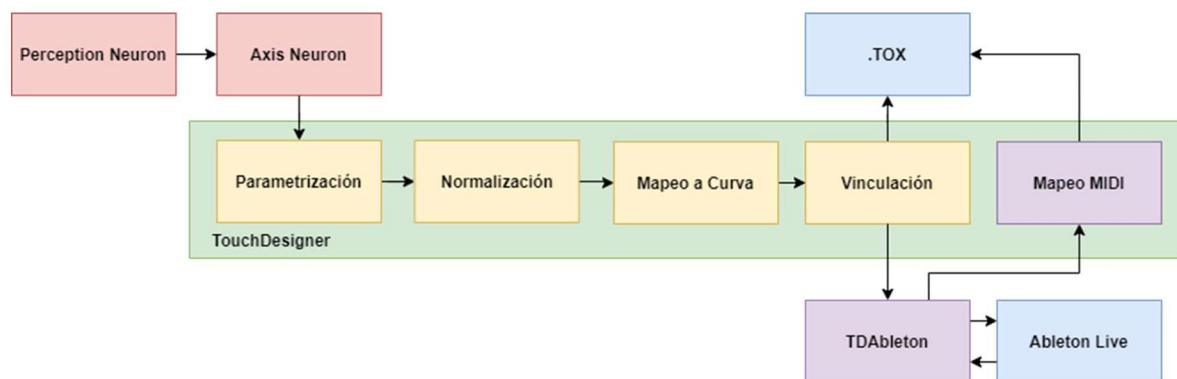


Figura 10. Diagrama de flujo de sistema de mapeo. Dirección de los datos y cómo estos interactúan con los diferentes programas de generación y control de contenido audiovisual.

Al momento de nombrar el sistema final no se encontró ningún otro programa que comuniqué a TouchDesigner con Axis Neuron, por lo cual se visionó el entregable del proyecto como esa API y publicarla como tal, por tanto, TDAxis es una sigla para TouchDesigner - Axis Neuron.

INTERFAZ GRÁFICA Y MÓDULOS

El diseño de la interfaz comenzó con la definición de las funciones más importantes que se consideró que un usuario debía tener. Esto se hizo en base al flujo de trabajo que previamente se definió: crear parámetros, configurarlos y mapearlos a un parámetro de TouchDesigner o Ableton Live. Estos requerimientos llevaban consigo otras necesidades. Por ejemplo, para poder configurar un parámetro de manera apropiada es necesario monitorear su efecto sobre la salida audiovisual. Para esto, se tomó como inspiración la interfaz de Ableton Live, la cual está dividida en diferentes módulos o secciones, cada uno con una función específica. En la figura 11 se pueden ver las interfaces de Ableton Live y TDAxis lado a lado.

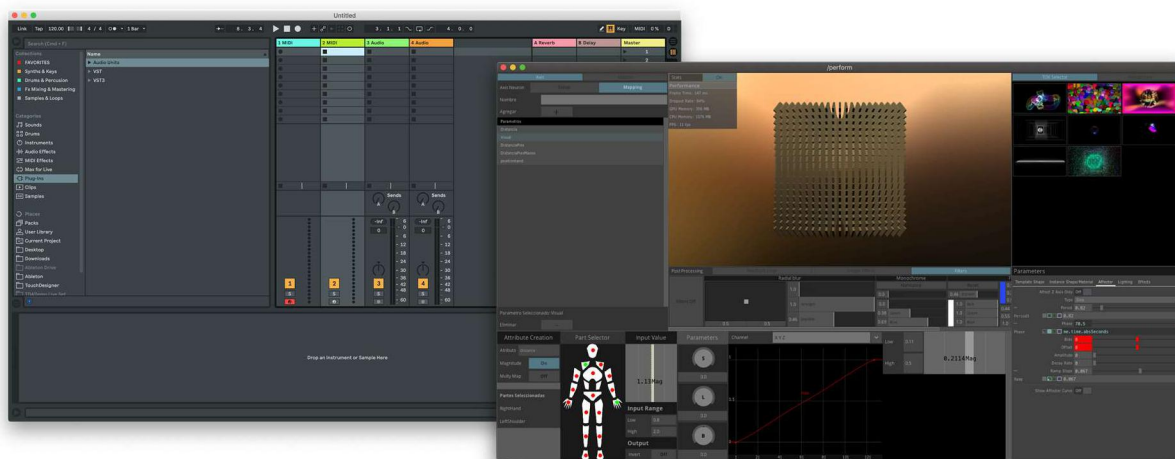


Figura 11. Comparación interfaz Ableton Live e interfaz TDAxis.

Una sección en la cual se pueden editar los parámetros, otra en la cual se pueden visualizar los archivos MIDI o de audio y sus niveles, paneos y envíos, y otra desde la cual se pueden seleccionar los efectos e instrumentos que se van a utilizar. Estos fueron los elementos que más influenciaron la interfaz del TDAxis, la cual tiene una sección dedicada para la creación y selección de todos los parámetros, otra para editar los valores de estos, otra para elegir que se quiere visualizar en la salida, otra para pre visualizar la salida, una sección de efectos de procesamiento, y otra en la cual se pueden editar los parámetros del .TOX que se esté visualizando en el momento.

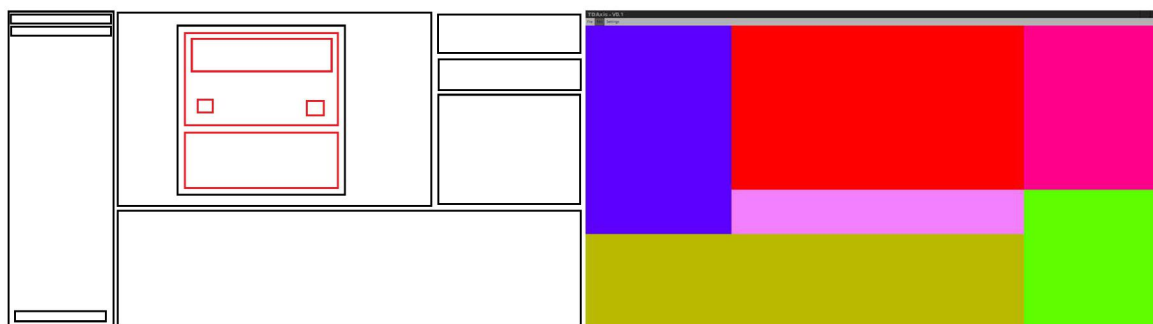


Figura 12. Layout de la interfaz. De izquierda a derecha iteraciones de la disposición de las ventanas o módulos de la interfaz final de TDAxis. Primera realizada en Microsoft Paint, y la segunda desarrollada en TouchDesigner. En la imagen de la derecha: azul es el panel de creación y selección de parámetros, y conexión con Axis Neuron y Ableton Live; rojo es el pre visualizador de la salida visual; rosado es el panel de post procesamiento; fucsia es el selector de .TOX cargados, y para recibir, enviar y controlar los parámetros de TDableton; ocre es el panel para el mapeo de parámetros de Axis Neuron; y finalmente, verde es el panel en el que se muestran los parámetros de salida del .TOX seleccionado.

DESARROLLO DEL SISTEMA

GESTIÓN DEL PROYECTO

Una vez definidos los requisitos con los que debía cumplir el software final para cumplir con los objetivos del trabajo, lo primero que se hizo fue crear un proyecto en Notion en el cual se registró y documentó todo el proceso de creación y desarrollo (anexo B, página principal). Se creó un calendario que sirvió como cronograma, se establecieron los entregables, se crearon bases de datos y relaciones entre ellas para vincular el contenido de las diferentes bases de datos. En el calendario y base de datos principal se registró el trabajo diario durante las etapas de exploración, pruebas y desarrollo. Esto cumplió el papel de diario de campo a lo largo del desarrollo de TDAxis.

Se crearon varias bases de datos para el registro audiovisual; los datos de las personas que sirvieron de usuarios de prueba; todos los .TOX desarrollados; videos, documentos, libros y blogs que sirvieron como fuente de información durante el desarrollo; los problemas encontrados y cómo fueron resueltos; los entregables de cada fase del proyecto; todos los FBX de las pruebas realizadas con el Perception Neuron y todas las piezas de software o hardware utilizadas durante el desarrollo del proyecto.

Sin embargo, aunque Notion es un servicio de almacenamiento en nube y permite compartir archivos, se encontró que, para el trabajo colaborativo, y en especial para compartir los proyectos de TouchDesigner y las sesiones de Ableton Live, no era la herramienta ideal, ya que implicaba estar cargando y descargando los archivos de la nube. Es por esto que en conjunto con el proyecto de Notion se creó una carpeta compartida en Google Drive, con la cual se sincronizan los computadores de trabajo. En esta carpeta se guardaban todos los archivos en los cuales se

estuviera trabajando, permitiendo el acceso a ellos a todos los miembros del equipo. La estructura de carpetas del Google Drive se puede ver en la figura 13.

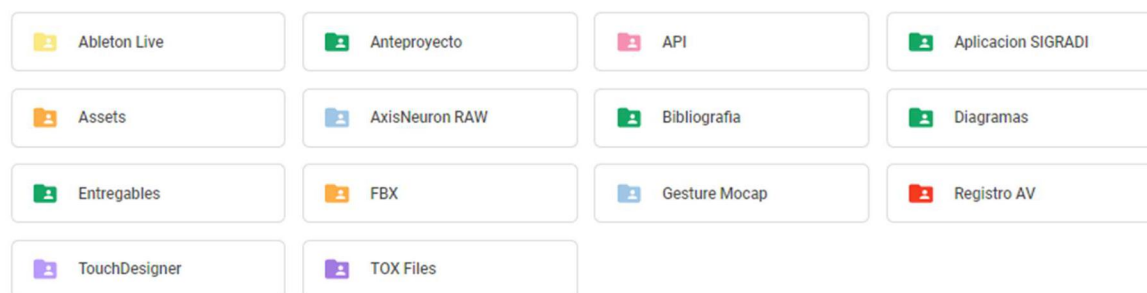


Figura 13. Estructura de carpetas en Google Drive

Una vez un .TOX se consideraba finalizado, se guardaba en una carpeta con todos sus assets, se comprimía y se subía a Notion. Esto servía, además, como un almacenamiento de respaldo de todo el proyecto.

ARQUITECTURA Y FUNCIONAMIENTO

Con todos los requisitos establecidos, se comenzó el desarrollo del sistema (anexo D, repositorio TD-Components). Para esto, el primer paso fue definir la arquitectura del mismo. En esta se definió un módulo principal por cada agente externo a TouchDesigner con el que se tuviera que comunicar. Es decir: un módulo para recibir, procesar y mapear los datos de Axis Neuron a un .TOX; un módulo para recibir, enviar e interpretar datos de Ableton Live; un módulo para cargar .TOX desde el sistema de archivos del equipo desde el cual se estuviera ejecutando el sistema; y finalmente, un módulo para encargarse del post procesamiento y la salida de video. En la figura 14 se puede ver el diagrama de conexión de estos módulos.

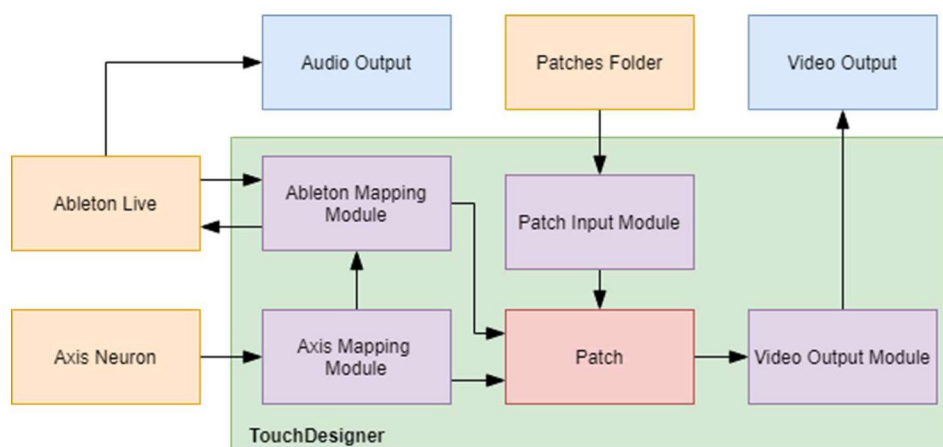


Figura 14. Arquitectura final del TDAxis. Los bloques naranjas representan las aplicaciones o estructuras externas, los azules las salidas de video y audio, los violetas representan los puntos de conexión y procesamiento de datos dentro del TDAxis, y el bloque rojo es donde se da el mapeo a parámetros con los cuales se controla y genera el contenido audiovisual.

Teniendo la arquitectura general del sistema lista, se procedió a diseñar cada uno de sus módulos.

El primero en desarrollarse fue el *Axis Mapping Module* (figura 15), ya que es el módulo que permite la interpretación de los movimientos del cuerpo dentro del sistema. Su función es conectarse con Axis Neuron, recibir sus datos, y crear parámetros para utilizarlos como elementos de control y generación de contenido multimedia. Este módulo posee una entrada por la cual se conecta a Axis Neuron, y entre una y tres salidas para cada parámetro creado. A nivel de interfaz, este módulo está compuesto por 3 elementos, un formulario para configurar los parámetros de conexión (*AXN Setup View*), un panel para crear parámetros y seleccionarlos (*AXN CalcParManager View*), y un panel para parametrizar datos del Axis Neuron y vincularlos a los parámetros de un .TOX o a TD Ableton (*AXN CalcParMapper View*). Estos paneles se pueden observar en la figura 16.

Internamente, este módulo contiene toda la infraestructura necesaria para procesar los datos recibidos, parametrizarlos, normalizarlos, mapearlos a una curva de expresión, escalarlos y vincularlos a otro parámetro.

Cada que el usuario crea un parámetro, el *Axis Mapping Module* crea una instancia del *AXN CalcParMapper*, que se encarga del procesamiento y vinculación del dato.

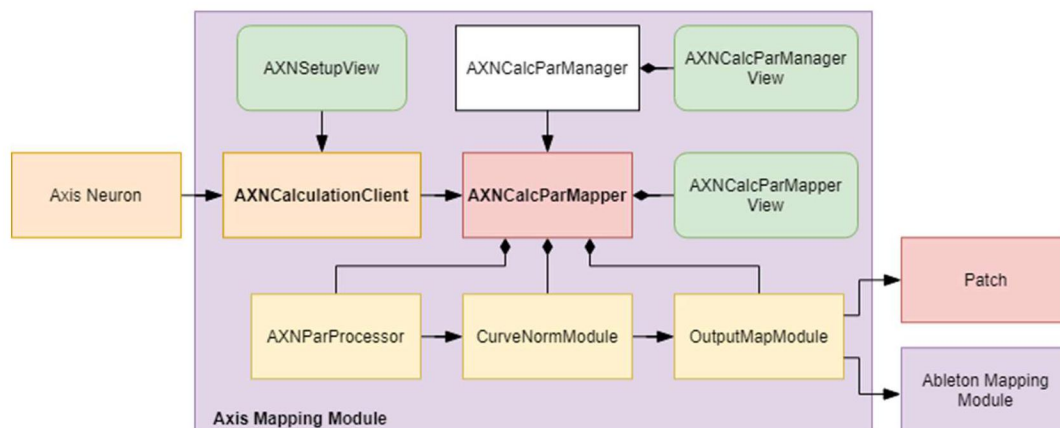


Figura 15. Arquitectura del *Axis Mapping Module*. En verde se pueden ver los bloques que hacen parte de la interfaz gráfica de TDAxis, en amarillo son los que hacen el procesamiento de los datos, en rojo son los centros de conexiones, en blanco el administrador de parámetros y naranja el software externos y puntos de conexión.

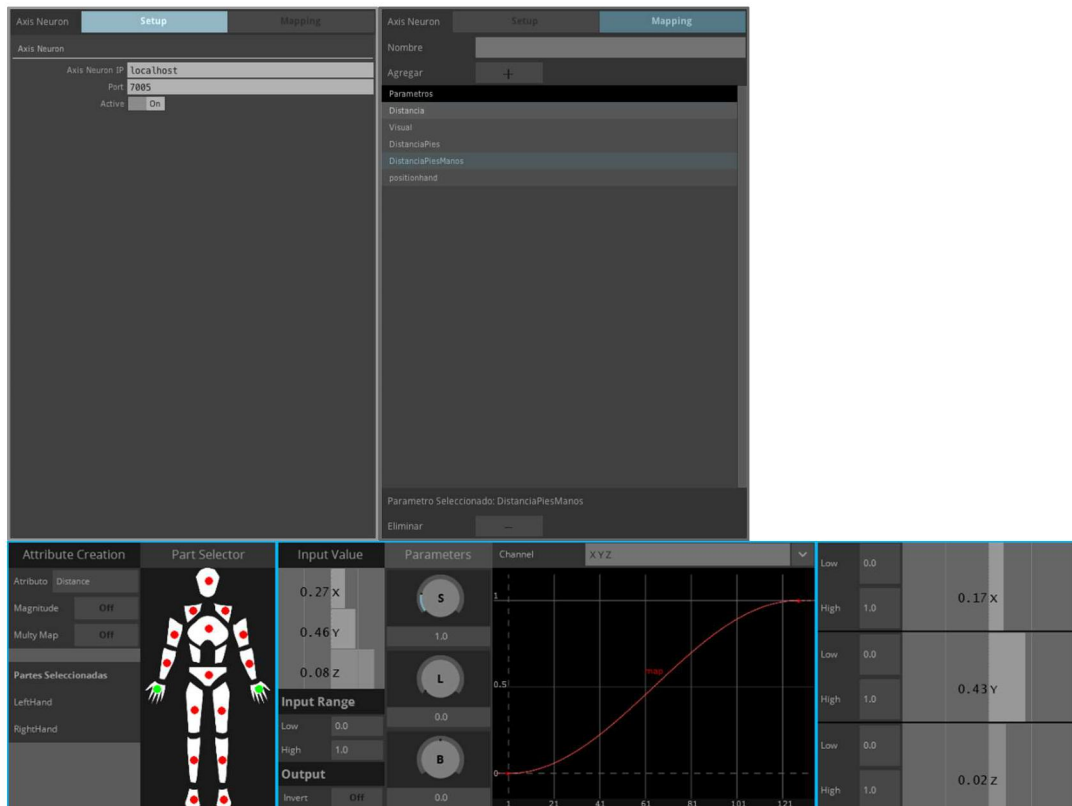


Figura 16. Paneles de la interfaz gráfica del Axis Mapping Module. De izquierda a derecha de arriba hacia abajo: AXNSetup View, AXNCalcParManager View y AXNCalcParMapper View. estando este último dividido en tres (3) secciones. De izquierda a derecha: parametrización, normalización del rango y mapeo a una curva de expresión, mapeo a rango final y salida de los datos.

Una vez desarrollado el *Axis Mapping Module*, se procedió crear un modelo que permite exponer archivos .TOX a otros módulos y componentes de TDAxis y un módulo encargado de cargarlos a TouchDesigner. Al modelo se le llamó *Patch* (figura 17) y al módulo se le llamó *Patch Input Module*. El *Patch* es un modelo en el que se exponen y se visualizan en la interfaz de TDAxis los parámetros internos de un .TOX seleccionado por el usuario.

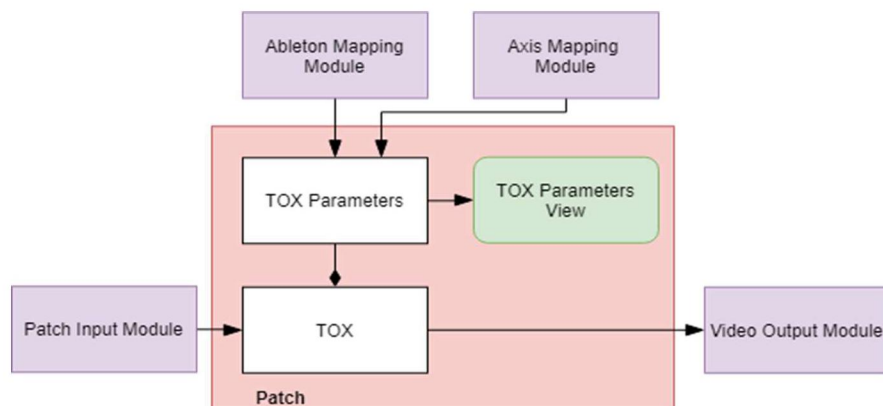


Figura 17. Modelo del Patch. Se exponen los Parámetros del TOX mediante una ventana de parámetros (TOX Parameter View). Estos pueden ser modificados por los demás módulos del sistema. El TOX entrega la salida de video al módulo correspondiente.

Por su parte, el *Patch Input Module* (figura 18) se encarga de acceder al sistema de archivos del computador, cargar de manera dinámica a TouchDesigner todos los .TOX que estén en una carpeta predefinida, permitir la selección de uno de estos y exponerlo al resto del sistema como un Patch; permitiendo tener varios .TOX en una misma sesión de TDAxis. La interfaz de esta módulo se puede ver en la figura 19.

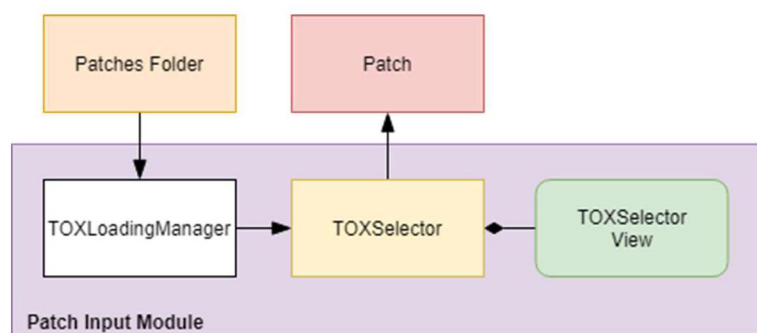


Figura 18. Arquitectura del Patch Input Module

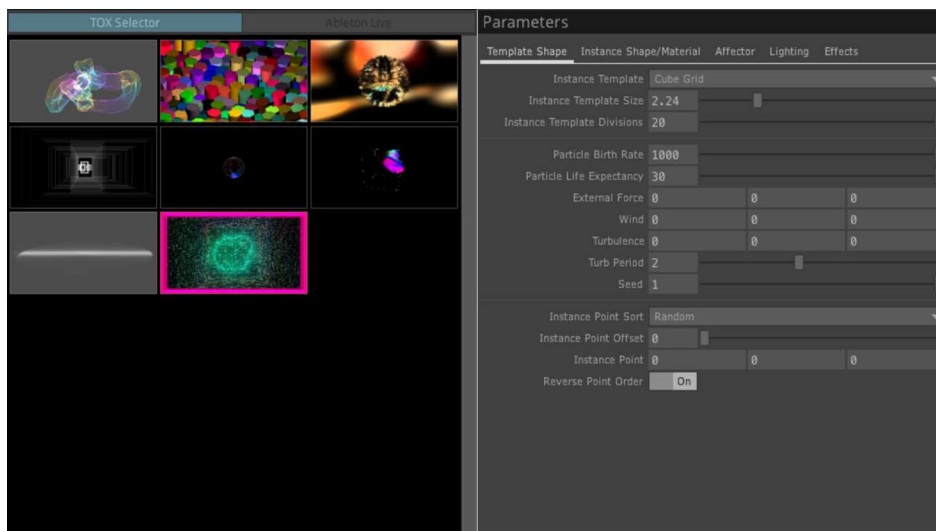


Figura 19. TOX Selector View y TOX Parameter View.

Luego de esto, se desarrolló un módulo encargado de aplicar efectos de post procesamiento a las imágenes generadas y controladas por el sistema, la pre visualización de este contenido y de la salida de video.

Este módulo toma la señal del *Patch* y la pasa por una serie de efectos visuales contenidos en el *Post Processing Rack* (figuras 20, 21 y 22). Estos efectos están divididos en tres (3) categorías descritas a continuación.

El *Feedback Loop* permite crear un efecto de superposición de la misma imagen con diferentes modos de fusión, da la impresión de trazos que se desvanecen.

La categoría Simple Effects permite aplicar efectos de espejos, transformaciones de forma y desenfoque. Finalmente, la categoría Filters es una serie de filtros como desenfoques radiales o cambios de hue. Estos están basados en los filtros que ofrece TouchDesigner en su paleta.

Una vez la señal ha pasado por todos los efectos se divide en 2, la primera es la salida principal de video que puede ser un proyector o una pantalla, la segunda salida va a la ventana de pre visualización, que permite monitorear la salida desde la interfaz de TDAxis.

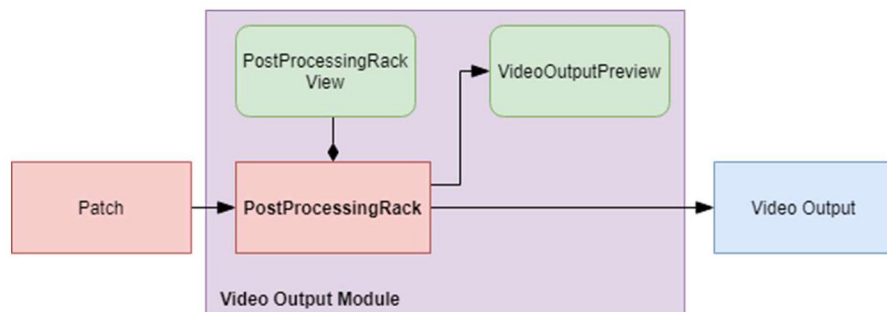


Figura 20. Arquitectura del Video Output Module



Figura 21. Post Processing Rack View en el TDAxis. Los efectos disponibles dentro del rack tienen un botón de encendido y control de sus parámetros.

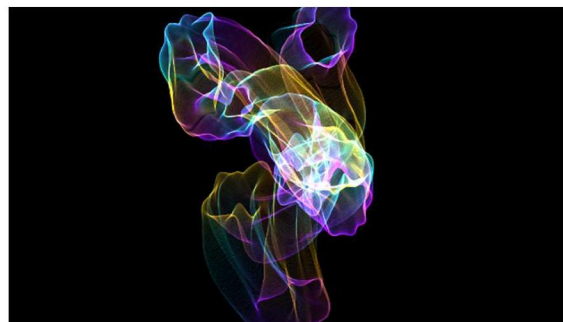


Figura 22. Resultados Post Processing Rack. De izquierda a derecha, arte generativo realizado en TouchDesigner, misma pieza luego de pasar por el Post Processing Rack con los efectos de mirror, blur, y feedback activados.

Finalmente, se desarrolló el módulo que comunica Ableton Live y TDAxis haciendo uso del paquete TDableton (figura 23). La arquitectura de este módulo está basada en la del *Axis Mapping Module*. A nivel de interfaz está compuesto por: una ventana desde la cual se configuran todos los parámetros para la conexión con Ableton Live (*ALSetup View*); un panel de creación y selección de parámetros, que en este caso está dividido en tres (3) secciones, una por

cada tipo de dato que se puede manipular, es decir, una sección para MIDI, otra para parámetros y otra para clips (*ALParManager View*); y un panel de edición y vinculación de los parámetros del dato seleccionado (*ALParMapper View*). Internamente este módulo crea una instancia de los nodos de comunicación de TDableton (*ALClipModule*, *ALParModule*, *ALMIDIModule*) por cada parámetro creado.

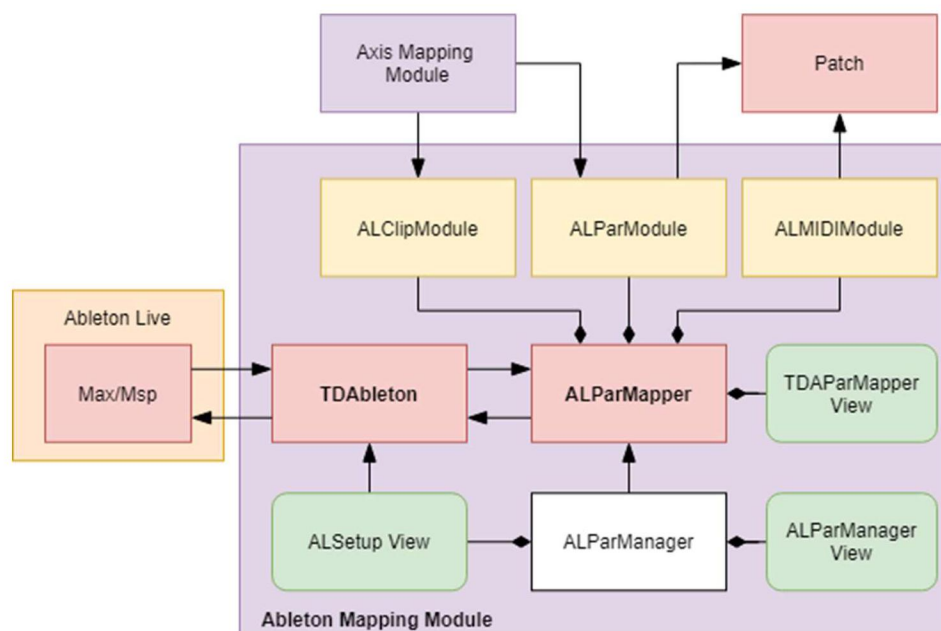


Figura 23. Arquitectura Ableton Mapping Module. En verde se pueden ver los bloques de la interfaz gráfica de TDAxis; en amarillo los Módulos de comunicación del TDableton, en rojo los centros de conexión y en blanco el administrador de parámetros.

CAPÍTULO 3

RESULTADOS

Todo el material audiovisual mencionado en este trabajo se puede encontrar en el anexo A en la sección “Repositorio Audiovisual” o en el anexo C en la sección de videos. Adicionalmente, los releases de TDAxis se pueden encontrar en el anexo D, sección de releases TDAxis. Finalmente, un demo con el resultado final del proyecto y su debida explicación se puede ver en la página principal de TDAxis (anexo A) o en la página principal de Notion (anexo B).

En las simulaciones realizadas (figura 24), se observó un gran potencial performático, permitiendo una clara relación entre los movimientos del cuerpo y el contenido audiovisual que se generaba siempre y cuando se hiciera una apropiada planeación de las relaciones entre el cuerpo y los sonidos, imágenes y efectos generados y controlados por el sistema.

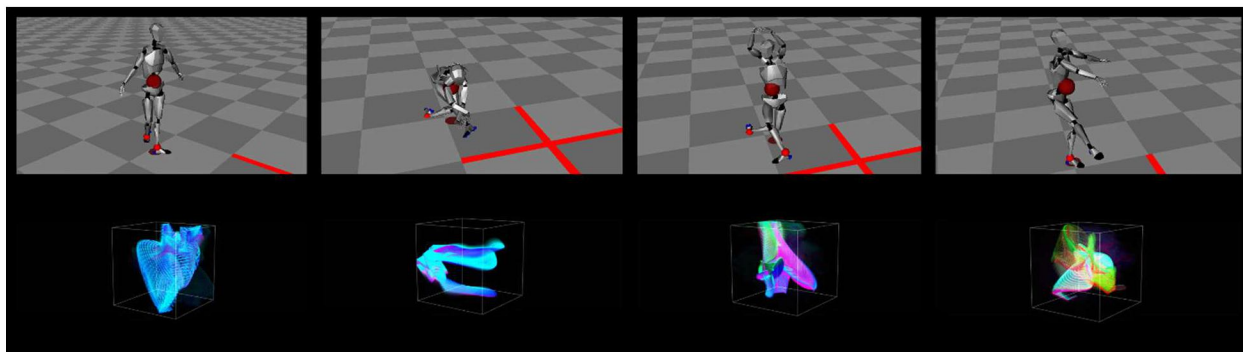


Figura 24. Generación visual. La densidad de la malla se mapeó a la distancia entre los pies. El ruido de la malla se mapeó a la distancia entre los manos. Y los cambios de color se mapearon a la distancia entre el pie izquierda y la mano derecha.

A nivel de la interfaz, si bien no se realizó un proceso iterativo para pulirla, el resultado fue altamente satisfactorio, ya que cumplió los requisitos del flujo de trabajo en torno al cual esta se diseñó.

Sin embargo, dada la pandemia por el COVID-19 que comenzó en marzo de 2020, y que al momento de escribir este trabajo de grado sigue vigente, afectando la fase final del proyecto e imposibilitando la realización de las pruebas con usuarios; no se puede decir que los resultados de este trabajo son concluyentes, por falta de información para contrastar con los resultados obtenidos durante la realización del mismo.

TDAxis presentó pocos problemas a nivel técnico. No presentó fallos ni conexiones intermitentes durante las simulaciones que se realizaron. Esto, siempre y cuando se ejecutara en un computador con tarjeta gráfica dedicada y que las conexiones por red se hicieran a través de un router dedicado.

La parte más inestable del sistema fue TD Ableton, ya que a presentó un funcionamiento impredecible. En algunos momentos funcionaba correctamente y en otros era necesario eliminar y volver a crear el parámetro seleccionado para que volviera a funcionar. Al intentar resolver este problema con ayuda de la documentación de TouchDesigner, se observó que este tipo de errores no son solucionables por los usuarios. Esto se debe a que la comunicación entre TouchDesigner y Ableton Live está programada en bajo nivel y la documentación del software recomienda no editar de manera interna los componentes de TD Ableton. Sin embargo, se cree que, en este caso particular, los errores no son causados por TD Ableton en sí, sino por la manera en la que se integró en TDAxis.

Finalmente, se envió una aplicación al SIGRADI 2020 con el propósito de hacer una ponencia sobre el trabajo de investigación que se realizó.

CAPÍTULO 4

CONCLUSIONES

Con respecto a las tecnologías elegidas, TouchDesigner demostró ser una herramienta ideal a lo que estabilidad de las conexiones por red y potencia gráfica en tiempo real se refiere. Aunque la integración con TD Ableton presentó un comportamiento impredecible, al momento de desarrollo este trabajo, Ableton Live sigue siendo software estándar en la industria del performance de música y sonido en vivo. Por lo tanto, se cree que replantear la arquitectura y el diseño de la implementación dentro del sistema es la mejor opción para mejorar significativamente la fiabilidad del TD Axis.

Una vez finalizado el desarrollo del proyecto, se encuentra que la metodología implementada permitió de manera progresiva y constructiva resolver los problemas que se presentaron a lo largo del desarrollo del proyecto, *“lo que se aprendía un día servía para resolver los retos o problemas del siguiente”*, describe a la perfección la sensación al final del trabajo.

A nivel técnico, se considera que el sistema que se implementó para permitir la creación de parámetros, demostró ser funcional y permitir tener un número virtualmente infinito de estos. Sin embargo, presenta un costo computacional muy alto, que se pudo evidenciar en los FPS del programa, ya que la relación entre el número de parámetros y los FPS era inversamente proporcional.

Utilizar el movimiento del cuerpo humano como una interfaz para controlar y generar contenidos audiovisuales en tiempo real, en el momento de realización de este trabajo todavía presenta varios desafíos técnicos y tecnológicos como para ser la única interfaz en un performance. Esto no significa que no se puede utilizar como una entre varias herramientas para el desarrollo de una puesta en escena. Es en la integración con otras técnicas de generación y control de contenido multimedia que está, lo que consideramos, el mayor potencial del *TDAxis*: su capacidad para vincular imágenes y sonidos con movimientos corporales. Cuando estas imágenes y sonidos son a su vez controlados o generados por otros sistemas o personas con una intención más narrativa o lineal se crea la oportunidad de una colaboración artística de creación e improvisación, en la cual el bailarín tiene un papel mucho más activo, se puede ver como sus movimiento se convierten en un pincel, un instrumento o una consola de mezclas extra con la cual intervenir las imágenes y sonidos creados por sus co-autores, generando así una obra única e irrepetible nacida de la colaboración entre técnicas más tradicionales de creación audiovisual y los movimientos del cuerpo de un bailarín.

Es aquí, en la intersección entre las modalidades artísticas y la tecnología, que este proyecto extiende nuestro entendimiento de qué tan lejos puede la ciencia empoderar a los artistas a avanzar en su capacidad expresiva.

CAPÍTULO 5

RECOMENDACIONES

Dadas las circunstancias presentes durante los meses de marzo y abril del 2020 por la pandemia del COVID-19, no fue posible realizar las pruebas finales con usuarios. Esto hace necesaria que en un futuro se realicen y documenten estas pruebas y que se compruebe en un contexto real el verdadero potencial de la herramienta.

Se considera que, para trabajos futuros, es pertinente hacer la inclusión de herramientas de Machine Learning en TDAxis, que permitan a usuarios no solo mapear movimientos corporales a parámetros, sino también el reconocimiento de gestos y acciones definidos por el usuario mismo. Además, se considera que introducir un sistema capaz de normalizar, de manera automática, los datos del cuerpo, simplifica considerablemente el flujo de trabajo que se propuso en este trabajo.

Además de esto, dados los resultados obtenidos con el TDableton, se recomienda replantear la arquitectura de integración, y su implementación en la interfaz gráfica del TDAxis ya que fueron estos aspectos los que presentaron el comportamiento más impredecible del sistema.

También se propone el desarrollo de más .TOX y programas en Max/Msp que le permitan a los usuarios sin experiencia en programación tener más opciones para la generación y control de contenido audiovisual.

BIBLIOGRAFIA

- [1] “TouchDesigner | Derivative.” <https://derivative.ca/product> (accessed Nov. 24, 2019).
- [2] “Ableton Live | Ableton.” <https://www.ableton.com/en/live/what-is-live/> (accessed Nov. 24, 2019).
- [3] R. Sers, S. Forrester, E. Moss, S. Ward, J. Ma, and M. Zecca, “Validity of the Perception Neuron inertial motion capture system for upper body motion analysis,” *Measurement*, vol. 149, p. 107024, Jan. 2020, doi: 10.1016/j.measurement.2019.107024.
- [4] M. A. Boden and E. A. Edmonds, “What is generative art?,” *Digit. Creat.*, vol. 20, no. 1–2, pp. 21–46, 2009, doi: 10.1080/14626260902867915.
- [5] M. H. Weik, *Computer science and communications dictionary*. 2000.
- [6] P. A. Soga, “Automatic Synthesizing System of Choreography for Supporting Contemporary Dance Creation.”
- [7] Comp, “Congreso Tendencias Escénicas,” vol. 24, 2015.
- [8] M. M. Mainsbridge, “Body as instrument : an exploration of gestural interface design,” 2016, [Online]. Available: <https://opus.lib.uts.edu.au/handle/10453/43474>.
- [9] N. Ltd., “AXIS NEURON User Guide,” 2016.
- [10] “Biovision BVH.” <https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html> (accessed May 07, 2020).
- [11] K. Tuuri, J. Parviainen, and A. Pirhonen, “Who Controls Who? Embodied Control Within Human–Technology Choreographies†,” *Interact. Comput.*, vol. 29, no. 4, pp. 494–511, Jan. 2017, doi: 10.1093/iwc/iww040.
- [12] K. C. Byas, “Embodied Interaction: Perception and Consciousness in Interactive 3-D

Audio-Visual Installations,” 2017, [Online]. Available:

<https://www.academia.edu/39105136>.

- [13] “Using Control Surfaces – Ableton.” <https://help.ableton.com/hc/en-us/articles/209774285-Using-Control-Surfaces> (accessed May 07, 2020).

ANEXOS

ANEXO A: página TDAxis

<https://tdaxis.github.io/>

ANEXO B: Notion

<https://tdaxis.github.io/notion.html>

ANEXO C: videos

<https://tdaxis.github.io/youtube.html>

ANEXO D: repositorios de GitHub

<https://tdaxis.github.io/github.html>