

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS Y COMPUTACIÓN

**“DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA INTEGRACIÓN DE
TECNOLOGÍA DE TRACKING 3D CON TECNOLOGÍA MIDI”**

**DISERTACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS Y COMPUTACIÓN**

AUTOR:

BRYAN MAURICIO ALVARADO CÁRDENAS

DIRECTOR:

ING. ANDRÉS JIMENEZ

QUITO, 2018

DEDICATORIA

A mis padres, ellos son mi principal ejemplo a seguir en todo lo que soy y en todo lo que me he convertido. Va por todo el apoyo que siempre me brindaron y por ser mis guías hacia lo correcto.

A mis profesores, a todos y cada uno porque me enseñaron que para alcanzar algo hay que saber esforzarse y tener dedicación. Va por su ayuda en mi formación profesional y en el presente trabajo.

A mis amigos, que han estado ahí siempre para darme una mano, Va por toda la ayuda que me brindaron en esta etapa.

Todo este trabajo ha sido posible gracias a ellos.

AGRADECIMIENTOS

A la Pontificia Universidad Católica del Ecuador por haberme dado la oportunidad de prepararme profesionalmente y a mi director de tesis: Ingeniero Andrés Jimenez, por su tiempo y su apoyo hacia el desarrollo y puesta en marcha de este trabajo.

A mis padres, hermana, profesores y compañeros de curso, que, a lo largo de toda mi carrera, han ayudado enormemente en el día a día de cualquier forma, con el fin de lograr esta meta.

Tabla de Contenido

DEDICATORIA.....	2
AGRADECIMIENTOS	3
INTRODUCCIÓN	8
ALCANCE	8
OBJETIVOS.....	9
OBJETIVO GENERAL.....	9
OBJETIVOS ESPECÍFICOS.....	9
JUSTIFICACIÓN	11
CAPÍTULO 1: TECNOLOGÍA DE TRACKING 3D	12
1.1. TRACKING 3D	12
1.2. HERRAMIENTAS PARA TRACKING 3D.....	13
1.3. LEAP MOTION	16
1.4. INTERCAMBIO DE INFORMACIÓN A TRAVÉS DE LEAP MOTION	17
1.5. TECNOLOGÍAS QUE SE PUEDEN INTEGRAR CON LEAP MOTION.....	20
1.5.1. Google Earth.....	20
1.5.2. TedCas.....	20
1.5.3. Oculus Rift	21
1.5.4. Casco OSVR de Razer	21
1.5.5. Computadoras Portátiles	22
CAPÍTULO 2: TECNOLOGÍA MIDI	23
2.1. CONTROLADORES MIDI	25
2.2. BANCO DE SONIDOS MIDI.....	27

2.3.	INTERCAMBIO DE INFORMACIÓN A TRAVÉS DE LOS DIFERENTES CANALES MIDI...	28
CAPÍTULO 3: DISEÑO Y DESARROLLO		31
3.1.	SELECCIÓN DE TECNOLOGÍAS	31
3.1.1.	Sensor	31
3.1.2.	Lenguaje de Programación	34
3.1.3.	Librerías MIDI	36
3.2.	SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO	38
3.2.1.	Introducción	38
3.2.2.	Tipos de Metodologías	39
3.3.	DOCUMENTACIÓN TÉCNICA	45
3.3.1.	HISTORIAS DE USUARIO	45
3.3.2.	DIAGRAMA DE CAPAS	46
3.3.3.	DIAGRAMAS DE INTERACCIÓN	48
3.3.4.	DOCUMENTACIÓN DE CÓDIGO	49
3.4.	DOCUMENTACIÓN DE USUARIO	52
3.4.1.	REQUERIMIENTOS TÉCNICOS	52
3.4.2.	ACCESO AL SISTEMA	53
3.4.3.	INSTALACIÓN DEL SISTEMA	53
3.4.4.	USO DEL SISTEMA	54
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES		58
4.1.	CONCLUSIONES	58
4.2.	RECOMENDACIONES	59
GLOSARIO DE TÉRMINOS		63
BIBLIOGRAFÍA		65

Índice de Imágenes

Ilustración 1. Kinect de Microsoft (Microsoft, 2018)	14
Ilustración 2. SMART DX (BTS Bioengineering, 2018)	15
Ilustración 3. Trajes de NANSENSE (NANSENSE, 2018)	16
Ilustración 4. Guantes NANSENSE (NANSENSE, 2018)	16
Ilustración 5. Leap Motion (Leap Motion, 2018)	17
Ilustración 6. Distorsión de Barril y Distorsión de Cojín (Belda, 2015)	19
Ilustración 7. Representación cartesiana de la obtención de la posición de los objetos captados por las dos cámaras del Leap Motion (Belda, 2015)	19
Ilustración 8. Transición entre los requisitos iniciales y finales que surgen en la creación de un proyecto de Prototipado (Lawrence, 2002)	42
Ilustración 9. Ciclo de Vida basado en un Prototipo (Lawrence, 2002)	43
Ilustración 10. Ventana del MIDI Controller (Virtual MIDI Piano Keyboard). Captura tomada por: Bryan Alvarado	54
Ilustración 11. Ventana principal del sitio web. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado	55
Ilustración 12. Ventana principal del sitio web. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado	56
Ilustración 13. Ventana de configuración de notas musicales. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado	57

Índice de Tablas

Tabla 1. Comparación entre las herramientas de Tracking 3D: Kinect vs. Leap Motion vs. SMART DX vs. NANSENSE. Elaborado por: Bryan Alvarado.....	33
Tabla 2. Comparación entre los lenguajes de programación compatibles con Leap Motion: JavaScript vs. C#. Elaborado por: Bryan Alvarado	35
Tabla 3. Comparación entre los tipos de metodologías de desarrollo de software: RUP vs. Agile Unified Process vs. Prototipado Rápido vs. XP vs. Scrum Elaborado por: Bryan Alvarado	41

INTRODUCCIÓN

ALCANCE

Este presente proyecto de disertación de grado terminará con la entrega de un documento que contenga toda la información necesaria para la posible integración de la tecnología de Tracking 3D con tecnología MIDI, el diseño y desarrollo de un prototipo funcional que compruebe dicha integración y un manual de usuario del mismo.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar y desarrollar un prototipo funcional que permita la integración de tecnología de tracking 3D con tecnología MIDI para la creación de sonidos.

OBJETIVOS ESPECÍFICOS

- Diseñar y desarrollar una aplicación que soporte integración con Leap Motion y tecnología MIDI.
- Integrar un controlador MIDI con una plataforma de desarrollo de software, de manera que permita intercambiar información recibida a través del movimiento de las extremidades y convertir dicha información en sonidos.
- Implementar un banco de sonidos MIDI que se pueda usar para intercambiarse por la información recibida a través del Tracking 3D.

- Documentar toda la información necesaria para lograr la integración de tecnología MIDI con tecnología de Tracking 3D.

JUSTIFICACIÓN

En la actualidad la innovación y la constante aparición de nuevas tecnologías es una realidad. La tecnología de Tracking 3D es una de estas, y al ser una tecnología inminente existe un sinfín de posibilidades de uso de la misma.

La integración con una tecnología que ya lleva varios años en el mercado y una documentación amplia como MIDI es una oportunidad para crear nuevas alternativas tanto para desarrolladores, cómo para usuarios generales. La idea es diseñar y desarrollar un prototipo de software en donde se pueda mostrar la utilidad y el alcance de la integración de dichas tecnologías.

CAPÍTULO 1: TECNOLOGÍA DE TRACKING 3D

En el presente capítulo se identificará el significado, contexto y variantes de la tecnología de Tracking 3D. Se examinará las diferentes alternativas que existen en el mercado para su desarrollo, así como algunos casos de éxito. Se enfoca el uso de Leap Motion como herramienta para desarrollo, así como también su facilidad de integración con otras tecnologías.

1.1. TRACKING 3D

Con la llegada de nuevas tecnologías y su avance en los últimos años, la inminente evolución de la realidad virtual, realidad aumentada e inclusive el avance en videojuegos, el campo de la tecnología de Tracking 3D ha evolucionado de manera impresionante (Pérez, 2011).

En general, existen varios tipos de tecnologías de captura de movimiento. Las que funcionan a través de la visión del computador (Gesture Recognition y Eye Tracking). Las que funcionan mediante señales de WiFi (Gesture Recognition, Seguridad y Domótica). Y las que funcionan a través de sensores que detectan los gestos corporales de las personas, de sus

músculos, extremidades, entre otros (Alfaro Vives & Velarde Robles, 2015).

El Tracking 3D es una tecnología de detección de movimiento, relacionada con la tendencia de interfaces humano-computadora (IHC) del tipo NUI (Natural User Interface). Al ser el objeto de estudio de las IHC, se encuentra incluida dentro de diversas áreas de conocimiento tales como: la ingeniería informática, la ingeniería de software, la ingeniería de sistemas, la psicología, neurociencia, psicología y el diseño (Alfaro Vives & Velarde Robles, 2015).

En otras palabras, el Tracking 3D básicamente consiste en la interacción entre una persona y una computadora a través de un sensor de movimiento y la persona la controla, sin necesidad de un contacto físico, es decir, permite sincronizar los elementos creados en la computadora con los movimientos originales que realiza el usuario con sus manos

1.2. HERRAMIENTAS PARA TRACKING 3D

La herramienta que es similar al Leap Motion que sirve para tracking 3D es el Kinect de Microsoft, que fue construido para revolucionar la forma en que las personas juegan con sus consolas y/o computadoras y cómo es la experiencia en el entretenimiento. El Kinect incorpora mucho hardware avanzado de sensibilidad. Más

notablemente, contiene un sensor de profundidad, una cámara a color y un matriz de cuatro micrófonos que proporciona todo el cuerpo que captura el movimiento 3D, reconocimiento facial y capacidades de reconocimiento de voz (Zeng, 2012).



Ilustración 1. Kinect de Microsoft (Microsoft, 2018)

El Kinect, no solamente, se dirige hacia los videojuegos, sino que también puede ser aprovechado en muchos otros aspectos, desde la investigación científica y médica, o en la edición y creación de animaciones o música (Marin, 2013).

Otra de las herramientas más utilizadas actualmente es SMART DX, la cual consiste en sistemas optoelectrónicos que realiza análisis de movimiento en diferentes ámbitos, ya sean: clínicos, deportivos o industriales. Estos sistemas utilizan cámaras digitales con potentes iluminadores infrarrojos que garantiza el funcionamiento eficiente y continuo (BTS Bioengineering, 2018).



Ilustración 2. SMART DX (BTS Bioengineering, 2018)

Otras herramientas conocidas son los trajes, guantes y zapatos, con su propio software, de NANSENSE. Los trajes contienen hasta 54 sensores colocados a lo largo de las extremidades superiores e inferiores del cuerpo. Los guantes incorporan soluciones de rastreo para animar el movimiento de las manos en una resolución muy alta. Estos, combinan variedad de sensores que incluyen IMU, presión y sensores flexibles para medir cada parte y forma de la mano y dedos (NANSENSE, 2018).



Ilustración 3. Trajes de NANSENSE (NANSENSE, 2018)



Ilustración 4. Guantes NANSENSE (NANSENSE, 2018)

1.3. LEAP MOTION

Leap Motion es un interfaz de captura de movimiento que se utiliza para la interacción entre computadoras y personas. La idea de creación de este dispositivo fue gracias a David Holz, que encontró problemas con la creación y manipulación de modelos en 3D

utilizando solamente el teclado y el mouse (Alfaro Vives & Velarde Robles, 2015).

El enfoque principal de este dispositivo es la interactividad personal, es decir, que personas logren interactuar (manejar, controlar, entre otros) con su computadora, de una forma más natural y de alguna forma real.



Ilustración 5. Leap Motion (Leap Motion, 2018)

Una de las razones por que se ha escogido este dispositivo para el desarrollo de este proyecto, es para identificar de mejor manera los gestos, obteniendo la mayor cantidad de información posible con respecto al movimiento realizado por las manos del usuario.

1.4. INTERCAMBIO DE INFORMACIÓN A TRAVÉS DE LEAP MOTION

El funcionamiento del Leap Motion se empieza a describir por su zona de cobertura. El dispositivo ilumina dicha zona, la cual está limitada por el ángulo de visión de los sensores y por la corriente máxima que puede entregar la conexión USB, a través de una luz infrarroja emitida mediante sus led's (Belda, 2015).

Cuando un objeto es iluminado, se produce una reflexión de luz que llega al dispositivo e incide sobre las lentes de las dos cámaras. Estas lentes, de tipo biconvexas, concentran los rayos en el sensor de cada cámara; y los datos recogidos por los sensores se almacenan en una matriz (imagen digitalizada) en la memoria del controlador USB, en donde se realizan los ajustes de resolución adecuados mediante el microcontrolador del dispositivo (Belda, 2015).

Después, los datos recogidos se envían directamente al driver instalado en la computadora. Estos datos representan un valor de intensidad luminosa por cada píxel de la imagen capturada y se guardan en un búffer (Belda, 2015).

Una vez que las imágenes llegan al driver, son analizadas para identificar manos y dedos a partir de un modelo matemático anatómico. Aquí se calcula la profundidad mediante un algoritmo de distorsión compleja, que es una mezcla entre la distorsión de barril (malla de puntos hacia afuera) y la distorsión cojín (malla de puntos hacia dentro), el cual es mejorada por el dispositivo,

superponiendo la imagen captada por el sensor en dichas mallas (Belda, 2015).

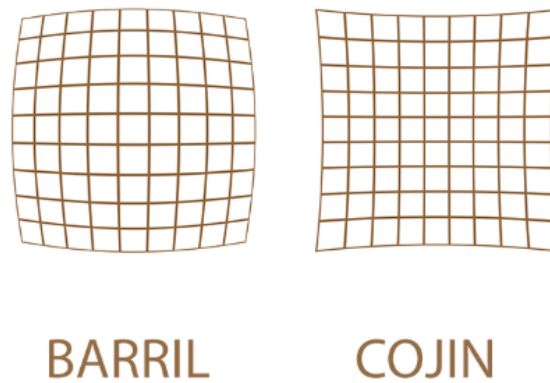


Ilustración 6. Distorsión de Barril y Distorsión de Cojín (Belda, 2015)

Con estas imágenes ya en el driver, se puede determinar la posición de las mismas en el sistema de coordenadas cartesianas del dispositivo a través de técnicas de visión estereoscópica entre cada par de imágenes para obtener la posición en el plano tridimensional (similar a unos binoculares) (Belda, 2015).

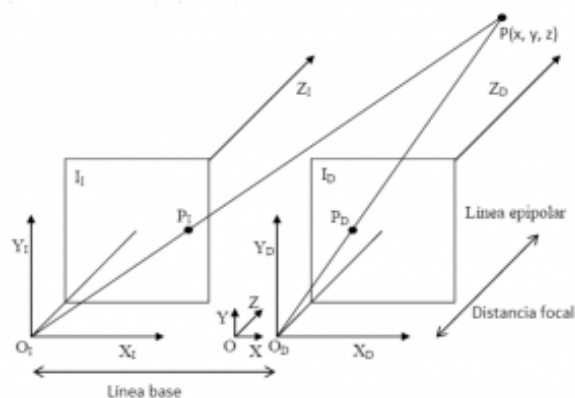


Ilustración 7. Representación cartesiana de la obtención de la posición de los objetos captados por las dos cámaras del Leap Motion (Belda, 2015)

El uso del dispositivo Leap Motion se ha dado con éxito en varias aplicaciones, dicho éxito se debe, en cierta parte, a la precisión que tiene este dispositivo para detectar el movimiento de manos y dedos del usuario.

1.5. TECNOLOGÍAS QUE SE PUEDEN INTEGRAR CON LEAP MOTION

Actualmente, existen varias tecnologías y herramientas que se integran con el Leap Motion, a continuación, se describirá cada una de ellas:

1.5.1. Google Earth

La integración de Leap Motion con Google Earth permite a los usuarios controlar Google Maps mediante gestos, lo que resulta mucho más cómodo y dinámico, además, se mejora la creación de mapas y el visor para calcular distancias. Lastimosamente, esta integración solo viene para la versión Pro de Google Earth (Velasco, 2013).

1.5.2. TedCas

TedCas es una empresa dedicada al sector de la salud. La integración de sus sistemas con Leap Motion permite que los cirujanos tengan un manejo de información más fácil y natural al

momento de realizar cirugías, donde el tiempo y la eficiencia pueden tener un impacto en la salud del paciente.

Los cirujanos pueden hacer zoom en un área de interés de una imagen, rotarla para obtener un nuevo punto de vista o navegar a través de una serie de imágenes (todo mientras mantienen sus guantes estériles para reducir el riesgo de infección y mejorar la eficiencia) (Martínez, 2013).

1.5.3. Oculus Rift

El Oculus Rift es un sistema que entrega a los ojos una visión periférica de realidad virtual. La integración de este sistema con Leap Motion resulta una combinación perfecta y complementaria, ya que el Oculus Rift es capaz de desplegar imágenes de realidad virtual al usuario de manera muy eficaz, y junto con Leap Motion, resulta en una experiencia donde se puede interactuar con los objetos virtuales utilizando el cuerpo (Oyanedel, 2014).

1.5.4. Casco OSVR de Razer

El casco OSVR de Razer es similar al Oculus Rift, solo que funciona con un software open-source. La integración de este dispositivo con Leap Motion se parece mucho a la que se puede tener con lo anteriormente descrito, la diferencia es que aquí se pueden pulsar teclas o controlar una rosca de volumen con el simple movimiento de los dedos sin que el casco lleve ningún accesorio añadido. El trabajo de integración de Leap Motion con este casco va más allá de trabajar solamente con las manos, se pretende utilizar más

partes del cuerpo para obtener una experiencia de realidad virtual más increíble (Engadget, 2015).

1.5.5. Computadoras Portátiles

Las computadoras portátiles no solamente sirven para jugar o para realizar actividades diarias y básicas (escribir un documento, navegar por Internet, entre otros). Leap Motion ofrece una integración casi global para realizar casi cualquier actividad o tarea en el ordenador. Entre estas actividades se tiene: crear objetos 3D, componer música, manejar el puntero del mouse con las manos, cambiar de pestaña en el navegador, jugar videojuegos compatibles con el dispositivo (por ejemplo, Fruit Ninja), entre otras muchas cosas (Tecnología de Tú a Tú, 2014).

CAPÍTULO 2: TECNOLOGÍA MIDI

La tecnología MIDI describe un protocolo, una interfaz digital y conectores, con los cuales varios instrumentos musicales electrónicos, computadoras y también dispositivos diversos, se pueden comunicar o conectarse y compartir información entre ellos. Con este sistema, se puede transmitir mensajes de eventos en donde se especifica una notación musical, un tono y la velocidad (247 Tecno, 2018).

Esta tecnología surge en el año de 1860 cuando Hermann Von Helmholtz, quien fue uno de los primeros en usar la electrónica para generar sonido, construye una serie de generadores electromecánicos que producían tonos puros. Luego, en 1897, Thaddeus Cahill construye el primer sintetizador de sonido gigante, el cual era capaz de producir corrientes alternas a frecuencias distintas. Este sintetizador gigante se llamaba "Telharmonium", el cual sirvió de base para la construcción posterior de sintetizadores relacionados en el ámbito de la electrónica. Mucho tiempo después, en el año 1961, Harald Bode demostró el potencial de esta tecnología de sintetizadores electrónicos, creando el Melochord, considerado como el primer sintetizador controlado por voltaje. En 1964, Robert Moog lanza la primera versión de su propio sintetizador y Donald Buchla presenta su sistema musical electrónico (Moreno, 2013).

Los años 80 fueron en donde aparecieron gran cantidad de instrumentos electrónicos digitales y aparecen samplers, sintetizadores,

secuenciadores, procesadores y controladores, y con el desarrollo de los sintetizadores digitales, hubo la necesidad de implementar un protocolo de comunicación que se convirtiera en un estándar para todos los fabricantes (Moreno, 2013).

Para 1981, varias empresas, en especial Sequential Circuits, desarrollan un protocolo de comunicación que pretendía, no solo comunicar sus dispositivos, sino que otros fabricantes pudieran usarlo y así convertirse en un estándar. Este protocolo, más tarde, se lo denominó USI (Universal Synthesizer Interface), el cual, ofrecía una transmisión serie a 19,2 kilobaudios (la misma que la del RS-232), niveles lógicos TTL (de 0 a +5 Voltios) y jacks estéreo convencionales como conectores. Pero, en 1982, varias compañías japonesas aportaron mejoras a dicho protocolo, haciéndolo más potente y mejor elaborado. Como resultado, surgió lo que hoy se conoce como MIDI (Musical Instrument Digital Interface). Con este protocolo mejorado, denominado MIDI, en diciembre de 1982, la compañía Sequential Circuits lanza al mercado el primer instrumento de la historia dotado con MIDI, el Prophet-600. Más tarde, se incursionó con el Prophet-5, incorporando un secuenciador analógico que permitía programar la reproducción de una frase corta (Moreno, 2013).

MIDI surgió de la necesidad de los diferentes músicos, artistas y compositores de mejorar e incrementar las posibilidades de comunicar sintetizadores y máquinas de ritmos para ejecutar sus creaciones. Con esto, las compañías fabricantes de este tipo de tecnologías respondieron

a esta necesidad y encontraron la manera de conectar, sincronizar y controlar sus dispositivos entre sí (Moreno, 2013).

2.1. CONTROLADORES MIDI

Un controlador MIDI es cualquier hardware o software que genera y transmite datos de la interfaz digital de instrumentos musicales a dispositivos electrónicos o digitales habilitados para MIDI, por lo general para activar sonidos y controlar parámetros de rendimiento de música.

Estos controladores, por lo general, tienen algún tipo de interfaz que el usuario presiona, golpea o toca. Esta acción genera datos (por ejemplo, en qué nota se reprodujo, por cuánto tiempo y con qué ritmo), que luego se pueden transmitir a un módulo de sonido o sintetizador compatible con MIDI mediante un cable también compatible con MIDI. El módulo de sonido o sintetizador a su vez produce un sonido que se amplifica a través de un altavoz (The MIDI Association, 2018).

Existen clases (hardware y software) de controladores tipo MIDI (The MIDI Association, 2018):

- El controlador del teclado es el más común de todos, ya que es un instrumento tradicional rediseñado como un dispositivo de control tipo MIDI.
- Instrumentos musicales electrónicos, incluidos sintetizadores, muestreadores, box de ritmos y baterías electrónicas, que se utilizan para interpretar música en tiempo real y que transmiten datos tipo MIDI.
- Los convertidores Pitch-to-MIDI que incluyen guitarra – sintetizadores analizan un tono y lo convierten en una señal tipo MIDI.
- Instrumentos tradicionales como tambores, pianos acústicos y acordeones que están equipados con sensores y un procesador de computadora que acepta la entrada de los sensores y transmite información de rendimiento en tiempo real como datos tipo MIDI.
- Secuenciadores, que almacenan y recuperan datos tipo MIDI.
- Dispositivos MIDI Machine Control (MMC), como equipos de grabación, que transmiten mensajes para ayudar en la sincronización de dispositivos habilitados para tecnología MIDI.
- Dispositivos MIDI Show Control (MSC), como los controladores de espectáculo, que transmiten mensajes para ayudar en el funcionamiento y la señalización de

producciones teatrales en vivo y de entretenimiento temático.

- La Interfaz Digital de Instrumentos Musicales permite que una variedad de instrumentos musicales interactúe con la PC en un entorno digital para la grabación o el rendimiento.

2.2. BANCO DE SONIDOS MIDI

Un banco de sonidos es básicamente un repositorio de sonidos de instrumentos musicales compatibles con determinada tecnología. En este caso, para la tecnología MIDI, existen varios sitios web en donde ofrecen dispositivos que integran varios sonidos de tipo MIDI. Junto con un programa secuenciador (Cubase, Sonar, Reaper, entre otros) se puede editar a gusto del usuario cada nota de la secuencia del sonido.

El uso de los sonidos tipo MIDI estaba originalmente limitado a aquellos que quisieran hacer uso de instrumentos electrónicos en una producción musical. El estándar permitió que diferentes instrumentos pudieran comunicarse con cualquier dispositivo. Esto causó una rápida expansión en las ventas y en la producción de instrumentos electrónicos, software musical, la llegada de las computadoras personales y los sintetizadores digitales, los cuales

permitían almacenar sonidos preprogramados para posteriormente ser utilizados.

Debido a que MIDI es un conjunto de comandos que crean sonidos, las secuencias MIDI pueden ser manipuladas de distintas maneras en comparación con el audio original. Es posible cambiar la tonalidad, la instrumentación o el tempo, y reacomodar sucesiones de manera individual (The MIDI Association, 2018).

El objetivo principal de este proyecto es conseguir la integración de la tecnología MIDI con la de tracking 3D. Es por ese motivo que para este proyecto se va a utilizar el banco de sonidos MIDI básico que incluye el controlador por defecto.

Para una segunda etapa de este proyecto se podría implementar un banco de sonidos mucho más variado y experimentar con instrumentos musicales mucho más producidos con sonido mucho más realístico.

2.3. INTERCAMBIO DE INFORMACIÓN A TRAVÉS DE LOS DIFERENTES CANALES MIDI

Una conexión MIDI puede llevar 16 canales independientes de información (en código binario del 0 al 15). Un dispositivo puede ser configurado para solo escuchar canales específicos e ignorar los mensajes enviados de otros (modo "Omni Off") o puede escuchar a todos los canales sin importar su dirección (modo "Omni On") (Huber, 2007).

Los mensajes MIDI pueden ser llamados "mensajes de canal", cuando son enviados a unos de los 16 canales y pueden ser escuchados solo por los dispositivos en ese canal, o "mensajes de sistema", cuando pueden ser escuchados por todos los dispositivos. Cualquier dato no relevante para un dispositivo receptor es ignorado. Existen cinco tipos de mensajes: "Channel Voice", "Channel Mode", "System Common", "System Real-Time" y "System Exclusive" (Brewster, 2003).

Los mensajes "Channel Voice" transmiten datos en tiempo real a través de un solo canal. Los mensajes "Channel Mode" incluyen los mensajes Omni/mono/poly mode on y off, así como mensajes que restablecen todos los controladores a su estado inicial o para enviar mensajes "note-off" para todas las notas. Los mensajes "System: Common, Real-Time, Exclusive" no incluyen los números de los canales y son recibidos por cada dispositivo MIDI conectado. El MIDI Time Code es un ejemplo de un mensaje System común.

Los mensajes System Real-Time contienen datos para sincronización e incluyen MIDI Clock y Active Sensing (Hass, 2012).

CAPÍTULO 3: DISEÑO Y DESARROLLO

El presente trabajo se enfocará en el diseño y desarrollo de un prototipo de integración de tecnologías de Tracking 3D y MIDI, el cual permitirá abrir un campo de investigación acerca de dichas tecnologías que aún se encuentran en progreso.

3.1. SELECCIÓN DE TECNOLOGÍAS

En este apartado, se escogerán las tecnologías para poder desarrollar el proyecto. Se hablará de qué sensor, herramienta de desarrollo de software y librerías de tipo MIDI se escogerán.

3.1.1. Sensor

En el capítulo 1 se mencionaban las herramientas de Tracking 3D, entre las cuales se mencionaban: el Kinect de Microsoft, el Leap Motion, el SMART DX y el traje de NANSENSE.

	Kinect	Leap Motion	SMART DX	NANSENSE
Empresa desarrolladora	Microsoft	Leap Motion, Inc.	BTS Bioengineering	Nansense, Inc.

Fecha de lanzamiento	Noviembre, 2010	Julio, 2013	2010	2016
Precio	\$350	\$80	\$580	\$6230
Dimensiones	(33x27x8) cm	(7,5x2,5x1,1) cm	Sin especificar	Sin especificar
Número de cámaras	1	2	1	Hasta 16 cámaras en una estación
Tipo de sensores	Sensor de profundidad (cámara RGB), Micrófono multiarreglo, Ajuste de sensor con su motor de inclinación	Dos sensores monocromáticos tipo CMOS, Iluminación infrarroja	Sensor con resolución 2048x2048 (4Mpixel), Full frame	Pequeños, con una tasa de actualización interna de 1000Hz y midiendo 2000 g / s, el giroscopio triaxial, el magnetómetro y el acelerómetro 16G
Campo de visión	Horizontal: 57°, Vertical: 43°, Inclinación física: 27°, Profundidad del sensor: 1,2 a 3,5 metros	Vertical: 151°, Semiesfera de 61 centímetros de radio	Sin especificar	Sin especificar
Flujo de datos	320 × 240 a 16 bits de profundidad a 30fps; 640 × 480 32-bit de	Hasta 290fps	Hasta 2000fps	Puede funcionar a 60fps, 90fps, 120fps y 240fps

	color a 30fps; Audio de 16-bit a 16kHz			
Sistema de seguimiento o rastreo	6 personas, 20 articulaciones por persona	Hasta dos objetos con precisión de 0.01 mm y ofrece un área de 8 pies cúbicos	Precisión/Volumen: <0,1mm en 6x6x3m	La precisión es excelente cuando se camina, salta, corre, se sienta, se mueven los dedos, entre otros movimientos naturales del ser humano
Sistema de audio	Chat en vivo, Cancelación de eco, Reconocimiento de voz múltiple	No tiene	Circuito TVC	No tiene

Tabla 1. Comparación entre las herramientas de Tracking 3D: Kinect vs. Leap Motion vs. SMART DX vs.

NANSENSE. Elaborado por: Bryan Alvarado

Basado en la tabla comparativa, se escogerá al dispositivo Leap Motion, porque es más barato que el Kinect, que el SMART DX y que el traje de NANSENSE. El Leap Motion es más portable, esto es, que es mucho más pequeño y posee dos cámaras con un tipo de sensores aceptables y adecuados a este proyecto.

3.1.2. Lenguaje de Programación

Una vez decidido la herramienta de Tracking 3D, que en este caso se escogió Leap Motion, ahora se debe escoger un lenguaje de programación que se adapten a la herramienta escogida.

Para este prototipo, se pueden utilizar varios lenguajes de programación, pero existen 2, que realmente son principales: JavaScript y C#. Se debe escoger solamente uno, por lo que se muestra a continuación una tabla comparativa entre estos lenguajes tomando en cuenta lo que se necesita implementar:

	JavaScript	C#
Librería de integración de Leap Motion para sistemas operativos	Una sola librería para todos los sistemas operativos.	libLeapC.dylib (Mac), LeapC.dll (Windows) y libLeapC.so (Linux)
Es interpretado	Si	No

Dependencia de un IDE para óptimo desarrollo	Javascript se puede codificar de manera óptima en cualquier editor de código.	C# depende del IDE de Visual Studio para su manejo óptimo.
Manejo de tipos de datos	Javascript permite utilizar cualquier tipo de dato bajo una variable de un solo tipo <i>var</i> .	C# optimiza el uso de memoria al declarar variables según su tipo de dato.
Implementación óptima según sistema operativo.	Se puede implementar Javascript de forma óptima en cualquier sistema operativo.	Está optimizado únicamente para Sistema operativo Windows.

Tabla 2. Comparación entre los lenguajes de programación compatibles con Leap Motion: JavaScript vs. C#. Elaborado por: Bryan Alvarado

Se busca que el proyecto sea multiplataforma, es decir, que funcione en cualquier computadora independientemente del sistema operativo de la misma. JavaScript es más portable, ya que se utiliza la misma librería en cualquier sistema operativo. Mientras que en C# se debe incluir una librería por cada sistema operativo lo cual complica el proceso: libLeapC.dylib (Mac), LeapC.dll (Windows) y libLeapC.so (Linux).

Entonces, el lenguaje de programación que se utilizará para este proyecto es JavaScript porque solo se requiere de una librería.

Además, basados en el cuadro se sabe que JavaScript es un lenguaje de programación de alto nivel que nos permite desarrollar mucho más rápido que en otros lenguajes gracias a su fácil adaptación con librerías de terceros.

3.1.3. Librerías MIDI

Ahora, como se escogió el lenguaje de programación de JavaScript, existen varias librerías para el manejo de la tecnología MIDI.

Entre las librerías más populares están:

- **MIDI Platinum**, cuyo autor es David Cuello, es una librería de loops y samples, en formato MIDI, WAV 16 bits, STEREO 44.1kHz, cuyo costo es de \$4,95 (Cuello, 2018).
- **Radio Bombs Vol 1. MIDI Edition**, cuyo autor es Refraction Productions, es una librería de electro en formato MIDI, WAV 16 bits, STEREO 44.1kHz, cuyo costo es de \$7,95 (Refraction Productions, 2018).
- **Refraction Essential Chords**, cuyo autor es Refraction Productions, es una librería de loops y

samples con mini chords, en formato MIDI, WAV y STEREO 16 bit, 44.1kHz, cuyo costo es de \$4,75 (Refraction Productions, 2018).

- **Future MIDI and Loops**, cuyo autor es Sugar Sounds, es una librería de sonidos profundos, loops y samples de música House en formato MIDI y WAV estéreo 16 bit, 44.1kHz, cuyo costo es de \$4,95 (Sugar Sounds, 2018).
- **MIDI.js**, cuyo autor es Michael Deal, es una librería realizada en JavaScript, disponible para desarrollo de aplicaciones con tecnología MIDI en web. Puede convertir sonidos para guitarra, bajo, batería, entre otros, en el código que puede leer el navegador. Admite múltiples instrumentos simultáneos y una sincronización perfecta. No tiene costo (Deal, 2015).
- **WebMidi.js**, cuyo autor es Jean-Philippe Coté, es una librería realizada en JavaScript, la cual ayuda a integrar MIDI con web a través de una API. Además, envía y recibe mensajes MIDI con facilidad. Incluye instrumentos de control con funciones fáciles de usar (playNote, sendPitchBend, etc.) y reacciona a la entrada MIDI con simples escuchas de eventos (nota, tono, control, cambio, entre otros). No tiene costo (Coté, 2018).

Entonces, las librerías que se utilizarán para el desarrollo de este proyecto serán: MIDI.js y WebMidi.js, ya que no tienen costo alguno, son fáciles de modificarlas para adaptarlas a lo que se requiera y son hechas en JavaScript.

3.2. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

3.2.1. Introducción

Una metodología de desarrollo de software es aquel proceso que se sigue para diseñar, construir y probar un sistema o programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas (cliente, desarrollador). Dichas metodologías rigen el desarrollo de programas de software en base a modelos de sistemas, reglas, estándares, sugerencias de diseño y guías (OBS, 2018).

Además, al momento de desarrollar un proyecto de software, se utiliza determinada metodología para estructurar, planificar y controlar el proceso de desarrollo, es decir, ayuda a definir: *quién* debe hacer *qué*, *cuándo* y *cómo* debe hacerlo (Enríquez, y otros, 2017).

3.2.2. Tipos de Metodologías

Se puede aplicar cualquier metodología de desarrollo de software para cualquier proyecto que se desee construir, eso sí, se debe de estudiar y tener conocimiento de aquellas metodologías más populares o utilizadas.

Los tipos de metodologías de desarrollo de software que se estudiarán para saber cuál aplicar a este proyecto son: RUP, Agile Unified Process, Prototipado Rápido, XP y Scrum.

Dichas metodologías coinciden en tener algunos aspectos que deben estar presentes durante el proceso de desarrollo de cualquier proyecto, dichos aspectos son: visión del producto, vinculación con el cliente, establecer un modelo de ciclo de vida, gestión de requisitos, plan de desarrollo, integración del proyecto, gestión de cambios, entre otros.

Entonces, se analizarán las cinco metodologías, anteriormente señaladas, en base a una serie de parámetros, en una tabla comparativa:

	RUP	AGILE UNIFIED PROCESS	PROTOTIPADO RÁPIDO	XP	SCRUM
Tipo de metodología	Tradicional	Ágil	Ágil	Ágil	Ágil

Posee documentación adecuada	Sí	Sí	No	Sí	Sí
Adaptable a cambios	No	Sí	Sí	Sí	Sí
Facilita la integración entre las etapas de desarrollo	No	Sí	Sí	Sí	Sí
Relación con UML	Sí	Muy poco	Sí	Muy poco	Muy poco
Permite desarrollo de software sobre cualquier tecnología	Sí	Sí	Sí	Sí	Sí
Tipo de proceso	Muy controlado, estricto	Menos controlado	Menos controlado	Menos controlado	Menos controlado
Permite grupos de desarrolladores	Sí, grupos grandes	Sí, grupos pequeños	Sí, grupos pequeños	Sí, grupos pequeños	Sí, grupos grandes
Número de artefactos o modelos	Muchos	Pocos	Muchos	Pocos	Pocos
Énfasis en la arquitectura del software	Sí	No	Sí	No	No
Presencia en proyectos de	Mucha presencia	Poca presencia	Mediana presencia	Mediana presencia	Mucha presencia

software a nivel empresarial					
---	--	--	--	--	--

Tabla 3. Comparación entre los tipos de metodologías de desarrollo de software: RUP vs. Agile Unified Process vs. Prototipado Rápido vs. XP vs. Scrum Elaborado por: Bryan Alvarado

Con base en la Tabla 3 se utilizará en este proyecto la metodología de Prototipado Rápido, ya que, este proyecto está enfocado al desarrollo de un nuevo producto y no hay un usuario final específico, ni lista de requerimientos previos, el tipo de metodología es ágil, por lo que el tipo de proceso es menos controlado, que permitirá realizar cambios durante el desarrollo, se relaciona con el Lenguaje Unificado de Modelado (UML), facilita la integración entre las etapas de desarrollo y provee gran cantidad de artefactos o modelos, los cuales servirán de base para el desarrollo del proyecto.

La metodología de Prototipado Rápido permite que todo el sistema, o algunas de sus partes, se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que se aseguren que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita, así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo (Lawrence, 2002).

Esta metodología se encarga del desarrollo de diseños para que estos sean analizados y prescindir de ellos a medida que se adhieran nuevas especificaciones, es ideal para medir el alcance del producto, pero no se asegura su uso real (Lawrence, 2002).

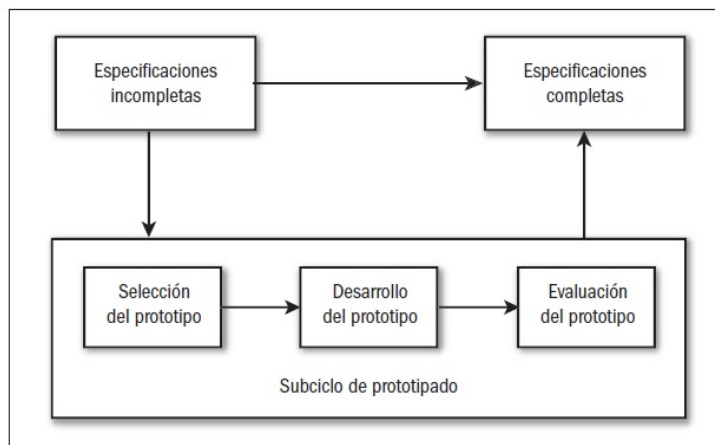


Ilustración 8. Transición entre los requisitos iniciales y finales que surgen en la creación de un proyecto de Prototipado (Lawrence, 2002)

Un factor muy importante que se debe tomar en cuenta en este tipo de metodología es que para que la construcción de o de los prototipos sea posible, se debe contar con la participación del cliente.

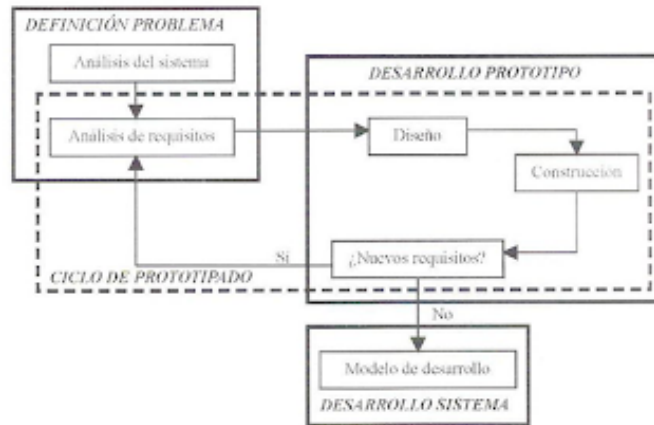


Ilustración 9. Ciclo de Vida basado en un Prototipo (Lawrence, 2002)

Esta metodología tiene la ventaja de ser útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos o especificaciones del software. También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro del funcionamiento o eficiencia de un algoritmo, de la adaptabilidad del sistema o de la forma en que el usuario debería interactuar con el software (Yáñez, 2010).

El cliente ve funcionando lo que para él es la primera versión del prototipo que ha sido construido de forma “beta”, y puede desilusionarse al decirle que el sistema aún no ha sido construido. El desarrollador puede ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente (Yáñez, 2010).

Cualquier desarrollo de software debe tener diagramas de requerimientos, de arquitectura y de interacción. En este caso se usará Historias de Usuario para levantar requerimientos, ya que en este caso existe un solo usuario final con el que se llega a un acuerdo mutuo, se puede agilizar el proceso a través de historias sencillas, priorizando las características del producto y se puede enrumbar todo en una sola dirección. Es una técnica que ayuda a agilizar procesos en el desarrollo ágil.

Para los casos específicos de diagramas de arquitectura y de interacción se irá por el lado de RUP, esto porque es mucho más formal y estructurado que otros métodos. En el caso de arquitectura se utiliza diagrama de paquetes porque es la representación interna de como está estructurado el software, y diagrama de despliegue para tener una representación externa del mismo.

Para interacción, el diagrama de actividades representa el modelo del flujo de trabajo que hay entre los componentes del sistema. También se usa el diagrama de secuencia es el diagrama que representa la relación entre los objetos del sistema.

3.3. DOCUMENTACIÓN TÉCNICA

3.3.1. HISTORIAS DE USUARIO

HISTORIA 1

Necesito configurar las notas que voy a interpretar	Para que exista armonía en la interpretación.	Porque necesito tener control de las notas que quiero escuchar.
---	---	---

HISTORIA 2

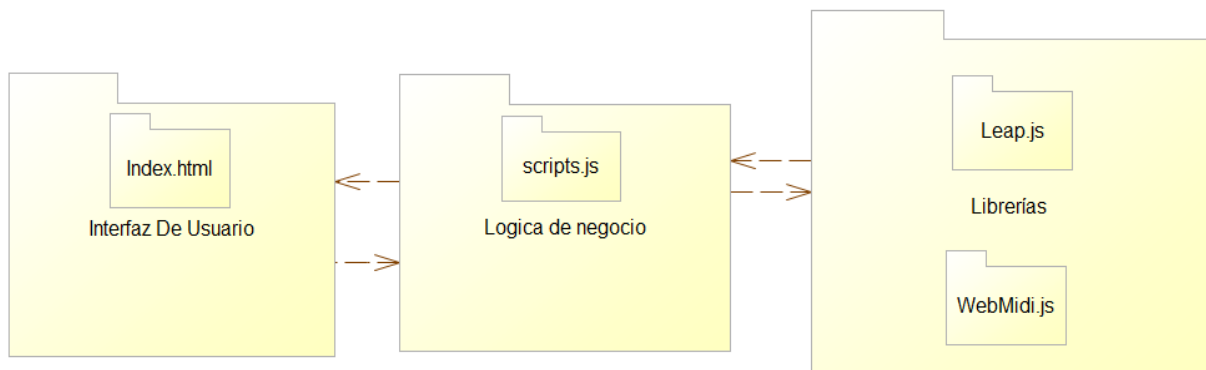
Necesito interpretar un instrumento musical sin tocarlo	Para generar mayor atractivo visual a la hora de la interpretación	Porque al momento no existen soluciones comerciales que permitan hacerlo de forma sencilla y con bajo costo.
---	--	--

CASO DE PRUEBA

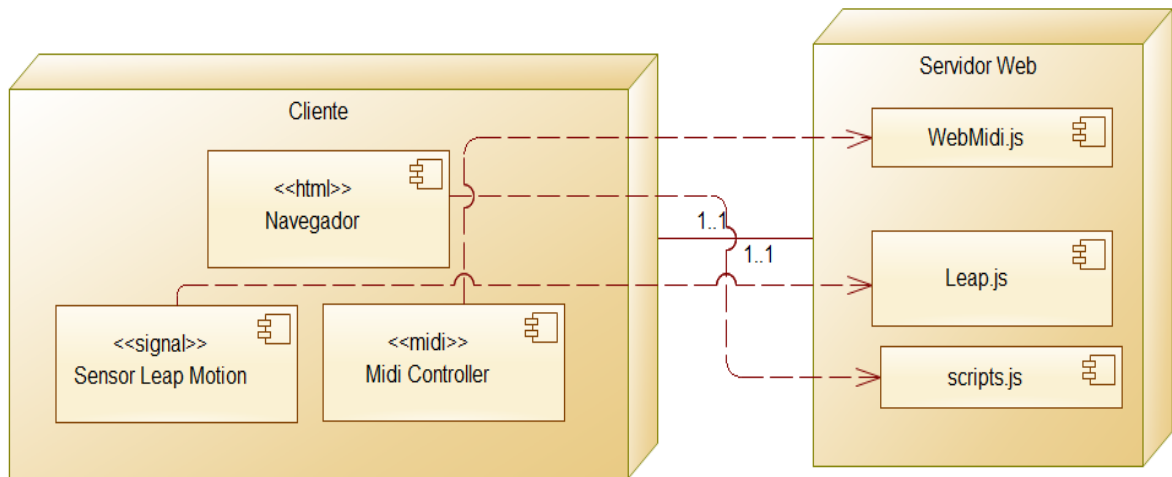
Historia	Acciones	Resultados
Historia 1	Configurar las notas a interpretar	Notas configuradas satisfactoriamente.
Historia 2	Poner la mano sobre el sensor y moverla según la interpretación	Sonidos correctos de acuerdo a la configuración y al movimiento del usuario

3.3.2. DIAGRAMA DE CAPAS

3.3.2.1. DIAGRAMA DE PAQUETES

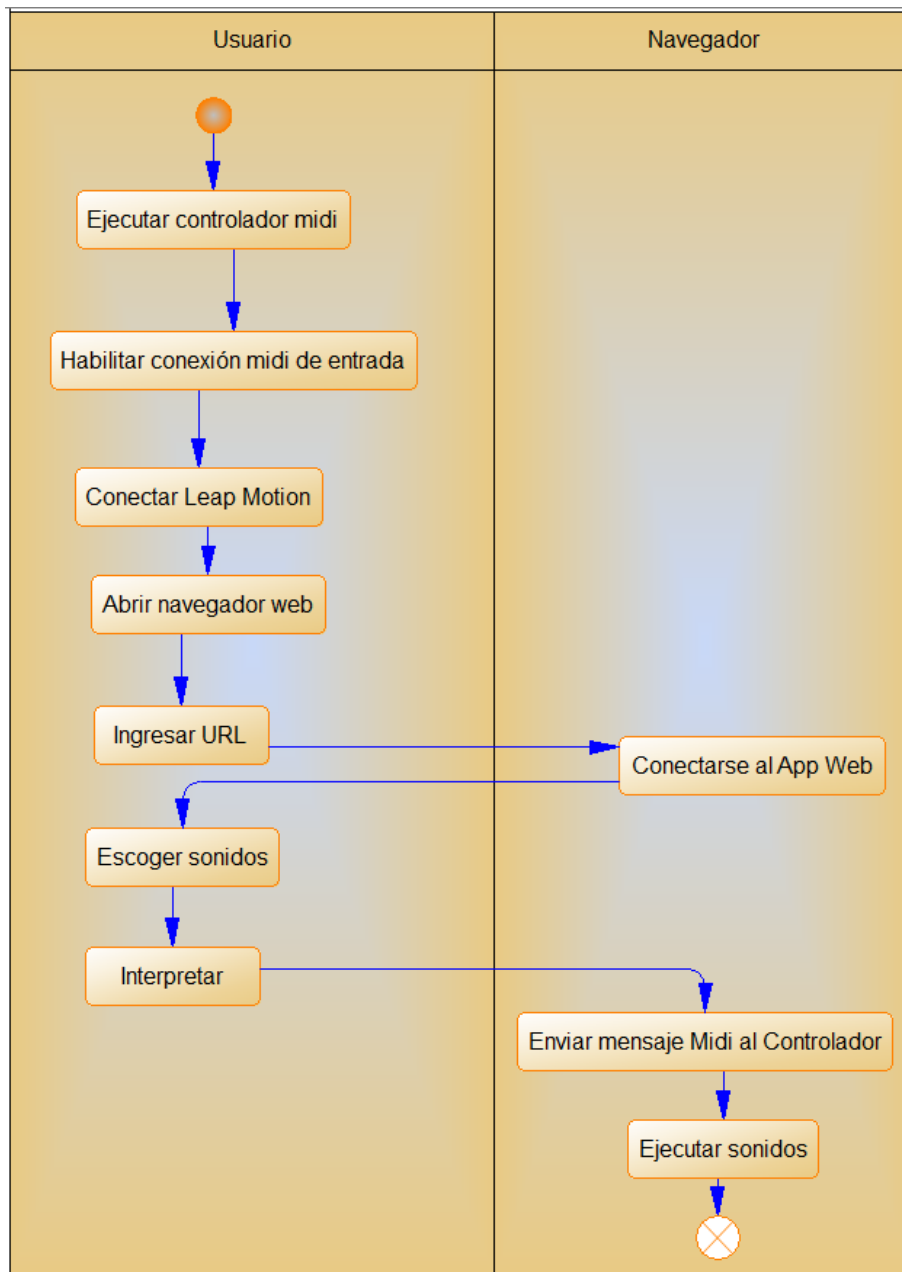


3.3.2.2. DIAGRAMA DE DESPLIEGUE

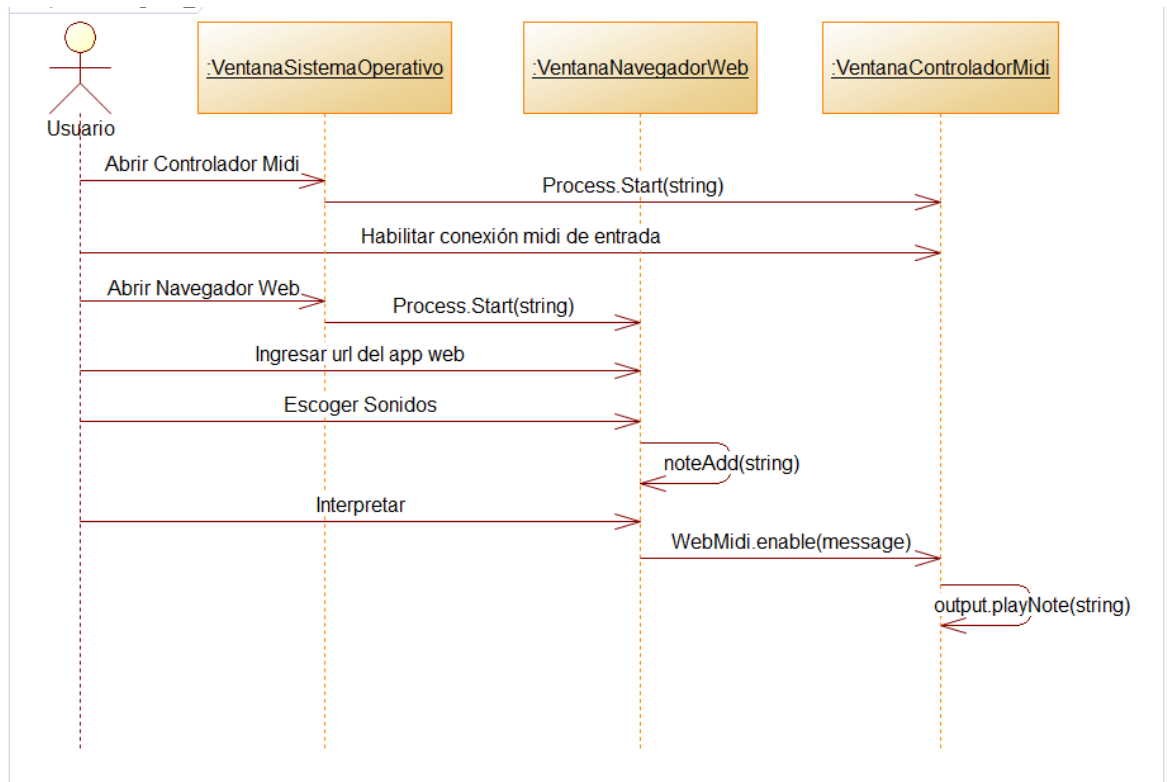


3.3.3. DIAGRAMAS DE INTERACCIÓN

3.3.3.1. DIAGRAMA DE ACTIVIDADES



3.3.3.2. DIAGRAMA DE SECUENCIA



3.3.4. DOCUMENTACIÓN DE CÓDIGO

WebMidi.js

WebMidi.js dispone de métodos fundamentales para el correcto funcionamiento de este prototipo

```
WebMidi.enable(function (err) {
    console.log(WebMidi.inputs);
    console.log(WebMidi.outputs);
});
```

El método `enable()` es donde se puede tener acceso a los controladores MIDI que estén disponibles en ese momento, sean entradas o salidas. Si existe más de un controlador, se crea un arreglo de controladores. Para este caso solamente se tiene uno conectado entonces se puede usar simplemente el primer controlador que encuentre la librería.

Como se va a enviar mensajes hacia el controlador se escoge la propiedad `outputs`, de esta manera se tiene control sobre lo que se va a mandar como mensaje al controlador MIDI.

```
output.playNote("F#1", "all", {velocity: 1});
```

`WebMidi.js` tiene a su disposición un método `playNote()`, que nos permite mandar las notas que se quiere que sean interpretadas, por que canal, y su duración.

Se utilizará esto para enviar el mensaje al controlador y que el mismo pueda reproducir el sonido que se enviará de parámetro.

Leap.js

```
star.setTransform = function(position, rotation)
```

`Leap.js` dispone de una función que se llama `setTransform()`, que da la posición y la rotación de la mano.

Con esto se puede utilizar la posición y rotación para mover el objeto que sirve de referencia dentro del sitio. En este caso una imagen.

```
img.style.transform = 'rotate(' + -rotation + 'rad)';  
img.style.webkitTransform = img.style.MozTransform =  
img.style.msTransform = img.style.OTransform =  
img.style.transform;
```

Basados en lo anterior se puede implementar una función que detecte la posición de la mano y clasificarla en cuatro cuadrantes principales que se reflejarían en la interfaz gráfica.

```
switch (cuadrante){  
  case 1:  
    stopNotes(output);  
    output.playNote(firstNote, 1);  
  break;
```

Dependiendo de eso se mande un mensaje hacia el controlador MIDI con la nota que se desea que se interprete, a través de la función playNote().

Se puede parametrizar las notas que se desee que se interpreten.

En conclusión, la estrella que se crea es una imagen, y a la misma se le agregan escuchadores para que se mueva y rote la misma a través de la señal que envía el Leap Motion detectando el movimiento de la mano.

Se detectan las coordenadas de donde se encuentra la imagen y dependiendo de en que cuadrante se encuentra se emite un mensaje al controlador con las Notas que se escogieron en la opción de configuración.

3.4. DOCUMENTACIÓN DE USUARIO

La documentación del usuario tiene como finalidad proporcionar una guía básica acerca de qué es y de cómo utilizar el prototipo desarrollado.

3.4.1. REQUERIMIENTOS TÉCNICOS

- Navegador de Internet (Chrome, Firefox, Edge, entre otros).
- Dispositivo Leap Motion
- Controlador MIDI
- Cualquier Sistema Operativo (Windows, Mac OS, Linux)

3.4.2. ACCESO AL SISTEMA

El ingreso al sistema se lo hará a través de un navegador web, desde la siguiente dirección: <http://leapmidi.com>. Una vez que entre al aplicativo web, se podrá utilizar el sistema.

3.4.3. INSTALACIÓN DEL SISTEMA

Antes de iniciar con el uso del sistema, se debe dejar en claro que se debe instalar el driver (controlador) del dispositivo Leap Motion, el cual se lo puede encontrar en la siguiente dirección web:
<https://www.leapmotion.com/setup/desktop/>

Y, además, se debe instalar un Controlador MIDI, el que sea de la preferencia del usuario. Se los puede encontrar en la siguiente dirección web:
http://www.hitsquad.com/smm/cat/MIDI_CONTROLLERS/

Finalmente se debe escoger el bando de sonidos de preferencia o se puede descargar un banco de sonidos para el controlador MIDI.

3.4.4. USO DEL SISTEMA

Se abre el MIDI Controller y se configura la entrada MIDI que tendrá ese controlador.

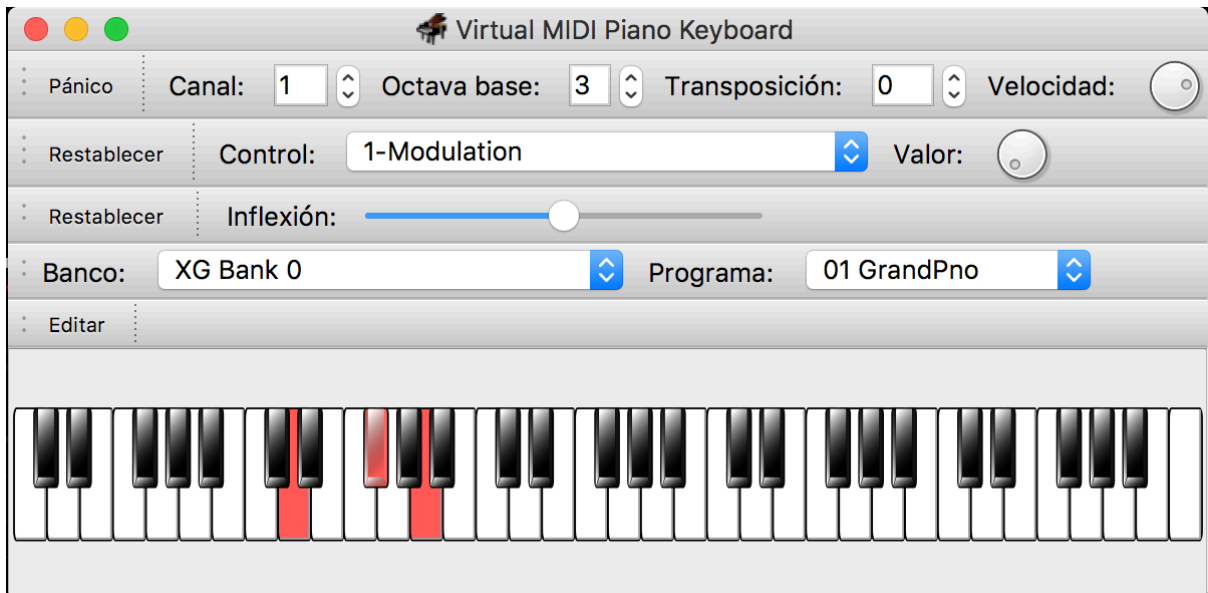


Ilustración 10. Ventana del MIDI Controller (Virtual MIDI Piano Keyboard). Captura tomada por: Bryan Alvarado

En este caso, el MIDI Controller es un piano virtual en formato MIDI, el cual servirá para que en la página principal del aplicativo web no exista ningún problema.

Una vez configurado el MIDI Controller, se ingresa a la dirección web del sistema (leapmidi.com), y se muestra una página principal como esta:

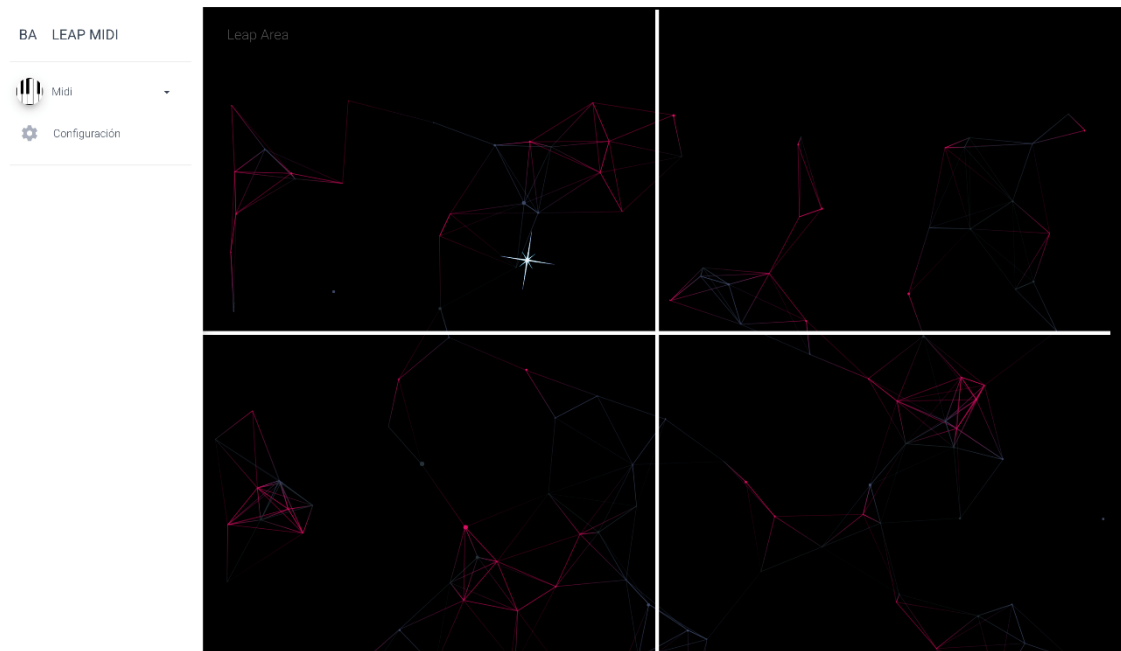


Ilustración 11. Ventana principal del sitio web. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado

En dicha interfaz se presentan 4 cuadrantes, los cuales serán el espacio en el cual, se pasará la mano por el sensor del Leap Motion y se capturará el movimiento que recorrerá dichos cuadrantes.

Estos 4 cuadrantes son configurables con todas las notas musicales (tres octavas), para que el usuario pueda componer diferentes melodías.

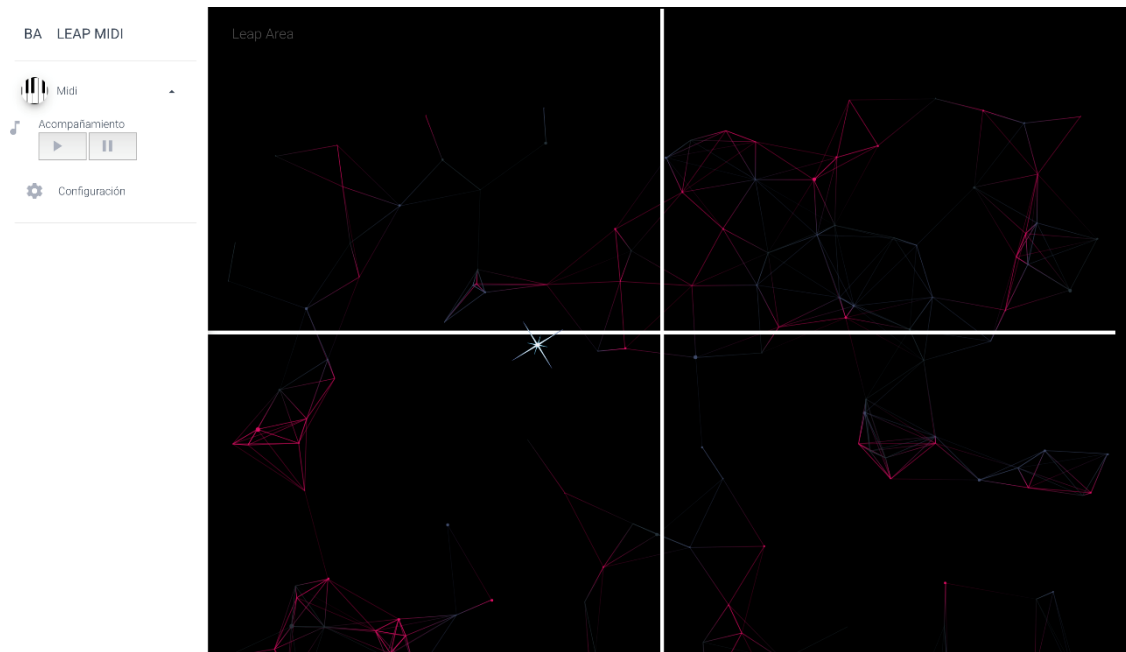


Ilustración 12. Ventana principal del sitio web. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado

En la parte izquierda de la pantalla principal del sistema se muestra una barra lateral en donde se pueden identificar dos botones: uno de Acompañamiento Musical y otro de Configuración.

El botón de Acompañamiento Musical, como su nombre lo dice, sirve para hacer un acompañamiento a la melodía que se está creando con el movimiento de las manos a través del Leap Motion. Esto es para, que se pueda crear y escuchar algo agradable. El acompañamiento se lo implementó con el sonido de una batería.

El botón de Configuración sirve para establecer las notas musicales de cada cuadrante. Cada cuadrante reproduce un sonido diferente, dependiendo de la configuración inicialmente realizada.

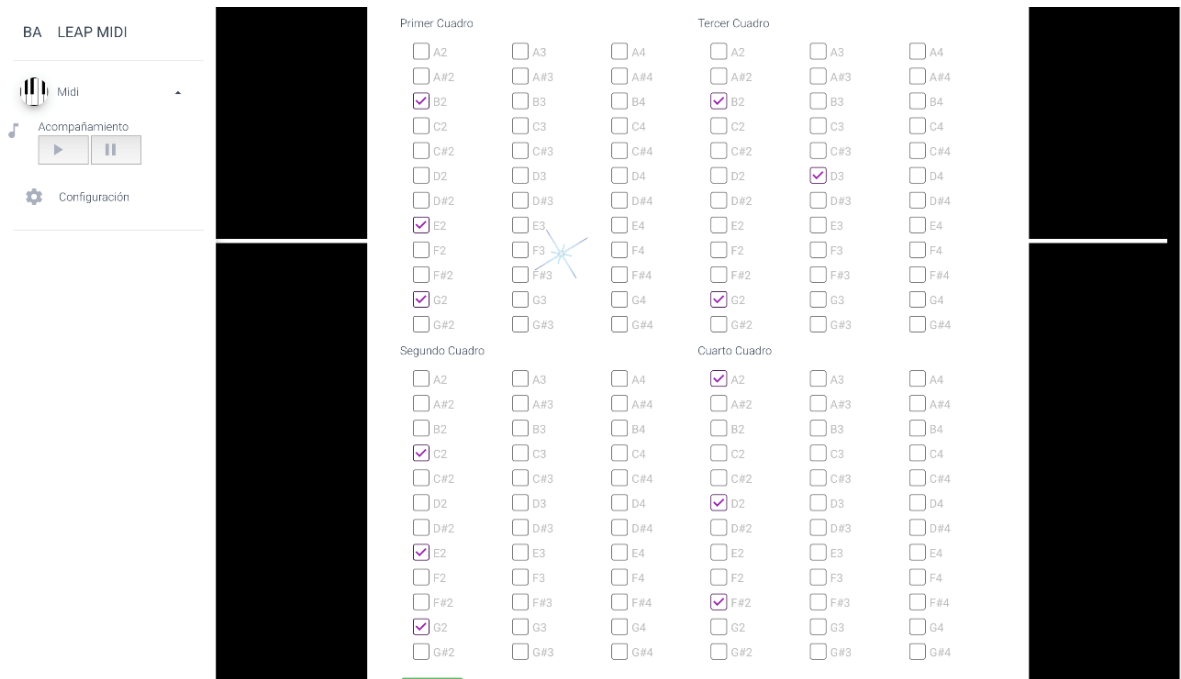


Ilustración 13. Ventana de configuración de notas musicales. Imagen obtenida del proyecto. Captura tomada por: Bryan Alvarado

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Como se explica en el Capítulo 1, al final del punto 1.1, la tecnología de Tracking 3D permite sincronizar los elementos creados en la computadora con los movimientos originales que realiza el usuario con sus manos.
- Se mencionó en el Capítulo 1 que, no solamente se puede utilizar la tecnología de Tracking 3D en ámbitos musicales, también se la puede utilizar en ámbitos mucho más profesionales, por ejemplo: diseños de infraestructura en 3D, estudios médicos, diseño de productos industriales, entre otros.
- Con este proyecto, se logra entrar en un área poco explorada actualmente, la integración de Tracking 3D con la tecnología MIDI, que resulta muy útil al momento de aprovechar sensores para poder realizar actividades que abarcan varios ámbitos de la vida cotidiana.
- Como se explica al final del Capítulo 3, en el punto 3.1.2, JavaScript es un lenguaje de programación de alto nivel que nos permite

desarrollar mucho más rápido que en otros lenguajes gracias a su fácil adaptación con librerías de terceros.

- Como se explica en el Capítulo 3, en el punto 3.1.3, la librería WebMidi.js ayuda a detectar controladores MIDI conectados a la máquina de forma rápida y eficiente, y el intercambio de información no presenta mucha latencia, lo cual la hace la librería óptima disponible para implementar este prototipo.
- Para lograr la conexión satisfactoria del controlador MIDI con este prototipo es necesario habilitar las conexiones de entrada de MIDI y definir el canal por el cual se va a realizar el intercambio de mensajes.
- WebMidi.js reacciona a la entrada MIDI con simples escuchas de eventos y determina a que canal pertenece cada uno, lo cual facilita el proceso de integración del controlador con este prototipo.

4.2. RECOMENDACIONES

- Como se explica en el Capítulo 3, en el punto 3.1.3, si se va a probar funcionamiento de librerías de terceros que se encuentra en Internet es importante tratar del código fuente de la manera más

organizada, de forma que se pueda evitar conflictos con lo que ya se tenga implementado antes.

- En el caso de integración de la tecnología MIDI con un navegador de Internet, es recomendable utilizar navegadores potentes como Chrome, Firefox, Edge o Safari, que son los que mejores herramientas nos brindan a los desarrolladores ya que se maneja una consola en la cual se puede depurar para encontrar errores más rápido y poder mejorar la calidad del producto que se está desarrollando.
- A la hora de escoger el controlador MIDI, primero probar con uno que tenga una integración bastante sencilla, el que se utilizó para este proyecto era bastante limitado en cuanto a sonidos, pero tenía una configuración fácil y se pudo adaptar rápidamente a la librería que se está usando. Una vez hecha la prueba de concepto, se puede usar mejores controladores con mejores bancos de sonidos.
- Si es que se desea realizar una segunda etapa del proyecto se recomienda parametrizar los canales para permitir que se pueda conectar más de un controlador MIDI y darle variaciones interesantes al software.
- Con base en el punto 3.3.4 se recomienda que, al ser el navegador el que se conecta directamente con el controlador, el computador

que se utiliza disponga de buenos recursos para que no se presenten problemas de latencia.

- Si es que se desea realizar una segunda etapa del proyecto se recomienda aprovechar al máximo las capacidades de Leap Motion para que se pueda leer más gestos de los dedos y utilizar dos manos al mismo tiempo, esto daría variaciones a los sonidos que se podrían implementar.
- Las coordenadas que da el SDK de Leap Motion con respecto al movimiento de las manos, no son las más optimizadas para desarrollar dentro de un sitio web. Se recomienda transformar esto con una relación de aspecto al espacio que se desee asignar. Esto ahorrara un tiempo significativo en desarrollo y más si se decide trabajar con varios gestos.
- Con base en el punto 3.3.4, se recomienda que, en caso de tener más de un controlador al mismo tiempo, es importante revisar a que canal están haciendo referencia los mismos para evitar que exista algún problema.
- Para una segunda fase del proyecto se puede también parametrizar los canales para permitir que se pueda conectar más de un controlador MIDI y darle variaciones interesantes al software. Además, se puede agregar control de los mensajes ya existentes y

parametrizar los mismos, agregando controles de volumen y reverberación con los mismos gestos.

GLOSARIO DE TÉRMINOS

- **Framework:** Es una estructura conceptual y tecnológica de soporte definido, normalmente con módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
- **Función o método:** Es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento.
- **Kinect:** Es un dispositivo que es capaz de capturar el esqueleto humano, reconocerlo y posicionarlo en el plano.
- **Leap Motion:** Es un dispositivo pequeño que se conecta, mediante USB, al ordenador, normalmente, entre el teclado y el cuerpo. Este dispositivo recibe y transfiere al ordenador toda la información de los dedos, posición en 3D, partes que lo forman (en dicha posición, cuanto miden, separación entre partes, entre otros).
- **Lenguaje de programación:** Es aquella estructura que, con una cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora.

- **Librería:** Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación y su fin es ser utilizada por otros programas, independientes y de forma simultánea.
- **MIDI:** Es un estándar tecnológico que permite que varios instrumentos musicales electrónicos, computadores y otros dispositivos relacionados se conecten y se comuniquen entre sí.
- **SDK (Software Development Kit):** Es un kit de desarrollo de software que reúne un grupo de herramientas que permiten la programación de sistemas, aplicaciones tanto de escritorio, web, y móviles.
- **Sistema operativo:** Es el conjunto de programas informáticos que permite la administración eficiente de los recursos de una computadora. Estos programas comienzan a trabajar apenas se enciende el equipo, ya que gestionan el hardware desde los niveles más básicos y permiten además la interacción con el usuario.
- **Tracking:** Es una tecnología que permite el rastreo o detección de objetos y de movimiento.

BIBLIOGRAFÍA

- 247 Tecno. (19 de 05 de 2018). *Tecnología MIDI*. Obtenido de 247tecno: <http://247tecno.com/tecnologia-midi/>
- Alfaro Vives, D. A., & Velarde Robles, D. M. (05 de 06 de 2015). *Desarrollo de Aplicaciones con Leap Motion*. Obtenido de Repositorio Académico Universidad Peruana de Ciencias Aplicadas (UPC): <http://repositorioacademico.upc.edu.pe/upc/handle/10757/582073>
- Aprende Web. (13 de 07 de 2018). *Librerías*. Obtenido de aprende-web: <https://aprende-web.net/librerias/>
- Belda, J. (04 de 05 de 2015). *Leap Motion: Principio de Funcionamiento*. Obtenido de Blog Show Leap Tech: <http://blog.showleap.com/2015/05/leap-motion-ii-principio-de-funcionamiento/>
- Brewster, S. (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Minneapolis: Julie A. Jacko; Andrew Sears. Mahwah: Lawrence Erlbaum Associates.
- BTS Bioengineering. (2018). *SMART DX*. Obtenido de BTS Bioengineering Productos: <http://www.btsbioengineering.com/es/products/smart-dx/>
- Ceballos, F. (2012). *Enciclopedia de Microsoft Visual C#*. Madrid: RA-MA Editorial.
- Coté, J.-P. (16 de 08 de 2018). *WebMidi.js*. Obtenido de GitHub: <https://github.com/djipco/webmidi>
- Cuello, D. (2018). *MIDI PLATINUM*. Obtenido de Refraction: <https://refractionproductions.com/lib/libreria-midi-platinum/>
- Deal, M. (13 de 07 de 2015). *MIDI.js*. Obtenido de GitHub: <https://github.com/mudcube/MIDI.js/>
- Engadget. (25 de 03 de 2015). *Razer integra los sensores de manos de Leap Motion en su casco OSVR*. Obtenido de engadget: <https://www.engadget.com/es/2015/03/25/razer-sensores-manos-leap-motion-osvr/>
- Enríquez, J., Farías, E., Flores, E., Honores, C., Llanos, R., López, W., . . . Zúñiga, A. (19 de 12 de 2017). *Metodología de Desarrollo de Software*. Obtenido de Universidad Católica de los Ángeles. Chimbote: <https://www.uladech.edu.pe/images/stories/universidad/documentos/2018/metodologia-desarrollo-software-v001.pdf>
- Flanagan, D. (2006). *JavaScript: The Definitive Guide*. USA: O'Really Media, Inc.
- Hass, J. (13 de 08 de 2012). *Introduction to Computer Music: Volume One. Chapter Three: MIDI*. Obtenido de indiana.edu: http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI3.shtml
- Huber, D. (2007). *The MIDI Manual*. Indiana: Carmel.
- Kruchten, P. (2004). *The Rational Unified Process: An Introduction*. Boston, MA: Pearson Education, Inc.

- Lawrence, S. (2002). *Ingeniería del Software*. Prentice Hall.
- Leap Motion. (2018). *Leap Motion*. (Leap Motion) Recuperado el 7 de Septiembre de 2018, de Leap Motion: <https://leapmotion.com>
- Marin, E. (08 de 10 de 2013). *¿Cómo funciona Kinect de Xbox One?* Obtenido de Hipertextual: <https://hipertextual.com/2013/10/kinect-xbox-one-funcionamiento>
- Martínez, P. (15 de 10 de 2013). *TedCas integra Leap Motion con Sistemas de Imagen Médica*. Obtenido de TodoStartups: <https://www.todostartups.com/bloggers/tedcas-integra-leap-motion-con-sistemas-de-imagen-medica>
- Microsoft. (2018). *Kinect*. (Microsoft) Recuperado el 7 de Septiembre de 2018, de Microsoft Developer: <https://developer.microsoft.com/es-es/windows/kinect>
- Moreno, R. (25 de 05 de 2013). *Historia del MIDI*. Obtenido de Diffusion Magazine: <http://www.diffusionmagazine.com/index.php/biblioteca/categorias/historia/335-historia-del-midi>
- NANSENSE. (2018). *R2 Black Suit*. Obtenido de NANSENSE Productos: <https://www.nansense.com/>
- OBS. (2018). *¿Qué son las metodologías de desarrollo de software?* Obtenido de OBS Business School: <https://www.obs-edu.com/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>
- Oyanedel, J. (28 de 08 de 2014). *Leap Motion estrena accesorio para integrarse al Oculus Rift*. Obtenido de FayerWayer: <https://www.fayerwayer.com/2014/08/leap-motion-estrena-accesorio-para-integrarse-al-oculus-rift/>
- Pérez, F. J. (03 de 2011). *Creatividad y Sociedad*. Obtenido de Revista Creatividad y Sociedad: <http://creatividadysociedad.com/articulos/16/4-Realidad%20Virtual.pdf>
- Refraction Productions. (2018). *RADIO BOMBS VOL.1 - MIDI EDITION*. Obtenido de Refraction: <https://refractionproductions.com/lib/libreria-midi-radio-bombs-vol-1-MIDI-edition/>
- Refraction Productions. (2018). *REFRACTION ESSENTIAL CHORDS*. Obtenido de Refraction: <https://refractionproductions.com/lib/essential-chords-secuencias-acordes-midi-estructura-tema/>
- Stallings, W. (1997). *Sistemas Operativos*. Madrid: Pearson Educación S.A.
- Sugar Sounds. (2018). *FUTURE MIDI AND LOOPS*. Obtenido de Refraction: <https://refractionproductions.com/lib/future-midi-loops-deep-house-sugar-sounds/>
- Tecnología de Tú a Tú. (15 de 05 de 2014). *Leap Motion, el reconocimiento de movimiento llega a los portátiles*. Obtenido de El Corte Inglés. Tecnología de Tú a Tú: <https://www.tecnologiadetuu.elcorteingles.es/actualidad/leap-motion/>

- The MIDI Association. (2018). *Why MIDI Matters*. Obtenido de midi.org: <https://www.midi.org/articles/why-midi-matters>
- Tinoco, Ó., Rosales, P., & Salas, J. (14 de 12 de 2010). Criterios de selección de metodologías de desarrollo de software. *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal*, 1-5. Obtenido de <http://www.redalyc.org/html/816/81619984009/>
- Velasco, J. (23 de 04 de 2013). *Google Earth integra el control por gestos de Leap Motion*. Obtenido de Hipertextual: <https://hipertextual.com/2013/04/google-earth-se-integra-con-leap-motion>
- Yáñez, S. (31 de 07 de 2010). *Modelo de Prototipo*. Obtenido de slideshare.net: <https://www.slideshare.net/yanezcabrera/modelo-de-prototipo>
- Zeng, W. (2012). Microsoft Kinect Sensor and Its Effect. *Multimedia at Work. University Of Missouri. IEEE*, 4-10.