

geminiflux: Música electrónica, improvisación y código abierto

Andrés Cabrera

Universidad Javeriana – Departamento de música

Cr. 7 #40 – 62 – Bogotá - Colombia

andres@geminiflux.com

***Sinópsis.** 'geminiflux' es un proyecto de música electrónica con una gran dosis de improvisación y tratamiento de audio en tiempo real. Es una exploración en técnicas de improvisación dentro de la música electrónica y en el control inusual de herramientas electrónicas acompañada de elementos visuales para buscar un nuevo estilo de 'teatro musical'. El desarrollo de la obra '01-radioreloj' se hizo usando herramientas muy variadas y la ejecución se hace principalmente con herramientas de código abierto. La creciente complejidad del proyecto durante su fase de composición generó necesidades y problemas adicionales que definieron la interfase de ejecución y condujeron a la adopción de Linux como sistema operativo para el procesamiento de audio.*

***Abstract.** The 'geminiflux' project plays electronic music incorporating plenty of improvisation and real-time audio processing. It explores improvisational techniques within electronic music and employs unusual control interfaces for the electronic tools, coupled with visual elements to create a new kind of 'musical theater'. Development of the piece '01-radioreloj' was done using many different tools and its performance employs primarily open source software. The growing complexity and needs of the project during composition, shaped the interface design, and led to the final adoption of Linux as operating system for audio processing.*

1. Introducción

El proyecto 'geminiflux'[4] (en mal latín 'fluir doble') surge como un deseo de explorar técnicas de interpretación en tiempo real usando software avanzado, llevando el resultado hacia un lenguaje musical directo y asequible, simultáneo con un montaje atractivo en vivo. 'Geminiflux' está compuesto por Andrés Cabrera, que cuenta con amplia experiencia en el campo de la post-producción audio-visual y la docencia, además de participar en el desarrollo de plug-ins para *csound* y Gonzalo Sagarmínaga quien es un reconocido intérprete y compositor de música para cine y televisión.

La influencia creativa que ha dictado el rumbo de la concepción musical es la obra 'Black on White' de Heiner Goebbels [5]. Esta obra es una emocionante mezcla de teatro e improvisación musical. Al inicio de la concepción de 'geminiflux', fue claro que se quería buscar un efecto similar al de esta obra, pero a través de herramientas muy diferentes. Mientras que 'Black on White' utiliza virtuosos músicos académicos que son al mismo tiempo músicos y actores, 'geminiflux' mezcla la experiencia de la música electrónica y el rock con herramientas tradicionalmente asociadas a la composición electroacústica, en un montaje en el que los músicos y aún el computador son actores. El resultado de la primera exploración de 'geminiflux' es '01-radio reloj'. Es un montaje marcadamente teatral, que requiere que la tecnología que soporta en gran medida el espectáculo, funcione sin que se evidencie un esfuerzo por parte de los interpretes por hacerla funcionar. Para lograr esto la mayoría del control de la ejecución

(incluyendo cambio de secciones) se hace vía MIDI -sin manipulación directa del computador-, así la tecnología adquiere un carácter propio e independiente de los músicos y se evitan pausas largas entre las secciones dando gran fluidez y una sensación de unidad. Aunque ésta fue desde el principio la idea original, los medios por los cuales se ha llevado a cabo han ido transformándose según las necesidades y a medida que se han tenido nuevos recursos disponibles (actualizaciones de software o hardware).

A pesar de ser una obra principalmente musical, por tener músicos interpretando instrumentos en vivo, '01-radioreloj' tiene una indispensable presencia de video y de teatralidad. Una proyección de video presenta diferentes propuestas visuales de acuerdo con la actividad musical, que tienen profunda relación con la propuesta musical, y en muchos casos aclaran o ayudan a definir esta propuesta. La práctica de que imágenes de video acompañen la música electrónica está usualmente asociada al trabajo de los VJ, y aunque algo de esta estética popular urbana esté presente en el video de 'gemini flux', la forma en que el video se entrelaza tanto temática y conceptualmente como rítmicamente con la música le da un carácter especial. La proyección de video, por incluir procesamiento de imágenes capturadas en vivo, subraya la relación entre músicos, actuación y tecnología. Se habla de teatralidad porque los músicos entran al escenario y se comportan en él como actores, es decir, sus acciones además de producir música, los definen como entidades en el escenario. Aunque estos elementos visuales son inseparables de la obra, este documento se centrará en los aspectos técnicos y estéticos que atañen a la creación sonora y no pretende por esto dar una visión completa de la obra.

1.1. Herramientas para la ejecución

Actualmente, aparte del sonido de los músicos en vivo, la mayoría de '01-radio reloj' proviene del proceso de dos computadores (uno corriendo Linux -Fedora Core 2 Planet CCRMA- [6] que se encarga del audio y otro corriendo Windows XP, encargado de la imagen). La razón de mezcla de sistemas operativos surge de necesidades específicas que sólo se satisfacen en uno de los dos sistemas, y que se discutirán más adelante. El procesamiento y generación de audio está a cargo de *csound* [1] (y en pequeña medida de *pure data*[9]). El video es generado por *pure data* usando su módulo de video *Gem* [14]. Los computadores están interconectados bidireccionalmente usando OSC [12] a través de UDP en cable ethernet.

Además del computador como generador y procesador de sonidos, están presentes diversos instrumentos como guitarras eléctricas, harmónica, bajo eléctrico, un teclado Korg Triton, rototoms, platillos y voces, todos conectados a la consola, y siendo amplificados con su sonido original y en algunos casos específicos con reverberación o flanger de una unidad de efectos externa.

Existe otra categoría de instrumentos, que fueron diseñados específicamente para '01-radio reloj', que involucran una mezcla de interacción acústica con procesamiento y síntesis electrónica. Fueron diseñados para ser efectivos tanto como efecto teatral e histriónico como para lograr timbres y posibilidades interpretativas inusuales. El primero es un sillín de batería que se menciona más abajo en la sección 2.5, el segundo una grabadora de periodista que se enciende y su reproducción es procesada en la sección 2.9, y un tercero que se ha llamado 'hidrófono' se menciona en la sección 2.11.

El control de la ejecución se hace desde tres controladores: Un teclado Radium49 (de 4 octavas y 16 controladores MIDI), un controlador MIDI infrarrojo Eyris (que puede enviar tanto notas como información de control) y un guante P5 (que por medio de USB puede enviar información de control).

El sonido de toda la obra es cuadrafónico, tanto en los procesos en tiempo real como en las secciones que tienen elementos pre-grabados. Los cuatro canales que salen del computador y los sonidos provenientes de los instrumentos entran a un mezclador principal donde son espacializados adecuadamente. En algunas secciones las voces entran directo a la consola y son amplificadas (con algo de reverberación de un procesador externo), mientras que en otras, las voces pasan por procesos electrónicos en el computador.

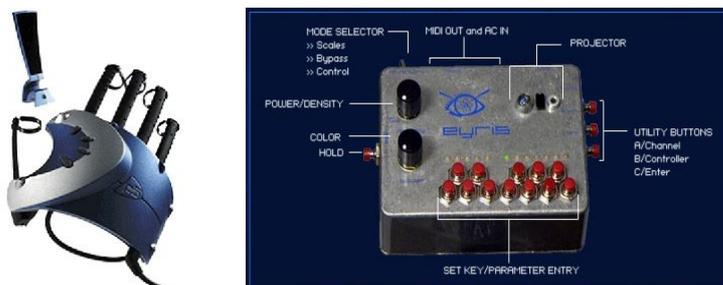


Figura 1. Guante P5 y controlador Eyriss

2. Estructura de '01-radio reloj'

La obra '01-radioreloj' intenta imitar la experiencia de escuchar música en radio presentando una serie de canciones continuas sin relación (ni estilística ni sonora) entre ellas. Cada canción tiene un estilo muy individual, y sus propios motivos histriónicos y experimentos interpretativos, lo que resulta en una serie de unidades totalmente independientes. Se hace un gran barrido por diferentes estilos de música electrónica, y aunque se presentan principalmente estilos populares de los últimos 20 años, también se escuchan sonidos reminiscentes de música electrónica académica -en la introducción y el sueño-. La presentación completa tiene una duración aproximada de una hora.

2.1. Introducción

La introducción, que inicia con el computador solo en escena (programado para arrancar a cierta hora específica), presenta un inicio que busca confundir al espectador sobre lo que va a suceder posteriormente. Presenta una serie de 'campanazos' realizados por medio de síntesis granular a partir de una muestra de audio de la segunda sección. Se inicia con unos sonidos profundos extraños -generados a partir de la granulación de la muestra transponiendo los granos y recorriéndola lentamente- que van transformándose hasta convertirse en uno de los motivos principales de la siguiente sección [10]. Esta transformación se logra acercando los parámetros de la síntesis cada vez más al punto en que el sonido se convierte en la muestra original -es decir sin transposición y usando la velocidad original-.



Figura 2: Leit-motiv

El motivo de la muestra es un leit-motiv en la obra y se compone de tres pares de notas sincopadas en un instrumento electrónico reminiscente a un xilófono. Tanto la música como el video de esta sección son simples y mecánicos, para introducir y dar carácter al 'personaje' del computador, presentándolo como una máquina fría que sin intervención humana, es capaz sólo de crear sonidos estériles. Este inicio contrasta fuertemente con el resto de la obra, que es muy viva e histriónica.

2.2. Primate melancólico

Esta sección surge a partir del motivo generado en la introducción. Es una pieza de elementos minimalistas y totalmente diatónica, que a pesar de ser más emotiva que la anterior conserva su carácter frío. En esta sección el video empieza a mostrar colores, que aún parecen primitivos, acompañando el surgimiento de armonías consonantes en la música. Se escuchan tres líneas instrumentales: el motivo generado en la introducción, que al repetirse constantemente genera una especie de pasacaglia, una línea alta de sonido similar a una flauta y un bajo que inician con motivos muy sencillos que van ganando movilidad melódica y armónica a lo largo de la sección. Los sonidos provienen del sintetizador Absynth [8] de Native Instruments, que por su versátil control de envolventes de parámetros de la síntesis permite crear sonidos muy orgánicos.

2.3. Ionosphere

Cuando empieza a morir el sonido de la sección anterior, se dispara el sonido de un radioreloj que ha sido previamente sincronizado para que inicie en este momento. Una emisora que ha sido seleccionada durante el ensayo empieza a sonar y luego empieza a ser amplificada. En este momento entra 'el técnico' (Andrés Cabrera) en escena. El personaje de 'el técnico' está a cargo de controlar los procesos electrónicos y de manejar la acción relacionada con la tecnología, y será siempre el puente entre el computador y el segundo personaje: 'el músico' (Gonzalo Sagarmínaga). Es un personaje frío que no interactúa con el público, y su relación en el escenario es siempre con el hardware.

En esta sección empieza a hacer un muestreo (usando líneas de retardo) de lo que esté sonando en radio en el momento, que luego se combina con estática y otros ruidos generados por el radioreloj, para comenzar a crear una textura rítmica, que se mezcla luego con una música previamente preparada con un ritmo similar a una champeta o un reggetón. El muestreo del radio se mezcla y se manipula por medio de filtros y otros efectos para dar variedad, y se vuelve prominente en dos secciones en las que se detiene la champeta, e inicia un ruido de ambulancias y luego de guerra. Aquí se hace fuerte uso de la espacialización cuadrafónica.

2.4. Blues

Mientras termina 'Ionosphere', entra al escenario 'el músico', el segundo personaje, que representa la parte emocional en la obra, porque su interacción con lo tecnológico es mínima (y cuando se da más adelante en 'electric cowboy' es cómica). Empieza a tocar una improvisación de estilo blues con guitarra y armónica, y luego comienza a ser muestreado igual que el radio para generar texturas rítmicas densas. 'El músico' también es muestreado en video, y se ven varias imágenes de él en bucles de diferente duración en la pantalla. Esta sección es improvisada sobre una estructura predefinida muy libre, y se genera un estilo algo contradictorio por combinarse la improvisación estilo blues con una rítmica y bucles característicos de música techno.

La voz en esta sección improvisa además de la melodía, la letra, en un idioma que suena como inglés pero que no tiene significado, lo que algunas veces se conoce como 'singlés'. Este 'singlés' -improvisación en melodía y texto- estará presente en todas las demás secciones con excepción de 'pussycat'.

2.5. Parsons

Cuando termina el blues, con una serie de ciclos arítmicos, se inicia la siguiente sección, que tiene un carácter totalmente electrónico, con una base rítmica que va aumentando en textura

mientras 'el músico' toca con baquetas de batería el sillón de una batería, que alimenta unos resonadores virtuales configurados desde el controlador MIDI, dando unos sonidos inusuales que son percutidos pero afinados. La percusión electrónica se hace más densa hasta que hay una explosión en la que entra un bajo sintetizado y una línea de sintetizador que parece generada por un arpegiador. En esta sección 'el músico' toca rototoms y platillos que son procesados con flanger. Luego 'el técnico' improvisa en guitarra eléctrica procesada para generar notas muy largas y de ataque lento, mientras 'el músico' improvisa en un teclado. Acercándose al final, 'el músico' grita y susurra al micrófono pasando por un waveshaper para darle distorsión, y enviándolo a un delay cuya presencia y feedback son controlados por MIDI [3].

2.6. ACDC

Cuando finaliza 'Parsons' se da comienzo a ACDC una canción que es mucho menos mecánica, y aunque de sonido muy electrónico, tiene más movilidad. Se exploran sobretodo efectos en la voz, como distorsión por waveshaping, vocoder y delay . El 'músico', además de improvisar la letra y la melodía (sobre un modelo básico), improvisa en el órgano. En esta sección se explora la relación entre video y música generando texturas visuales contrastantes que son paralelas a los procesos contrastantes que sufre la voz (distorsión-vocoder).

2.7. Nobody wan

Esta especie de interludio es una pieza lenta que busca dar un toque de tranquilidad y emotividad. Tiene dos secciones muy claras que aunque relacionadas tímbricamente, pueden categorizarse como oscura y clara. Representa una salida a la claridad y la tranquilidad a partir de un estado de angustia pasiva. Para mantener el carácter contemplativo, esta sección no tiene ningún elemento histriónico como las anteriores, sino que se centra en la emotividad de la música únicamente. Aquí la improvisación se da en guitarra y voz.

2.8. Electric cowboy

Esta sección presentada por el 'músico', sin acompañamiento de video o electrónica, hace un comentario tácito y humorístico sobre la tecnología, porque el intérprete para generar acordes con el teclado, sobre los que improvisará con la armónica, pega con cinta las teclas para que queden sonando. Esta sección 'primitiva' es un fuerte contraste con la siguiente altamente tecnológica.

2.9. Sueño

En 'Sueño' se presenta una experimentación con controladores inusuales y programación de gestos para la generación de notas. Todo el sonido es generado por el 'técnico' a partir de dos controladores (un Eyriss -controlador MIDI infrarrojo- y un guante P5), que dirigen tanto el sonido como el video. La programación de la respuesta del computador a estos controladores se da de dos maneras: la manera tradicional que es mapeando un parámetro del controlador a un parámetro que controla la síntesis, y la otra es a partir de definición de gestos. Los gestos son simples comparaciones de estado del controlador (el guante P5), que cuando se cumplen generan una acción determinada. Por ejemplo, cuando se flexionan dos dedos (el anular y el meñique), se activa la posibilidad de generar notas de un instrumento virtual que dependen de la altura (eje y) del guante. Otro caso es que cuando se cierran los dedos como un puño, se envía información a *csound* para que dispare unas muestras de la pieza granular introductoria y de 'Primate melancólico'. El gesto de flexionar el pulgar, por ejemplo, apaga todas las notas.

La ejecución tiene un orden definido, ya que el mismo gesto cambia de función (luego de varias veces de hacer el mismo gesto, este produce diferente resultado), y aunque se tiene mucha

libertad en el tiempo, el orden se ajusta a un texto (ver más abajo).

El 'técnico' utiliza en una mano el guante P5 y con la otra utiliza el controlador infrarrojo Eyris, lo que resulta en una extraña 'danza' con los brazos. El Eyris se configura para que envíe mensajes de noteon y noteoff, que aunque pertenecientes a la escala de do mayor, se les convierte usando *pure data* en un arpeggio de un acorde disminuido.

En esta sección, el 'músico' enciende una pequeña grabadora de periodista frente a un micrófono, y un texto grabado es luego procesado por varios delays con tiempos y filtros diferentes, creando unas extrañas texturas rítmicas a partir de la voz. Algunos de los sonidos generados vienen de 'primate melancólico' y la introducción, aunque son difícilmente reconocibles en este contexto. Esto acompaña los otros sonidos generados por los controladores (a través de plugins VST [11]) creando una extraña textura surrealista y onírica.

La sección comienza usando una grabación de los samples del radio que se hicieron en 'Ionosphere'. Inician claramente reconocibles hasta que se transforman en sonidos extraños, como el momento en el que se entra en el sueño. Toda la actividad es realizada por el 'técnico' que se convierte en una especie de hipnotizador controlando el sueño del 'músico' (esto es evidente en el video en el que salen figuras de su cabeza) y llevándolo a una especie de pesadilla.

2.10. Pussycat

'Pussycat' es la continuación de la pesadilla. Es una pieza en estilo 'electro', construida exclusivamente a partir de muestras procesadas (transposición con envolventes, filtros, etc.) de sonidos de una fábrica de mantenimiento de aviones, y una simple línea de bajo sintetizado. Por esto tiene un sonido agresivo y mecánico. Está cantada por una mujer y es la única que tiene un texto 'coherente'. Hace extenso uso de paneos cuarafónicos para crear la textura percusiva.

2.11. Party

Esta sección inicia con una introducción de estilo cinematográfico, que desemboca en una canción alegre con influencia del tecno pop de los años 80. Es una canción que busca terminar en clímax por ser la última sección en la que los músicos están en el escenario. En esta sección aparece el 'hidrófono', un instrumento diseñado para esta obra, que consiste de una olla de cocina con un pick-up de teléfono que luego es procesada por un agresivo waveshaper diseñado en *csound*, controlando volumen y profundidad del waveshaper desde MIDI. El efecto es a la vez emocionante y cómico, como un gran solo de guitarra eléctrica 'a la Hendrix' con una olla. Estos solos de 'hidrófono' están intercalados con las secciones 'pop', produciendo una estructura inusual. Esta canción está conectada con la siguiente por transición al estilo DJ (por crossfade).

2.12. Primate alegre

Esta sección final es una repetición idéntica de la sección 2.2 'Primate melancólico', pero esta vez, se ha agregado un beat electrónico y otras texturas rítmicas (a partir de elementos percusivos procesados con modulación y delay), que junto con el aumento del tempo, le dan gran claridad rítmica a la pieza (particularmente al motivo que se ha repetido a lo largo de la obra), y mucha alegría. Este gran contraste de atmósfera con la primera aparición de 'Primate', representa el resultado de la inspiración agregada a la tecnología. Es despojar la tecnología de su aspecto frío e inhumano para convertirla en algo vibrante y emocional. El computador cierra la obra solo en el escenario tal como empezó.

3. El proceso de composición

3.1. Inicio

El primer paso hacia '01-radio reloj' fue el diseño de varios algoritmos para *csound* a finales del 2003. Estos algoritmos permanecen en las secciones 'Ionosphere' y 'Blues' y generan las líneas de delay independientes. También en esta primera fase se hizo la composición de la introducción usando síntesis granular -también en *csound*- y 'Primate melancólico' que se compuso en Sonar 3¹ usando principalmente el sintetizador de software Absynth de Native Instruments. También se hicieron los primeros bocetos para el video de estas dos secciones usando las herramientas gráficas de *csoundAV*.

Al unirse Gonzalo al proyecto se inicia la composición del grueso de la obra usando principalmente Sonar. La composición se hizo en un secuenciador comercial tradicional, pero siempre se tenía presente el rol de la improvisación y la interpretación en vivo, dejando siempre espacio para éstos. A lo largo de este proceso se planeaba generar el audio y el video usando un solo computador corriendo *csoundAV* en Windows XP y haciendo la edición del código usando winXsoundPro. La razón por la que se eligió el software de código abierto *csound* fue por su gran versatilidad tanto para el procesamiento y generación de sonido como por la posibilidad de tener gran interactividad. Otra razón por la que no se eligió algún software comercial como Live o Reason fue que se quería que las secciones de la obra fluyeran ininterrumpidamente y sin manipulación del computador, algo imposible con estos programas ya que se tendría que cargar un nuevo archivo para cada sección. Sin embargo, pronto surgieron las siguientes necesidades adicionales:

- Se vio la necesidad de usar en tiempo real plugins VST para tener una paleta versátil de sonidos, y *csoundAV* no tenía (ni tiene aún) la capacidad de correrlos. Como primera solución se escribió un módulo (en C++) que permitía usar plugins VST en *csound5*, pero *csound5* aún no era apto para uso en tiempo real en Windows en esa época, así que se debió recurrir a *pure data* para alojar estos plugins. Esto implicó que se debía tener un computador adicional ya que no es posible en Windows XP que dos programas independientes accedan a una tarjeta de sonido cuando se quieren usar drivers de baja latencia como ASIO. Desde este momento se comenzó a separar el proceso de audio entre *csound* y *pure data* en dos computadores, sincronizándolos a través de MIDI. La sincronización se hizo a través de mensajes note-on en el canal 15, con un valor de nota para encender y otro diferente para marcar el final de sección. La información de controladores MIDI pasaba directamente del computador con *csound* al de *pd* para controlar parámetros de los plugins VST, que en un principio fueron Cyanide2 (distorsión por waveshaping, gratuita) y Orange Vocoder (de Prosoniq).
- Se comenzó a presentar dificultad para hacer ensayos que no iniciaran desde el principio, por la estructura de la sección de score de *csound*. Se intentó armar un sistema usando macros para simplificar esto, pero seguía siendo engorroso. Adicionalmente, el código había alcanzado un tamaño considerable, y no era fácil hacer ajustes con un editor tradicional. La solución llegó con blue [13] -un front-end para *csound* escrito en java- que permite iniciar la ejecución de *csound* desde cualquier punto en el tiempo.² Blue sigue siendo el editor actual de archivos de *csound* en '01-radio reloj'.

1 Sonar es un secuenciador MIDI y grabador multi-track de audio digital.

2 Esta limitación no se ha superado del todo, y siempre toca tocar secciones desde el principio.

- Las herramientas gráficas de *CsoundAV* empezaron a ser insuficientes, y se transportó con considerable esfuerzo el trabajo ya hecho a *pure data+gem* que ofrece una gama más amplia de herramientas de video y sobretodo la posibilidad de procesar el video capturado por una cámara en tiempo real. Esto implicó nuevos retos de sincronización, ya que se vió que no era confiable correr el audio y el video en una sola instancia de *pd*, sino que debían correr en instancias separadas. Esto supuso la necesidad de instalar un cable virtual (MidiyokeNT), y un patchbay MIDI virtual (Midiox) para intercomunicar las dos aplicaciones que estuvieran en un mismo computador.
- Empezó a notarse un comportamiento errático e impredecible de *CsoundAV* al usar muchos *if...then* dentro de un instrumento, que se requerían para seleccionar la pista que debía sonar según la sección, así que se asignó el trabajo de reproducción de pistas a *pd*.

En esta fase se empezó a asentar la primera versión funcional de '01-radio reloj', en la que se usaron dos computadores de la siguiente forma:

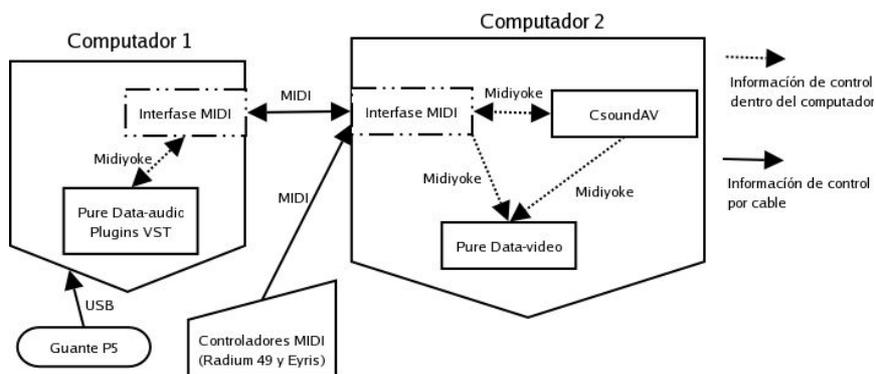


Figura 3. Esquema de la primera versión funcional

Se diseñó un simple protocolo de sincronización generado por *csound*, que no incluía ninguna información de posición sino solo gatillos de inicio, que ha demostrado ser suficiente, ya que no se requiere en ningún momento una sincronización precisa entre las aplicaciones, sino sólo que cada una ejecute la tarea adecuada según el momento. Este protocolo de mensajes de note-on, permanece en la versión actual.

El protocolo cuenta principalmente con dos partes. Una que usa el canal 15 indica a los programas qué sección de la obra está sonando. La sección determina qué algoritmos deben activarse o apagarse, tanto en audio como en video. Los canales 1-8 determinan controles específicos de cada sección, siendo la primera 'Ionosphere' y la octava 'party' (la sección 'Electric cowboy' no tiene ningún control MIDI). Por esto en cada cambio de sección se envía un mensaje desde *csound* que indica que una sección termina -e inicia la siguiente-, y que se debe escuchar en el canal MIDI siguiente, teniendo que configurar al controlador MIDI (Radium49) para que transmita por el canal siguiente. La ventaja de separar las secciones por canal MIDI garantiza que no se pueden activar funciones indeseadas -como activar por error una función de otra sección-.

La organización de la función de las teclas del controlador para ayudar a la memoria se hizo agrupando funcionalidad y separando funciones por octava. El controlador usado es un Radium49, que tiene 49 teclas (4 octavas), 8 sliders y 8 perillas (pots- numeradas 9 a 16). Por ejemplo la sección 'Ionosphere' -que debe transmitir por el canal MIDI 1- se controla así:

Nota	Función
C4 a A4 (teclas blancas)	Estas notas controlan el muestreo del radio, enviando la entrada de audio a una de las 6 líneas de delay, cada una representada por una tecla blanca. Cuando la tecla esta presionada, el audio es enviado a la línea de delay.
F#4, G#4 y A#4	Estas tres teclas negras controlan respectivamente el filtro, flanger y gating que se aplica al sonido. Cuando la tecla está presionada, el efecto está activo.
C5	Inicio del kick drum rítmico. Tiene efecto en el video también.
D5	Inicia la secuencia pregrabada. Debe iniciar sincronizada al kick que se disparó con C5. Produce un cambio en el video.
F5	Re-dispara el kick en el caso de que haya quedado desajustado o impreciso con respecto a los loops (del radio) que ya están sonando.
A5	Graba al disco duro el sonido del loop más largo que luego se usará en la sección 'sueño'.
F#5, G#5, A#5, C#6, D#6, F#6	Controlan diferentes efectos y secciones del video.

Tabla 1. Teclas de control para la sección 'Ionosphere'

Adicionalmente a estas teclas, controladores MIDI tienen efecto en el proceso. En el caso de Ionosphere están distribuidos así:

Slider	Función
1 al 6	Controlan independientemente la retroalimentación de cada una de las líneas de delay.
7 y 8	Controlan la frecuencia y resonancia del filtro resonante pasa-bajos que se activa con F#4.

Tabla 2a. Controles MIDI para la sección 'Ionosphere'

Perilla	Función
9, 10, 11	Nivel de las líneas de delay 1,2 y 3
12	Nivel de las líneas 4, 5 y 6
13	Nivel master de las líneas de delay
16	Nivel de la señal del micrófono para amplificar el sonido del radio

Tabla 2b. Controles MIDI para la sección 'Ionosphere'

La configuración de teclas y controladores para las demás secciones sigue patrones similares organizando funciones en la misma octava en las teclas blancas o las negras. La tecla C7 (96) indica en todos los casos cambio de sección. Esta opción de cambiar de sección por medio de comandos en tiempo real en vez de estar programada es necesaria porque algunas de las secciones no tienen una duración definida, sino que dependen de cómo se desarrolle la improvisación.

La elección de usar el computador 2 para *csound* y el video obedeció a que el trabajo que realizaba el *csound* (líneas de delay, control maestro) era menos exigente para el procesador que el que realizaba pure data (que debía alojar plugins VST, y reproducir archivos de audio largos), y se escogió el computador con el hardware de gráficas más apropiado para este trabajo.

Con esta disposición se terminó la primera versión funcional de '01-radio reloj' que ha sido presentada en el Teatro Colón de Bogotá como concierto de cierre del ciclo anual 'Colón Electrónico' en diciembre de 2004 y luego en la Corporación Colombiana de Teatro en febrero de 2005.

Sin embargo este sistema presenta algunas complicaciones que se han querido corregir:

- El hecho de usar MIDI ha sobrecargado el sistema en secciones donde se envía mucha información, particularmente cuando se envía información acerca del nivel *rms* del sonido. Esta información que genera el *pd* de audio se envía 60 veces por segundo con una precisión de 14-bits usando dos controladores diferentes. Como esta información debe entrar por la interfase MIDI al computador 2, y luego ser repartida a través de dos cables MIDI virtuales -uno conectado a *csound* y el otro a *pd*, el impacto en el desempeño del sistema es de consideración. Experimentos usando OSC para la transmisión de esta información demostraron que es más ligero para el sistema además de ofrecer mayor precisión en el tiempo y en los valores. Sin embargo *csound* hasta hace muy poco tiempo (y solo en versiones beta) soporta OSC.
- Tener el audio separado en dos computadores implicaba tener que usar una consola de más de 16 canales, que tiende a ser más costoso e impráctico. Lo ideal sería que un solo computador produjese el audio, ya que la limitación no es la complejidad del proceso, sino el hecho de que en Windows XP dos programas no pueden acceder simultáneamente al hardware con la baja latencia requerida para trabajo en tiempo real.
- Ocasionalmente ocurría que el sistema se caía cuando recibía una nota MIDI, por causas que parecen apuntar a algún problema en el subsistema o librerías MIDI de Windows cuando se corren varias aplicaciones MIDI y hay una carga alta en el procesador. Se ha visto que este problema está relacionado con el disco duro externo que aloja los archivos de audio.
- Otro problema molesto surgía por un bug en *gem* con el objeto [separator] que aún no ha sido corregido. La ejecución del video debió separarse en dos partes, teniendo que cerrar *pd* entre una y otra. Sin embargo, al cerrar *pd* en estas circunstancias, por alguna razón aún desconocida, la información MIDI dejaba de viajar entre las aplicaciones a través de los cables virtuales. Esto implicó que se debía reiniciar el computador 2 en medio de la presentación, algo que se escondió con la improvisación acústica de armónica y teclado por parte del 'músico' -llamada 'electric cowboy'- mientras el 'técnico' reiniciaba y arrancaba de nuevo el procesamiento de audio y de video, con las correspondientes conexiones MIDI virtuales y la configuración de los plugins VST.

El problema era una paradoja, porque ¿Cómo separar el video del audio si se deben correr dos programas de audio que deben estar en computadores separados sin usar un computador más? En ese momento, me encontraba experimentando con *Linux* hacía algún tiempo, y la posibilidad de usar *csound* y *pd* a través de *Jack* [2] permitiría hacerlo. Sin

embargo el cambio no sería sencillo.

En primer lugar, la versión de *Csound* que se estaba utilizando era *CsoundAV* 0.043, [7] que contenía funciones aún no disponibles en la versión oficial de *Csound 4* para *Linux*, así que se trabajó para portar los opcodes de *CsoundAV* a *Csound5* (que tiene una arquitectura más moderna, y es más fácil de extender). Afortunadamente, por la misma época (a principios del 2005), la comunidad de desarrolladores llegó una versión estable (aunque aún beta), que corría eficientemente en tiempo real y era capaz de conectarse a *Jack*.

Los plugins VST (Absynth, Orange Vocoder, Cynanide2 y JX220) que se habían usado son programas de Windows, que después de algún trabajo se hicieron correr sobre Linux usando WINE y *jackfst*. Sólo uno de ellos no funcionó (JX220) pero como no era uno de timbre demasiado importante, se substituyó por *polyblit* que dió unos timbres similares. Para realizar la mezcla y nivelación de estos plugins se conectan a través de Jack a *csound* y *pd* que se encargan de la nivelación.

Inicialmente se pensó en conectar el guante P5 directamente a *Linux*, pero las versiones más recientes del kernel (se está usando 2.6.10) no soportan aún el guante, así que se debió escribir un pequeño programa en *pd* (en Windows) que convirtiera la información MIDI del guante a OSC. El montaje definitivo está presentado en la figura 4.

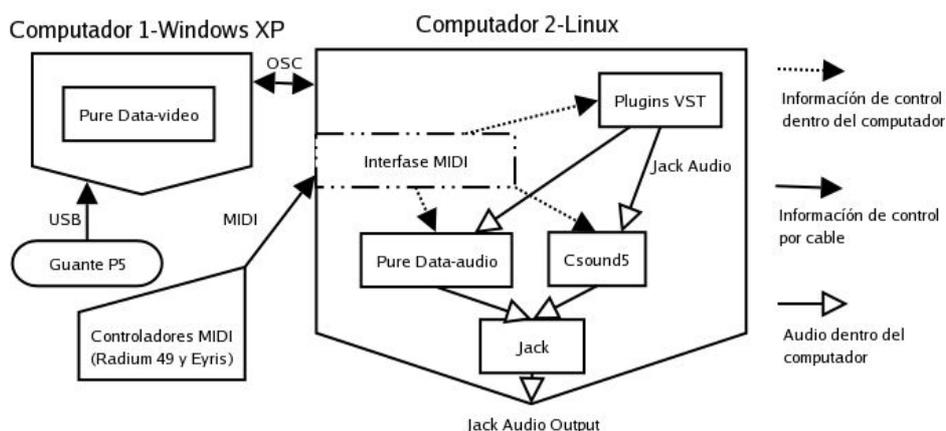


Figura 4. Esquema de la versión final de montaje

4. Conclusiones

El proceso de desarrollo de 'gemini flux' por sus intensos requerimientos técnicos, y por sus ambiciosas intenciones estuvo lleno de obstáculos impredecibles e inesperados. Como la intención y la dirección del proyecto siempre estuvo clara se hicieron todos los esfuerzos para que las limitaciones técnicas nunca frenaran el proceso y la visión creativa. El software de código abierto estuvo presente desde la concepción del proyecto, pero la adopción de *Linux* como sistema operativo de uno de los equipos fue una consecuencia inesperada. Con este documento se quiere demostrar la madurez y versatilidad de estas herramientas gratuitas, en un contexto en el que brillan sus mayores fortalezas.

La experiencia con Linux ha sido inmensamente positiva, y aunque es cierto que es un sistema operativo que puede tomar tiempo para conocer, ha posibilitado la ejecución de 'gemini flux' dando una estabilidad que no se había logrado en Windows. Para futuras producciones de 'gemini flux' se espera usar linux en todo el proceso, incluso para el video. De esta forma se simplificará la interacción entre los diferentes programas, y se facilitará la compatibilidad

futura por no usarse software propietario. Además, el hecho de usar software de código abierto permite extender y modificar la funcionalidad del software según sea necesario.

5. Referencias

- [1] **Boulanger, R.** 2000. ed. *The Csound Book*. MIT Press. Cambridge, Massachusetts.
- [2] **Davis, P, et. al.** JACK- Jack Audio Connection Kit. <http://jackit.sourceforge.net/>
- [3] **Dodge, C. , Jerse, T.** 1997. *Computer Music*. 2nd Edition. Schirmer Thompson Learning.
- [4] **Gemini Flux.** <http://www.geminiflux.com>
- [5] **Goebbels, Heiner.** 1998. *Black on White*. <http://www.heinergoebbels.com>.
- [6] **Lopez-Lezcano, F.** 2002. The Planet CCRMA Software Collection. Proceedings of ICMC 2002. International Computer Music Association. San Francisco.
- [7] **Maldonado, Gabriel.** 2001. *CsoundAV*. <http://www.csounds.com/maldonado/>
- [8] **Native Instruments Inc.** *Absynth*. http://www.native-instruments.com/index.php?id=absynth3_us
- [9] **Puckette, M.** 1996. *Pure Data: another integrated computer music environment*. Proceedings of ICMC 1996, Goeteborg, Sweden. International Computer Music Association.
- [10] **Roads, C.** 2001. *Microsound*. MIT Press. Cambridge, Massachusetts.
- [11] **Steinberg.** *VST Plugin standard*. <http://www.steinberg.de>
- [12] **Wright, M., A. Freed.** *Open Sound Control: A New Protocol for Communicating with Sound Synthesizers*, Proceedings of ICMC 1997. International Computer Music Association. San Francisco.
- [13] **Yi, S.** Blue. <http://www.csounds.com/stevenyi/blue/>
- [14] **Zmoeling, I.** maintainer. *Gem*. <http://gem.iem.at/>

5.1. Hardware de control

Eyris Infrared MIDI controller. <http://www.synesthesiacorp.com/indecks.html>

P5 glove. http://www.zzz.com.ru/index.php?area=pages&action=view_page&page_id=11

Radium49. http://www.midiman.com/products/en_us/Radium49-main.html