

# **Sistemas interactivos en tiempo real. Softwares Isadora, EyeCon, MaxMsp, Pure Data y otros. Prof. Maximiliano Wille<sup>1</sup>.**

Sistemas, Interactividad y Tiempo Real: Para hablar sobre los **Sistemas interactivos en tiempo real** (SITR) debemos, primeramente, separar y diferenciar algunos términos. La profundización respecto de teorías sistémicas excede los propósitos de este trabajo, pero desarrollaremos brevemente conceptos generales sobre los *Sistemas*, para nuestro encuadre teórico sobre los SITR.

Un *Sistema* es un objeto compuesto, cuyos componentes se relacionan con al menos algún otro componente. Todos los *Sistemas* tienen composición, estructura y entorno. En este caso hablaremos de un *Sistema Informático* que, como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano (humanware/usuario) que permite almacenar y procesar información. Dentro de este sistema encontramos diferentes componentes. Hardware: todo lo que sea material físico "duro" y tangible que incluye computadoras, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. El software: se conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

Los componentes lógicos incluyen, entre muchos otros, las aplicaciones informáticas; tales como el procesador de texto, que permite al usuario realizar todas las tareas concernientes a la edición de textos; el software de sistema, tal como el sistema operativo, que, básicamente, permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una interfaz con el usuario.

Por último se encuentra el usuario que es el responsable de la puesta en marcha del funcionamiento del *Sistema* en sí mismo que dará como resultado el inicio del proceso interactivo.

Dentro del contexto de este trabajo debemos ampliar la aplicación en el uso de la palabra usuario: en general cuando pensamos en un usuario hablamos del que hace uso de tal o cual sistema. Esto es así pero, también, hablamos del creador

---

<sup>1</sup> Es profesor nacional de música, percusionista, maestro acompañante en danza contemporánea, compuso música para obras de danza, investigador en tecnologías aplicadas a la danza.

mismo del sistema interactivo que actúa como usuario. Pero lo más importante a tener en cuenta es al performer y al público dentro de la experiencia artística misma. Éstos son, en la actualidad, los protagonistas del arte interactivo. Cada vez más el público es parte del hecho artístico, participando activamente, modificando y alterando el desarrollo de la performance. Adquiere un rol activo, convirtiéndose en un eslabón más del *sistema*.

### Sistema Interactivo

Hardware      Software

### Intérprete/Público

Se denomina **sistema interactivo** a aquel sistema que se interrelaciona y depende de las acciones de un usuario para realizar una tarea, es decir, todo sistema en el que interactúan persona y máquina. Podríamos considerar interactivo desde un reproductor de DVD hasta un juego de computadora en el que nuestras acciones determinan el trascurso de la acción<sup>2</sup>.

Respecto de la interactividad, podríamos pensarla como **la demanda de acción que efectúa el sistema interactivo multimedia al usuario**. Esta acción/reacción puede tener distintos medios de expresión: tocar la pantalla, entrar texto en un determinado campo, realizar determinadas manipulaciones usando el mouse –click, doble click, arrastrar y tirar, etc., hasta acciones de mayor complejidad que involucren a otros sistemas relacionados entre sí. Desde el punto de vista del usuario, interactividad es la posibilidad de control que éste tiene sobre los objetos y contenidos del sistema, dependiendo del grado –nivel de complejidad- de interactividad que puede tener el sistema. La interactividad pensada como un diálogo entre un *Sistema* (uno o más softwares) y uno o varios usuarios.

Un *Sistema en Tiempo Real* (STR) es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, controlabilidad y alcanzabilidad.

Los sistemas en tiempo real están presentes en nuestra vida diaria, prácticamente en todo lo que nos rodea. La principal característica que distingue a

---

<sup>2</sup> Ver las categorías propuestas por Todd Winkler en (PAINE, 2002)

los STR de otros tipos de sistemas es el tiempo de interacción. Sin embargo, antes de continuar es necesario aclarar el significado de las palabras "tiempo" y "real". El tiempo es la magnitud física con la que medimos la duración o separación de acontecimientos sujetos a cambio, de los sistemas sujetos a observación. De tal manera que la palabra "tiempo" refiere a que el correcto funcionamiento de un sistema depende no sólo del resultado lógico que devuelve la computadora, sino también del tiempo en que se produce ese resultado. La palabra "real" quiere decir que la reacción de un sistema a eventos externos debe ocurrir durante su evolución. Como consecuencia, el tiempo del sistema (tiempo interno) debe ser medido usando la misma escala con que se mide el tiempo del ambiente controlado (tiempo externo).

La utilización e inclusión de la Tecnología en las artes escénicas no es algo nuevo. Se presenta como una solución y respuesta ante las necesidades artísticas de las personas.

Cuando pensamos en/la Tecnología -conjunto de conocimientos técnicos, ordenados científicamente, que permiten diseñar y crear bienes y servicios que facilitan la adaptación al medio ambiente-, tendemos a pensarla únicamente en términos actuales, es decir, en algo que es contemporáneo a nosotros, algo que sucede en *este* momento en el que vivimos. Esto es en parte cierto, pero no debemos olvidarnos del "pasado" de la Tecnología, ya sea remoto o de cuatro décadas atrás, ya que eso nos ayudará a tener una visión global y comprender el porqué de su importancia y su constante implementación.

Información digital, computadoras, animación en 3d, pantallas táctiles, hologramas, dispositivos de realidad virtual, robótica y cibernética, etc. son algunos de los desarrollos de las tecnologías actuales, pero no las únicas. El desarrollo tecnológico está presente desde el desarrollo mismo del hombre. Podríamos buscarlo en cualquier momento, en la época que quisiéramos, y encontrarnos con ejemplos donde las diversas tecnologías disponibles de ese momento, y el arte, se entrecruzan, retroalimentándose, dando origen a nuevas expresiones artísticas o variantes de la misma.

Por ejemplo podemos nombrar la utilización de máscaras en las representaciones de la tragedia griega; éstas poseían enormes dimensiones, no solo para que fueran mucho más visibles al público, sino que funcionaban como megáfonos aumentando la voz del actor. Tal vez un *sistema* mecánico, que permitía la elevación de un contrapeso, accionando de tal modo el desplazamiento de un decorado o parte de una pesada escenografía. En las artes del movimiento,

especialmente la danza, nos encontramos con algunos ejemplos que aporta la Doctora Ludmila Pimentel<sup>3</sup>.

*"... hay muchos ejemplos de coreografías de danza en las cuales fueran usadas las tecnologías, no solamente las digitales, para trascender las limitaciones biológicas del cuerpo, como por ejemplo la introducción de los zapatos de punta en 1832 en el ballet "La Sífide"... También se podrían citar los artefactos creados por Loie Fuller a finales del siglo XIX para multiplicar su cuerpo en múltiples imágenes a través de espejos, o incluso transformándolo en formas animadas de flores, orquídeas y serpentinas, a través de los efectos lumínicos y trajes especiales, que creó y patentó" (PIMENTEL, 2008)*

Es cierto que ahora ha cambiado la intencionalidad o el orden de origen de las ideas; muchas veces se antepone la tecnología al hecho artístico en sí mismo. Dispositivos o entornos tecnológicos que quieren convertirse en hechos artísticos. La necesidad de usar Tecnología dentro de un trabajo artístico simplemente porque es posible hacerlo, es divertido o está de moda...Creemos que el uso de la tecnología –más allá de la especificidad tecnológica misma- en trabajos de índole artística debe ser un medio para lograr los resultados esperados, o por lo menos que nos acerque de la mejor manera a nuestros objetivos como creadores. No debería ser un fin en sí mismo.

De la misma manera que los *sistemas* mecánicos evolucionaron en eléctricos, luego en electromecánicos, electrónicos y digitales, hay una constante evolución y perfeccionamiento en los SITR y los softwares asociados a éstos, que son muchos y variados, dependiendo de la función que cumplen, cómo se manejan y de cómo se organizan internamente.

Como ya dijimos, el software se refiere al equipamiento lógico o soporte lógico de una computadora digital, un conjunto de componentes lógicos necesarios que hacen posible la realización de tareas específicas. Algunos de éstos necesitan cierto grado de programación por parte del usuario para poder realizar la tarea deseada. Éstos le exigen al usuario un nivel mayor de conocimiento para poder manipularlos pero, también, le otorgan mayor flexibilidad y límites más amplios a la hora de crear, dependiendo de la habilidad e imaginación del programador. Otros softwares son más simples e intuitivos, donde parte de la programación ya está

---

<sup>3</sup> La Doctora Ludmila Pimentel es directora del Grupo de Investigación

<sup>3</sup>en Ciberdanza del Laboratorio de Investigaciones Avanzadas del Cuerpo (LaPAC) del Programa de Postgrado en Danza de la Universidad Federal de Bahía, Salvador, Bahía, Brasil; equipo que colabora con el INTAD (Equipo de Investigación en Tecnología Aplicada a la Danza, DAM, I.U.N.A)

resuelta y por esa misma razón son más cerrados o rígidos, con menos capacidades expansivas.

Antes de describir algunos softwares con capacidades interactivas, y para que resulten menos ajenos términos y funciones, hablaremos sobre cómo y de qué manera éstos se gestan, desenvuelven y trabajan. Desde ya que el tema es sumamente complejo y excede los límites de este trabajo, pero es necesario una breve aproximación.

Para eso hablaremos de dos formas o maneras de programación:

Programación orientada a "*objetos*" y Programación por "*Líneas de Código*"

La programación orientada a "*objetos*", usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos; un objeto se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. Estos objetos interactúan unos con otros, en contraposición a la visión tradicional en la cual un programa es una colección de subrutinas (funciones o procedimientos), o simplemente una lista de instrucciones para el computador. Cada objeto es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos de manera similar a un servicio. Los objetos tienen características fundamentales que nos permiten conocerlos mediante la observación, identificación y el estudio posterior de su comportamiento; estas características son: Identidad, Comportamiento y Estado.

Los objetos se interrelacionan entre sí creando objetos más complejos, formando así un *sistema*, que podrá unirse e interactuar con otro u otros objetos o sistemas dentro del mismo software como externos a éste. Algunos ejemplos de esta clase de programas son: ISADORA, MAX/MSP y PURE DATA, entre otros.

La programación por "*Líneas de Código*" (por ej. PROCESSING, OPEN FRAME WORKS) funciona a través de una determinada y específica sintaxis para el desarrollo y la creación de tareas y objetos. La programación se rige por reglas y un conjunto más o menos reducido de órdenes, expresiones, instrucciones y comandos que tienden a asemejarse a una lengua natural acotada (en inglés). Las características propias y capacidad de acción del nuevo elemento estarán dadas por el código mismo. Básicamente el *Código* es una serie de instrucciones que serán ejecutadas por una computadora cuando el código esté funcionando (andando). Está escrito en lenguaje de programación que, como todo lenguaje, es un contrato entre dos partes. Esas dos partes son el programador (programmer) y el compilador (compiler) -es el que escucha y entiende nuestro *código*. Un código puede ser algo tan simple como sumar dos valores, o algo sumamente complejo. Lo importante es escribir correctamente las instrucciones para nuestro *Compilador*.

Teniendo en cuenta estos conceptos generales sobre *Sistemas*, *Interactividad* y *Tiempo Real*, vamos a describir el software *Isadora*, que se encuadra dentro de los parámetros previamente descritos.

*Isadora* fue creado y desarrollado por el estadounidense Mark Coniglio, compositor musical y artista multimedial en el año 2001, especialmente para las performances de la compañía Troika Ranch.<sup>4</sup>

Desde sus inicios como artista e investigador estuvo interesado en la vinculación del arte con las tecnologías multimediales, desarrollando entre otros, un sistema inalámbrico que permitía a los intérpretes manipular el video, audio y luces en tiempo real, llamado Midi Dancer. Este sistema consistía en un traje especialmente diseñado con fibras plásticas que medía las flexiones y extensiones sobre las articulaciones del cuerpo. Cuando una articulación se flexionaba una señal era enviada a la computadora. Esa señal se transformaba en código permitiendo interactuar y manipular el video, audio y luces. Por ejemplo cuando el intérprete estiraba un brazo el sonido era suave y cuando lo flexionaba aumentaba la intensidad.

Este sistema llamado Midi Dancer lo condujo a su desarrollo más importante que fue *Isadora*.

*"El software Isadora es un entorno de programación gráfica (para Macintosh y Windows), que proporciona control interactivo sobre los medios digitales, con especial énfasis en la manipulación en tiempo real del video digital"*<sup>5</sup>.

Pero no solamente del video digital también de la música o cualquier "material" sonoro digital, imágenes, información MIDI, video cámaras, sistema físicos lumínicos y mecánicos, etc. Recopila la información de movimiento recibida por diferentes tipos de sensores: de movimiento, infrarrojos, de presencia etc., utilizando esos datos para manipular el video digital, sintetizadores de música, dispositivos de modulación de música, luces y piezas robóticas.

Es un programa relativamente nuevo -tiene diez años desde su primera versión- y a lo largo de este período ha ido evolucionando, y continúa, expandiéndose y mejorando sus posibilidades funcionales y operativas. Esto se debe no solo porque su creador continúa trabajando sobre *Isadora*, sino gracias al aporte que realizan los usuarios, como así mismo la incorporación al software de todos los desarrollos tecnológicos y el avance en las mejoras de los sistemas operativos y del hardware que ofrece el mercado. Es un programa comercial (no gratuito), pero se puede bajar e instalar y utilizarlo en modo "demo" con todas sus funciones excepto la de "salvar" el proyecto en un archivo.

---

<sup>4</sup> <http://www.troikaranch.org>

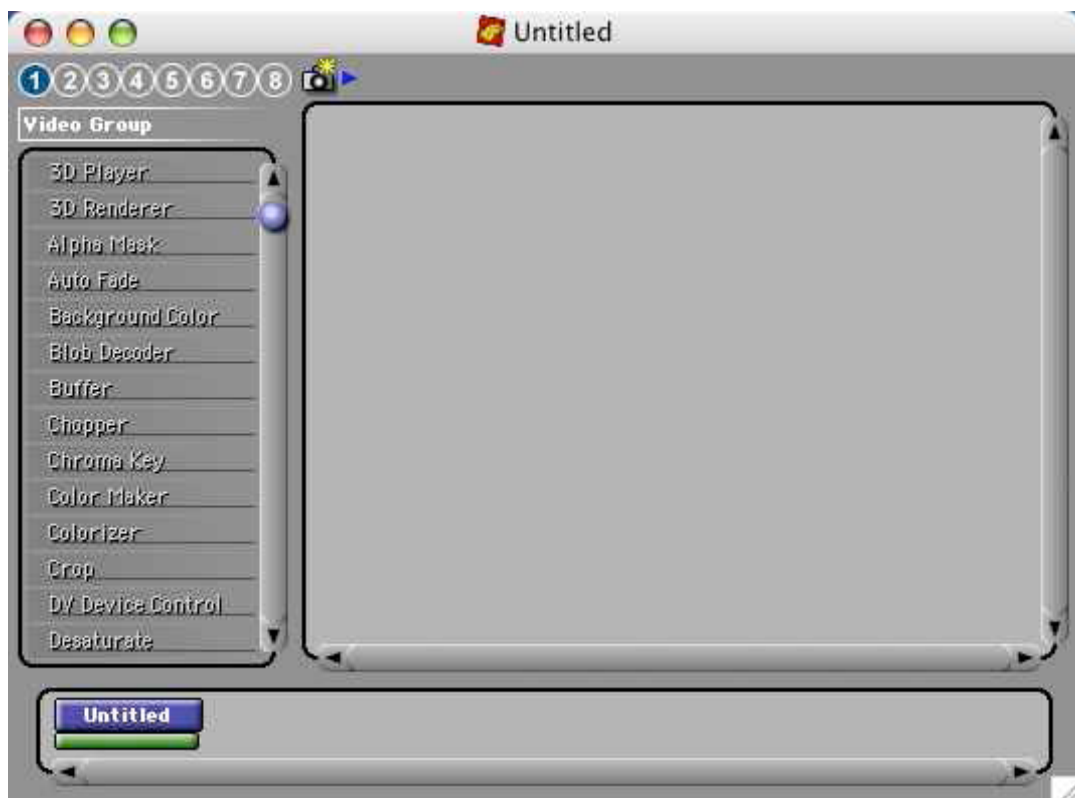
<sup>5</sup> <http://www.troikatronix.com/>

Para comprender cómo trabaja *Isadora* podríamos comenzar con el reconocimiento y descripción de su interfaz.

La interfaz gráfica es la que nos permite acceder e interactuar con el programa mismo. Actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles del programa. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema. Habitualmente las acciones se realizan mediante manipulación directa, para facilitar la interacción del usuario con la computadora. El diseño de la interfaz de *Isadora* está especialmente desarrollado para su utilización "en vivo".

Como ya dijimos *Isadora* es un entorno de programación gráfica donde sus herramientas de trabajo están divididas y separadas en categorías según sus funciones y atributos. Estas herramientas, o mejor dicho *objetos* (ya que *Isadora* está desarrollado como un entorno de programación orientada a objetos), se llaman Actores.

Los Actores son seleccionados y arrastrados dentro de un "espacio" llamado Editor de Escena (Scene Editor). En este "espacio" de trabajo o edición, los Actores



se conectan (literalmente se enganchan) entre sí, de una manera muy intuitiva.

En líneas generales todos los Actores tienen una "salida" (Out) que se conecta con la "entrada" (In) de otro Actor. De este modo los Actores se ensamblan

generando un "sistema" que puede llegar a ser muy complejo dependiendo, entre otras cosas, de la cantidad de Actores que se utilicen y demás variantes. Las conexiones pueden estar dadas en formas lineales, paralelas o ambas. También puede haber dos o más grupos de conexiones que no estén relacionados entre sí.

La decisión de nombrar a las herramientas/objetos **Actores**, y de nombrar al espacio o área de trabajo **Escena**, no es casual. Esta denominación nos da una clara idea de la importancia que tienen dentro del programa y también de la que tendrán en la performance, a la hora de utilizar este software en un trabajo artístico. Podemos intuir el por qué de la analogía con verdaderos actores y verdaderos escenarios de cualquier disciplina de las artes escénicas: en *Isadora* sus Actores poseen características similares tales como la respuesta en tiempo real e interactividad.

El diseño gráfico/visual de los Actores nos permita acceder fácilmente a sus componentes, funciones y parámetros, que podrán ser modificados según nuestro criterio y necesidad. Estas modificaciones se realizan cuando estamos armando nuestra Escena o durante la performance (tiempo real).

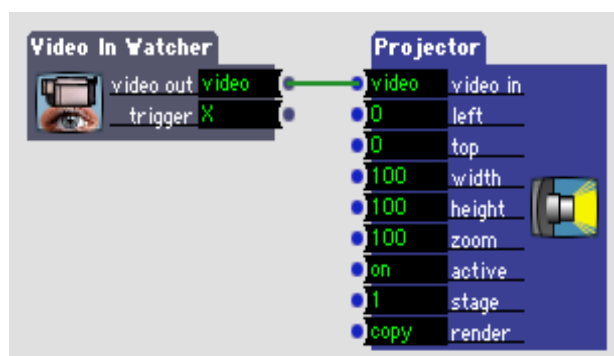
Para acceder a algunos de los trabajos de la compañía Troika Ranch donde utilizaron el *Isadora*: <http://www.troikaranch.org/vid-loopDiver.html>. Para acceder a algunos de los trabajos por parte de la Dra. Pimentel:

<http://iarchbodies.wordpress.com/>

Imaginemos una sencillísima performance dividida en dos momentos o "actos". Para el 1er momento deberemos crear la Escena 1 (ya veremos con qué Actores); para el 2º momento crearemos la Escena 2.

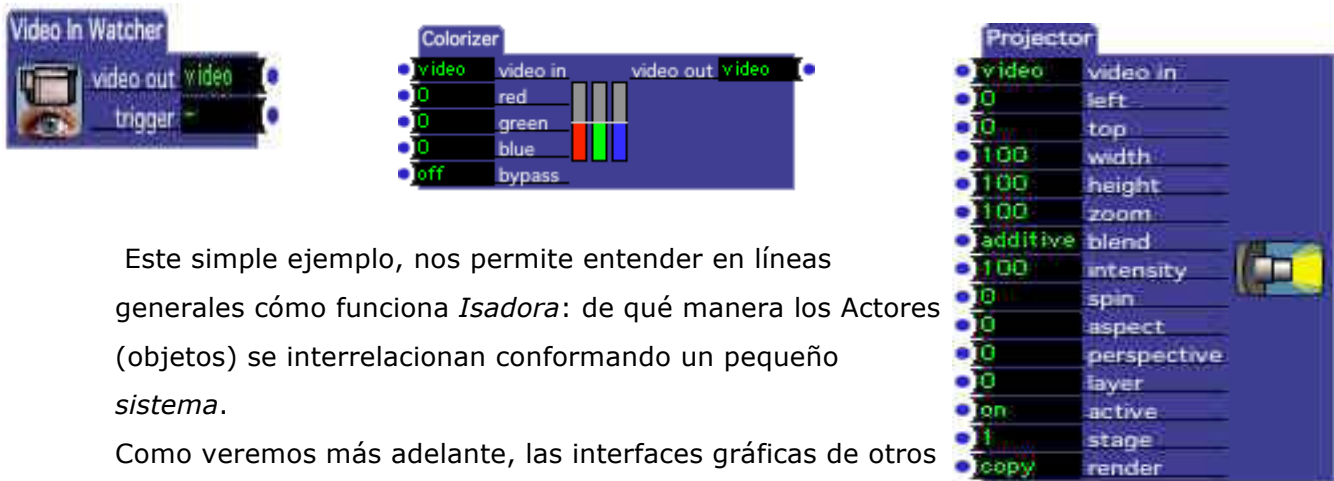
La Escena 1 consistirá de una proyección de un video. Para eso necesitaremos seleccionar y arrastrar dentro del Editor de Escena a un actor llamado Movie Player (reproductor de video) que es el que nos permitirá importar o "cargar" un archivo de video, y por otro lado un Actor llamado Projector (proyector) que es el que permitirá la visualización del video y posterior salida hacia un proyector externo.

La Escena 2 consistirá en la captura, en tiempo real, de la performance del intérprete, a través de una video cámara, que previamente manipulada y modificada por algún Actor, será proyectada. Para eso necesitaremos seleccionar un Actor llamado Video in Watcher (el que ve o registra la cámara de video, que está conectada a la computadora) y al Projector.





Entre ambos Actores incluiremos al Actor Colorizer (“coloreador”) que modificará los colores reales de la captura por los que nosotros le asignemos para luego ser proyectados.



Este simple ejemplo, nos permite entender en líneas generales cómo funciona *Isadora*: de qué manera los Actores (objetos) se interrelacionan conformando un pequeño *sistema*.

Como veremos más adelante, las interfaces gráficas de otros programas como Pd o Max/Msp son más simples o “despojadas” estéticamente, pero no por eso menos potentes en cuanto a prestaciones, todo lo contrario, son programas muy potentes ya que permiten la modificación y creación de nuevas acciones y objetos según las necesidades del programador, a diferencia de *Isadora*.

Otros Programas: Reseñas.

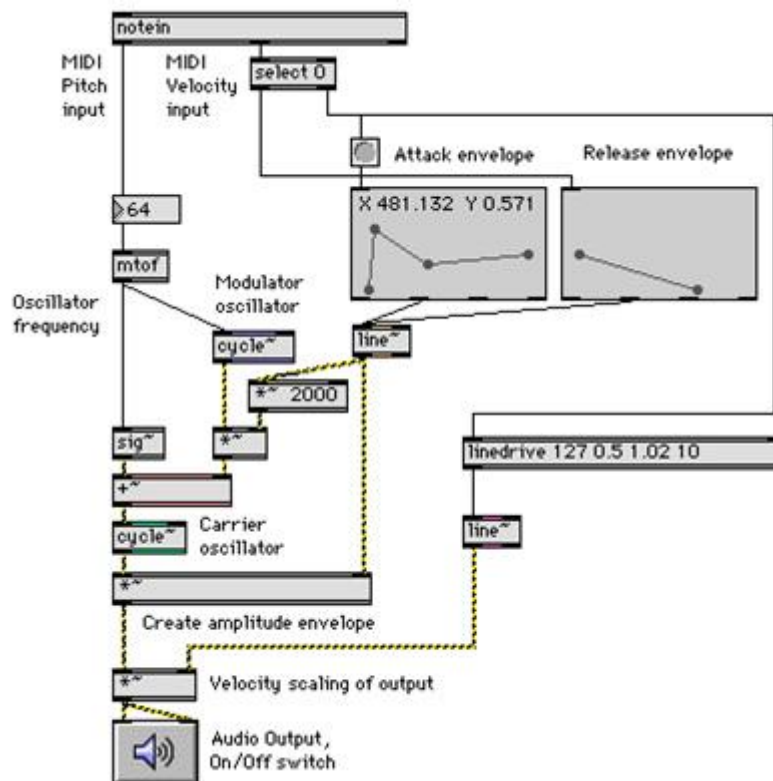
**MaxMSP** es un software gráfico desarrollado por Miller Puckette desde mediados de la década de los 80, utilizado para la creación de música electroacústica principalmente, pero que también tiene una extensión de procesamiento de imágenes llamada Jitter, y muchas capacidades para realizaciones interactivas y procesamiento en tiempo real. Es un programa comercial, mantenido por Cycling '74<sup>6</sup>, una empresa de programas situada en San Francisco.

En este programa también como en *Isadora* el usuario dispone de una superficie de trabajo en la cual dispone “objetos”, “mensajes”, “números” y otros elementos, que como son muy simples permiten una gran flexibilidad, pero también exigen un cierto entrenamiento. Es un software muy actualizado y muy documentado, además que está enriquecido por el trabajo de muchos programadores que han creado gran cantidad de aplicaciones para él. Muchas de estos desarrollos, llamados librerías externas, son libres y de código abierto.

---

<sup>6</sup> <http://cycling74.com/>

En la siguiente imagen vemos un patch de Max donde se ha diseñado un sintetizador muy elemental de frecuencia modulada



### Pd<sup>7</sup> (Pure Data)

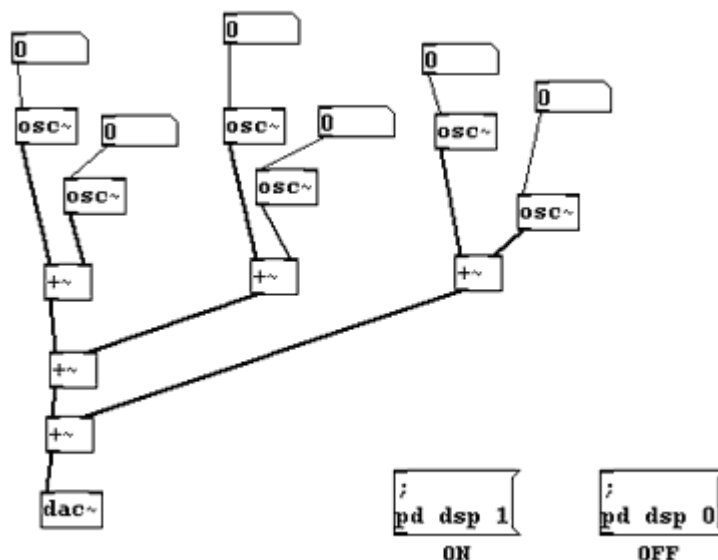
Es un lenguaje de programación gráfico desarrollado por Miller Puckette durante los años 90 para la creación de música computarizada interactiva y obras multimedia. Aunque Puckette es el principal autor del software, Pd es un proyecto de código abierto y tiene una gran base de desarrolladores trabajando en nuevas extensiones al programa. Es la tercera rama importante de la familia de lenguajes de programación conocidos como Max (MaxFTS, ISPW Max, MaxMSP, jMax, etc.).

Como Max, Pd tiene una base modular de código con externos u objetos que son utilizados como bloques de construcción para programas escritos en el software. Esto hace el programa arbitrariamente extensible y alienta a los desarrolladores a añadir sus propias rutinas de audio y control, escritas en distintos lenguajes. Sin embargo, Pd es un lenguaje de programación en sí mismo. Unidades de código modulares y reusables, escritas nativamente en Pd, llamadas "patches" o "abstracciones", son usadas como programas independientes y compartidas libremente entre la comunidad de usuarios de Pd. También posee una extensión para manejo de imagen llamada GEM, con la cual es posible crear y manipular

<sup>7</sup> <http://puredata.info/>

vídeo, gráficos e imágenes en tiempo real con aparentemente infinitas posibilidades de interactividad con audio, sensores externos, etc.

En la imagen siguiente vemos un ejemplo de un dispositivo de PD (Patch) de síntesis digital de sonido.



### Moldeo

Haremos referencia sobre un desarrollo informático hecho en argentina llamado Moldeo. Desarrollado originalmente por Fabián Kesler (Compositor licenciado en la UNQ, diseñador sonoro) y Fabricio Costa (programador y artista digital).

*"Moldeo es una plataforma de código fuente abierto que posibilita la realización de entornos interactivos con elementos de video y animación -en dos y tres dimensiones- y con efectos digitales en tiempo real".<sup>8</sup>*

La artista argentina Alejandra Ceriani<sup>9</sup> en colaboración con los responsables del proyecto Moldeo, ha creado la obra "Speak 3.0". Es quizá la obra más importante, hasta el momento, llevada a cabo con este software. Para esta obra se utiliza el software Moldeo como una plantilla de conversión a datos de los movimientos captados por la cámara web<sup>10</sup>. Entonces, el sonido de Speak 3.0 consta de la reinterpretación estética de estos datos más la música propiamente

<sup>8</sup> <http://www.moldeo.org>

<sup>9</sup> Alejandra Ceriani es graduada en la Facultad de Bellas Artes, Universidad Nacional de La Plata, Profesora y Licenciada en Artes Plásticas. Magíster en Estética y Teoría de las Artes. Investigadora categorizada y becada por la UNLP, para el estudio de las interrelaciones entre las disciplinas del cuerpo y el espacio escénico.

<sup>10</sup> <http://www.youtube.com/watch?v=XdTwtHgOKk>

generada por Fabián Kesler, sumado al diseño de fondos y plantillas de animación generados Fabricio Costa.<sup>11</sup>

### **EyeCon**

Es un software comercial desarrollado por Robert Wechsler y Frieder Weiss dentro del grupo Palindrome<sup>12</sup> (Palindrome Dance Company). Palindrome es un grupo performático que usa la tecnología de captura de movimiento (Motion Tracking technology) como eje principal para la concepción de sus investigaciones y trabajos artísticos. A partir del uso de video cámaras y sensores infrarrojos se convierte al espacio escénico en sí mismo sensible el movimiento. El sistema por el cual EyeCon detecta el movimiento está representado a través de estructuras gráficas llamadas "elementos".

La particularidad de este programa reside en poder "dibujar" esos elementos (que son puntos, líneas y figuras) dentro de su interfaz sobre la imagen que nos devuelve el monitor interno a partir de la captura de la cámara de video. Esas líneas y figuras sobre impuestas en la imagen que nos da el video, ocupan y delimitan un espacio virtual, que al ser traspasado o "tocado" por el intérprete generan un tipo de información. De esta manera el programa es "sensible" al movimiento del performer. El programa nos permite decidir qué tipo de información queremos generar cuando son activados los elementos. Esa información puede ser datos del protocolo MIDI u OSC y otros, pero también directamente EyeCon nos permite "disparar" imágenes, material sonoro, videos etc. cuando se activan los elementos.

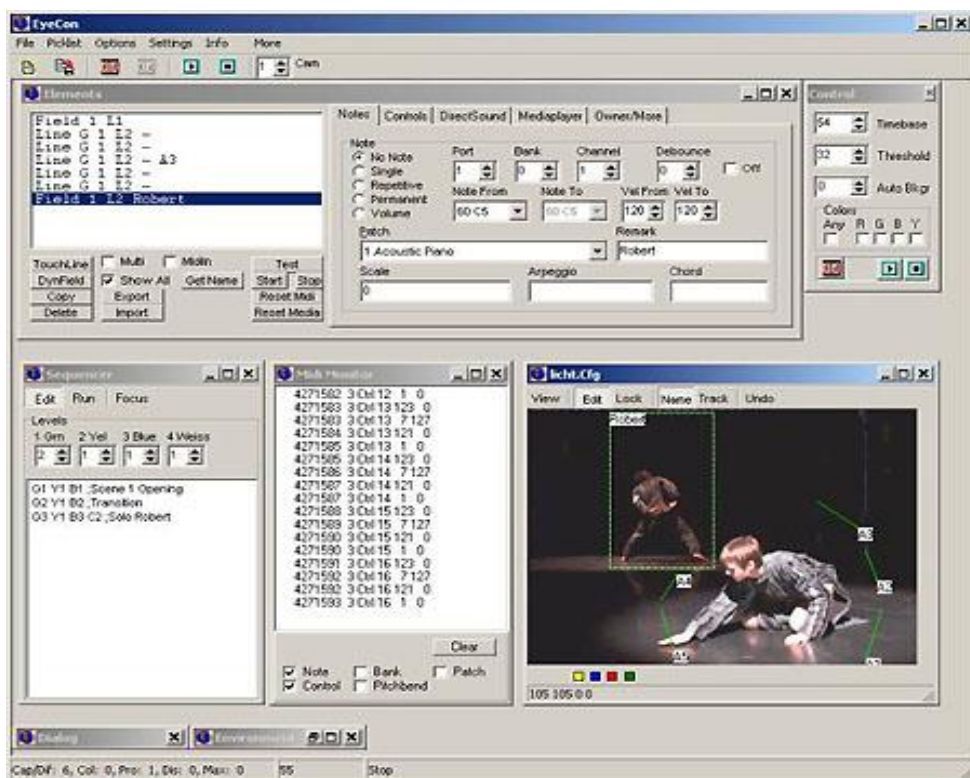
Algunos de los trabajos realizados con EyeCon pueden encontrarse en

<http://www.palindrome.de/>

---

<sup>11</sup> <http://www.alternivateatral.com/obra15578-speak-30>

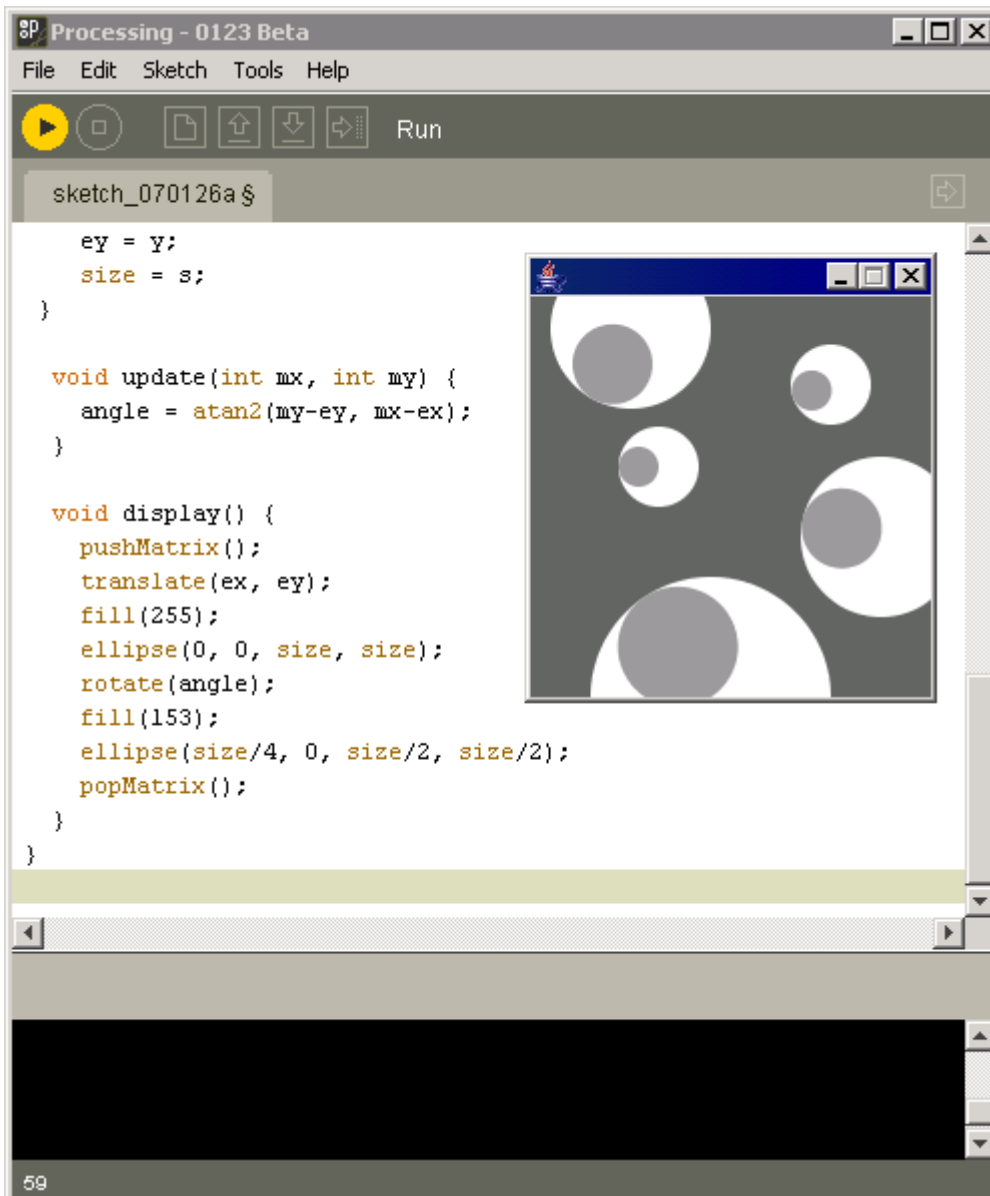
<sup>12</sup> <http://palindrome.de/>



En la imagen anterior podemos observar cómo se dibujan las líneas en el espacio, generando un campo virtual para el intérprete. Esas líneas son sensibles al tacto y responden ante la presencia u ausencia del intérprete u objetos. A diferencia de las líneas, las figuras actúan como un campo dinámico para ser usados como disparadores de sonido, control de volumen etc.; por ejemplo a mayor velocidad se mueve el intérprete mayor es el volumen.

**Processing**<sup>13</sup> es un proyecto desarrollado por Ben Fry (Broad Institute) y Casey Reas (UCLA Design | Media Arts), consiste en un entorno y lenguaje de programación de código abierto (basado en Java) orientado a la programación de imágenes, animaciones y sonido en tiempo real, y está diseñado especialmente para realizar aplicaciones multimedia en entornos web e instalaciones. Es un lenguaje de gran potencia y muy bien documentado que se está extendiendo a través de diferentes plataformas como Windows, Linux y Mac. En la siguiente imagen vemos la interface de Processing y un pequeño código y el resultado visual del mismo.

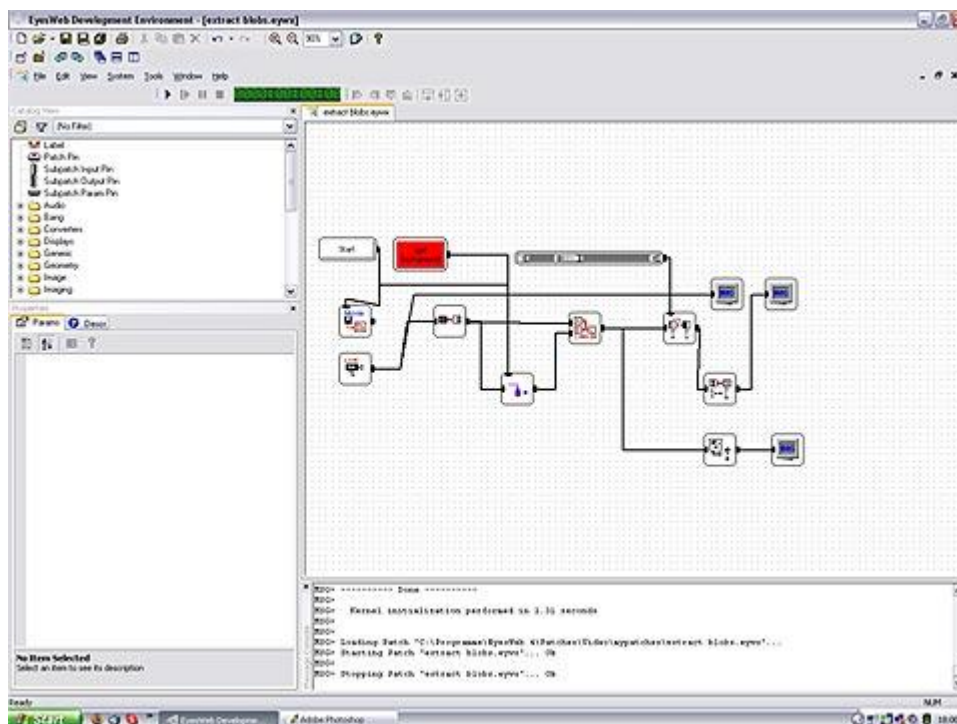
<sup>13</sup> <http://processing.org/>



**EyesWeb<sup>14</sup>**: EyesWeb es una plataforma abierta desarrollada por el Laboratorio de Informática Musical de la Universidad de Génova (<http://www.eyesweb.org/>). Se especializa en la captación de diferentes patrones de movimiento y gestualidad del cuerpo humano y la interpretación musical, y está orientado a la producción de sistemas multimedia interactivos para el análisis de movimientos escénicos en tiempo real y controlar la síntesis de sonido y la ejecución en vivo de instrumentos. En la siguiente imagen vemos cómo es la interface del EyesWeb y un ejemplo de un patch (código).

---

<sup>14</sup> <http://www.infomus.org/EywMain.html>



### **Eukinetic:**

Dentro de la escena latinoamericana encontramos el desarrollo del software Eukinetic-tech.

El proyecto se encuentra bajo la dirección, desarrollo de investigación y creación de la artista chilena Brisa MP<sup>15</sup>.

*“Eukinetic-tech es un proyecto de creación de un software prototipado para danza y performance diseñado bajo un sistema interactivo en tiempo real que involucra imagen, sonido y cuerpo en movimiento. Eukinetic-tech experimenta con herramientas de tecnología informática y electrónica (physicalcomputing) teniendo como base el estudio realizado por el coreógrafo Rudolf Laban en 1928, el que posteriormente repuso Ledeer sobre la “Eukinética”. El estudio de la Eukinética a partir de la propuesta de Ledeer, posibilitaría de manera simplificada una aproximación al cuerpo en movimiento desde una perspectiva científica tomando como estructura los tres factores del movimiento: “tiempo, espacio y energía”, con la idea de experimentar la posibilidad de mapear, analizar, anotar y traducir un movimiento de danza en tiempo real.”*<sup>16</sup>

Hemos realizado un recorrido por algunos de los programas más utilizados hoy en día. Un relevamiento por todos los softwares y sistemas interactivos sería imposible, además que escapa a las necesidades e intenciones de este trabajo.

<sup>15</sup> <http://caidalibre.cl/cv.htm>

<sup>16</sup> <http://caidalibre.cl/eukinetic.html>

Sería redundante decir que todos estos softwares están especialmente diseñados para un uso interactivo, pero tal vez no lo sea pensar el por qué. Diremos que en parte son una posible solución o respuesta a las nuevas necesidades comunicacionales de las personas y los artistas.

Los softwares con capacidades interactivas que se utilizan para conformar un Sistema en Tiempo Real (SITR) son muchos y cubren un amplio abanico de necesidades.

Difieren en qué cosas pueden hacer, o mejor dicho para qué fueron creados; además varían según el nivel de complejidad que demandan al usuario. También debemos tener en cuenta quién o quienes desarrollan tal o cual software. Según el propósito de sus creadores, encontraremos softwares gratuitos y de código abierto, permitiendo un acercamiento masivo de usuarios, y otros comerciales y cerrados.

Hay mucho por explorar y aprender, motivo por el cual es importante el apoyo por parte de las instituciones académicas en el fomento de las investigaciones sobre tecnologías aplicadas a las disciplinas escénicas y artísticas.

### **Bibliografía**

PAINE, G. (2002). *activatedspace.com/*. (C. U. Press, Ed.) Recuperado el 3 de octubre de 2011, de Organised Sound: [http://www.activatedspace.com/papers/files/interactivity-os7\\_3.pdf](http://www.activatedspace.com/papers/files/interactivity-os7_3.pdf)  
PIMENTEL, L. (2008). *EL CUERPO HÍBRIDO EN LA DANZA*. Recuperado el 5 de 10 de 2011, de UNIVERSIDAD POLITECNICA DE VALENCIA: <http://riunet.upv.es/handle/10251/3838>