

EDUCACION ARTISTICA Y PROGRAMACION.

Lenguajes de programación gráfica. Software Isadora.

Prof. Maximiliano Wille y Anibal Zorrilla.

La generalización del uso de la tecnología digital y su presencia en prácticamente todas las actividades humanas abarca desde hace tiempo las realizaciones artísticas, no sólo en disciplinas tradicionales como la música o el cine sino también creando las nuevas variantes de arte tecnológico, performático o no. Esta realidad se refleja en la educación artística hoy que no es ajena al proceso de tecnificación que se está viviendo. Sin embargo, es evidente la manifestación de distintas trabas y dificultades al momento de aplicar recursos informáticos en el campo de la formación de los estudiantes.

Las falencias y dificultades que aparecen al momento de realizar la labor de formación artística-tecnológica tiene muchos aspectos y un análisis exhaustivo de los mismos no es el objetivo de este escrito, pero podemos mencionar simplemente dos situaciones que ejemplifican el tipo de problemas que se presentan. Por un lado existen deficiencias y carencias estructurales que en gran medida se originan en dificultades económicas: falta de equipamiento básico como las mismas computadoras, proyectores, equipos de audio y red de internet entre otras. Es cierto que los planes estatales nacionales y locales están revirtiendo en parte este problema y constituyen un gran avance. En segundo lugar aparece la dificultad que plantea a los docentes, por diversos motivos, el empleo de los recursos tecnológicos disponibles y el establecimiento de nuevos canales y estrategias didácticas basadas en éstos. Ya sea por falta de conocimiento, estímulo o interés, introducir nuevos o no tan nuevos dispositivos tecnológicos dentro del ámbito educativo es todavía un desafío.

Alfabetización digital.

De todas maneras hay una dificultad que es de distinto nivel y que tiene significado, origen y problemáticas diferentes. Es el del concepto de alfabetización digital y de su desarrollo o ampliación de acuerdo al avance de las tecnologías digitales, y del uso que se hace de éstas.

A pesar de la existencia de planes estatales para la inclusión informática, “... los jóvenes haciendo uso del Chat, navegando en distintas páginas de Internet no adecuadas a su nivel educativo o utilizando ciertos programas multimedia, pensamos erróneamente que tienen amplios conocimientos y habilidades en esta área, cuando están usando solo un 10% de lo que ofrece la herramienta informática”. (Sandra Spitzer, 2012).

Esto implica no sólo un desaprovechamiento de las posibilidades de la tecnología, sino que también una falta de adecuación a las condiciones que exige una adecuada alfabetización digital. El concepto de alfabetización es dinámico en general, pero mucho más en relación a la tecnología cuyos tiempos de renovación y modos de penetración en todos los ámbitos humanos están en una franca aceleración sin que se pueda prever otra situación.

Además la mayor parte de los contenidos y de las habilidades que las personas adquieren sobre la tecnología están en su mayor parte a su función como usuario de aplicaciones dirigidas al ámbito de la producción, la administración y la provisión de servicios, como procesadores de textos, bases de datos, planillas de cálculo entre otras. Y los otros usos en general consisten en navegar en las redes sociales, emplear buscadores para conseguir información y ver videos y escuchar música.

Naturalmente la importancia de estos aspectos del empleo de la tecnología es muy grande y sus beneficios son importantes, pero hay otros que están cobrando una importancia creciente y sobrepasan estos límites, entre éstos la habilidad para programar.

Por lo que esta situación podría traer como consecuencia una doble insuficiencia: tanto en la formación en campos que en poco tiempo van a ser mucho más importantes para la inclusión del sujeto en el medio y la otra en la incorporación de la herramienta informática y el mundo digital por el sujeto desde una perspectiva meramente utilitaria relacionada con el mercado de trabajo.

En estas definiciones existen dos perspectivas: la de asociar al sujeto como una pieza más del ordenador, cuya conducta y por lo tanto su subjetividad está adaptada y programada de acuerdo a las necesidades del dispositivo, o como un sujeto creador cuya relación con este dispositivo sea no sólo la de un operador sino la de un participante activo en su conformación y empleo.

En el campo de la actividad artística ocurre el mismo fenómeno. La mayor parte de la utilización de la tecnología está dirigida a editores de audio, video y procesamiento de imágenes. Esto tiene como razón fundamental el enorme abaratamiento de los costos de las producciones artísticas debido al fácil acceso a estos softwares y a la digitalización de la mayor parte de los medios audiovisuales que se producen y difunden. Sin embargo colocan al artista en una posición similar en relación al uso de la tecnología a la que tiene un administrador que usa una planilla de cálculo.

En ese horizonte la programación utilizando lenguajes de programación gráfica orientados a la actividad artística tiene mucho que aportar en el campo educativo y social, como una herramienta crítica y creativa para considerar a las tecnologías como formas culturales. Estos softwares permiten otras formas de apropiación de la herramienta tecnológica mucho más activa por parte del creador y no lo limita al rol del usuario, similar al de una pieza de una cadena de producción. Por medio de ellos puede incorporar a su creación una herramienta informática adaptada a sus necesidades artísticas, además de abrir la posibilidad de la existencia de nuevas disciplinas como el arte interactivo, que si bien existía desde antes tiene en este momento un desarrollo y unas posibilidades que lo hacen cualitativamente diferentes.

Programas y programación.

Cuando se piensa en programación, programas y programadores, generalmente se los concibe como actividades que pueden ser realizadas únicamente por profesionales altamente especializados y capacitados, alejando por lo tanto a muchas personas de la posibilidad siquiera de intentar conocer ese mundo.

Es cierto que parte de este supuesto tiene sustento. Saber programar significa, entre muchas otras cosas, conocer y dominar un lenguaje, y luego saber aplicarlo. Es un lenguaje artificial que tiene reglas y principios, que es necesario conocer y entender. No sólo eso sino que del mismo modo que existe una inmensa variedad de idiomas que se hablan en el mundo, son también diversos y variados los lenguajes de programación. Cada lenguaje posee específicas reglas y principios, pero al igual que los idiomas, generalmente un lenguaje de programación tiene a otro como antecesor, de manera tal que compartirán muchísimos elementos en común. A su vez, no todos tendrán las mismas funciones y/o campos de aplicación específicos, y tampoco la misma complejidad para aprenderlos. Es por este motivo que ciertos tipos de lenguajes son más propicios para la iniciación en el campo de la programación. Uno de estas clases de softwares son los lenguajes de programación gráfica (en inglés VPL, por "visual programming language"), que por sus características son muy adecuados para el acercamiento a la experiencia de la programación.

Un lenguaje de programación es un "lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar". Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. El lenguaje utilizado por el procesador se denomina "lenguaje máquina". Se trata de datos que llegan al procesador de tal manera que éste los puede interpretar, y consiste en la serie de ceros y unos del código binario. El lenguaje máquina, por lo tanto, no es comprensible para los seres humanos, razón por la cual se emplean lenguajes intermediarios comprensibles para el hombre. El código escrito en este tipo de lenguaje superior se transforma en código máquina para que el procesador pueda procesarlo.

El ensamblador fue el primer lenguaje de programación utilizado, por lo que es muy similar al lenguaje máquina, con la diferencia de que los desarrolladores pueden comprenderlo. No obstante, este lenguaje se parece tanto al lenguaje máquina que depende estrictamente del tipo de procesador utilizado, de manera que cada tipo de procesador debe tener su propio lenguaje ensamblador. Así, un programa desarrollado para un equipo no puede ser portado a otro tipo de equipo, entonces para poder utilizar un software escrito en un código ensamblador en otro tipo de equipo, será necesario volver a escribir todo el programa.

Para enfrentar esta dificultad aparecieron los programas de alto nivel, en los cuales una instrucción corresponde a muchas instrucciones del programa ensamblador, y también estas instrucciones pueden ser interpretadas por ensambladores correspondientes a distintos procesadores. El término "portabilidad" describe esta capacidad de usar un programa de software en diferentes tipos de equipos, algo común en las máquinas de hoy.

Para que los ordenadores puedan cumplir las tareas para las cuales se utilizan es necesario disponer de programas ejecutables. El programador escribe una secuencia de instrucciones siguiendo las reglas de un lenguaje de programación y esa secuencia de instrucciones se guarda en uno o varios archivos que constituyen el código fuente del programa. Mediante otro programa especial llamado compilador o intérprete el ordenador convierte los archivos de texto en archivos ejecutables, es decir traduce la secuencia de instrucciones escritas por el programador en instrucciones, transformando un conjunto de archivos de código fuente en programas ejecutables. Por ejemplo, un lenguaje de programación puede tener una instrucción para calcular raíces cuadradas, entonces el compilador o intérprete tiene que ser capaz de convertir la instrucción que componen el algoritmo de la raíz cuadrada que

hubiera escrito el programador en una serie de pasos que acaben dando el resultado deseado.

Normalmente se utiliza el término programa para referirse a las dos cosas:
al archivo de texto escrito por un programador en algún lenguaje de programación
y al archivo ejecutable que ejecuta el ordenador para producir un resultado.

Eso puede causar alguna confusión, pero en general el uso de la palabra, a pesar de la ambigüedad de su significado, permite determinar mediante el contexto de cuál de los dos conceptos se está hablando.

Los ordenadores siguen al pie de la letra las instrucciones de los programas ejecutables. Por tanto, los programas deben tener en cuenta todos los posibles resultados de todas las instrucciones e incluir las instrucciones a seguir en cada caso, de manera que el ordenador no se quede atascado en algún punto del programa. En el ejemplo citado de las raíces cuadradas, el programa tiene que tener en cuenta que no se pueden calcular las raíces cuadradas de números negativos (sin recurrir a los números complejos) y no tiene que pedirle al ordenador que lo haga porque seguramente se producirá un error y el programa se interrumpirá.

Los lenguajes de programación gráfica.

Un lenguaje de programación gráfica o visual (VPL) es cualquier lenguaje de programación que permite a los usuarios crear programas manipulando sus elementos a través de representaciones gráficas en lugar de tener que especificarlos textualmente. Un VPL permite programar con expresiones visuales, empleando textos y símbolos gráficos como elementos de la sintaxis o nota secundaria. Por ejemplo, muchos VPL están basados en la idea de “cajas y flechas”, donde las cajas u otros objetos de pantalla son tratados como entidades, relacionadas por flechas, líneas o arcos que representan relaciones. Otros presentan una interface gráfica compuesta de bloques de distintos tipos que se arman como rompecabezas construyendo grupos a través de los cuales circulan las instrucciones. También hay programas específicos con símbolos de circuitos electrónicos, matemáticos u orientados a la robótica.

En el ámbito educativo existe un interés creciente en desarrollar herramientas y lenguajes que mediante el juego faciliten un aprendizaje de las bases de la programación. Uno de los que ya alcanza mayor tradición es Lego Mindstorms (1998) un juego de robótica que además de las tradicionales piezas de esta marca incluye un pack con sensores de tacto, luz y sonido y un motor eléctrico. Todo este pack permite la construcción de robots complejos y programar las acciones que debe realizar. El lenguaje de programación (ROBOTC) que utiliza el juego es gráfico, basado en bloques que se van sumando para construir las instrucciones. Tiene sus orígenes en Logo, un lenguaje de programación ideado por Seymour Papert en su libro “MindStorms: Children, Computers, and Powerful Ideas” y muy usado como herramienta educativa. La estructura del lenguaje ROBOTC se basa en la unión de bloques que contienen acciones y así se van construyendo las instrucciones que ha de ejecutar cada parte del robot. Este producto ha acercado la creación de robots tanto al público generalista sin conocimientos de programación como a los expertos, puesto que los robots permiten la recepción de instrucciones escritas en lenguajes más complejos permitiendo que los jugadores avancen en su aprendizaje.

Otra herramienta es el LittleBits, un juguete basado en la misma filosofía: un módulo que permite programar múltiples acciones a partir de la interacción física entre los distintos bloques. LittleBits fomenta el aprendizaje de la programación experimentándola a partir de gestos simples e intuitivos, y permite a los niños entender el funcionamiento de los dispositivos que les rodean. En esta misma línea de trabajo, se sitúa la placa MaKey MaKey. Bajo licencia openhardware y diseñada por Jay Silver y Eric Rosenbaum del MIT MediaLab, permite ejecutar cualquier programa instalado en un ordenador o en internet, mediante cualquier superficie imaginable, como frutas, partes del piso, telas y muebles, entre otros.

Uno de los lenguajes de programación orientados al aprendizaje para niños más populares es Scratch, de distribución gratuita, del MediaLab del MIT, incluido entre otros en los equipos que proporciona el programa “Conectar Igualdad”. Un desarrollo complementario al que se puede acceder on-line es Snap!, de la Universidad de Berkeley. En la misma línea Google ha presentado recientemente su propio lenguaje para el aprendizaje. Se trata de Blockly, un editor de programación gráfica on-line que permite construir aplicaciones mediante bloques. Una de las grandes ventajas que ofrece esta última herramienta es que permite exportar a lenguajes más complejos como JavaScript, Dart, Python y XML.

Programar para bailar. Software Isadora. Experiencias educativas.

El Equipo de Investigación en Tecnología Aplicada a la Danza, InTAD, desde hace cinco años lleva adelante experiencias no solo educativas y de formación sino artísticas. Para lo cual se apoya en las tecnologías multimedias interactivas. Uno de los softwares que utilizamos es Isadora, que está desarrollado íntegramente sobre las bases del lenguaje de programación gráfico.

El software Isadora es un lenguaje de programación gráfica basado en la idea de cajas y cables, que tiene su origen en el software Max escrito por Miller Puckette en los años 80, del cual derivaron Max/Msp, Pure Data y muchos otros. Si bien tiene herramientas de procesamiento de sonido está fuertemente orientado hacia el procesamiento de imagen en tiempo real y a la interacción con el medio o el performer. Se caracteriza por tener una interface gráfica muy intuitiva, muy adecuada para su empleo por artistas que no tienen formación en informática, matemáticas o ciencia. Es muy potente ya que tiene partes del programa bastante importantes representadas por gráficos llamados “actores” que se colocan en un espacio de trabajo llamado “escena” y se conectan entre sí mediante cables para configurar los flujos de datos. Permite lograr resultados impactantes con poco trabajo de programación, por lo cual ayuda a romper la desconfianza y la lejanía que en general tiene un bailarín o un actor hacia la tecnología digital.

Es un software propietario que requiere comprar una licencia para acceder a guardar los trabajos o patches. De todas maneras muchos artistas lo usan en modo demo, que permite de todas maneras su uso. Existen otros desarrollos que tienen algunas de las funciones de Isadora, como el ARGOPd programado en Pure Data, pero su empleo es bastante más complejo y si bien tiene una potencia y posibilidades similares necesita un aprendizaje mucho mayor. Una de las posibilidades del uso de Isadora es permitir la práctica de la programación gráfica en un entorno amigable y adiestrar así al programador en sus procedimientos antes de abordar otros softwares.

Este equipo de investigación realizó actividades de vinculación con las escuelas de danza de nivel medio de la Ciudad de Buenos Aires y a través del Conectar/Lab del programa Conectar Igualdad, introduciendo el software Isadora como herramienta específica de aprendizaje y experimentación para las artes del movimiento, a través del uso de las netbooks que poseen los alumnos. Estas experiencias tienen dos objetivos principales teniendo en cuenta que participan tanto alumnos como docentes, y teniendo en cuenta los problemas planteados al principio de este trabajo. En primer lugar acercar a los docentes y alumnos esta herramienta para la programación artística, difundiendo sus características y los principales conceptos para su manejo. Y en segundo lugar potenciar y estimular el uso de las netbooks y el vínculo entre los alumnos y éstas, cambiando el uso de la herramienta de una forma utilitaria a otra más participativa y creativa, modificando la posición del alumno frente a ella y acercando nuevos horizontes de inclusión y oportunidad.

Referencias:

Lego Mindstorms:

<http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>.

LittleBits: <http://littlebits.cc/>

MaKey MaKey: <http://www.makeymakey.com/>

Scratch: <http://scratch.mit.edu/>

Snap!: <http://snap.berkeley.edu/>

Bloclly: <https://code.google.com/p/blockly/>

InTAD: <http://intad1.wordpress.com/>

Isadora: <http://troikatronix.com/>

ARGOPd: <http://paresys.pagesperso-orange.fr/ARGOPd/index.html>

Pure Data: <http://puredata.info/>

Max/Msp: <http://cycling74.com/>

Spitzer, Sandra. *LA INFORMATICA -INCLUSION Y RETENCION EDUCATIVA-*; Sitech Sudeste, Sindicato De Trabajadores De La Educación del Chaco. <http://es.scribd.com/doc/170116288/La-Informatica-Inclusion-y-Retencion-Educativa>, Resistencia, 2012