

LA PROGRAMACIÓN DE VIDEOJUEGOS CON PYTHON COMO ESTRATEGIA PARA UNA ALFABETIZACIÓN DIGITAL INTEGRAL

Francisco Palm

Universidad de Los Andes, Av. Las Américas, Núcleo Liria, FACES, IEAC

E-mail: fpalm@ula.ve,

Palabras clave

Aprendizaje, Conocimiento Libre, Videojuegos, Python, Soberanía Tecnológica

Resumen

En este documento se enumeran algunos factores estructurales que dificultan la construcción de la soberanía tecnológica venezolana, y se desemboca en el problema fundamental del cuidado del cuidado del conocimiento como bien público, entendido bien público en el sentido utilizado por Fuenmayor (2002). A continuación, se presenta el esbozo de una propuesta de enseñanza masiva de la programación usando como elemento motivador los videojuegos y como principal herramienta el lenguaje de programación Python, esto con la finalidad de coadyuvar una alfabetización digital y multidimensional de los ciudadanos, que habilite su incorporación en procesos de aprendizaje que buscan dar cuenta de las destrezas básicas como el procesamiento de información y la generación de nuevo conocimiento en entornos de profusa interacción social presencial o telemática. Esta propuesta busca también sentar las bases para construir un tejido social endógenamente desplegado que responda a la amenaza global de los modos de producción, los patrones culturales y el poder mediático que imponen las corporaciones transnacionales. La visión de este proyecto es iniciar la discusión hacia la construcción de organizaciones sociales basadas en conocimiento libre de las que puedan emerger sinergias emancipadoras.

1 Introducción

1.1 Factores de fracaso en la construcción de la soberanía tecnológica.

Después de casi 5 años desde el Decreto 3390 que establece el uso prioritario del software libre en la Administración Pública Venezolana encontramos resultados bastante modestos, y una presencia casi nula del software libre en la escena pública nacional. Si bien existen éxitos y fracasos puntuales que generalmente distraen la atención, no dan cuenta de cuestiones mucho más profundas:

- Ausencia hasta los momentos de una relación coherente entre los contenidos de la enseñanza y el proyecto de país en los términos establecidos por Varsavsky (2006). En su lugar se ha constituido una política tecnológica desarrollista que mide el progreso en términos de imitar a los países que se consideran avanzados.
- Patrón cultural globalizado que desliga la ciencia y la tecnología de la realidad cotidiana de la mayoría de las personas y los pone a cargo de elites tecnocientíficas. Por ejemplo, es común que como primeros

ejemplos de ciencia se habla a los niños y niñas de cuestiones como los Dinosaurios y la Astronomía que no pueden constatarse de forma directa.

- Adoctrinamiento tecnológico orientado únicamente al **uso** de tecnología "avanzada", que delega la **creación** de ciencia y tecnología en manos de la elite tecnocientífica mencionada antes. No hay preocupación por enseñar los principios básicos de tecnología de consumo masivo, y se ha permitido que por intereses particulares se privatice el conocimiento de la misma.
- Des-conocimiento y des-reconocimiento del Estado como "arbitro" en la implantación de las políticas en ciencia y tecnología. Esto debido en parte a la pérdida progresiva de credibilidad de los entes encargados de esta tarea al no cubrir con las expectativas ni requerimientos básicos de la sociedad venezolana. Tampoco ha funcionado dejar que los órganos de la administración pública tomen sus propias decisiones, ni tampoco esperar que las universidades ofrezcan esas respuestas tal y cómo ha ocurrido con la Misión Ciencia.

Podría resumirse, tal como lo establece Fuenmayor (2002), en el devenir de una sociedad que no solamente ha sufrido una sistemática enajenación del **bien público**, sino que incluso ha perdido la condición de posibilidad para que existan bienes públicos. En el marco de este documento, y en la realidad actual de la sociedad venezolana, se considera al conocimiento sobre las tecnologías de información como un bien público fundamental, precisamente como condición generatriz de nuevas dinámicas de aprendizaje transversales a un amplio abanico de áreas del saber.

1.2 Elementos básicos de una estrategia educativa

Entonces, cuando se hace referencia en el título a *alfabetización digital integral*, se quiere denotar un proceso de *adquisición de capacidades para la ciudadanía digital* que pueda coadyuvar a la *existencia y cuidado de bienes públicos*, tanto en el propio ámbito del conocimiento de las tecnologías de información como en las posibilidades que estas ofrecen.

Con frecuencia, el estudio de las ciencias de la computación se considera una capacitación reservada para algunas carreras específicas de la Educación Superior. Este enfoque, frecuente también en otras áreas de la ciencia, conduce a la creación de condiciones en la conformación de una elite tecnológica endógena, desde el punto de vista del diseño de una política tecnológica nacional este tipo de soluciones es sumamente frágil en vista de:

- La recurrente problemática de la "fuga de cerebros" en la que parte de las personas beneficiarias de las políticas de ayuda a la formación pasan a engrosar a las elites tecnológicas de los países más desarrollados.
- Las elites tecnocientíficas pasan a ser un grupo con intereses particulares y con frecuencia con una agenda propia que con frecuencia no coincide con los intereses de la nación.

- Las debilidades institucionales propias de un sector de ciencia y tecnología sustentado durante muchos años en el clientelismo y el rentismo, en cuyo caso resulta mucho más sencillo comprar que hacer.

En este marco de ideas, en este trabajo se intenta hacer un aporte en base a una **estrategia de formación masiva de los primeros niveles de educación formal** que permita un acercamiento más intenso y profundo de los niños y niñas hacia la ciencia y la tecnología desde la educación primaria. Esta estrategia trata de dar cuenta de los factores indicados arriba, a saber:

- Establecer procesos de aprendizaje desde una aproximación que implique una re-creación constante y progresiva, **evitando el rol de consumidores de tecnología**, y particularmente de todo conocimiento fundamentado en "cajas negras" constituido por conocimiento privativo por causa de patentes y licenciamientos restrictivos.
- Hacer **énfasis en las tecnologías de información**, dado el enorme impacto transversal que tiene sobre un gran número de áreas de conocimiento. En este sentido, la re-creación en el ámbito de las tecnologías de la información implica entender conceptos básicos de manejo de la información y algoritmia para la resolución de problemas.
- Debe constituirse en una alternativa con un gran atractivo para los niños y niñas, en virtud del impactante bombardeo de los medios de comunicación masiva (donde está incluido Internet). Muchos estudios muestran como los niños pasan más tiempo frente a la pantalla del TV o la computadora personal que en las aulas de clase. Y una parte importante de este tiempo se dedica a los videojuegos. Así, **diseñar estrategias educativas alrededor de los videojuegos** puede resultar muy atractivo, y puede asegurar un impacto significativo en los niños.
- El **uso exclusivo de software libre y recursos bajo licenciamiento libre** que por un lado potencian en gran medida los procesos de aprendizaje al no imponer ninguna restricción sobre el estudio y modificación de todas las herramientas tecnológicas. Y por el otro lado forma a los estudiantes bajo las premisas de un entorno donde se concibe al conocimiento como un bien público
- En el marco de la introducción de proyectos como "Canaima Educativo" va a resultar necesario elaborar contenidos y actividades que estén a la altura de las necesidades y exigencias de niñas, niños y jóvenes venezolanos. Precisamente la enseñanza de la programación y de todas las áreas de conocimiento involucradas en el desarrollo de videojuegos va a permitir un marco de referencia común para la enseñanza de la informática, lo cual será un aporte muy significativo para **fortalecer la imagen de las instituciones que actúan como árbitros en política de ciencia y tecnología**.

2 Diseño de una estrategia masiva de educación basada en la programación de videojuegos

2.1 El desarrollo de software como elemento fundamental de la alfabetización digital

La programación es vista con frecuencia como un campo de ejercicio profesional para personal altamente capacitado, y en muchos casos lo es, pero lo cierto es que existen tantos grados de pericia y especialización como los hay en casi todas las actividades humanas. Por ejemplo, no todas las personas realizan ejercicio físico con la finalidad de ser atletas de alta competencia.. Sin embargo, es de común acuerdo que es muy beneficioso tanto para los individuos como la sociedad en su conjunto la difusión amplia de la práctica de ejercicio físico. La programación es una destreza fundamental para el funcionamiento de la sociedad actual, y en lo individual puede ser una forma de ejercicio mental tan beneficioso como lo es el ejercicio físico.

En la tesis "Promoviendo la Alfabetización Digital mediante la Programación en Python"¹ (Miller, 2006) argumenta que haciendo una analogía entre la alfabetización digital y la alfabetización a secas, usar las computadoras sin saber programar es como leer sin saber escribir. Usar la computadora implica interpretar la simbología que ofrece su interfaz (leer), y la programación requiere el ejercicio de articular símbolos para crear un artefacto cognitivo que pueda ser interpretado por otros usuarios (escribir).

Este asunto es sumamente importante porque al fin y al cabo *la alfabetización lo que se busca es habilitar a la persona para aprender*. Hay muchos modos de aprendizaje que no dependen de la lectura y la escritura, pero el acceso a una buena parte del conocimiento humano acumulado en la historia depende de estas destrezas básicas. De igual modo, en el contexto de la sociedad actual la alfabetización digital es crucial para entender de modo individual y colectivo el sentido de las nuevas tecnologías de información y comunicación en nuestras vidas, lo cual tiene una importancia trascendental en la construcción de modelos alternativos de sociedad para superar la divisoria digital planetaria (Castells, 2001). Sin conocimientos de programación, aunque sean básicos, nuestro conocimiento sobre las tecnologías de información siempre será incompleto. Y la programación es tan solo una parte de un ámbito mucho más amplio como lo es el *desarrollo de software*, que tiene muchas implicaciones organizativas y económicas, y por ende sociológicas y culturales. También hay una tendencia de hablar de desarrollo de software trivializando la programación, y llevándolo a un terreno exclusivo de la gestión administrativa, este sería un error tan dramático como el caso de un editor de un medio impreso que pasara por alto "detalles" como redacción y ortografía.

Cuando se reserva el estudio de la programación a la educación superior al parecer existe una confusión entre las necesidades de formación para cubrir las necesidades de personal de las empresas con las necesidades educativas en función de un proyecto de país, y esto ocurre porque las instituciones educativas publicas y privadas parecen llamadas a dar respuesta a las necesidades de las empresas, lo que en el marco del neoliberalismo parece ser una contribución indirecta a la sociedad. También muestra la lentitud y falta de flexibilidad de las instituciones públicas venezolanas para adaptarse a los cambios, dado que uno de los pilares fundamentales para una economía planificada, sustancial en la construcción del socialismo, es una

1 Traducción libre del título: "Promoting Computer Literacy through Programming Python"

infraestructura informática adecuada (Cockshott & Cottrell, 2003), si Venezuela pretende realizar su proyecto de país requiere, para empezar, ponerse al día con el desafío que supone la sociedad red.

Un reto fundamental para responder al reto que no lanza la sociedad red es, como señala Castells (2001), poner en práctica una política educativa masiva, especialmente dirigida a los niños, orientada al procesamiento de la información y la generación de nuevos conocimientos, y una parte enorme del conocimiento de las tecnologías de información se genera programando, de allí la gran importancia de esta destreza.

2.2 Importancia del uso exclusivo de herramientas libres.

Del mismo modo en que la alfabetización digital habilita a las personas para procesar y generar conocimiento que reside en el ciberespacio, la libertad del software es un requisito básico para poder acceder a buena parte de este conocimiento. El movimiento del software libre surge como una respuesta a las prácticas abusivas de las empresas de desarrollo de software que empezaron a negar el código fuente de sus aplicaciones, y por tanto el acceso al conocimiento sobre los mismos, con la finalidad de imponer un modelo de negocio alrededor del software como producto (Stallman, 1999; Williams, 2002). Stallman (1999) puntualiza *"otra suposición implícita es que la única cuestión importante acerca del software es cuáles trabajos nos permite hacer--que nosotros los usuarios de las computadoras no debemos preocuparnos por el tipo de sociedad que se nos permite tener"*. Nótese que en una sociedad donde no se imponen los intereses comerciales de las compañías sobre los derechos fundamentales de los ciudadanos, desde el punto de vista de las tecnologías de la información, es fundamental el acceso al código fuente, y este código fuente por otra parte es prácticamente inútil si los ciudadanos como colectivo no poseen la capacidad básica de entender este código fuente y esta es una destreza básica de la programación.

Sin embargo, con el devenir de los años las grandes corporaciones del software han conseguido la manera de sacar partido del Software Libre como una estrategia para ganar espacios en el mercado, y la propuesta inicial de Stallman basada en la libertades individuales enmarcada en una ideología liberal no ha podido contrarrestar esta situación, aún cuando han ido evolucionado los esquemas de licenciamiento de software libre. Existen casos llamativos como la tecnología Java que si bien se encuentra bajo licencias libres todo el proceso de desarrollo esta controlado de forma implacable por la corporación Sun (Otero & Sanchez-Mariscal, 2004; LaMonica, 2005). El lenguaje de programación Java es uno de los más utilizados en la actualidad después de una agresiva estrategia de mercadotecnia global para introducirlo en los programas de estudio de los institutos de educación superior. Además, Sun realiza una considerable inversión para mostrarse como una empresa que favorece al software libre. Con este fin contrata personas para promocionar esta imagen en eventos de software libre, y participar en las comunidades de desarrolladores, donde con ayuda de los recursos que les provee la compañía obtienen prestigio e influencia en los procesos de toma de decisiones (Dahlander & Wallin, 2006).

La lección que no se ha querido entender es que la naturaleza del software libre, y al fin al cabo del conocimiento libre, no puede circunscribirse a un ámbito de licenciamientos. Si bien licencias como GPL o Creative Commons han representado logros significativos, son al fin y al cabo ideas que surgieron desde un trasfondo liberal y que obedecen a un marco legal consecuente con este, y no da cuenta de una sociedad

endógenamente desplegada basada en un *conocimiento emancipado* liberado de toda clase de subordinación o dependencia (Petrizzo, 2007). El uso exclusivo de software libre y el libre acceso al código es un requisito fundamental para la comprensión de las tecnologías de la información, y a partir de allí la masificación de destrezas como la programación permitirá generar procesos de aprendizaje y formas de organización social alternativas con capacidad de afrontar los retos de los nuevos tiempos, en particular, contrarrestar el poder hegemónico de las corporaciones.

2.3 Python como lenguaje de programación para niños, niñas y jóvenes.

En la enseñanza de la programación se requiere seleccionar con mucho cuidado un lenguaje de programación adecuado, que permita a los estudiantes interactuar y experimentar, y desarrollar capacidades básicas como la depuración de errores. Pero por el otro lado, se debe evitar en lo posible que el proceso de aprendizaje sea entorpecido por tecnicismos propios del lenguaje. Python es el único lenguaje de programación de propósito general involucrado en un proyecto de formación masiva diseñado para todos los niveles educativos, tal y como se plantea en la propuesta "Computer Programming for Everybody (CP4E)" (van Rossum, 1999) elaborada con el fin de empoderar a **todos** los usuarios de las computadoras con conocimientos de programación, y bajo la visión de generar cambios con beneficios de gran alcance comparables a la alfabetización, y con resultados tan palpables como las aplicaciones desarrolladas de forma colaborativa por las comunidades de software libre.

Existen varias herramientas para enseñar programación a los niños, niñas y jóvenes, entre estas destacan: Phrogram¹, Scratch², Alice³, Toontalk⁴, Gamemaker⁵, Greenfoot⁶, Logo⁷. Todos estos proyectos tienen en común que son lenguajes diseñados con capacidades muy limitadas exclusivamente para la enseñanza de niños y jóvenes, por lo que en muy raras ocasiones serían utilizados en el ejercicio profesional de un programador: En estas circunstancias, los estudiantes avanzados se verían obligados a utilizar los lenguajes base que en comparación son mucho más difíciles, perdiendo así todas las ventajas de la herramienta. Solamente Scratch, Alice, Greenfoot y algunas implementaciones de Logo se encuentran bajo licencias libres, en particular Scratch tiene una licencia con importantes limitaciones, aunque pone a disposición el código fuente establece que las aplicaciones derivadas del mismo no pueden subir los proyectos al portal del MIT (MIT Media Lab, 2009), y en consecuencia quedarían aislados de la mayor parte de la comunidad Scratch. Otros software tienen características interesantes, por ejemplo Phrogram basado en la plataforma .Net puede utilizarse para

1 <http://phrogram.com/>

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

3 <http://www.alice.org/>

4 <http://www.toontalk.com/>

5 <http://www.yoyogames.com/gamemaker>

6 <http://www.greenfoot.org>

7 [http://en.wikipedia.org/wiki/Logo_\(programming_language\)](http://en.wikipedia.org/wiki/Logo_(programming_language))

programar aplicaciones para las consolas XBOX, y Gamemaker resulta muy atractivo por las facilidades que provee para el desarrollo de videojuegos.

La única desventaja significativa de Python es que en una primera impresión no cuenta con una interfaz gráfica tan llamativa como la que tienen Scratch o Phrogram. Sin embargo, esta interfaz gráfica de Scratch hace que aún para las aplicaciones más sencillas se requiere una buena cantidad de clics. por lo que el proceso en realidad es bastante lento. El resultado de la programación es un diagrama del código, pero no el código propiamente dicho, los proyectos de Scratch se almacenan en formato ".sb" binario específico para el software Scratch. El hecho de no disponer de un "codigo fuente" de proyectos Scratch es una gran desventaja, porque la escritura y lectura de código fuente son destrezas fundamentales de la programación. Hay diferencias cognitivas importantes entre trabajar con diagramas de colores y texto, como señala Sartori (1998), el lenguaje perceptivo concreto tiene un contenido connotativo muy inferior al que tiene el lenguaje abstracto de un texto, aunque estemos hablando un texto escrito en un lenguaje de programación.

Python ofrece ventajas panorámicas sobre todas las opciones mencionadas arriba (y sus similares), a saber:

- Puede emplearse en la enseñanza de la programación ya sea en primaria, secundaria o educación superior. Permite escribir los programas más sencillos de forma muy clara y sencilla, y ofrece soporte para distintos paradigmas de programación, incorpora estructuras de datos avanzadas y soporta conceptos avanzados como la metaprogramación.
- Es un lenguaje utilizado ampliamente en el campo profesional, organizaciones como Google, Yahoo, NASA y CERN, lo emplean en gran cantidad de sus servicios y soluciones informáticas. La capacitación en Python sienta las bases de una formación profesional, y los alumnos pueden explorar progresivamente características más avanzadas sin cambiar de lenguaje.
- Es un lenguaje de propósito general, puede utilizarse para prácticamente cualquier tipo de aplicaciones: juegos y aplicaciones multimedia, aplicaciones web y de escritorio, bases de datos, cómputo científico, etc.
- Existen una buena cantidad de herramientas didácticas para facilitar el uso de Python en la enseñanza de la programación. Algunos ejemplos destacados son: RUR-PLE¹ y Guido Van Robot², que introducen conceptos de programación controlando el movimiento de un robot en un espacio bidimensional; Pyro³, un entorno para explorar conceptos en inteligencia artificial y robótica; Turtle⁴ y xturtle⁵, implementaciones del clásico lenguaje didáctico Logo; y la plataforma de desarrollo de actividades de aprendizaje Sugar⁶.

1 <http://rur-ple.sourceforge.net/>

2 <http://gvr.sourceforge.net/esp/>

3 <http://pyrorobotics.org/>

4 <http://docs.python.org/library/turtle.html>

5 <http://xturtle.rg16.at/index.html>

6 <http://www.sugarlabs.org/>

- Una cantidad enorme de material de aprendizaje de libre acceso en Internet, libros, tutoriales y videos demostrativos y clases. Destacan algunos libros diseñados específicamente para niños y jóvenes como:
 - *Snake Wrangling for Kids: Learning to Program with Python*¹. Autor: Jason R. Briggs. Libro bajo licencia Creative Commons² para personas desde 8 años en adelante.
 - *How to Think Like a Computer Scientist: Learning with Python 2nd Edition*. Autores: Jeffrey Elkner, Allen B. Downey, y Chris Meyers. Libro bajo la licencia GNU Free Document License, pensado para estudiantes de los últimos años de secundaria.
 - *Invent Your Own Computer Games with Python*. Autor: Albert Sweigart. Libro bajo Licencia Creative Commons³, diseñado para personas desde 9 años en adelante.
 - Estos tres libros están en continua revisión y actualización, se pueden descargar de forma totalmente legal y gratuita gracias a su licenciamiento y los autores agradecerían que fuesen traducidos a cualquier idioma.
- Existen unos cuantos libros similares que no se encuentran bajo licencias libres, como por ejemplo:
 - *Computer Programming is Fun!: Beginning Computer Programming with Python*⁴. Autor: David Handy. Libro escrito para estudiantes de secundaria, viene junto a un programa de estudio semanal, y ejercicios.
 - *Hello World!: Computer Programming for Kids and Other Beginners*⁵. Autores: Warren D. Sande y Carter Sande. Un libro muy dinámico y sugerente, lleno de imágenes y viñetas divertidas, para personas desde los 12 años.
 - Es importante destacar que aunque estos libros no se encuentran bajo licencias libres, el código fuente de todos los programas utilizados en los distintos libros se encuentran disponibles desde cada uno de los sitios web, y proveen herramientas y guías de trabajo que pueden ser muy útiles.
- Es utilizado con frecuencia para automatizar procesos cotidianos de oficina, generación de cartas e informes. También se emplea con frecuencia para integrar distintas aplicaciones. Estas son destrezas muy útiles en cualquier ambiente de oficina, y para mejorar la productividad cotidiana de cualquier usuario de computación.
- Es una parte sustancial de casi todas las distribuciones Linux en distintas arquitecturas, de forma que viene preinstalado en prácticamente cualquier computadora personal, y se encuentra disponible en

1 <http://www.briggs.net.nz/log/writing/snake-wrangling-for-kids/>

2 Específicamente: Creative Commons Attribution-Noncommercial-Share Alike 3.0 New Zealand License.

3 Específicamente: Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.

4 <http://www.handysoftware.com/cpif/>

5 <http://www.manning.com/sande/>

las consolas portátiles más interesantes basadas en hardware y software libre como GP2X¹ y OpenPandora².

Existen muchas características del lenguaje derivadas de su naturaleza dinámica que facilitan en gran medida el aprendizaje:

- No es necesario declarar las variables
- La sintaxis es muy simple al punto que a veces el código Python es incluso más reducido que el pseudocódigo de los textos de programación para lenguajes como Pascal.
- Inculca buenas prácticas de codificación como la documentación y la indentación.
- Incorpora estructuras de datos dinámicas que permiten escribir código muy complejo de forma intuitiva.
- Provee un interprete interactivo de comandos mediante el cual el estudiante obtiene respuesta inmediata de cada línea de código que escribe, y de los errores de transcripción, sintaxis o ejecución. El interprete interactivo funciona como entorno de desarrollo y depuración y manual en línea.
- Provee una biblioteca estándar extensa y comprensiva para realizar una gran cantidad de tareas, además existe una enorme cantidad de bibliotecas de terceros disponibles en Internet.

2.4 Carácter multidimensional de los videojuegos como estrategia educativa

Los videojuegos representan por sí mismos un área de conocimiento sumamente amplia y diversa. La discusión sobre los beneficios y perjuicios que causan los videojuegos a niños, niñas y jóvenes, ha sido extensa y profunda y está muy lejos de tener fin.

De cualquier manera, estamos ante dos hechos innegables:

1. En la actualidad una gran cantidad de niñas, niños y adolescentes utilizan regularmente juegos de vídeo y esta tendencia va en aumento, a un punto que el tiempo dedicado a los videojuegos está superando al tiempo dedicado a la televisión (y en muchos casos el tiempo que niños, niñas y adolescentes dedican a la TV es superior al tiempo que permanecen en la escuela).
2. Los videojuegos ya están influyendo en el desarrollo de destrezas, capacidades y patrones culturales de muchas personas en distintos ámbitos, desde los simuladores de vuelo que utilizan los pilotos comerciales y militares, pasando por los juegos multijugador en línea que requieren una gran cantidad de interacción social, hasta el desarrollo de motricidad fina para manipular distintos tipos de dispositivos.

En una primera instancia hay que tener muchísimo cuidado de generalizar cuando se trata el asunto de los videojuegos. El término juego por sí solo ya escapa a definiciones precisas. considerando que hace referencia a *una actividad de naturaleza lúdica que impone desafíos individuales o colectivos bajo un sistema de reglas de común acuerdo*. En cualquier caso, la ambigüedad está latente, un juego puede realizarse bajo distintos modos de

1 <http://wiki.gp2x.org/wiki/Python>

2 <http://pandorapress.net/2008/12/28/official-python-will-come-preinstalled-on-the-pandora-os/>

coerción social, y aún cuando se participe libremente puede ser poco divertida para algunos de sus participantes. Más allá de lo que suceda durante un juego en particular, es una actividad que se acepta socialmente como recreativa y la mayoría de las personas la realizan de buen grado. En ese orden de ideas, el videojuego es una aplicación computacional diseñada para que el usuario de la misma realice actividades recreativas, el beneficio educativo es un efecto colateral. En la actualidad, las posibilidades de la informática hacen que las aplicaciones computacionales puedan representar prácticamente cualquier faceta de la cultura humana, y esa misma diversidad se refleja en los videojuegos.

Los videojuegos generalmente se clasifican desde la perspectiva de los desarrolladores, muchas veces en términos de los desafíos técnicos que representan para estos, y otro criterio usual es la estructura de las actividades que ofrecen a los usuarios y usuarias. Así, entre los géneros de juegos más aceptados se habla de: Acción, Aventura, Simulación, Rol, Estrategia y Simulación de vehículos. Además, géneros como Acción abarcan a su vez varios subgéneros que son considerados géneros de primer orden desde el punto de vista comercial, tales como Lucha, Disparos, Deportes o Plataformas. Del mismo modo, los videojuegos pueden ser individuales o permitir varios jugadores simultáneos de forma presencial o remota, teniendo como extremo los juegos masivos por Internet donde pueden llegar a participar miles de jugadores.

Los videojuegos han marcado la pauta en la simulación de situaciones reales o ficticias y en el diseño de interfaces humano-máquina, así, prácticamente cualquier actividad educativa es susceptible de ser simulada. Sin embargo, excepto en situaciones muy concretas, los videojuegos diseñados explícitamente con un sentido educativo tienen escasa difusión, precisamente porque la faceta educativa del juego (entendida bajo la perspectiva de la educación formal tradicional) opaca casi por completo el sentido lúdico del juego. Esto ha llevado a concebir al videojuego educativo con frecuencia como un género aparte, diferenciado de los videojuegos "propiamente dichos".

Sin embargo, el tema de la educación basada en videojuegos es (aún) una idea radical que se sostiene en dos factores claves (Prensky, 2001):

1. La influencia de las tecnologías de información han modificado en los últimos años la forma en que las personas aprenden, incluso la forma en que estructuran sus procesos mentales.
2. Las nuevas generaciones han desarrollado por primera vez en la historia una amplia experiencia en el uso de interfaces humano-máquina, desde los electrodomésticos hasta las computadoras personales, y esto ha dado forma a nuevas preferencias y habilidades que ofrecen un gran potencial para su aprendizaje.

Así, lo que en el año 2001 se presentaba como una idea muy radical ha ido lentamente alcanzando nuevos espacios al punto que para este año se anunciaba en Nueva York, EEUU *Quest to Learn*¹ la primera escuela pública con un programa de estudio diseñado en base a videojuegos. Los beneficios comprobados y potenciales de la educación basada en videojuegos ha sido reseñada por distintos autores con un saldo bastante positivo (Shaffer,

1 <http://q2l.org/>

2006; Dumbleton & Kirriemuir, 2006), bajo consideraciones razonables en el cambio del método educativo, y en los que el rol de los profesores y profesoras es fundamental.

Sin embargo, la trampa del Hommo Videns del que nos habla Sartori (1998) sigue vigente y al acecho, aunado al riesgo permanente que impone la privatización del conocimiento en beneficio especialmente de las corporaciones transnacionales y en detrimento de la humanidad como un todo, cortando el flujo de sinergias latentes en todos los tejidos sociales. Una clave en esta encrucijada está en la insistencia de circunscribir a los ciudadanos y ciudadanas al rol de usuarios de software, ofreciendo interfaces gráficas cada vez más llamativas y funcionalidades más amigables. El nuevo paso que hay que dar está en una idea mucho más radical: *desarrollar destrezas para la programación de videojuegos*.

Schell (2008) enumera las destrezas requeridas por un diseñador de juegos: Animación, Antropología, Arquitectura, Tormenta de ideas, Negocios, Cinematografía, Comunicación, Escritura creativa, Economía, Ingeniería, Historia, Gerencia, Matemáticas, Música, Psicología, Oratoria, Diseño de sonido, Escritura técnica, y Artes visuales. Por su parte, Salen (2009) señala que un videojuego exitoso implica reunir habilidades de pensamiento de sistemas, solución de problemas críticos iterativos, arte y estética, escritura y narrativa, diseño interactivo, lógica y reglas de los juegos, y habilidades de programación. Y la pretensión, aunque parezca ambiciosa, es formar en múltiples alfabetizaciones con capacidades básicas como: definir las reglas de un sistema y probar sus límites, comprender como funcionan los sistemas y cómo pueden ser transformados, negociar las demandas variables de equidad, y colaborar en espacios colectivos donde el conocimiento se encuentra distribuido (Salen, 2009).

No es difícil llegar a la conclusión que lo que se requiere es la conformación de equipos multidisciplinarios, pero incluso en estos casos es necesario elaborar canales de comunicación transversales a todas estas disciplinas. Este hilo conductor parece ser la actitud del jugar, la resolución de problemas con una actitud lúdica, de desparpajo y alegría, lo cuál no excluye la rigurosidad y la excelencia, cuestiones en las que las máquinas son jueces insobornables.

Los videojuegos en el siglo 21 son lo que una década antes eran los textos hipermedia, y su construcción involucran las mismas destrezas y procesos que requiere la producción de medios audiovisuales en general. Es de esperar que en los próximos años aparezcan iniciativas equivalentes a Thinkquest¹ basadas en el desarrollo de videojuegos educativos. La construcción de videojuegos es una práctica que reporta más recompensas y aprendizaje que el simple uso de videojuegos, y puede ofrecer nuevas formas para comunicar información y demostrar conocimiento y comprensión para todos los estudiantes, pero especialmente para aquellos que encuentran serias dificultades en el modelo tradicional que hace un énfasis excesivo en la disciplina, la lectoescritura básica y el pensamiento lógico-matemático, ya que apela a un rango de inteligencias múltiples mucho más amplio (Jung & Park, 2009).

1 <http://en.wikipedia.org/wiki/ThinkQuest>

2.5 Papel y búsqueda de reconocimiento de las instituciones de educación, ciencia y tecnología

Uno de los hechos más destacados del inicio del año escolar venezolano 2009 fue el arranque del proyecto "Canaima Educativo" que consiste en la introducción de computadoras Classmate PC¹ en las aulas de clase venezolanas de la Escuela Primaria, empezando por 50 mil máquinas pero este número se aumentará progresivamente hasta llegar a un 1 millón. Se trata en particular de la segunda generación de Classmate PC que se fabrican en Portugal bajo el nombre de Magalhaês Classmate PC. Son computadoras que diseñó Intel como respuesta al Proyecto OLPC² (One Laptop Per Child). La elección ha sido adecuada debido al crecimiento y respaldo que ha tenido el Classmate PC en el último año en comparación con el OLPC, y gracias a la decisión del Estado venezolano de utilizar el sistema operativo Linux, en particular la distribución "Canaima Educativo" que da nombre al proyecto, hecha a la medida a partir de la distribución GNU/Linux Canaima³ diseñada en principio para cubrir las necesidades de la administración pública venezolana. Otro factor positivo ha sido un esfuerzo por dar identidad venezolana a distintas aplicaciones educativas existentes para Linux.

A pesar de esto, destacan varios aspectos que van en detrimento del proyecto:

- Excesiva lentitud en la toma de decisiones claves, aún cuando hace más de un año (en septiembre 2008) se anunciaba la compra de las computadoras a Portugal no es sino hasta ahora que se está convocando a un foro presencial⁴ para empezar a hablar sobre la generación de contenidos educativos. Es muy bueno que aún con retraso se inicie esta discusión, pero la realidad del país demanda mayor agilidad y planificación proactiva.
- Llama la atención la incapacidad manifiesta de utilizar las propias tecnologías de información y comunicación para la construcción de comunidades, y sin este punto de anclaje social es muy difícil alcanzar los objetivos mínimos. Se debe entender que aunque las comunidades se construyen mediante procesos emergentes que se van dando forma a sí mismos, no se trata de un proceso espontáneo en su origen. Se requiere una intención clara y manifiesta, y un acompañamiento de acceso a recursos materiales y de gestión para que las comunidades puedan avanzar de forma autónoma. No se trata de poner un wiki o un portal en Internet, sino crear instituciones dedicadas de forma exclusiva al cuidado de estas comunidades.
- Quizás el problema más profundo que se debe atacar es que aunque en el país se habla en términos de revolución, las instituciones apelan a concepciones totalmente conservadoras de la educación. Incluso las tecnologías libres más sofisticadas puestas al servicio de un modelo educativo desfasado de la actualidad no conseguirán sino reforzar los modelos existentes. En la promoción del Foro mencionado arriba se señala "potenciar lectura, escritura y pensamiento lógico-matemático de niñas y niños

1 <http://www.classmatepc.com/>

2 <http://olpc.com/>

3 <http://canaima.softwarelibre.gob.ve/cms>

4 <http://www.foroeducativo.cnti.gob.ve/>

venezolanos"¹. Esto parece una extensión de la preocupación simplista de reducir el esfuerzo de la educación primaria a construir capacidades de lectoescritura y aritmética básica, Esto es inaceptable en momentos en que se habla de múltiples inteligencias y múltiples alfabetizaciones, en las que se deben atender, entre otras, capacidades críticas, estéticas, de valores e identidad. No se trata de llenar a niñas y niños de contenidos sino de mostrar como se construyen esos contenidos. Todo esto bajo procesos donde los docentes se dispongan a crecer y aprender junto con las nuevas generaciones.

Las instituciones de la administración pública nacional que buscan reconocimiento como árbitros legítimos en la implantación de las políticas de ciencia, tecnología y educación no tienen más remedio que apurar el paso, y adaptarse a los tiempos que corren. Las respuestas son simples y directas:

1. Agilizar los procesos administrativos que conllevan a las decisiones técnicas y la alimentación de recursos a proyectos claves,
2. Abocarse la construcción de comunidades de práctica alrededor de los proyectos de uso de las nuevas tecnologías en la educación formal,
3. Reformular de forma sustancial los modelos educativos en función de una educación liberadora y creativa, que desafía los límites de los sistemas existentes y llena los contenidos de sensibilidad hacia sus semejantes, hacia el entorno, y hacia el cuidado del propio conocimiento humano como un bien público.

3 Elementos de un programa de estudios

3.1 Estrategia base

La alfabetización digital exige a niñas, niños y jóvenes la capacidad de programar, y en lo posible, esto debe ocurrir bajo un enfoque atractivo y envolvente, sin abandonar nunca la necesidad de acceder al código fuente, interpretarlo, interiorizarlo y modificarlo de forma creativa. Si bien se recomienda el uso de interfaces gráficas que faciliten el manejo del sistema, y la edición del código, las capacidades cognitivas que desarrolla la reconstrucción creativa a partir de elementos abstractos es muy valiosa e insustituible por elementos perceptuales por muy llamativos que estos sean.

Hay muy pocos lenguajes de programación que ofrecen una sintaxis con la sencillez suficiente para que las niñas, niños y jóvenes puedan adquirir destrezas de implantación de algoritmos sin la ominosa interferencia de tecnicismos intrascendentes, entre esos destacan lenguajes dinámicos como Lua, Python o Ruby. Entre estos lenguajes Python es el único que cuenta con las herramientas de desarrollo de juegos y simulación más maduras y coherentes.

Entonces la estrategia base consiste en explorar como los videojuegos desarrollan actitudes, destrezas, conocimientos, valores e identidad en las personas, y como el enorme potencial de motivación de los videojuegos se pone al servicio de construir nuevos modelos educativos, pasando por el uso planificado y

1 <http://www.foroeducativo.cnti.gob.ve/>

consciente de los videojuegos en el programa de estudios, y de forma simultánea hacer a los estudiantes partícipes de la formulación de los marcos epistemológicos que subyacen en los videojuegos, que como se ha dicho antes, son construcciones multidimensionales interactivas con el potencial de representar prácticamente cualquier faceta expresiva del ser humano.

Tres componentes fundamentales deben subsistir entonces en cualquier diseño de un programa de estudios basado en general en tecnologías de información, y en particular usando los videojuegos como canal motivacional:

- La posibilidad de acceder al conocimiento sin restricciones, de comprenderlo de forma cabal, realizar experimentos y modificaciones ilimitadas, y generar nuevo conocimiento. Si se trabaja con cualquier tipo de contenidos educativos, estos deben quedar susceptibles a críticas, sugerencias, modificaciones y añadidos.
- No prescindir del desarrollo de la interpretación, recreación y construcción abstracta, en favor del uso exclusivo de imágenes y otras elaboraciones concretas. La imagen no puede reemplazar por completo a la palabra, porque aunque una imagen pudiese valer más de mil palabras, en la palabra subyace el significado y sin palabras las imágenes quedan huecas de contenido.
- La construcción de valores de participación social y ciudadanía activa, crítica y solidaria. Los mundos virtuales son espacios ideales para mostrar que sin la interferencia de un marco histórico y cultural que ha propiciado sistemáticamente prácticas mezquinas y agendas particulares, el potencial constructivo del trabajo colectivo en un marco de equidad es inconmensurable.

3.2 Estrategia de transición

Con la introducción masiva de los computadores en el salón de clase gracias a proyectos como Canaima Educativo, y a otros proyectos que buscan sacar partido de las posibilidades de las nuevas tecnologías, una de las principales restricciones para la aplicación efectiva de los videojuegos como estrategia educativa va a quedar satisfecha de forma progresiva. La otra restricción, y quizás la más compleja, consiste en involucrar a los docentes en estas nuevas prácticas.

El enfoque de aprender mediante la construcción de videojuegos calza exactamente en un modelo educativo constructivista socialmente situado. Se habla entonces de una 3^{era} generación en el uso educativo de los videojuegos que demanda una mejor comprensión de los potenciales y las limitaciones de los videojuegos como herramienta educativa, y donde ya no se conciben como elementos ajenos al salón de clase sino incorporados como una entre tantas herramientas de las que dispone el docente en su trabajo (Egenfeldt-Nielsen, 2009).

- Se sugiere entonces el inicio de experiencias piloto en las que se seleccionen e incorporen videojuegos al aula de clase para reforzar procesos de adquisición de conocimientos fundamentales y exploración, elaborando para ellos contenidos de una forma más o menos tradicional. que propicien el acercamiento de docentes y estudiantes a las nuevas tecnologías.

- Iniciar la formación de docentes en programación Python, y su aplicación en metodologías para el diseño y desarrollo de videojuegos como experiencia educativa. Esta experiencia permitirá recoger requerimientos para el desarrollo de herramientas que faciliten la puesta en práctica de estas ideas.
- Elaboración de tutoriales y otros recursos educativos presenciales y a distancia para desarrollar destrezas básicas en la construcción de videojuegos, y el manejo de métodos y herramientas para su seguimiento y gestión. Por ejemplo, el uso de herramientas como Celtx¹, aplicación libre para la pre-producción de medios audiovisuales.
- Simultáneamente, diseñar estrategias involucrando estas nuevas tecnologías en la elaboración de proyectos de aula, espacio donde los estudiantes tienen la posibilidad de desarrollar temas de forma iterativa-incremental con la asistencia del docente.
- Dedicar un apoyo decidido a la construcción de comunidades de práctica, virtuales y tangibles, donde confluyan inquietudes, conocimientos y experiencias, entre todos los practicantes.

4 A modo de conclusión

Volviendo al punto inicial, de lo que se trata es construir las bases de un modelo de soberanía tecnológica apoyado en un estilo tecnológico de tipo socialista creativo, y sustentado en estructuras sociales endógenamente desplegadas y basadas en conocimiento libre emancipador. Las herramientas y los métodos pueden cambiar, pero ese es el objetivo.

La propuesta consiste en empoderar masivamente niñas, niños y jóvenes venezolanos con conocimientos de programación, usando la creación de videojuegos como elemento motivador, y el lenguaje de programación Python como herramienta pivote. Este plan demanda importantes requerimientos en cuanto a infraestructura tecnológica, y un esfuerzo considerable para la aproximación de los docentes a nuevos modelos educativos.

En el peor de los casos se contará a mediano plazo con una cantidad creciente de personas con destrezas mucho más efectivas y una visión más amplia respecto a la informática. Sin embargo, la visión que se persigue es potenciar el aprendizaje desde primaria y secundaria, en procesos que permitan crear y re-crear todos los contenidos hacia nuevas formas de expresión creativa, y re-plantear concepciones, identidades y valores para la construcción de nuevas destrezas y estructuras de organización social emancipadora.

Es evidente que la distancia entre la propuesta y el objetivo es enorme, pero toda excursión, corta o larga, empieza por el primer paso.

5 Recursos iniciales a considerar

5.1 Ideas

- Grupo de Especial Interés de Python en Educación:
<http://www.python.org/community/sigs/current/edu-sig/>

¹ <http://celtx.com>

- Educational Games Don't Have to Stink!
http://www.gamasutra.com/features/20050126/adams_01.shtml
- El Programa de Estudios Comparativos en Medios del MIT
<http://cms.mit.edu/>

5.2 Herramientas para diseñar herramientas de aprendizaje

- Renpy: motor multiplataforma para utilizar palabras, imágenes y sonidos para contar historias con el computador
http://renpy.org/wiki/renpy/Home_Page
- Sage: alternativa libre a software matemático como Magma, Maple, Mathematica y Matlab
<http://sagemath.org/>
- Sugar Learning Platform: plataforma para la construcción y distribución de actividades de aprendizaje
<http://www.sugarlabs.org/> y http://wiki.sugarlabs.org/go/Sugar_on_a_Stick

5.3 Herramientas para el aprendizaje de Python

- GASP Python Course: Taller introductorio a Python para personas sin conocimientos previos de programación (GASP es el acrónimo de Graphics Api for Students of Python)
<http://openbookproject.net/pybiblio/gasp/course>
- Invent your own computer games with Python: Libro libre que enseña a programar en Python en base la programación de juegos
<http://pythonbook.coffeeghost.net/index.html>

5.4 Herramientas para desarrolladores

- Pygame: herramientas para el desarrollo de videojuegos en Python basados en la biblioteca de programación SDL
<http://pygame.org>
- Pyglet: biblioteca de desarrollo de contenidos multimedia para Python sin dependencias externas.
<http://pyglet.org/>
- Libregamewiki: la enciclopedia de los juegos libres
<http://libregamewiki.org>
- Panda3D: motor de juegos y herramientas para el desarrollo de de juegos
<http://www.panda3d.org/>
- Soya3D: motor de juegos 3D para python de alto nivel
<http://home.gna.org/oomadness/en/soya3d/index.html>
- Python embebido en Blender
<http://www.blender.org/education-help/python/>

6 Referencias

- Abraham Otero Quintana, & Alvaro Sanchez-Mariscal. (2004). Análisis de la "libertad" en la plataforma Java. *JavaHispano*. Recuperado Octubre 15, 2009, a partir de http://www.javahispano.org/contenidos/es/analisis_de_la_libertad_en_la_plataforma_java/
- David Williamson Shaffer. (2006). *How Computer Games Help Children Learn*. Palgrave MacMillan.
- Giovanni Sartori. (1998). *Homo videns: La sociedad teledirigida*. Madrid: Taurus.
- Guido van Rossum. (1999). Computer Programming for Everybody. Recuperado a partir de <http://www.python.org/doc/essays/cp4e.html>
- Jae Yeob Jung, & Hyung Sung Park. (2009). Learning by Doing via Game Making. En *Digital Simulations for Improving Education: Learning Through Artificial Teaching Environments*. Information Science Reference.
- Jesse Schell. (2008). *The Art of Game Design, A Book of Lenses*. Morgan Kaufmann Publishers.
- John Alexander Miller. (2006). *Promoting Computer Literacy through Programming Python* (Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Education)). The University of Michigan. Recuperado a partir de <http://www.python.org/sigs/edu-sig/miller-dissertation.pdf>
- Katie Salen. (2009). Gaming Literacies: A Game Design Study in Action. En *Learning and Teaching with Electronic Games* (págs. 87-108). Association for the Advancement of Computing in Education (AACE).
- Linus Dahlander, & Martin W. Wallin. (2006). A man on the inside: Unlocking communities as complementary assets, 35, 1243-1259.
- Manuel Castells. (2001). *La Galaxia Internet*. Areté.
- Marc Prensky. (2001). *Digital Game-based Learning*.
- Mariángela Petrizzo. (2007). Conocimiento Emancipado para el Desarrollo Endógeno. *Essentia Libre*, 10, 26-29.
- Martin LaMonica. (2005). What's wrong with the Java community process? *CNET News*. Recuperado Octubre 15, 2009, a partir de http://news.cnet.com/Whats-wrong-with-the-Java-community-process/2008-1082_3-5607094.html
- MIT Media Lab. (2009). Scratch Source Code. Recuperado Octubre 15, 2009, a partir de http://info.scratch.mit.edu/Source_Code
- Oscar Varsavsky. (2006). *Hacia una política científica nacional*. Monte Ávila Editores Latinoamericana.
- Paul Cockshott, & Allin Cottrell. (2003). Economic planning, computers and labor values. Presented at the I Conferencia Internacional La Obra de Carlos Marx y los desafíos del Siglo XXI., La Habana, Cuba.
- Ramsés Fuenmayor. (2002). Venezuela: Su enfermedad y su crisis actual. *Cuestiones de América*, (12). Recuperado a partir de <http://www.cuestiones.ws/revista/n12/dic02-fuenmayor.htm>
- Richard Stallman. (1999). The GNU Operating System and the Free Software Movement. En *Open Sources : Voices from the Open Source Revolution* (pág. 280). O'Reilly. Recuperado a partir de <http://oreilly.com/catalog/opensources/book/stallman.html>

- Sam Williams. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. O'Reilly. Recuperado Octubre 15, 2009, a partir de <http://oreilly.com/openbook/freedom/ch01.html>
- Simon Egenfeldt-Nielsen. (2009). Third Generation Educational Use of Computer Games. En *Learning and Teaching with Electronic Games* (págs. 49-67). Association for the Advancement of Computing in Education (AACE).
- Tim Dumbleton, & John Kirriemuir. (2006). Digital games and education. En *Understanding Digital Games* (págs. 221-240). SAGE Publications.