

Pertinencia de la tecnología Java por y para el Estado Venezolano

Centro Nacional de Desarrollo e Investigación en Tecnologías Libres.

Índice de contenido

I La tecnología no es neutra.....	1
II El proyecto país de Venezuela.....	2
III La naturaleza corporativa de la tecnología Java.....	3
IV Adecuación de Java al proyecto país.....	6
IV.1 Construcción del tejido socioprodutivo endógeno.....	7
IV.2 Plena soberanía tecnológica, oportuna y relevante.....	9
IV.3 Aspectos técnicos.....	10
V Hacia una política tecnológica socialista creativa.....	11
VI Referencias.....	13

I La tecnología no es neutra

Antes que nada, es fundamental entender que la tecnología no es neutra. Con esto se quiere decir que en la propia naturaleza de la tecnología hay elementos desde los cuales se expresan los intereses y los valores de las personas que crean, modifican y usan tecnologías. No se puede separar la tecnología de la sociedad, la cual, al igual que el conocimiento científico es, en esencia, un producto humano, de dicha sociedad en donde está inserta. ¿Cuántas veces hemos escuchado “La tecnología no es buena ni mala, todo depende de cómo se use”?, estas son palabras que nos remiten al discurso de la “libertad de elección individual” del liberalismo. Feenberg (1996) señala que el enfoque esencialista según el cual es posible en una primera instrumentalización abstraer a la tecnología de la sociedad ha generado consecuencias que experimentamos actualmente tales como homogeneización cultural, anomia social y crisis ambiental. No es casualidad que precisamente estas sean cuestiones comúnmente asociadas como manifestaciones de la globalización y el capitalismo. Como bien señala Varsavsky (1972) si la

tecnología y la investigación científica fuesen neutras servirían indistintamente a cualquier “estilo científico”, lo que el mismo demuestra que no es cierto. Cada estilo científico determina y promueve tecnologías diferentes, según los valores sociales en los que se fundamenta ese estilo.

II El proyecto país de Venezuela

Sin lugar a dudas, el espacio donde aparece plasmado de la forma más clara y concisa el proyecto de país que se adelanta en la Venezuela actual es en el Proyecto Nacional “Simón Bolívar”, PPSN (Despacho de la Presidencia, 2007), denominado sin reparos “Primer Plan Socialista de la Nación” (PPSN). Todas las líneas estratégicas planteadas en ese documento pueden ser vinculadas directamente con el enfoque que se pretende lograr sobre la tecnología nacional: nueva ética socialista, suprema felicidad social, democracia protagónica revolucionaria, modelo productivo socialista, nueva geopolítica nacional, Venezuela: potencia energética mundial, y nueva geopolítica internacional .

Puede parecer en una primera instancia que cuestiones como la “nueva ética socialista” no tienen que ver con la tecnología (obviamente, desde una concepción limitada y “tradicional” de la tecnología), pero cuando se encuentra en la estrategia “superar la ética del capital” como primer punto “fomentar el trabajo creador y productivo”, se puede deducir sin ambigüedades que esto requiere de proveer las condiciones más favorables para la apropiación del conocimiento y la tecnología necesaria para ello.

A lo largo del documento del PPSN se encuentran de manera reiterada referencias a la necesidad de un profundo cambio en las políticas tecnológicas del país. Sin pretender ser exhaustivos con respecto a lo señalado en el plan, y mucho menos en sus implicaciones y los detalles de su implantación, se pueden considerar algunos ejemplos que dan sustento a esta última afirmación. Particularmente para el desarrollo de estrategias y políticas, tales cómo:

- “Fortalecer la accesibilidad a los alimentos”,

- “Propiciar la seguridad y soberanía farmacéutica”,
- “Fortalecer e incentivar la investigación en el proceso educativo”,
- “Fomentar la actualización permanente de nuestro pueblo en el entendimiento del mundo contemporáneo”,
- “Incentivar un modelo de producción y consumo ambientalmente sustentable”,
- “Apoyar el equilibrio entre productores, poder popular y el Estado en la planificación, gestión económica y distribución de excedentes”,
- “Fomentar la capacidad de toma de decisiones de la población”,
- “Garantizar la transparencia y democratización de la información”,
- “Expandir la Economía Social cambiando el modelo de apropiación y distribución de excedentes ”,
- “Consolidar el carácter endógeno de la economía”,
- “Incrementar la producción nacional de ciencia, tecnología e innovación hacia necesidades y potencialidades del país ”,
- “Ordenar el territorio asegurando la base de sustentación ecológica”,
- “Mejorar el hábitat de los principales centros urbanos”,
- “Desarrollar la industria del gas natural libre”,
- “Ampliar y mejorar la red de transmisión y distribución de la electricidad”,
- “Desarrollar alianzas energéticas en el contexto de la integración regional”,
- “Profundizar el intercambio cultural y la independencia científica y tecnológica”.

En cada una de estas estrategias puede apreciarse, por un lado, una clara relación con el establecimiento de una política tecnológica autónoma y soberana, y además, en términos de Varsavsky, un “estilo tecnológico” muy bien definido de tipo “socialista creativo”. Se pueden apreciar entre las distintas líneas estratégicas lugares comunes, tales como: el fortalecimiento de las capacidades locales, fomento de la equidad y la justicia social, establecimiento de relaciones internacionales ante el bloque hegemónico mundial, cuidado de la armonía con el medio ambiente, etc.

El tema que se quiere abordar en este documento es la necesidad inminente de abordar el análisis sobre la pertinencia de utilizar la tecnología Java por y para el Estado venezolano, en qué medida y de qué manera, sin olvidar en ningún momento que de acuerdo al artículo 5 la constitución vigente de la República Bolivariana de Venezuela “La soberanía reside intransferiblemente en el pueblo, quien la ejerce directamente en la forma prevista en esta Constitución y en la ley, e indirectamente, mediante el sufragio, por los órganos que ejercen el Poder Público. Los entes del Estado emanan de la soberanía popular y a ella están sometidos”. Todo esto debe responder a los lugares comunes de las estrategias del PPSN que se enumeran en el párrafo anterior.

III La naturaleza corporativa de la tecnología Java

Cuando se habla de Java es importante distinguir entre el lenguaje de programación Java¹ y la tecnología Java que sirve de soporte a este. El lenguaje de programación Java se puede considerar libre, tan libre como el lenguaje C# de Microsoft. Existen muchos sitios web y libros que describen en detalle la sintaxis y los conceptos detrás de estos lenguajes. Al hacer referencia a la tecnología Java, se incluyen todos los componentes que conforman lo que Sun Microsystems denomina la “Plataforma Java” en sus distintas “Ediciones”: Micro, Standard y Enterprise, los cuales son necesarios para ejecutar cualquier aplicación desarrollada con el lenguaje Java. De igual manera, ha de incluirse aquí todos los factores humanos y organizaciones que arrojan los mecanismos de toma de decisiones empleados en el desarrollo de esta tecnología.

La tecnología Java se ha originado desde diciembre de 1990 como un proyecto interno de Sun Microsystems, la cual es una corporación transnacional que con poco más de 30 mil empleados, realiza operaciones en más de 100 países y se encuentra en el lugar 187 de las más grandes corporaciones estadounidenses, una lista en donde Exxon Mobil se encuentra en el 2^{do} lugar, y Microsoft en el lugar 49 (Fortune Magazine, 2007). El desarrollo de la tecnología Java surge ante la

¹ El cual es, al igual que cualquier otro lenguaje de programación, un conjunto de reglas sintácticas y semánticas usadas para controlar el comportamiento de un sistema computacional.

necesidad de plantear una respuesta a la supremacía de Microsoft en el mercado del software. Uno de los principales objetivos para la creación de Java fue el desarrollo de software que pudiese ejecutarse en distintos sistemas operativos (Wikipedia, 2008), lo cual abría la posibilidad que aplicaciones que se pudiesen ejecutar en el sistema operativo Windows, eventualmente pudiesen prescindir del mismo. En otras palabras, la tecnología se origina en una corporación sin otro objetivo que ofrecer un producto que pudiese otorgarle ventajas competitivas sobre sus competidores, nunca ha estado en los intereses de Sun incorporar valores similares a aquellos preponderantes en el proyecto país de Venezuela expuesto en el PPSN y mencionado en la sección anterior.

A lo largo de los años, fundamentalmente en virtud de las estrategias de mercadeo de Sun, el lenguaje de programación Java ha pasado a ser el lenguaje de programación más utilizado (TIOBE, 2008), para lograr esto ha tenido como una de sus principales vías de penetración los entornos universitarios. En Venezuela, por ejemplo, se utiliza la tecnología Java de forma preponderante en las carreras universitarias relacionadas con las tecnologías de información y comunicación. La elección de la tecnología Java se justifica por lo general para dar respuesta a las supuestas demandas de un mercado laboral, pero nadie se plantea la conveniencia de esta tecnología para el desarrollo de un modelo de país alternativo.

La estrategia de Sun Microsystems en la actualidad, al igual que IBM y Google, consiste en utilizar a las herramientas y a las propias comunidades de software libre como un herramienta en su lucha por obtener ventajas comparativas, sin otro fin que el de mejorar sus beneficios económicos, y a través de estos, sus cuotas de poder, tal y cómo se establece en el recetario del capitalismo. A la transnacional Sun nunca le ha dejado de interesar aumentar su capital y mejorar su posición en el mercado, y en ese sentido ha comprendido a lo largo de los años cómo utilizar al movimiento del software libre en su provecho. Por otra parte, Sun realiza una considerable inversión para mostrarse como una empresa que favorece al software libre, y para este fin contrata personas específicamente con la finalidad de promocionar esta imagen en cualquier evento relevante de software libre. De igual

modo, contratan personas que forman parte de las comunidades, o que penetran de forma intencionada en distintos proyectos de software libre, y con ayuda de los recursos que les provee la compañía ganan con cierta facilidad prestigio e influencia en los procesos de toma de decisiones (Dahlander y Wallin, 2006).

Con frecuencia se promociona la apertura de Java a las comunidades a través del denominado Java Community Process (JCP), y aquí nuevamente nos encontramos con las prácticas corporativas y burocráticas de Sun. Para formar parte del JCP se puede hacer a título individual o como representante de una organización, las personas que deciden entrar a título individual no tienen que pagar nada, pero no pueden ser ciudadanos de alguno de los países que los EEUU han colocado en el “eje del mal”, y tienen prohibido comunicar los conocimientos que adquieran en los procesos de discusión a sus respectivas organizaciones. El derecho para participar en el JCP para organizaciones tiene un costo de varios miles de dólares (Sun Microsystems, 2008). Por ejemplo si el miembro de un grupo de usuarios de Linux desea participar, el grupo debe cancelar 2000 US\$ anuales, de lo contrario tendría prohibido legalmente comunicar cualquier conocimiento que adquiriera en su participación en el JCP a sus compañeros del grupo de usuarios. El JCP no garantiza de ninguna manera un proceso de decisión justo, aunque las discusiones de los estándares se realiza con todos los miembros del JCP, las decisiones las toman los Comités de Expertos (CE) de cada conjunto de especificaciones. De forma obligatoria, siempre hay un representante de Sun en cada CE que tiene derecho de veto sobre el resto, y 10 miembros más son propuestos por Sun. Toda la gestión administrativa del JCP (denominados Program Management Office -PMO-) es realizada por empleados de Sun. Siempre hay un miembro de Sun en cada comité de 3 miembros que decide si un organismo sin fines de lucro puede formar parte del JCP.

En fin, aunque Sun se esfuerce en demostrar lo contrario (y prueba de esto es la gran cantidad de artículos que indican la apertura del JCP, con autores siempre en asociación directa con Sun u grupos de usuarios de Java), existen notas de prensa que señalan lo contrario. Por ejemplo en el artículo *“What's wrong with the Java*

community process?" (LaMonica, 2005) donde un ejecutivo de IBM señala: *"Nuestra posición es que Java podría estar mejor servido si existiera un proceso de generación de estándares más democrático, en lugar de uno donde la compañía tiene derechos de super-mayoría sobre cualquier otro. Pienso que el JCP funciona bien a veces. Y las otras veces queda empantanado, ¿y entonces qué es lo que Sun quiere? deja de ser comunitario en ese punto"*. Si esta es la opinión de un representante de una mega corporación como IBM, ¿qué se podría esperar de un representante de una comunidad de software libre?.

Ante todo esto, los distintos voceros de Sun repiten en forma reiterada que la empresa *"no usará su cuota de poder para otro fin distinto que no sea el velar por la plataforma"*. Pero esto lamentablemente es una cuestión de fe casi ciega, que no resulta razonable si se tiene cierto conocimiento del mundo de las transnacionales. Aquí cabe preguntarse, ¿Qué se entiende por velar por la plataforma?, ¿el fin de la plataforma es el mismo para una transnacional que para una comunidad de software libre?, ¿Cómo se vela que la transnacional la use para la evolución armónica de la tecnología y no para sus intereses particulares?. La única forma de estar seguros es que las comunidades tengan total y absoluta autonomía. Aunque Sun ha empezado a liberar algunos componentes de Java bajo licencia GPL, es importante entender que tal y cómo señalan Otero y Sánchez-Mariscal (2004), miembros de la comunidad javaHispano, más allá de las licencias es *"...más grave la falta de libertad plena en la creación y mantenimiento de las especificaciones de la plataforma. La rescisión de los privilegios de Sun, así como la gestión del JCP por una organización sin ánimo de lucro, evitando así que posibles intereses particulares de una empresa puedan llevar el JCP por el camino que no conviene a la mayor parte de la comunidad, son condiciones necesarias para lograr la libertad de la plataforma."*

IV Adecuación de Java al proyecto país

En base a las dos secciones anteriores, pasemos a contrastar en qué medida la tecnología Java se adecúa al proyecto de país planteado para Venezuela. El

proyecto de país, tal y cómo se mencionó arriba, es una cuestión extensa y compleja en extremo. Así, con la finalidad de simplificar el análisis se puede trabajar en base a dos elementos fundamentales comunes, que cómo se ha visto van surgiendo en las distintas políticas y estrategias del PPSN, a saber: la construcción del tejido socioproductivo endógeno para el asentamiento de las bases para la equidad y la justicia social, y el establecimiento de una soberanía tecnológica plena, oportuna y relevante.

IV.1 Construcción del tejido socioproductivo endógeno

Se requiere la construcción de una industria nacional del software en base, como debería ser obvio, al Software Libre. Y esta industria nacional debe tener características establecidas con claridad en el PPSN con un predominio de Empresas de Economía Social, a través del impulso de un sistema de innovación de redes de ciudadanos que acceden y comparten conocimiento en condiciones de libertad, con la finalidad expresa de reducir las desigualdades sociales.

El uso eficaz de la tecnología Java requeriría tener la posibilidad que la población venezolana se apropie de esta tecnología hasta en sus más pequeños detalles. No se está diciendo con esto que cada persona conozca todos los detalles de la plataforma, y tampoco en que sería suficiente que tan sólo una elite muy reducida conozca los detalles de implementación. De cualquier manera, debería establecerse un modelo de formación masiva que hiciera posible la apropiación. Ahora bien, la decisión desde hace algunos años de enseñar Java en las universidades venezolanas ha tenido muy poco éxito en contribuir a la construcción de una industria venezolana del software.

En distintas universidades extranjeras en las que se ha contado con una larga experiencia en el uso de Java para la enseñanza de la programación, y una experiencia mucho mayor en el desarrollo de grandes proyectos de software, donde se ha usado Java en la enseñanza superior, en particular como primer lenguaje de programación, han recibido quejas de las industrias por los resultados obtenidos.

Dewar y Schonberg (2008), a través de su experiencia en la Universidad de Nueva York, señalan que las destrezas de programación para la resolución de problemas son restringidas desde el enfoque de la plataforma Java² a una simple selección del objeto más apropiado para una determinada tarea (usan como analogía, el modo en que un plomero encuentra soluciones en una ferretería). Señalan también que el uso de lenguajes que ofrecen soluciones rápidas, como Javascript o PHP, generan problemas similares a los atribuidos a Java. De igual manera, hace poco se ha anunciado que en la Universidad de Waterloo en Canadá se empezará a utilizar Scheme en lugar de Java (Ragde, 2008), para satisfacer la necesidad de utilizar lenguajes de programación que permitan expresar conceptos que pueden trasladarse a otros lenguajes de programación y capacitar a los estudiantes en el estudio de temas más avanzados. Spolsky, autor del popular blog “Joel on Software”, ya se adelantaba a estas situaciones en su artículo “Los peligros de las escuelas de Java” (Spolsky, 2005), donde habla del auge de Java en universidades donde se utiliza en el 100% de las asignaturas, y como esto les imposibilitaría para trabajar, por ejemplo, en el kernel de Linux, en entender los algoritmos que hacen a Google escalable, y en particular, cercena la capacidad de abstracción que permite tomar las decisiones apropiadas en el desarrollo de grandes sistemas³.

Otra cuestión que hay que considerar, es que la población venezolana debería también contar con la posibilidad de participar de forma activa y protagónica en el diseño e implementación de esta tecnología para poder modificarla según sus necesidades. Para lograr este fin, los desarrolladores venezolanos deberían participar en el JCP sin estar sometidos a todas las condiciones mencionadas antes. Otra opción sería iniciar desarrollos paralelos haciendo implementaciones propias de la tecnología, lo que parece en gran medida irrealizable, una evidencia de esto es que los desarrolladores del proyecto Harmony, una de las implantaciones libres de las especificaciones de la plataforma Java, aunque son utilizados continuamente por Sun como evidencia de las bondades de su modelo abierto a las comunidades,

2 Aunque en el artículo se habla del lenguaje Java, claramente se hace referencia a la tecnología Java en general, es decir, a la manera que se ha planteado que se debe utilizar dicha tecnología en contextos determinados.

3 Precisamente esos “grandes sistemas” en donde se espera que Java sea fuerte.

han tenido que publicar en internet una “Carta abierta a Sun Microsystems” por la incapacidad de obtener una licencia aceptable para realizar las pruebas de compatibilidad de los desarrollos de Harmony con la especificación Java 2 SE 5. Este el trato que recibe la importantísima Fundación Apache, los desarrolladores del servidor web más utilizado en el planeta. ¿Qué podría esperarse entonces para los desarrolladores venezolanos?, imagínese entonces el trato que recibirían varias aplicaciones proyectadas, e incluso operativas, para la administración pública tales como SIGEFIRRH, cuando se requiriese probar funcionalidades de la edición Enterprise de la plataforma Java.

En resumen, la apropiación real de la tecnología Java por el tejido productivo venezolano que se proyecta en el Primer Plan Socialista de la Nación es una alternativa inviable, y en la medida en que sea utilizada esta tecnología se seguirá acentuando la dependencia de la política tecnológica endógena a las decisiones de una corporación transnacional norteamericana. Los actuales desarrollos a gran escala basados en java en la administración pública, y los intentos de establecer esta tecnología como una suerte de estándar para el país, deberían disminuirse tanto y tan rápido como sea posible por el bien del país.

IV.2 Plena soberanía tecnológica, oportuna y relevante

Si nos apegamos a la concepción profunda de soberanía popular que se expresa en la constitución, se debería entender que la imposibilidad de apropiación efectiva de la tecnología Java por parte del pueblo venezolano ya es un argumento suficiente para calificar la propagación descontrolada del uso de la tecnología Java como una seria amenaza para la soberanía del país. Sin embargo, también existen rasgos complementarios de la soberanía de tipo nacional en los que se concibe esta como el derecho exclusivo del gobierno del Estado para ejercer control sobre un área de gobernancia, en particular en relación a la seguridad nacional. En este orden de ideas, es crucial considerar las proyecciones que se desprenden del PPSN en relación a hacer frente al bloque hegemónico mundial mediante la conformación del bloque geopolítico regional y de un mundo multipolar.

Ya se ha señalado que la tecnología Java se origina en una corporación transnacional estadounidense, Sun Microsystems, que utilizando estrategias de mercadeo, tal y cómo si se tratara de un perfume o un electrodoméstico, ha pretendido globalizar dicha solución tecnológica como si fuese una panacea, cuando dista mucho de serlo. Se ha tratado de mostrar, además, como esta corporación mediante su poder económico utiliza para su propio beneficio los esfuerzos de las comunidades de software libre e intenta intervenir en sus decisiones, mientras impone serias restricciones burocráticas y económicas para que las comunidades intervengan en las decisiones sobre las especificaciones de la tecnología Java.

Si acaso esto fuera poco, todos y cada uno de los componentes de cualquiera de las ediciones de la plataforma Java tienen en sus términos de licencia “Regulaciones de exportación”⁴, que es una de las formas en la que el gobierno de los EEUU trata de ejercer su rol de policía del mundo. Estas regulaciones prohíben expresamente la exportación a países en una lista donde aparecen en forma recurrente Irán y Cuba, los cuáles según el PPSN tienen un rol estratégico clave en la conformación del nuevo mapa geopolítico mundial. Y además, el gobierno de los EEUU ha amenazado de forma insistente con incluir a Venezuela dentro de países de que apoyan acciones terroristas (Lobe, 2005), ¿qué sucedería entonces?, ¿se tendrían que abandonar de inmediato todas las aplicaciones desarrolladas en Java o utilizarlas a escondidas?, la improvisación no debería aparecer nunca como alternativa para la protección de la soberanía venezolana en ningún ámbito. Es necesario pensar en un modo de protección de la Soberanía venezolana que pasa por la tecnología, y que nos provea de la suficiente autonomía en los procesos de construcción del modelo de sociedad deseada.

IV.3 Aspectos técnicos

Si se incluyen consideraciones técnicas se puede encontrar que las industrias

4 Un ejemplo se puede encontrar en: <http://java.sun.com/javase/6/jdk-6u4-license.txt>. Y para constatar a cuáles países se encuentran bajo dichas restricciones: <http://www.bis.doc.gov/licensing/exportingbasics.htm>

básicas y la industria petrolera requieren en lo fundamental el desarrollo de controladores para dispositivos electrónicos como los sistemas SCADA y otro tipo de sensores, y sistemas que permitan el procesamiento, integración y análisis de grandes cantidades de información, los procesos administrativos tienen una importancia secundaria para estas industrias. La gestión de los entes de la administración pública requiere integrar información heterogénea y distribuida, y facilitar mecanismos de interacción con los ciudadanos y para integrarse con procesos emergentes de tipo administrativo y productivo que habrán de surgir de las empresas de economía social y los consejos comunales.

Este nuevo panorama requiere desarrollos sumamente flexibles, basados en tecnologías mixtas que incorporen por un lado lenguajes de bajo nivel que permitan la implantación de controladores y sistemas de seguimiento en tiempo real, y por el otro lado lenguajes dinámicos de muy alto nivel que permitan integrar información y aumentar al máximo la productividad de los desarrolladores. Se requiere también la asistencia de sistemas inteligentes que sirvan de soporte a la toma de decisiones colaborativa, con la intervención protagónica de la ciudadanía, y en general, de sistemas que utilicen los recursos computacionales de forma óptima.

Ante este panorama, sería más conveniente contar con una combinación razonable de tecnologías interoperables fáciles de hacer funcionar en Clusters Linux y estándares de comunicación sencillos que evitan al máximo la redundancia. Incorporando por un lado lenguajes de bajo nivel como C, C++ o D; lenguajes dinámicos como Perl, Python, Ruby y Javascript; lenguajes de tipo funcional y concurrente como Ocaml y Erlang, etc. son una opción mucho más apropiada en relación a los requerimientos técnicos mencionados que las tecnologías Java diseñadas para ser utilizadas en la implantación de enormes aplicaciones monolíticas requeridas por las empresas transnacionales para sus procesos de negocio, las cuales no tienen reparos en inversiones cuantiosas de hardware para satisfacer el hambre de recursos de las máquinas virtuales de Java.

El tema se planteó recientemente en la lista de discusión de SOLVE, y allí José Luis

Rey (2008) argumentó que el lenguaje de programación Java *“es un lenguaje mediocre, que no permite implementar técnicas de programación, que todo programador debería conocer y dominar. Lo peor es que ahora estoy seguro de que eso es así por imposición”*. Señala que aunque el diseñador de Scheme fue contratado en la concepción de Java, en el resultado final se eliminaron todas las capacidades poco convencionales como la herencia múltiple, para evitar que el lenguaje fuera *demasiado* avanzado. Así, Java la mediocridad de java no es casual sino intencionada⁵. Además, el lenguaje Java cuenta con un sistema de tipo de datos estático y simplista que no permite el manejo de características más avanzadas como metaprogramación o programación funcional.

Rey asegura, al igual que muchos artículos disponibles en Internet, que una de las grandes trampas de la plataforma Java es que esta se fundamenta en bibliotecas de programación muy extensas que dan la impresión a algunos programadores que cuentan con herramientas de gran poder. Estas bibliotecas en su gran mayoría cuentan con APIs⁶ sumamente idiosincráticas de gran complejidad que requieren un alto grado de experticia y especialización en algunas bibliotecas específicas. Así pues, se trata de manejar la complejidad intrínseca en el modelado de sistemas con un lenguaje simplista mediante bibliotecas sumamente extensas y complicadas, que en muchos casos solamente si se cuenta con conocimientos que el propio lenguaje Java niega, como la programación funcional y la programación genérica.

Rey continua su argumentación señalando que la plataforma Java ofrece un lenguaje pobre con muchas bibliotecas, mientras que en el mundo del software libre auténtico se pueden tener muchos lenguajes interesantes, que además, entre todos ofrecen más bibliotecas de las que se puedan concebir en ningún proyecto en particular. Mejor aún, cuando se tiene que resolver un nuevo problema es posible escaparse de las limitaciones de un lenguaje/plataforma en particular, sino se puede contar con la plena libertad de elegir las herramientas idóneas para ese

5 De acuerdo, es de suponer, al “estilo tecnológico” imperante en el Sun Microsystems de entonces.

6 Acrónimo del término en inglés Application Programming Interface, que son el conjunto de parámetros y funciones que ofrece la biblioteca de programación para ser utilizada desde otra aplicación como una capa de abstracción.

problema, desde el lenguaje a las bibliotecas, pasando por los paradigmas de programación.

V Hacia una política tecnológica socialista creativa

Muchos dirigentes del actual gobierno venezolano, al tomar sus decisiones parecen tomarse muy a la ligera los preceptos básicos de la actual revolución, sus fundamentos ideológicos, muy bien establecidos en “El árbol de las tres raíces”. Parece ser que algunas urgencias, y principalmente mucha ignorancia, justifican la utilización de la tecnología Java, entre otras, contraviniendo la esencia del Proyecto “Simón Bolívar”. Es importante recordar entonces la máxima Robinsoniana (Rodríguez, 1842):

“¿Dónde iremos a buscar modelos? La América española es original. Originales han de ser sus instituciones y su gobierno. Y originales, los medios de fundar uno y otro. O inventamos o erramos”.

Como obra humana no es diferente la tecnología de la política, y por lo tanto, aducir que es posible llevar a buen puerto a la revolución bolivariana usando cualquier tipo de tecnología es prueba de un claro condicionamiento ideológico, del mismo tipo de condicionamiento desde donde se añade el adjetivo de “trasnochado” al socialismo. Venezuela no puede hacer ya caso a ese tipo de condicionamientos.

Por fortuna, para problemas en tecnologías de información como los que tiene el país hay disponibles una enorme cantidad de herramientas y soluciones desde el software libre **auténtico**, que subyace en las comunidades de desarrolladores distribuidos en el planeta, la mayoría de las veces utilizando buena parte de su tiempo libre para ofrecer su conocimiento a la humanidad. Existen claros ejemplos de proyectos sumamente exitosos en este sentido, y a través de su estudio detallado podríamos entender con mucha mejor propiedad los temas más avanzados de las tecnologías de información: los sistemas operativos Linux y BSD; los proyectos GNU; las distribuciones Linux Debian y Gentoo; el servidor web Apache; los lenguajes de programación Haskell, Lisp, Scheme, Ocaml, Perl, PHP, Python, Ruby, entre otros; el navegador web Firefox; los manejadores de bases de datos

BerkleyDB, Sqlite y PostgreSQL; los sistemas de información geográfica GRASS y Quantum GIS; los sistemas de planificación de recursos empresariales (ERP) TinyERP, ERP5 y Oratio, todo esto tan sólo por nombrar los más conocidos.

En lo que respecta a las herramientas de desarrollo viniendo desde el mundo de las aplicaciones privativas, existe una inclinación excesiva hacia la búsqueda de Entornos de Desarrollos Integrados, precisamente como los que ofrecen plataformas como Java y .Net. Cuando se trata del desarrollo colaborativo de software libre, sin duda el eje de este modelo no son los lenguajes de programación ni las herramientas para escribir código, sino las herramientas que coadyuven a la convergencia del trabajo colaborativo, donde destaca de forma predominante los sistemas de control de versiones tales como Subversion o Mercurial. El esquema en el software libre auténtico nunca ha sido imponer tecnologías específicas o establecer certificaciones⁷, sino proveer a los expertos en las distintas tecnologías de los mecanismos para tomar en forma abierta y concertada las decisiones que permitan el avance de la tecnología.

Todos estos elementos conducen a un replanteamiento de una industria nacional del software donde fortalezcan los valores y los modos de funcionamiento de las comunidades de software libre auténticas. Se trata, entonces, como señala Feenberg (1991), de una revisión de la tecnología que incorpore las habilidades propias de los trabajadores y las trabajadoras (usuarios y hacedores de la tecnología), los procesos de la comunicación humana, y los límites ambientales, en el diseño de su estructura. Esta revisión conlleva a procesos que no obedecen a urgencias ni campañas electorales, pero que sin duda si contribuyen de forma significativa a la construcción de la plena soberanía de un pueblo.

VI Referencias

Dahlander, L., Wallin, M. W., (2006). *A man on the inside: Unlocking communities as*

⁷ Las certificaciones sobre las tecnologías libres surgen por lo general como respuestas a la demandas de la industria, conformada esta industria por empresas de tipo capitalista que quieren evitar al máximo cualquier situación de riesgo que pueda afectar la rentabilidad de sus negocios.

complementary assets. Research Policy 35, pp. 1243-1259

Despacho de la Presidencia (2008). Proyecto Simón Bolívar, Primer Plan Socialista de la Nación, Líneas Generales del Plan de Desarrollo Económico y Social de la Nación 2007-2013 www.embavenez-paris.com/docs/g04ot1c21.pdf

Feenberg, A. (1991). *Critical Theory of Technology*. New York: Oxford Univ. Press.

Feenberg, A. (1996). *From Essentialism to Constructivism: Philosophy of Technology at the Crossroads*. . URI: <http://www.sfu.ca/~andrewf/talk4.html>.

Fortune Magazine (2007). *Annual ranking of America's largest corporations*. <http://money.cnn.com/magazines/fortune/fortune500/2007/index.html>

LaMonica, M. (2005). *What's wrong with the Java community process?*. URI: http://www.news.com/Whats-wrong-with-the-java-community-process/2008-1082_3-5607094.html

Lobe, J. (2005). *Washington Focuses on Southern 'Axis of Evil'*. URI: <http://www.antiwar.com/lobe/?articleid=5334>.

Otero, A. y Sánchez-Mariscal, A. (2004). *Análisis de la "libertad" en la plataforma Java*. URI: http://www.javahispano.org/contenidos/es/analisis_de_la_libertad_en_la_plataforma_java/

Prabhakar Ragde. (2008). *Computer Science at Waterloo: the new Scheme of things*. URI: <http://compsci.ca/blog/computer-science-at-waterloo-the-new-scheme-of-things/> (Accedido el 24/03/2008).

Rey, J. L. (2008). *Distintas intervenciones en el tema "nuevos aliados anti-java"*. Lista de discusión Software Libre de SOLVE. <http://solve.net.ve/pipermail/softwarelibre/2008-January/thread.html>

Rodríguez, S. (1842). *O inventamos o erramos*. Monte Ávila Editores, (Edición del

1988).

TIOBE Software (2008). TIOBE Programming Community Index. URI: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. (Accedido el 25/03/2008)

Varsavsky, O. (2006). *Hacia una política científica nacional*. Monte Ávila Editores.

Wikipedia (2008). *Java (programming language)*. URI: http://en.wikipedia.org/wiki/Java_%28programming_language%29#Primary_goals. (Accedido el 23/03/2008).