

Estudio de factores conductuales que contribuyen al cultivo de prácticas virtuosas en el Desarrollo de Software Libre

Johanna Álvarez, Nelevis Báez, Víctor Bravo, Daisy Villasana

Centro Nacional de Desarrollo e Investigación en Tecnologías Libres – CENDITEL
Mérida, Venezuela

jalvarez@cenditel.gob.ve, nbaez@cenditel.gob.ve, vbravo@cenditel.gob.ve,
dvillasana@cenditel.gob.ve

Fecha de recepción: 26/10/2016

Fecha de aceptación: 03/11/2016

Pág: 112– 138

Resumen

Este artículo describe el estudio realizado por el equipo de Aseguramiento de Calidad en el Desarrollo de Software Libre de CENDITEL, en el año 2014, sobre factores conductuales que podrían contribuir al cultivo de prácticas virtuosas en el desarrollo de Software Libre (SL). Dicho estudio se fundamenta en una serie de entrevistas realizadas a miembros de algunas comunidades de desarrollo de SL en Venezuela y en el exterior, así como en la revisión de referencias relacionadas a este tema. El análisis de las entrevistas y de los contenidos revisados, se realizó en base a las siguientes dimensiones: interacción, colaboración, cumplimiento de las políticas definidas por las comunidades, liderazgo, motivación y “Ética Hacker”.

Este análisis permitió identificar factores conductuales tales como la fraternidad, el respeto, la humildad, el trabajo, la honestidad, la libertad y el amor; cuya presencia en los integrantes de una práctica de desarrollo de SL parecen favorecer, de manera significativa, no sólo la producción de aplicaciones de acceso libre sino el cultivo de procesos colaborativos de enseñanza- aprendizaje. Estos procesos buscan mejorar las prácticas de desarrollo de SL, tanto a nivel técnico como a nivel social, en pro de alcanzar la construcción colectiva del conocimiento que tiene como propósito principal contribuir al bien común de la humanidad.

Palabras Clave: software libre, comunidades de software libre, factores conductuales, prácticas virtuosas.

Introducción

Los procesos de construcción y desarrollo de software suelen ser complejos, ya que conllevan generalmente la interacción de muchas personas para la consecución de un objetivo en común,

cuyo requerimiento es, fundamentalmente, que los participantes demuestren capacidades para contribuir de forma conveniente a la construcción del producto.

En el caso del SL, estos procesos tienen una visión dinámica para la coordinación de las actividades, enmarcados en un modelo que propicia la maduración de los productos a través del “bazar” o construcción colectiva con relaciones jerárquicas débiles, orientadas en función del gusto por el aprender, la contribución a la sociedad y la mejora continua del software que se desarrolla.

El modelo de construcción colectiva que caracteriza el estilo de desarrollo de SL, implica una serie de aspectos a nivel del comportamiento humano que hacen posible un esfuerzo en comunidad, esfuerzo que tributa al cultivo de bienes de uso común, en este caso el software y el conocimiento implícito en este, así como en su práctica de desarrollo. Dichos aspectos representan factores a nivel conductual que tienen incidencia significativa para promover procesos de construcción colectiva de conocimiento, entre los cuales la colaboración, la interacción, el cumplimiento de políticas, el liderazgo, la motivación y la posible praxis de la “Ética Hacker” parecen representar la base que posibilita estos procesos. En este sentido, el objetivo de este artículo se encuentra dirigido al estudio de tales factores conductuales en algunas comunidades de desarrollo de SL, ello, a través de un primer acercamiento que permita tener una visión sobre el papel de estos en las prácticas de desarrollo de SL.

En este trabajo, como primer punto, se realiza una revisión de los fundamentos teóricos que están vinculados con las prácticas de desarrollo. En el apartado correspondiente, se incluyen el origen de las primeras comunidades de SL, así como también los hitos más importantes relacionados con la construcción colectiva.

Luego se expone un análisis del estudio realizado sobre algunas comunidades venezolanas, constituidas como personas con personalidad jurídica y sin fines de lucro, que trabajan bajo los esquemas de SL, así como algunas comunidades a nivel internacional, tomando como insumo un conjunto de entrevistas realizadas para este fin. Por último, con fundamento en el análisis se propone una serie de recomendaciones para promover mejoras en las prácticas de desarrollo orientadas en los factores conductuales estudiados.

Cultivo de prácticas virtuosas en el desarrollo de software libre

A principios de los años setenta, con el proyecto de desarrollo del software UNIX, llevado a cabo por el Instituto Tecnológico de Massachusetts, los Laboratorios Bell de ATT y otros actores, se pueden registrar los primeros indicios de las prácticas de desarrollo de software. Tanto los colaboradores del proyecto UNIX, como los desarrolladores de la Universidad de California en Berkeley, llevaron a cabo ciertas prácticas que se fueron consolidando en el tiempo, ya que primero usaron el código fuente de UNIX y le fueron haciendo modificaciones, hasta que el permiso para usarlo fue restringido por la compañía ATT y crearon su propia distribución, denominada Berkeley Software Distribution (BSD). Interpretando a Leonard, el trabajo en esta nueva distribución y las subsiguientes, se caracterizó por la inclusión de otros actores en el desarrollo, nutriendo el proceso con sus contribuciones, llegando a crear redes distribuidas

de colaboración y logrando establecer la estructura de la metodología de desarrollo de fuente abierta [1].

Luego, con el anuncio inicial del Proyecto GNU de Richard Stallman en 1983, quedaron plasmados los principales fundamentos de lo que sería a posteriori las Comunidades de Software Libre (CSL): La libertad y la colaboración. Así, para justificar por qué debía escribir GNU, Stallman planteaba:

(...) la regla de oro exige que si yo quiero un programa debo compartirlo con otras personas que también lo quieren. No puedo, conscientemente, firmar un acuerdo de confidencialidad o un acuerdo de licencia de software. Para que yo pueda continuar utilizando las computadoras sin violar mis principios, he decidido reunir suficiente software libre de manera que podré continuar sin necesidad de utilizar algún software que no sea libre[2].

“La regla de oro” como está descrita, consiste en que el software como un producto generado en colectivo debe seguir construyéndose así y para ello el desarrollo debe darse de manera colaborativa, haciéndose imprescindible el libre acceso al código fuente que da vida al software.

Según lo que plantea Stallman[3], el desarrollo del software UNIX, no se daba bajo ningún tipo de restricciones y, por consiguiente, se hizo necesario adherir el adjetivo “libre” a la definición software, una vez que grandes compañías tecnológicas como Digital y Symbolics, Inc., en los años ochenta, comenzaron a implementar ciertas medidas para monopolizar su producción privatizándolo. Por consiguiente, la libertad es una condición propia del software el cual nació sin restricciones de acceso, copia, mejora y distribución; siendo entonces una creación del intelecto humano que llega a ser privatizado por una minoría a través de los derechos de explotación exclusiva, bien conocidos como Derechos de Propiedad Intelectual. Dicha privatización niega el desarrollo colaborativo e instaura la condición de competitividad y explotación, negando a su vez la condición sine qua non del conocimiento como lo es ser un bien público.

Ahora bien, en 1985 se crea la Fundación para el Software Libre, en inglés Free Software Foundation (FSF) llegando a ser una organización sin fines de lucro, libre de impuestos cuya misión principal es luchar para defender las 4 libertades del SL:

- La libertad de ejecutar el programa para cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello[4].

El mecanismo utilizado para poder cumplir con esta misión ha sido impulsado, en principio, por el método del copyleft el cual promueve que las copias y modificaciones de un programa sean de dominio público imposibilitando que llegue a ser un software privativo. Para establecer esta prédica como norma, a posteriori surgió la Licencia Pública General, en inglés General Public License (GPL) y sus distintas versiones, originándose todo un movimiento con variedad de protecciones a las tendencias de privatización, entre ellas la Licencia Libre para Documentación (GNU Free Documentation License FDL)[5]. Esto evidencia la lucha existente ante la privatización del conocimiento, haciéndose imperante la generación de mecanismos normativos en defensa del conocimiento como bien público mediante la protección de las tecnologías libres, por ejemplo, lo cual debería ser una de las tareas a cumplir por parte de las distintas CSL a nivel mundial.

Con respecto al segundo fundamento de las CSL, la colaboración, se podría destacar que el desarrollo del SL plantea el trabajo en conjunto para la consecución de un fin común. La colaboración es lo que a su vez da sustento a la defensa del conocimiento como un bien común, es decir, al darle supremacía a la comunidad y no al individualismo, las CSL ponen en evidencia que ningún conocimiento pertenece a un solo individuo, debido a que es producto de la construcción social que responde a un contexto cultural en particular. El esfuerzo por incrementar la organización del trabajo colaborativo coloca el énfasis en que los resultados pueden ser mejores, ya que una amplia participación de sujetos activos es clave para el bien común.

Para ampliar esta perspectiva, la reflexión realizada por León [6] respecto a la relación entre “Socialismo y Software Libre”, plantea que aparte del control de los medios de producción, otro de los atributos en común entre el socialismo y el SL es la fraternidad. Sin embargo, esta fraternidad que promueve el SL se diferencia mucho de lo que en su momento Stuart Mill y Herbert Spencer, por ejemplo, promovieron en el siglo XIX como la unión de intereses, teniendo solo un sentido utilitarista. Por el contrario, se considera que la fraternidad a la que apunta el SL es la que se refiere a la importancia de la comunidad como núcleo central para el alcance de importantes logros en pro de un objetivo en común, como lo es generar más conocimiento por parte de todos y para todos.

En este sentido, según el citado autor: “La fraternidad del socialismo de hoy en día gira en torno al compartimiento del conocimiento. Aquí, el movimiento del SL es consecuente sobre el conocimiento libre (...)” (p. 60)[6]. Esto induce a pensar que las CSL no solo reclaman el compartir el software producido sino también desarrollarlo en colectivo, lo cual presenta la necesidad de describir los modelos de organización del trabajo que se pueden dar en el desarrollo de software planteados por Eric Steven Raymond.

Raymond, promotor de la Iniciativa del Código Abierto, en inglés Open Source Initiative (OSI)¹, fracción de la FSL, a partir de 1998 origina otra organización fundada para dar mayor

¹“La Iniciativa del Código Abierto (OSI), es una corporación sin fines de lucro con alcance global que está constituida para educar y recomendar acerca de los beneficios del código abierto y para construir puentes entre los distintos distritos de la comunidad del código abierto. El Código Abierto, es un método del desarrollo para software que aprovecha el poder de revisión igual distribuida y transparencia del proceso. La promesa del código

relevancia al acceso del código fuente instaurando una visión más instrumental que filosófica del SL, en su ensayo sobre la defensa de la iniciativa OSI “La Catedral y el Bazar”, escrito en 1997, describe dos modelos en la producción de software:

En el “Modelo Catedral”, el desarrollo se lleva a cabo por unos cuantos desarrolladores teniendo una organización centralizada donde lo producido no se da a conocer al público hasta no estar culminado, de esta manera se trata de ocultar los errores cometidos por ciertas fallas en la planificación. La negación a que otros formen parte del proceso de desarrollo produce que se obvien las consideraciones de terceros, los cuales en muchos casos terminan siendo usuarios directos y, en consecuencia, son fuente esencial para otorgar información sobre lo que en realidad se necesita. En este modelo, la participación de la mayoría no es un factor clave para la consecución del objetivo común, es así como se duda que el mismo sea “común” a todos, siendo diseñado y planificado bajo los lineamientos de una política a puertas cerradas [8].

La innovación se concibe como la capacidad para generar nuevas necesidades ante los posibles usuarios, es decir, el nuevo producto tecnológico surge como un insumo más para ampliar el mercado. La objetividad científica es la que determina las normas a seguir en el desarrollo, por ende, la tecnología se convierte en un simple instrumento para el desarrollo económico de la organización productora de software, y, para resguardar este sistema de producción, hay que establecer normas que garanticen los Derechos de Propiedad Intelectual. Programar bajo este modelo, requiere de un ambiente de competencia entre pocos programadores y es por ello que la información necesita ser secuestrada para mantener la planificación cerrada.

Por otra parte, en el “Modelo Bazar”, el modo de organización de trabajo se basa en el trabajo colaborativo donde lo que se produce proviene del conocimiento generado en colectivo, considerando como fuente esencial las necesidades presentadas por los usuarios. De esta manera, el conocimiento libre es clave no solo para tener acceso a la información sino para participar de manera activa en la construcción colectiva de un software, por tanto, en contraposición con el “Modelo Catedral”, la planificación es descentralizada instaurando una “política de puertas abiertas”, permitiendo la participación de la mayoría: “otorgando suficientes miradas, todos los errores estarán a la vista” (Traducido de [8], p.1).

Bajo este modelo, la innovación no va a ser un simple instrumento del mercado, va a consistir en un mecanismo generador de soluciones que responda a las demandas de los usuarios, contribuyendo a la construcción sociocultural del conocimiento. La seguridad, por su parte, va a concretarse una vez que la información esté bajo el dominio público y de esta manera se pueda mantener el control de los productos generados y de su utilidad para el bien común. La programación entonces se nutre de la “construcción comunal” y aunque surgen ciertos liderazgos, el conocimiento aportado por cada uno de los miembros es fuente esencial para la transformación de la realidad existente.

Ambos modelos poseen divergentes concepciones de planificación organizacional y desarrollo de software: El “Modelo Catedral”, establece parámetros unidireccionales para llevar a cabo la programación donde la dirección de los proyectos está al mando de un grupo cerrado de

abierto es mejor calidad, alta autenticidad, más flexibilidad, bajo costo y el fin de los bloqueos impuestos por los vendedores depredadores” (Traducido de [7]).

desarrolladores, únicos privilegiados de poseer el conocimiento necesario para alcanzar los objetivos pautados a priori, muchas veces con poco conocimiento con respecto a las necesidades de los usuarios. Muy por el contrario, el “Modelo Bazar” otorga una programación basada en la colaboración de una mayoría activa y transformadora, capaz de producir los cambios necesarios en el sistema para mejorar su estado, proyectando lo producido al bien común.

En cuanto a la estructura de producción en proyectos de SL, como los casos de BSD y Linux, esta ha sido descrita por varios investigadores, entre ellos Cano [1], quien hace referencia a los distintos actores y sus respectivos roles dentro de un proyecto de desarrollo de SL:

- En primer lugar, están quienes dirigen el proyecto, equipo núcleo (core team) en el desarrollo de BSD y/o los hackers del kernel en Linux, estos poseen un mapa de la ruta que va a seguir el desarrollo y ejercen influencia en la direccionalidad de la misma, acceden a escribir en los repositorios donde está el código fuente y otorgan permiso a otros desarrolladores, colaboradores, para escribir bien sea en todo el código, solo un módulo o algunas áreas.
- Luego, están quienes logran acceder a la escritura en el código fuente sin ser específicamente líderes del proyecto (committers), miembros de la comunidad que ejercen influencia, no en toda la escritura del código pero sí en algún módulo o área, de tal manera que pueden ser considerados como enlace con el proyecto por aquellos quienes no participan del desarrollo en sí.
- También, están los desarrolladores (developers) cuyas funciones consisten en contribuir con el desarrollo de alguna parte del código, muchas veces a través de los committers, agregar documentación, hacer traducciones o el conjunto de instrucciones para automatizar alguna tarea (scripts).
- Finalmente, aunque no por ello menos importante, están los usuarios quienes en el desarrollo de SL constituyen un actor fundamental ya que no son considerados simples observadores sino copartícipes de la construcción del software, al tomar en cuenta las sugerencias respecto a las necesidades de funcionalidades a desarrollar, mediante reuniones presenciales o en línea, etc.

Otras características que menciona el citado autor respecto a la estructura de producción de SL, referente a la participación de los actores, son: El carácter descentralizado de dicha participación ya que contribuyen mediante la red de redes, Internet, desde cualquier lugar del mundo. También, en cuanto al cambio de roles, pasar de ser colaborador a líder del proyecto, dependerá de los méritos obtenidos por sus contribuciones y en consecuencia, se dará el reconocimiento de la comunidad de desarrollo. Todo esto aporta al desarrollo de un software en módulos, lo que permite avanzar en el proyecto sin depender del desarrollo de una parte en específico, además de coordinar el desarrollo teniendo una versión mediante un software de control sin necesidad de un servidor central, así como también, el poder reutilizar los módulos en otros desarrollos.

Por otro lado, la descripción de cómo debería darse el desarrollo del software bajo un paradigma que en realidad incluya los dos fundamentos del SL (libertad y colaboración), es decir, el “Modelo Bazar” descrito previamente, da pie a indagar sobre las características de la “Ética Hacker”, en este sentido Himanen [9] señala que:

El diccionario del argot hacker, el «jargon file», compilado de forma colectiva en la Red, define a los hackers² como personas que se dedican a «programar de forma entusiasta»(1) y creen que «poner en común la información constituye un extraordinario bien, y que además para ellos es un deber de naturaleza ética compartir su competencia y pericia elaborando software gratuito³ y facilitando el acceso a la información y a los recursos de computación siempre que ello sea posible»,(2) Ésta ha sido la ética hacker desde que un grupo de apasionados programadores del MIT [Massachusetts Institute of Technology] empezaron a llamarse hacker a principios de la década de 1960.(3) Con posterioridad, a mediados de la década de 1980, los medios de comunicación, empezaron a aplicar el término a los criminales informáticos. A fin de evitar la confusión con aquellos que dedican su tiempo a escribir virus informáticos y a colarse en los sistemas de información, los hackers empezaron a denominar crackers a estos usuarios destructivos o piratas informáticos (4) (p. 5, El énfasis es de los autores).

Varios son los aspectos a destacar del estudio de Himanen. En primer lugar, hace mención a que la actividad del hacker informático está relacionada a compartir no sólo la información sobre lo que ha creado sino así como también su competencia y pericia, es decir, el sentido de lo que ha producido, el conocimiento previo adquirido, las capacidades que ha puesto en práctica para conseguir su objetivo y sobre todo la contribución a la comunidad. Todo ello implica mucho más que otorgar información básica sobre un producto en particular para poder ser utilizado (proceso de transferencia tecnológica en la sociedad de consumo) muy por el contrario, requiere de ciertas virtudes para disminuir las relaciones sociales de manipulación existentes en la producción de conocimiento hoy en día.

Según Contreras[10]: “(...) Los hackers de los MSL precisamente procuran esto, desplegar una vida de mejoramiento y crítica continua, aprendizaje incesante y retos cada vez más complejos (...) La vida hacker exige coherencia en la búsqueda por la liberación (...)” (p. 25). De lo expuesto por el autor, se podría reflexionar que para el logro de la libertad es necesaria “la pasión” que Himanen menciona en la “Ética hacker del trabajo”. Se deduce que la “búsqueda por la liberación” no se trata de ir simplemente colocando a la luz pública información oculta por supuestos motivos de seguridad pública, va más allá, ya que si el hacker posee pasión por una práctica en específico, la misma se va a nutrir de conocimiento y dicha práctica trascenderá cuando otro se apropie del conocimiento generado, le haga mejoras y lo comparta con otros, esto implica hacer público más que la información. Al mencionar entonces la palabra pasión son

² Si bien en esta definición el autor hace referencia al hacker en el campo de la informática, en el transcurrir de su obra deja claro que muchas personas con las virtudes que él describe pueden ser llamadas hackers.

³ Probablemente la versión en español tradujo de forma errónea la palabra free en inglés y el autor en realidad hacía referencia al SL en vez de software gratuito.

muchos otros adjetivos los que van de la mano: constancia, creatividad y esfuerzo, al contrario de lo que se suele pensar, el ocio no es algo directamente relacionado con la pasión.

Otro aspecto que describe Himanen es la relación entre el tiempo y la “Ética Hacker”, siendo la flexibilidad la característica principal ya que el hacker organiza sus actividades sin límites preestablecidos por otro. De allí que, si bien los horarios clásicos del “espíritu del capital” (la ética protestante) al que alude el autor en su obra, hacen culto a la optimización del tiempo en términos de réditos económicos, la cultura hacker insiste en que el ritmo de trabajo, visto como un placer más de la vida y no como un castigo, debe ser definido por nosotros mismos. Los límites que se impusieron con los horarios disminuyeron la relevancia de las metas planteadas y de la creatividad, lo cual el hacker sí considera trascendental llegar a obtener, sea en el horario que sea.

En cuanto a la motivación hacia el trabajo, el autor menciona que el valor del trabajo como actividad de reconocimiento social ha sido dejado de lado por la motivación de acumular cada vez más dinero, más allá de la satisfacción de necesidades básicas, de modo tal que los verdaderos hackers contradicen la motivación monetaria en la sociedad de la información actual estando entonces movidos por la pasión por el quehacer, el reconocimiento de sus semejantes, el cultivo de la creatividad y el amor por la comunidad.

Conociendo la estructura de la producción de SL y las motivaciones que se podrían encontrar en la “Ética Hacker”, se pueden indicar algunas características de la práctica de desarrollo de SL, aclarando que, en principio, se entenderá por práctica:

(...) cualquier forma coherente y compleja de actividad humana cooperativa, establecida socialmente, mediante la cual se realizan los bienes inherentes a la misma mientras se intenta lograr los modelos de excelencia que le son apropiados a esa forma de actividad y la definen parcialmente, con el resultado de que la capacidad humana de lograr la excelencia y los conceptos humanos de los fines y bienes que conlleva se extienden sistemáticamente (p. 248)[11].

Dichas características de la práctica de desarrollo de SL se podrían resumir en: la liberación del código fuente para cualquier persona, lo que conlleva a una revisión del mismo para evaluar su aporte a otros desarrollos, sea este significativo o no, es decir, puede ser mejorado, posibilitando realizar actualizaciones siempre y cuando se consideren las observaciones de los usuarios. Otro aspecto importante, es contribuir con el código original y esto implica que en la administración del proyecto se trate de mantener el interés de otros actores, lo cual se logra mediante el otorgamiento de reconocimientos por las contribuciones dadas gracias a la liberación temprana del código y a la necesidad de resolver ciertas fallas (bugs). Evidentemente, como se mencionó antes, la colaboración es esencial para llevar a cabo el objetivo común planteado. Colaboración no solo en todas las etapas de la metodología de desarrollo sino con mayor énfasis en la etapa de construcción, ya que es importante que muchos desarrolladores conozcan cómo se hizo dicha construcción para que la misma pueda replicarse en otros espacios y las fallas que se encuentren en esta puedan resolverse rápidamente sin depender de un solo desarrollador [1].

Un esquema resumen a considerar de lo que se busca en las prácticas de producción de SL, son las reglas numeradas por Raymon, citado por Turner[12]:

- Si pierdes interés en un programa, herédalo a un sucesor competente.
- Tratar a tus usuarios como co-desarrolladores es la mejor ruta para mejorar rápidamente el código y depurar de modo efectivo.
- Libera temprano y seguido, y escucha a tus usuarios.
- Si la base de co-desarrollo y de probadores-beta es amplia, casi cualquier problema se clasificará rápidamente y su arreglo será obvio para alguien.
- Si tratas a tus probadores-beta como tu recurso más valioso, ellos responderán convirtiéndose justo en eso.
- La siguiente mejor cosa de tener buenas ideas es reconocerlas en tus usuarios.
- Si el coordinador de desarrollo tiene un medio, de menos, tan bueno como internet, y sabe cómo guiar sin coerción, muchas cabezas son inevitablemente mejor que una sola (p. 160).

Ante el panorama descrito, desde los inicios de las prácticas de SL, los fundamentos del SL, los modelos de producción de software, la estructura de producción en SL, la “Ética Hacker” y la dinámica de las prácticas de desarrollo de SL, es conveniente aproximarse a vislumbrar cómo esto se está llevando a cabo en Venezuela:

En el año 2004, se promulgó el Decreto Nro 3.390 [13] en cuyo artículo 1er se lee: “La Administración Pública Nacional empleará prioritariamente SL desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional (APN) iniciarán los procesos de migración gradual y progresiva de éstos hacia el SL desarrollado con Estándares Abiertos”. La génesis de esta decisión data del año 2002, cuando en el paro petrolero fueron vulnerados los sistemas informáticos de la industria petrolera nacional (Petróleos de Venezuela, S.A.), los cuales siendo privativos estaban en manos de centros económicos de poder foráneos.

Se evidenció del paro de la principal industria nacional la vulnerabilidad de su sistema y, con esta, la debilidad que implica la dependencia tecnológica, sobre todo en el principal sector productivo del país. Comienza entonces todo un impulso al uso, capacitación, desarrollo, investigación, normalización y reflexión en tecnologías libres con la creación de varias instituciones coordinadas, para entonces, por el ya existente Ministerio de Ciencia y Tecnología, creado desde el año 1999. Entes adscritos al actual Ministerio del Poder Popular para Educación Universitaria, Ciencia y Tecnología (MPPEUCT) como el Centro Nacional de Innovación Tecnológica (CENIT), la Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL) y el Centro Nacional de Tecnologías de Información (CNTI) han contribuido con la misión de impulsar el SL en el país.

En el año 2013, fue promulgada la Ley de Infogobierno [14], cuyo objeto consiste en establecer las bases, principios y lineamientos que regirán el uso de las tecnologías de información en el Poder Público y el Poder Popular, con la finalidad de mejorar la gestión pública por un lado y principalmente, para que la base del desarrollo, adquisición, implementación y uso de las tecnologías de información sea el conocimiento libre. La importancia de esta ley radica, no en el simple hecho de la migración a SL por parte de la APN sino, en que se pueda impulsar el desarrollo e investigación en tecnologías libres, para ir generando espacios donde se fomente la libertad y la colaboración inherentes a estas.

La misión de impulsar el SL en Venezuela, no solo ha sido liderada por los entes adscritos al MPPEUCT y no se ha limitado a la migración al SL de las aplicaciones utilizadas por la APN, también ha ido creciendo en otros espacios, por ejemplo, el desarrollo del sistema operativo GNU/Linux Canaima y la participación de las distintas CSL en Venezuela: Comunidad Debian Venezuela, Comunidad Canaima GNU/Linux, Comunidad Python Venezuela, Ubuntu Venezuela.

La Comunidad Debian Venezuela trata de promover el proyecto Debian GNU/Linux, el cual establece en su contrato social cinco ideas básicas: el sistema operativo permanecerá 100 % libre, contribuirán a la comunidad de SL, no se ocultarán los problemas, la prioridad son los usuarios y el SL y otorgarán a los usuarios la posibilidad de usar trabajos en los que no se adopten las directrices del SL Debian, para lo cual la comunidad creó las secciones (contrib y «non-free») en sus archivos. Así mismo, establecieron las orientaciones de lo que debe ser el SL en la distribución Debian: los componentes deben tener libertad de distribución, se debe tener el código fuente completo, la licencia debe permitir la posibilidad de generar otros trabajos (como por ejemplo lo fue Canaima GNU/Linux) y la distribución de software a partir de código fuente modificado, no debe haber discriminación contra personas o grupos, no se debe limitar su uso en base a la finalidad perseguida, la licencia será distribuida a los trabajos derivados y no debe colocar restricciones a los mismos siendo GPL, BSD y Artística las licencias que el proyecto considera pertinentes.

El proyecto en Venezuela cuenta con la participación de desarrolladores, mantenedores y colaboradores quienes en principio se mantienen comunicados continuamente y luego se encuentran en el evento anual “Día Debian”, donde llevan a cabo debates e imparten talleres (referentes a virtualización, instalación, desarrollo, empaquetado de software, entre otros), haciendo instalaciones de sistema, tanto estable como de prueba [15].

La Comunidad Canaima GNU/Linux está conformada por un conjunto de colaboradores y desarrolladores que trabajan en el “proyecto socio-tecnológico abierto construido de forma colaborativa” Canaima GNU/Linux, teniendo este como distribución de origen el sistema Debian. A su vez otros proyectos se derivan de Canaima, como: Canaima Educativo (desarrollo de contenidos educativos, aplicaciones y funciones en SL que contribuyan a un modelo educativo integral), Canaima Universitario (comunidad de estudiantes, profesores y activistas del SL que pretenden promocionar el uso del sistema Canaima para la “formación integral de profesionales”), Canaima Comunal (distribución concebida para brindar herramientas tecnológicas a los consejos comunales), Canaima Colibrí (pretende ser una aplicación ligera

para computadoras de “bajos recursos”) y Canaima Forense (herramienta para la investigación forense).

La participación en este proyecto puede hacerse adquiriendo responsabilidades a través de los roles existentes: desarrollador, relacionista público, documentador, administrador, facilitador, diseñador, traductor [16]. El modo de comunicación que se establece entre los usuarios y los colaboradores del proyecto se da a través de varios medios: Chat IRC (Chat de la Comunidad Canaima GNU/Linux”), Wiki (Enciclopedia colaborativa), lista de correo (soporte, desarrolladores, discusión, servicios) y el formulario de contacto. Dicha comunicación ocurre de manera virtual pero también la comunidad posee otro modo de atender las necesidades de los usuarios a través de encuentros presenciales de desarrolladores, denominados “Cayapas Canaima”, en los que estos, mediante el trabajo colaborativo, intercambian experiencias para así ir generando mejoras tanto organizacionales como técnicas [16].

La Comunidad Python, por su parte, se constituye a partir de la creación de dicho lenguaje de programación como sucesor del lenguaje llamado ABC, a principios de 1990 por Guido van Rossum, en los Países Bajos. Su creador sigue siendo su autor principal aunque incluye colaboraciones. Todas las versiones de Python son Open Source. La mayoría, no todos, los comunicados de Python han sido compatibles con la GPL; sin que ello implique distribución de Python bajo la GPL debido a que, todas las licencias de Python, a diferencia de la GPL, permiten distribuir una versión modificada sin realizar cambios de código abierto. Las licencias compatibles con la GPL hacen posible combinar Python con otro software que se distribuye bajo la GPL [17].

La Comunidad Python Venezuela se constituye “ (...) con el objetivo de promocionar y difundir las bondades del lenguaje de programación Python” [18], definido por la comunidad como un lenguaje de programación poderoso y fácil de aprender, ideal para programación a nivel de comando (scripting) y de rápido desarrollo en diferentes áreas, compatible con la mayoría de las plataformas en virtud de contar con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. A dicha comunidad la identifica un trabajo colaborativo de aprendizaje, el impulso de iniciativas y el reclutamiento de programadores.

Otorgan acceso a sus desarrollos en la biblioteca estándar de Python en forma binaria y de código fuente para su libre distribución. Igualmente, se puede acceder a distribuciones y enlaces de módulos libres de Python de terceros, programas y herramientas y documentación adicional. La comunidad permite que el intérprete de Python pueda extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C) y el uso como lenguaje de extensiones para aplicaciones personalizables.

La comunidad proyecta su constitución en Fundación a objeto de cumplir con actividades de promoción del uso del lenguaje Python como herramienta de desarrollo, tales como el Pycon, que se constituye como el espacio de conferencia de desarrolladores, el PyDay, caracterizado por ser una jornada de iniciación al lenguaje y a algunas de las herramientas basadas en Python, el PyGouts, que permite el desarrollo de talleres interactivos vía Google Hangout de interés para los miembros de la comunidad y, el PyMeeting, que es el espacio ordinario de

reuniones para compartir conocimientos y conversaciones sobre los proyectos de la comunidad. Los colaboradores son incorporados al proyecto a través de redes sociales, vgr, LinkedIn (Python Venezuela en LinkedIn) y se denominan Pythonistas Venezolanos.

Por último, la Comunidad Ubuntu, surge basada en el Manifiesto Ubuntu que establece que el software es disponible sin costo alguno, con la posibilidad de usarlo en la “ (...) lengua materna del usuario independientemente de cualquier discapacidad” [19] y de adaptarlo y modificarlo de conformidad con los requerimientos del particular. El sistema operativo de Ubuntu se basa en Debian y está disponible en forma libre con soporte para la propia comunidad y otros colaboradores.

La Comunidad Ubuntu desarrolla el postulado bajo el cual se formó: “humanidad hacia otros” [19], a partir del reconocimiento público de: a) la gratuidad, inclusive en las denominadas “ediciones comerciales” asumiendo la calidad del producto ofrecido, b) la inclusión de traducciones y mayores opciones de accesibilidad para la comunidad de SL, c) liberaciones regulares y predecibles de las versiones que vayan surgiendo, liberándose cada seis meses la nueva versión, dejando a libertad del usuario la elección entre la versión estable o la versión en desarrollo, garantizando el correspondiente soporte técnico durante un plazo mínimo de dieciocho (18) meses, d) compromiso con el SL y las libertades que lo informan animando a los usuarios a su uso, mejora y compartimiento.

La comunidad se define en su Código de Conducta como “productiva, feliz y ágil” [20], ganada a las nuevas ideas y mejora de los procesos así como al fomento de la colaboración entre grupos, aún con requerimientos, intereses y habilidades diversas.

Del documento normativo de las conductas de los miembros, se destacan los principios de: a) consideración, en virtud del reconocimiento de las afectaciones que pudieran surgir en los usuarios por los trabajos de la comunidad; b) respeto, con la expresa admisión de las divergencias y resolución de conflictos, en aras de la productividad de la comunidad; c) responsabilidad por las palabras y las acciones; d) colaboración, en el reconocimiento del carácter holístico que debe caracterizar el producto; e) valor de decisión, claridad y consenso, en el entendido de la solución empática de los desacuerdos sin que se contaminen los proyectos sociales, dejando al árbitro designado la decisión sobre cualquier conflicto que pudiere surgir sin que exista acuerdo; f) receptividad y colaboración para con quienes tengan dudas; g) participación oportuna de abandono de proyectos con la consecuente responsabilidad de conducción del sucesor.

La comunidad reconoce el liderazgo emergente y la autoridad que de este surge, considerado dentro de la comunidad como un privilegio, una responsabilidad y un mandato, siempre en beneficio de la innovación y de los proyectos. En este sentido, reconocen la figura de lo que llaman un “dictador benevolente” quien es el encargado de la dirección y coordinación de los equipos de trabajo dentro de los proyectos. Igualmente, reconoce el adquirir méritos de los más capaces y comprometidos, la delegación en la toma de decisiones, la gobernanza y liderazgo de órganos superiores. Su órgano de dirección lo constituye el Consejo Comunitario, quien tiene a su cargo las nominaciones para las juntas y consejos, cuyo nombramiento contará con el visto bueno de la comunidad.

Otro aspecto importante para la comunidad es la sinergia, vista como las puertas al éxito, mediante la cual el líder como conductor del equipo demostrará su condición de tal si logra los objetivos. Se conducen bajo la máxima: “Un virtuoso es juzgado por sus acciones, un líder es juzgado por las acciones de su equipo” [20]. Un líder sabe cuándo actuar y cuándo dar un paso atrás, sabe cuándo delegar el trabajo y cuándo hacerse cargo él mismo. En Ubuntu, tratan los créditos del líder, el coraje y consideración de este y los conflictos de intereses que dentro del equipo pudieran surgir, siempre centrados en principios del liderazgo basados en la armonía y el consenso.

Todas las características de las CSL previamente expuestas, fueron registradas de acuerdo a las publicaciones en sus distintas páginas web, en el exterior y en Venezuela, sin embargo es imprescindible plantear la interrogante: ¿se encuentran los elementos descritos en el modelo de producción de software “Bazar”, la estructura clásica de producción de SL, la “Ética Hacker” y la dinámica de las prácticas de desarrollo de SL, en el quehacer de las CSL?

Estudio de factores conductuales que contribuyen al cultivo de prácticas virtuosas en el desarrollo de software libre

Fundamentación

Desde el Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL)⁴ se ha concebido que el desarrollo de tecnologías libres debe visualizar al conocimiento como libre más allá del acceso al mismo y de las cuatro libertades de la FSF, mediante elementos liberadores que permiten enriquecer el conocimiento a través de su construcción colaborativa, lo cual deriva en una búsqueda constante de la mejora de este y en una lucha contra las pretensiones de su mercantilización y privatización.

De acuerdo al modelo de desarrollo de SL planteado en la sección II del presente artículo, se consideran en este estudio aspectos que describen la práctica de desarrollo de SL, los cuales encuentran justificación en las definiciones de libertad del conocimiento y práctica virtuosa referidos en este trabajo. En este sentido, en la consulta realizada a CSL se indagó en aspectos como la interacción, la colaboración, el cumplimiento de políticas, el liderazgo, la motivación y la posible praxis de la “Ética Hacker” en las comunidades estudiadas. Dicha investigación se planteó con el propósito de identificar elementos a nivel organizacional y conductual que determinen los aspectos mencionados para el cultivo de conocimiento en colectivo, en pro del bien común.

Con relación a la colaboración se quiso conocer si las comunidades consultadas no solo comparten conocimiento sino que también lo construyen en colectivo. Por otro lado, la información sobre la interacción en las comunidades permitiría saber si la misma se corresponde con una estructura de producción de tecnologías descentralizada; donde el reconocimiento a los miembros se otorga por la búsqueda de la excelencia, los líderes orientan el camino hacia la mejora continua de la práctica de desarrollo tecnológico (no por ello toman todas las decisiones)

⁴Disponible en: <http://www.cenditel.gob.ve/>

y la motivación la encuentran en la producción de conocimiento como un bien de y para la comunidad.

De igual forma, se pretende observar si el cumplimiento de las políticas establecidas en las comunidades corresponde con el deber ser de la construcción de SL, así como también si su accionar coincide con el espíritu de la “Ética Hacker”, el cual se fundamenta en el compartir su competencia y pericia.

Con respecto a la visión de práctica virtuosa considerada por MacIntyre[11], previamente expuesta, se pretendió ver si los miembros de las comunidades estudiadas poseían ciertos valores éticos relacionados con dicha visión de práctica, como por ejemplo, si intentan lograr modelos de excelencia en conjunto con sus pares en lugar de logros individuales.

Variables en estudio

Para el presente estudio resultaron de interés las siguientes variables: interacción, colaboración, cumplimiento de las políticas definidas por la comunidad, liderazgo, motivación y “Ética Hacker”. Cada uno de estos aspectos se tratan a continuación desde el enfoque de desarrollo de SL.

Interacción: con base a la noción del individuo en sociedad y el análisis de sus acciones en términos relacionales, se destaca la interacción como uno de los aspectos o variable principal que sirve de fundamento a la práctica de SL dentro de las comunidades. Al respecto, Christakis y Fowler [21] indican:

(...) creemos que nuestras conexiones con otras personas son lo que más importa y que, al vincular el estudio de los individuos con el estudio de los grupos, la ciencia de las redes sociales tiene mucho que decir sobre la experiencia humana (p. 15).

Así, la interacción da cuenta de las relaciones entre los actores y del cómo dichas relaciones devienen en una estructura caracterizada por su dinamismo, en el que el motor de innovación tecnológica se encuentra representado precisamente por el intercambio de conocimientos entre desarrolladores y usuarios y por el intercambio de conocimientos entre usuarios.

Colaboración: en esta investigación no se establece diferencia entre los conceptos de colaboración y cooperación. El concepto colaboración encuentra su fundamento en el modelo de desarrollo de SL denominado “Modelo Bazar”, referido anteriormente, que se resume en tres (3) máximas: la liberación rápida y a menudo del software, la distribución de tareas y responsabilidades y la máxima apertura para la estimulación de la cooperación, lo que implica un entorno de libertad, cooperación y comunidad.

La novedad que introduce el software libre es que pone en funcionamiento un modelo de cooperación sin mando. No hay intereses empresariales o directos, es general intellect puro,

ingobernable y libre del mando⁵. Es más la ausencia de mando, de control corporativo o jerárquico, parece condición sine qua non: allí donde reaparece el mando —sea en forma de interés propietario, sea en su variante autoritaria—, el modelo se marchita, se agota y acaba por desaparecer (p. 7) [22].

Cumplimiento de políticas de la comunidad: las políticas en las CSL tienen como objetivo establecer las normas, procesos y actividades dentro de la comunidad para conseguir un modelo tecnológico autoregulado y sostenible basado en SL.

Liderazgo: dada la amplitud del tema relacionado con el liderazgo, a los fines de la investigación, se consideró el término a partir de la innovación propia del SL. En este sentido, el liderazgo se concibe como la facilitación de un proceso adaptativo y de asunción de riesgos, de superación de obstáculos.

Liderar no sólo significa motivar a la gente y movilizarla, ni basta con tener un gran sueño hacia el cual caminar. Se requiere eso y, además, actuar estratégicamente, para lo cual se ha de saber dónde estoy, para dónde voy y cómo voy a ir (. . .) Para ejercer liderazgo es necesario atreverse, quebrar el statu quo, desafiar los supuestos y las normas, romper la resistencia natural de la gente a cambiar, movilizar y escuchar (. . .) (pp. 59-60) [23].

Motivación: Turner [12], señala que investigaciones realizadas alrededor de la motivación de los actores dentro de las CSL y las redes de producción, indican entre las motivaciones más frecuentes las de índole ideológica y las de carácter técnico; sin embargo, no excluye la remuneración económica de las motivaciones.

Entre las primeras, se encuentran nociones e ideas sobre la libertad, el acceso libre al conocimiento, la solidaridad y la justicia social. Por otro lado, también son importantes la búsqueda de reconocimiento y pertenencia a una comunidad, la diversión y sensación de reto al resolver un problema específico, la posibilidad de aprender colaborando y de obtener algún beneficio, como puede ser una oportunidad laboral o un pago monetario (p. 213).

A los efectos del estudio, se consideran dentro de la motivación los aspectos citados por la referida autora.

“Ética Hacker”: el hacker, ha sido definido por Raymond, citado por Turner [12] como “(. . .) la persona que disfruta explorando los detalles de sistemas programables y en saber

⁵El autor alude al texto en el que Karl Marx recurre al término general intellect (o “intelecto general”) para designar el conjunto de los conocimientos abstractos (de “paradigmas epistemológicos”, diríamos hoy), que, al mismo tiempo, constituyen el epicentro de la producción social y organizan todo el contexto de la vida. Un “cerebro” o intelecto general, basado en la cooperación y el saber abstracto, incluyendo el saber científico, que tiende a volverse, en virtud precisamente de su autonomía en relación a la producción, ni más ni menos que la principal fuerza productiva, relegando a una posición marginal al trabajo parcelizado y repetitivo de la producción industrial.

cómo expandir sus propias capacidades, en oposición a la mayoría de los usuarios que prefieren sólo aprender lo mínimo necesario” (p. 148).

En esta investigación se propone la consideración de la “Ética Hacker” como el comportamiento asumido por la persona, en los términos descritos por Raymond [8], que se adhiere a una comunidad y a sus reglas tácitas o explícitas de la cual dimana un sentido de pertenencia.

Las comunidades hackers siguen en mayor o menor medida los ideales de libertad para compartir información y conocimiento. Las variaciones éticas entre estas comunidades se definirían por la radicalidad de sus posiciones con respecto a esta libertad y en contra del software privativo (p. 150).

Metodología

Para realizar este estudio se siguió un conjunto de acciones orientadas a describir las interacciones o “prácticas” que realizan diferentes grupos de trabajo dentro del quehacer tecnológico actual bajo el enfoque del conocimiento libre, específicamente el que tiene que ver con la producción, mantenimiento y promoción de programas para computadoras (software) y su relación con los procesos que estas herramientas atienden.

En primer lugar, se establecieron las líneas teóricas para diseñar los instrumentos utilizados en el estudio, las cuales también sirvieron para los procesos de descripción y análisis de resultados. En este sentido, teorías expuestas por MacIntyre, Himanen, Raymon, Cano y Turner fundamentaron el estudio. Igualmente, escritos de León y Contreras[6], referentes más cercanos a los investigadores, fueron consultados a efectos de construir una base suficientemente descriptiva de la investigación.

En la investigación se utilizaron los siguientes instrumentos: entrevistas y análisis de documentos publicados en páginas web. Las entrevistas fueron aplicadas a integrantes de la comunidad Wikipedia, así como a un grupo de personas que pertenecen a cuatro CSL en Venezuela: Ubuntu, Debian, Canaima y Python. Las mismas tuvieron como objetivo describir o determinar, según lo aportado por los entrevistados, las “prácticas” de cada comunidad. Por otro lado, para la comunidad Qt se realizó la revisión de documentos en páginas web⁶.

La entrevista fue de tipo estructurada⁷. A tales efectos, se elaboró una serie de preguntas que servirían para evaluar aspectos de interés relacionados con la práctica virtuosa en el SL, de acuerdo con criterios apreciados a partir del modelo de referencia descrito en la sección II. Las dimensiones del estudio estuvieron representadas por la interacción, la colaboración, el cumplimiento de políticas de la comunidad, el liderazgo, la motivación y la “Ética Hacker”.

La selección de los entrevistados fue realizada tomando como criterio a personas que son consideradas como miembros activos dentro de su comunidad, con logros verificables y que

⁶Disponible en: <http://qt\discretionary\-\}\}\project.org/>

⁷Disponible en: http://calidad\discretionary\-\}\}\sl.cenditel.gob.ve/files/2011/06/preguntasConsulta_2014.pdf

estuviesen dispuestos a brindar la información necesaria para el estudio. Algunas entrevistas se realizaron en forma presencial y otras en forma remota, utilizando en este último caso el correo electrónico o la videoconferencia. La técnica para la recolección, en el caso de la entrevista presencial y la videoconferencia, fue el audio.

Se aplicaron siete (7) entrevistas a personas relacionadas con las CSL para conocer sus visiones sobre las prácticas e interacciones humanas características de los procesos de producción de software.

Tomando como insumo la información obtenida en las entrevistas y bajo los criterios de prácticas expuestos en la sección II, se extrajeron los puntos considerados como significativos y que pueden contribuir a conformar una práctica humana basada en valores que mejoren los procesos y productos vinculados con el desarrollo de SL.

Finalmente, los resultados se analizaron tomando como elemento central el planteamiento de una lista de recomendaciones a ser utilizadas por grupos de trabajo de SL. Estas recomendaciones tienen el objetivo de servir como herramientas de uso constante en las actividades que realizan los grupos de trabajo de SL en su día a día, con lo cual se busca lograr la mejora de los niveles de calidad de los productos intermedios y finales que estos construyen.

Resultados

Se definieron indicadores para cada dimensión mencionada en la metodología.

Los indicadores de la dimensión interacción fueron: los logros, los fracasos, los roles, las prácticas grupales, las normas de interacción, los modos de relación entre los miembros y entre estos y los usuarios, los conflictos y sus modos de resolución, las expresiones de solidaridad y fraternidad, los modos de transferencia del conocimiento, la actitud con respecto al proceso enseñanza-aprendizaje y la importancia del desarrollo, la investigación, la promoción de las tecnologías y la discusión de ideas en torno al desarrollo y uso de las tecnologías.

La dimensión colaboración se evaluó a través de indicadores basados en ejemplos expuestos por los entrevistados en torno al trabajo colaborativo y la facilitación del proceso colaborativo mediante la documentación y el diseño del software, la práctica de la libertad, la colaboración, la solidaridad y la fraternidad, la actitud con respecto al proceso enseñanza-aprendizaje.

El objetivo general de la comunidad, los mecanismos de castigo, el conflicto y su resolución, sirvieron de indicadores para la evaluación del cumplimiento de las políticas de la comunidad.

El liderazgo se analizó mediante la dirección de las acciones de la comunidad, los parámetros para liderar proyectos y la claridad del entrevistado con el objetivo de la comunidad.

La motivación la definieron los aspectos relacionados con el desarrollo y la publicación del conocimiento, así como la importancia del desarrollo, la investigación, la promoción de las tecnologías y la discusión en torno al desarrollo y uso de las tecnologías.

Por último, para evaluar la “Ética Hacker”, se consideró la opinión de los entrevistados en torno a la presencia de sus elementos distintivos: trabajo voluntario, entusiasta, activo, búsqueda de conocimiento como medio de compartir; tanto en los integrantes de la comunidad

como solo en las personas que la dirigen y la pasión y contribución con el éxito de los proyectos.

El análisis de los resultados obtenidos se encuentra disponible en: <http://tinyurl.com/gvlkkvg>

Sin embargo, podemos mencionar brevemente algunos de estos resultados: en la dimensión interacción se encontró que en la mayoría de las comunidades en estudio (Ubuntu, Canaima, Qt, Wikipedia, Python, Debian) se consideraron logros; la ampliación de la comunidad de usuarios, el desarrollo de algunas herramientas en el país y la organización de eventos para difundir no solo las distintas distribuciones sino el sentido de usar SL. Como fracasos, dichas comunidades plantearon: la falta de organización y no continuidad de los colaboradores por diversas controversias entre estos. En cuanto a los roles, en su mayoría se encuentran divididos por tareas o áreas de desempeño. En las prácticas grupales, mencionaron las reuniones virtuales continuas y las reuniones periódicas cada cierto tiempo, las discusiones tanto técnicas como políticas a través de los chats o foros, entre otros. Las normas de interacción, como en el caso de Ubuntu se presentan en un Código de conducta y en otros casos, se establecen de manera consuetudinaria, como el procurar un ambiente armónico entre los miembros.

Por otro lado, en la dimensión colaboración como ejemplos en torno al trabajo colaborativo y la facilitación de dicho proceso mediante la documentación y el diseño del software la mayoría de las comunidades, algunos de los entrevistados dijeron que la utilización de plataformas para la construcción en conjunto de documentación y control de versiones de un sistema es un factor que contribuye significativamente al trabajo colaborativo. En cuanto a práctica de la libertad, la colaboración, la solidaridad y la fraternidad argumentaron seguir la concepción que se tiene de libertad en movimientos como la Free Software Foundation y Open Source, colaborar en cuanto a construcción de conocimiento (documentación y código) en conjunto, ser solidarios y fraternos aunque en algunos casos se planteó la falta de estas dos virtudes, como en el caso de la comunidad Python. En la actitud con respecto al proceso enseñanza-aprendizaje, describieron que se da de manera informal en foros de discusión, eventos, entre otros, donde los miembros con más experiencia de las comunidades van guiando a los más nuevos.

En el cumplimiento de las políticas de la comunidad, como objetivo general de la misma se obtuvieron diversas respuestas: primero, el contribuir a la libertad del conocimiento, compartir conocimiento, luego el mantenimiento de las distintas distribuciones y el dar a conocer su respectiva comunidad. En lo referente a los mecanismos de castigo, en su mayoría dijeron que quien comete una falta se expulsa de los foros y chats de discusión y si la falta ha sido muy grave, se expulsa de la comunidad. Ante cualquier conflicto (suelen darse por diferencias entre algunos miembros y búsqueda de protagonismo), se trata de resolverlo a través del diálogo, código de conducta (en el caso de Ubuntu), apelando a la jerarquía de roles, foros de discusión, etc.

Ahora bien, en la dimensión liderazgo la dirección de las acciones en algunas comunidades la dirigen los miembros que tienen roles más destacados como los desarrolladores con más tiempo en la comunidad, es decir, el liderazgo se establece por rol y en otras comunidades depende del área de trabajo (diseño gráfico, codificación, etc.). En otros casos, como Python y Debian se argumentó la horizontalidad y el consenso en la toma de decisiones. Como parámetro para

liderar proyectos, mencionaron la experiencia acumulada, aportes técnicos, la capacidad para sumar usuarios y colaboradores al proyecto, es decir, capacidades no sólo técnicas sino políticas.

En la dimensión de motivación, se evidenciaron algunas coincidencias en las respuestas como el considerar la publicación y el compartir conocimiento como principal elemento motivador para contribuir en la comunidad, aportar a las 4 libertades del SL, luego el obtener reconocimiento por sus pares, el trabajo colaborativo, entre otros. En cuanto a la importancia del desarrollo, la investigación, la promoción de las tecnologías y la discusión en torno al desarrollo y uso de las tecnologías, algunos argumentaron que en su comunidad se le otorga importancia a los cuatros procesos por igual, otros priorizaron la investigación y el desarrollo, otros la promoción.

Finalmente, con lo que respecta a la dimensión “Ética Hacker” cuando se solicitó opinión en torno a la presencia de sus elementos distintivos (trabajo voluntario, entusiasta, activo, búsqueda de conocimiento como medio de compartir) en los integrantes de la comunidad o solo en las personas que la dirigen, la mayoría de los entrevistados consideraron que los miembros de sus comunidades poseen dichos principios éticos y solo algunos miembros no. Les apasiona lo que aprenden en comunidad y así como también, lo que aportan a la misma como una contribución a un proyecto común.

En esta sentido, se puede connotar que en algunos casos encontramos coincidencia entre lo que se planteó como descripción teórica de las variables en estudio y en otros no. Claro está, las entrevistas fueron realizadas a un representante por cada comunidad, así que no se puede dar por sentado si en opinión de otros miembros hay coincidencia entre lo que se considera ideal como práctica virtuosa en el SL (teoría) y lo que en realidad se presenta en el día a día de las comunidades (praxis).

Recomendaciones a las comunidades para el cultivo de prácticas virtuosas en el desarrollo de tecnologías libres

Con base al estudio realizado, se plantea a continuación una serie de recomendaciones dirigidas a mejorar la práctica de desarrollo de SL en las distintas comunidades donde se desarrollen este tipo de aplicaciones, las cuales abarcan aspectos del comportamiento que podrían influir positivamente en el logro de una práctica de trabajo colaborativo que tribute al bien común.

Interacción

Para lograr niveles de interacción que favorezcan el trabajo colaborativo y el esfuerzo conjunto en torno al desarrollo de SL se debe prestar especial atención a:

- La organización de eventos en los que se lleven a cabo actividades más allá de la promoción del uso del SL, en especial, actividades de producción de conocimiento y prototipos de software, mediante las cuales se fomente el ejercicio del trabajo que implica el desarrollo de software, permitiendo a su vez formar colaboradores que puedan sumarse a la comunidad.

- La generación de incentivos que favorezca el aumento de la participación en las prácticas de desarrollo de SL.
- El apoyo constante en la superación de obstáculos que se puedan presentar durante el ejercicio de la práctica de desarrollo, el cual debe ir orientado en la identificación oportuna de tales obstáculos, así como en procesos de enseñanza-aprendizaje que permitan superarlos. Este apoyo requiere de seguimiento continuo tanto de las actividades de la práctica de desarrollo como de sus productos intermedios y finales. Para la superación de obstáculos es fundamental el ejercicio de la fraternidad entre los miembros de un equipo de desarrollo, en tanto que, esta crea lazos de unión que permiten brindar apoyo en la enseñanza de mejores prácticas que contribuyan a superar estos obstáculos.
- El reconocimiento y el llevar a cabo las mejores prácticas conlleva a obtener buenos resultados en el desarrollo de software. En este caso el reconocimiento de la competencia y pericia en los miembros de un equipo de desarrollo, representa un incentivo de significativa importancia que no solo permite reconocer en el otro su trabajo, sino que promueve la mejora continua de la práctica de desarrollo, en tanto que, quienes reciben este reconocimiento deben ser los que dirijan los procesos de enseñanza-aprendizaje entre los integrantes de un equipo de desarrollo. El hecho de reconocer en el otro sus habilidades y aprender de estas, requiere el ejercicio del respeto y la humildad por parte de quien reconoce y aprende. En este sentido, el respeto se gana en función de los méritos obtenidos por las contribuciones que se realicen al desarrollo de aplicaciones de software y la humildad, se ejerce cuando se es capaz de reconocer en el otro mejores formas de hacer software, que valen la pena aprender para mejorar el quehacer.
- El establecimiento de roles organizacionales que favorezcan la toma de decisiones efectiva en el desarrollo de las actividades programadas.
- La utilización de herramientas de apoyo a las prácticas grupales que sean de fácil uso y que permitan cubrir las necesidades de interacción de los equipos de trabajo.
- La transparencia en el quehacer de las prácticas grupales, lo cual implica la publicación de las decisiones que se tomen junto con su fundamentación, así como de los productos que se generen. Para que tal transparencia ocurra es necesario la honestidad entre los miembros del equipo de desarrollo.
- La existencia de normas expresas dirigidas a mantener el orden entre los integrantes de la práctica de desarrollo y a facilitar el trabajo colaborativo.
- El respeto entre los miembros de la práctica de desarrollo es un elemento esencial para la convivencia entre estos.
- Los encuentros presenciales, aunque no son muy recurrentes en prácticas de desarrollo a distancia como los presentes en las comunidades de SL, deberían llevarse a cabo con más frecuencia, a fin de fortalecer la relación entre sus miembros.

- Los conflictos en las comunidades tienden a generarse por visiones distintas, de tal manera que, para abordarlos, se recomienda el diálogo continuo que permita el entendimiento, basado este en el respeto por las posturas distintas.
- La fraternidad y la solidaridad constituyen principios esenciales para la vida en comunidad, por tanto deberían ser estos aspectos del comportamiento que se destaquen entre los integrantes de la práctica de desarrollo de SL. Estos principios junto al ejercicio del amor al prójimo, que implica el empleo de la libertad para hacer público el conocimiento y ofrecer las herramientas para mejorarlo libremente, tributan a un proceso que busca mejorar la vida en comunidad, basado en la enseñanza y aprendizaje de todo aquello que permita cultivar el bien común.
- La publicación de los modos como se llevan a cabo las aplicaciones de software resulta esencial para una verdadera transferencia del conocimiento.
- El monitorear las tareas que ejecutan los miembros de la práctica de desarrollo es fundamental para reforzar el proceso de enseñanza-aprendizaje.

Colaboración

En la construcción del conocimiento como bien común, el trabajo colaborativo constituye un elemento fundamental, por lo tanto:

- Para facilitar la colaboración en el quehacer de la práctica de desarrollo, es imprescindible un proceso de apropiación orientado no solo al uso del software sino a la forma en que este se desarrolla, lo cual implica generar documentación de análisis y diseño que facilite la comprensión del software y, por ende, su mejora.
- La claridad en el planteamiento del objetivo trascendental de la organización que desarrolla software, resulta fundamental para definir la concepción de colaboración y libertad a la cual se debe dicha organización.
- Para mejorar la calidad de los productos que se aportan desde un proceso colaborativo de desarrollo de software, es necesario el seguimiento del proceso de enseñanza-aprendizaje que se da entre los integrantes de dicho proceso.

Políticas

Toda organización que desarrolla SL debe contemplar, dentro de sus políticas organizacionales, ciertos elementos que tienen un peso significativo para mantener el orden y el equilibrio en las relaciones que surgen entre sus miembros, entre estos elementos se mencionan:

- Compartir el conocimiento generado como filosofía de vida.

- Generar conocimiento para el bien común.
- Establecer mecanismos de castigo, de manera equitativa, siempre y cuando se fundamente y comprueben las causas de su aplicación.
- Asumir los errores en los que se incurra en la práctica de desarrollo, pues ello facilita la corrección de los mismos, y mejora las relaciones entre los miembros de la práctica.

Liderazgo

El liderazgo en la práctica de desarrollo, así como en otro tipo de prácticas, resulta fundamental no solo para orientar la toma de decisiones, sino para fortalecer el proceso de enseñanza-aprendizaje dentro de la práctica, dado que como líder se debe dar ejemplo, tanto en comportamiento como en lo que respecta al conocimiento referido al quehacer de la práctica. En este sentido, se indican algunas recomendaciones a tomar en cuenta por los líderes dentro de la práctica de desarrollo:

- Las decisiones deberían ser tomadas a través del consenso y la discusión de ideas, lo cual genera un liderazgo compartido.
- El líder debe ser modelo de excelencia en la práctica de desarrollo de SL, enalteciendo ciertas cualidades, como por ejemplo: poder de convocatoria, iniciativa, motivación, aporte técnico, respeto por los otros, comunicación, participación efectiva en el proceso de enseñanza-aprendizaje, etc.

Motivación

La colaboración depende en gran medida de la motivación que tengan los que colaboran en la realización de una práctica. Teniendo ello en consideración se presentan algunas sugerencias con respecto a este asunto:

- El poner a disposición del público lo que se crea en la práctica de desarrollo debería ser la principal motivación de toda organización que desarrolla SL.
- El reconocimiento entre pares constituye otra fuente importante de motivación.
- Otro tipo de motivación podrían ser los incentivos económicos por la buena labor realizada, al igual que otro tipo de premios o reconocimientos que generen entusiasmo entre los colaboradores.

Ética Hacker

La “Ética Hacker” constituye un importante elemento dentro del movimiento de SL, en tanto que, gracias a esta filosofía de trabajo el conocimiento es percibido como motor de toda actividad productiva, que tiene como fin el bienestar de la vida en comunidad. En este sentido, existen algunas características de un Hacker que deben ser tomadas en cuenta como modelo a seguir por los integrantes de una práctica de desarrollo de SL, estas son:

- La búsqueda de la excelencia en la práctica de desarrollo, lo cual implica ir más allá de lo establecido cumpliendo con un modelo de mejora continua.
- La pasión por lo que se hace, lo cual se percibe como una actitud a seguir para contribuir con la comunidad, siempre que esta tribute al trabajo en equipo.

Consideraciones finales

Desde el origen del SL, que se remonta a inicios de los años setenta, se evidencian los principios sobre los que descansan los fundamentos de las CSL: la libertad y la colaboración, desde la práctica de desarrollo de software, pilares de la construcción del conocimiento para el logro del bien común.

Más allá de otorgar acceso a la producción intelectual, es importante la construcción de esta en colectivo, generándose un espacio propicio para el desarrollo de relaciones de fraternidad, lo cual se observó en la descripción de los modelos de producción de software propuestos por Raymond, el “Modelo Catedral” y el “Modelo Bazar”, cuyas características principales son el modo de organización del trabajo, que en el primero conduce a la planificación centralizada, al desarrollo a puerta cerrada y, en el segundo, a la planificación descentralizada, en virtud de un trabajo colaborativo.

Para el estudio de factores conductuales que contribuyen al cultivo de prácticas virtuosas en el desarrollo de SL, fue importante conocer cómo en algunas de las CSL en Venezuela, Comunidad Debian Venezuela, Comunidad Canaima GNU/Linux, Comunidad Python Venezuela, Ubuntu Venezuela, se visualizan las relaciones entre sus miembros y entre estos y los usuarios; a tales efectos, se realizó investigación documental y se aplicaron entrevistas. Para el análisis de estas últimas, se establecieron unidades de análisis a partir de las palabras o expresiones de los entrevistados, las cuales se categorizaron para evaluar la práctica llevada a cabo en algunas de las CSL que participan en procesos de construcción del software.

La unidad de análisis estuvo enfocada en ciertas categorías, a criterio de los investigadores, en relación con los logros y fracasos; los roles; las prácticas grupales; las normas de interacción; los modos de relación entre los miembros y entre estos y los usuarios; los conflictos y sus modos de resolución; las expresiones de solidaridad y fraternidad; los modos de transferencia del conocimiento; la actitud con respecto al proceso de enseñanza-aprendizaje; la importancia del desarrollo, la investigación, la promoción de las tecnologías y la discusión de ideas en torno al desarrollo y uso de las tecnologías; los diversos ejemplos en torno al trabajo colaborativo y

la facilitación del proceso colaborativo mediante la documentación y el diseño del software; la práctica de la libertad, la colaboración, la solidaridad y la fraternidad, el objetivo general de la comunidad; los mecanismos de castigo; el conflicto y su resolución; la dirección de las acciones de la comunidad; los parámetros para liderar proyectos; la claridad del entrevistado con el objetivo de la comunidad; el desarrollo y publicación del conocimiento; la opinión en torno a la presencia de sus elementos distintivos (trabajo voluntario, entusiasta, activo, búsqueda de conocimiento como medio de compartir) en los integrantes de la comunidad o solo en las personas que la dirigen y, la pasión y contribución con el éxito de los proyectos.

Los indicadores previamente mencionados, formaron parte de las dimensiones desde las cuales se posibilitó la definición de factores conductuales que contribuyen al cultivo de prácticas virtuosas en el desarrollo de SL: la interacción, la colaboración, el cumplimiento de las políticas definidas por las comunidades, el liderazgo, la motivación y la “Ética Hacker”. Los resultados obtenidos sirvieron como fundamento para las recomendaciones.

Destacaron en los resultados la contribución y la organización como medios para los logros; por el contrario, la imposibilidad y la ausencia de comunicación se constituyen en obstáculos; por consiguiente, en motivos que conllevan al fracaso de cualquier proyecto emprendido o por emprenderse.

En relación a la estructura de producción en proyectos de SL, se refleja la importancia de la distribución de los roles para cumplir con los objetivos propuestos dentro de una comunidad.

La comunicación y el uso de herramientas se constituye en paradigma sobre el que se fundamentan las prácticas grupales. Igualmente, se acentúan el respeto y el reconocimiento como normas de interacción que se integran a los códigos de conducta observados en algunas de las CSL.

Los ejercicios de desarrollo colaborativo se convierten en modos de transferencia del conocimiento y, la libertad, la colaboración la solidaridad y la fraternidad en principios que informan a las CSL que, junto con la pasión con el éxito de los proyectos, se conjuga para dar sentido a lo que se hace.

Resalta la “Ética Hacker”, descrita por Himanen, como una cultura propia de los hackers la cual define modos de actuar en los que destaca la transferencia de la competencia y la pericia en la práctica de desarrollo, así como también la pasión por el quehacer, el reconocimiento de sus semejantes, el cultivo de la creatividad y el amor por la comunidad, conductores hacia la liberación del conocimiento.

El concepto de práctica virtuosa de MacIntyre fue de interés para el estudio realizado, en tanto que, en la búsqueda de la excelencia del bien que se produce, se requieren ciertas cualidades que posibilitan la presencia de los factores conductuales analizados en la práctica de desarrollo de SL, que podrían llegar a constituir las bases en las que se cimentan los principios de libertad y colaboración.

Del análisis de los conceptos y elementos mencionados en este artículo se puede concluir que una práctica de desarrollo colaborativo de SL, que implica procesos de enseñanza-aprendizaje que buscan el bien común, tanto con la liberación del software como con el conocimiento implícito en su creación, demanda en el ser humano el ejercicio de conductas tales como

la fraternidad, el respeto, la humildad, el trabajo, la honestidad, la libertad y el amor, que constituyen virtudes.

Es así como, en la construcción colectiva del conocimiento que tributa al bien común, se requiere de la fraternidad entre las personas para compartir conocimiento, además del respeto y humildad para reconocer en el otro capacidades que se pueden aprender. Adicionalmente, se necesita del trabajo constante que implica el “aprender haciendo” y la mejora continua de este hacer, así como de la honestidad con los otros para brindar transparencia en todo el proceso de generación de conocimiento, lo cual favorece el aprendizaje y la confianza en dicho proceso.

Finalmente, se requiere de amor a nuestros semejantes, reflejado en la búsqueda incesante del bien común, razón principal por la cual se mejora y libera el conocimiento. En este caso, dicha liberación implica adicionalmente el ejercicio de la libertad en pro del bienestar de la humanidad.

Como se puede apreciar una práctica virtuosa en el desarrollo de SL, demanda en sus integrantes no solo el ejercicio y mejora continua de habilidades técnicas, sino, con mayor énfasis, el ejercicio de virtudes como las mencionadas en este trabajo, que permiten al ser humano ejercer una lucha constante por alcanzar el bien común.

Bibliografía

- [1] Cano, A. (2010). Nuevas formas de valorización del conocimiento en el esquema de realización de ganancia del capital. Caso: Asociación Mexicana Empresarial del Software Libre. Tesis para obtener el grado de Maestro en Ciencias Sociales, FLACSO, México. Disponible en: <http://tinyurl.com/grfxvzg> [Consulta: 2014, octubre 14].
- [2] Stallman, R. (1983). El Manifiesto de GNU. Disponible en: <https://www.gnu.org/gnu/manifiesto.es.html> [Consulta: 2014, octubre 8].
- [3] Stallman, R. (2010). El Proyecto GNU. Disponible en: <https://www.gnu.org/gnu/thegnuproject.es.html> [Consulta: 2014, octubre 8].
- [4] Free Software Foundation (2009). ¿Qué es el software libre? Disponible en: <https://www.gnu.org/philosophy/free-sw.es.html> [Consulta: 2014, octubre 14].
- [5] Free Software Foundation (2014). Licencias. Disponible en: <https://www.gnu.org/licenses/licenses.es.html> [Consulta: 2014, octubre 14].
- [6] León, L. (2008). Socialismo y Software Libre en Reflexiones desde Cenditel. Vol. 2. Tecnología Socialista, pp. 37-76. Disponible en: <http://www.cenditel.gob.ve/node/415> [Consulta: 2014, octubre 14].
- [7] Open Source Initiative. (2012). About the Open Source Initiative. Disponible en: <https://opensource.org/about> [Consulta: 2014, octubre 14].

- [8] Raymond, E. (2001). The Cathedral and the Bazaar. Disponible en: <http://www.understein.net/su/docs/CathBaz.pdf> [Consulta: 2014, octubre 14].
- [9] Himanen, P. (2001). La ética del hacker y el espíritu de la era de la información. Disponible en: <http://eprints.rclis.org/12851/1/pekka.pdf> [Consulta: 2014, noviembre 20].
- [10] Contreras J. (2009). Una aproximación al sentido de los Movimientos por el Software Libre Contra la Mundialización de la Sociedad de Consumo, en rescate de la Sociedad Liberal. Disponible en: <http://tinyurl.com/jdcxjbh> [Consulta: 2014, noviembre 20].
- [11] MacIntyre, A. (2004). Tras la virtud. A M Gràfic, S.L., Santa Perpetua de Mogoda (Barcelona). Versión digital. Disponible en: <http://tinyurl.com/nf9ofoc> [Consulta: 2014, noviembre 20].
- [12] Turner, T (2012). Software Libre y abierto: comunidades y redes de producción digital de bienes comunes. Tesis propuesta como cumplimiento de los requisitos para la Maestría en Estudios Políticos y Sociales . UNAM. México. Disponible en: <http://flosshub.org/sites/flosshub.org/files/Tesis.pdf#page=1&zoom=auto,-187,515> [Consulta: 2014, octubre 14].
- [13] Decreto N^o 3.390. (2004). Gaceta Oficial de la República Bolivariana de Venezuela, 336.626, diciembre, 28.
- [14] Ley de INFOGOBIERNO. (2013). Gaceta Oficial de la República Bolivariana de Venezuela, 40274, octubre 17.
- [15] Portal Canaima GNU/Linux. (2010). Celebrado el 9no Día Debian en la ciudad de Barquisimeto en conmemoración de su 20 Aniversario. Disponible en: <http://tinyurl.com/jxym2gv> [Consulta: 2014, noviembre 8].
- [16] Portal Canaima GNU/Linux. (2014). Canaima GNU/LINUX. Disponible en: <http://canaima.softwarelibre.gob.ve/> [Consulta: 2014, noviembre 8].
- [17] Python Software Foundation. (2014). History and License. Disponible en: <https://docs.python.org/3/license.html> [Consulta: 2014, noviembre 8].
- [18] Python Venezuela. (2014). Python Venezuela. Disponible en: <http://pyve.github.io/> [Consulta: 2014, noviembre 8].
- [19] Ubuntu Venezuela. (2014a). Ubuntu. Disponible en: <http://www.ubuntu.org.ve/ubuntu> [Consulta: 2014, noviembre 8].
- [20] Ubuntu Venezuela. (2014b). Código de Conducta. Disponible en: <http://www.ubuntu.org.ve/?q=codigodeconducta> [Consulta: 2014, noviembre 8].

- [21] Christakis, N. A., Fowler, J. H. (2010). Conectados. El sorprendente poder de las redes sociales y cómo nos afectan. México: Editorial Taurus.
- [22] Vidal, M. (2000). Cooperación sin mando: una introducción al software libre. Disponible en: <http://www.trasversales.net/i62mvidal.htm> [Consulta: 2014, noviembre 17].
- [23] Barrera y otros. (2010). Programa Compartim. El trabajo colaborativo en la administración. Documento en línea. Disponible en: <http://tinyurl.com/gpyh1lm> [Consulta: 2014, noviembre 20].