
**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE OCCIDENTE**

Reconocimiento de Validez Oficial de Estudios de Nivel Superior según Acuerdo Secretarial
15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976

DEPARTAMENTO DE ESTUDIOS SOCIOCULTURALES
MAESTRÍA EN COMUNICACIÓN DE LA CIENCIA Y LA CULTURA



Procesos de comunicación en el desarrollo de software libre.

El caso de Moblin v2.

Tesis que para obtener el grado de
Maestro en Comunicación de la Ciencia y la Cultura

Presenta

Lic. Esteban Enael Contreras Vázquez

Director de tesis: Mtra. Magdalena López de Anda

Tlaquepaque, Jalisco. Noviembre 2010

Abstract

El software libre refiere a un software donde, distintos tipos de usuarios tienen la posibilidad de participar en el proceso productivo y los desarrolladores retroalimentarse con la colaboración de la comunidad que cultivan a su alrededor. Los 32 trabajos revisados, coinciden en señalar que los usuarios forman lazos entre sí y con desarrolladores, organizaciones no lucrativas, empresas y otro tipo de instituciones y personas. En esta investigación, se abordará esta relación para explorar: qué caracteriza a los procesos de comunicación empleados en el desarrollo de Moblin v2 desde tres ejes de análisis: a) herramientas de comunicación y de trabajo; b) modelos de trabajo y; c) necesidades de los participantes.

Agradecimientos

En primer lugar, este trabajo no hubiera sido posible, por un lado; sin la existencia del programa de la Maestría en Comunicación de la Ciencia y la Cultura y Estudios Socioculturales del ITESO, de sus profesores y compañeros. En especial se extiende el agradecimiento a, la confianza y apoyo brindados por el Dr. Raúl Fuentes Navarro y; la inspiración constante y seductora de la Dra. Rossana Reguillo. Por el otro lado, este trabajo no existiría sin el apoyo económico (beca) que el CONACYT brinda a los alumnos de posgrado registrados en el Padrón Nacional de Posgrados.

En segundo lugar, es de agradecer la inagotable paciencia de la directora de esta tesis, la Mtra. Magdalena López de Anda, que tuvo que soportar: múltiples cambios de enfoque durante las primeras etapas del proyecto de tesis y; retrasos crónicos en las entregas, a lo largo de las etapas subsecuentes.

En tercer lugar, la labor de LinuxCabal A.C. y de sus participantes, que aceptaron jugar el rol de; primero, objetos de estudio y, después; asesores y guías en el mundo del Software Libre, todos los sábados y sin goce de sueldo, desde mediados del 2008 en que esta tesis comenzó.

En cuarto lugar y no al último, un agradecimiento especial a la familia de Esteban (osea yo), que durante ya hace mucho tiempo ha lidiado con que, haya estudiado Sociología y luego Comunicación! Por fortuna, Elena Vázquez Larumbe (mi mamá) ha extendido una línea de crédito sólida, constante y puntual. Por otra parte, se agradece el ejemplo impecable brindado por Vicente Contreras Arteaga (mi papá), siempre amable, sereno y firme en sus creencias.

Indice

1	Introducción.....	6
1.1	Software Libre.....	6
1.2	Moblin.....	8
1.3	Esta investigación (Ruta a seguir).....	8
2	Moblin, contexto de emergencia y objetivos.....	11
2.1	Contexto.....	11
2.2	Alianzas comerciales, competencia y convenios.....	13
2.3	Moblin v2.....	16
3	Representar el software libre desde las ciencias sociales.....	18
3.1	Punto de partida, reconocer legados.....	19
3.2	Perspectiva demográfica y motivacional.....	19
3.3	Perspectiva organizacional.....	21
	Investigaciones descriptivas.....	22
	Investigaciones teóricas.....	23
	Trabajos que recuperan a las comunidades de práctica.....	23
	Trabajos que recuperan elementos de política y gobernanza.....	24
	Balance de la perspectiva organizacional.....	25
3.4	Perspectiva Económica.....	26
3.5	Ley, Cultura y Sociedad.....	28
3.6	Conclusiones a la representación del software libre desde las ciencias sociales.....	30
4	Lentes conceptuales.....	32
4.1	Comunidades de práctica (CdP).....	33
4.2	Comunidades de práctica en el contexto de la comunicación mediada por computadora.....	36
4.3	Redes de comunicación y trabajo en el desarrollo de software libre.....	39
4.4	Modelo de trabajo Bazar, principales características.....	41
	Acceso.....	41
	Economía del regalo.....	43
	Meritocracia.....	44
4.5	Interpretación de necesidades subyacentes.....	45
4.6	Balance de los lentes conceptuales.....	49

5	Objetivo, pregunta, método.....	50
5.1	Objetivo.....	50
5.2	Pregunta.....	50
5.3	Método.....	51
	Primera etapa: Observación como punto de partida para el análisis.....	53
	Segunda etapa: Análisis de contenido (AdC).....	54
	Desglose de variables.....	55
	Forma expresiva.....	56
	Propósito.....	56
	Flujos de conversación.....	58
	Estructura de la conversación.....	59
6	Primera etapa. Observación como punto de partida para el análisis.....	63
6.1	Moblin.org (sitio web).....	64
6.2	Moblin-dev (lista de correo).....	66
6.3	Moblin-Bugzilla.....	69
6.4	Moblin-Git.....	72
6.5	Lanzamientos y puntos de encuentro entre las herramientas de comunicación y trabajo.....	75
7	Segunda etapa. Conversaciones.....	78
7.1	Sujetos.....	78
	Equipo central.....	78
	Vendedores, desarrolladores de segundo nivel y participación legítima.....	79
	Vendedores.....	80
	Desarrolladores de segundo nivel.....	81
	Participantes legítimos.....	82
	Usuarios finales y participantes periféricos.....	82
7.2	Estructura interna de los mensajes.....	83
	Reglas explícitas.....	83
	Prácticas (reglas implícitas).....	84
7.3	Conversaciones y desarrollo de Moblin v2.....	88
	Temas tratados.....	88
	Temas y desarrollo de Moblin.....	91

Interacción y construcción de sentido.....	94
Etapa 1: Forma expresiva de la acción inicial.....	94
Etapa 2: Intercambios de información.....	96
Etapa 3: Argumentación.....	97
Etapa 4: Construcciones grupales.....	97
Etapa 5: Cierres.....	100
Flujos de conversación.....	101
7.4A manera de cierre.....	103
8Conclusiones generales.....	104
8.1Procesos de comunicación en el desarrollo de Moblin v2.....	104
8.2Pendientes.....	107
9Bibliografía.....	107
9.1Bibliografía Básica.....	107
9.2Sitios de donde se extrajo la información para la investigación.....	113
Sobre Moblin.....	113
Sobre los sistemas de información.....	113
Sobre los participantes.....	113
9.3Bibliografía Complementaria.....	113

1 Introducción.

1.1 Software Libre.

El software libre es un término utilizado para hacer referencia a un software registrado bajo licencias públicas que siguen el modelo de la licencia pública GNU-GPL. Por otra parte, el término es también utilizado para hacer referencia al movimiento de productores de software que promueven prácticas, como, acompañar a los programas con su código fuente y; considerar a los usuarios de su software, como co-desarrolladores (Raymond 2000).

El uso que se hace en esta tesis con el término "software" incluye dos sentidos; el primero, hace referencia a un programa de computadora listo para utilizarse, por ejemplo, el navegador web Firefox tal como puede ser descargado de Internet, que se instala en la computadora y se usa. El segundo sentido hace referencia al código fuente de los programas, escrito en lenguajes de programación. Para diferenciar estos dos sentidos del término software, cuando se hace referencia al código fuente, se utilizará la palabra "código" y; cuando se hace referencia a un programa listo para usarse, se utilizan las palabras "programa o aplicación".

Para entender al software libre como modelo de trabajo, es conveniente identificar a los principales actores y parte de su historia. Primero, el software libre engloba a dos movimientos tan similares que es difícil separarlos. El primero, es el movimiento del software libre, encabezado por el proyecto GNU y la figura de Richard Stallman; el segundo es el *Open Source Software*, encabezado por Linux y la figura de Linus Torvalds.

El proyecto GNU es formulado por Richard Stallman, a finales de la década de 1970, cuando el mercado toma un papel activo en el desarrollo de innovaciones informáticas; esto contempló un proceso de desregulación y privatización de los desarrollos realizados en Universidades y Centros de Investigación, financiados en conjunción por el gobierno de los Estados Unidos de América y empresas. Al entrar los bienes informáticos en el mercado, el software dejó de ser considerado un bien público y se convirtió en un bien privado, propiedad de alguien y sujeto a licencias de propiedad intelectual y patentes comerciales. En palabras de Williams (2001), Richard Stallman actuó al momento en que su estilo de vida y prácticas de trabajo fueron desestructuradas, en especial, por las políticas de ocultamiento del código, por parte de las empresas.

GNU fue formulado como un proyecto para comenzar desde cero, toda la serie de bienes que fueron privatizados por las empresas en la década de 1970, en especial, el sistema operativo UNIX. Para asegurarse de que ningún actor pudiera volver a arrebatarse sus desarrollos, Stallman formuló un

tipo de licencia pública conocida como GNU-GPL (*GNU Not Unix-General Public Licence*), la cual, sirvió como parteaguas para que otros sujetos adoptaran sus prácticas de trabajo y juntos desarrollaran un proyecto independiente.

Con la llegada de la World Wide Web (www) a principios de la década de 1990, el movimiento del software libre conoció un nuevo impulso: Linux, el primer proyecto de gran envergadura de desarrollo de software, realizado con la colaboración voluntaria de cientos de personas alrededor del mundo, registrado bajo la licencia pública GNU-GPL v.2. Linux significó una revolución en las formas de imaginar el desarrollo de software y su influencia ha sido tal, que sus prácticas de trabajo fueron tomadas como modelo, por la GNU y por muchos otros proyectos. Interesa señalar que Linux, entendido como modelo de trabajo, es institucionalizado con la *Open Source Initiative* (OSI) y el término *Open Source Software* (OSS).

El *Open Source Software* (OSS), tiene dos sentidos de importancia para el conocimiento de las prácticas de trabajo del software libre. El primer sentido, es relacionado con el modelo de trabajo utilizado para desarrollar Linux, es decir, un modelo de trabajo que utiliza la participación de voluntarios (usuarios) para desarrollar, mejorar y mantener al software. La primera formulación del OSS como modelo de trabajo, fue plasmado en el trabajo "La Catedral y el Bazar" (*The Cathedral and the Bazaar*) de Erick Raymond (2000, primera versión de 1997). Este trabajo presenta la experiencia de Raymond, al utilizar las prácticas que aprendió de Linux, para el desarrollo de uno de sus proyectos de software libre. Según su perspectiva, este modelo considera a los usuarios como co-desarrolladores, para lo cual, hace accesibles por medio de Internet los informacionalismos (código, documentación, herramientas, manuales) necesarios para que participen, junto con canales de comunicación y prácticas, que integren sus propuestas, resuelvan sus dudas y en general, los tomen en cuenta; de forma que los usuarios se motiven a participar, socializar y, constituirse como un grupo, equipo, red o, comunidad (de práctica) alrededor del proyecto.

El segundo sentido del OSS, es formulado por la *Open Source Initiative* (OSI) en 1998; el cual, puede ser entendido como un intento por distanciarse del término software libre, para dar paso, a un término cuyo objetivo es el gran mercado informático. Esta propuesta, surge de la necesidad de un término que no confunda a los empresarios con libertad y gratuidad y, que deje claro que su propuesta es, una nueva forma de hacer software, de mayor calidad a menor precio. Cabe señalar que uno de los principales objetivos de la OSI, es el dar a conocer su propuesta, para lo cual financiaron una campaña publicitaria en medios de comunicación masiva electrónicos, donde se ocultó al software libre y, se

resaltó al OSS como opción comercial.

Respecto a la diferencia entre software libre y OSS; por un lado, el software libre se relaciona con un conjunto de principios ideológicos y legales sobre la libertad del software; por otro lado, el término OSS surge como una propuesta para comercializar el software libre, ya que el término *free software*¹, confundía a los posibles clientes. La controversia surge cuando no hay elementos técnicos o legales para diferenciar al software libre y al OSS, debido a las mezclas, hibridaciones y mutuas dependencias entre ambos movimientos. Por lo observado en la bibliografía consultada sobre el tema², cada investigador usa el término software libre, *open source* o la combinación *Free and Open Source Software* (FOSS); sin presentar una justificación clara. Para el caso de ésta tesis, se hará uso del término software libre, con la aclaración de que también contempla la propuesta del OSS.

1.2 Moblin

El caso en el cual se sitúa esta tesis, es el proyecto Moblin, el cual, es un sistema operativo desarrollado por Intel, para promover su familia de procesadores *Atom*. En términos más precisos, Moblin es una distribución GNU & Linux; es decir una pluralidad de proyectos de software libre, realizados por distintos actores sociales, unidos para dar forma a un sistema operativo; el concepto de proyectos sirve para diferenciar a los distintas instituciones, comunidades, grupos; con pluralidad de posturas; licencias públicas y no públicas; formas de organización y; de intereses; que convergen en las distribuciones GNU & Linux, las cuales, son desarrolladas con ese objetivo, el de acoplar los distintos elementos disponibles, en un nuevo proyecto, para satisfacer necesidades específicas.

Como recuento, en esta tesis se usa el término software libre, el cual incluye también al *Open Source*, que juntos, forman las distribuciones GNU & Linux; por distribuciones se entiende a un sistema operativo registrado casi por completo con licencias públicas compatibles con la GNU-GPL. A su vez, Moblin, es una distribución GNU & Linux, formada por una multiplicidad heterogénea de proyectos, donde intervienen actores como empresas, organizaciones no lucrativas, comunidades, Estados, universidades, voluntarios, entre los más importantes.

1.3 Esta investigación (Ruta a seguir)

El objetivo de la investigación es realizar una caracterización, de los procesos de comunicación que

1 En Inglés *free* significa al mismo tiempo "libre" y "gratis", lo cual se presta a malos entendidos; de allí que sea común la expresión "*free as in free speech, not as in free beer*" (libre como en libertad de discurso, no como en cerveza gratis).

2 Que se expone en el apartado "Representar al software libre desde las ciencias sociales" de esta tesis.

tienen lugar, en el desarrollo de Moblin v2, desde la articulación de: las herramientas de comunicación y de trabajo que, posibilitan y condicionan las relaciones entre los sujetos; los modelos de trabajo y; las necesidades de los participantes.

Las *herramientas de comunicación y de trabajo*, son vinculadas con dos nociones, la primera, es *red híbrida* de Sack et.al. (2006) y Ducheneaut (2005), que contempla, que las herramientas informáticas utilizadas en el desarrollo de un proyecto de software libre, soportan una red de relaciones entre sujetos (participantes), sistemas de información o herramientas y, contenidos (informacionalismos). Derivado de la noción de red híbrida, se considera que, un proyecto de software libre es soportado en una red de sistemas de información o herramientas de comunicación y de trabajo, que contemplan distintos sistemas de información independientes desde lo técnico e; interconectados desde lo organizacional, dicha red se compone de: sitios web; listas de correo; canales de chat (IRC); wikis; sistemas de administración de errores (Bugzilla); sistemas de control de versiones (Git) y; sistemas de administración de traducciones (Transifex).

La segunda noción, es la de *informacionalismos*, de Elliot y Scachi, los cuales, son definidos como, conjunto de "recursos, objetivados en información digital, accesibles por medio de la Internet cuya utilidad reside en ser soporte para usuarios y desarrolladores" de software libre (2003:5). Los recursos que componen a los informacionalismos, incluyen código, herramientas, manuales, guías, documentación, discusiones, enlaces, entre los más relevantes.

Para definir los *modelos de trabajo* del software libre, se recurre al modelo de trabajo Bazar desarrollado por Erick Raymond (2000), el cual es caracterizado por: la distribución de los niveles de acceso en los sistemas de información; la economía del regalo y; la meritocracia, como forma de gobierno. Las configuraciones entre estos elementos se orientan para promover, la construcción del usuario como co-resarrollador, mediante la apertura a los recursos del proyecto y, (teóricamente)voz y voto en la toma de decisiones que afectan al proyecto.

Las *necesidades de los participantes* es relacionado con el concepto de *interpretación de necesidades subyacentes* de Habermas (2001), el cual, problematiza al sujeto desde sus intereses, deseos, gustos, ambiciones y; los modelos, códigos, normas y valores que sigue o promueve. Recurrir al concepto de necesidades subyacentes, permite por un lado; superar el concepto de las motivaciones y; por el otro, considerar al sujeto en un equilibrio contingente, subjetivo e intersubjetivo con sus deseos, los de otros sujetos, instituciones, normas y valores compartidos.

El tejido conceptual utilizado en esta tesis, para dar coherencia a esta serie de elementos, es el

de las comunidades de práctica desarrollado en, Lave y Wegner (1991), Wenger (2001) y, Wenger y McDermott (2002). Según esta perspectiva, un proyecto de software libre puede ser entendido como una comunidad de práctica que contempla, como uno de sus elementos más importantes, el cultivo de una comunidad, compuesta por distintos tipos de participantes, como usuarios, voluntarios, vendedores y patrocinadores; en el proceso de desarrollo y uso del software; en este sentido, el organigrama del proyecto, puede ser representado como un modelo de círculos concéntricos en cuyo centro hay un grupo central (*core team*) de desarrolladores, encargados de la organización del proyecto, ellos sitúan las condiciones de partida, los códigos y las herramientas, son quienes poseen mayor conocimiento sobre el proyecto; quienes realizan la mayor parte del trabajo y; quienes tienen mayor acceso a los recursos de la comunidad. Alrededor de éste núcleo, gravitan con distintos grados de adscripción, compromiso y niveles de acceso a las herramientas y recursos; participantes, voluntarios, patrocinadores y usuarios; que participan de distintas formas e intensidades, con el proyecto.

Es de señalar que el tejido conceptual de las comunidades de práctica, involucra un reordenamiento de los papeles y jerarquías de los actores involucrados; este reordenamiento plantea que el *core team*, para cultivar comunidad, debe de construir roles (llamadas identidades) ligados a distintos niveles de acceso, obligaciones y derechos; para los sujetos que los asuman y, se sientan parte de Moblin.

A los conceptos que conforman las comunidades de práctica, en esta tesis se les da más importancia a, participación; periferalidad, legitimidad de la participación y; acceso. A partir de estos conceptos se pretende: concebir a los sujetos como participantes; situar los diferentes roles, adscripciones y niveles de compromiso entre los participantes y; situar las relaciones de poder objetivadas en los niveles de acceso en las herramientas de comunicación y de trabajo de Moblin.

En lo que refiere a la metodología, el proceso se dividió en cuatro etapas distintas. La primera, consistió en la revisión hemerográfica del proyecto Moblin para conocer por quién, cómo, bajo qué condiciones y con qué objetivos surge el proyecto; se consultaron revistas y sitios web como *Wired*, *Ars Technica*, *Linux Journal*, *PC World*, *PC Magazine*; también, blogs de expertos y personas relacionadas con innovación tecnológica y, en particular, dispositivos móviles de Internet.

La segunda etapa, es una revisión panorámica de los historiales de participación registrados en la red de herramientas de comunicación y de trabajo de Moblin v2. Por principio, se definen cuáles son los sistemas de información utilizados; se exploran sus características; se observan sus usos; los informacionalismos producidos; los participantes; los niveles de participación y, su relación con las

fechas de lanzamientos del proyecto.

La tercera etapa, consiste en la observación de los procesos de comunicación que tuvieron lugar en la lista de correo Moblin-dev, para lo cual se utilizó la técnica del Análisis de Contenido de Krippendorf (1990). En esta etapa se observaron los temas tratados; la dinámica de las conversaciones; las reglas y prácticas en la redacción de mensajes y; los procesos de construcción social de sentido sobre Moblin y su uso, por parte de los participantes.

Para complementar, se planeó una cuarta etapa, la cual consideraba realizar un seguimiento de las relaciones que construyen los sujetos dentro y fuera de la red de comunicación y de trabajo de Moblin, con el objetivo de, construir una caracterización de las necesidades de los sujetos; para lograrlo se planteó reconstruir trayectorias de participación y entrevistas semi-estructuradas por *chat* (IRC) a participantes. Al final, esta etapa no fue realizada.

2 Moblin, contexto de emergencia y objetivos.

2.1 Contexto

El proyecto Moblin surge en el primer cuarto del año 2007 impulsado por Intel Corp. como parte de la promoción de su familia de micro-procesadores Atom; cuya característica más importante, es su reducido tamaño, su precio bajo y limitada capacidad en el procesamiento de datos³. Moblin es el nombre corto de "*Mobile Linux*", un *sistema operativo libre*, para *netbooks*, MID's (*Mobile Internet Devices*), *smartphones* e IVI's (*In-Vehicle Infotainment* o, info-entretenimiento para automóviles).

El proyecto Moblin comienza con el lanzamiento de la *Mobile Linux Initiative* por Intel, su propuesta es construir un software atractivo y funcional orientado a satisfacer la demanda de los usuarios, no por computadores más poderosos, sino, por lo que Tim Bajarin de *PCMagazine* denomina, "Internet en tu bolsillo"⁴. Este concepto plantea que los usuarios quieren acceder a Internet desde cualquier lugar en el que se encuentren, con dispositivos ligeros, que les permitan realizar operaciones sencillas como usar su correo electrónico y navegar por Internet.

La entrada de Intel en el mercado de estos nuevos dispositivos de Internet móvil, se desarrolla a la sombra de Nokia y su *Nokia 770 Internet Tablet* que aparece en el mercado en el 2005; este dispositivo permite conexión a Internet mediante *wireless*, navegar por Internet, consultar correo electrónico, abrir imágenes, escuchar música y radio por Internet. El sistema operativo que la Nokia

3 Ver: <http://linuxfordevices.com/c/a/News/Intel-launches-mobile-Linux-initiative/> consultado el 15 de Noviembre de 2009.

4 Ver: <http://www.pcmag.com/article2/0,2817,2115380,00.asp> consultado el 15 de Noviembre de 2009.

770 utiliza software libre, que se adapta a los estándares de la distribución *GNU & Linux Debian*; su interfaz de usuario retoma las librerías de *GNOME* y el estándar *GTK* con el manejador de ventanas *Matchbox*; el Kernel que controla el procesador *ARM-Texas Instruments* es la versión 2.6 de Linux. El nombre dado a este conjunto de aplicaciones es la distribución *GNU & Linux Maemo* y, para ser desarrollado, *Nokia* lanzó el proyecto *Maemo* y el sitio web *maemo.org*, como una forma de articular una comunidad de desarrolladores a la manera que, otros proyectos de software libre lo han realizado⁵.

La entrada de Intel en el mercado de la Internet de bolsillo, se ha caracterizado por una fuerte competencia con otros fabricantes de procesadores como "*AMD, Via, Freescale, y Qualcomm*"⁶; por otro lado, existen además de *Maemo* otros proyectos de software libre basados en Linux para MID's, entre los cuales resalta: *Google Android* y *LiMo (Linux Mobile)*. Esta doble competencia ha llevado a un rezago de Intel, más claramente observable en el plano de los MID's y de los *smartphones*, donde hasta finales del 2009 no había uno de estos dispositivos con procesadores de la familia Atom⁷. En este contexto Intel ha optado por algunas estrategias comerciales articuladas alrededor del software libre, para hacer frente a este plano de competencia, por ejemplo, promover el uso de procesadores Atom en *netbooks* apoyado en el uso de software libre⁸; adquirir empresas desarrolladoras de software libre para MID's como *Opened Hand*⁹; apoyarse en la legitimidad y prestigio de la Fundación Linux para promocionar Moblin entre los desarrolladores¹⁰ y; realizar alianzas con gobiernos para desarrollar tecnologías de software libre, como la realizada con el gobierno de Taiwan¹¹.

Recurrir al software libre para promocionar el desarrollo de la infraestructura lógica de estos dispositivos de Internet móvil, se muestra como una tendencia, *Motorola, Nokia, Google* y otras

5 Ver: <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Building-consumer-products-with-open-source/> consultado el 15 de Noviembre de 2009.

6 Ver: <http://www.rahulsood.com/2009/01/intel-atom-vs-via-nano-vs-netbook.html> consultado el 15 de Noviembre de 2009.

7 Ver: <http://www.wired.com/gadgetlab/2009/07/atom-processor-phones/> consultado el 15 de Noviembre de 2009.

8 Ver: <http://www.linuxfordevices.com/c/a/News/First-Atom-based-notebook-runs-Linux/> consultado el 15 de Noviembre de 2009.

9 Ver: http://www.freesoftwaremagazine.com/columns/interview_matthew_allum_openedhand consultado el 15 de Noviembre de 2009.

10 Ver: <http://arstechnica.com/open-source/news/2009/04/linux-foundation-to-host-development-of-intels-moblin-os.ars> consultado el 15 de Noviembre de 2009.

11 Ver: <http://linuxpundit.wordpress.com/2008/10/30/intel-and-taiwan-inc-partner-for-oss-research-wimax-rollout/> y <http://www.linuxfordevices.com/c/a/News/Intel-Taiwan-tagteam-Moblin-WiMAX/> consultados el 15 de Noviembre de 2009.

empresas lo están haciendo; según la interpretación de Ari Jaaksi, responsable del proyecto *Maemo*, en el caso del *Nokia 770*¹², el uso de software libre responde a una necesidad por contar con software dinámico y adaptable que responda a costos competitivos para su modificación y evolución. El software libre, visto como un ecosistema compuesto de varios componentes, que permiten combinarse (Watson et al. 2008), es utilizado como base ideal para trabajar, una característica que resalta Jaaksi de este ecosistema, es que permite la selección en las licencias entre proyectos similares para combinarlas, con otros desarrollos que no tienen por fuerza que ser libres o abiertos; esta practicidad se complementa con los beneficios que tiene el abrir el proyecto al mundo y cultivar una comunidad que se enriquezca con propuestas, detección de errores y, promuevan el consumo en el mercado. El potencial de esta propuesta contempla que para el 2013 un 23% del mercado de los *smartphones* estará utilizando Linux¹³, según esta perspectiva, el software libre estará llegando al usuario final por medio de los MID's y los *smartphones*¹⁴.

2.2 Alianzas comerciales, competencia y convenios.

En Abril de 2007 Intel lanza una nueva línea de procesadores llamada *Ultra Mobile Platform 2007*, orientada a las *Ultra-Mobile PC* (UMPC) y, los MID's; la diferencia entre ambos tipos de dispositivos, es que los primeros están diseñados siguiendo la lógica de una PC que utiliza como sistema operativo principal Windows XP o Vista y, busca que los sujetos puedan trabajar en dispositivos más poderosos y de mayor precio que los MID's. Por su parte, los MID's son creados siguiendo la lógica de la *Nokia 770 Internet Tablet*, que tiene funciones limitadas, son más pequeños, de menor precio y, orientadas al entretenimiento por sobre el trabajo¹⁵. Los primeros MID's que Intel desarrolló utilizaban un sistema operativo simplificado basado en Linux para cubrir cuatro necesidades del usuario: estar en contacto; estar entretenido; estar informado y localizado y; ser productivo. Las herramientas que se contemplaron para cubrir estas necesidades combina elementos de software libre y, de software propietario, a tal grado que resulta difícil para los propios desarrolladores distinguir cuál es cuál¹⁶.

12 Ver: <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Building-consumer-products-with-open-source/> consultado el 15 de Noviembre de 2009.

13 Ver: <http://arstechnica.com/open-source/news/2008/06/23-of-smartphone-market-to-be-linux-powered-by-2013.ars> consultado el 15 de Noviembre de 2009.

14 Ver: <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Building-consumer-products-with-open-source/> consultado el 15 de Noviembre de 2009.

15 Ver: <http://www.pcmag.com/article2/0,2817,2115380,00.asp> consultado el 15 de Noviembre de 2009.

16 Ver: <http://www.linuxfordevices.com/c/a/News/Intel-debuts-Linuxbased-Mobile-Internet-Device/> consultado el 15 de Noviembre de 2009.

En Julio de 2007 Intel lanza la *Mobile Linux (Moblin) Initiative* para promocionar sus procesadores, con esto pretendía que Moblin fuera un punto de integración entre proyectos similares, su modelo de desarrollo de software libre estaba compuesto por distintas partes Kernel Linux; *GTK-based UI* (Interfaz de usuario basada en GTK); Firefox; Gstreamer Helix y; *Moblin Image Creator (MIC)*¹⁷. En esta fase del proceso la comunidad de la distribución GNU & Linux *Ubuntu*, colabora en la integración de los elementos.

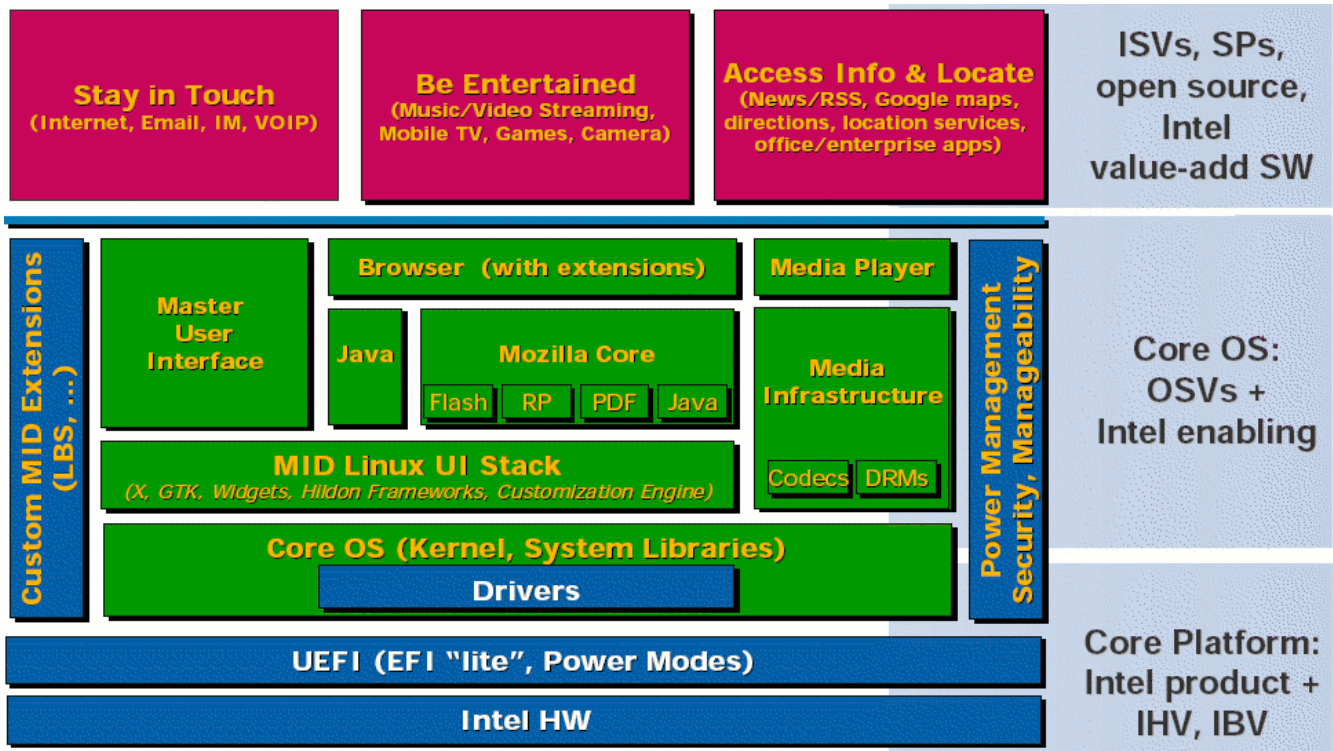


Imagen 1 extraída de <http://www.linuxfordevices.com/c/a/News/Intel-debuts-Linuxbased-Mobile-Internet-Device/>

En Julio 2007 se crea la página moblin.org que albergaría al proyecto, esto obedece una lógica de integración de una comunidad a partir del sitio web; las intenciones de Intel es convertir a Moblin en un parteaguas para la comunidad que trabaja con aplicaciones basadas en procesadores Atom¹⁸. Para inicios de 2008 Nokia adquiere a *Trolltech ASA*, el cual desarrolla las librerías de la interfaz de usuario

17 Ver: <http://www.linuxfordevices.com/c/a/News/Intel-debuts-Linuxbased-Mobile-Internet-Device/> consultado el 15 de Noviembre de 2009.

18 Ver: <http://arstechnica.com/open-source/news/2007/07/intel-launches-site-for-open-source-mobile-linux-development.ars> consultado el 15 de Noviembre de 2009.

Qt, el mayor competidor de GTK¹⁹, que utiliza Moblin; por su parte, *Verizon Wireless* se convierte en miembro del centro de desarrollo de la fundación *LiMo*²⁰.

Intel, por su parte, participa en el lanzamiento de la primera *netbook* que sale al mercado, la "Asus Eee PC" en Enero de 2008, la computadora utiliza el procesador "Intel Celeron mobile-M ULV 900MHz" y la distribución GNU & Linux *Xandros*²¹. Para Mayo, Intel concreta una alianza con *Wind River*, para que la segunda, desarrolle un software para sistemas de info-entretenimiento para automóviles (IVI's) con Linux y procesadores de la familia Atom. En palabras de Bryan Thomas Director de *Global Communications* de *Wind River*, los IVI revolucionarán la forma en que la tecnología es usada en el automóvil²². Este lanzamiento se nutre además de Linux, de la plataforma gráfica GNOME y del "*GTK-based Hildon framework*"²³. Siguiendo la línea de lanzamientos de Intel, para el mes de Junio de 2008 se lanza al mercado la primera *netbook* que utiliza uno de los procesadores de la familia Atom, la *Acer Aspire One*, cuya primera versión utilizaba la distribución GNU & Linux *Linpus Linux Lite*²⁴ de la empresa taiwanesa *Linpus Technologies*, la cual, en ese mismo mes de Junio se une al proyecto Moblin²⁵.

A la popularidad que ganan las *netbooks*, Ubuntu, que colaboraba en este momento con Moblin lanza su distribución *Ubuntu Remix* especial para *netbooks*, compatible con procesadores Intel y ARM²⁶; para el mes siguiente, Moblin deja de lado a *Ubuntu* y se acerca con la comunidad del proyecto *Fedora*²⁷, en parte, por cuestiones legales con *Microsoft* y por los buenos resultados obtenidos con la

19 Ver: <http://linuxpundit.wordpress.com/2008/02/> consultado el 15 de Noviembre de 2009.

20 Ver: <http://www.linuxjournal.com/content/verizon-joins-linux-mobile-devices-limo> consultado el 15 de Noviembre de 2009.

21 Ver: http://www.freesoftwaremagazine.com/columns/asus_eee_pc_ultra_portable_laptop_gnu_linux_pre-installed consultado el 15 de Noviembre de 2009.

22 Ver: <http://www.windriver.com/news/press/pr.html?ID=6041> consultado el 15 de Noviembre de 2009.

23 Ver: <http://arstechnica.com/open-source/news/2008/05/intel-wants-to-pimp-your-ride-with-prodigious-penguin-power.ars> consultado el 15 de Noviembre de 2009.

24 Ver: <http://www.linuxfordevices.com/c/a/News/First-Atom-based-notebook-runs-Linux/> consultado el 15 de Noviembre de 2009.

25 Ver: <http://www.linuxfordevices.com/c/a/News/Linux-distributor-joins-Moblin/> consultado el 15 de Noviembre de 2009.

26 Ver: <http://arstechnica.com/hardware/news/2008/06/hands-on-with-the-ubuntu-netbook-remix.ars> consultado el 15 de Noviembre de 2009.

27 Ver: <http://www.linuxfordevices.com/c/a/News/Moblin-switching-from-Ubuntu-to-Fedora/> consultado el 15 de Noviembre de 2009.

Linpus Linux Lite que esta basada en *Fedora*²⁸. Moblin en este punto comienza a desarrollar una versión para MID's²⁹.

Durante el periodo de Junio a Octubre de 2008 hay señalamientos sobre la gran demanda que está teniendo el procesador Atom, debido a su bajo precio y buen rendimiento, al grado de hacer *laptops* con procesadores Atom³⁰. En Septiembre de 2008 Intel adquiere al desarrollador de software libre *Opened Hand*. Para Octubre se hace público el acuerdo entre Intel y MOEA (*Ministry of Economic Affairs*) de Taiwan, cuyo objetivo es construir un centro de desarrollo para Moblin, que también investigue en VMAX, un operador de telefonía móvil. Por su parte el gobierno de Taiwan contempla desarrollar una red WiMAX³¹.

2.3 Moblin v2.

En Enero 2009 se libera a través de la página moblin.org la versión *Alpha* de Moblin v2, el último desarrollo que integra todos los movimientos empresariales de Intel del 2008, la nueva versión, retoma los avances de la versión 1 y, basa buena parte de sus herramientas y formatos en la distribución GNU & Linux *Fedora*, en lugar de *Ubuntu*; el corazón de Moblin, según los términos de su sitio web³², está construido sobre *GNOME Mobile* enriquecido con *Clutter* y *GUPnP* de *Opened Hand*.

28 Ver: <http://www.linuxfordevices.com/c/a/News/Linux-MID-stack-to-debut-at-IDF/> consultado el 15 de Noviembre de 2009.

29 Ver: <http://arstechnica.com/open-source/news/2008/08/linux-everywhere-mobile-internet-devices-and-the-cloud.ars> consultado el 15 de Noviembre de 2009.

30 Ver: <http://arstechnica.com/hardware/news/2008/08/limited-atom-supply-shows-strong-netbook-demand.ars> y <http://www.infoworld.com/t/hardware/atom-supply-still-stymied-testing-bottleneck-346> consultado el 15 de Noviembre de 2009.

31 Ver: <http://arstechnica.com/open-source/news/2008/10/intel-taiwan-teaming-up-on-mobile-linux-development-lab.ars> consultado el 15 de Noviembre de 2009.

32 Ver: <http://moblin.org/documentation/moblin-overview/moblin-core> consultado el 15 de Noviembre de 2009.

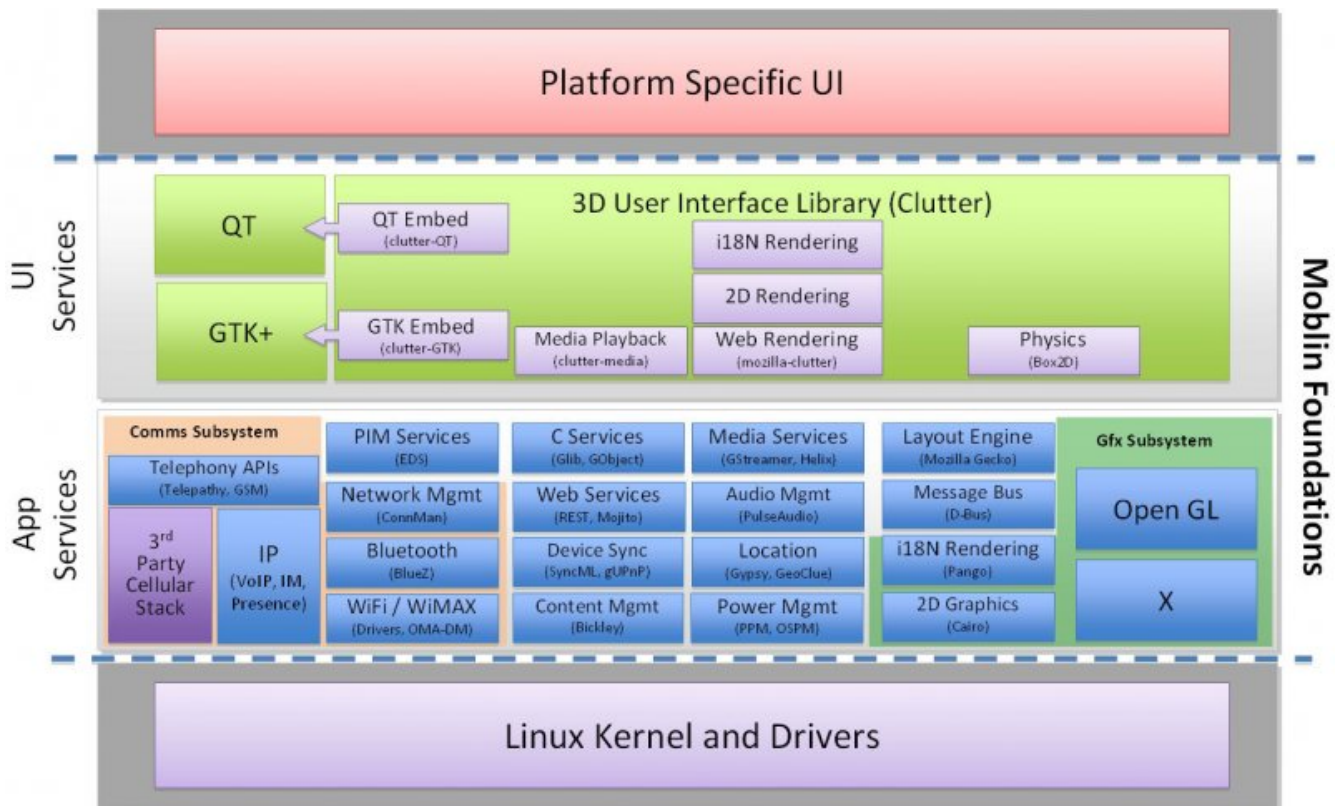


Imagen 2: Moblin Core Arquitectura extraída de: <http://moblin.org/documentation/moblin-overview/moblin-core>.

Esta versión *Alpha* (o versión en desarrollo) de Moblin v2 estuvo orientada a *netbooks*, su lanzamiento buscó articular una comunidad alrededor de ella, objetivo en parte logrado cuando otros distribuidores de GNU & Linux como "*Mandriva*", "*Novell*", "*Open Suse*", "*Linpus*" y "*GoS*" adoptan Moblin v2 para sus distribuciones³³. Una de las características que distingue a Moblin v2 de otros sistemas operativos, es su rapidez para cargar (*fast-boot*), que según estimaciones ronda los 5 segundos³⁴, también; se distingue por proponer una interfaz para el usuario pensada en una lógica de pantallas pequeñas que requieren un aprovechamiento máximo del espacio, donde no se busca que el usuario sustituya su computadora de escritorio o su *laptop* por una *netbook*³⁵, aun y los usuarios así lo quieran, ya que las

33 Ver: <http://arstechnica.com/open-source/news/2009/01/intel-releases-linux-based-moblin-2-alpha-for-netbooks.ars> consultado el 15 de Noviembre de 2009.

34 Ver: <http://www.linuxfordevices.com/c/a/News/Intel-Taiwan-tagteam-Moblin-WiMAX/> consultado el 15 de Noviembre de 2009.

35 Ver: <http://www.rahulsood.com/2009/01/intel-atom-vs-via-nano-vs-netbook.html> consultado el 15 de Noviembre de 2009.

características de una *netbook* no permite realizar operaciones demasiado complicadas.

Uno de los acuerdos de mayor trascendencia, es que Intel consiguió una alianza con la Fundación Linux para el desarrollo de Moblin v2, la propuesta, en palabras de Jim Zemlin, director ejecutivo de la Fundación Linux, es servir como espacio de desarrollo neutro que estimule la participación de terceras personas; por medio del alojamiento del proyecto en sus servidores y, la toma de ciertos criterios para administrar la colaboración, por ejemplo, uno de estos criterios es la utilización de GIT (*Global Information Tracker*), desarrollado por Linus Torvalds, como el SCM (*Source Code Manager*) de Moblin v2, este anuncio se hizo público en el mes de Abril de 2009³⁶.

La tendencia de Intel para entrar en el mercado de los MID's y de los smartphones sigue en la lista de las prioridades de Moblin v2 y en Febrero de 2009 se ve reforzada al asociarse con LG para el lanzamiento de un MID³⁷ y; en Junio de 2009 establecer un acuerdo con Nokia para desarrollar de manera conjunta un *Intel chipset architecture* orientado a telefonía móvil, de igual forma, Intel adquirió de Nokia la licencia IP del modem "Nokia HSPA/3G" que será usado en futuros productos de Intel. En Septiembre de 2009 Intel da a conocer una versión simplificada de Moblin diseñada para MID's y *smartphones* y en Noviembre una diseñada para ser utilizada en IVI's.

La relación con Nokia va más allá del solo hardware y ambas compañías lanzan una propuesta para unir, Intel Moblin y Nokia Maemo en un solo producto llamado MeeGo³⁸, contemplado para ser utilizado en *netbooks*, *smartphones*, *laptops* y prácticamente cualquier tipo de dispositivo que utilice un procesadores miniatura de AMD e Intel. Con este proyecto la actividad de Moblin v2 se da por finalizada y los esfuerzos son concentrados en MeeGo, que en Abril de 2010 libera su primera versión³⁹.

3 Representar el software libre desde las ciencias sociales.

La propuesta de esta sección es construir un estado de la cuestión, de investigaciones académicas, que permita contextualizar, el análisis de los procesos de comunicación que tuvieron lugar en el proceso productivo de Moblin v2. Para ello, se toman cuatro distintas perspectivas de análisis del software libre,

36 Ver: <http://arstechnica.com/open-source/news/2009/04/linux-foundation-to-host-development-of-intels-moblin-os.ars> consultado el 15 de Noviembre de 2009.

37 Ver: <http://www.wired.com/gadgetlab/2009/02/new-lg-netbook/> consultado el 15 de Noviembre de 2009.

38 Ver: <http://arstechnica.com/open-source/news/2009/06/new-intelnokia-partnership-a-huge-win-for-mobile-linux.ars> consultado el 02 de Mayo de 2010.

39 Ver: <http://arstechnica.com/open-source/news/2010/04/meego-gets-going-source-code-and-developer-builds-available.ars> consultado el 02 de Mayo de 2010.

que reproducen la organización de la compilación de Lerner et.al. (2005); éstas perspectivas son: motivacional; organizacional; económica; ley, cultura y sociedad.

El recorrido tiene un doble objetivo, primero, ubicar actores y tendencias de cada una de las perspectivas, segundo, recuperar los elementos que cada perspectiva aporta al estudio de la comunicación mediada por computadora.

3.1 Punto de partida, reconocer legados.

El lector recordará que al inicio de esta tesis, se menciona que se hace uso del término, software libre con la aclaración de que también hace referencia al término *Open Source Software* (OSS). También, que los autores que tratan el tema, hacen referencia a uno u otro, y también, al término que los une: el *Free and Open Source Software* (FOSS). La razón principal por la cual, en esta tesis se usa el término software libre, es para subrayar que la licencia pública dominante en las distribuciones GNU & Linux (incluida Moblin v2), es la GNU-GPL de la *Free Software Foundation*. Puede decirse, que las investigaciones que abordan el lado comercial u organizacional del software libre, utilizan el término OSS y, aquellas que abordan aspectos ideológicos o de valores, conservan el término software libre; sin embargo, esto no ocurre en todas las ocasiones. Para dar elementos para que el lector formule sus propias conclusiones al respecto, se conservará en la revisión de cada trabajo, el término al cual recurre, con la aclaración de que, en las generalizaciones, se usará el término software libre.

3.2 Perspectiva demográfica y motivacional.

En el año 2000 en Harvard y en el *Journal First Monday*, aparecieron los primeros trabajos (Ghosh 2005; Lerner y Tirole 2000; Lakhani y Wolf 2003 y; Hars y Ou 2001) que abordaron al software libre desde el campo de la demografía y la psicología; por medio de encuestas realizadas a desarrolladores y participantes, los objetivos de estos primeros trabajos, fueron, resolver la interrogante por las motivaciones que tienen los sujetos por donar tiempo y desarrollos a proyectos de software libre; un segundo objetivo, fue sentar bases estadísticas que permitieran conocer a los sujetos involucrados en el desarrollo del software libre.

Uno de los trabajos más influyente en esta perspectiva es Lerner y Tirole (2000), su aporte es mostrar que los participantes voluntarios en proyectos de OSS dedican tiempo y esfuerzo a esta actividad porque así, obtienen beneficios como, reconocimiento de la comunidad, apoyos brindados por empresas a sus empleados para participar en proyectos en horarios laborales o, el dominio de nuevas habilidades. El prestigio juega un papel importante, ya que es considerado como un elemento que

permite mejorar las condiciones de vida de quién lo posee.

Un segundo trabajo es “La Encuesta Orbiten” del 2000 (Ghosh 2005), realizada a 2,280 participantes de distintos proyectos de *Free Libre Open Source Software* (FLOSS). Su aporte fue construir una figura estadística de este sujeto colectivo, que diera cuenta de sus edades, niveles escolares, lugares de residencia y sus motivaciones para participar de manera voluntaria en los proyectos. Sus resultados apuntan a que la mayoría son hombres, con edad promedio de 27 años; poco menos de la mitad se declara soltero; un tercio cursaron estudios universitarios, una proporción igual menciona maestría y un décimo doctorado; casi un cuarto es estudiante y más de la mitad tiene un trabajo relacionado con tecnologías de la información. Las motivaciones de los sujetos para participar son agrupadas en:

1. Sociales (aprender y desarrollar nuevas habilidades; compartir conocimiento y habilidades, participar en un nueva forma de cooperación, participar en la escena del software libre).
2. De carrera o monetaria (mejorar productos de otros desarrolladores, mejorar mis oportunidades de empleo, ganar reputación en la comunidad).
3. Política (creer que el software no debe ser un bien propietario, limitar el poder de las grandes compañías).
4. Software (exclusivamente interés en el software) (Ghosh 2005:35)

Las motivaciones más señaladas por los participantes fueron las sociales, las cuales, fueron mencionadas por la mitad de los encuestados, y de éstas, la más recurrente fue la de *aprender nuevas habilidades y compartir el conocimiento*. La reputación y los beneficios económicos no son la motivación principal, pero son mencionados como un incentivo importante, sobre todo, en la construcción de una carrera como desarrollador.

Otro cuestionario realizado a 648 desarrolladores en el 2003, llega a resultados similares con la diferencia de dividir a las motivaciones en intrínsecas y extrínsecas (Lakhani y Wolf 2005:4-6). La contribución de este trabajo es mostrar la importancia que los participantes le dan a la participación como experiencia creativa, y que, aunque la mayoría afirmaba no recibir paga directa, la mitad contribuían en su tiempo laboral, muchos bajo consentimiento del patrón. Las motivaciones principales que mencionan son: distribuir software no mercantilizable, hacer dinero, limitar el poder de las grandes compañías de software, ganar prestigio y explotar la revisión entre pares.

Los resultados de Hars y Ou (2001) se formulan a partir una encuesta realizada a 389 desarrolladores de distintos proyectos, éste trabajo, aporta la construcción de tres grupos básicos de encuestados: 1. Estudiantes y aficionados; 2. Programadores asalariados y; 3. Programadores pagados para desarrollar OSS. Los resultados presentan algunas variaciones entre los grupos, conservando, una complementación entre motivaciones externas (recompensas futuras como vender productos, adquirir nuevas habilidades, y prestigio) e internas (decisión propia, altruismo y compromiso con la comunidad). Las motivaciones externas son más apreciadas por el tercer grupo, sin que ello implique un abandono de las motivaciones internas; en cambio, el grupo de los aficionados y estudiantes les brinda más importancia a las motivaciones internas, en especial, la de adquirir nuevas habilidades.

Los procesos de comunicación en el desarrollo del software que se pueden inferir, de los trabajos revisados en esta perspectiva son:

1. La naturaleza inmaterial del software y la posibilidad que brindan las computadoras para replicar la información, sobre una base de redes informáticas globales interconectadas permite;
2. la comunicación, interacción y trabajo, entre sujetos distribuidos geográficamente (principalmente entre Estados Unidos y Europa Occidental), los cuales,
3. se comunican de forma dinámica, rápida y a bajo costo, lo cual, les permite participar y colaborar en el hogar, universidad o el trabajo.

3.3 Perspectiva organizacional

Este apartado comprende investigaciones que abordan la organización en los proyectos de software libre (Ducheneaut 2005; Elliot y Scachi 2003; Fang y Neufeld 2009; Garzarelli et.al. 2008; German 2004; Jørgensen 2001; Markus 2007; Mockus et al. 2002 y; Sack et al. 2006; Stewar & Gosain 2006), desde distintas perspectivas, como: administración, gobernanza, socialización, y valores en la resolución de conflictos.

Un rasgo en común que tienen las investigaciones de la perspectiva organizacional, es que prácticamente todas, recurren a proyectos de software libre consolidados como *Apache*, *Free-BSD*, *GNOME*, *GNUe*, *Mozilla o*, *Python*. Estos proyectos han atraído la atención de distintos investigadores, lo que los ha convertido en casos de estudio recurrentes, cuyas formas de organización han sido abordadas desde distintas perspectivas.

La perspectiva organizacional es dividida en dos grupos, en el primero (German 2004; Jørgensen 2001 y; Mockus et al. 2002), se contemplan trabajos que abordaron las formas de organización del software libre, desde la administración de proyectos, con un enfoque descriptivo.

Las investigaciones del segundo grupo (Ducheneaut 2005; Fang y Neufeld 2009 y; Sack et al. 2006) poseen una integración mayor de elementos provenientes de las ciencias sociales y de las humanidades, en especial, el marco teórico de las comunidades de práctica de Ettiene Wenger, que se adapta y revisa para entender los proyectos de software libre y; de los *scientific technology studies* de Bruno Latour, de los cuales se recupera la teoría del actor-red.

Investigaciones descriptivas

Las investigaciones de Jørgensen (2001); Mockus et.al. (2002); y German (2004) son representativas de esta forma de analizar los proyectos, comparten entre sí preguntas de investigación y objetivos con poca variación, las tres exploran: ¿Cómo se organizan los contribuidores?, ¿cuáles son los procesos que siguen para solucionar sus dislocaciones?, ¿qué tipo de medios de comunicación utilizan?, ¿cómo interactúan? Esto condujo a descripciones muy detalladas de proyectos de software libre, la mirada que brindan de su organización es un modelo de "círculos concéntricos":

Altamente jerarquizado compuesto por distintas identidades: a) equipo central (*core team*), se encarga de coordinar el proyecto y de integrar las partes en un todo funcional; b) mantenedores (*maintainers*), encargados las sub-partes que componen el proyecto; c) parchadores (*patchers*), encargados de realizar correcciones; d) reporteros de errores (*bug reporters*), se encargan de realizar pruebas al software y sus partes para encontrar errores e identificar sus causas, pueden ellos mismos hacer sugerencias para solucionarlas o canalizarlas a los parchadores y; e) usuarios, que no se interesan por participar en el proyecto (Ducheneaut 2005:325).

El modelo de las jerarquías en estos trabajos se relaciona con el diseño de la arquitectura del software, lo cual da paso a la planeación modular, descrita como una de las actividades más complicadas y que toma más tiempo en los proyectos de software libre, Jørgensen hace referencia a la misma como "partir el proyecto en partes" (2001:228) interdependientes que al ser ensambladas toman su forma final. Esta planeación contempla que cada módulo es desarrollado o mantenido por un equipo de trabajo mínimo (una o dos personas) que reproduce el modelo de los círculos concéntricos. Esta interdependencia entre los distintos equipos de trabajo se complementa con el uso de múltiples sistemas de información que

posibilitan la revisión entre pares gracias, a que los sistemas de información hacen accesible el estado de cada uno de los módulos. Desde la perspectiva de Jørgensen (2001) esto conduce a que, pese la poca vinculación formal de los módulos, informalmente los sujetos se encuentran vinculados entre sí por sistemas de información que posibilitan la comunicación y revisiones constantes.

De este grupo de investigaciones se retoman las construcciones jerárquicas y de organización de los proyectos, de utilidad para contextualizar el tejido conceptual de las comunidades de práctica, aún y no lo recuperan. La diferencia con estos trabajos, es tomar como centro de análisis a los procesos de comunicación y, la inclusión de las herramientas de comunicación y trabajo como una variable de análisis.

Investigaciones teóricas

Este segundo grupo se caracteriza por su sólida integración de perspectivas teóricas provenientes de las ciencias sociales y de estudios sobre Internet. La organización en estas investigaciones, es representada no ya como un plano o un modelo, sino como un proceso cambiante, flexible y adaptable. A partir de sus referentes teóricos, se construyen dos grupos: a) aquellas que recurren al tejido conceptual de las Comunidades de Práctica de Etienne Wenger (2001) y; b) otras perspectivas relacionadas con crecimiento económico y gobernanza.

Trabajos que recuperan a las comunidades de práctica.

Los trabajos que recuperan los planteamientos de la participación periférica y de comunidades de práctica, lo hacen basándose en una visión que considera a la organización, como el resultado de un proceso de socialización/integración de largo plazo, que se construye con la participación de los sujetos en los proyectos.

Un segundo elemento que caracteriza a estas investigaciones, es el tratamiento de la organización en base a las teorías de redes, en especial, con la Teoría del Actor-Red de Bruno Latour. Estos trabajos (Sack et al. 2006; Ducheneaut 2005 y; Fang y Neufeld 2009), construyen un planteamiento que busca reconstruir el proceso de socialización/integración de los participantes en los proyectos; su método toma como eje la participación de ciertos sujetos a lo largo del tiempo, con los datos que quedan registrados en los sistemas de información, reconstruyen trayectorias, y a partir de ellas reconstruyen la evolución de las relaciones que establecen los participantes, entre sí y con el proyecto. En algunos casos el proceso se interrumpe y en otros continúa, esto lleva a una transformación identitaria de los sujetos, producto de la socialización, la construcción de sentidos de

afiliación con la comunidad y, procesos de aprendizaje.

La propuesta de Sack et al. (2006) divide los proyectos OSS en tres espacios: a) discusión, b) implementación y c) documentación. Antes de abordar cada espacio, cabe hacer la aclaración de que cada uno está relacionado con uno o varios sistemas de información interconectados, que en conjunto, componen una red de sistemas de información, denominada híbrida, porque en ella, se hibridizan, conversaciones y código; la acción de máquinas y personas y; los tres espacios descritos, de forma que, su división es una construcción del investigador.

El espacio de discusión comprende el lugar donde se establecen diálogos, luchas y negociaciones mantenidas en distintos sistemas de información como listas de correo electrónico, foros y, *chat* (IRC). Los trabajos abordados en este grupo (Sack et al. 2006; Ducheneaut 2005 y; Fang y Neufeld 2009), coinciden en señalar que las listas de correo son el medio principal por el cual se comunican los participantes de los proyectos, de forma que, metodológicamente hablando, las listas de correo son el medio principal a partir del cual se construye el espacio de discusión.

Por espacio de implementación, se entiende al espacio posibilitado por el sistema que administra el código fuente del proyecto (sistemas de control de versiones como CVS y GIT); en este espacio se puede observar, a través del análisis del código, la objetivación de los procesos de negociación y de toma de decisiones que tienen lugar en el espacio de discusión.

El espacio de documentación complementa las interpretaciones de los espacios de implementación y de discusión; la documentación podría considerarse como un nivel intermedio que registra, al mismo tiempo, los espacios de discusión y de implementación.

El trabajo realizado por Elliot y Scacchi (2003), presenta un armado teórico que contempla la perspectiva de las comunidades de práctica de Wenger (2001) en diálogo con el concepto de economía del regalo de Kollock y Smith (1998), lo cual, brinda elementos para enriquecer la comprensión del software libre como fenómeno cultural; Elliot y Scacchi, parten del supuesto de que los participantes del proyecto *GNU Enterprise (GNUe)*, no tienen la oportunidad de convivir cara a cara en su vida laboral cotidiana y, que mediante una cultura organizacional y valores relacionados con el software libre, logran superar sus conflictos.

Trabajos que recuperan elementos de política y gobernanza.

Entre los trabajos que recuperan elementos de política y gobernanza, resaltan los de Garzarelli et.al. (2008) y Markus (2007). Por un lado, Garzarelli et.al. (2008) muestra que el modelo de trabajo utilizado por el OSS presenta características de las economías redundantes (similar a la economía del

regalo de Kollock y Smith 1998), donde los ingresos y egresos, por la inmaterialidad del software, las redes informáticas y, las particularidades de las licencias públicas, impulsan la retroalimentación y construcción de conocimiento.

Por otro lado, Markus (2007), aborda a la gobernanza en los proyectos, como un híbridismo que combina formas orgánicas y sintéticas, de organización multidimensional, enraizada en: 1. Estructuras y procesos; 2. Reglas formales, informales y codificadas; internalizadas y externalizadas y; 3. En mecanismos de confianza, control y verificación. También, este trabajo muestra que la organización del equipo de trabajo en un OSS, sea exitosa: a) el tamaño del equipo, es mejor, cuando es mayor en etapas posteriores y, menor en etapas iniciales; b) un clima de trabajo agradable promueve una identidad grupal reforzada por sujetos prototipo (más que líderes, estereotipos positivos); c) durante las fases iniciales es más importante, contar con personal, que ofrecer resultados y en fases posteriores es al revés; d) en fases iniciales los esfuerzos de los desarrolladores, en identificar nuevas tareas, desborda su capacidad de realizarlas y conforma avanza el desarrollo se igualan; e) las variables relacionadas con el clima emocional de trabajo, son más importantes en las fases iniciales, que en fases posteriores donde procesos y rutinas ya están establecidos y; f) el promover una ideología en fases iniciales del proyecto, es contrario a la confianza de los participantes y al tamaño del equipo, en cambio, mientras el proyecto crece se va construyendo un sentido de pertenencia entre los participantes, sin necesidad de que sea promovido explícitamente.

Balance de la perspectiva organizacional

Los trabajos de la perspectiva teórica que recuperan el tejido conceptual de las comunidades de práctica, son la fuente principal de la la cual se nutre esta tesis. En el plano conceptual son de utilidad las definiciones de: los espacios de discusión, documentación e implementación de Sack et.al. (2006); la división del poder en un proyecto de software libre de Mahendran (2002)⁴⁰; el tratamiento que hace de las comunidades de práctica Ducheneaut (2005) y, Elliot y Scacchi (2003). En el plano metodológico, es espacialmente útil el trabajo de Sack et.al. (2006) y su concepto de red híbrida, que recupera elementos de la teoría del actor-red de Bruno Latour.

En lo que refiere a las referencias implícitas de la comunicación mediada por computadora en la perspectiva organizacional; la naturaleza informática e inmaterial del software, es un elemento estructural de los proyectos de software libre, que se extiende al ámbito de la investigación, como elemento metodológico central, ya que es, sobre la base de las redes de sistemas de información de los

40 Que se describe con detenimiento en el tejido conceptual.

proyectos, que los investigadores realizan su trabajo. Hacer uso de los sistemas de información como fuentes de obtención de datos, es una ruta utilizada de forma regular por los trabajos de la perspectiva organizacional, los cuales, retoman de forma explícita o implícita la diferenciación de los espacios de discusión, implementación y documentación de Sack et.al. (2006). Un elemento final a señalar, es que las listas de correo, son utilizadas en los trabajos revisados en esta perspectiva como la fuente de información por excelencia para la extracción de datos.

3.4 Perspectiva Económica

Los trabajos seleccionados en la perspectiva económica abordan proyectos, que combinan modelos comerciales de empresa, con modelos comunitarios de desarrollo y, viceversa. Los planteamientos de esta perspectiva se formulan a partir de preguntas como: ¿Cuáles son y qué efecto tienen en el OS y en el mercado del software, los cambios en los modelos de producción de empresas que comercializan y usan OSS? (Fitzgerald 2006); ¿Cuáles son las principales características de los modelos de negocio de las empresas OSS de segunda generación (OSSg2)? (Watson 2008); ¿Se puede considerar al modelo de innovación del OSS una mejor combinación del modelo privado y de acción colectiva?, ¿Qué beneficios trae para los desarrolladores este modelo? (von Hippel y von Krogh 2002); ¿Aumentará la calidad del software de una empresa productora de software cerrado de manera distinta, cuando enfrenta la competencia de una empresa que produce OSS, que cuando no lo tiene?, ¿Habrá un cambio en la respuesta de la firma de software cerrado cuando hay otro competidor de software cerrado, aparte del competidor OSS? (Jaisingh et al. 2009).

Fitzgerald (2006) denomina como OSSg2 a empresas productoras de OSS que adoptaron un modelo híbrido de producción que combina los esfuerzos de empleados, con el de voluntarios; en palabras de Fitzgerald (2006), híbridos privado-colectivos, orientados al desarrollo de OSS, de manera comercial. Watson (2008) por su parte, retoma el nombre OSSg2 y su problemática, para profundizar en el fenómeno de la producción de OSS, desde un modelo que combina la empresa con las comunidades. Por ejemplo, *Trolltech*, empresa que desarrolla las librerías para interfaces gráficas Qt, se considera a sí misma como propietaria de las librerías que desarrollan, pese a que otras empresas poseen también parte de *Trolltech*. Su balance lo obtuvieron mediante la venta de un tercio de la firma y, el reparto de los dos tercios restantes entre sus 230 empleados (Watson 2008:44). El elemento que más distingue al OSSg2, según la perspectiva de este autor, es la capacidad de innovación y dinamismo de estas empresas.

Sobre el balance "empresa y comunidad", von Hippel y von Krogh (2002), definen a los

proyectos OSS como redes de innovación, cuyo dinamismo se sustenta en la práctica generalizada de compartir las innovaciones con otros miembros del equipo y, de distribuirlas de manera directa con el usuario, o indirecta, con intermediarios independientes, de manera rápida y económica, gracias a la naturaleza inmaterial del software y a la existencia de la Internet. Este postulado contempla que el modelo de innovación OSS es un balance entre intereses, personales y empresariales, donde el bien o la innovación, se maneja como un bien público.

Un último aporte brindado por Jaisingh et.al. (2009), es que el modelo de negocios de las empresas OSSg2, se basa en lo que denominan “efecto red”, el cual; contempla que los ingresos de estas empresas no vienen de vender el software como un producto, sino, de comercializarlo como un servicio, donde, el valor del mismo aumenta en relación con el número de sus usuarios donde; a mayor número de usuarios, más confianza y más valor, de forma que, los desarrolladores ganan cuando otros utilizan sus programas y, los intermediarios (entre desarrolladores y usuarios) ganan, pues mientras más programas conocen y usan, más amplia es la gama de servicios que pueden ofrecer. Aunado a esto, mientras más extendido es el uso de un programa, más probable que los proyectos OSSg2 reciban apoyo de la comunidad.

Las conclusiones del análisis de la perspectiva comercial son:

1. Los proyectos de software libre comerciales, promueven la formación de una comunidad alrededor del software.
2. En los casos en que son empresas con fines comerciales las que promueven los proyectos de software libre, sus actividades contemplan el cultivo de comunidades de usuarios, que a su vez.
3. Conforman redes de uso, distribución y comercialización del software, el cual;
4. Es representado como un servicio, de forma que las ganancias, se generan a partir de la venta de servicios como instalación, configuración, mantenimiento y capacitación.

Cabe señalar que la capacidad para reproducir el software y su ubicuidad, promueven una forma de producción y comercio que está cambiando los paradigmas de la industria del software; la comunicación mediada por computadora juega un papel fundamental en este proceso, pues es a través de la interacción mediada por computadora entre sujetos y, sistemas de información, que el modelo de comercio del software libre es posible.

3.5 Ley, Cultura y Sociedad

Este grupo de investigaciones (Kelty 2000 y 2008; Lessig 2000 y McGowan 2000) abordan al software libre desde las perspectivas cultural⁴¹ y jurídica⁴² del software; los trabajos seleccionados, representan algunas de las posiciones más influyentes en estos ámbitos, en especial los autores Lessig y Kelty aportan elementos jurídicos y culturales que, interrelacionados, resultan de utilidad para comprender los fundamentos sociales de las prácticas de trabajo, de distribución y de comercio del software libre. Ambos autores, extienden la problematización al mercado de bienes digitales en la Internet, derechos de propiedad, libertad de discurso y, de acceso a la información y al conocimiento.

Limitado al ámbito del F/OSS, McGowan (2000) presenta un trabajo que aborda las implicaciones legales que conlleva, realizar trabajos derivados de software registrado bajo licencias públicas, su análisis se sitúa en juicios legales donde se disputan violaciones a la GNU-GPL. Sus conclusiones apuntan a que, las licencias públicas no están relacionadas con el abandono de los derechos de propiedad, sino con su reformulación. Entre las obligaciones que conlleva esta reformulación, resalta el llamado, "carácter viral" de la GNU-GPL, el cual, obliga a que todo trabajo derivado de un software registrado bajo esta licencia, sea registrado bajo una licencia compatible y, a que el código fuente sea accesible a sus usuarios.

Por su parte, Lessig (2000) construye una relación entre las licencias públicas y lo público, donde público es definido, como un conjunto de bienes de acceso abierto en Internet, registrados bajo licencias públicas, que protegen su accesibilidad y uso para realizar nuevos productos. La discusión de este autor parte del estatuto de propiedad de bienes como el OSS, los cuales, sirven como una reserva de bienes compartidos, a partir de la cual se facilita la creación de nuevos bienes, que a su vez, alimentan esta reserva de bienes compartidos.

Las consideraciones de Lessig (2000) sobre la propiedad y los bienes, al situarlos en el contexto de la comunicación mediada por computadora, les confiere una nueva naturaleza, más fluida y difícil de controlar, de forma que, surge un debate para definir los límites de lo público y lo privado de los

41 La perspectiva cultural, desarrollada por Kelty (2002 y 2008), es formulada a partir de los registros antropológicos y simbólicos de la cultura, por un lado, el registro antropológico se centra en el estudio de los grupos que desarrollan software libre, como "culturas" o grupos. Esta visión se complementa con el registro simbólico que se centra en la construcción de sentido que *geeks*, formulan sobre el mismo.

42 Por perspectiva jurídica se busca resaltar a aquellas investigaciones que abordan el tema de las licencias y sus implicaciones sociales y culturales, es decir, las implicaciones que tienen las licencias en el uso y apropiación que los sujetos hacen con el software.

productos. La posición de Lessig en el debate, resalta el beneficio que traen consigo el uso de licencias públicas como la GNU-GPL, para el enriquecimiento de lo público frente a la tendencia de control y vigilancia de las industrias culturales, en particular, frente a propuestas legales como la DMCA (*Digital Millenium Copyright Act*) de 1998.

Es de utilidad señalar los aportes del antropólogo Christopher M. Kelty (2002 y 2008), para entender al software libre desde una perspectiva cultural. Su trabajo del 2002 es formulado con la recuperación del trabajo de Marcel Mauss y de Bronislaw Malinowski. El interés de Kelty (2002) se centra en realizar un análisis de *The Cathedral and the Bazaar* de Erick Raymond (2000) desde la mirada del visitante, del extranjero, del recién llegado; que se sumerge en un contexto cultural que no conoce y desde el cual, discute los presupuestos del modelo de trabajo del software libre, con los conceptos de economía del regalo y de memoria de Mauss.

En el trabajo del 2008, Kelty aborda con más detenimiento las implicaciones culturales del software libre. Para comenzar, define al sujeto colectivo relacionado con el software libre como "geeks", un término que, si bien hace referencia la relación entre el sujeto y las tecnologías de información y comunicación, su elemento distintivo es una concepción sobre, una serie de valores compartidos relacionados con la Internet, entre los cuales resalta la libertad de decisión, de discurso, de acceso, de copia de la información y, de forma especial, la concepción de la tecnología como "elemento retórico en el cual, es irrelevante argumentar sobre la tecnología, lo que vale, es argumentar a través de la tecnología" (Kelty 2008:92). Esta posición, contempla que la tecnología es un medio para llegar a determinados fines, que dependen en última instancia, de una o varias voluntades.

Con respecto a la comunicación mediada por computadora, Kelty (2008) la problematiza a partir de la relación entre "sistemas operativos y sistemas sociales", la cual, sirve para abordar una interdependencia entre las características y posibilidades de los sistemas de información y comunicación y, los sujetos que los construyen y utilizan. El caso en el cual se sitúa para esta formulación, son las listas de correo electrónico, utilizadas por los *geeks*, las cuales; sirven como medio que posibilita un tipo de interacción, similar a salones públicos, donde los sujetos intercambian opiniones, discuten y construyen acuerdos en temas de interés general, como por ejemplo, la Internet, la libertad de expresión, o el software libre.

Para cerrar el apartado de la perspectiva de ley, cultura y sociedad, cabe señalar que los trabajos seleccionados aportan elementos para argumentar por la interdependencia de los sistemas de información, con respecto a los sistemas sociales que los usan; los señalamientos de Kelty (2008) y de

Lessig (2000) sirven para argumentar que la cultura y las leyes fueron y son, elementos fundamentales en la configuración de los sistemas.

Al respecto, la comunicación medida por computadora podría parecer que depende, de los sistemas de información que la posibilitan, sin embargo, esta conclusión se debilita cuando se recupera el ejemplo de las RFC (*Requests For Comments*), las cuales, son los primeros ejemplos de la comunicación mediada por computadora y, fueron utilizadas como un instrumento para construir acuerdos, sobre los estándares de Arpanet (Williams 2002). A lo anterior y bajo el supuesto de la independencia cultural desarrollada por el programa fuerte de la sociología cultural de Alexander (2000); los sistemas de información pueden ser definidos, como objetivación de las negociaciones que tuvieron lugar a través de la comunicación mediada por computadora, que a su vez, es definida por (o depende de) una o varias culturas; en este sentido, existe un alto grado de dependencia de los sistemas de información, en relación con las formas culturales de los participantes y, de los procesos de comunicación, a través de los cuales se formulan los acuerdos para configurarlos.

3.6 Conclusiones a la representación del software libre desde las ciencias sociales

Desde estas cuatro perspectivas relacionadas a las ciencias sociales, el software libre es representado como:

1. **Motivacional:** Un sistema de desarrollo de software en donde participan y colaboran, de manera voluntaria, cientos de usuarios y desarrolladores distribuidos alrededor del mundo; en algunos casos como parte de su trabajo, pero en muchos otros, como un tipo particular de práctica a partir de la cual, los colaboradores aprenden nuevas habilidades y se dan a conocer en el mercado del software libre.
2. **Organizacional:** Los proyectos de software libre, lejos de representar un modelo de organización sin jerarquías e igualitario; representan, redes complejas que a partir de los niveles de acceso en los sistemas de información que componen su red de herramientas de comunicación y de trabajo, establecen estructuras jerárquicas bien definidas. Una particularidad de estas estructuras, es que están formuladas para promover la colaboración de sus usuarios, para lo cual, hacen accesibles los medios y herramientas (informacionalismos) para que el usuario, de forma ideal: aprenda, participe, socialice y se integre al equipo de trabajo.
3. **Económica:** Los proyectos de software libre, en su mayoría no son organizaciones de caridad

que de forma desinteresada liberan sus productos en Internet; sino, organizaciones con fines comerciales que buscan recibir dinero a cambio de su trabajo. La forma de comercialización más frecuentemente utilizada (no la única), contempla al software como un servicio, no como un producto, de forma que; los proyectos de software libre venden la instalación y soporte a sus clientes. Por ejemplo, la empresa X quiere construir una red para sus empleados, así que contrata Red-Hat, los cuales, no le cobran su distribución GNU & Linux, sino la instalación, configuración y mantenimiento de la red, por ejemplo, por un año; de forma que, si la empresa X requiere soporte o asistencia de Red-Hat, tiene que pagar por ella. Por otra parte, si la empresa B requiere del software para construir una red y, uno de sus empleados es experto en redes GNU & Linux, descarga de forma gratuita la distribución GNU & Linux de su preferencia y construye la red, sin necesidad de pagar por el software que están utilizando. Este modelo de negocio, beneficia de las dos formas a las distribuciones GNU & Linux, ya que, aunque en el segundo caso no reciben retribución económica directa, el que exista un usuario más les beneficia, ya que, el valor de sus servicios aumenta según el número de sus usuarios, así, mientras más usuarios, más valor; por otro lado, es muy probable que el empleado de la empresa B que instaló GNU & Linux, sea un miembro activo de un proyecto de software libre, al cual contribuye, ayuda y participa de forma desinteresada.

4. Ley, cultura y sociedad: Esta agrupación, pareciera englobar a todas aquellas investigaciones, cuyas perspectivas no encontraron lugar en las perspectivas anteriores⁴³, sin embargo, al comparar los trabajos de Kelty y de Lessig, resultan complementarios. Por un lado Kelty maneja una perspectiva cultural que aborda el sentido de la adopción de las prácticas del software libre, lo cual comprende, el uso y explotación de herramientas y medios de comunicación relacionados con Internet; en este sentido, la problemática de Kelty, aunque trata el tema de las licencias lo hace de forma periférica. Por otra parte, la perspectiva de Lessig retoma una perspectiva cultural, pero la aborda desde las posibilidades que brindan los modelos de licencia intelectual imperantes, en particular, la diferencia entre licencias restrictivas y licencias públicas (cultura de solo lectura y; cultura de lectura y escritura). De ambas perspectivas surge una complementariedad que sugiere líneas sólidas para comprender las implicaciones legales de una cultura libre y abierta; que promueve el intercambio dinámico y viral (en el sentido del carácter viral de la GNU-GPL) de bienes digitales.

43 Al menos así parece la compilación de Lerner et.al. (2005), de la cual se obtiene este modelo de clasificación.

Múltiples perspectivas y trabajos fueron dejados de lado en favor de la selección realizada, la cual, busca dar cuenta de los trabajos más influyentes, realizados sobre software libre, a este respecto, no quedan deudas.

Donde sí quedan deudas en esta revisión, es a la comunicación mediada por computadora, la cual, dada su amplitud, merece su propia revisión; en este sentido, dicha revisión fue sacrificada en favor de la comprensión de la producción académica social del software libre. El principal riesgo de haber tomado esta decisión, es que, reproduce el modelo de dar por supuesta, a la comunicación mediada por computadora.

Dejar de lado la revisión documental de la producción académica de la comunicación mediada por computadora, en una investigación que se dice tener por objetivo, los procesos de comunicación que tienen lugar en un proyecto de software libre, representa una contradicción. Para mitigar los efectos negativos de esta contradicción, se integra la perspectiva de Gunawardena (1994) como un eje metodológico principal, para el abordaje del diálogo mediado por computadora y el trabajo colaborativo.

4 Lentes conceptuales

El tejido conceptual que se presenta, busca ser imaginado como cristales (Berdou 2007) a partir de los cuales observar los procesos de comunicación en el proyecto Moblin v2, esta forma de entender a la teoría y los conceptos, considera que el punto de encuentro entre realidad y teoría es la observación. Se considera que esta perspectiva permite imaginar que, al recurrir a una teoría o concepto, como lente a través del cual se observa; una realidad emerge, se destruye o transforma en el observador, mediante el privilegio de uno o varios aspectos de la realidad observada, esto implica que, ciertos aspectos son enfocados, otros semi-observados y muchos más, dejados de lado.

Los lentes que se proponen utilizar para observar –en general- al fenómeno del software libre y –en particular- a Moblin v2 y sus procesos de comunicación, comienzan con el marco conceptual de las comunidades de práctica de Lave y Wenger (1991), Wenger (1998) y Wenger y McDermott (2002); de los cuales, serán rescatados los conceptos de perifericalidad y participación legítima, como una forma de referirse a los sujetos que participan en Moblin v2. Para situar este marco conceptual en el contexto del software libre; se recurre a la interpretación que se realiza del mismo, en los trabajos de Berdou (2007) y, Elliot y Scacchi (2003).

Para definir comunicación mediada por computadora, se recupera la perspectiva de la

interacción humano-humano de Jensen (2001), en el marco del proceso productivo del proyecto Moblin v2, entendido como constelación de comunidades de práctica (Wenger 1998). Ésta problematización de la comunicación, se construye a partir de tres elementos que se consideran principales: a) la red de herramientas de comunicación y de trabajo (sistemas de información) que utiliza el proyecto Moblin v2; b) el modelo de trabajo Bazar y; c) la interpretación de necesidades subyacentes de los participantes (centrales y periféricos), del proyecto.

Para la reconstrucción conceptual de la red de herramientas de comunicación y de trabajo, se recupera el concepto de red híbrida de Sack et.al. (2006) y, de informacionalismos de Elliot y Scacchi (2003). Por un lado, con el concepto de red híbrida busca clarificar a los sistemas de información como herramientas que permiten la comunicación y el trabajo colaborativo y; por el otro, los informacionalismos, como producto y recursos, de y para, la comunicación y el desarrollo de Moblin v2.

Para situar el modelo de trabajo Bazar, se recupera el trabajo de Raymond (2000), revisado y pulido por los trabajos de Elliot y Scacchi (2003) y Berdou (2007), los cuales, ayudan a situar las lógicas que guían a este modelo de trabajo a la luz de las comunidades de práctica y de la comunicación mediada por computadora.

El concepto de interpretación de necesidades subyacentes de Habermas (2001), es utilizado para situar, los objetivos de los participantes sin reducirlos a una perspectiva racionalista de las intenciones, sino que; contemple valores y normas que condicionan, tanto la construcción de los objetivos, como los medios para lograrlos. En este sentido, el concepto de necesidades subyacentes aporta elementos para explorar, tanto los móviles de la acción, como los medios para realizarla mediante la comunicación mediada por computadora en el marco de Moblin v2.

4.1 Comunidades de práctica (CdP)

Las comunidades de práctica son grupos de personas que comparten una experticia sobre un oficio u actividad (llamado dominio), mediante el ejercicio de dicha actividad (práctica) y, comparten también, uno o varios objetivos delimitados, los cuales son denominados "empresa compartida"⁴⁴. A la acción de los sujetos, en el ejercicio de la práctica, en una comunidad determinadas, se le llama participación, a partir de la misma, los sujetos ejercitan la práctica, aprenden el dominio y la comunidad está un poco más cerca de la obtención de su empresa (Wenger y McDermott 2002).

⁴⁴ El término "empresa compartida" es usado por Wenger (1998) para definir los objetivos que la comunidad de práctica persigue.

El modelo de organización, que contempla las comunidades de práctica, en el caso de proyectos de software libre, según la interpretación de Berdou (2007), es imaginado en círculos concéntricos; en el centro, se encuentran los maestros, alrededor, los participantes legítimos (también llamados jóvenes maestros) y a su alrededor los participantes periféricos (llamados aprendices, nuevos o, recién llegados).

El grupo de los maestros es el más activo y el poseedor del dominio, son el grupo con mayor legitimidad, reconocimiento y quienes tienen mayor agencia en la definición de la empresa y los lineamientos de la comunidad. El grupo de los jóvenes maestros se constituye por aprendices que, por medio su participación sostenida y su aprendizaje del dominio de la comunidad, han logrado el reconocimiento de los maestros y han recibido mayor acceso a los recursos de la comunidad y, el derecho a desempeñar tareas que involucran responsabilidad y brindan mayor prestigio a quienes las realizan. Los aprendices o recién llegados, son el grupo con menor reconocimiento y menor acceso a los recursos de la comunidad, su posición es definida como de perifericalidad con respecto a la comunidad. Su tarea es aprender de sus maestros, pero su aprendizaje involucra algún tipo de aportación a la comunidad, "generalmente en la ejecución de tareas rutinarias" (Berdou 2007:29), conforme aprenden el dominio y adoptan los modelos de trabajo, presentan, un desplazamiento de la periferia hacia el centro.

En este caso, la perifericalidad se manifiesta como la serie de restricciones que la comunidad implementa a las prácticas más maduras de la comunidad (también las que gozan de mayor legitimidad). La perifericalidad es concebida como:

una experiencia de no participación que resulta en un sentido de exclusión. En el caso del FOSS la participación es cuidadosamente estructurada a través de la regulación formal e informal en los diferentes grados de acceso a los proyectos y recursos que, resulta en un renovado sentido de jerarquía (Berdou 2007:29)

La perifericalidad, es el sentido que sustenta la división del trabajo y configura las jerarquías en la comunidad; es posible observarla en los niveles de acceso a los sistemas de información que componen los proyectos. Por ejemplo, el tener o no permisos para participar de manera directa en la edición del código, habla de la posición y prestigio de un sujeto, ya que, es una de las actividades más delicadas y está reservada solo a ciertos participantes; de manera similar, tener la capacidad para editar el sitio web o los manuales y guías, aunque no goza del mismo prestigio, goza de reconocimiento, y además, es un

incentivo en el camino desde la periferia, hacia las posiciones centrales (Berdou 2007).

Una aclaración que debe realizarse sobre las comunidades de práctica, es que es posible que distintos tipos de práctica se realicen en una misma comunidad, en el caso de los proyectos de software libre, su actividad no se limita a la programación, hay artistas, mercadólogos, administradores y diseñadores que participan y que ejercitan distintos tipos de práctica al interior del mismo proyecto. Para estos casos, Wenger (2001) recurre al concepto de "constelaciones de comunidades de práctica", que refiere a conglomerados de múltiples comunidades de práctica que comparten la misma empresa.

Para esta investigación, es de utilidad recuperar del planteamiento de las comunidades de práctica, la definición que hacen autores como Berdou (2007) y Elliot y Scacchi (2003) de un proyecto de software libre, como un modelo jerárquico que combina códigos y reglas, formales e informales, cuyo sustento emana del reconocimiento mutuo, objetivado y ejercido en niveles de acceso a los recursos de la comunidad, en particular, los sistemas de información (herramientas de comunicación y de trabajo).

Por el lado metodológico, el planteamiento de las comunidades de práctica se centra en la participación (como un tipo particular de acción) de los sujetos, por ejemplo; Fang y Neufeld (2009) realizan su investigación de un proyecto de software libre, con los historiales que los sistemas de información registran sobre las acciones de los sujetos, a partir de ellos y de su contextualización como participación, determinan grados de adscripción a la comunidad, roles, y jerarquías.

Para terminar, cabe señalar la crítica que presenta Ducheneaut (2005) y Sack et.al. (2006), a las interpretaciones simplistas sobre la organización de los proyectos de software libre; la cual, consiste en poner en entredicho la interpretación lineal del modelo de círculos concéntricos construido con las comunidades de práctica. Según la perspectiva de Ducheneaut (2005), sobre-simplificar la organización de un grupo, a un modelo lineal de círculos concéntricos, es un error común cuando se aborda el software libre, para solucionarlo, propone no limitarse al reconocimiento que la comunidad brinda a sus participantes, al compromiso de los sujetos con la comunidad o a la eficiencia de las aportaciones, sino también, a factores como el carisma de los participantes.

La crítica al modelo de círculos concéntricos de Ducheneaut (2005), se basa en un estudio realizado al proyecto Python (que produce el lenguaje de programación Python); según explica este autor, los sujetos, a partir de un proceso de socialización fundado en la participación mantenida a lo largo del tiempo, son reconocidos por la comunidad y se les brinda prestigio y accesos; sin embargo, no es la única forma de desplazarse hacia el centro, los participantes, tienen la capacidad de formar

alianzas con otros miembros del proyecto o, de movilizar a segmentos de la comunidad para alcanzar posiciones más prestigiosas.

Por su parte, Sack et.al. (2006) argumenta que el carisma y la política, entendidos como una relación que configura la agencia de los sujetos; son elementos que no pueden ser dejados de lado al considerar a los elementos que intervienen en la configuración de las estructuras jerárquicas de un proyecto. Aunado a lo anterior, cabe recuperar del análisis de Berdou (2007), que en proyectos donde intervienen empresas y comunidades, surge una diferencia entre empleados y voluntarios que interviene en la configuración de las agencias de los participantes del proyecto.

De los planteamientos de Berdou (2007), Ducheneaut (2005) y Sack et.al. (2006), es de utilidad recuperar una visión sobre la división jerárquica en una comunidad de práctica que contemple, al mismo tiempo, elementos como el prestigio, el compromiso y el mérito con otros como el carisma, la política o el mercado.

4.2 Comunidades de práctica en el contexto de la comunicación mediada por computadora.

Para construir una definición que permita comprender los procesos de comunicación que tienen lugar en proyectos de desarrollo de software libre, se parte de la noción de *interacción* utilizada por Jensen (2001) para describir la comunicación mediada por computadora. Esta definición busca ser relacionada con la comparación de Bacon (2009), de un proyecto de software libre con un barrio o vecindario, donde, los sujetos se conocen, intercambian información, y bienes; construyen relaciones de amistad, solidaridad, arraigo, compromiso, amor, resentimientos y; aunque sean pocos los vecinos, no siempre todos conocen a todos, ni intercambian de manera directa bienes; aunque, en determinado momento, pueden prestarse ayuda por el solo hecho de ser vecinos.

La perspectiva de Jensen (2001) señala tres metáforas para situar los niveles de la interacción de los sujetos, a través de la comunicación mediada por computadora:

1. Comunicación humano-máquina: proceso de interactividad y de construcción de sentido interno; esta metáfora de comunicación, es situada en un nivel individual, cognoscitivo o psicológico, del estudio de las relaciones entre, mensajes transmitidos por computadora y el o los usuarios de los mismos.
2. Comunicación humano-humano: contempla el intercambio de mensajes, la construcción de sentido, y también, contempla el medio por el cual se transmiten los mensajes; esta

metáfora se sitúa en el plano grupal de la comunicación mediada por computadora en donde, las máquinas son instrumentos que sirven para mediar conversaciones entre individuos.

3. Comunicación humano-máquina-sociedad: contempla un análisis macro-social, construido desde distintas disciplinas de las ciencias humanas, naturales y sociales, el objetivo de este análisis es llegar a la comprensión de las transformaciones que provoca la sociedad sobre si misma a partir del uso de las tecnologías de la información y comunicación.

En la construcción de estas metáforas Jensen (2001) propone recuperar, la relación de entre agente-estructuras, para el análisis y comprensión del proceso de comunicación. Este modelo, es retomado de la teoría de la estructuración de Giddens, donde, las estructuras sociales más que ser situadas en las instituciones que conforman al Estado o controlan la economía, centran su actuar en las representaciones que hacen las personas sobre dichas instituciones y la sociedad, en un sentido similar a reglas o normas; uno de los beneficios que ofrece esta perspectiva, es que resalta la capacidad del agente para influir, conservar o cambiar, las instituciones; de allí que los sujetos sean llamadas agentes.

Jensen (2001) agrega a la relación estructuras-agentes, al medio o los medios de comunicación a través de los cuales, los agentes se comunican entre sí, para la construcción, reformulación o mantenimiento de las estructuras; en este sentido, la visión de la estructuración de Giddens toma una nueva dimensión de utilidad para la comprensión de la comunicación mediada por computadora que contempla tres tipos de interactividad que los agentes establecen con los mensajes y entre sí:

- Interactividad 1 (agente-medio): esta primera interactividad sirve para definir al agente como usuario, en relación y rompimiento, con la definición de receptor de los estudios sobre medios de comunicación masiva (*mass media research*); esta visión es construida a partir de la actividad, intercambios y construcción de sentido, que el agente establece con y a partir de mensajes transmitidos y mediados por la computadora, que en palabras de Jensen:

Involucra las secuencias estructuradas de selecciones, que deben los humanos de realizar para comunicarse con los mensajes embebidos en programas computacionales, donde además, se ha de contemplar no solo a los creadores de los contenidos, sino también, a los desarrolladores de las herramientas, que de una forma menos visible (pero no por eso menos activa), participan en

los procesos de comunicación (2001:78).

- Interactividad 2 (estructura-medio): hace referencia a la metáfora de la sociedad como sistema de comunicación a través de cuatro modos de comunicación mediada por computadora:
 - Alocución, refiere a un tipo de comunicación similar a los medios de comunicación de masas (unidireccional y de uno a muchos);
 - Conversación, lo más cercano a la comunicación interpersonal;
 - Consulta, el usuario interactúa con un conjunto de programas y sistemas de información a través de secuencias de órdenes, por ejemplo, con un juego en línea que exige una constante toma de decisiones por parte del usuario y;
 - Registro, que refiere a las trayectorias que los usuarios construyen al navegar en Internet.
- Interactividad 3 (agente-estructura): en este caso la comunicación fluye de los usuarios hacia las instituciones a través de la comunicación que establecen con otras personas, sean pares, mediante la comunicación directa con las instituciones, en este tipo de interactividad, es donde el tema de las transformaciones sociales acontecidas con la entrada de las computadoras encuentra lugar.

Para los fines de esta investigación, se sitúa el análisis de los procesos de comunicación en el nivel humano-humano o bajo la terminología de Giddens, agente-agente, donde: el interés reside en caracterizar cómo los participantes se comunican a través de los sistemas de información, cómo se relacionan estos usos con modelos de organización y cómo esta relación entre sistemas de información y modelos de trabajo, facilitan o dificultan la satisfacción de las necesidades de los participantes, en el marco de la empresa, de Moblin v2, como constelación de comunidades de práctica.

La utilidad de recuperar la perspectiva analítica de la "Interactividad 2", y en especial, los modos de comunicación mediada por computadora "conversacional" y de "registro"; radica (por el lado del modo conversacional) en su compatibilidad con la metáfora de Kelty (2008) y de Bacon (2009); de que las interacciones entre los participantes en el proceso de desarrollo de software libre, son similares a las que tienen lugar en reuniones públicas (*meeting halls*), donde se discuten temas relacionados con una empresa en relativa igualdad de circunstancias, lo cual contempla que ciertos sujetos tienen más prestigio que otros, pero que, su voz no evita que voces divergentes participen en las discusiones y

conversaciones.

La utilidad del modelo de registro de los sujetos en los sistemas de información, desde la perspectiva de Berdou (2007); radica en que dichos sistemas realizan un registro de todo lo que los sujetos hacen en ellos. Esta formulación, es relacionada con el panóptico de Foucault, es decir, un sistema de vigilancia desde el cual, es posible observar todo cuanto sucede en determinado espacio y/o institución; desde esta perspectiva, un proyecto de desarrollo de software libre es similar a un panóptico, porque, permite la observación y registro de las acciones de cada uno de los participantes. La diferencia principal entre el panóptico y el proyecto de software libre, es que la capacidad para estudiar los historiales de los participantes se mantiene abierta a los mismos, de forma que, esta apertura, facilita la revisión entre pares, la resolución de conflictos y, el prestigio de los participantes.

4.3 Redes de comunicación y trabajo en el desarrollo de software libre.

El concepto de red híbrida, tal como lo manejan Sack et.al. (2006), en la representación de un proyecto de software libre, se formula sobre la base de la teoría de la acción-red (ANT por sus siglas en Inglés) de Bruno Latour. Sin pretender profundizar en los planteamientos de la ANT, una red, es definida como el total de las relaciones que pueden establecer entre sí distintos actores, donde; por actores, no se considera solamente a los sujetos, sino también, a aquellos elementos, objetos u objetivaciones, que de alguna forma o de otra, participan en las relaciones que los sujetos establecen entre si (Wenger 1998).

Un proyecto de software libre entendido como una red híbrida está conformada por participantes, sistemas de información utilizados, interfaces, Internet, información producida y enlazada y; procesos de socialización entre los participantes, es decir, los lazos que construyen, los conflictos que crean, los acuerdos a los que llegan, sus luchas, diálogos, consensos, entre otros (Sack et.al. 2006:234).

Sack et.al. (2006) utiliza el calificativo de híbrida, a una red de actores que intervienen en el desarrollo de software libre, para resaltar dos cualidades de la red; la primera, es que existe un hibridismo inherente a los sistemas de información sobre los cuales se desarrolla el proyecto; por un lado, son herramientas, por medio de las cuales se trabaja, y por el otro; los sistemas de información son también los medios de comunicación por medio de los cuales los sujetos conversan, construyen acuerdos, crean y resuelven disputas, sobre el desarrollo del software. Esta perspectiva sirve para dar cuenta de un proceso recursivo en el cual, las herramientas son desarrolladas por medio de sí mismas.

La segunda cualidad, es que el código y las conversaciones se encuentran hibridizados; por un lado, el código, da cuenta de los procesos de discusión y negociación de los participantes durante el

proceso de desarrollo del código. Y por el otro, las conversaciones entre los sujetos, recuperan y dan cuenta del código, al cual se hace referencia (Mahendran 2002).

Una red híbrida, según la construcción de Sack et.al. (2006), puede ser dividida en tres espacios: de discusión, documentación e implementación. En el primero, se discute sobre el proyecto y el software; en el segundo, se documenta el proceso de desarrollo del software y se crean manuales y guías para poder entenderlos y utilizarlo; el tercero, es el espacio en el que se desarrolla el código. Esta división de espacios, conlleva una división de funciones para los sistemas de información, donde, por ejemplo; las listas de correo electrónico sirven para discutir; las wikis, sirven para documentar y; Git sirve para implementar.

Para esta tesis, no será utilizado el término red híbrida de Sack et.al. (2006), sino, que será utilizado el término *red de comunicación y de trabajo*, del cual se desglosan los términos: a) red de sistemas de información (herramientas y canales de comunicación) y; b) informacionalismos (recursos y productos, imaginados como objetivación de la práctica de los participantes).

Por red de sistemas de información se entiende a, un conjunto de sistemas de información en línea (*on line*) articulados técnica y organizacionalmente; se caracteriza porque cada una de los sistemas que la conforman, cumple una o varias funciones diferentes, que en conjunto, buscan brindar el soporte tecnológico a partir del cual, satisfacer el total de necesidades de los participantes y la empresa de la comunidad. Las funciones principales que posibilitan los sistemas de información son, comunicar, documentar e implementar.

El concepto de informacionalismo de Elliot y Scacchi (2003), se utiliza para agrupar a todas aquellas objetivaciones de la interacción, comunicación y trabajo, de los participantes de proyectos de software libre, acumulados a lo largo del tiempo, disponibles a muy bajo costo, para todos aquellos interesados que requieren de recursos, herramientas o canales de comunicación, para satisfacer una o varias necesidades.

Los informacionalismos son repertorios de recursos, objetivados en información digital, accesibles por medio de la Internet, cuya utilidad reside en ser soporte para usuarios y desarrolladores de software libre. Los recursos que componen a los informacionalismos, incluyen código, herramientas, manuales, guías, documentación, discusiones, enlaces, canales de ayuda, entre los más relevantes (Elliot y Scacchi 2003).

Bajo la consideración de que un proyecto de software libre hace accesible a través de Internet: a) el software que produce y su código fuente; b) el registro de la comunicación y trabajo de los

participantes; c) documentación sobre el código producido; d) manuales de uso para el software producido; e) enlaces a otros tipos de información utilizada y; f) canales de ayuda. Puede proponerse que informacionalismos, es una noción relacionada con la objetivación de la interacción, comunicación y trabajo de un proyecto software libre, soportados por una (o varias) red(es) de sistemas de información, con la aclaración de que; el concepto de informacionalismos, involucra objetivos, utilidad y accesibilidad.

Para concluir, la red de comunicación y de trabajo contempla que, la red de sistemas de información que posibilita la interacción y comunicación entre los participantes, sea entendida como espacios de interacción y; a los informacionalismos como recursos y productos objetivados, de la interacción de los participantes.

4.4 Modelo de trabajo Bazar, principales características.

Bazar es la metáfora con la cual, fue dado a conocer al mundo, el modelo de trabajo utilizado para el desarrollo del kernel Linux, a principios de la década de 1990. La perspectiva que brinda Raymond (2000), sobre el Bazar, es un modelo de desarrollo y mantenimiento del software, en el que los usuarios son tratados como co-desarrolladores por los administradores del proyecto y, como tales, se les otorga un papel activo, como agentes del proceso productivo. Para caracterizar este modelo de trabajo, se recuperan de Elliot y Scacchi (2003) y de Berdou (2007), tres elementos: acceso; economía del regalo y; meritocracia.

Acceso

La construcción de la noción de acceso contempla tres niveles, a partir de los cuales se formulan tres sentidos del acceso; el primer nivel de acceso, es la capacidad que tiene un sujeto para leer y copiar información de un sistema de información; el segundo, es el acceso que tiene un sujeto a realizar modificaciones a la información contenida en un sistema; el tercero, es la capacidad que tiene un sujeto para configurar el sistema y determinar los accesos de otros sujetos.

En el primer nivel de acceso, también llamado de "solo lectura", hace referencia a la capacidad que tiene uno o varios sujetos para acceder y copiar, por ejemplo, un documento, programa, archivo o enlace, por medio de Internet. El segundo nivel, llamado de "lectura y escritura", hace referencia a la capacidad que tiene uno o varios sujetos para realizar modificaciones, por ejemplo, en el código fuente que un proyecto desarrolla, con la advertencia de que todo error o accidente, repercutirá de forma directa en todo el proyecto. El tercer nivel, llamado de "administración", es el que tiene uno o varios

sujetos para configurar un sistema de información y asignar, los niveles de acceso a los sujetos.

Derivados de esta tipología de los niveles de acceso, se formulan tres sentidos para definir al acceso: a) la apertura de solo lectura, que los proyectos de software libre mantienen en Internet, de su código fuente y de distintos informacionalismos que componen al proyecto (Elliot y Scacchi 2003); b) como objetivación de la perifericalidad de los participantes en las comunidades de práctica (Berdou 2007) y; c) como la objetivación del poder de los participantes, en la red de sistemas de información de un proyecto de software libre (Mahendran 2002).

El primer sentido es relacionado por Elliot y Scacchi, con la serie de interfaces públicas a través de las cuales, los proyectos de software libre hacen accesible los informacionalismos del proyecto; dichas interfaces, son denominadas como *OS Support Environment (OSSE)* (2003:6), ésta noción, está relacionada con el concepto de red de comunicación y de trabajo, arriba descrito, con la aclaración de que, subraya el objetivo de mantener accesible, la mayor cantidad de informacionalismos a través de la red de sistemas de información; para facilitar de esta manera, el desarrollo del proyecto y la socialización de nuevos participantes. El acceso desde esta perspectiva, puede ser relacionado con la apertura del proyecto para atraer usuarios y socializar participantes, por medio de herramientas en línea, que le permiten a los distintos participantes, acceder a informacionalismos que, a su vez, les permitan satisfacer alguna necesidad relacionada con el software y su desarrollo.

El segundo sentido, se centra en las estrategias por medio de las cuales los proyectos administran la participación de los sujetos. Berdou (2007) relaciona el acceso, con la noción de perifericalidad para señalar que, el acceso es el medio por el cual, las actividades prestigiosas y recursos de la comunidad, son separadas de la periferia de la comunidad, donde se encuentran los recién llegados y aprendices; para centrarlos, en el centro de la comunidad, donde se encuentran los maestros (equipo central) y jóvenes maestros (participantes legítimos).

La interpretación de perifericalidad, de Berdou (2007), es situada en privilegios de lectura y privilegios de escritura; con ellos, se construyen niveles de acceso, que se relacionan con identidades; donde, la escritura en la versión central, se encuentra restringida; otras menos delicadas, como la redacción de manuales o traducciones, tienen mayor permisividad y; así, en un movimiento de centro a periferia.

El tercer sentido relaciona la estructuración de los niveles de acceso, como objetivación de estructuras sociales y de gobernanza de los proyectos; según la interpretación de Sack et.al. (2006), los niveles de acceso son una objetivación del poder de los sujetos en la estructura del proyecto, a partir de

la cual, se sostiene la división social del trabajo y la construcción de jerarquías. En este sentido, Mahendran (2002) propone dividir las relaciones de poder en tres tipos: material, discursivo y técnico.

El poder material, hace referencia al poder objetivado en los sistemas de información, por medio de los niveles de acceso, el cual, se sitúa en la capacidad que tiene un sujeto para decidir sobre, las modificaciones que se pueden o no realizar, a los informacionalismos que componen un proyecto, en particular, al código.

El poder discursivo, hace referencia a la capacidad que tienen los sujetos para manejar las discusiones del proyecto; esta visión del poder, contempla que la capacidad para manejar el discurso, está relacionado con las estructuras organizacionales de los proyectos, donde, la formulación de grupos y canales de comunicación se refuerza con el poder material, de forma que, quienes detentan el poder material, se encuentran en una posición discursiva privilegiada, frente a quienes no lo detentan.

El poder técnico refiere a las habilidades técnicas que tienen los sujetos para acceder, manejar y reformular los desarrollos (informacionalismos) de un proyecto, las habilidades, son un criterio de diferenciación entre individuos, sirve como elemento de configuración para las relaciones y las posiciones en un proyecto y también; sirve de complemento o elemento para equilibrar los otros tipos de poder. Las habilidades si son pensadas como conocimiento, en la línea de las comunidades de práctica, es el elemento central para situar, la distribución de accesos a los recursos de la comunidad.

Economía del regalo

La economía del regalo se refleja en dos soportes principales: en el acceso gratuito o a muy bajo costo, que los sujetos tienen a los informacionalismos de un proyecto de software libre y; en la disposición de los sujetos para participar de manera gratuita (voluntariado) en dichos proyectos. Por un lado, la gratuidad en el acceso (de lectura y copia) a los informacionalismos, es definida por Elliot y Scacchi (2003), como un ambiente de soporte y desarrollo de herramientas, recursos y servicios de software libre, imaginado y administrado como un bien público, compartido y alimentado colectivamente por cientos de constelaciones de comunidades de práctica de software libre. Por el otro lado, la economía del regalo, se manifiesta en la donación de esfuerzo y compromiso por parte de los participantes de los proyectos software libre, los cuales lo hacen, sin esperar una recompensa económica por la labor que realizan.

Si se relaciona la economía del regalo, como una forma de administrar los informacionalismos, con el concepto de recursividad de Kelty (2008), es posible interpretar al conjunto de informacionalismos del software libre, como una plataforma de herramientas, recursos y

conocimiento, que se desarrolla a sí misma; es decir, el conglomerado de informacionalismos del software libre (llamado también ecosistema), está diseñado para alimentarse a sí mismo, por medio del uso que la comunidad hace con ellos; cabe señalar que este proceso se facilita, porque los usuarios de estos informacionalismos, son quienes los desarrollan y mantienen, de forma que, "los requerimientos del sistema de software y su diseño toman poco tiempo en ser articulados y negociados" (Elliot y Scacchi 2003:6).

Por su parte Barbrook (2005) señala que la economía del regalo, puede ser interpretada como el libre flujo y acceso a la información en la Internet, bajo la consideración de que, ni es completamente libre, ni recubre de manera homogénea, ni a la Internet, ni a los proyectos de software libre. Al respecto, este autor señala que la economía del regalo convive, con la economía del intercambio económico, como tendencias complementarias en la Internet; desde su perspectiva, ambas posturas forman un hibridismo de intercambios.

Meritocracia

La meritocracia, sin intención de incluir una problematización teórica sobre la misma, es descrita por Bacon, un ex *Community Manager* del proyecto *Ubuntu*, como "una atribución social (forma de gobierno) basada en el talento, los logros y el mérito, que aplica para muchas comunidades" de software libre (2009:37).

La meritocracia como forma de gobierno, está relacionada con la comunicación y la colaboración, orientadas a la evaluación de las contribuciones de los participantes de un proyecto, no solo por quienes ocupan las posiciones centrales en las comunidades de práctica, sino por todos los miembros, de forma que, si los sistemas de información generan historiales detallados de los usos y aportaciones de los participantes y, el acceso de lectura está abierto, la revisión de las contribuciones de cada participante está en posibilidad de ser cuestionada o reafirmada por el conjunto de la comunidad.

En esta situación, si se involucra la noción de campo de Bourdieu, es posible interpretar al prestigio como el capital social en disputa y al ecosistema de software libre como el campo simbólico de negociación; en este sentido, los méritos de los participantes es la forma de apropiar prestigio y; la comunidad de práctica, en especial los sujetos más prestigiosos, los encargados de asignarlo.

La meritocracia no debe de considerarse exenta de la influencia de otro tipo de atributos, que intervienen en la distribución de los capitales al interior de un proyecto de software libre, por ejemplo, están aquellos proyectos híbridos, compuestos por voluntarios organizados en comunidades y empresas, donde la empresa, como institución fundamentada en la economía del intercambio

económico, negocia los objetivos y medios a seguir con los participantes voluntarios desde una posición de ventaja. Las modalidades y tipos de hibridaciones toman distintas formas, desde aquellas que mantienen las posiciones más importantes sólo con empleados (como Moblin v2), hasta aquellas que construyen órganos de diálogo con los voluntarios (organizados en comunidades) por medio de comités y consejos y, a los cuales, les es asignado posiciones poderosas en el sentido de Mahendran (2002), como ejemplo están los proyectos: *Fedora*, *Ubuntu* y *Maemo*; los cuales, trabajan empresa y comunidad de forma cercana donde, representantes de la comunidad tienen asientos en los comités organizadores.

4.5 Interpretación de necesidades subyacentes.

Interpretación de necesidades subyacentes, es un término que Habermas (2001) utiliza, para enlazar conceptualmente, la acción intencional y la acción regulada por normas. Su argumento sobre la utilidad de este enlace, radica en que, tanto las nociones de acción intencional (comportamiento dirigido fines), como de acción regulada por normas (convenciones); tomadas por separado adolecen de limitantes explicativas, sobre todo, cuando se busca enmarcar conceptualmente, la acción como un proceso dinámico, mutuamente relacionado entre sujetos individuales (subjetividad) y sujetos colectivos (intersubjetividad).

Desde la problemática de la participación, como acción-interacción entre sujetos en el marco de un proyecto de software libre, la pregunta por los motivos de la participación y, por las normas y valores que guían la participación; aparecen aislados en las discusiones teóricas de las comunidades de práctica, de la comunicación mediada por computadora y del modelo de trabajo Bazar. En este sentido, resulta de utilidad utilizar, el concepto de interpretaciones de necesidades para abordar la relación entre, los objetivos de los sujetos para participar y, las normas de la comunidad que guían la participación.

Adicionalmente, trabajar sobre con concepto de interpretación de necesidades subyacentes, de la forma como Habermas (2001), como encuentro entre intenciones, normas, valores y códigos; brinda elementos para comprender, de una manera más productiva, los procesos de comunicación utilizados en el desarrollo de Moblin v2.

Habermas (2001) en su trabajo *Intención, convención e interacción lingüística* (original de 1976), plantea la problemática de las limitantes, en los tejidos conceptuales de acción intencional y de acción regulada por normas. Según su perspectiva, ambos conceptos, además de estar entrelazados, al tratarse por separado, llegan a finales analíticos, como el concepto de "motivos" del sujeto para la acción, o el de "normas" que constriñen la acción; que no permiten llegar a una comprensión más

profunda de la acción social.

El argumento de Habermas (2001) se sitúa en dos planos, el primero, es el plano lingüístico y, el segundo, es el plano social de la acción, desde su perspectiva ambos planos están relacionados, sin embargo, es un error considerar que el tránsito analítico de uno a otro sea directo o transparente. En esta tesis se toma el plano lingüístico, como un referente a partir del cual comprender el plano social de la acción.

El punto de partida de la discusión de Habermas, al respecto de la acción de los sujetos, es que, el comportamiento regido por normas sigue una intención, y la acción orientada a fines, sigue reglas, normas y códigos; en este sentido, el sujeto al actuar, está accionando simultáneamente, una convención y una intención por medio de "seguir una interpretación". Si se toma el ejemplo del lenguaje, para expresar lingüísticamente una intención es requisito hacer uso de un conjunto de reglas del lenguaje, que permitan a otros sujetos la comprensión de una intención.

Las intenciones entendidas desde un plano de acción teleológica, es decir, "al designio, voluntad o propósito que un sujeto tiene de conseguir algo o realizar un fin"; comprenden esperanzas, temores, expectativas, emociones, inclinaciones, deseos, odio, vergüenza, amor y, "en un sentido lato" todo aquello relacionado con lo subjetivo de los individuos (Habermas 2001:264).

Para fundamentar esta relación entre intenciones y subjetividad, Habermas señala que toda intencionalidad está compuesta por dos elementos. La i1, contempla la relación cognoscitiva que el sujeto establece con la realidad objetivada en la forma de una pretensión de verdad, que puede ser entendida como el elemento racional que fundamenta la intención. La i2, por su parte, hace referencia a la toma de postura emotiva, del sujeto frente a la pretensión de verdad (i1).

Por ejemplo, en relación con el software libre, un sujeto puede decidir participar con un proyecto, porque necesita una aplicación o librería, así, tiene una pretensión de verdad sustentado por un elemento racional: el software sirve (i1) para satisfacer un objetivo determinado. Pero además, el mismo sujeto, elige una opción entre distintas opciones igual de útiles, por ejemplo, porque considera que en una de ellas, el código es elegante, optimizada o, eficiente (i2). El sujeto, por un lado, establece una relación directa con el software (realidad objetivada) que le permite saber, que el software le es de utilidad para satisfacer una necesidad (elemento racional). Por el otro lado, el sujeto sigue un patrón de subjetividad relacionado con una interpretación, compartida por una comunidad de programadores, que le permite relacionar ciertas características técnicas del software, con los adjetivos elegancia, optimización o eficiencia; los cuales, llevan consigo una toma de postura del sujeto, con respecto a

determinadas formas de entender al software.

Habermas aclara que, para que un sujeto esté en condiciones de "actuar intencionalmente", antes tiene que estar en "condiciones de expresar su intención en circunstancias apropiadas" (2001:267); esto implica que el sujeto posee un lenguaje intencional a partir del cual construir su situación y su perspectiva de acción. Al punto de encuentro que constituye el acto lingüístico de discursivizar una intención de un sujeto, Habermas lo llama *vivencia intencional*, este último concepto es de importancia ya que, es en la vivencia que el sujeto incorpora tanto su relación con la realidad como con sus adscripciones a ciertas interpretaciones.

Esta postura contempla que la vivencia intencional, como construcción discursiva, es situada después de la acción, de forma que, las intenciones que provocan una acción, al ser recordadas o justificadas por los sujetos, se les integran interpretaciones sobre las metas que buscaban lograr y, las reglas, normas y valores que intervinieron como condicionantes de la acción. (Habermas 2001:269).

Un ejemplo para ilustrar las construcciones lingüísticas, está en el trabajo de Ghosh (2005), que se interroga sobre las motivaciones que tienen los sujetos para participar de manera voluntaria en proyectos de software libre, las respuestas que recupera contempla expresiones como: "lo necesito", "me divierte" o, "de esa forma regreso algo a la comunidad". En estos casos se observa una relación cognitiva con la realidad "lo necesito", un sentimiento hacia la participación "me divierte" y, una justificación que involucra valores "regresar algo a la comunidad".

Para objetivar el vínculo entre intención y valores del sujeto, en la justificación discursiva que realiza de una acción, Habermas (2001:270) regresa al concepto de *interpretación de necesidades subyacentes*; si se toma como verdadera la afirmación de que los fines, son elementos descriptivos de la justificación y, los valores, son elementos evaluativos; se tiene que una interpretación de necesidades subyacentes contiene armónica y conflictivamente, *elementos descriptivos*, como fines e intenciones y, *elementos evaluativos*, como sentimientos y estados de ánimo.

Habermas argumenta, que la acción orientada a valores, desde la perspectiva de la interpretación de necesidades subyacentes, involucra a la moral como un nivel de validez que entra en juego en la elección de los medios que un sujeto toma para realizar una acción intencional. Esta postura relaciona una ética empirista al esquema teleológico de la acción, de forma que, el supuesto de que una acción pueda estar únicamente orientada por intenciones u objetivos, no sea plausible (2001:207).

La debilidad del modelo de la acción intencional radica en la valoración que los sujetos hacen sobre su situación. Dicha valoración, está relacionada con creencias del sujeto sobre un estado en el

mundo, construidas a partir de proposiciones que otros sujetos han realizado, desde el supuesto de que otros sujetos tienen "situaciones de vivencia motivadoras de ese mismo tipo" (Habermas 2001:272).

Cuando la justificación de la acción que el sujeto realiza, implica a alguna de estas interpretaciones compartidas sobre situaciones o vivencias, recurren a caracterizaciones de objetos o situaciones que los llevaron a reafirmar una intención, por medio de *estándares generales de valoración*, los cuales se constituyen como tales, en la medida en que esas caracterizaciones de objetos y situaciones sean reconocidas por otros. Cuando estos estándares de justificación se encuentran lo suficientemente compartidos, Habermas (2001:273) comienza a hablar de normas; las cuales, sirven como elemento común a la justificación de los agentes, de forma que, todo agente al ser interrogado por sus acciones, "tiene el derecho" de recurrir a determinados valores para ponerlos en la base de sus necesidades para justificar sus actos.

Como ejemplo, en el caso de estudio de *GNUe* de Elliot y Scacchi (2003), su observación indica que, valores como la libertad de elección o de libre acceso a la información, son valores respetados y defendidos por los participantes del proyecto, al grado que son un elemento objetivado, en sus sistemas de información y en la formulación de sus jerarquías (accesos).

Si se considera que, este modelo de administración de los sistemas de información del software libre, no es exclusivo del proyecto GNUe, sino que es una práctica compartida que conforma un ambiente de soporte de software libre (ecosistema), constituido por una pluralidad de proyectos que comparten y defienden la libertad de elección y de acceso; según los elementos que Habermas (2001) brinda, podrían ser pensados como una norma fundada en valores que configura prácticas y sistemas de información, bajo la consideración que:

1. Sea posible, reconstruir intersubjetivamente las necesidades que están en la raíz de las intenciones de los sujetos, cuando toman postura frente a objetos y situaciones. Bajo la consideración de que, las necesidades aparecen bajo interpretaciones que suponen:
 - a. Comunidad de lenguaje.
 - b. Lenguaje con expresiones evaluativas.
 - c. Una tradición de valores culturales intersubjetivamente compartida.
2. Muestren que estos valores compartidos son candidatos a quedar materializados en normas, es decir, como orientaciones de valor que rigen en determinadas situaciones (Habermas 2001:276).

Un elemento fundamental para la problematización de las normas fundadas en valores, es la percepción compartida de que la norma responde a un interés común, o en palabras de Habermas es "expresión de un interés general" (2001:277). Este último punto integra la legitimidad, al escenario de la construcción de la intersubjetividad de una comunidad.

4.6 Balance de los lentes conceptuales

Del tejido de las comunidades de práctica, se pone el acento, primero, en el concepto de perifericalidad, que permite entender los mecanismos por medio de los cuales se formulan las jerarquías y la división del trabajo en Moblin. Y segundo, en el concepto de participación, a partir del cual se definen a los sujetos como participantes y, a los procesos de comunicación que establecen, como formas de interacción.

Del concepto de red de comunicación y de trabajo, el acento se pone; por un lado, en la red de distintos sistemas de información, como canales de comunicación en línea, especializados en satisfacer necesidades particulares de organización, comunicación y trabajo, en especial, soportar los espacios de discusión, documentación e implementación. Por el otro, esta red de sistemas, busca hacer accesibles, producir, organizar y administrar informacionalismos, pensados, como recursos necesarios para la participación y, como objetivación de la interacción y trabajo de la comunidad de práctica.

Del modelo de trabajo, el foco se centra en la distribución del acceso, como objetivación del poder que cada participante posee, de forma que, las jerarquías sean representadas como articulación entre poder material, discursivo y técnico; que posibilitan y condicionan, la interacción de los participantes. La economía del regalo y la meritocracia, por su parte, sirven como elementos que movilizan el significado de la participación, como recurso que nutre la distribución del poder al interior de los proyectos de software libre.

Del concepto de interpretación de necesidades subyacentes, se retoma a las necesidades, como la articulación de: objetivos enmarcados por criterios de verdad técnicos; gustos, deseos y emociones, que relacionan los objetivos con la subjetividad de los participantes y; valores y normas, que posibilitan y condicionan las acciones que los distintos tipos de sujetos realizan para obtener sus objetivos.

Estos lentes conceptuales, se orientan para comprender al proyecto como un proceso en el cual, los participantes participan de manera diferenciada, con intensidades y frecuencias variables, que pueden delimitadas en base a su perifericalidad, mientras más cercanas al centro, más intensas.

La participación es producto de una o varias necesidades, las cuales, para ser satisfechas

requieren existir como acciones comunicativas sujetas a reglas contextuales de la red de comunicación y de trabajo del proyecto; muchas de estas reglas, dependen del modelo de trabajo Bazar (acceso, economía del regalo y meritocracia), pero otras, pertenecen a otros contextos que escapan al software libre, como el uso del lenguaje Inglés o códigos para utilizar de manera correcta las listas de correo.

La participación como proceso contempla que los participantes tienen negocian con la red la satisfacción de sus necesidades y a su vez, la red negocia con los participantes el mantenimiento de su forma (Sack et.al. 2006) y, durante el proceso, la práctica se objetiva en informacionalismos, los cuales, servirán como base para nuevas negociaciones, de forma que el proceso se sostenga a lo largo del tiempo, hasta que actores externos intervengan y cambien la forma, procesos y relaciones de la red.

Respecto a las limitantes de los lentes conceptuales, está en primer lugar, que la propuesta de las comunidades de práctica (Wenger 2001), dificulta problematizar la lucha y el conflicto en su interior, como perspectiva, tienden a privilegiar las relaciones armoniosas y los puntos de acuerdo.

Con respecto al resto de los conceptos utilizados, el principal problema es la simplificación en el uso que se hace de ellos; en el modelo de trabajo Bazar, se privilegia la atención en el acceso y se descuida la economía del regalo y, la meritocracia; en el concepto de red de comunicación y de trabajo, se descuida el análisis en las relaciones que establecen entre sí los participantes, para central las relaciones entre participantes y sistemas de información y, participantes con informacionalismos; en el caso de las interpretaciones de necesidades subyacentes, es más notoria esta simplificación, ya que se deja de lado la complejidad del planteamiento teórico de Habermas (2001), para solo hacer uso del término necesidades.

5 Objetivo, pregunta, método.

5.1 Objetivo

Caracterizar los procesos de comunicación utilizados en el desarrollo de Moblin v2 a partir de: las herramientas de comunicación y de trabajo utilizadas y; las necesidades de los distintos tipos de participantes. Dicho análisis, enmarcado en el el modelo de trabajo Bazar, expuesto por Raymond (2000), Berdou (2007) y Ducheneaut (2005).

5.2 Pregunta

¿Cómo se articulan, en los procesos de comunicación que tienen lugar, en el proceso productivo del proyecto Moblin v2: las herramientas de comunicación y de trabajo; los modelos de trabajo; y las interpretaciones de las necesidades subyacentes (Habermas 2001) de los participantes?

5.3 Método

Para construir la caracterización de los procesos de comunicación, cercanos al concepto de participación, como se ha mencionado repetidamente, se recurre a herramientas de comunicación y trabajo, modelo de trabajo Bazar e, interpretación de necesidades subyacentes. Desde la consideración de que cada uno de estos elementos es relativamente independiente de los otros, se ha optado por dividir la metodología en dos etapas.

La primera etapa se centra en brindar los elementos para caracterizar las herramientas de comunicación y de trabajo y, consiste en, la observación y conteo de la red de sistemas de información y, de los informacionalismos de Moblin v2. La segunda etapa, busca brindar elementos para la caracterización de los participantes y sus necesidades; la articulación entre procesos de comunicación y procesos de producción y; la puesta en práctica del modelo de trabajo Bazar en el proyecto Moblin⁴⁵; todo ello, a partir de un análisis de contenido de las conversaciones mantenidas entre Mayo y Diciembre del 2009, en la lista de correos Moblin-dev.

La operacionalización que se proponen para caracterizar la articulación en los procesos de comunicación de Moblin, de herramientas de comunicación y de trabajo, modelo de trabajo Bazar e, interpretación de necesidades subyacentes de los participantes; es:

- Herramientas de comunicación y de trabajo
 - Red híbrida de sistemas de información
 - Sitio web
 - Lista de correo Moblin-dev
 - IRC (chat)
 - Transifex
 - Bugzilla
 - Git
 - Informacionalismos
 - Manuales
 - Guías
 - Documentación
 - Repositorios
 - Herramientas de desarrollo

45 Quedó pendiente la realización de una tercera etapa, cuyo objetivo era, complementarla caracterización del modelo de trabajo Bazar y, aportar elementos para la exploración de las necesidades subyacentes de los participantes, a partir de, primero, una reconstrucción de ciertas conversaciones que ilustren la forma en que los participantes hibridizan los informacionalismos y la red de sistemas de información y; segundo, entrevistas con ciertos participantes en el canal IRC (chat) de Moblin v2.

- Librerías
 - Aplicaciones
- Canales de ayuda
 - Lista de correo Moblin-dev
 - IRC (chat)
 - Informacionalismos externos a Moblin v2
- Código
 - Repositorios
 - Garage
 - Bugzilla
 - Git
- Participantes
 - Participantes centrales
 - Administradores (core team)
 - Desarrolladores
 - Parchadores
 - Participantes periféricos
 - Vendedores
 - Patrocinadores
 - Empleados de Intel
 - Usuarios finales
- Modelos de trabajo
 - Etapas del proyecto
 - Fechas de entrega
 - Lanzamientos
 - Procesos
 - Usos de los sistemas de información
 - Tipos de sistemas de información
 - Trabajo
 - Comunicación
 - Tipos de participación que posibilitan
 - Acceso
 - Control
 - Participación
 - Tipos
 - Jerarquías
 - Dirección de correo electrónico
 - Agencias
 - Responsabilidades
- Interpretación de Necesidades Subyacentes (Habermas 2001)
 - Intenciones
 - Relación cognoscitiva con la realidad
 - Conocimiento sobre los objetivos

- Postura
 - Deseos, sentimientos, opiniones, prejuicios, que guían el objetivo
- Normas
 - Justificaciones discursivas de las acciones
 - Reglas
 - Valores

Primera etapa: Observación como punto de partida para el análisis

Se enfoca a contabilizar los elementos que componen al proyecto Moblin v2, el conteo comienza por identificar las herramientas de comunicación y trabajo, como redes híbridas de sistemas de información, infomacionalismos y participantes.

El periodo de tiempo seleccionado para la observación es de Enero a Diciembre de 2009, el año de actividad del proyecto Moblin v2. Del universo que componen las herramientas informáticas utilizadas, la muestra se reduce a las que se considera, son centrales:

1. Sitio web
2. Lista de correo Moblin-dev
3. Bugzilla
4. Git

Se tomó la decisión de eliminar a Transifex y al IRC (chat), por no tener el tiempo suficiente, ni la capacidad para hacerlo, por tal, han sido dejados de lado con la consideración de que de esta forma se reducen al mínimo los efectos negativos de dejar de lado un sistema.

Los objetivos que se esperan cubrir con esta etapa son:

1. Describir las herramientas informáticas de comunicación y de trabajo utilizadas para el desarrollo del proyecto Moblin v2.
2. Describir a los participantes: número, tipos y posiciones.
3. Describir en términos generales el proceso de producción de Moblin v2, desde una mirada que enfoque, las relaciones de las herramientas en el proceso productivo, con las etapas de producción y los contenidos producidos.

Segunda etapa: Análisis de contenido (AdC)

Al avanzar en la primera etapa de observación, fue haciéndose evidente que los informacionalismos no podían ser abordados solo mediante su conteo, y, que la variedad y amplitud de los mismos, no permitiría profundizar más en cada uno de ellos. Esto condujo a elegir unos informacionalismos, de un sistema de información, que dieran cuenta del proceso general de Moblin v2. Se eligió la lista de correo Moblin dev, por considerar que en ella: a) es posible observar la interacción entre los distintos tipos de participantes, gracias a que todos los mensajes se archivan y son accesibles por Internet; b) se maneja un lenguaje accesible al observador externo; c) se discuten y anuncian temas relacionados con el desarrollo de Moblin v2 y; d) a partir de esas discusiones, es posible caracterizar, los procesos de desarrollo.

Bajo la consideración de que la lista de correo Moblin-dev contiene una gran cantidad de mensajes y conversaciones, se utiliza el análisis de contenido (de aquí en adelante AdC) porque es una técnica de investigación que trabaja con muestras estadísticas de análisis. El AdC es utilizado, en base al trabajo de Krippendorff (1990), que lo define como "una técnica de investigación con orientación empírica y exploratoria, de utilidad para abordar fenómenos simbólicos en forma de mensajes".

La unidad de sentido, sobre la que se hace el análisis, son las conversaciones, las cuales; se componen de uno o varios mensajes, intercambiados por uno o varios participantes. La utilidad de hacerlo de esta forma, es que el sentido de los mensajes se construye con base en otros mensajes, y juntos, constituyen una unidad de sentido, que comparte un mismo nombre y una misma problemática. Se toma como eje articulador al primer mensaje de cada conversación, al mismo, se le llama acción inicial.

La selección temporal de la muestra para las conversaciones de la lista de correo Moblin-dev, va del mes de Mayo a Diciembre de 2009, se toma este periodo de tiempo por ser, el que más actividad, por parte de participantes periféricos, registró (851 conversaciones) y; porque en este periodo se liberaron dos nuevas versiones de Moblin v2, lo cual permitió observar parte del proceso de desarrollo, en especial, la participación de los sujetos denominados "intermediarios".

El muestreo, por conglomerados y aleatorio, según la terminología de Krippendorff (1990), fue formulado con el 17% del total de las conversaciones, es decir, 145 conversaciones; la forma de extraerlos, fue tomar al azar 20 conversaciones de cada mes, de las cuales, las conversaciones repetidas (3% del universo muestral), fueron eliminadas. De la muestra, se buscó en primer lugar definir los temas que se discutían, y en segundo, abordar cualitativamente los procesos dialógicos en las,

conversación, es decir, el tipo de procesos de interpretación y de negociación de sentido que tomaron lugar en la lista de correo Moblin-dev.

A continuación se presenta un desglose general de las variables y, un desglose particular de las categorías utilizadas en las variables que se consideran más importantes, las mismas, pueden verse en negritas en la tabla que a continuación se presenta.

Desglose de variables.

Inicio-Fin	Señalan la fecha del primer mensaje y el último.
Duración	Señala el número de días que dura cada conversación.
Título de la conversación	Asignado por el autor del primer mensaje
Forma expresiva	Tipifica el primer mensaje de la conversación (acción inicial); por lo observado éste mensaje caracteriza a la conversación, le da nombre y, según el tema que trata y su forma en que se expresa, la conversación se desenvuelve de maneras distintas.
Propósito	Acción o necesidad que el sujeto busca realizar o satisfacer, por medio de la ayuda que otros sujetos brindan a través de Moblin-dev.
Tema general	Objeto, componente o dispositivo técnico sobre el cual se discute. Trata de ser lo más general posible.
Tema particular	Nombre del objeto, componente o dispositivo técnico al cual se hace referencia en las conversaciones, busca ser lo más específico posible.
Proceso de conversación	Describe el flujo de la comunicación a partir de números, flechas, paréntesis, signos de suma y de resta. Más adelante se describe con más detalle la codificación.
Estructura de la conversación	Describe, en base al modelo de trabajo colaborativo de Gunawarderna (1994), las distintas etapas por las que, las conversaciones, en la comunicación mediada por computadora, transcurren; de lo simple a lo complejo. Donde la acción inicial, desencadena ciertos tipos de construcciones cognitivas relacionadas con el intercambio e interpretación de la información. Este modelo tiene la virtud de permitir abordar posturas, acuerdos, construcciones y

	reformulaciones.
Mensajes	Número de mensajes de la conversación.
Participantes	Número, nombre o alias de los sujetos que participaron en la conversación.

Tabla 1. "Desglose de variables"

Forma expresiva

La variable "forma expresiva", busca agrupar las distintas formas discursivas a las que los sujetos recurren para expresarse, en la acción inicial de las conversaciones. Se considera de utilidad centrar la mirada, en la acción inicial y su forma expresiva, ya que, sirve como eje de articulación a las conversaciones que se construyen, en la lista de correos Moblin-dev. El análisis de la acción inicial será formulado con articulación de la forma expresiva, el propósito y los temas; de forma que, haya los suficientes elementos para caracterizar a las acciones iniciales de las conversaciones, que a su vez, serán tomadas como un elemento que permanece a lo largo de las conversaciones.

Las categorías de la forma expresiva son las siguientes:

Pregunta	Expresa una necesidad, problema, duda o solicitud de ayuda en forma de pregunta.
Solicitud de ayuda	Redacta su necesidad en forma de solicitud, por ejemplo, un participante rompe un archivo y solicita que alguien le pase el suyo, o, solicitan ayuda para resolver una duda.
Anuncio	Dar a conocer algo, un lanzamiento, un proyecto, aplicación
Sugerencia	Sugerir una acción a realizar por el equipo de Moblin.
Queja	Expresa descontento.
Reporte	Expresa un mal funcionamiento, como un mal funcionamiento, se distingue de la pregunta por su carácter afirmativo.

Tabla 2: "Categorías de la forma expresiva".

Propósito

Busca dar cuenta de las necesidades que los participantes buscan satisfacer a través de la acción inicial. A partir de su categorización, se busca inferir los usos que hacen los participantes de la lista de correo Moblin-dev, pensada, por un lado, como un canal de ayuda que el proyecto Moblin v2 pone a disposición de sus usuarios y, por el otro, un instrumento de socialización.

Las categorías del propósito son las siguientes:

Desarrollar	Generar un nuevo producto, en la mayoría de los casos, desarrollo de aplicaciones para Moblin por parte de participantes periféricos. En esta categoría se incluyen las necesidades referentes a las estaciones de trabajo utilizadas por dichos participantes, en especial, la relación entre sistemas operativos, procesadores y tarjetas de video.
Usar	Utilizar un hardware con Moblin, por ejemplo, un modelo de computadora, un periférico como touchscreen, una tarjeta (video o wireless); por ejemplo, probar Moblin en una computadora y que todo funcione bien (no incluye problemas relacionados con desarrollo).
Modificar	Hace referencia a un cambio al repositorio central de Moblin, como, cambios a la Interfaz Gráfica; cambiar una de sus aplicaciones; o integrar una aplicación.
Configurar	Manipular de cierta forma a Moblin para lograr un resultado específico, como activar una aplicación al bootear. Puede pensarse de forma similar a "modificar", pero en lugar de intervenir en el repositorio de Moblin v2 del Intel-OTC, se modifica el Moblin del participante.
Arreglar	Solucionar problema con un componente o aplicación ya existente en Moblin v2; por ejemplo, la tarjeta wireless, que se puede usar, pero que su desempeño es deficiente (hay que arreglarlo).
Soportar	Integrar una aplicación al repositorio central de Moblin, lo cual incluye, que el proyecto le de mantenimiento.
Integrar	Aplica a re-configurar aplicaciones ya existentes para poder ser utilizadas en Moblin v2 (excluye a desarrollos de los participantes).
Aportar	Brindar un parche o desarrollo, para que sea utilizado por la comunidad y/o integrado a Moblin v2.
Promocionar	Anunciar que hay una nueva aplicación, desarrollo o proyecto. Busca que otros participantes estén al tanto, lo usen o participen.
Conocer	Preguntan el por qué de la arquitectura y configuración de Moblin v2. Incluye dudas de configuración, ubicación de archivos o, ciertos procesos.
Investigar	Da cuenta de pretensiones de los participantes por conocer a profundidad a Moblin v2 o, una de sus aplicaciones.

Tabla 3: "Categorías de la variable propósitos".

Flujos de conversación

Buscan representar gráficamente los flujos de la comunicación por medio de símbolos que no guardan relación con formulas matemáticas, pero que recurren a paréntesis, números, signos de suma, resta, mayor que y menor que; sin embargo, un nuevo significado es asignado a cada signo, para ayudar a que describan procesos de comunicación, a continuación la codificación para entenderlos:

Números	(1) (3)	Representan a los sujetos, si se encuentran solos, no representan una conversación; si se encuentra entre paréntesis, un número superior a uno, significa que hay una conversación entre el número representado de participantes.
Paréntesis	()	Representan una conversación; se usan para hablar de la conversación en general, y también, para referir a conversaciones que tienen lugar dentro de la conversación mayor.
Flechas mayor que y menor que	<- -> <= =>	Representan el flujo de la conversación, si es “menor que”, quiere decir que miembros de la lista están ayudando o dándole respuestas al sujeto que hace el primer mensaje. Si es “mayor que”, se hace referencia a que la acción inicial es una sugerencia, queja, contribución, bug o mensaje del participante hacia la lista. Si es menor que, indica que los participantes de la lista dan consejos, respuestas o soluciones al autor de la acción inicial. Cabe señalar que gráficamente, representan flechas.
Suma	+	Sirve como enlace para distintos momentos por los cuales transcurre una conversación.
Menos	-	Significa que los mensajes, no aportan elementos para la construcción de soluciones o de problemáticas.
Separador.		Significa que los mensajes están completamente

		separados.
--	--	------------

Tabla 4: "Categorías de la variable flujos de conversación".

Así por ejemplo: $(1+2 \leq 7)$. Da cuenta de: un primer participante que hace una pregunta, de la cual, se desata una diálogo entre dos sujetos y, otros siete, aportan elementos para responder a la pregunta del primer sujeto y resolver o complementar el diálogo de los dos sujetos.

Otro ejemplo: $((1 \leq 1)+1)$. Da cuenta de un participante, que hizo una pregunta y otro la respondió, sin embargo, un tercer sujeto hizo un tercer comentario que, al estar posicionado fuera de los paréntesis, su participación se mantuvo periférica al proceso de pregunta-respuesta de los otros sujetos, pero no tan separada, ya que hay un signo de suma que sirve de enlace.

Un tercer ejemplo: $(1 \leq 4)$. Este ejemplo da cuenta del flujo más común de las conversaciones, en donde, el primer participante hace una pregunta y otros cuatro responden, de forma complementaria a su pregunta.

Un último ejemplo: $(1 \Rightarrow (1) \leq 1)$. En este caso, el participante que realiza la acción inicial (A1) se posiciona en el centro y mantiene pláticas separadas con un par de participantes (B1 y B2) que responden separadamente a su pregunta. Puede interpretarse como que, el A1 primero mantiene una conversación con B1 y, por separado, sostiene otra con B2.

Estructura de la conversación

Esta variable busca clasificar las conversaciones, según la calidad de la construcción de sentido que, los participantes consiguieron; la construcción de sentido, sin pretender profundizar en su definición, es considerada un proceso de negociación, construido a partir del intercambio e interpretación de mensajes, por parte de los participantes de una conversación. Desde la perspectiva de Gunawarderna (1997), este proceso de intercambio de mensajes e interpretación, es denominado "interacción".

La calidad de la construcción de sentido o, la calidad de la interacción, es categorizada en base a los procesos de interpretación que cada participante, da cuenta a través de sus mensajes en una conversación. Un elemento y limitante de utilidad, para situar estos procesos, es que en la lista de correo de Moblin v2, solamente se tratan temas relacionados con el uso, adaptación y desarrollo de software; el cual, puede ser entendido como un límite temático que circunscribe las conversaciones. Se dice que es una limitante, porque por ejemplo; negociaciones de sentido relacionadas a relaciones de poder en el proyecto, que aunque son observables, no permiten una reconstrucción satisfactoria.

Desde la consideración de que el límite temático es el software que compone, o puede ser

utilizado en Moblin v2; las conversaciones dan cuenta de un proceso de aprendizaje colectivo en el cual, quienes tienen el conocimiento, ayudan a quienes tienen una necesidad y, utilizan la lista de correo de Moblin-dev para satisfacerla; también, para ayudar a quienes tienen la responsabilidad del proyecto, a darse cuenta de lo que sus usuarios (o co-desarrolladores según la interpretación de Raymond 2000), necesitan; de forma que, unos reciben ayuda de los expertos y los expertos, observan sus fallos, omisiones y éxitos a través de los mensajes de los participantes periféricos.

La forma final que toma la adaptación del modelo de Gunawardena (1997), tiene cinco niveles: exposición; intercambios de información; discusiones; construcciones y; cierres:

Etapa 1: Forma expresiva de la acción inicial		
Pregunta	1P	Cualquier tipo de pregunta, en general, pregunta para satisfacer una necesidad.
Reporte	1R	Reportar un asunto, fallo, omisión o error en el código, que pueda ser observado a través del uso que le da el participante de la acción inicial (A1).
Anuncio	1A	Un anuncio emitido por cualquier tipo de participante al resto de los suscriptores de la lista de correo, en general, emitido por los participantes centrales (administradores y desarrolladores primarios).
Aportación	1Ap	Es un tipo de anuncio en el que A1 da a conocer una mejora, configuración o modificación, realizada al código de Moblin v2.
Sugerencia	1S	Sugiere nuevas funcionalidades o, regresar a viejas funcionalidades de Moblin v2 o de alguno de los elementos que lo componen.
OT (fuera de tópico)	1OT	Es un mensaje que no tiene relación con Moblin v2 y es señalado como tal por los participantes, en especial, los centrales.
Etapa 2: Intercambios de información a partir de la acción inicial.		
Intercambio de información	2II	Aunque información es un término amplio, se consideran a aquellos mensajes que dan instrucciones para que A1 pueda satisfacer su necesidad.
Intercambio de enlaces	2Ie	Considera a todo aquel enlace que sea intercambiado, sea para formular la acción inicial o, para resolver la necesidad expresada en la acción inicial. En general, son enlaces a otros proyectos o, a

		sistemas de información de Moblin v2.
Intercambio de Documentos	2Id	Aunque en términos generales, es un intercambio de enlaces (documentos enlazados), se realiza la distinción para limitar la generalidad de la categoría de enlaces.
Intercambio de código	2Ic	Incluye comandos y código, escritos en el mensaje, por lo general, complementa al intercambio de información.
Solicitudes de más información	2Si	En muchos casos, A1 no da la suficiente información para que otros sujetos entiendan su necesidad, en estos casos, los participantes solicitan información detallada (<i>logs</i>) sobre la necesidad de A1.
Solicitudes de documentación	2Sd	Hace referencia a solicitudes de los participantes a A1 para que revise los informacionalismos disponibles antes de realizar la acción inicial. Una forma de señalar que no les haga perder su tiempo con algo que solo puede solucionar.
Etap 3: Discusiones en donde se negocia, la satisfacción técnica de las necesidades expresadas en la acción inicial; la utilidad de cierto proyecto o elemento en o para Moblin y; el futuro del proyecto Moblin v2.		
Solicitud de clarificación	3Cl	Se le brinda esta categoría a aquellos mensajes, en los que, aunque hay la información suficiente, hay elementos no bien explicados o, entendidos, que un participante solicita que se profundice. Es común que estas clarificaciones sean acompañadas de interpretaciones que buscan evaluar una proposición.
Argumentación	3Ar	Son mensajes que sustentan una proposición mediante uno o varios argumentos racionales, coherentes entre sí.
Contra-Argumentación	3Ca	Son mensajes que buscan poner en cuestión o discutir una o varias proposiciones, mediante uno o varios argumentos racionales, coherentes entre sí y con la proposición que busca desestabilizar.
Uso argumentativo de evidencias	3Ev	Se considera evidencia a todo aquel recurso utilizado para sustentar una proposición, generalmente, son mensajes que el sistema operativo arroja (<i>logs</i>), como informes de errores. También, son comandos o programas que sirven para despejar una controversia, duda o necesidad.
Soluciones	3Csol	Un sujeto brinda una solución a la necesidad planteada por A1, puede ser un enlace, un comando o una instrucción. Se distingue

		por ser un tipo de cierre a la discusión, que no implica la construcción cognitiva o informativa de la discusión.
Etapa 4: Construcciones cognitivas e informativas realizadas con los intercambios de información y las discusiones.		
Soluciones grupales	4Sg	Refiere a una solución construida grupalmente a partir de argumentaciones, contra-argumentaciones y, o, evidencias.
Síntesis de discusiones	4Sd	Síntesis de la discusión neutral, es decir, sin la afirmación de acuerdos o no acuerdos.
Síntesis de no acuerdos	4Sna	Síntesis de puntos de diferencias, posiciones encontradas o no acuerdos.
Afirmación de posturas	4Ap	Los sujetos afirman posiciones valorativas o subjetivas en relación con las argumentaciones o proposiciones expresadas en la conversación.
Construcción de acuerdos	4Ca	Expresa una posición consensuada sobre el tema discutido, un punto de encuentro entre lo fáctico y lo valorativo, en relación con las proposiciones expresadas en la conversación.
Cambio de enfoque sujeto	4CeS	Hace referencia a cambios de perspectivas, posiciones o creencias por los participantes periféricos a Moblin, en varias de las ocasiones por parte de A1. También puede entenderse como un aprendizaje que reconfigura los marcos interpretativos de dichos participantes.
Cambio de enfoque de Moblin	4Ce M	Expresa un cambio de perspectiva de uno de los participantes centrales de Moblin v2, lo cual, puede entenderse como un cambio que cambiará la estructura del desarrollo de Moblin. Este tipo de cambios de enfoque son poco comunes y significan una reformulación, mayor o menor del proyecto, producto de la agencia de los participantes.
Etapa 5: Desenlace de la discusión, también puede entenderse como cierre.		
Desenlace	5	Utilizado para señalar el final de la conversación cuando no hay mayores elementos para señalar que la conversación se terminó.
Aplica el Sujeto	5AS	Se utiliza cuando A1 u otro participante que comparte la necesidad expresada por A1, aplica con éxito las soluciones brindadas por los participantes.
Bug	5B	Señala que hay consenso sobre un mal funcionamiento, omisión o

		error en el código; generalmente es canalizado a Bugzilla para darle seguimiento.
En construcción Moblin	5CM	Es utilizado por los participantes centrales para señalar que, la situación discutida en la conversación, está siendo desarrollada y próximamente estará disponible.
Abandono Sujeto	5AbS	A1 abandona la conversación, no responde más.
Abandono Lista	5AbL	La lista de correo deja de responder a los mensajes de A1.

Tabla 5: "Categorías de la variable estructura de la conversación".

6 Primera etapa. Observación como punto de partida para el análisis

Se enfoca en contabilizar los elementos que componen la red de comunicación y de trabajo del proyecto Moblin v2; el conteo comienza por identificar las herramientas de comunicación y trabajo utilizadas (también llamadas sistemas de información); los informacionalismos producidos e intercambiados y; los participantes.

El periodo de tiempo seleccionado para la observación es de Enero a Diciembre de 2009; el universo que componen las herramientas de comunicación y de trabajo, son seis (sitio web; listas de correo; canal de chat; transifex; bugzilla y git); de las cuales, se seleccionan cuatro (sitio web; lista de correo Moblin-dev; bugzilla y git) para la observación. A su vez, de éstas cuatro se toma a la lista de correo Moblin-dev, para realizarle un análisis de contenido.

Herramientas	Función	Descripción	Análisis de contenido
Sitio Web moblin.org	Aparador para dar a conocer al proyecto	Sí	No
Lista de correos Moblin-dev	Canal de ayuda, donde los distintos participantes pueden conversar unos con otros	Sí	Sí
Canal de chat (IRC)	Canal de ayuda, donde los distintos participantes pueden conversar unos con otros	No	No

Transifex	Sistema para administrar las traducciones del proyecto	No	No
Bugzilla	Sistema de rastreo y administración de errores	Sí	No
Git	Sistema de control de versiones distribuido, para administrar, el desarrollo del código, del proyecto.	Sí	No

Tabla 6: "Sistemas de Información considerados".

La razón por la cual no se abordará en el análisis al canal de IRC, es que no genera historiales que permitan acceder a, las conversaciones que allí tienen lugar, lo que no permitió realizar una reconstrucción o análisis. En el caso de *Transifex* (un sistema para administrar traducciones), fue uno de los últimos sistemas en ser implementados y, no hubo tiempo suficiente para investigarlo.

La forma de abordar la descripción de las herramientas, comienza con una caracterización; un recuento de los informacionalismos que contienen, y; un recuento de los participantes. Para cerrar, se brindarán elementos generales sobre la relación entre, etapas de desarrollo del proyecto, con los niveles de participación registrados.

6.1 Moblin.org (sitio web)

El sitio web de Moblin fue dado de alta en Julio de 2007 para Moblin v1 y, dura vigente hasta Enero de 2009, cuando se libera la versión *alpha* de Moblin v.2, la transición en el sitio web comienza a mediados del año 2008, como preparación para recibir la nueva versión de Moblin. El administrador del sitio es Michael Shaver⁴⁶, el cual, rediseñó el sitio de Moblin.org, en la transición de Moblin v1 a Moblin v2 a mediados del 2009; él fue el encargado de determinar las necesidades de las partes interesadas y de los usuarios. Por medio del análisis de sus tareas, diseñó mapas del sitio, cuadros y prototipos⁴⁷. La arquitectura o base utilizada, para diseñar y construir el sitio web es *Drupal*⁴⁸.

El sitio moblin.org, es conveniente que sea pensado, como la vidriera de contenidos que Scolari (2008:194) señala, es decir, un mostrador al cual, sujetos familiarizados o no con el proyecto Moblin, se acercan para conocer el proyecto, de forma que, por medio de su acceso de solo lectura al sitio web,

46 Ver: <http://moblin.org/users/mshaver> consultado el 22 de Abril de 2010.

47 Ver: <http://www.michaelshaver.com/> consultado el 10 de Abril 2010.

48 Ver: <http://moblin.org/documentation/moblinorg-drupal/drupal-architecture> y <http://drupal.org/> consultados el 22 de Abril de 2010.

puedan aprender lo que es el proyecto, sus objetivos, las instituciones que participan, los elementos que lo componen, las dinámicas de trabajo y, en general, toda aquella información de utilidad para que los visitantes conozcan, a grandes rasgos, al proyecto y, si lo desean, tengan elementos para participar en Moblin v2; es de utilidad señalar que, los flujos de comunicación en el sitio web, es de uno a muchos (unidireccional), con participación restringida, donde el administrador decide, qué entra y qué no.

El sitio web está dividido en seis secciones, cada una, responde a objetivos diferenciados. La primera sección es la "Portada"⁴⁹, está pensada como la primera impresión que los usuarios reciben del proyecto Moblin v2, por ello, resume al proyecto y enlaza a sus principales secciones. La siguiente sección, "Proyectos"⁵⁰, contiene los distintos proyectos que el proyecto Moblin v2 desarrolla, brinda información sobre las características de cada uno, documentación, nombre del o los administradores y, enlaces para que los participantes se suscriban. La sección "Descargas"⁵¹, contiene enlaces para descargar completo o por partes, Moblin v2 o cada uno de sus proyectos, en distintos formatos de empaquetamiento, y también, enlaces para descargar el código fuente de Moblin v2. "Documentación"⁵² es la sección más numerosa del sitio web, contiene manuales y guías sobre Moblin v2 y, de casi todos los proyectos que lo componen, brinda información sobre la arquitectura⁵³, sobre el corazón del proyecto⁵⁴, fotografías sobre la apariencia de Moblin v2⁵⁵, videos⁵⁶, etc. "Comunidad"⁵⁷ contiene los enlaces a los blogs de los proyectos, a las listas de correo, al chat (IRC), a Bugzilla, a eventos y, a perfiles de participantes. Finalmente, "Garage"⁵⁸, contiene los enlaces a los distintos programas, herramientas y aplicaciones, listos para ser utilizados en Moblin v2⁵⁹. Cada una de las secciones que componen al sitio web, puede ser entendida como un índice o mapa, que le sirve al visitante o explorador, para conocer y enlazarse con los distintos actores (sistemas, informacionalismos y sujetos) que componen la red híbrida del proyecto Moblin v2.

Por la observación realizada, aquellos sujetos que quieren participar en el sitio web, considerado en este caso como un proyecto parte de Moblin v2, han de acercarse al administrador y presentar sus propuestas, en base a las cuales, se les brindan o no, los permisos para editar páginas en el sitio web. En

49 Ver: <http://moblin.org> consultado el 12 de Junio de 2010.

50 Ver: <http://moblin.org/projects> consultado el 22 de Abril de 2010.

51 Ver: <http://moblin.org/downloads> y <http://moblin.org/downloads/project> consultados el 22 de Abril de 2010.

52 Ver: <http://moblin.org/documentation> consultado el 22 de Abril de 2010.

53 Ver: <http://moblin.org/documentation/moblin-overview/architecture-overview> consultado el 22 de Abril de 2010.

54 Ver <http://moblin.org/documentation/moblin-overview/moblin-core> consultado el 22 de Abril de 2010.

55 Ver <http://moblin.org/documentation/moblin-overview/netbook-screenshots> consultado el 22 de Abril de 2010.

56 Ver: <http://moblin.org/documentation/moblin-netbook-intro> consultado el 23 de Abril de 2010.

57 Ver: <http://moblin.org/community> consultado el 22 de Abril de 2010.

58 Ver: <http://garage.moblin.org/welcome-moblin-garage> consultado el 22 de Abril 2010.

59 Ver: <http://repo.moblin.org/moblin/garage/2.1/oss/source/> consultado el 22 de Abril 2010.

base al modelo de participantes de Wenger (2001), el participante central es Michael Shaver; los participantes legítimos, son otros miembros formales del equipo de Intel-OTC, con los accesos correspondientes para crear páginas y editar contenidos, podría considerarse como participantes legítimos aquellos sujetos que, externos al OTC, deciden colaborar con Michael; los participantes periféricos, podrían ser considerados a los visitantes, que solo leen y, aquellos visitantes que deciden dejar con comentario⁶⁰. Para más información sobre el sitio web, consultar en los archivo anexos la tabla: moblin.org.ods

6.2 Moblin-dev (lista de correo)

La lista de correo es un sistema que sirve, para que los suscriptores del sistema, intercambien correo electrónico. El programa que el proyecto Moblin utiliza es, **GNU Mailman 2.1.12**, según el manual del mismo, la suscripción sigue una dinámica similar a la suscripción de una revista o periódico; las personas se registran y reciben en su correo electrónico todos los mensajes publicados en la lista, pueden escribir mensajes y los correos que envíen a la lista, llega a todos los suscriptores; este tipo de listas (llamadas de discusión) le dan mayor importancia al primer mensaje de una conversación, el cual, define el tema a tratar y permite, mantener un hilo de discusión; si un suscriptor quiere tratar un tema distinto, se le recomienda que cree una nueva discusión.

El manual de *Mailman* señala que hay cuatro posiciones (tipos de participantes) principales en una lista de correo: administradores de la lista (que se encargan de administrar la lista), moderadores (que leen los mensajes y deciden si deben de ser enviados a los suscriptores), suscriptores o miembros (personas que reciben los mensajes y/o participan en las discusiones) y, administradores del sitio (que se encargan de darle mantenimiento al software).

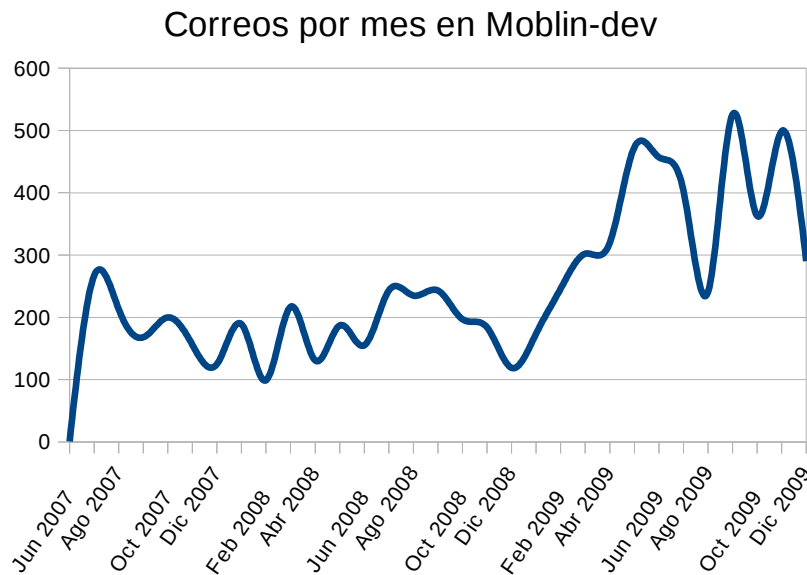
Mailman ofrece dos interfaces de usuario, la primera es una página web donde se brinda información a los usuarios, da la opción para suscribirse, acceder al archivo de las conversaciones y; a los administradores, la posibilidad de ingresar al sistema para realizar cambios. La segunda interfaz, es el manejador de correo de cada usuario, a partir de allí, se puede interactuar con *Mailman*, ya sea para administrar la cuenta, para enviar mensajes a la lista de correo o, para activar opciones de filtrado de mensajes. Con los correos que componen una lista de discusión, *Mailman* genera un archivo, al cual se puede acceder y revisar todos los mensajes intercambiados, con la posibilidad de ordenarlos según fecha, tema, autores y conversaciones.

La lista de correo Moblin-dev comienza su historial en Junio 2007 y hasta Diciembre 2009 hay

⁶⁰ Para más los árboles de las secciones del sitio web, consultar en los anexos la tabla desarrollada en hoja de cálculo.

registrados 7,614 mensajes de 1,110 personas distintas, donde se trataron 2,212 temas distintos, con un promedio de 5.88 participaciones por persona y 3.48 respuestas por tema. En 2007 hubo 1,130 mensajes, en 2008 hubo 2,192 y en 2009 fueron 4,292 mensajes. El participante que más mensajes envió desde 2007 son 229 y los que solo han mandado uno o dos mensajes suman el 56.49% del total. La conversación que más respuestas tuvo fueron 42 mensajes, un 31.28% de los mensajes no tuvieron respuesta y el 21.2% solo tuvo una respuesta.

Los mensajes en la lista de correo han tenido un aumento constante desde que comenzó la lista, el año 2009 fue especialmente activo, en especial a partir del mes de Mayo, esto concuerda con la liberación de Moblin v2.0 en Enero y, con el alojamiento del proyecto en la Fundación Linux en Abril y, la liberación de Moblin v2.1 en Noviembre del mismo año.



Gráfica 1. Elaboración propia con datos de lists.moblin.org/pipermail/dev/ el 22 de Abril de 2010

La lista de correo Moblin-dev es definida, por los propios participantes, como el lugar central para las discusiones de los desarrolladores⁶¹, es un espacio creado para discutir sobre el desarrollo de Moblin, para obtener retroalimentación y atender las inquietudes de los usuarios que, aunque no obtengan respuesta en este espacio, desarrolladores, participantes y otros usuarios mantienen una revisión constante sobre lo que se escribe.

Lo que más interesa resaltar de las posibilidades comunicativas de una lista de correo, es que

61 Ver: <http://moblin.org/community/mailling-lists> consultado el 22 de Abril de 2010.

brinda a cada uno de los suscriptores, la posibilidad de que un solo mensaje llegue a todos los suscriptores, que algunas veces responderán, otras no, sin embargo, mantiene enterados a los suscriptores y, les permite elegir, en qué conservaciones se involucran.

La lista de correo Moblin-dev ha gozado de una actividad que se incrementa año con año especialmente a partir de que la Fundación Linux entra en el proyecto a mediados del 2009 (ver gráfica 1), este movimiento coincidente con el periodo de pruebas de Moblin v2.0 que impulsa la retroalimentación de la comunidad hacia el proyecto. Para los participantes centrales de Moblin, esta alianza con la Fundación Linux resultó benéfica, pues obtuvieron visibilidad entre las comunidades del software libre, lo cual, representó obtener el interés de sujetos con competencias y objetivos que pudieron beneficiar al desarrollo del proyecto, cuenta de ello, es observable a través de mensajes que solicitan ayuda para resolver una problemática determinada y, que en varias de las ocasiones, desembocan en la detección de un error en el código o en la documentación.

Un último detalle que conviene señalar, es que la lista de correos Moblin-dev, no fue utilizado por los participantes centrales para construir acuerdos sobre el diseño de Moblin v2, la revisión apunta a que solo en ciertos casos, las decisiones tomadas por el equipo central de Moblin, fueron puestas a discusión, un ejemplo de ello fue la solicitud de abrir foros o wikis para estimular la interacción de los usuarios⁶², este caso particular derivó en la formación de foros no oficiales por parte de usuarios que no encontraron escucha a sus solicitudes⁶³.

De todo lo anterior, es posible formular tres usos principales en Moblin-dev:

1. El proceso comunicativo se articula al proceso de producción de Moblin v2.
2. Moblin-dev, es un espacio anuncios, que el equipo central de Moblin utiliza para realizar anuncios.
3. Es utilizado por los participantes periféricos y legítimos (no miembros del equipo central), para pedir ayuda, reportar problemas y resolver dudas.
4. El equipo central, recibe retroalimentación del resto de los participantes sobre omisiones, errores y sugerencias, en el código o en la documentación.
5. La lista Moblin-dev no es un espacio de negociación de las decisiones del proyecto.

62 Ver: <http://lists.moblin.org/pipermail/dev/2009-October/006515.html> consultado el 10 de Abril de 2010.

63 Ver los canales no oficiales formados a falta de atención por parte del equipo de Intel: <http://www.Moblincommunity.com> <http://openMoblin.com/forum/> consultados el 10 de Abril de 2010.

Para más información sobre la lista de correo Moblin-dev, consultar en los archivos anexos, la tabla `moblin.dev-list.ods`

6.3 Moblin-Bugzilla

Bugzilla es un sistema⁶⁴ diseñado para rastrear defectos en el software, fue producido y registrado bajo la licencia pública Mozilla-Licence en 1998 por la Fundación Mozilla. Este sistema es utilizado por el proyecto Moblin para registrar, dar seguimiento, encontrar soluciones a defectos y malos funcionamientos en el código; en el caso de Moblin-Bugzilla, también se utiliza como foro para recibir propuestas. La lógica que sigue la anatomía del “bug”, es decir, el defecto específico en el software, según **The Bugzilla Guide - 3.4.5 Release**⁶⁵, comienza por una distinción en la división del trabajo entre administradores y usuarios.

Los administradores se encargan del buen funcionamiento de Bugzilla y asignan a cada bug reportado, un módulo específico, donde uno o más administradores se encarguen de resolverlo. Esta división conduce a que los usuarios, sean los encargados de reportar los *bugs*, para lo cual, se contempla que los reportes deben de contar, con la información suficiente para que los encargados, tengan los elementos necesarios para reproducir el error, estudiarlo y eventualmente solucionarlo, para ello se le recomienda al usuario:

1. identificar el componente que tiene el error,
2. brindar un sumario del problema,
3. asignar palabras clave,
4. indicar la versión del componente del bug,
5. el tipo de error que produce,
6. dependencias de ese componente,
7. mensajes que el sistema brinda cuando se presenta el error y,
8. las instrucciones o pasos que el usuario siguió para poder producir el error.

Junto al sistema de clasificación y la capacidad de anexar información, Bugzilla actúa como un foro en el cual usuarios y administradores establecen conversaciones con motivo del *bug*, así, los administradores incitan a los usuarios a que se les provean con información puntual, para así poder

64 Se toma la definición que ellos mismos hacen sobre su carácter de software libre o código abierto.

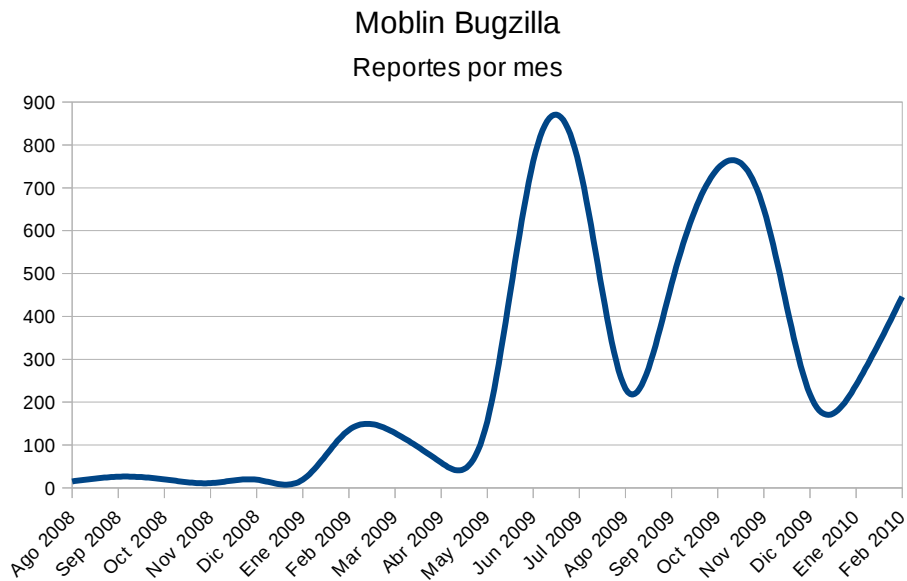
65 Ver: <http://www.bugzilla.org/docs/3.0/html/index.html> consultado el 10 de Abril de 2010.

construir soluciones grupales, donde intervienen procesos de negociación entre los distintos participantes, en especial, en relación con las fechas de lanzamiento de las nuevas versiones.

Desde Agosto 2008 hasta el 28 de Febrero 2010, Moblin-Bugzilla tiene registrados 5,220 *bugs* (aprox.), reportados por 1,332 participantes y, asignados a 138 administradores. Es de resaltar que la distribución de la participación, tanto de reporteros como de administradores, se agrupa en cuentas de correo electrónico de Intel (intel.com y linux.intel.com); dichos reporteros, concentra el 80% de los reportes y, del lado de los encargados de corregir los *bugs*, concentran el 97% .

Hasta el 28 de Febrero 2010, fecha en la cual se levantaron los datos, el estatus de los *bugs* reportados se distribuía de la siguiente manera: nuevos 360; necesita más información 32; asignado 319; esperando implementación 32; reabiertos 39; resueltos 780; verificados 3,186; cerrados 454; total 5,220. En cuanto a su resolución: sin estatus 796, arreglados 1,898, integrados 929, inválidos 309, no arreglados 290, duplicados 778, pospuestos 94, “*Works for me*” (funciona para mí, o, veo que no hay tal error) 126; total 5,220.

Los reportes de bugs a diferencia de la lista de correo, surge hasta Agosto de 2008, casi un año después de arrancado el proyecto de Moblin v1, esto puede relacionarse con que la fase de prueba y corrección de un programa, se realiza hasta que se cuenta con una versión utilizable. En cuanto a la intensidad de la participación, es de resaltar que a partir del mes de Abril de 2009, aumenta para llegar a su máximo en Julio 2009; Abril es una fecha significativa porque en ese mes, en el *Linux Collaboration Summit* se anunció la unión de Moblin con la Fundación Linux y, se presentaron la versión *Alpha* de Moblin v2.



Gráfica 2. Elaboración propia con datos de bugzilla.moblin.org el 22 de Abril de 2010

Bugzilla es un sistema cuya utilidad, radica en las posibilidades que brinda a usuarios y administradores, para organizar y rastrear errores en el software (*bugs*), a diferencia de la lista de correo, en donde se deja bastante libertad en la forma en que deben de ser redactados los mensajes, Bugzilla cuenta con reglas y lineamientos, para que los usuarios al dar de alta los reportes, sean lo más claros posibles y ocupen la menor cantidad de recursos. Para dar de alta un reporte, hay que estar registrado en el sistema, después, se llega a la página de registro de reportes, que pide seleccionar el componente que falla, la versión, severidad, plataforma en la que se produce el error, el sistema operativo, el estado del reporte y, la o las personas a las que podría ser asignado y/o dirigido. Después hay que darle un nombre al error, que según la documentación debe de, en 60 caracteres, identificar de forma rápida y única un error; posteriormente está un sumario en el cual se ingresa información que describa al error, los pasos que hay que seguir para reproducirlo, el resultado esperado y posibles causas. Finalmente está la posibilidad para anexar un archivo.

Una vez levantado el reporte es asignado, verificado y comentado, en búsqueda colaborativa de una solución, que se construye, mediante comentarios en los reportes; esto permite a reporteros y administradores comunicarse, preguntar detalles, hacer sugerencias y, llegar a acuerdos que permitan solucionar el problema, los cuales en algunos casos, vienen de los mismos reporteros.

Un detalle a señalar, es lo sofisticado del sistema de búsqueda de Bugzilla, que permite generar gráficas y tablas de los reportes, según distintos criterios, como fecha, reporteros, severidad, estatus,

resolución, componentes, etc. Tal es la capacidad de Bugzilla para tratar los datos que hizo innecesario darles un tratamiento extra para presentarlos en esta investigación. Para una aproximación a las estadísticas del proyecto, consultar en los archivos anexos, la tabla moblin-bugzilla.ods

6.4 Moblin-Git

Git es un Sistema de Control de Versiones (VCS por sus siglas en Inglés) desarrollado por Linus Torvalds, para el desarrollo del código, del Kernel Linux. Un VCS es un sistema que utilizan los desarrolladores para, administrar su código, dividir el trabajo, administrar los cambios, distribuirlo, unirlo (*merge*) con el trabajo de otros. Git se distingue por:

1. Proveer una plataforma de trabajo descentralizado en el cual los participantes (desarrolladores) no trabajan en una versión central, sino que, *cada uno trabaja en su propia versión* (llamadas clones).
2. En cada clon, los participantes, pueden crear distintas versiones (*ramas*) del proyecto, donde una rama es denominada maestra o cabeza (*head*) y; otras versiones, son llamadas ramas secundarias. El objetivo es tener una rama maestra y ramas secundarias, orientadas a satisfacer necesidades particulares, que llegado el caso, pueden ser unidas con la rama maestra; unidas entre sí, al margen de la rama maestra o; convertirse en la nueva versión maestra.
3. El modelo de clones, ramas, cabezas y uniones, funciona como proceso individual y, social, en el desarrollo del proyecto.

Git permite el intercambio y revisión, descentralizada, del trabajo de la comunidad; esta visión, hay que cuidar no relacionar la descentralización, con la disolución de las estructuras jerárquicas; sino con la recomposición del modelo de control, intercambio y desarrollo del software.

A partir de la revisión de un manual de Git (progit.org 2010), hay seis comandos que resultan de utilidad para caracterizar de forma mínima, las posibilidades que brinda Git, como herramienta de trabajo colaborativo; la descripción se propone sea mínima, porque Git, es un sistema de control, sofisticado y complejo, que ofrece muchas más opciones de las que es conveniente analizar, por ello, se toma la decisión de concentrarse en estos seis comandos, ya que se considera, son suficientes para reflejar, las lógicas que sustentaron el diseño de la plataforma y guían su uso:

1. Clonar (*to clone*)
2. Jalar (*to pull*)
3. Empujar (*to push*)
4. Unir (*to merge*)
5. Ramas (*branch*)
6. Revisar historial (*git-log*)

El primer paso para trabajar con Git, es realizar un clon del repositorio maestro, es decir, de los repositorios oficiales del proyecto. Esto es, realizar una copia del código fuente del proyecto, en la computadora, para trabajar sobre de ella, sin tener que mantener la conexión con el repositorio maestro o central; trabajar con Git, equivale a tener un repositorio principal y, un otras versiones del repositorio principal, pertenecientes a cada uno de los participantes, que han clonado el proyecto; cada participante tiene su propia copia y, realiza los cambios que mejor crea convenientes, sin alterar el repositorio maestro o central, sin necesidad de permisos.

Git no solo permite clonar el repositorio maestro completo, también, permite jalar (*to pull*) determinadas partes, en teoría, de cualquiera de los repositorios existentes; esto equivale, a que los participantes tienen la posibilidad de intercambiar de manera dinámica desarrollos e innovaciones, con el repositorio maestro y/o con otro participante.

Lo contrario de jalar, es empujar (*to push*) los desarrollos, innovaciones o, versiones propias a otros repositorios, sea al maestro o, a un clon; mediante la función empujar, se puede solicitar a un participante o administrador, que reciba el trabajo realizado que, puede o no, ser integrado a su versión o al repositorio maestro.

Esta forma de trabajo contempla que los cambios son libremente realizados sin ataduras y sin compromisos, hasta el momento en el que se construye en conjunto una versión, entonces, se implementan estrategias múltiples de selección, que en teoría recuperan lo mejor de cada una de las distintas versiones.

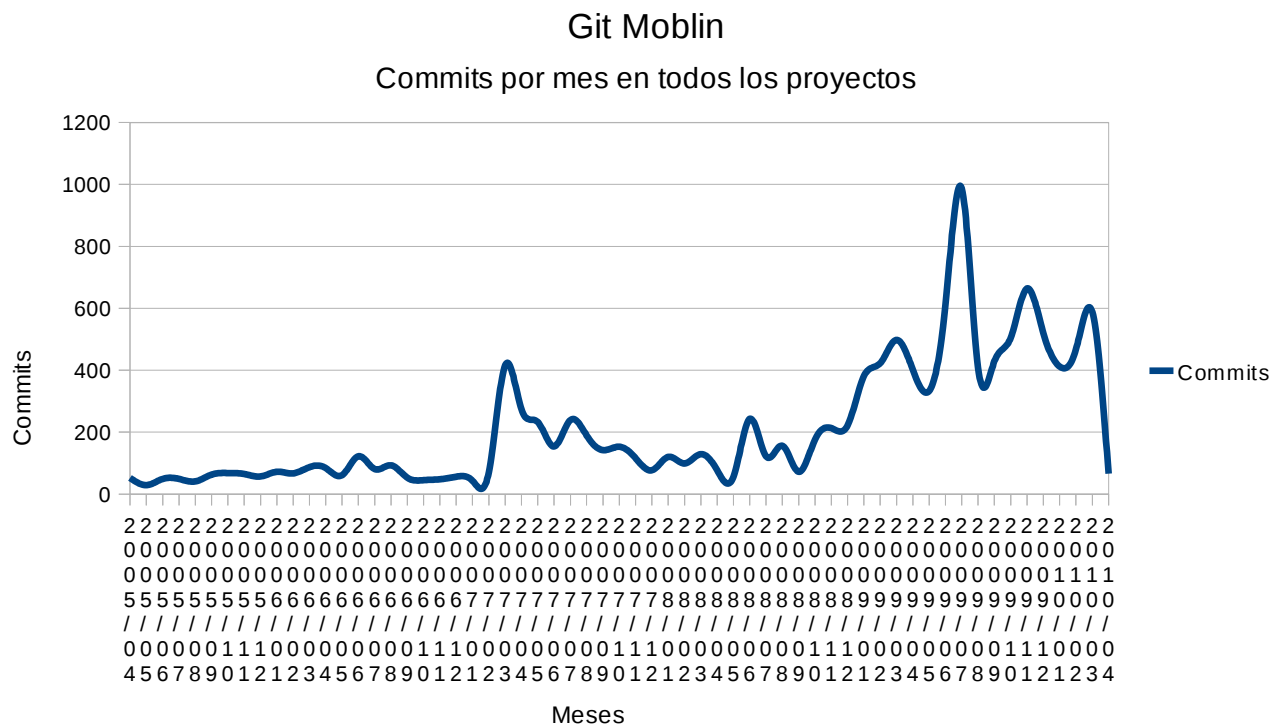
Este proceso de selección es auxiliado por las ramas (*branch*) y las uniones (*merge*). Las ramas son versiones modificadas o, especializadas, que coexisten con la versión maestra o cabeza (*head*); si cumplen los requerimientos pueden unirse (*merge*) con la cabeza (*head*), convertirse en la nueva cabeza, unirse con otras ramas o, desaparecer. Esto implica que no solamente existen alrededor de los repositorios maestros versiones múltiples, sino que, cada versión o repositorio, puede tener dentro de si,

distintas ramas, que coexisten, compiten y se destruyen, con el objetivo de poder tomar lo mejor de cada una, para con ello formar una cabeza (*head*).

Para complementar este proceso, Git produce un historial (*git-log*) de cambios (*commits*), a partir del cual, es posible reconstruir el recorrido seguido por el proyecto. El historial registra los cambios, fechas, descripciones sobre los cambios, autores (los autores del código de los cambios) y *committers* (los que tienen los accesos de escritura, para realizar los cambios). Además, Git puede producir estos historiales en gráficas, de forma que, faciliten su lectura y, control.

Para el levantamiento de los datos en Moblin-Git, se clonó el código de los proyectos que componían a Moblin v2, entre la semana del 28 de Marzo, al 3 de Abril del 2010. Los proyectos que componían al Moblin v2, en el periodo analizado, eran 47 (al mes de Octubre, cuando se redacta este apartado, hay 74 proyectos) y, según el historial (*git-log*) que Git generó, participaron alrededor de 487 autores; el promedio de *committers* por proyecto, llega a casi tres, que en su gran mayoría, participan desde sus cuentas de Intel.

La configuración que toma Moblin, visto desde los proyectos que lo componen, es la de un conglomerado de aplicaciones y librerías. El trabajo de Moblin, comprende ensamblar los distintos proyectos, para lo cual, estudian a cada uno, realizan modificaciones para que, como sistema operativo (unidad), maximicen el ahorro de recursos de *netbooks*, mediante; una interfaz gráfica que aproveche al máximo una pantalla pequeña, ahorre energía y, se prenda y cargue los programas a la mayor velocidad posible.



Gráfica 4: "Git Moblin". Elaboración propia con datos extraídos de git.moblin.org, del 28 de Marzo, al 3 de Abril del 2010.

6.5 Lanzamientos y puntos de encuentro entre las herramientas de comunicación y trabajo.

La participación en la generación de los informacionalismos del proyecto Moblin v2, sigue la lógica de las etapas de lanzamiento de Moblin v2 y, otros factores contextuales, como el éxito comercial de las *netbooks* y, en especial, el alojamiento del proyecto en los servidores de la Fundación Linux.

Los lanzamientos durante el 2009, contemplan 5 versiones de Moblin en su versión 2.0 y tres en su versión 2.1. De Noviembre del 2009 a Febrero de 2010 se realizó trabajo a versiones de 2.1 diseñadas para IVI (*In Vehicle Infotainment*).

Moblin	Versión	Inicio	Fin
2.0	Alpha 1	Enero 2009	17 de Marzo 2009
2.0	Alpha 2	17 de Marzo 2009	21 de Abril 2009
2.0	Alpha 3	21 de Abril 2009	03 de Agosto 2009
2.0	Beta	03 de Agosto 2009	23 de Septiembre 2009
2.0	Final		23 de Septiembre 2009
2.1		23 de Septiembre 2009	03 de Noviembre 2009

2.1	Final		03 de Noviembre 2009
2.1 IVI FC	Final		11 de Febrero 2010

Tabla 7: "Lanzamientos Moblin v2 y 2.1". Elaboración propia con datos extraídos de mobin.org.

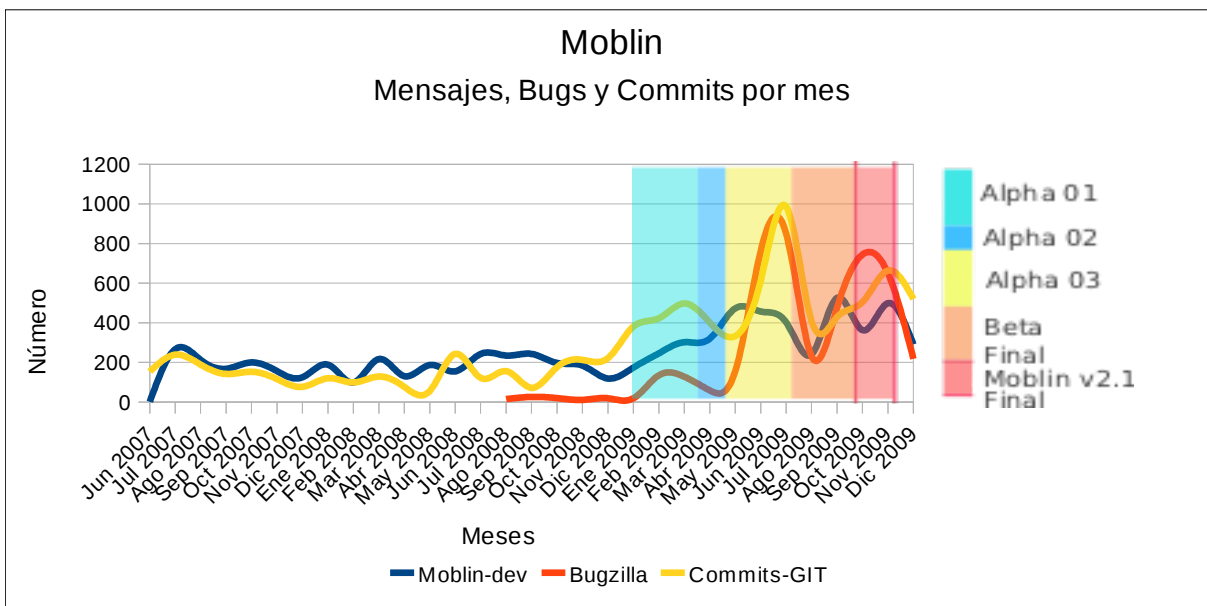
Catalogar una versión como *Alpha*, representa que es una versión utilizable, pero incompleta, que en este caso, pasó por tres etapas antes, antes de ser medianamente estable; a este punto de "mediana estabilidad", se le llama *Beta* y, contempla, que es una versión de pruebas considerada relativamente estable, pero aún incompleta. Cuando la versión *Beta*, se encuentra lo suficientemente madura, se le llama "estable"; las versiones estables, son aquellas que han sido probadas y mejoradas, lo cual, no excluye que aún tengan errores y deficiencias, sin embargo, son mínimos; también se les llama estables a versiones viejas. Las versiones *Beta Final*, pueden ser representadas como versiones estables, con la aclaración de que, no se seguirá trabajando en ellas. Para aclarar este último punto es de utilidad recordar un poco sobre el proyecto Moblin.

Moblin es un proyecto de Intel, que comenzó en el 2007, su objetivo fue desarrollar un sistema operativo GNU & Linux, para promocionar la familia de procesadores Intel-Atom; Moblin v1 fue diseñado para MID's, sin embargo, el proyecto no brindó los resultados esperados, aparentemente debido al poco interés del mercado por los MID's, que se hizo más notorio, frente al éxito de las *Netbooks*.

Moblin v2, fue desarrollado durante el 2009, para aprovechar el interés por las *Netbooks* que usan procesadores Intel; su propuesta, contempló, maximizar el espacio en pantallas pequeñas; ahorrar energía y; *bootear* (cargar el sistema) rápidamente (en 5 segundos). Moblin al finalizar el año 2009, se une con el proyecto *Maemo* de Nokia; para formar *Meego*; su propuesta es impactar en el mercado de los *Smartphones*, que podría decirse, son un tipo de MID's con teléfono integrado; según palabras del coordinador de *Maemo* y de *Meego* (por Nokia), Ari Jaaksi (2010), *Meego* es un sistema operativo de computadora completo, diseñado para MID's. Moblin al unirse con *Meego*, regresó a una versión de desarrollo *Alpha*.

En este contexto, muchos son los elementos que intervinieron en el desarrollo de Moblin v2, en este caso, interesa resaltar el impacto que tuvo en el desarrollo la Fundación Linux (Linux). Por un lado, Moblin al unirse con Linux, se convirtió en un proyecto visible para la comunidad de Linux. Por el otro lado; adoptar las prácticas de trabajo y sistemas de información (herramientas) administrados por Linux, brindó confianza entre la comunidad del software libre e, interés por Moblin. Esto se reflejó en un crecimiento en el desarrollo del proyecto.

La gráfica 3, ilustra el incremento en la participación a partir de Abril del 2009, cuando se anunció en el *Annual Collaboration Summit* la colaboración entre Linux y Moblin:



Gráfica 3: "Desarrollo de Moblin". Realización propia con datos extraídos de Moblin-dev; Moblin-Bugzilla y; Moblin-Git.

Es ilustrativo señalar que, cada vez que se señala una nueva fase de desarrollo, se registra un pico en la actividad, de Moblin-dev, Bugzilla y Git; al cual, una vez terminada registra un descenso en la actividad. Resulta de importancia señalar que, la relación más estrecha entre los sistemas, se registra entre Bugzilla y Git, ya que, en Bugzilla se formulan los acuerdos para corregir el código y, en Git, se implementan dichos acuerdos.

La lista de correo Moblin-dev, jugó un papel distinto al esperado; en las primeras fases de esta investigación, se llegó a considerar que la lista de correo era un espacio de negociación, donde se discuten y negocian acuerdos, que afectan el desarrollo de un proyecto; similar a las *Python Enhancement Proposals* (PEP's), descritas por Sack et.al. (2006) o, las *Request For Comments* (RFC).

Se observó que Moblin-dev es utilizado como un canal de ayuda, que los participantes centrales utilizan, para ayudar a los participantes periféricos, a satisfacer las necesidades que tienen con Moblin; por ejemplo, ayudarles a adaptarlo a un hardware específico, o, asistirlos para usar y adaptar una aplicación en Moblin. El giro que se propone realizar en este punto, es dejar de lado la comunicación que mantuvieron los participantes centrales entre sí y, enfocar, la interacción entre participantes centrales y participantes periféricos, donde la ayuda de los primeros a los segundos, será abordada

como una característica de los proyectos de software libre.

7 Segunda etapa. Conversaciones.

Esta sección da cuenta de los hallazgos del análisis de contenido de las conversaciones mantenidas, en la lista de correos Moblin-dev, durante los meses de Mayo a Diciembre del 2009. Las unidades de significado que se privilegian son las conversaciones, con esto se busca que las preguntas sean dirigidas, a las construcciones grupales de sentido, que se formulan a través de la interacción de los participantes.

Estos hallazgos y reflexiones, serán presentados en base a tres ejes: sujetos; forma y; contenido. Este orden busca, definir una tipología general sobre los sujetos; la estructura interna de los mensajes, en base a reglas y prácticas de uso del correo electrónico; los principales temas tratados en las discusiones, y la forma en que la lista de correos se vincula con el desarrollo de Moblin y con procesos de construcción de sentido en las conversaciones.

7.1 Sujetos

Hay tres elementos a partir de los cuales se construye la diferenciación entre los participantes; el primero, es la posición que juegan en la conversación, es decir, si son los que escriben el primer mensaje (el mensaje que desata la conversación) o, si son los que responden al primer mensaje. El segundo, es la dirección de correo electrónico desde la cual escriben. El tercero, son es un criterio doble, el primero, se sitúa en la relación desarrolladores principales; desarrolladores de segundo grado y, vendedores y; usuarios finales. El segundo, de las comunidades de práctica: participantes centrales; participantes legítimos y; participantes periféricos. Con el primero se busca formular la diferencia en el uso de Moblin; en el segundo, adscripción del sujeto a la comunidad.

Cabe hacer la aclaración de que la formulación que se presenta está incompleta, porque la tercer etapa metodológica no fue realizada, sin embargo, existen los suficientes elementos como para construir una aproximación confiable sobre la posición que ocupan los sujetos, en los procesos de comunicación, que tuvieron lugar, en el desarrollo de Moblin v2.

Equipo central

Solo algunos de los miembros del equipo central participan en la lista Moblin-dev, entre los que registran mayor participación, se encuentran: AjV (linux.intel.com); KA (intel.com); WJJ (intel.com); RL (intel.com); RW (intel.com); RB (burtonini.com); EB (linux.intel.com).

Este grupo central, en su mayoría escribe desde cuentas de Intel; AjV y EB, escriben desde cuentas de "linux.intel"; al igual que RB, que en la lista de correos escribe desde su sitio personal, "burtonini.com" y, en Bugzilla, escribe desde "linux.intel".

El papel que juegan éstos miembros del equipo central en la lista de correo, es la de dar asistencia a quién la solicite, solo ellos, concentran poco más del 35% de los mensajes, que responden, preguntas y solicitudes de ayuda. Aunque no se cuentan aún con elementos para profundizar el análisis, resulta preciso subrayar, que el modelo de trabajo del software libre, contempla que, sujetos con posiciones centrales en el proyecto, dedican parte de su tiempo a responder preguntas simples, complejas, tontas o, brillantes, de desconocidos⁶⁶.

Vendedores, desarrolladores de segundo nivel y participación legítima.

Existe un nivel intermedio entre el equipo central de desarrolladores y, los usuarios finales de Moblin v2; en este nivel, podrían señalarse tres posiciones no excluyentes o exclusivas. Éstas figuras, pueden ser definidas por los términos, de vendedor; de desarrollador de segundo nivel y; de participante legítimo. La primera hace referencia a un intermediario comercial que utiliza Moblin v2 para satisfacer la necesidad de un cliente.

Un desarrollador de segundo nivel, de manera similar, toma a Moblin v2, experimenta, lo adapta, transforma y modifica, pero, esta actividad no se encuentra necesariamente ligada a un objetivo comercial, puede ser por gusto o diversión que lo hace. Otra acotación, es que los desarrolladores de segundo nivel, se orienten al desarrollo de aplicaciones y, su objetivo, es que sus aplicaciones, puedan ser utilizadas también en Moblin v2, con la consideración de que, si esas aplicaciones son parte fundamental de un producto final que se ofrece a un cliente, sería correcto ubicar al sujeto como desarrollador de segundo nivel y como vendedor; en cambio, si la aplicación desarrollada, es multiplataformas y Moblin v2 es una de tantas, es correcto nombrarlo como desarrollador de segundo nivel, porque desarrolla una aplicación para Moblin v2, aunque no distribuye un producto final basado en Moblin v2.

Según la nomenclatura de las comunidades de práctica de Wenger (2002), la participación

66 El acceso que los participantes periféricos tienen para participar en Moblin-dev, es abierto, no solo técnicamente, sino organizacionalmente, ya que, no importa el participante, la cuenta de correo o su identidad, sino, su pregunta; la cual, si es clara y brinda la información suficiente, es respondida. Para los otros sistemas de información, las redes de confianza que implementan los participantes centrales se manifiestan en, acceso de escritura restringido solo a participantes legítimos, los periféricos, tienen que desplazarse hacia el centro para obtenerlo.

sostenida en una comunidad, lleva a un cambio identitario de los participantes dentro de la comunidad de práctica; su objetivación no necesariamente lleva consigo un cambio en los niveles de acceso, sino, algún tipo de reconocimiento por parte de la comunidad; a los sujetos que reciben este reconocimiento, se les llama, participantes legítimos.

En Moblin v2, hay tres elementos que caracterizan a los participantes legítimos; el primero, es que en determinado punto, estos participantes decidieron ayudar al equipo central, en la resolución de dudas y solicitudes de ayuda de otros participantes, en la lista de correo Moblin-dev; el segundo, que las direcciones desde las que escriben, en la mayor parte de las veces, son de empresas orientadas al desarrollo de soluciones tecnológicas y; en algunos casos, dichas cuentas de correo, son de distribuciones GNU & Linux, como *Suse*, *Mandriva* y *Xandros*.

La legitimidad brindada a los participantes legítimos, sin tocar el tema de accesos en los sistemas, proviene de los participantes periféricos, que logran resolver preguntas y dudas gracias a su ayuda, esto puede entenderse; como un desplazamiento de éstos participantes, de la periferia hacia el centro, construido sobre la base de su aprendizaje de Moblin v2, lo suficientemente consistente, como para ayudar a otros participantes, menos experimentados, a satisfacer sus necesidades y, realizar nuevos aprendizajes.

Antes de continuar, cabe hacer la aclaración de que, no serán tratados los sujetos con sus nombres, sino, con sus direcciones de correo electrónico, de esta forma, se evita; hacer referencia a datos personales y, posibilita visibilizar a las instituciones participantes. Existen dos problemas con esta forma de proceder, el primero es; que muchos mensajes, escritos desde cuentas de empresas relacionadas con el ramo tecnológico, siguen intereses personales y no institucionales, de forma que, es complicado saber en qué momentos es un proyecto comercial y en qué otros, un proyecto personal. Una solución parcial a este problema, es investigar el tipo de productos o servicios que ofrece la empresa, desde la cual se emiten los mensajes; si se encuentra relacionada con MID's o IVI's, es probable que los mensajes obedezcan objetivos comerciales, sino, no.

El segundo problema son las cuentas de correo de gmail, yahoo y hotmail; que, aunque brindan una mayor certeza sobre los objetivos personales que persiguen los participantes, oscurecen si sus objetivos personales, son de usuario final, desarrollador o de vendedor independiente. La forma de reducir esta incertidumbre es, por la información que dejan en los mensajes, sin posibilidad de triangularla.

Vendedores

Dentro del grupo de los participantes que escriben el primer mensaje, se encuentran aquellas empresas especializadas en MID's, IVI's y otros sistemas de info-entretenimiento que utilizan la familia de procesadores Atom, éstas empresas son: 126.com, appliedminds.com, hildebrand.co.uk, imgtec.com, innes.fr, intel.com, mobica.com, open-plug.com, redflag-linux.com, speaktech.com, wipro.com. Este grupo realiza preguntas relacionadas con formas de configurar Moblin v2 para distintos tipos de hardware.

Dentro del grupo que escriben en respuesta al primer mensaje, se encuentran: harman.com, videon-central.com, mobica.com, collabora.co.uk, fusiongarage.com, sigarden.com. Este grupo, a diferencia del anterior, en vez de realizar preguntas, las responde; sin embargo su participación se limita a uno o dos mensajes.

Desarrolladores de segundo nivel

En el grupo de los desarrolladores de segundo nivel hay un fenómeno particular, los objetivos de estos participantes dan cuenta de un mayor interés personal, comercial o no comercial, para participar en la lista de correo. Esto puede interpretarse como un interés por adaptar sus aplicaciones en Moblin v2, como proyectos independientes de empresas; en este caso, el número de cuentas de correo relacionadas con empresas prácticamente no figuran; en su lugar, aparecen cuentas de gmail, yahoo, hotmail y sitios web personales.

Por lo observado en las discusiones, el desarrollo de aplicaciones en Moblin v2, se encuentra enmarcado en el ecosistema de sistemas operativos orientados a MID's, como *Android* de *Google*; *Symbian* y *Maemo* de *Nokia*; *iOS* (iphone OS) de *Mac*; entre otros. Lo anterior, puede entenderse como una actividad no dedicada a Moblin v2, sino, una actividad que se orienta a distintas los distintos sistemas operativos de los MID's, a partir de los cuales, se articula la actividad de los participantes. En este sentido, relaciones de compromiso o participación sostenida, se ven disminuidas de forma considerable, ya que, una vez que la aplicación funciona en un sistema operativo, la necesidad es satisfecha y el participante deja de serlo hasta que una nueva necesidad aparezca. A excepción de aquellos casos, en los intereses de los usuarios trascienden el mero desarrollo de aplicaciones y, se articulan con el uso de Moblin v2, como usuarios finales.

El tema que más trata este grupo de participantes, es el relacionado con la virtualización de Moblin v2 en plataformas de desarrollo como *Ubuntu*, *Fedora* y *Windows xp*. La virtualización implica que, sobre un sistema operativo, se crea ejecuta máquina virtual como *VMware*, *KVM-Qemu* o, *Virtual Machine*; sobre la cual, otro sistema operativo es ejecutado. Del proceso de virtualizar Moblin v2,

puede entenderse que, los desarrolladores utilizan un sistema operativo, sobre los cuales desarrollan aplicaciones y, por medio de máquinas virtuales, prueban sus aplicaciones en los sistemas operativos para los cuales las adaptan.

El proyecto Moblin v2, hace accesible las herramientas necesarias para crear a Moblin v2 (*Moblin Image Creator*) y, para adaptar (*Moblin Package Creator*) las aplicaciones a Moblin v2. Esto puede interpretarse, como una forma de promover a Moblin v2 en el mercado, de forma que, mientras más aplicaciones puedan ser utilizadas, más posibilidades hay de que usuarios finales se interesen en Moblin v2, como el sistema operativo de su vida diaria.

Participantes legítimos

La legitimidad proviene de dos fuentes y, como tal, da paso a dos tipos de participantes legítimos; la primera legitimidad es de participantes que representan a proyectos de software libre, con los cuales el proyecto Moblin ha establecido contratos; estos participantes son: AW de *OpenSuse*; GKH de *Suse*; MM y AB de *Novell*; KW de *Xandros* y; OB de *Mandriva*. Estos participantes, representan a otras distribuciones GNU & Linux, que consideran que Moblin v2 es una propuesta que puede serles útil de alguna forma; los participantes legítimos, trabajan con Moblin para beneficiar a la distribución a la que pertenecen y, los avances, desarrollos o innovaciones que formulen para su distribución, los comparten con Moblin v2.

El segundo tipo de participantes legítimos, son aquellos que han mantenido una participación constante y, que no cuentan con un respaldo institucional, sino que, solo colaboran con el proyecto. En este caso, esta colaboración o, participación orientada a ayudar al proyecto es, ayuda a quienes solicitan ayuda para usar o configurar Moblin v2 y; retroalimentación a Moblin v2, por medio de reportes de errores y participación en las ocasionales discusiones suscitadas por, modificaciones al código de Moblin o, a la interfaz gráfica.

El papel de ambos tipos de participantes periféricos, apunta al desarrollo de una red de ayuda que complementa, la labor de los participantes centrales; los participantes que se considera la conforman, sin contar a los participantes legítimos arriba mencionados y a partir de la revisión realizada, son: DLF de *fridu.net*; JC de *linuxlighthouse.com*; GG de *lincor.com* y; EW de *gmail.com*.

Usuarios finales y participantes periféricos

Los usuarios finales son aquellos que utilizan Moblin v2, como el sistema operativo de su *netbook*; ésto se manifiesta en la lista de correo, como una relación del sujeto con Moblin v2, el sistema operativo y,

Moblin, el proyecto. Esta relación, si la consideramos una especie de contrato entre usuarios y desarrolladores, da cuenta de un proceso de aprendizaje del sujeto con el sistema, de forma que, si algo se complica, el sujeto recurre a los canales de ayuda del proyecto Moblin. En este sentido, los usuarios finales, se encuentran cercanos, tanto con los participantes legítimos, como con los participantes periféricos, ya que, pocos son los sujetos que usan distribuciones GNU & Linux, en especial, porque requiere de conocimientos particulares que en no pocas ocasiones, solo se obtienen en canales de ayuda como Moblin-dev.

En cuanto al abandono de la participación de los usuarios finales, de manera similar a la de los desarrolladores de segundo nivel, recurren a los canales de ayuda de Moblin v2 para satisfacer una necesidad transitoria, que no genera compromiso y que una vez resuelta, permite distanciarse. Esta participación por tal, se caracteriza por su corta temporalidad, sea porque el usuario satisface su necesidad y abandona Moblin, o porque no puede solucionarla, se da por vencido y, abandona Moblin.

7.2 Estructura interna de los mensajes

La interacción de los participantes en una lista de correo, se logra a partir de mensajes intercambiados por el correo electrónico, en este apartado, se describen las reglas que siguen los participantes en sus mensajes, de forma que aporten elementos para abordar las discusiones.

Los correos intercambiados prácticamente solo constan de texto, si los participantes lo consideran necesario, se anexan imágenes o archivos. En segundo lugar, la economía del lenguaje es una pieza clave para caracterizar sus mensajes; los mensajes son claros y van directo al grano sin rodeos, sin formalidades, pero con respeto. En tercer lugar, existen algunas reglas que se siguen para iniciar discusiones y, para participar en las discusiones.

Reglas explícitas

La primera regla, es que antes de crear una nueva conversación, hay que buscar si el tema ha sido tratado con anterioridad, en caso de que así sea, se escribe en el tema y no se crea uno nuevo, por ejemplo, este mensaje escrito por KW de Xandros:

This is such a dumb question that I was sure **it must have been asked before but I didn't find it in the mailing list**. How does one eject a usb key? I couldn't even manually umount it because fuser claims that bkl-orbiter and metacity are using it.⁶⁷

⁶⁷ Esta es una pregunta muy tonta y **estoy seguro de que fue preguntada antes, pero no encontré nada en la lista de**

Cabe mencionar que este ejercicio de revisar lo realizado antes de crear una nueva discusión, puede relacionarse con una práctica de desarrollo de software, que busca evitar hacer *forks* de los proyectos, es decir, en lugar de dividir los esfuerzos de la comunidad del software libre, en el desarrollo de distintos proyectos similares, es mejor concentrar esfuerzos el menor número de proyectos. Esta práctica, al menos en la revisión realizada, no se practica con frecuencia y, es común que los mismos temas sean preguntados una y otra vez, en especial, los relacionados con los drivers de las tarjetas de video y la compatibilidad con los procesadores.

La segunda regla a seguir, es que cada mensaje en cada discusión, debe de seguir el tema del primer mensaje, sino, debe de crear una nueva discusión; a su vez, cada nueva discusión, debe de orientarse a tratar el tema general de la lista de discusión, en caso de que no lo haga, es indispensable señalar que no lo hace con el encabezado en el título "*Off Topic*"⁶⁸ (OT). Si es spam o un mensaje completamente fuera de tópico, el participante corre el riesgo de perder su suscripción.

La tercera regla es el respeto a los suscriptores, el lenguaje duro y directo es permitido (y promovido), sin embargo, insultos innecesarios o agresiones fuera de lugar están terminantemente prohibidos, los participantes que incurren en estas faltas son eliminados. En los mensajes analizados, llegan a observarse tensiones, pero en ningún punto agresiones o insultos.

Prácticas (reglas implícitas)

Una primera práctica promovida en la redacción de los mensajes, es utilizar títulos (*subjects*) que den cuenta del tema tratado, de forma que, los lectores, de manera sencilla y rápida, puedan representar la necesidad que se expresa en el mensaje. Esta práctica, no es practicada lo suficiente en Moblin-dev.

Una segunda práctica, es exponer con claridad y brindar la información suficiente, para que los lectores sepan cuál es la necesidad que busca ser satisfecha; en principio, la redacción y ortografía, no parece ser decisiva en la evolución de las conversaciones, pero, sí se toman en cuenta. En cuanto a la suficiencia en la información, de las 145 conversaciones analizadas, en 12, se solicita mayor información, de forma que, en casi un 10% de los casos, se invirtió tiempo y energía, para solicitar que preguntas y problemas fueran clarificados.

Un ejemplo de las solicitudes de más información, es el mensaje enviado en la conversación

correo. Cómo puedo expulsar una llave usb? Ni siquiera puedo hacerlo manualmente, porque fuser dice queblk-orbiter y metacity la están usando.

68 Fuera de tópico.

"connection manager fail in moblin V2 beta":

What kind of connman fail you met? Please provide more info.

1. Does connmand daemon be launched?
2. When you click connman icon, what you see?⁶⁹

La forma más común de definir problemas (necesidades) es por medio de los *logs*, los cuales, son mensajes que ofrece el sistema sobre un dispositivo, aplicación o problema particular; los *logs*, sirven como información estandarizada y reproducible, que sirve como base para definir con precisión el tema tratado. Por ejemplo, aquí un caso en donde se hace una solicitud de información, en donde se pide que el sujeto introduzca un comando en la terminal y mande por correo los resultados:

Please help do below command under terminal

```
>>>Dbus-send --system --print-repl --dest=org.moblin.connman /
org.moblin.connman.Manager.GetProperties
See what's the output?70
```

Un extracto del *log* en esta conversación es:

output for version 15:

```
[mm at localhost ~]$ dbus-send --system --print-reply
--dest=org.moblin.connman / org.moblin.connman.Manager.GetProperties
method return sender=:1.9 -> dest=:1.87 reply_serial=2
array [
  dict entry(
    string "ActiveProfile"
    variant object path "/profile/default"
  )
]
```

69 Qué clase de error en connman encontraste? por favor brinda más información

1. Es lanzado el demonio connmand?
2. Cuando das click en el icono de connman, qué vez?

70 Por favor ayuda, ejecuta el siguiente comando en la terminal

```
>>>Dbus-send --system --print-repl --dest=org.moblin.connman / org.moblin.connman.Manager.GetProperties
Vez cuál es el output?
```

```
dict entry(  
  string "Profiles"  
  variant      array [  
    object path "/profile/default"  
  ]  
)
```

Una tercera práctica promovida, es la forma *inter-posting* para responder a los mensajes, el cual, responde a un código heredado de Usenet, para formular maneras más eficientes de responder a los mensajes: la *Netiquette*; que según manuales consultados⁷¹, pertenece a una de cuatro formas de responder a un correo electrónico:

- *Clean-posting*: borrar mensaje original y responder.
- *Top-posting*: responder sobre el mensaje original.
- *Bottom-posting*: responder abajo del mensaje original.
- *Inter-posting*: responder punto por punto al mensaje original.

Pudo observarse que los sujetos más experimentados, como participantes centrales y legítimos, utilizaron más que otros grupos, la forma de *inter-posting*. Esta forma de respuesta, permitió discusiones puntuales, claras y, con mayor profundidad que con las otras formas de respuestas. La forma del *inter-posting*, consiste en conservar los elementos a los cuales se responde y, eliminar el resto del o los mensajes. Un ejemplo de la forma de estos mensajes, es tomado de la discusión "*separate partition trouble*":

DM at gmail.com

Thu Nov 12 07:55:22 CST 2009

2009/11/12 WJJ <wjj at intel.com>

> You mean partition of your SD card is broken even take it on another
> machine?

Yes, exactly.

> Could you provide the HW info of your platform, better is lspci output?

71 Ver: <http://charlescurley.com/netiquette.html>, <http://www.catb.org/~esr/jargon/html/index.html>,

<http://www.caliburn.nl/topposting.html> y, <http://www.faqs.org/rfcs/rfc1855.html> consultadas el 20 de Octubre de 2010.

this is the lspci:
00:00.0 Host bridge: Intel Corporation Mobile 945GME Express Memory
Controller Hub (rev 03)
00:02.0 VGA compatible controller: Intel Corporation Mobile 945GME Express
Integrated Graphics Controller (rev 03)
00:02.1 Display controller: Intel Corporation Mobile 945GM/GMS/GME,
943/940GML Express Integrated Graphics Controller (rev 03) ...⁷²

En este caso, se discute sobre un problema con una tarjeta SD; el participante DM, responde a preguntas y solicitudes de *logs* que hace el participante WJJ; el participante DM, según los manuales de Netiquette, conserva el texto suficiente para dar contexto a las respuestas y responde abajo de los mismos. Si se contempla esta respuesta desde la economía del lenguaje, se ahorran palabras y se alimenta el buen entendimiento, al conservar el contexto del mensaje al cual se responde.

El código para entender qué mensajes son los anteriores, los da por *default* los clientes de correo electrónico; el símbolo utilizado es el "mayor que" (>)⁷³, el cual, representa un mensaje anterior; ">>" son dos mensajes anteriores, ">>>" tres mensajes anteriores y así sucesivamente; si no hay ">", significa que es el mensaje más actual. Cabe mencionar que arriba de los mensajes, el sistema señala quién es el participante que escribe y cuándo mandó el mensaje.

72 DM at gmail.com
Jueves Nov 12 07:55:22 CST 2009
2009/11/12 WJJ <wj at intel.com>
> Dices que la partición de tu tarjeta SD está rota, aún si la usas en otra
> máquina?
Sí, exactamente.
> Podrías proveer la información del HW (hardware) de tu plataforma, mejor si es el *output*
> de lspci?
este es el lspci:
00:00.0 Host bridge: Intel Corporation Mobile 945GME Express Memory
Controller Hub (rev 03)
00:02.0 VGA compatible controller: Intel Corporation Mobile 945GME Express
Integrated Graphics Controller (rev 03)
00:02.1 Display controller: Intel Corporation Mobile 945GM/GMS/GME,
943/940GML Express Integrated Graphics Controller (rev 03) ...

73 No confundir con la codificación utilizada en la metodología para ubicar los flujos de conversación.

Una cuarta práctica promovida, es la lógica de las respuestas, que recupera la lógica para crear nuevas discusiones y, nuevos proyectos, es decir; la de no repetir lo ya dicho. En la mayoría de las discusiones, un participante lanza una pregunta o un problema a la lista y recibe, si la información que da es eficiente, soluciones, consejos, enlaces, comandos y en general, distintos tipos de informacionalismos, que le permitan satisfacer su necesidad; en este sentido, es común que más de un participante responda, lo interesante consiste en que, cada uno de estos participantes, se toma la molestia de leer las posibles soluciones ya brindadas, para, de manera ideal, solo participar si su solución es una alternativa sólida para dar solución a la necesidad planteada en la acción inicial.

La práctica de investigar un poco por lo ya existente, para no hacer dos veces el mismo trabajo, permite que las respuestas que se brindan en Moblin-dev, logren soluciones, sino individuales, grupales, formuladas a partir de la atención a la participación de otros sujetos y, a la búsqueda de complementar y mejorar lo ya existente; esto involucra, que si la solución ya está dada, no tiene caso aportar nuevos elementos. En teoría esta práctica debería de ser aplicada por todos los participantes, sin embargo, no es así y, en algunos casos es común encontrar mensajes que lanzan soluciones alternativas, después de formulada una solución aceptable, dichos mensajes son, en el mejor de los casos agradecidos e ignorados; en la mayoría de las ocasiones, solo ignorados.

7.3 Conversaciones y desarrollo de Moblin v2

El objetivo de la discusión en este apartado, es formular una interpretación sobre las interacciones que los sujetos establecen en Moblin-dev; se parte del supuesto de que, existe una relación indirecta entre la lista de correo, con el desarrollo de Moblin v2, la cual; es explorada a partir de la revisión de los temas tratados, en relación con las fechas de lanzamientos de Moblin v2.

El recorrido que se propone comienza con una descripción de los temas tratados, la función social de los sujetos en los procesos de comunicación y, cierra con; una exploración de las construcciones de sentido producto de la interacción de los sujetos, desde la perspectiva de los procesos de interpretación de la información intercambiada y; los flujos de comunicación, preguntas-respuestas-diálogo-negociación.

Temas tratados

Al primer mensaje se envía se le llamará de aquí en adelante, acción inicial; a las respuestas que obtiene la acción inicial, se les llamará solo respuestas. La preeminencia de la acción inicial se le brinda en base a, que sitúa el tema a tratar a lo largo de la conversación, de forma que, se constituye

como un eje de articulación alrededor del cual, las respuestas (si las hay) se acomodan.

El segundo criterio de clasificación, es la forma expresiva de los mensajes, si son preguntas, solicitudes de ayuda, anuncios, quejas o sugerencias. La distribución apunta a, que la lista de correo fue utilizada como un espacio en el cual se le brinda ayuda a quienes lo solicitan y, en menor medida, se realizan reportes, sugerencias y quejas.

Forma expresiva	No.	Frecuencia
Pregunta	104	72.22%
Reporte	21	14.58%
Anuncio	8	5.56%
Solicitud ayuda	7	4.86%
Sugerencia	3	2.08%
Queja	1	0.69%
Total	144	100.00%

Tabla 8: "Formas expresivas". Elaboración propia con datos de Moblin-dev (mayo-diciembre 2009) el 20 de octubre de 2010.

Sobre la interpretación de los mensajes, las acciones iniciales, apuntan a una preeminencia de estos participantes por, adaptar, modificar y usar Moblin v2. Este proceso, se encuentra caracterizado por preguntas relacionadas a las formas para personalizar Moblin, a partir de la modificación o configuración de alguno de sus elementos.

Una segunda tendencia detectada, es la necesidad de integrar aplicaciones y controladores que no vienen instalados en la versión original. En este sentido, los usuarios se encargan de tomar sus aplicaciones favoritas, generalmente el código fuente de las mismas, compilarlas e instalarlas. Este proceso, al requerir de ciertas habilidades y conocimiento sobre Moblin, es facilitado en la lista de correo y, se complementa con manuales, guías y la ayuda de los participantes de Moblin-dev.

Un punto intermedio entre usar e integrar, es dado por la integración de controladores y codecs. Por ejemplo, Moblin v2 no contiene los *codecs* para tocar mp3, porque no son libres, así que hay que conseguirlos de otros proyectos e, instalarlos, un caso similar sucede con películas en *dvd's*. Esta situación se extiende a tarjetas *wireless*, tarjetas de video y procesadores; por tal, para poder usar un hardware particular, es necesario integrar controladores y *codecs*. A continuación, la tabla sobre las acciones buscadas:

Acción buscada	No.	Frecuencia
Usar	33	22.917%
Configurar	26	18.056%
Arreglar	19	13.194%
Integrar	19	13.194%
Desarrollar	19	13.194%
Modificar	10	6.944%
Conocer	9	6.250%
Lanzar	2	1.389%
Compilar	1	0.694%
Promocionar	1	0.694%
Soportar	1	0.694%
Investigar	1	0.694%
Contribuir	1	0.694%
Felicitar	1	0.694%
OT	1	0.694%
Total	144	100%

Tabla 9. Elaboración propia con datos de Moblin-dev (mayo-diciembre 2009), consultada el 20 de octubre de 2010.

Es de resaltar que, en la tabla de las acciones buscadas, solo se registra en una acción, el objetivo explícito de contribuir al proyecto, esta contribución, la hizo un participante legítimo de *Mandriva*. Su contribución es un parche (código) para mejorar el funcionamiento del navegador web de Moblin; esta contribución se distingue de las demás, porque es el único mensaje que contiene código para ser integrado. Esto no excluye que haya otras formas de contribución, por ejemplo, el reporte de errores, que alcanza poco más de un 13% de las acciones buscadas, en la tabla de las formas discursivas se le llaman reportes, en la de las acciones buscadas, se le llaman "arreglar".

Entrando de lleno a los temas tratados, se utilizan 32 categorías para agruparlos, la que mayor concentración tiene es un 12.5% y las menores, alcanzan el 0.694% de la muestra. Los principales temas tratados, giran alrededor de la adaptación de aplicaciones a Moblin; la interfaz gráfica; los *codecs*, de audio, video, de tarjetas *wireless* y tarjetas de video; desarrollo y virtualización de imágenes; entre las más importantes.

Es importante señalar, que aquellas conversaciones que tocan el tema de "*GMA500*" y "Procesador", dan cuenta de un problema importante, el de las restricciones al acceso a los *codecs* de la

tarjeta de video *GMA500* y, las limitaciones que los diseñadores imprimieron en Moblin, para que solo trabaje con procesadores Intel que soportan el set de instrucciones *Supplemental Streaming SIMD Extensions 3*. Esta acción, excluye de Moblin a buena cantidad de procesadores Intel y, a todos los procesadores AMD, su mayor competencia.

Por un lado, no liberar los *codecs* de la tarjeta *GMA500*, generó molestias entre los participantes, que adquirieron *netbooks* con procesadores Intel, con la intención de poder usar en ellas distribuciones GNU & Linux, objetivo logrado con distribuciones como *Ubuntu* y *Fedora*; pero no con Moblin v2, proyecto promovido por Intel.

Por el lado de los procesadores, el problema es que solo soportar aquellos que cuentan con *SSSE3*, excluye a todos aquellos desarrolladores de segundo nivel, cuyas estaciones de trabajo, cuentan con procesadores no soportados, lo cual limita, el desarrollo de aplicaciones para Moblin. Sobre usos comerciales de Moblin, se encuentran aquellos mensajes que tratan a los temas de *IVI*; consumo de energía; teléfono y; *copyright*.

Temas y desarrollo de Moblin

A través de los meses de análisis (Mayo a Diciembre de 2009), las acciones buscadas y los temas generales presentan cambios; estos cambios son relacionados en este apartado con las fases de desarrollo y lanzamientos de Moblin v2.

El análisis de las conversaciones comienza en el mes de Mayo, cuando la versión Alpha 3 de Moblin lleva un mes de haber sido lanzada, este periodo de tiempo registra la mayor actividad en Bugzilla y en Git; lo cual puede pensarse, como un periodo de pruebas, correcciones e implementaciones en preparación al lanzamiento de Moblin v2.0 *Beta* y Moblin v2.0 *Final*. Las acciones buscadas más mencionadas, fue en primer lugar "desarrollar" y en segundo "usar"; esto puede interpretarse que, por un lado, mientras la actividad del equipo central se enfoca en corrección de errores y desarrollo; por un lado, los desarrolladores de segundo nivel, presentan necesidades relacionadas al ajuste de sus estaciones de trabajo (en especial compatibilidad con los procesadores), uso de las herramientas que Moblin brinda para el desarrollo (como *MIC2*) e, interfaz gráfica. Por el otro lado, los usuarios finales, aprender y usar es la principal acción buscada y, la interfaz gráfica el tema más tratado; estos participantes, junto con los desarrolladores de segundo nivel, se unen para hacer sugerencias y expresar sinsentidos, malentendidos y dudas sobre la forma en que ha de utilizarse y, desarrollarse, para una nueva interfaz que representa, una nueva forma de entender la pantalla de las *netbooks*.

En el mes de Junio, los participantes centrales alcanzan el punto más alto en reportes de errores en Bugzilla, mientras los niveles de implementación en Git aumentan. Para los desarrolladores de segundo nivel y los usuarios, la preocupación por la incompatibilidad de los procesadores sigue siendo el tema más mencionado, sin embargo, las acciones buscadas apuntan a usos más que a desarrollo, junto con un aumento en el interés por reportar errores y; conocer como integrar *codecs* para audio y video, el proyecto *fastboot* (el encargado de modificar los distintos elementos que componen Moblin para que arranque más rápido) y, formas de utilizar dispositivos periféricos relacionados con *touchscreen*.

En Julio el equipo central alcanza el tope en desarrollo de Moblin y, los reportes comienzan a disminuir. La actividad de los desarrolladores de segundo nivel se ve disminuida; en cambio, los usuarios finales comienzan a ser más insistentes, sobre todo, en lo relacionado con *codecs* de audio y video y, controladores para tarjetas *wireless*.

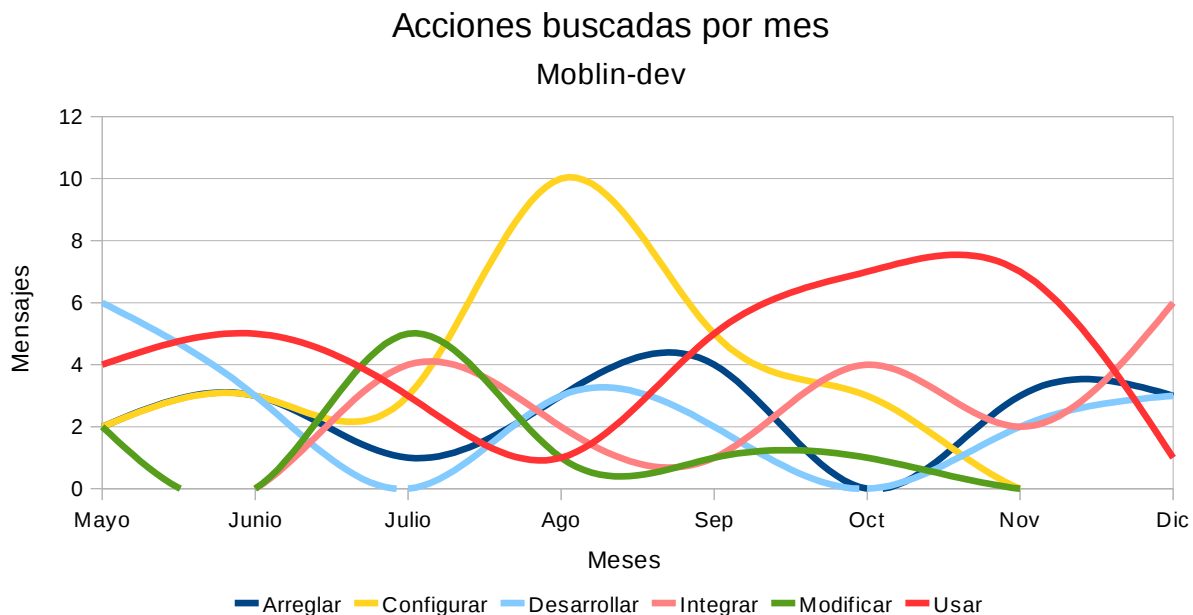
En Agosto la fase de desarrollo Moblin v2.0 *Alpha 3* llega a su fin y se libera la versión *Beta* de Moblin v2.0; esto se refleja en la disminución de la participación en implementación y corrección de errores, de los participantes centrales. En el ámbito de los participantes periféricos, los desarrolladores de segundo nivel se activan un poco, después de que en Julio no participaran; los usuarios casi no participan y; los vendedores aparecen en el escenario. La principal acción buscada es la configuración de *Mutter* y *Clutter*, las aplicaciones encargadas de controlar y desarrollar la interfaz gráfica; cambiar colores, logos, distribución de ventanas o integrar efectos.

En Septiembre llega a su fin la fase de desarrollo *Beta* de Moblin v2.0, se lanza la versión final de Moblin v2.0 y, se da inicio al desarrollo de Moblin v2.1. Los grupos de los participantes periféricos que se mantienen más activos son los vendedores y usuarios; los cuales, buscan acciones centradas en uso, reparación, virtualización y configuración de Moblin. Los temas tratados siguen relacionados con la interfaz de usuario, *codecs* para audio y video y, problemas relacionados con la tarjeta *wireless*.

En el mes de Octubre, se registra un tope de reportes de errores en Bugzilla menor al mes de Junio, seguido con un aumento en la actividad del equipo central en Git; esto concuerda con una disminución de la participación en Moblin-dev. El mes de Octubre parece especialmente inactivo para los desarrolladores de segundo grado, para los usuarios y vendedores, el uso y la implementación de nuevas aplicaciones, son las acciones buscadas que reciben más atención; al igual que el mes anterior, las conversaciones giran en torno a *codecs* de audio y video, la interfaz gráfica y, la (in)compatibilidad de Moblin con los procesadores de los participantes periféricos.

En el mes de Noviembre el levantamiento de errores en Bugzilla disminuye y la producción en Git alcanza un tope menor que el registrado en Julio, la versión 2.1 de Moblin sigue en desarrollo. Lo más relevante en este mes, es que los vendedores y usuarios comienzan a tomar el papel de "conocedores" de Moblin y a ayudar a nuevos participantes periféricos a satisfacer necesidades, sin desplazar al equipo central. Las las principales acciones buscadas por los usuarios periféricos fueron usar y arreglar. Los objetos sobre los cuales se discutió más, fueron *codecs* para audio y video, (in)compatibilidad de Moblin con procesadores y; comienza a discutirse más sobre IVI's. Un detalle importante fue que los vendedores y usuarios que comenzaban a desplazarse hacia el centro, por medio de su asistencia a participantes más periféricos, fue que, felicitaron al equipo de Moblin por la calidad y cantidad de correcciones y mejoras que hicieron en Moblin v2.1.

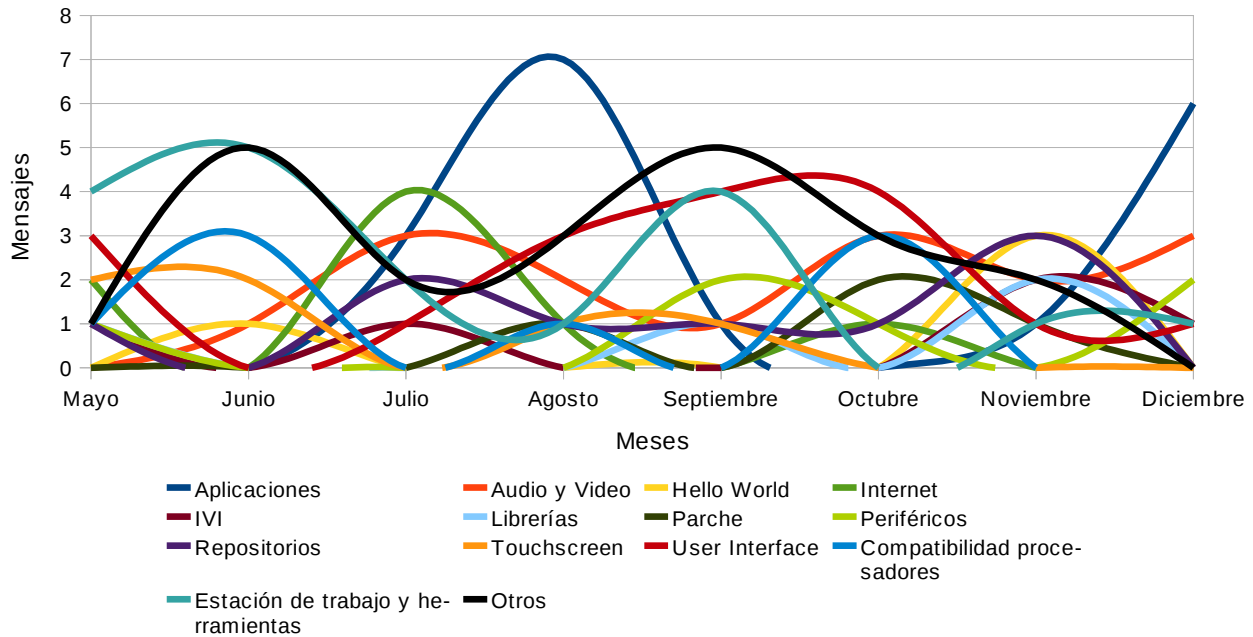
En Diciembre los niveles de participación del equipo central disminuyen en Bugzilla, Git y, en Moblin-dev. En este mes se termina el desarrollo de Moblin v2.1 y se libera su versión final. Los vendedores y usuarios con participación mantenida toman un papel más activo en la ayuda a nuevos participantes, los cuales, se acercan al nuevo lanzamiento de Moblin y, que tienen complicaciones para integrar nuevas aplicaciones a Moblin; instalar *codecs* para audio y video y; usar dispositivos periféricos.



Gráfica 4 "Acción buscada por mes". Elaboración propia con datos extraídos de list.moblin.org

Temas discutidos por mes

Moblin-dev



Gráfica 5 "Temas discutidos por mes". Elaboración propia con datos extraídos de list.moblin.org

Interacción y construcción de sentido

Las conversaciones en Moblin se encuentran abiertas a todos los participantes, se respetan pocas reglas explícitas y otras implícitas se siguen, para lograr, mantener un espacio de apoyo que permita a los participantes satisfacer necesidades relacionadas con el uso, desarrollo de aplicación y adaptación de Moblin, con fines comerciales y personales.

En el proceso de la construcción de soluciones, se fundamenta en la conversación escrita, en forma de mensajes que viajan por el protocolo del correo electrónico. La estructura de las conversaciones más común, está formada por una acción inicial y respuestas de parte de la lista de correo; la acción inicial tiene un papel preponderante, ya que es la que determina el tema y, el participante que la escribe tiene la responsabilidad de expresar con claridad su problema para que otros puedan entenderlo y proponer soluciones.

Etapa 1: Forma expresiva de la acción inicial

Se definieron seis formas expresivas de las acciones iniciales (preguntas, reportes, anuncios, aportaciones, sugerencias y fuera de tópico), sin embargo, preguntas y reportes agrupan poco más del 90% del total. Las preguntas son recursos que utilizan, en su mayoría participantes periféricos, para

solicitar ayuda en Moblin-dev; los reportes, son utilizados para informar que hay un error u omisión en Moblin o, alguno de sus aplicaciones.

De forma general, las acciones iniciales desatan las discusiones en Moblin-dev, el Diagrama 1 ilustra el proceso por el cual transita una pregunta. Por lo observado, el primer impedimento para la solución de las necesidades de los participantes que hacen la pregunta, es la claridad con la cual la expresan, solo hay soluciones cuando se expresan con la claridad suficiente y, aportan la información necesaria, con frecuencia les regresaban sus preguntas para que fueran reformuladas o, anexaran mayor información.

El segundo impedimento para las soluciones, era el tránsito entre la teoría y la práctica, el cual exige de los participantes que hacían las preguntas, tuvieran conocimiento de como implementar las soluciones recibidas, en casos en donde no tenían estas habilidades, fue común observar que se solicitara más información para poder implementarlas.

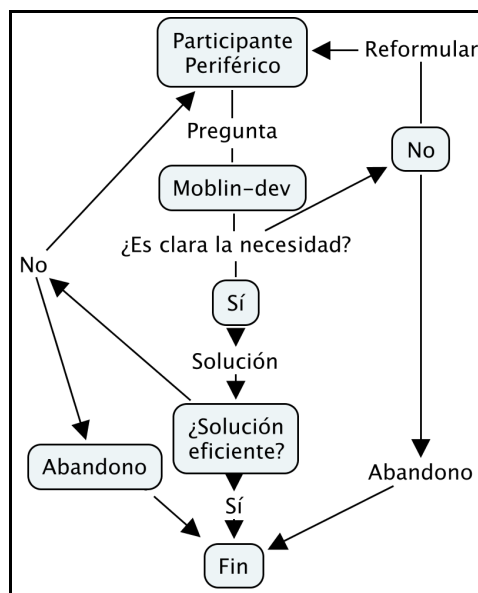


Diagrama 1: "Flujo de conversación en exposición de necesidades con forma expresiva de pregunta".

En el caso de los reportes la discusión se centró en la mayoría de las veces en, la veracidad del problema, en los casos en que el problema era comprobado, se le solicitaba al participante que reportara el error en Bugzilla; sino, se formulaban soluciones o se abandonaba la discusión; en ambos casos, se presentaban argumentaciones para evaluar la veracidad del reportes. El Diagrama 2 muestra en rasgos

generales el flujo.

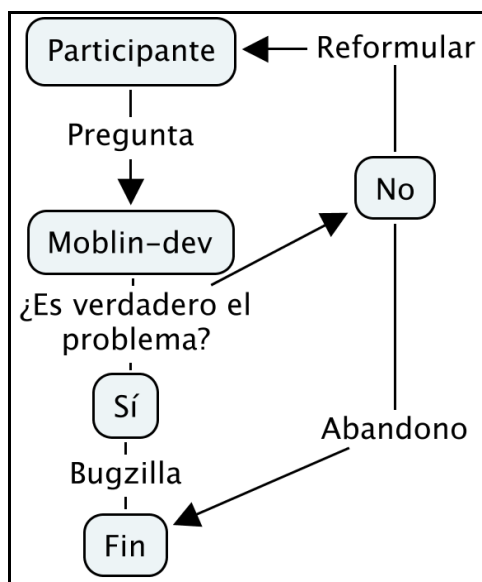


Diagrama 2: "Flujo de conversación en exposición de reportes".

Nombre	Clave	No. de conversaciones que lo usan
Pregunta	1P	99
Reporte	1R	33
Anuncio	1A	8
Aportación	1Ap	1
Sugerencia	1S	2
OT (fuera de tópico)	1OT	1

Tabla 9. "Formas expresivas". Elaboración propia con datos de Moblin-dev.

Etapa 2: Intercambios de información

A partir de respuestas y contra-respuestas, se forman conversaciones a partir del intercambio de información y de la prueba de la veracidad y efectividad de las afirmaciones. En todos los casos que tienen respuesta, existen intercambio de información, sin embargo, esta información se constituye en su mayoría de intercambios de nombre de aplicaciones, o de instrucciones que no involucran comandos y; solo en alrededor de un cuarto de los casos, pueden observarse intercambios de enlaces y de comandos.

Nombre	Clave	No. de conversaciones que lo usan
Intercambio de información	2II	123

Intercambio de enlaces	2Ie	19
Intercambio de Documentos	2Id	0
Intercambio de código	2Ic	27
Solicitudes de documentación	2Sd	0

Tabla 10. "Intercambios de información".

Elaboración propia con datos de Moblin-dev.

Etapa 3: Argumentación

Casi dos terceras partes de las conversaciones logran trascender el intercambio simple de información y, llegar a un nivel de argumentación y de negociación de las soluciones a las necesidades de los participantes. El primer elemento que resalta, es la clarificación de la información, que aparece en casi un cuarto del total de conversaciones que llegan al nivel de argumentación; en una proporción poco menor, las argumentaciones son un elemento que caracteriza a las conversaciones, en ellas, se brindan argumentos con base racional, que buscan la definición de los problemas y de las soluciones, sin llegar a la construcción colaborativa de soluciones. En casi un tercio de las conversaciones, las conversaciones encuentran una solución sin necesidad de la intervención de mayor complejidad en la interacción de los participantes.

Nombre	Clave	No. de conversaciones que lo usan
Solicitud de clarificación	3Cl	25
Argumentación	3Ar	20
Contra-Argumentación	3Ca	21
Uso argumentativo de evidencias	3Ev	9
Soluciones	3Csol	32

Tabla 11. "Argumentación". Elaboración propia con datos de Moblin-dev

Etapa 4: Construcciones grupales

En una proporción similar a un cuarto de las conversaciones se logra llegar a la etapa de las construcciones grupales. En solo ocho casos se formulan soluciones grupales a las necesidades de los participantes, en el resto de los casos que cuentan con una solución, es un sujeto el que propone una solución efectiva, esto sucede porque, en general las respuestas que ofrecen soluciones, se muestran como opciones de las cuales el autor de la acción inicial elige; en estos casos, no puede hablarse de una construcción grupal de soluciones, sino, de intercambios de soluciones.

Los casos más comunes de construcciones grupales en las conversaciones, son las afirmaciones

de posturas, las cuales, son consideradas una autoreflexión que los participantes formulan a partir de las discusiones; las posturas son tomadas como una construcción individual que busca situar al participante frente a un grupo, es decir, un proceso a partir del cual el participante como sujeto, toma una posición frente a otros actores.

Un ejemplo de las tomas de posturas, es la conversación "*Broadcom Driver for Dell Mini 9 users*", la cual presenta una discusión sobre las implicaciones que tiene liberar un driver que combina el Kernel Linux con software para las tarjetas *wireless* Broadcom. La problemática surge cuando Linux está registrado bajo la GNU-GPL v2 y, exige que todo trabajo derivado sea acompañado de su código fuente, en este caso, no todo el controlador de la tarjeta Broadcom es código fuente, por tal, hay una violación de la GPL.

La forma en que los dos participantes (participante legítimo y participante periférico) conversan, los hace reflexionar sobre el valor de las licencias, la importancia de respetarlas y, reflexionar las consecuencias que tiene, tanto individual como colectivamente, el que Broadcom no libere todo el código fuente de los controladores para sus tarjetas. Este proceso reflexivo, del participante periférico, que es instruido por el participante legítimo, busca ser representado como una afirmación de postura.

Sobre la construcción de acuerdos, solo en siete se considera que se formulan, en buena medida, porque Moblin-dev no es un espacio de negociación, sino de ayuda, sin embargo, podría decirse que en un 4.86% lo es. Un ejemplo para abordar la formulación de acuerdos, es la conversación "*mutter-moblin / ccss speedup ...*", en la cual, el participante legítimo MM comparte un desarrollo (parche) y un razonamiento que sirve para reducir el tiempo que Moblin tarda en encender. En este caso, a raíz de este anuncio, otros participantes legítimos y centrales intercambian razonamientos e interpretaciones sobre el trabajo que cada uno realiza por su cuenta. Se le llama un acuerdo, aunque no haya plan de trabajo o calendario, porque los participantes aportan ideas y propuestas útiles que les permiten obtener retroalimentación para realizar su trabajo y, saber qué es lo que cada uno está haciendo, en dónde y cómo.

La única síntesis de no acuerdo, fue formulada en la discusión "*where is the system tray icon ?*" referente a la interfaz gráfica; en este caso, se intercambiaron perspectivas, se brindaron argumentos técnicos, se formularon problemáticas y se brindaron posibles soluciones, a un problema que no estaba a discusión, es decir; distintos participantes coincidieron en que la interfaz gráfica había olvidado un espacio donde situar el ícono (*small systray icon*) de un programa, cuando éste se encuentra activo,

pero el usuario no quiere verlo, por ejemplo, el programa Skype, que puede ser minimizado y no ocupa espacio en la barra de tareas, como un programa abierto, ni en la pantalla, sino que, se minimiza en un ícono que se aloja en la "bandeja del sistema" y, desde allí el usuario, si lo desea, lo invoca. La problemática surge cuando la interfaz de usuario de Moblin no cuenta con una bandeja del sistema, al menos no de la misma manera como otras interfaces lo hacen.

En este caso, primero se fundamentó la necesidad de una bandeja del sistema convencional, segundo; que existen ciertas soluciones que pueden implementarse, pero que no son convenientes; tercero, que esto no es tarea del equipo de Moblin, sino de los desarrolladores que hacen los programas y; cuarto, se definieron las aplicaciones, programas, librerías y otros recursos implicados en la solución del problema, de forma que, los participantes interesados puedan desarrollar una solución que sirva.

La síntesis de no acuerdo, es el tercer punto, que esta necesidad no corresponde ser solucionada por Moblin, para fundamentar este punto hubo que hacer una recapitulación de los principales argumentos que apuntaban a que así fuera. En este caso, la síntesis de no acuerdos también sirvió como un contra-argumento formulado a partir de una síntesis de los argumentos refutados.

Las síntesis de discusiones solo son formuladas en cuatro ocasiones y, pueden considerarse como una forma de Netiquette, utilizada como una atención al destinatario de un mensaje, para que esté enterado de lo que ha sido discutido y pueda participar en la conversación. Un ejemplo ilustrativo, es la conversación "*low power state of atom*", en la cual, participantes periféricos "L_U" y "YL_U" desean medir el consumo del procesador Atom en sus distintas fases de funcionamiento, en especial, cuando se encuentra en modo de suspensión. La síntesis de la discusión es realizada por YL_U, que, después de que L_U había desesperado a los participantes centrales AZ y AjV; participa con una recapituación de la discusión precisa y extensa, que actúa como una especie de tributo para la lista y, que provoca nuevas respuestas amables y efectivas.

La conversación "*low power state of atom*" también sirve para ilustrar los cambios de enfoque de los sujetos, los cuales, representan cambios de perspectivas o, modos de hacer las cosas que no ofrecen buenos resultados, por ejemplo, L_U buscaba inducir el modo de suspensión de manera al procesador y, AZ y AjV una y otra vez le argumentaron que eso no era recomendable, ya que el Kernel es el único encargado tomar esa decisión, que lo más conveniente, era esperar a que el Kernel indujera la suspensión. Al final de la conversación y con ayuda de la síntesis de YL_U, los participantes centrales hicieron que L_U y YL_U, cambiaran de opinión.

El cambio de enfoque de Moblin, es utilizado para indicar que Moblin, como proyecto decide

realizar un cambio o, desarrollar una aplicación por sugerencia de un participante periférico o legítimo. Este cambio en la perspectiva de Moblin, es expresada por uno de los participantes más centrales de los participantes centrales (Administradores con cuentas de correo linux.intel.com), como por ejemplo, por Bob Spencer, que a partir de la discusión "*where's the shutdown button?*", que trata sobre la ausencia de un botón de apagado en la interfaz gráfica. Bob Spencer, al ver esta discusión, participa para reafirmar la necesidad de, si no un botón, al menos de un cuadro de diálogo que aparezca cuando se oprime el botón de encendido y apagado de la *netbook*. Este caso también sirve para ilustrar el poder discursivo de los administradores.

Nombre	Clave	No. de conversaciones que lo usan
Soluciones grupales	4Sg	8
Afirmaciones posturas	4Ap	16
Construcción acuerdos	4Ca	7
Síntesis de no acuerdos	4Sna	1
Síntesis de discusiones	4Sd	4
Cambio de enfoque sujeto	4CeS	5
Cambio de enfoque de Moblin	4CeM	2

Tabla 12. "Construcciones grupales". Elaboración propia con datos de Moblin-dev

Etapa 5: Cierres

Todas las conversaciones tienen un cierre, casi un 40% de los mismos, son cierres simples, solo la conversación se interrumpe y deja poco espacio para la interpretación. Poco más del 20%, son cierres en donde el participante que realizó la acción inicial implementa las soluciones brindadas por otros participantes en Moblin-dev, obtiene resultados positivos y en muchos de los casos agradece la ayuda.

Cuando la forma expresiva de la acción inicial es un reporte tiene al menos tres desenlaces, el reporte es falso, el error es comprobado y, el error ya fue reportado y se encuentra en desarrollo. En los casos en que el error fue comprobado y reconocido como reporte; en general el mensaje de cierre es escrito por un participante central como: "podrías dar de alta este error en Bugzilla?", solo en ocho ocasiones sucede esto. Por su parte, en casi un 14% los errores que se reportan están siendo solucionados.

Para cerrar, el abandono del sujeto representa una interrupción en los mensajes del sujeto que inicia una conversación, esto ocurre en veinte ocasiones. En otras siete, pasa lo contrario, que los participantes de la lista no responden a los mensajes de un participante, esto llega a suceder cuando se

ha dado respuesta o solución a la acción inicial y, el participante no quiere entenderlo y sigue preguntando lo mismo; también, cuando este participante espera que le solucionen el problema sin esforzarse.

Nombre	Clave	No. de conversaciones que lo usan
Desenlace	5	57
Aplica el Sujeto	5AS	31
Bug	5B	8
En construcción Moblin	5CM	20
Abandono Sujeto	5AbS	20
Abandono Lista	5AbL	7

Tabla 13. "Cierres". Elaboración propia con datos de Moblin-dev.

Flujos de conversación

El tema del poder al pensar a Moblin-dev como un canal de ayuda, es observable a través del flujo de conocimiento; en este caso, de los participantes que lo poseen a aquellos que no lo poseen; en términos más precisos, de los participantes centrales a participantes periféricos.

En esta investigación antes se recupera de Mahendran (2002) tres formas de poder: material, técnico y discursivo. Se coincide con Mahendran en la determinación que el poder material tiene en la configuración de las relaciones entre participantes en la red de sistemas de información de los proyectos de software libre, sin embargo, su forma de abordar al poder técnico y discursivo, al centrarse en formas de negociar el conflicto (poder discursivo) en discusiones acaloradas (llamadas *flames*, que podrían entenderse como peleas), oculta la importancia que tiene el conocimiento (poder técnico) como poder.

Moblin-dev al ser un espacio centrado en la ayuda entre participantes, administrado por la Fundación Linux y financiado por Intel, no da pie a *flames*, o al menos, no hay evidencia de que las discusiones deriven en peleas. Entonces, el poder se muestra en dos formas principales, la primera, en el poder discursivo de los miembros del equipo central, en especial los más importantes (Administradores) y; segundo, en el poder técnico, el cual en rasgos amplios, refleja autonomía relativa frente a las otras formas de poder.

El poder técnico, pensado como conocimiento, fluye en poco más del 75% de los casos, de los participantes centrales hacia los participantes periféricos; se espera que esto suceda por ser Moblin-dev

un canal de ayuda, esta situación abona la suposición de que los participantes periféricos mantienen relaciones de poder desiguales con los participantes centrales, desde la perspectiva del poder material, expresada en los sistemas de información; desde la del poder discursivo y; también, desde la del poder técnico, en este caso representada en el Diagrama 3 en color morado.

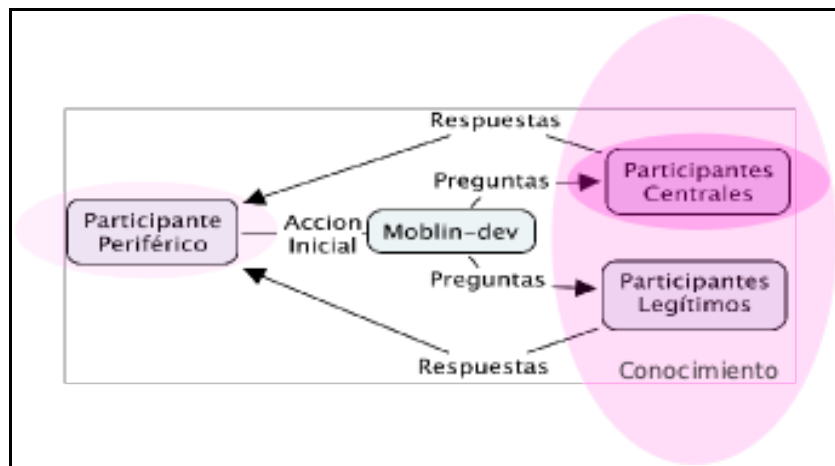


Diagrama 3: "Flujo del conocimiento cuando la acción inicial es una pregunta".

El desequilibrio en la relación de poder entre participantes centrales y participantes periféricos es nivelada por medio de, la disposición de los primero a ayudar a los segundos, bajo la condición de que los segundos expongan con claridad sus necesidades o problemas. Sin embargo, esta forma de proceder, por sí sola, no empodera a los participantes periféricos, en cambio los mantiene en un nivel de perifericalidad con respecto al equipo central; lo cual puede interpretarse como funcionalidad de Moblin-dev, como canal de ayuda, pero, no como espacio de socialización para el cultivo de una comunidad⁷⁴.

El cultivo de comunidad desde la perspectiva de Bacon (2009), requiere de elementos para que los participantes periféricos socialicen, se involucren, y desarrollen compromiso con la empresa de la comunidad. Uno de los medios que señala de utilidad para cultivar una comunidad, es la meritocracia. El fundamento de la meritocracia, como forma de gobierno, implica que el poder sea asignado en base al mérito por sobre intereses empresariales, personales, políticos o económicos, sin embargo, en Moblin-dev, no se presentan evidencias de una distribución basada en el mérito, sino, en base a

⁷⁴ Para conocer la veracidad de esta afirmación se planteaban las entrevistas, sin embargo, en este punto habrán que quedar de esta manera, bajo la consideración de que han de ser revisadas en un futuro con mayor detenimiento.

compromisos empresariales, como se muestra en el Diagrama 4, donde el equipo central en sí mismo los procesos de toma de decisiones, dejando de lado a otros participantes, en especial, a los participantes periféricos.

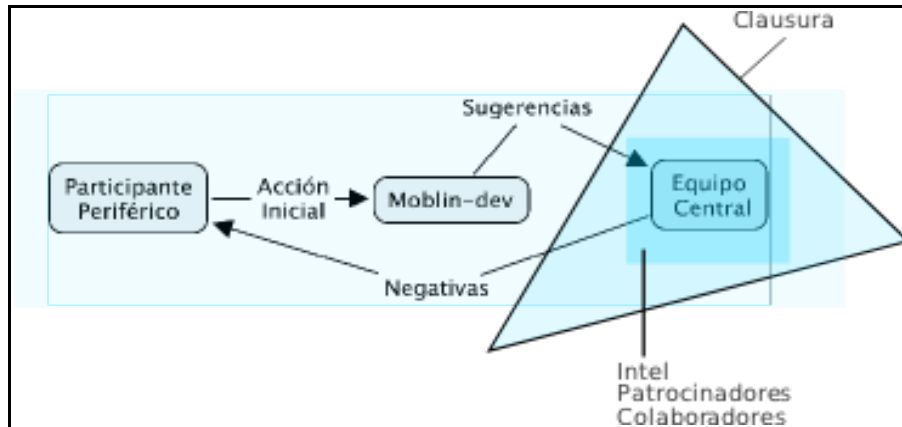


Diagrama 4: "Relaciones de poder entre participantes periféricos y equipo central, cuando la acción inicial es una sugerencia".

Lo anterior sirve para interpretar a Moblin como un proyecto altamente centralizado que no promueve o, promueve de forma periférica, el desplazamiento de los participantes periféricos hacia posiciones centrales en la comunidad. Esto se refleja en la concentración de acceso en el equipo central; en su poca disposición a cambiar sus posiciones sobre el desarrollo del proyecto en favor de los deseos de los participantes periféricos y legítimos y; la preeminencia discursiva, en Moblin-dev, de los participantes centrales sobre el resto y de los Administradores sobre los participantes centrales.

Una de las formas observadas de la cual los participantes periféricos y legítimos utilizaron, para empoderarse mediante la participación, fue brindar ayuda a otros participantes en la resolución de problemas, esta forma de proceder, al menos les brindaba elementos para participar en la distribución del poder discursivo en Moblin-dev.

7.4 A manera de cierre

La concentración del poder en el equipo central no es un caso aislado en los proyectos de software libre, Mokus (2005) en su investigación con los proyectos Apache y Mozilla, observó que el grupo central concentró, alrededor del 80% del trabajo; Sack et.al. (2006) plantea que una de las funciones del proyecto Python, pensado como una red híbrida, es la de conservar su forma por medio de la

concentración del poder en Guido van Rossum (*Python's BDFL*⁷⁵) y, en el grupo Python-dev.

Esta forma de concentración del poder, es mitigada en el caso de Python con las PEP's (*Python Enhancement Proposals*)⁷⁶, un canal de comunicación por medio del cual los participantes hacen llegar sus propuestas e inquietudes al equipo central de Python (Sack et.al. 2006). En Moblin el canal utilizado para las sugerencias es Bugzilla, sin embargo, por exceder los límites del análisis de contenido, no fue analizado⁷⁷.

8 Conclusiones generales

Las conclusiones son divididas en dos apartados, el primero, busca responder la pregunta de investigación, a partir de la descripción de las características que se consideran más importantes, de los procesos de comunicación que tuvieron lugar en el desarrollo de Moblin v2. El segundo apartado, contiene una explicación de los pendientes que quedan en la investigación, en concreto, la realización de una tercera etapa metodológica.

8.1 Procesos de comunicación en el desarrollo de Moblin v2.

La pregunta de investigación planteada es:

¿Cómo se articulan, en los procesos de comunicación que tienen lugar, en el proceso productivo del proyecto Moblin v2: las herramientas de comunicación y de trabajo; los modelos de trabajo; y las interpretaciones de las necesidades subyacentes (Habermas 2001) de los participantes?

Los procesos de comunicación que tienen lugar en el desarrollo del proyecto Moblin, se caracterizan por llevarse a cabo en (ser soportados por) una red de sistemas de información en línea, a partir de los cuales los sujetos trabajan e interactúan. Estas herramientas (sitio web, lista de correo, IRC, Transifex, Bugzilla y Git) se especializan en un proceso productivo distinto, en palabras de Sack et.al. (2006), discutir (planeación del proyecto, brindar soporte, negociación de sentido), documentar (informar, desarrollar manuales y guías, fundamentar decisiones) e, implementar (programar, desarrollar, integrar, realizar los acuerdos).

75 Benevolent Dictator For Life del proyecto Python. Para más información ver: <http://neopythonic.blogspot.com/> consultado el 15 de Noviembre del 2010.

76 Ver: <http://www.python.org/dev/peps/> consultado el 15 de Noviembre del 2010.

77 En la revisión de Moblin-Bugzilla no se fijó como un propósito encontrar esta sección y, por eso o porque no es fácil de encontrar, no fue posible observarla.

Estas herramientas, además separar el trabajo, objetivan las jerarquías de los participantes, por medio de la relación entre personas y accesos de lectura y de escritura. Los accesos de lectura se encuentran, en su mayoría abiertos a Internet, es decir, se encuentran disponibles, para que la gente los descargue, los utilice y/o modifique. Los accesos de escritura, dependiendo del sistema, se encuentran cerrados en distintos niveles, como la Tabla 10 lo muestra.

Sistema o herramienta	Lectura	Escritura
Git	Abierta	Altamente restringida
Bugzilla	Abierta	Restringida
Sitio web	Abierta	Restringida
Moblin-dev	Abierta	Abierta

Tabla 10: "Herramientas y acceso". Elaboración propia.

Esta restricción de acceso, se relaciona con el modelo de trabajo "Bazar", el cual, es caracterizado por Raymond (2000), porque los desarrolladores (equipo central) consideran a sus usuarios como co-desarrolladores. En Moblin el modelo de trabajo se manifiesta en la capacidad que tienen los participantes periféricos y legítimos para colaborar en el proyecto, pero siempre con la mediación de uno o varios miembros del equipo central, en este sentido, el Bazar no está relacionado con la eliminación de las jerarquías, al menos no en lo referente a la administración de los sistemas de información; sino, con la capacidad que tienen los sujetos externos a Moblin (no participantes) en tomar lo ya desarrollado y, con ello, crear su propio proyecto de software libre, es decir, en el marco legal de las licencias públicas utilizadas en Moblin, en especial, la GNU-GPL.

La capacidad que tienen sujetos externos para retomar Moblin y utilizarlo a su favor, es consecuencia de que Moblin, hace lo mismo con otros proyectos. Uno de los beneficios que tiene este modelo de trabajo, es que las innovaciones que se hacen de los trabajos que se retomados, en teoría, pueden regresar al trabajo original, esta decisión la toma el equipo central de cada proyecto, lo importante es señalar que la posibilidad existe. Esta libertad en el flujo del código entre proyectos, permite que cada proyecto reciban retroalimentación directa o indirecta de sus usuarios, en este caso, otros proyectos.

Por ejemplo, el Kernel Linux es retomado por cientos de proyectos, que lo utilizan, modifican, comparten y transforman; estos cambios, por causa de la GNU-GPL se hacen públicos en Internet y, por medio de la red de participantes de Linux, pueden o no regresar al Kernel. Cada innovación, una

vez en el Kernel es retomada por los mismos u otros cientos de proyectos y el proceso sucede otra vez.

Si se sitúa este modelo de trabajo en Moblin, por lo observado en Moblin-dev, el desarrollo del código dependió más de los proyectos de los cuales se nutre Moblin (como Linux, GNOME o Mozilla, por mencionar algunos), que de los participantes periféricos de Moblin. En este sentido, sería más correcto pensar al proyecto Moblin como ese usuario-codesarrollador que participa con los proyectos de los cuales se nutre.

Las necesidades de los participantes, pueden ser divididas en tres grupos: aquellas que pertenecen a los proyectos de los cuales se nutre Moblin; aquellas pertenecientes a Moblin, administradas por su equipo central y; aquellas relacionadas a los desarrolladores de segundo nivel, vendedores y usuarios finales de Moblin.

Esta tesis se sitúa en el tercer grupo, desarrolladores de segundo nivel, vendedores y usuarios finales; y como tal, identifica que hay necesidades diferenciadas para cada uno de sus integrantes. Para los desarrolladores de segundo nivel, su necesidad principal es que sus aplicaciones puedan ser utilizadas en Moblin; para los vendedores, su necesidad principal es adaptar Moblin a un hardware determinado para formar un nuevo producto y comercializarlo; para los usuarios finales, su necesidad principal es utilizar Moblin en sus *netbooks*.

Para los desarrolladores de segundo nivel, por lo observado en el análisis realizado en Moblin-dev, las principales necesidades fueron, primero, utilizar las herramientas de desarrollo que el proyecto hizo disponibles para adaptar aplicaciones a Moblin, como MIC2 (*Moblin Image Creator 2*) y MCM (*Moblin Package Manager*); y segundo, virtualizar Moblin en sus estaciones de trabajo y probar cómo funcionaban sus aplicaciones en Moblin, el principal impedimento para lograrlo, fueron las limitaciones de Moblin para trabajar con procesadores que no fueran compatibles con el set de instrucciones *Supplemental Streaming SIMD Extensions 3* (SSSE3).

Para los vendedores su principal necesidad fue adaptar Moblin de la mejor manera a hardwares específicos, en particular *netbooks*. Sus principales complicaciones fue la instalación de controladores de dispositivos periféricos como tarjetas *wireless* y *touchscreen*; la compatibilidad de Moblin con procesadores (el problema del SSSE3) y tarjetas de video (en particular la GMA500) y; conocer cómo trabajar con la interfaz gráfica de Moblin (en concreto, Clutter y Mutter).

Los usuarios finales comparten, los problemas relacionados con controladores, la incompatibilidad de los procesadores y las tarjetas de video y; el entendimiento de la interfaz gráfica. Una necesidad extra que tienen, es instalar y en algunos casos adaptar, los programas a los cuales están

acostumbrados en Moblin, por ejemplo Skype.

8.2 Pendientes

El principal pendiente que deja esta investigación, es el acercamiento a los participantes, el no tener esta etapa metodológica desarrollada, deja importantes lagunas en la interpretación de los procesos de comunicación, en especial, en lo que refiere a la interpretación de las necesidades subyacentes y, a los modelos de trabajo.

Por el lado de las necesidades que los participantes buscan satisfacer mediante su participación en Moblin, resulta complicado lograr una interpretación veraz, sin la reflexión de los sujetos sobre sus acciones. Esta es la razón principal por la cual, la interpretación queda en un nivel superficial, producto de la revisión de sus conversaciones.

Por el lado de los modelos de trabajo, quedan ensombrecidos casi por completo, los procedimientos por medio de los cuales los sujetos se desplazan de la periferia de Moblin, como comunidad de práctica, hacia el centro; es decir, reciben el reconocimiento y los accesos en uno de los sistemas de información, por parte del equipo central para desarrollar una actividad. En el momento en que se redactan estas conclusiones, no hay prácticamente elementos confiables a partir de los cuales interpretar la forma en que se distribuyen los accesos en los sistemas, se asignan responsabilidades, se negocian jerarquías, se administra a los participantes y la participación periférica. El principal beneficio de obtener esta información, es la formulación clara de posiciones, roles, jerarquías y procesos de socialización en el proyecto.

Otro punto final que queda pendiente y, que no parece ser solucionado solo con entrevistas a participantes de Moblin es, el de las relaciones entre los proyectos, en especial, el tema del *mainstream* (proceso de liberar el código de una aplicación o proyecto en la Internet) y el *downstream* (proceso de retomar y adaptar una aplicación o proyecto), por ejemplo, el tomar el Kernel Linux directo de la Fundación Linux (*mainstream*), configurarlo y adaptarlo a las necesidades de Moblin (*downstream*).

9 Bibliografía

9.1 Bibliografía Básica

Anderson, P., & Tushman, M. L. (1990). Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change. *Administrative Science Quarterly*, 35, 604-633.

Bacon, J. (2009). *The Art of Community*. Sebastopol: O'Reilly

Bakardjieva, M., & Feenberg, A. (2002). *Communitie Technology and Democratic*

Rationalization. *The Information Society*, 18, 181-192.

Beecher, K., Capiluppi, A., & Boldyreff, C. (n.d.). Identifying exogenous drivers and evolutionary stages in FLOSS projects. *The Journal of Systems and Software*, (82). doi: 10.1016/j.jss.2008.10.026.

Berdou, E. (2007). *Managing the Bazaar: Commercialization and peripheral participation in mature, community-led Free/Open source software projects*. PhD in Media and Communications, London School of Economics and Political Science.

Bugzilla (2006). The Bugzilla Guide - 2.18.6 Release. Retrieved November 02, 2010, from <http://www.bugzilla.org/docs/2.18/html/>

Carillo, K., & Okoli, C. (2008). The Open Source Movement: a revolution in software development. *Journal of Computer Information Systems*.

Castells, M. (2006a). *La Era de la información*. Vol. I. México: Siglo XXI.

----- (2006b). *La Era de la información*. Vol.II. México: Siglo XXI.

----- (2000). La ciudad de la nueva economía. *La Factoria*, 12. Retrieved April 10, 2009, from <http://www.lafactoriaweb.com/articulos/castells12.htm#>.

Code is Law | *Harvard Magazine* January-February 2000. (n.d.). Retrieved October 16, 2009, from <http://harvardmagazine.com/2000/01/code-is-law.html>.

Coleman, G. (2004). The political Agnosticism of Free and Open Source Software and Inadvertent Politics of Contrast. *Anthropological Quarterly*, 77(3), 507-519.

Coleman, G., & Hill. (2005). *The Social Production of Ethics in Debian and Free Software Communities: Anrthropological Lessons for Vocational ethics*. Idea Group.

Dalle, J., & David, P. (n.d.). The Allocation of Software Development Resources in 'Open Source' Production Mode | *SIEPR*. Retrieved October 23, 2009, from <http://siepr.stanford.edu/publicationsprofile/451>.

Ducheneaut, N. (2005). Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work*, 14. doi: 10.1007/s10606-005-9000-1.

Elliot, M. S., & Scacchi, W. (2003, April). *Free Software: A Case Study of Software Development in a Virtual Organizational Culture*. University of California Irvine.

Fang, Y., & Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, 25(4). doi: 10.2753/MISO742-1222250401.

- Fitzgerald, B. (2006). The transformation of Open Source Software. *MIS Quarterly*, 30(3).
- Flitchy, P. (2007). The internet imaginarié. EUA: MIT.
- Garzarelli, G., Limam, R., & Bjorn Thomassen, Y. (2008). Open Source Software and Economic Growth: A classical division of labor Perspective. *Information Technology for Development*, 14(2). doi: 10.102/itdj.20092.
- German, D. (2004). Decentralized open source global software development, the GNOME experience. *Journal of Software Process: Improvement and Practice*, 8(4). doi: 10.1002/spip.189.
- Ghosh, R. A., & Prakash, V. V. (2000). Orbiten Free Software Survey. *First Monday*, 5(7). Retrieved November 17, 2009, from <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/769/678>.
- Giddens, A. (2006). *La constitución de la sociedad. Bases para la teoría de la estructuración*. (p.412). Buenos Aires: Amorrortu.
- Grudin, J. (n.d.). CSCW: Its History and Participation. research.microsoft. Retrieved October 25, 2009, from <http://research.microsoft.com/en-us/um/people/jgrudin/past/papers/ieee94/ieeecomplastsb.html>.
- Himanen, P. (2002). *La ética hacker y el espíritu de la era de la información*. Imago Mundi (p. 257). Barcelona: Destino.
- von Hippel, E. (2002). *Open source projects as horizontal innovation networks - by and for users*. MIT Sloan School of Management Working.
- (2005). Open Source Software Projects as User Innovation Networks. In J. Feller (Ed.), *Perspectives on Free and Open Source Software*. Cambridge/ London: MIT Press.
- von Hippel, E., & von Krogh, G. (2002). *"Private-Collective" Innovation Model: Issues for Organization Science*. MIT Slon School of Management.
- Jaisingh, J., See-To, E., & Tam, K. Y. (n.d.). The impact of Open Source Software on the Strategic Choices of Firms Developing Proprietary Software. *Journal of Management Information Systems*, 55(3). doi: 10.2753/MIS0742-1222250307.
- Jensen, C. (2001). Modelos comunicantes: la importancia de los modelos para la investigación sobre los mundos de la Internet. *Comunicación y Sociedad*, 40. México: Universidad de Guadalajara.
- Jordan, T. (2004). *Hactivism and Cyberwars: Rebels with a Cause*. Londres: Routledge.
- Jørgensen, N. (2001). Putting it all in the trunk: incremental software development in the FreeBSD open source project. *Information Systems Journal*, 11(4). doi: 10.1111/j.1365-

2575.2001.00113.x. .

Jørgensen, N. (2005). Incremental and Decentralized Integration in FreeDBS. In J. Feller (Ed.), *Perspectives on Free and Open Source Software*. Cambridge/ London: MIT Press.

Kaminsky, D. (1999). Core Competencies: Why Open Source Is The Optimum Economic Paradigm for Software? *Doxpara Research*. personal, . Retrieved November 18, 2008, from <http://www.doxpara.com/read.php/core.html> .

Kelemen, M. (2001). Communitie and its "virtual" promises. *Information, Communication & Society*, 4(3), 370-387.

Kelty, C. (2001). Free Software/ Free Science. *First Monday*, 6(12-3). Retrieved November 6, 2009, from <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/902/811>.

----- (2004). *Punt To Culture*. *Anthropological Quartely*, 77(3).

----- (2008). Two bits. The cultural significance of free software. *Tecnological Lives, Scientific Arts, Antropological Voices*. Durham and London: Duke University Press.

Kollock, P. (1999). The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace. In Smith and Kollock (Ed.), *Communities in Cyberspace*. London: Routledge. Retrieved from http://www.sscnet.ucla.edu/soc/faculty/kollock/papers/communities_01.htm.

Lakhani, K. R., & Wolf, R. G. (2003). *Why hackers do what they do: Understanding motivation and effort in free/Open Source Software Projects*. MIT Sloan Working Paper . doi: 10.2139/ssrn.443040.

----- (2005). Why hackers do what they do: understanding motivation and effort in Free/Open Source Software projects. In J. Feller (Ed.), *Prspectives on Free and Open Source Software*. Cambridge/ London: MIT Press.

Lerner, J., & Tirole, J. (2000). *Economic perspective on open source*. Boston: Harvard Business School.

Lessing, L. (2005). Open Code and Open Societies. In J. Feller (Ed.), *Perspectives on Free and Open Source Software*. Massachusetts: MIT Press.

Lloyd, A. (2007). "A system that works for me" - an anthropological analysis of computer hackers' chared use and development of the Ubuntu Linux system. MA thesis in Anthropology, Copenhagen.

Mattelart, A. y Mattelart, M. (1997). *Historias de las teorías de la comunicación*. Barcelona: Paidós

Mahendran, D. (2002). *Serpents and Primitives. An ethnographic excursion into an Open Source community*. MA in Information, Mgmt. and Systems, University of California at Berkeley.

Markus, M. L. (2007). The governance of free/open source software projects: monolithic, multidimensional, or configurational? *Journal of Manage Governance*, (11). doi: 10.1007/s10997-007-9021-x.

McGowan, D., & Feller, J. (2005). Legal aspects of free and open source software. In *Perspectives on Free and Open Source Software*. Massachusetts: MIT Press.

McGowan, D. (2001). Legal aspects of free and open source software. *SSRN*. doi: 10.2139/ssrn.243237.

Mills, W. (1997). *La imaginación sociológica*. México: FCE.

Mockus, A., Fielding, R., & Herbsleb, J. (2005). Two case studies of Open Soure Software Development: Apache adn Mozilla. In J. Feller (Ed.), *Perspectives on Free and Open Source Software*. Cambridge/ London: MIT Press.

Mockus, A., Fielding, R., & Herbsleb, J. (2002). Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3).

Mounier, P. (2002). *Los dueños de la Red. Una historia política de Internet*. Rompeolas. Madrid: Popular.

Negroponte, N. (1996). *Being Digital*. Londres: Coronet.

Noonan, D., Baker, P. M., & Moon, N. W. (2008, June). Open source software potential index (OSPI): development consideratios. *Red Hat/ Georgia Tech OSPI Project*. Retrieved from http://www.redhat.com/f/pdf/OSSI_Research.pdf.

Pfaffenberger, B. (1992). Social Anthropology of technology. *Annual Review of Anthropology* No 21.

Raymond, E. (2000). The Cathedral and the Bazaar. *Eric S. Raymond's Home Page*. Retrieved November 18, 2008, from <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>.

----- (2008). Halloween Document I (Version 1.17). *Eric's Home Page. Personal*, . Retrieved November 18, 2008, from <http://www.catb.org/esr/halloween/halloween1.html>.

Robbins, J. (2005). Adopting Open Source Software Engineering (OSSE). Practices by Adopting OSSE Tolls. In J. Feller (Ed.), *Perspectives on Free and Open Source Software*. Massachusetts: MIT Press.

Sack, W. (2001). Conversation Map: An interface for very large-scale conversations. *Journal of Management Information Systems*, 17(3).

Sack, W., Détienne, F., & Ducheneaut, N. (2006). A methodological framework for socio-cognitive analyses of collaborative design of open source software. *Computer Supported Cooperative Work*, 250. doi: 10.1007/s10606-006-9020-5.

Scolari, C. (2008). *Hipermediaciones. Elementos para una Teoría de la Comunicación Interactiva*. Barcelona: Gedisa.

Sen, R., Chandrasekar, S., & Mathew, L. N. (2009). Determinants of the Choice of Open Source Software Licence. *Journal of Management Information Systems*, 25(3). doi: 10.2753/MIS0742-1222250306.

Stewart, K. M., & Gosain, S. (2006). The moderating role of development stage in Free/Open Source project performance. *Journal of Software Process Improvement Practice*, (11). doi: 10.1002/spip.258.

Tapscott, D., & Williams, A. D. (2007). *Wikinomics. La nueva economía de las multitudes inteligentes*. Barcelona: Paidós.

Teil Geneviève, & Latour, B. (1995). The Man-machine and Artificial Intelligence. *Stanford Humanities Review* vol 4.2. Retrieved September 17, 2009, from <http://www.stanford.edu/group/SHR/4-2/text/mazlish.html>.

Thompson, J. B. (1993). *Ideología y cultura moderna. Teoría crítica social en la era de la comunicación de masas*. México: UAM X.

Trejo-Delarbre, R., & Sosa-Plata Gabriela. (2009). Campo nuevo, problemas viejos. La investigación mexicana en materia de Sociedad de la Información, Internet, Cibercultura y Telecomunicaciones. In Vega (Coord.): *La comunicación en México. Una agenda de investigación* (pp. 1-23). México: UNAM-FCPyS-CEIICH-UJAT-UABC-AMIC.

Turkle, S. (1999). What Are We Thinking About When We Are Thinking About Computers? In M. Biagioli (Ed.), *The Science Studies Reader*. New York: Routledge.

Watson, R., Bourdreau, M., York, P., Greiner, M., & Wynn, D. (n.d.). The bussines of Open Source. Tracking the changing competitive conditions of the software industry. *Communications of the ACM*, 51(4).

Wenger, E. (1998). *Communities of Practice, Meaning and Identity*. Cambridge: Cambridge University Press.

----- (2001). *Comunidades de práctica. Aprendizaje, significado e identidad*. Cognición y desarrollo humano. Barcelona: Paidós.

Williams, S. (2001). *Free as in Freedom. Richard Stallman's crusade for free software*. (p. 240). O'Reilly. Retrieved November 15, 2008, from <http://oreilly.com/openbook/freedom/>

9.2 Sitios de donde se extrajo la información para la investigación

Sobre Moblin

Moblin-website: <http://moblin.org/> consultado de 2009 a 2010.

Moblin-dev: <http://lists.moblin.org/pipermail/dev/> consultado de 2009 a 2010.

Moblin-Bugzilla: <http://bugzilla.moblin.org/> consultado de 2009 a 2010.

Moblin-Git: <http://git.moblin.org/> consultado de 2009 a 2010.

Sobre los sistemas de información

Chacon, S. (n.d.). *Pro Git. Profesional Version Control*. Retrived November 02, 2010, from <http://progit.org/>

Mailman Documentation (n.d.). Retrived november, 02, 2010, from <http://www.gnu.org/software/mailman/docs.html>

The Bugzilla Guide (2006). *The Bugzilla Team*. Retrived November 02, 2010, from <http://www.bugzilla.org/docs/2.18/html/>

Sobre los participantes

Burtonini.com: <http://burtonini.com> consultado el 02 de Noviembre de 2010.

Fridu.org: <http://www.fridu.org/> consultado el 02 de Noviembre de 2010.

Lincor Solutions Ltd.: <http://lincor.com> consultado el 02 de Noviembre de 2010.

LinuxLighthouse: <http://linuxlighthouse.com> consultado el 02 de Noviembre de 2010.

Suse Linux: <http://suse.de> consultado el 02 de Noviembre de 2010.

Xandrox: <http://xandros.com> consultado el 02 de Noviembre de 2010.

9.3 Bibliografía Complementaria

Bajarin, T. (2007, April 18). Intel Divulges Mobile-Computing Roadmap - Columns by PC Magazine. *PC Magazine. News*, . Retrieved November 12, 2009, from <http://www.pcmag.com/article2/0,2817,2115380,00.asp>.

Birckner, J., & Reddig, F. (2009, May 10). Business Source: The Revolution Will Be Mobile. *Business Source Complete (EBSCOhost)*. Info. Source, . Retrieved November 14, 2009, from <http://web.ebscohost.com/bsi/detail?vid=5&hid=111&sid=b17c549e-7a75-471d-b820-d3e80b65176c%40sessionmgr113&bdata=JnNpdGU9YnNpLWxpdmU%3d#db=bth&AN=44745867>.

Brown, E. (2008a, November 14). HOT TOPIC: Moblin - Linux For Devices Articles - Linux for Devices. *Linux For Devices*. News, . Retrieved September 9, 2009, from <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/HOT-TOPIC-Moblin/>.

----- (2008b, July 30). Linux MID stack to debut at IDF - News - *Linux for Devices*. Linux For Devices. News, . Retrieved November 12, 2009, from <http://www.linuxfordevices.com/c/a/News/Linux-MID-stack-to-debut-at-IDF/>.

----- (2008c, June 3). Linux distributor joins Moblin - News - *Linux for Devices*. Linux For Devices. News, . Retrieved November 12, 2009, from <http://www.linuxfordevices.com/c/a/News/Linux-distributor-joins-Moblin/>.

----- (2008d, June 7). Moblin switching from Ubuntu to Fedora - News - *Linux for Devices*. Retrieved November 12, 2009, from <http://www.linuxfordevices.com/c/a/News/Moblin-switching-from-Ubuntu-to-Fedora/>.

Commit History for juanje's moblin-image-creator - *GitHub*. (n.d.). . Retrieved September 9, 2009, from <http://github.com/juanje/moblin-image-creator/commits/master>.

Crits-Christoph, A. (2009, October 1). Innovative Interfaces with Clutter. *Linux Journal*. HOWTOs, . Retrieved November 15, 2009, from <http://www.linuxjournal.com/article/10318>.

Fairchild, C. (2008, May 14). Verizon Joins Linux for Mobile Devices | *Linux Journal*. Linux Journal. News, . Retrieved November 16, 2009, from <http://www.linuxjournal.com/content/verizon-joins-linux-mobile-devices-limo>.

Ganapati, P. (2009a, January 30). Intel Pushes New Operating System For Netbooks | Gadget Lab | *Wired.com*. *Wired*. Retrieved November 12, 2009, from <http://www.wired.com/gadgetlab/2009/01/intel-bets-on-m/>.

Ganapati, P. (2009b, July 6). Why Intel's Processors Aren't Big on Cellphones | Gadget Lab | *Wired.com*. *Wired*. News, . Retrieved November 13, 2009, from <http://www.wired.com/gadgetlab/2009/07/atom-processor-phones/>.

Ganapati, P. (2009c, June 23). Intel, Nokia To Create New Mobile Architecture and Devices | Gadget Lab | *Wired.com*. *Wired*. News, . Retrieved November 13, 2009, from

<http://www.wired.com/gadgetlab/2009/06/intel-nokia/>.

Ganapati, P. (2009d, September 22). Intel Hints at Its Own Smartphone Operating System | Gadget Lab | *Wired.com*. *Wired*. News, . Retrieved November 13, 2009, from <http://www.wired.com/gadgetlab/2009/09/intel-new-version-of-moblin/>.

Gray, J. (2008, December 1). Linux Device Roundup. *Linux Journal*. News, . Retrieved November 15, 2009, from <http://www.linuxjournal.com/article/10258>.

HP Announces Linux Support On HP NetServer and Future IA-64 Architecture-based Systems | *Business Wire* | Find Articles at BNET. (n.d.). . Retrieved October 23, 2009, from http://findarticles.com/p/articles/mi_m0EIN/is_1999_Jan_27/ai_53650520/?tag=content;coll.

Icaza, M. (n.d.). *The Story of the GNOME project*. Retrieved May 11, 2009, from <http://primates.ximian.com/~miguel/gnome-history.html>.

Intel debuts Linux-based "Mobile Internet Device" - News - *Linux for Devices*. (n.d.). . Retrieved September 9, 2009, from <http://www.linuxfordevices.com/c/a/News/Intel-debuts-Linuxbased-Mobile-Internet-Device/>.

Intel Developer Forum – Embedded Technology. (n.d.). . Retrieved November 12, 2009, from <http://developer.intel.com/idf/us/fall2008/training/embedded.htm?iid=idf+MarqueeNav+Subnav+Training+Embedded>.

Intel GMA950 VS GMA500 | *The Netbook Market*. (n.d.). . Retrieved November 15, 2009, from <http://www.netbookmarket.net/intel-gma950-vs-gma500/>.

Intel launches mobile Linux initiative - News - *Linux for Devices*. (n.d.). . Retrieved September 9, 2009, from <http://www.linuxfordevices.com/c/a/News/Intel-launches-mobile-Linux-initiative/>.

intel_mid_arch.gif (GIF Image, 965x546 pixels). (2007, April 16). *Linux For Devices*. News, . Retrieved November 12, 2009, from http://www.linuxfordevices.com/files/misc/intel_mid_arch.gif.

intel_moblin_media_artist_screen02.gif (GIF Image, 500x301 pixels) - Scaled (88%). (2009, July 16). *Linux For Devices*. News, . Retrieved November 12, 2009, from http://www.linuxfordevices.com/files/misc/intel_moblin_media_artist_screen02.gif.

intel_moblin_ppm_architecture.jpg (JPEG Image, 840x555 pixels). (2007, July 16). *Linux For Devices*. News, . Retrieved November 12, 2009, from http://www.linuxfordevices.com/files/misc/intel_moblin_ppm_architecture.jpg.

intel_ultra_mobile_2007_chipset_diag.gif (GIF Image, 693x875 pixels). (2007, April 18). *Linux For Devices*. News, . Retrieved November 12, 2009, from

http://www.linuxfordevices.com/files/misc/intel_ultra_mobile_2007_chipset_diag.gif.

Lehrbaum, R. (2009, June 4). *DeviceGuru* » Intel swallows Wind River. *deviceguru.com*. Blogg, . Retrieved November 15, 2009, from <http://www.deviceguru.com/intel-swallows-wind-river/>.

----- (2008, April 23). *DeviceGuru* » Netbooks to take the market by storm. *deviceguru.com*. Blogg, . Retrieved November 15, 2009, from <http://www.deviceguru.com/netbooks-to-take-the-market-by-storm/>.

Lemon, S. (2008, August 29). Atom supply still stymied by testing bottleneck | *Hardware - InfoWorld*. *Infoworld*. News, . Retrieved November 13, 2009, from <http://www.infoworld.com/t/hardware/atom-supply-still-stymied-testing-bottleneck-346>.

Linux Device Roundup. (n.d.). *Linux Journal* . Retrieved November 15, 2009, from <http://www.linuxjournal.com/article/10258>.

Linux Devices. (2008, October 29). Intel, Taiwan tag-team Moblin, WiMAX - News - Linux for Devices. *Linux For Devices*. News, . Retrieved November 12, 2009, from <http://www.linuxfordevices.com/c/a/News/Intel-Taiwan-tagteam-Moblin-WiMAX/>.

Linux Devices. (2007, April 18). Intel launches mobile device chips, roadmap - News - Linux for Devices. *Linux For Devices*. News, . Retrieved November 12, 2009, from <http://www.linuxfordevices.com/c/a/News/Intel-launches-mobile-device-chips-roadmap/>.

Linux Devices. (2006, December 11). Building consumer products with open source - Linux For Devices Articles - *Linux for Devices*. Retrieved November 13, 2009, from <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Building-consumer-products-with-open-source/>.

LinuxJournal. (2009, March 1). UpFront. *Linux Journal*. News, . Retrieved November 15, 2009, from <http://www.linuxjournal.com/article/10363>.

Meyer, D. (2008, June 12). Nokia: Linux Needs to Learn Business - *BusinessWeek*. *BusinessWeek*. Analisis, . Retrieved November 13, 2009, from http://www.businessweek.com/globalbiz/content/jun2008/gb20080612_288518.htm?campaign_id=rss_topStories.

moblin. (n.d.). About Moblin | *moblin.org*. *Moblin.org*. *Official web site*, . Retrieved November 15, 2009, from <http://moblin.org/about-moblin>.

Montalbano, E. (2009, April 1). Intel to turn Moblin over to Linux Foundation - *LinuxWorld*. *LinuxWorld*. Retrieved November 12, 2009, from <http://www.linuxworld.com/news/2009/040109-intel->

[to-turn-moblin-over.html](#).

Nystedt, D. (2009, June 2). Acer may be first with Android netbook. *Computer World Hardware*. News, . Retrieved November 15, 2009, from http://www.computerworld.com/s/article/9133800/Acer_may_be_first_with_Android_netbook.

Powers, S. (2009, October 27). How To Kick Your Friends in the Face: GMA500 | Linux Journal. *Linux Journal*. News, . Retrieved November 15, 2009, from <http://www.linuxjournal.com/content/how-kick-your-friends-face-gma500>.

redhat.com | Open Source World Map. (n.d.). *Red Hat*. Retrieved May 11, 2009, from <http://www.redhat.com/about/where-is-open-source/activity/>.

Richmond, G. (2008a, July 2). Acer's Linpus Linux Lite (Fedora) ultra portable laptop piles the pressure on Microsoft. *FSF Magazine*. Analisis, . Retrieved November 15, 2009, from http://www.freesoftwaremagazine.com/columns/acers_linpus_linux_lite_ultra_portable_laptop.

Richmond, G. (2008b, January 31). The Asus Eee PC: An Ultra-portable laptop PC with GNU/Linux pre-installed. *FSF Magazine*. Analisis, . Retrieved November 15, 2009, from http://www.freesoftwaremagazine.com/columns/asus_eee_pc_ultra_portable_laptop_gnu_linux_pre-installed.

Roberts, J. (2007, June 18). Interview With Matthew Allum of OpenedHand. *FSF Magazine*. Retrieved November 15, 2009, from http://www.freesoftwaremagazine.com/columns/interview_matthew_allum_openedhand.

Ryan, J. (2009, June 2). Invasion of the Android Snatchers | Linux Journal. *Linux Journal*. News, . Retrieved November 15, 2009, from <http://www.linuxjournal.com/content/invasion-android-snatchers>.

Ryan, P. (2009a, January 28). Intel releases Linux-based Moblin 2 Alpha for Netbooks - *Ars Technica*. *Ars technica*. Retrieved November 12, 2009, from <http://arstechnica.com/open-source/news/2009/01/intel-releases-linux-based-moblin-2-alpha-for-netbooks.ars>.

----- (2009b, May 20). Intel wants to pimp your ride with prodigious penguin power - *Ars Technica*. *Ars technica*. News, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/05/intel-wants-to-pimp-your-ride-with-prodigious-penguin-power.ars>.

----- (2009c, July 19). Intel launches site for open source mobile Linux development - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2007/07/intel-launches-site-for-open-source-mobile-linux-development.ars>.

----- (2009d, June 3). 23% of smartphone market to be Linux-powered by 2013 - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/06/23-of-smartphone-market-to-be-linux-powered-by-2013.ars>.

----- (2009e, July 19). Intel launches site for open source mobile Linux development - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2007/07/intel-launches-site-for-open-source-mobile-linux-development.ars>.

----- (2008a, November 1). This week in open source: Ubuntu 8.10 released, Chrome, and Android - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/11/this-week-in-open-source-ubuntu-chrome-and-android.ars>.

----- (2008b, November 3). Motorola and Google become GNOME sponsors - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/11/motorola-and-google-become-gnome-sponsors.ars>.

----- (2008c, November 28). Intel's Linux-based Moblin platform arrives on devices - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/11/intels-linux-based-moblin-platform-arrives-on-devices.ars>.

----- (2008d, January 28). Intel releases Linux-based Moblin 2 Alpha for Netbooks - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2009/01/intel-releases-linux-based-moblin-2-alpha-for-netbooks.ars>.

----- (2008e, December 3). Nokia's OSS strategy gets boost as Symbian deal completed - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2008/12/nokias-oss-strategy-gets-boost-as-symbian-deal-completed.ars>.

----- (2009a, May 8). Novell to offer own Moblin-based distro for hardware makers - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2009/05/novell-boosts-commitment-to-intels-moblin-linux-platform.ars>.

----- (2009b, April 2). Linux Foundation to host development of Intel's Moblin OS - *Ars Technica*. *Ars technica*. Analisis, . Retrieved November 13, 2009, from <http://arstechnica.com/open-source/news/2009/04/linux-foundation-to-host-development-of-intels-moblin-os.ars>.

----- (2009c, May 19). Hands-on: Intel brings rich UI to Moblin Linux platform - *Ars Technica*. *Ars technica*. News, . Retrieved November 16, 2009, from <http://arstechnica.com/open-source/news/2009/05/intel-brings-rich-ui-to-moblin-linux-platform.ars>.

source/reviews/2009/05/hands-on-intel-brings-rich-ui-to-moblin-linux-platform.ars.

Shureih, T. (2009, May 21). Announcing Moblin V2 Core Alpha Release | *moblin.org*. Retrieved November 12, 2009, from <http://moblin.org/community/blogs/tshureih/2009/announcing-moblin-v2-core-alpha-release>.

Sood, R. (2009, January 21). rahulsood: Could the Appetite for Netbooks Curb PC Innovation? *rahulsood.com. Personal Blogg*, . Retrieved November 12, 2009, from <http://www.rahulsood.com/2009/01/intel-atom-vs-via-nano-vs-netbook.html>.

Sorrel, C. (2008, October 8). Linux Eee PC Boots in Five Seconds | Gadget Lab | *Wired.com*. *Wired*. News, . Retrieved November 13, 2009, from <http://www.wired.com/gadgetlab/2008/10/linux-eee-pc-bo/>.

Sousou, I. (2009, February 23). Moblin Core | *moblin.org*. *Moblin.org. Oficial web site*, . Retrieved November 15, 2009, from <http://moblin.org/documentation/moblin-overview/moblin-core>.

Takahashi, D. (2009, June 22). Has Intel finally broken into the cell phone market? | *VentureBeat. DigitalBeat. News*, . Retrieved November 13, 2009, from <http://digital.venturebeat.com/2009/06/22/has-intel-scored-nokia-as-a-customer-for-cell-phone-chips/>.

The Linux Information Project. (2006, Feb 14). Source Cod Definition. *linfo.org*. Retrived April 27, 2010, from http://www.linfo.org/source_code.html

The Story of the GNOME project. (n.d.). *Primates Ximian. Personal Web-site of the Ximian folks* . Retrieved May 11, 2009, from <http://primates.ximian.com/~miguel/gnome-history.html>.

Thomas, B. (2008, May 20). Wind River - Wind River Announces Disruptive Infotainment Platform For Automotive Industry. *windriver.com. Company page*, . Retrieved November 13, 2009, from <http://www.windriver.com/news/press/pr.html?ID=6041>.

Torvalds, L. (2007). *YouTube - Tech Talk: Linus Torvalds on git*. Retrieved May 11, 2009, from <http://www.youtube.com/watch?v=4XpnKHJAok8>.

UNCTAD. (2008). *Creative Economy Report 2008*. The Challenge of Assessing the Creative Economy: towards Informed Policy-making. ONU.

Vance, A. (2009, April 1). Intel Turns Over Its Linux Operating System - Bits Blog - *NYTimes.com. NY Times. News*, . Retrieved November 9, 2009, from <http://bits.blogs.nytimes.com/2009/04/01/intel-hands-over-the-keys-to-its-linux-operating-system/>.

Vepstas, L. (2001). Is Free Software Inevitable? *The Home Page of Linas VEPSTAS. Personal*, . Retrieved November 18, 2008, from <http://www.linas.org/theory/freetrade.html> .

Virki, T. (2008, December 2). Nokia eyes wider use of Linux software in phones | Technology | Internet | Reuters. *Reuters.com. News*, . Retrieved November 15, 2009, from <http://www.reuters.com/article/internetNews/idUSTRE4B16IO20081202>.

Weinberg, B. (2009, June 4). Intel to Acquire Wind River – Embedded Industry Realignment Coming « *LinuxPundit Weblog. LinuxPundit Weblog. Blog*, . Retrieved November 15, 2009, from <http://linuxpundit.wordpress.com/2009/06/04/intel-to-acquire-wind-river-embedded-industry-realignment-coming/>.

Weinberg, B. (2008, January 28). Nokia and Trolltech: Out loud and on Qt . *LinuxPundit Weblog. Blog*, . Retrieved November 15, 2009, from <http://linuxpundit.wordpress.com/2008/01/>.

Wheeler, D. A. (2007). Why open Source Software/Free Software (OSS/FS, FLOSS or FOSS)? Look at the Numbers! *David A. Wheeler's Personal Home Page. Personal*, . Retrieved November 18, 2008, from http://www.dwheeler.com/oss_fs_why.html .