Gestión del conocimiento mediante comunidades de práctica virtuales: aplicación a proyectos de software de código abierto

Autor: Sergio Toral Marín

Directora: Profª. Dra. Mª del Rocío Martínez Torres

Departamento: Administración de Empresas y Comercialización e Investigación de Mercados (Marketing)

Universidad de Sevilla

Sevilla, 27 de Junio de 2010

La Directora de la Tesis Doctoral

Fdo.: Mª del Rocío Martínez Torres

Fdo.: Sergio Toral Marín

Gestión del conocimiento mediante comunidades de práctica virtuales: aplicación a proyectos de software de código abierto

Autor: Sergio Toral Marín

Directora: Profª. Dra. Mª del Rocío Martínez Torres

Departamento: Administración de Empresas y Comercialización e Investigación de Mercados (Marketing)

Universidad De Sevilla

Sevilla, 27 de Junio de 2010

Contenido

Agradecimientos 11

Introducción 13

Capítulo 1. Gestión del conocimiento 19

1.1 Concepto de conocimiento 19

1.2 Términos relacionados con el conocimiento 40

1.3 Conocimiento tácito y explícito 42

1.4 Conocimiento soft y conocimiento hard 48

Capítulo 2. Comunidades de Práctica 51

2.1 Introducción a la Teoría del Aprendizaje Social 51

2.1.1 Teorías del aprendizaje 51

2.2.2 Teoría del aprendizaje social 52

2.2 Concepto de Comunidad de Práctica 53

2.2.1 Tipologías de participantes 56

2.3 Comunidades de práctica y gestión del conocimiento 58

2.4 Comunidades virtuales 61

2.4.1 Creación de conocimiento en comunidades distribuidas 62

2.5 Beneficios e inconvenientes de las comunidades de práctica 65

Capítulo 3. Proyectos de Software de código abierto 67

3.1 Introducción 67

3.1.1 Software propietario, abierto y libre 68

3.1.2 Otras definiciones de términos 71

3.2 Licencias de software de código abierto 73

3.2.1 GNU Public License – GPL 74

3.2.2 GNU Lesser General Public License – LGPL 75

3.2.3 BSD License 75

3.2.4 Mozilla Public License 76

3.3 Teorías sobre el software de código abierto 76
3.4 Proyectos de software de código abierto 81
3.4.1 Ciclo de vida 82
3.4.2 Modelos de negocio 84
3.4.3 Ejemplos 86
3.4.4 Fortalezas y debilidades de los proyectos de código abierto 90
3.5 Comunidades de software de código abierto 93
3.5.1 Participantes 93
3.5.2 Tecnologías de desarrollo 95
Capítulo 4. Gestión del conocimiento soft y del conocimiento hard: participación y cosificación 99
4.1 Introducción 99
4.2 Representación de las comunidades como redes sociales 100
4.3 Características de la participación en comunidades virtuales 104
4.3.1 Desigualdad participativa 105
4.3.2 Estructura 107
4.4 Hipótesis y modelo de participación 111
4.5 Análisis del proceso de cosificación 114
Capítulo 5. Metodología 117
5.1 Análisis factor 117
5.2 PLS 130
5.2.1 El modelo PLS 135
5.2.2 Procedimiento a seguir para la construcción de una medida con indicadores formativos 139
5.2.3 Funcionamiento del modelo 140
5.2.4 Análisis de la validez y la fiabilidad 142
5.3 Análisis semántico 146
Capítulo 6. Resultados: modelo de participación en comunidades de software de código abierto 159
6.1 Caso de estudio 159
6.2 Modelo de participación 165

6.2.1 Indicadores 167
6.2.2 Modelo estructural y de medida 171
6.3 Análisis semántico: cosificación 176
6.3.1 Aplicación a la comunidad Debian-ARM 177
6.3.2 Identificación de las dimensiones latentes 183
Capítulo 7. Conclusiones, limitaciones y futuros trabajos 189
7.1 Conclusiones 189
7.2 Limitaciones 195
7.3 Futuras líneas de investigación 197
Bibliografía 201

Índice de figuras

Figura 1. Creación de conocimiento. 40
Figura 2. Continuum de conocimiento. 47
Figura 3. Dualidad de los aspectos *soft* y *hard* del conocimiento. 49
Figura 4. Modelo de éxito en sistemas de información (DeLone y McLean, 2003) 99
Figura 5. Foro de discusión del proyecto ARM Debian Linux en Octubre de 2007 101
Figura 6. Red social de la comunidad de ARM Debian Linux durante 2007 103
Figura 7. Distribución de las contribuciones para la comunidad ARM Debian Linux durante 2007 105
Figura 8. Áreas para calcular el coeficiente de Gini 106
Figura 9. Curva de Lorentz y diagonal de igualdad para las aportaciones a la lista de distribución de la comunidad ARM Debian Linux 107
Figura 10. Evolución de los miembros activos de la comunidad ARM Linux entre 2002 y 2007 109
Figura 11. Evolución de los gestores de conocimiento de la comunidad ARM Linux entre 2002 y 2007 111
Figura 12. Modelo de éxito de las comunidades basado en la participación 114
Figura 13. Modelo PLS genérico 136
Figura 14. Descomposición de la matriz principal en las dos matrices de vectores singulares y una matriz diagonal de valores singulares 149
Figura 15. Proceso subyacente en los modelos probabilísticos de semántica latente 155
Figura 16. Evolución de los sistemas operativos para sistemas embebidos (fuente: Linuxdevices.com, Snapshot of the embedded Linux market -- Mayo, 2006) 160
Figura 17. Distribuciones de Linux más utilizadas (fuente: Linuxdevices.com, Snapshot of the embedded Linux market -- Mayo, 2006) 161
Figura 18. Diagrama de flujo de la extracción de información de participación 166
Figura 19. Cabecera de un mensaje de la comunidad Debian-arm (Enero 2008). 167
Figura 20. Comunidad Debian PPC durante el año 2005 169
Figura 21. Modelo de participación en comunidades de software de código abierto 174
Figura 22. Diagrama de flujo de la aplicación de técnicas LDA 176

Figura 23. Variación de la perplejidad con el número de tópicos para la comunidad Debian ARM 179

Índice de tablas

Tabla 1. Comparación de las características de las Comunidades de Práctica con otras formas de organización 56
Tabla 2. Creación de conocimiento en el modelo basado en la comunidad frente al modelo basado en la organización 64
Tabla 3. Licencias de software de código abierto más importantes. 74
Tabla 4. Principales diferencias entre indicadores reflectivos y formativos 138
Tabla 5. Listado de comunidades consideradas en el caso de estudio propuesto 163
Tabla 6. Indicadores de medida de los constructos del modelo de participación 170
Tabla 7. Fiabilidad de ítems y constructos 172
Tabla 8. Validez discriminante 173
Tabla 9. Parámetros del algoritmo LDA 177
Tabla 10. Parámetros del algoritmo LDA para la comunidad Debian ARM 178
Tabla 11. Distribución de tópicos por año para la comunidad Debian ARM 181
Tabla 12. Indicadores extraídos del análisis semántico de las comunidades consideradas 183
Tabla 13. KMO y prueba de esfericidad de Bartlett 184
Tabla 14. Varianza total explicada 185
Tabla 15. Cargas de los factores rotadas por el método Varimax 185
Tabla 16. Test de Kruskal-Wallis 187

Agradecimientos

En primer lugar deseo expresar mi agradecimiento a la directora de tesis, Rocío Martínez Torres, por su apoyo y motivación en la realización de la tesis y su valiosa asesoría. Este agradecimiento es extensible al departamento de Administración de Empresas y Comercialización e Investigación de Mercados (Marketing), donde los conocimientos adquiridos han abierto nuevos horizontes investigadores.

Finalmente, y por no extenderme más, agradecer a mi grupo de investigación y, en especial, a Federico Barrero, el apoyo y ánimo para concluir los trabajos aquí presentados.

Introducción

Aunque el concepto de comunidad de práctica fue tratado y desarrollado durante la década de los 90, su verdadero auge ha venido de la mano de las comunidades virtuales, caracterizadas por hacer un uso intensivo de los medios electrónicos como herramienta básica de contacto y de puesta en común de aportaciones. Este auge se debe básicamente al desarrollo de nuevos modelos de funcionamiento de Internet, que es lo que generalmente se denomina bajo el nombre de Web 2.0. La Web actual es muy diferente de la que existía tan sólo hace una década. El motivo es que se ha producido un punto de inflexión por el que se ha pasado de un modelo 'top-down' de creación de información e interacción a un modelo 'bottom-up', gracias a las nuevas aplicaciones Web que otorgan mayor poder a los usuarios. Hasta hace poco Internet se entendía como un gran repositorio de información, donde los usuarios eran consumidores pasivos de esa información. Ahora son los individuos los generadores de información, agrupados en comunidades virtuales que han dado lugar a grandes redes sociales. La aparición de estas redes sociales ha transformado Internet en un medio para conectar personas y compartir información y conocimiento.

En este contexto, este trabajo pretende profundizar en el análisis de las comunidades virtuales como herramientas de gestión del conocimiento y del capital intelectual, en particular, del capital social. Para ello, se estudiarán los dos procesos básicos subyacentes de gestión del conocimiento según la teoría de Hildreth *et al*. (1999). El primero de ellos tiene que ver con la denominada cosificación, que significa hacer concreto lo abstracto. Esto se refiere a los contenidos generados y puestos en común por las comunidades, que se analizarán según las técnicas de análisis semántico basado en modelos generativos para extraer indicadores sobre el conocimiento creado y compartido por los usuarios. El segundo de los procesos se refiere a las actividades de participación, que se analizarán mediante técnicas de análisis de redes sociales. En particular, se medirán aspectos tales como la topología de la comunidad y su cohesión, y se identificarán perfiles clave de usuarios necesarios para el buen desarrollo de la comunidad. En este caso, también se generarán indicadores que permitan medir los procesos de participación.

A partir de los indicadores obtenidos se pretenden validar diversas hipótesis que expliquen los procesos de participación y cosificación.

- El proceso subyacente en la gestión del conocimiento *hard* es la cosificación. Este proceso se analizará utilizando técnicas de análisis semántico. En particular, se emplearán técnicas de Análisis Semántico Latente (LSA, Latent Semantic Análisis) (Zha et al., 2001; Landauer, 2002) como los modelos basados en tópicos (Blei *et al*., 2003; Toral *et al*., 2009d). A partir del análisis semántico se obtendrán los distintos tópicos de discusión de las comunidades virtuales y se definirán indicadores que permitan medir el conocimiento *hard* (Toral *et al*., 2009e).
- El proceso subyacente en la gestión del conocimiento *soft* es la participación. Este proceso se analizará utilizando técnicas de análisis de redes sociales (ver apartado de metodología). Este tipo de técnicas permite modelar una comunidad como un conjunto de vértices y arcos interconectados entre sí (Kautz *et al*., 1997; Toral *et al.*, 2010a). Los vértices representan las personas y los arcos las relaciones entre ellas (Yang y Chen, 2008). El principal objetivo del análisis de las redes sociales es detectar e interpretar patrones de conexiones entre individuos (Nooy *et al*., 2005; Martínez Torres *et al.*, 2009b). En particular se analizarán las estructuras y topologías de las redes formadas por las comunidades virtuales, los principales perfiles dentro del núcleo de la comunidad y se obtendrán indicadores que permitan medir el conocimiento *soft*.

En base a los indicadores obtenidos se tratarán de obtener modelos que justifiquen las hipótesis principales de la propuesta. Los resultados de esta investigación tienen implicaciones importantes sobre la industria del software, que está pasando de esquemas de ingresos basados en la generación de nuevos productos y cobro de licencias, a esquemas de ingresos basados en servicios de valor añadido sobre software libre o de código abierto, cuyo desarrollo se fundamenta en las comunidades virtuales de soporte. Actualmente coexisten el software de código abierto y el software propietario, así como sus modelos de negocio. No obstante, se aprecia una tendencia creciente a la integración de ambas alternativas en unas estrategias globales más amplias.

Los principales objetivos que se persiguen con este trabajo son:

- Caracterizar el comportamiento de las comunidades virtuales como herramientas de desarrollo del conocimiento *hard* y del conocimiento *soft*
- Definir indicadores que permitan medir el conocimiento *hard* mediante el uso de técnicas de análisis semántico.
- Definir indicadores que permitan medir el conocimiento *soft* mediante el uso de técnicas de análisis de redes sociales.
- Obtener un modelo causal de participación, que permita medir el éxito de las comunidades y sus principales antecedentes.
- Identificar las principales dimensiones relacionadas con los procesos de cosificación.

Como principales aportaciones de trabajo realizado cabe destacar:

- La formalización de los procesos de gestión del conocimiento *soft* y *hard* en comunidades de práctica virtuales sobre la base del análisis de redes sociales, para los procesos de participación, y del análisis semántico, para los procesos de cosificación.
- La obtención de un modelo estructural y de medida que justifica el éxito de las comunidades de software de código abierto a partir de los procesos de participación.
- La identificación de las dimensiones relacionadas con los procesos de cosificación en comunidades de software de código abierto.
- La definición de las características fundamentales por las que deben regirse las comunidades virtuales para alcanzar el éxito, de modo que sirvan a empresas interesadas en fomentar el desarrollo de líneas de producto basadas en software de código abierto

Para conseguir estos objetivos y estos resultados, el trabajo se ha estructurado en los siguientes capítulos:

- El capítulo 1 está dedicado al concepto de conocimiento y las distintas teorías sobre su gestión. Se hace una mención especial a la dicotomía de conocimiento tácito y explícito de Nonaka y Takeuchi (1995), para concluir con la dualidad de conocimiento *soft* y conocimiento *hard* definida por Hildreth *et al.* (1999).

- El capítulo 2 se centra en las comunidades de práctica, que se introducen a partir de la Teoría del Aprendizaje Social, de Wenger (1998). Se definen sus características como herramientas de gestión del conocimiento y se introducen las comunidades de prácticas virtuales, capaces de atraer participantes de todo el mundo, superando todo tipo de barreras físicas al realizar un uso masivo de herramientas electrónicas.
- El capítulo 3 está dedicado a los proyectos de software de código abierto, que constituyen un paradigma novedoso y revolucionario en el ámbito de la Ingeniería del software. Este tipo de proyectos constituye un claro ejemplo de comunidades de prácticas virtuales en las que la realización del software se apoya en una comunidad de usuarios subyacente. El capítulo define las principales características de los proyectos de software de código abierto y las tipologías y motivaciones de los usuarios que contribuyen a su desarrollo.
- El capítulo 4 está dedicado a los procesos de participación, mediante los cuales se desarrolla el conocimiento *soft*, y a los procesos de cosificación, mediante los cuales se desarrolla el conocimiento *hard*. Se introduce la representación de las comunidades como redes sociales y se definen las hipótesis y el modelo de participación propuesto.
- El capítulo 5 se centra en la metodología mediante la cual se desarrollará el estudio empírico. En particular, el modelo de participación se obtendrá a partir de una serie de indicadores obtenidos de las comunidades representadas como redes sociales y utilizando Modelos de Ecuaciones Estructurales como PLS. La identificación de las dimensiones relacionadas con los procesos de cosificación se realizará mediante Técnicas de Análisis Semántico y aplicando el Análisis Factorial Exploratorio.
- El capítulo 6 contiene la parte empírica, resultado de aplicar la metodología descrita a las comunidades Debian-Linux para sistemas embebidos. Para el modelo de participación se define en primer lugar una serie de indicadores, aplicando medidas típicas del análisis de redes sociales sobre las comunidades modeladas como grafos interconectados. El modelo estructural y de medida se obtiene aplicando PLS y, finalmente, se analiza su validez y fiabilidad. Para el modelo de cosificación se obtiene una serie de indicadores aplicando técnicas de análisis semántico sobre el contenido de

los mensajes de las distintas comunidades Debian-Linux y se identifican sus principales dimensiones mediante la técnica del análisis factorial.

- Por último, el capítulo 7 recopila las principales conclusiones del trabajo y sus implicaciones para la industria del software, así como sus limitaciones y futuras líneas.

Capítulo 1. Gestión del conocimiento

La propia historia del conocimiento nos sugiere la importancia de asumir conciencia del juego que jugamos al definir lo que es conocer. El hecho de definir es, en sí mismo, un acto de conocimiento, una operación para la que se han encontrando reglas de sintaxis. No es nuestro interés el emprender aquí el esfuerzo de la definición de conocimiento como un sistema formal, sino meramente el de identificar elementos significativos en un proceso de definición. Baste decir que, en tal sintaxis, hay elementos lógicos, elementos semánticos y elementos (meta)sistémicos que hay que discernir y formalizar. En la medida que lo logremos, contaremos con contribuciones diversas y complementarias (las definiciones) relativas a sistemas específicos, antes que agobiarnos por la fatua búsqueda de "La Definición". Éstas, sin embargo, pueden ser la base para una convención acerca de lo que entendemos por Conocimiento y Gestión del Conocimiento en un grupo determinado, es decir, para nuestra definición.

Para ser la base de una Teoría de la Empresa, el conocimiento debe ser definido de forma lo suficientemente precisa, de manera que nos permita ver qué empresa tiene el conocimiento más significativo y explique cómo se consigue esa ventaja competitiva (Spender, 1996a)

1.1 Concepto de conocimiento

El conocimiento es algo abstracto, difícil, aunque no imposible de definir. Muchas han sido las definiciones dadas al término "conocimiento", según el punto de vista con el que ha sido tratado.

Para comenzar, la Real Academia Española, en su Diccionario de la Lengua Española, nos define el conocimiento como: m. 1. Acción y efecto de conocer. || 2. Entendimiento, inteligencia, razón natural. || 3. Conocido, persona con quien se tiene algún trato, pero no amistad. || 4. Cada una de las facultades sensoriales del hombre en la medida en que están activas. Perder, recobrar el conocimiento. || 5. (desusado) Papel firmado en que se confiesa haber recibido de otro alguna cosa, y se obliga a pagarla o devolverla. || 6.

Reconocimiento, gratitud. || 7. (Comercio) Documento que da el capitán de un buque mercante, en que declara tener embarcadas en él ciertas mercaderías que entregará a la persona y en el puerto designado por el remitente. || 8. (Comercio) Documento o firma que exige o se da para identificar la persona del que pretende cobrar una letra de cambio, cheque, etc, cuando el pagador no le conoce. || 9. (plural) Noción, ciencia, sabiduría || venir en conocimiento de una cosa. (francés) llegar a enterarse de ella.

A nosotros nos interesan sólo algunas de las acepciones dadas al término. Más concretamente, las dos primeras: "Acción y efecto de conocer", es decir, de averiguar por el ejercicio de las facultades intelectuales la naturaleza, cualidades y relaciones de las cosas; y "Entendimiento, inteligencia, razón natural", es decir, capacidad de comprender o concebir las cosas, compararlas, juzgarlas, e inducir y deducir otras de las que ya conoce

Desde el punto de vista filosófico, preguntas como: qué es el conocimiento, en qué se funda el conocimiento, cómo es posible el conocimiento, etc. pertenecen a una disciplina filosófica llamada de varios modos: "teoría del conocimiento", "crítica del conocimiento", "gnoseología", "epistemología".

A continuación se considerarán varios aspectos ya clásicos en teoría del conocimiento: descripción o fenomenología del conocimiento; posibilidad del conocimiento; fundamentos del conocimiento; formas posibles de conocimiento.

En el sentido muy amplio de "pura descripción de lo que aparece o de lo que es inmediatamente dado", la fenomenología del conocimiento sugiere poner de manifiesto el "fenómeno" o el "proceso" del conocer. Se ha intentado hacer esto independientemente de, y previamente a, cualesquiera interpretaciones del conocimiento y cualesquiera explicaciones que puedan darse de las causas del conocer. Por tanto, la fenomenología del conocimiento no es una descripción genética y de hecho, sino "pura". Lo único que tal fenomenología aspira a poner en claro es lo que significa ser objeto de conocimiento, ser sujeto cognoscente, aprehender el objeto, etc.

Un resultado de tal fenomenología parece obvio: conocer es lo que tiene lugar cuando un sujeto (llamado "cognoscente") aprehende un objeto (llamado "objeto de conocimiento" y, para abreviar, simplemente "objeto"). Sin embargo, el resultado no es ni obvio ni tampoco

simple. Por lo pronto, la pura descripción del conocimiento o, si se quiere, del conocer, pone de relieve la indispensable co-existencia, co-presencia y, en cierto modo, co-operación, de dos elementos que no son admitidos, al menos con el mismo grado de necesidad, por todas las filosofías. Algunas filosofías insisten en el primado del objeto (realismo en general); otras, en el primado del sujeto (idealismo en general); otras, en la equiparación "neutral" de sujeto y objeto. La fenomenología del conocimiento no reduce ni tampoco equipara; reconoce la necesidad del sujeto y del objeto sin precisar en qué consiste cada uno de ellos, es decir, sin detenerse en averiguar la naturaleza de cada uno de ellos o de cualquier supuesta realidad previa a ellos o consistente en la fusión de ellos.

Conocer es, pues, fenomenológicamente hablando, "aprehender", es decir, el acto por el cual un sujeto aprehende un objeto. El objeto debe ser, pues, por lo menos gnoseológicamente, trascendente al sujeto, pues de lo contrario no habría "aprehensión" de algo exterior; el sujeto se "aprehendería" de algún modo a sí mismo. Decir que el objeto es trascendente al sujeto no significa todavía, sin embargo, decir que hay una realidad independiente de todo sujeto. La fenomenología del conocimiento, decíamos, no adopta por lo pronto ninguna posición idealista, pero tampoco realista. Al aprehender el objeto, éste está de alguna manera en el sujeto. No está en él, sin embargo, ni física ni metafísicamente. Está en él sólo representativamente. Por eso decir que el sujeto aprehende el objeto equivale a decir que lo representa. Cuando lo representa tal como el objeto es, el sujeto tiene un conocimiento verdadero (si bien posiblemente parcial) del objeto; cuando no lo representa tal como es, el sujeto tiene un conocimiento falso del objeto.

El sujeto y el objeto del que aquí se habla son, pues, "el sujeto gnoseológico" y el "objeto gnoseológico", no los sujetos y objetos "reales", "físicos" o "metafísicos". Por eso el tema de la fenomenología del conocimiento es la descripción del acto cognoscitivo como acto de conocimiento válido, no la explicación genética de dicho acto o su interpretación metafísica.

Sin embargo, aunque la fenomenología del conocimiento aspira a "poner entre paréntesis" la mayor parte de los problemas del conocimiento, ya dentro de ella surgen algunos que no pueden ser ni solucionados, ni siquiera aclarados por medio de una pura descripción. Por lo pronto, está el problema del significado de 'aprehender'. Se puede "aprehender" de muy

diversas maneras un objeto. Así, por ejemplo, hay una cierta aprehensión (y aprehensión cognoscitiva) de un objeto cuando se procede a usarlo para ciertos fines. No puede descartarse sin más este aspecto de la aprehensión de objetos por cuanto un estudio a fondo del conocimiento requiere tener en cuenta muy diversos modos de "capturar" objetos. Sin embargo, es característico de la fenomenología del conocimiento el limitarse a destacar la aprehensión como fundamento de un enunciar o decir algo acerca del objeto. Por este motivo, la aprehensión de la que aquí se habla es una representación que proporciona el fundamento para enunciados.

En segundo lugar, está el problema de cuál es la naturaleza de "lo aprehendido" o del objeto en cuanto aprehendido. No puede ser el objeto como tal objeto, pero entonces hay que admitir que el objeto se desdobla en dos: el objeto mismo en cuanto tal y el objeto en cuanto representado o representable. La clásica doctrina de las "especies" —especies sensibles, especies intelectuales— constituyó un esfuerzo con vista a dilucidar el problema del objeto en cuanto representado o representable. Han sido asimismo esfuerzos en esta dirección las diversas teorías gnoseológicas (y a menudo psicológicas y hasta metafísicas) acerca de la naturaleza de las "ideas" —teorías desarrolladas por la mayor parte de autores racionalistas y empiristas modernos—. También han sido esfuerzos en esta dirección los intentos de concebir la aprehensión representativa del objeto desde el punto de vista causal (como ha sucedido en las llamadas "teorías causales de la percepción").

Finalmente, está el problema de la proporción de elementos sensibles, intelectuales, emotivos, etc., en la representación de los objetos por el sujeto. De acuerdo con los elementos que se supongan predominar se proponen muy diversas teorías del conocimiento. Puede verse, pues, que tan pronto como se va un poco lejos en la fenomenología del conocimiento se suscitan cuestiones que podrían llamarse "meta fenomenológicas".

"Posibilidad del conocimiento": A la pregunta de si es posible el conocimiento se han dado respuestas radicales. Una es el escepticismo, según el cual el conocimiento no es posible. Ello parece ser una contradicción, pues se afirma que se conoce algo y, al mismo tiempo, que nada es cognoscible. Sin embargo, el escepticismo es a menudo una "actitud" en la

cual no se formulan proposiciones, sino que se establecen, por así decirlo, "reglas de conducta intelectual". Otra es el dogmatismo, según el cual el conocimiento es posible. Más aún, las cosas se conocen tal como se ofrecen al sujeto.

Las respuestas radicales no son las más frecuentes en la historia de la teoría del conocimiento. Lo más común es adoptar variantes del escepticismo o del dogmatismo, por ejemplo, un escepticismo moderado o un dogmatismo moderado, que muchas veces coinciden. En efecto, en las formas moderadas de escepticismo o de dogmatismo se suele afirmar que el conocimiento es posible, pero no de un modo absoluto, sino sólo relativamente. Los escépticos moderados suelen mantener que hay límites en el conocimiento. Los dogmáticos moderados suelen sostener que el conocimiento es posible, pero sólo dentro de ciertos supuestos. Tanto los límites como los supuestos se determinan por medio de una previa "reflexión crítica" sobre el conocimiento. Los escépticos moderados usan con frecuencia un lenguaje psicológico o, en todo caso, tienden a examinar las condiciones "concretas" del conocimiento. Así, por ejemplo, los límites de los que se habla son límites dados por la estructura psicológica del sujeto cognoscente, por las ilusiones de los sentidos, la influencia de los temperamentos, los modos de pensar debidos a la época o a las condiciones sociales, etc. Cuando lo que resulta es sólo un conocimiento probable, el escepticismo moderado adopta la tesis llamada "probabilismo". Los dogmáticos moderados, en cambio, usan un lenguaje predominantemente "crítico-racional"; lo que tratan de averiguar no son los límites "abstractos", es decir, los límites establecidos por supuestos, finalidades, etc. Es fácil ver que mientras los escépticos moderados se ocupan predominantemente de la cuestión del origen del conocimiento, los dogmáticos moderados se interesan especialmente por el problema de la validez del conocimiento.

Los autores que no se han adherido ni al escepticismo ni al dogmatismo radical y que, por otro lado, no se han contentado con adoptar una posición moderada, estimada como "meramente ecléctica", han intentado descubrir un fundamento para el conocimiento que fuese independiente de cualesquiera límites, supuestos, etc. Tal ocurrió con Descartes, al proponer el *Cogito, ergo sum* y con Kant, al establecer lo que puede llamarse el "plano trascendental". En el primer caso, conocer es partir de una proposición evidente (que es a

la vez resultado de una intuición básica). En el segundo caso, conocer es sobre todo "constituir", es decir, constituir el objeto en cuanto objeto de conocimiento; nuestro conocimiento se construye a partir de las impresiones de nuestros sentidos y no puede, por tanto, decirnos nada sobre una realidad más allá de estas impresiones. Kant cree que nuestra experiencia se forma por la realidad, nuestro conocimiento de ésta se basa en las intuiciones a priori y consecuentemente están delimitadas por las categorías disponibles de la comprensión humana (Spender, 1996a).

Una vez admitido que el conocimiento (total o parcial, ilimitado o limitado, incondicionado o condicionado, etc.) es posible, queda todavía el problema de los fundamentos de tal posibilidad.

Algunos autores han sostenido que el fundamento de la posibilidad del conocimiento es siempre "la realidad" (o, como a veces se dice, "las cosas mismas"). Sin embargo, la expresión "la realidad" no es en modo alguno unívoca. Por lo pronto, se ha hablado de "realidad sensible", a diferencia de una, efectiva o supuesta, "realidad inteligible". No es lo mismo decir que el fundamento del conocimiento se halla en la realidad sensible (en las impresiones, percepciones sensibles, etc.), como han hecho muchos empiristas, que decir que tal fundamento se halla en la realidad inteligible (en las "ideas", en sentido más o menos platónico), como han hecho muchos racionalistas (especialmente los que han sido al mismo tiempo "realistas" en la teoría de los universales). Por otro lado, aun adoptándose una posición empirista o racionalista al respecto, hay muchas maneras de presentar, elaborar o defender la correspondiente posición. Así, por ejemplo, el empirismo llamado a menudo "radical" propone que no sólo el conocimiento de la realidad sensible está fundado en impresiones, sino que lo está también el conocimiento de realidades (o cuasi-realidades) no sensibles, tales como los números, figuras geométricas y, en general, todas las "ideas" y todas las "abstracciones". Pero el empirismo "radical" no es ni mucho menos la única forma aceptada, o aceptable, de empirismo. Puede adoptarse un empirismo a veces llamado "moderado" (que a menudo coincide con el racionalismo también llamado "moderado", tal como sucede, por ejemplo en Locke), según el cual el fundamento del conocimiento se halla en las impresiones sensibles, pero éstas sólo proporcionan la base primaria del conocer —una base sobre la cual se montan las ideas generales—. Puede adoptarse un

empirismo que a veces se ha llamado "total": es el empirismo que rehúsa atenerse a las impresiones porque son sólo una parte, y no la más importante, de la "experiencia". La "experiencia" no es únicamente para este empirismo experiencia sensible; puede ser también experiencia "intelectual", o experiencia "histórica", o experiencia "interior", o todas ellas a un tiempo. Puede adoptarse asimismo un empirismo que no deriva de las impresiones sensibles el conocimiento de las estructuras lógicas y matemáticas justamente porque estima que tales estructuras no son ni empíricas ni tampoco racionales. Son estructuras puramente formales, sin contenido. Tal ocurre con Hume y diversas formas de positivismo lógico. Puede abrazarse también un empirismo que parte del material dado a las impresiones sensibles, pero admite la posibilidad de abstraer de ellas "formas"; es el empirismo de sesgo aristotélico y los derivados del mismo. En cuanto al llamado grosso modo "racionalismo", ha adoptado asimismo muy diversas formas de acuerdo con el significado que se haya dado a expresiones tales como 'realidad inteligible', 'ideas', 'formas', 'razones', etc. No es lo mismo, en efecto, un racionalismo que parte de lo inteligible como tal para considerar lo sensible como reflejo de lo inteligible, que un racionalismo para el cual el conocimiento se funda en la razón, pero en donde ésta no es una realidad inteligible, sino un conjunto de supuestos o "evidencias", una serie de "verdades eternas", etc.

Las posiciones empiristas y racionalistas, y sus múltiples variantes, son sólo dos de las posiciones fundamentales adoptadas en la cuestión del fundamento del conocimiento. Otras dos posiciones capitales son las conocidas con los nombres de "realismo" e "idealismo". Indiquemos aquí únicamente que lo característico de cada una de estas posiciones es la insistencia respectiva en tomar un punto de partida en el "objeto" o en el "sujeto". Aun así, no es fácil esclarecer el significado propio de realismo y de idealismo en virtud de los muchos sentidos que adquieren dentro de estas posiciones los términos objeto y sujeto. Así, por ejemplo, en lo que toca al "sujeto", la naturaleza de la posición adoptada depende en gran parte de si el sujeto en cuestión es entendido como sujeto psicológico, como sujeto trascendental en el sentido kantiano, como sujeto metafísico, etc. En algunos casos, el partir del sujeto puede dar lugar a un subjetivismo. Pero en otros casos el término sujeto designa más bien una serie de condiciones del conocimiento como tal, que no son

precisamente “subjetivas”. Por eso, cuando se habla, por ejemplo, de idealismo, no es lo mismo entenderlo en sentido subjetivista u objetivista, crítico, lógico, etc. En otros casos, el partir del objeto puede dar lugar a lo que se llama “realismo fotográfico”, pero en muchas ocasiones el admitir que el fundamento del conocimiento se halla en el objeto no equivale a hacer del sujeto un mero “reflejo” del objeto.

No todas las actitudes adoptadas en el problema que nos ocupa pueden clasificarse en posiciones como las reseñadas. En rigor, todas estas posiciones tienen en común el dar de algún modo el conocimiento por supuesto. Además, casi todas tienden a concebir el conocimiento no sólo como una actividad intelectual, sino también como una actividad fundada en motivos intelectuales, aislados, o aislables, con respecto a cualesquiera otros motivos. En cambio, ciertas posiciones, especialmente desarrolladas en la época contemporánea pero precedidas por algunos autores (entre los cuales cabe mencionar a Nietzsche y a Dilthey), han intentado preguntarse por el fundamento del conocimiento en función de una más amplia “experiencia”. Como resultado de ello la teoría del conocimiento no ha consistido ya en una “filosofía de la conciencia” como “conciencia cognoscente”. Ejemplos de estos intentos los tenemos en varios autores: pragmatistas (Dewey, James); existencialistas (Sartre) y otros no fácilmente clasificables, como Ortega y Gasset, Heidegger, Gilles-Gaston Granger, etc. Nos limitaremos a subrayar aquí la doctrina de Ortega en la cual el conocimiento es examinado como un saber: el “saber a qué atenerse”. Se niega con ello que el conocimiento sea connatural y consustancial al hombre, es decir, que el hombre sea últimamente “un ser pensante”. Esto no equivale a defender una teoría “irracionalista” del conocimiento; equivale a no dar el conocimiento por supuesto y a preguntarse del modo como “se funda” (Ferrater, 1985).

A nosotros nos va a interesar para nuestra definición de conocimiento la perspectiva de Kant, al considerar “que nuestra experiencia se forma por la realidad y nuestro conocimiento de ésta se basa en las intuiciones a priori y consecuentemente están delimitadas por las categorías disponibles de la comprensión humana”. Es por ello que en el presente trabajo se adoptará un punto de vista empirista, es decir, aquél que opina que el fundamento del conocimiento se halla en las impresiones sensibles, en la experiencia,

refiriéndose a la experiencia "sensible", o experiencia "intelectual", o experiencia "histórica", o experiencia "interior", o todas ellas a un tiempo.

Desde la perspectiva psicológica (Sánchez, 1983), el conocimiento se concibe como un proceso, que recibe el nombre de cognición o proceso cognitivo, que es todo aquél que transforma el material sensible que recibe del entorno, codificándolo, almacenándolo y recuperándolo en posteriores comportamientos adaptativos.

Las principales formas de actividad en que se realiza el conocimiento son la percepción, la imaginación, la memoria y el pensamiento.

La corriente psicológica que ha estudiado actualmente con más profundidad el conocimiento se denomina psicología cognitiva, que se interesa fundamentalmente por los procesos humanos y constituye un intento de integración entre la psicología de la forma y el conductismo.

Tres son las características principales de esta concepción señaladas por el profesor H. Carpintero:

1 Representa la recuperación del plano de la experiencia individual, inmediatamente vivida por el sujeto.

2 Representa, por otra parte, una renovación del paradigma Estímulo (E) - Organismo (O) - Respuesta (R).

3 Restablece la consideración del organismo como una realidad activa, es decir, como un organismo capaz de procesar la información que recibe, orientando así al sujeto hacia un determinado tipo de conducta.

En el desarrollo de esta corriente han influido profundamente los estudios de cibernética, las teorías de N. Chomsky sobre la psicolingüística y las aportaciones de J. Piaget con su psicología genética.

A este respecto, Piaget define el conocimiento como una relación entre los objetos y el sujeto, interviniendo en él elementos diversos como los puramente biológicos, adaptativos, elementos de tipo lógico-formal, que entrañan funciones psíquicas cognitivas.

Piaget realiza el estudio del conocimiento válido desde el prisma de la denominada por él "epistemología genética". Frente a las posiciones empirista y racionalista Piaget propugna una tercera: la consideración "genética" del conocimiento, según la cual, éste se halla constantemente enlazado con acciones u operaciones. De ahí que lo más importante para él sea el estudio del desarrollo cognitivo.

Respuestas a preguntas como qué significa saber, cómo utilizan las personas su conocimiento, o cómo lo aprenden, influyen en la elección de qué enseñar, cómo deben ser organizadas las aulas y en las expectativas de las instituciones educativas. Desde la perspectiva pedagógica se entiende por conocimiento tanto el "saber" como el conjunto de los saberes que constituyen el curriculum de cada una de las ciencias (Sánchez, 1983). Basándonos en la perspectiva constructivista, los aprendices —o estudiantes— son los constructores de su propio conocimiento. Por tanto, la enseñanza de éste no debe ser vista como la colocación de información en las cabezas de los estudiantes, sino como el posibilitarles su construcción por ellos mismos.

Durante años, debido particularmente a la influencia de las interpretaciones de los "Piagetianos" sobre el desarrollo cognitivo, el constructivismo consideraba que no existía una enseñanza "didáctica". Por el contrario, lo que interesaba era crear un entorno en el que los estudiantes descubrieran o inventaran el conocimiento por ellos mismos. Hoy en día se sabe que la creación de este entorno es más complejo de lo que parece, consecuencia de los cambios derivados de la naturaleza de la experiencia y del aprendizaje.

Se ha descubierto que los grandes pensadores y solucionadores de problemas poseen grandes cantidades de conocimientos específicos (Glaser, 1984). Los expertos consideran que ese conocimiento produce habilidades y resultados eficientes. Sin embargo, parece que los educadores no pueden construir la experiencia haciendo memorizar a sus estudiantes el conocimiento de los expertos. Tal método de aprendizaje produce un conocimiento "inerte", que es improbable de ser utilizado en situaciones complejas. Es por ello que el conocimiento experto debe ser construido por cada individuo.

El conocimiento nuevo es muy dependiente de lo que ya se conoce. Las personas necesitan esquematizar las cosas para entender y retener nueva información (Resnick *et al*., 1996).

Cuanto más ricos y apropiados sean estos esquemas, más rápido y completamente se asimilarán las nuevas ideas.

Aprender bien pocas ideas y conceptos importantes tiene un mayor poder educativo que aprender un programa extenso pero con ideas y conceptos superficiales (Resnick *et al*., 1996). El problema está en identificar cuales son esos conceptos poderosos y generadores de conocimiento.

Una revisión metodológica y conceptual de la bibliografía existente sobre utilización del conocimiento centrada en la educación (Dunn y Holzner, 1982) llega a la conclusión de que las cuatro proposiciones siguientes aportan un marco integrador:

1 La utilización del conocimiento es interpretativa. Ello implica que los resultados del conocimiento potencialmente transferibles, ya se basen en la investigación o en la experimentación, "no hablan de sí mismos". Al contrario, son interpretados por los diversos protagonistas en términos de sus propios marcos de referencia.

2 La utilización del conocimiento está socialmente limitada. Los procesos interpretativos de la utilización del conocimiento están integrados en una estructura social y están limitados por las responsabilidades del rol, las interrelaciones y demás convenios institucionales, así como por las "racionalidades" que generan.

3 La utilización del conocimiento es sistemática. Los problemas derivados de la utilización del conocimiento rara vez pueden descomponerse en partes, dado que la utilización del conocimiento incluye de modo usual un conjunto de problemas en cuanto a la producción, organización, almacenamiento, recuperación, transferencia y utilización del conocimiento (Holzner y Marx, 1979).

4 La utilización del conocimiento es transaccional. No puede decirse realmente que el conocimiento sea "intercambiado", "comercializado" o "transferido", es decir, términos que sugieren un proceso unidireccional de mover discretas parcelas de información entre aquellos que comparten 'a priori' una definición común de "conocimiento". Por el contrario, el conocimiento se transfiere entre aquellas partes que están unidas en un acto de negociación, simbólico o comunicativo, sobre la

adecuación, la relevancia y la legitimidad de las demandas del conocimiento (Dunn, 1982).

El conocimiento puede ser socialmente distribuido, es decir compartido a través de varios individuos (Levine *et al*., 1993; Resnick *et al*., 1991). Destacan aquí dos aspectos importantes. En primer lugar, el aprendizaje vía interacción: al interactuar con otros —por ejemplo, para resolver problemas de matemáticas o dirigir una pieza compleja de una maquinaria o leer e interpretar un texto— es la base para ser capaz de resolver tareas por sí solo. Por tanto, una cuestión importante en el trabajo de un educador es diseñar cuidadosamente las interacciones que promuevan la internalización de las estrategias particulares, las formas de razonar y las actitudes conceptuales (Rogoff, 1990).

El otro aspecto importante es aprender a interactuar. Fuera de las aulas, la mayoría del trabajo intelectual se realiza interactuando directamente con otros. En estas situaciones —trabajo, vida cotidiana, dentro de la familia— la competencia cognitiva de la persona se juzga no sólo por lo que ella sabe, sino también por cómo de cuidadosamente utiliza este aprendizaje en la actividad conjunta con los demás. Una característica de este trabajo es la atención prestada a las diferencias culturales, influidas por las organizaciones e instituciones en las que las personas trabajan.

La estructura social de los sistemas de conocimiento está relacionada de manera compleja con la creación y la utilización del conocimiento, pero también queda limitada a la cultura moral de la sociedad y a su sentido de identidad (Robertson y Holzner, 1980). Así pues, los procesos de creación y utilización del conocimiento pueden contemplarse desde su interdependencia.

Los sistemas de conocimiento pueden analizarse en términos de funciones del conocimiento, dominios institucionales y estructuras de conocimiento, así como en términos de la situación céntrica o periférica de los componentes del sistema o "regiones". Las principales funciones del conocimiento pueden describirse bajo los cinco apartados siguientes:

1. Producción de conocimiento, por ejemplo, en la investigación científica y la escolaridad.

2. Organización y estructuración del conocimiento como, por ejemplo, en la elaboración de teorías, pero también en la elaboración de textos, currícula y similares.
3. Distribución del conocimiento, por ejemplo, a través de publicaciones o de agentes intermediarios
4. Almacenamiento del conocimiento en archivos así como en la memoria de los individuos y las colectividades.
5. Utilización del conocimiento a través de diversos tipos de relaciones de retroacción con cualquiera de las demás funciones.

Los principales sectores institucionales, como por ejemplo, la agricultura, la educación, la medicina u otras áreas de política nacional pueden desarrollar sistemas propios y especializados de conocimiento social. Las profesiones establecidas constituyen un buen ejemplo de ello (Freidson, 1970).

La utilización del conocimiento puede considerarse primordialmente como conceptual, definida y medida en términos de procesos mentales de diversa índole, y puede representarse y medirse en términos de conducta abierta.

El método predominante para la obtención de datos en los estudios sobre la utilización del conocimiento es el cuestionario de auto-administración. Es relativamente raro el empleo del análisis del contenido, la observación naturalista y la entrevista, mientras que muy pocos estudios son cualitativos en el sentido específico de que intenten captar los significados contextuales subyacentes, unidos al conocimiento y a su utilización. Los estudios sobre la utilización del conocimiento, siempre que puedan basarse firmemente en el empleo de cuestionarios cuya fiabilidad sea fácilmente evaluada, se apoyan con frecuencia en procedimientos con una fiabilidad y una validez desconocidas o no expresadas. Dado que los estudios sobre la utilización del conocimiento están estrechamente relacionados con la evaluación de propiedades cognoscitivas (subjetivas) de diversos tipos, la ausencia de información sobre la fiabilidad de los procedimientos y la validez de los constructos representa un serio problema sin resolver para la mayor parte de la investigación del área.

En la Teoría de la Organización, algunos autores consideran el conocimiento una función o una herramienta directiva para realizar una tarea en relación con el entorno; una herramienta al servicio del saber y no algo que, una vez que se posee, es suficiente para llevar a cabo una acción o una práctica. El conocimiento es algo que se utiliza en las acciones, pero no es una acción (Cook *et al*., 1999). Da forma, significado y disciplina nuestras interacciones con el mundo real. Los métodos, reglas, creencias y teorías que utiliza para ello son herramientas intelectuales, las cuales son diferentes de las herramientas físicas porque se basan en un contexto social, es decir, se necesita un contexto social para poder utilizarlas (Polanyi, 1966; Sveiby, 1994). No todo lo que sabemos como consecuencia de la interacción con el mundo descansa en nuestro conocimiento: algunas cosas también descansan en nuestras propias acciones. Existen distintas formas de conocimiento, cada una de las cuales es utilizada por el saber cuando el conocimiento es utilizado como una herramienta en la interacción con el mundo. El conocimiento es una mezcla fluida de la experiencia, valores, información contextual y visión experta que proporciona un marco teórico para evaluar e incorporar nuevas experiencias e información. Éste se origina y es aplicado en las mentes de los conocedores. En las organizaciones a menudo está embebido, no sólo en documentos y reposiciones, sino también en las rutinas, procesos, prácticas y normas organizativas (Prusak *et al*., 1998 en Viedma, 2001). Por tanto, los individuos y grupos utilizan el conocimiento en su interacción con las cosas y actividades del mundo social y físico (Cook *et al*., 1999).

El concepto de competencia obtiene en este momento una especial atención. Los teóricos de la organización distinguen la competencia como una característica organizativa, que Philip Selznick (1957; Sveiby, 1994) asimila al término "competencia distintiva", análogo al concepto de "ventaja competitiva" de una organización definida por Porter (1985). La competencia se define como la capacidad de las personas para actuar en distintas situaciones, o en otras organizaciones similares. Es una relación entre el individuo y las reglas del sistema social. Ello es posible gracias a que la persona tiene poder sobre su propio conocimiento, es decir, sobre el sistema de reglas que decide utilizar (Rolf, 1991). Incluye las habilidades técnicas y directivas, así como la organización del trabajo y el reparto de valor, que posibilitan la supervivencia de una organización, a las que algunos

autores (Prahalad y Hamel, 1990; Sveiby, 1994; Snow y Hrebiniak, 1980) denominan "competencia clave" (core competence), así como la educación, la experiencia, los valores y las habilidades sociales (Sveiby, 1998). Es la suma del know-how y la habilidad para reflexionar. Es el lazo de unión entre Conocimiento y Estrategia.

En gran medida, la competencia depende del entorno. Esto es particularmente cierto para los elementos vinculados a la experiencia y para la red social de la competencia. Si una persona se encuentra situada en un nuevo entorno, pierde la competencia (Sveiby, 2000). Sin embargo, también podría ocurrir que la competencia sea más valiosa en un nuevo entorno si dicha competencia es relevante y escasa.

La perspectiva constructivista considera la competencia como un concepto individual y ve la tradición de la competencia (y del conocimiento) entre individuos como un elemento clave en la "organización". En este contexto, la competencia no es algo que una organización "tiene", sino que es un concepto subordinado al Conocimiento o al Saber.

La perspectiva constructivista se basa en la idea de que el individuo construye la realidad, no la descubre. Por tanto, el individuo construye el conocimiento al ser un experimentador activo. No existe ninguna forma de transferir conocimiento, pues cada persona debe construir su propio conocimiento. Estructuras cognitivas que ayudan a las personas a construir su propia realidad son los conceptos, las reglas, los esquemas, las metáforas, etc. (Sveiby, 1994). El conocimiento corporativo comprende esos hechos, reglas, modelos y conceptos derivados de las decisiones del día a día tomadas en cualquier nivel de la organización (Taylor, 1996).

Otros autores (Winter, 1995; Szulanski, 1996; Spender *et al*., 1996) definen el conocimiento asimilándolo a un recurso más de la organización sujeto a problemas complejos para que cualquiera pueda apoderarse de él (Grant, 1996b) y plasmado en la calidad e intensidad del entendimiento sobre la estrategia de negocio, las tecnologías, los productos, una base de clientes particular o incluso del uso de nuevas técnicas de producción (Pfeffer *et al*., 1999; Khanna *et al*., 1998). El conocimiento —intuición, entendimiento y know-how práctico que todos poseemos— es el recurso fundamental que nos permite actuar de forma inteligente (Wiig, 1997). Penrose lo define como el proceso

experto de apalancar recursos, donde ese conocimiento está permanentemente embebido en la organización (Spender, 1996a). A lo largo del tiempo, el conocimiento ha sido transformado en otras manifestaciones —como libros, tecnologías, prácticas y tradiciones— dentro de las organizaciones de todo tipo y en la sociedad en general. Estas transformaciones dan lugar a una experiencia acumulada y, cuando se utiliza de forma apropiada, aumenta la eficacia. El conocimiento es uno, o incluso el principal factor que hace posible el comportamiento inteligente de las personas, las organizaciones y la sociedad, por lo que debe ser cultivado, preservado y utilizado lo más posible, tanto por los individuos como por las organizaciones. Es por ello que los procesos relacionados con el conocimiento —para crear, construir, recopilar, organizar, transformar, combinar, aplicar y salvaguardar el conocimiento— deben ser dirigidos de forma cuidadosa y explícita en todas las áreas afectadas (Wiig, 1998).

El conocimiento puede ser considerado como un "bien público", a diferencia de los otros factores que las empresas poseen y que son considerados como "bienes privados" (tierra, trabajo y capital), entendiendo como tal su infinita extensibilidad y que su uso por una persona no priva a otros del mismo. Sin embargo, esto no es del todo correcto, ya que el conocimiento relevante para la empresa se conceptualiza a través de las habilidades del trabajo o el capital intelectual, por lo que puede ser convertido en un bien privado (Spender, 1996a; Polanyi, 1966).

Al hablarse de conocimiento como un recurso, éste se podrá almacenar. El stock de conocimiento de un individuo consiste en:

1. las expectativas normativas relacionadas con el rol que ha de desempeñar;
2. las disposiciones formadas en el curso de pasadas socializaciones; y
3. el conocimiento local de circunstancias particulares de tiempo y espacio.

Una empresa puede tener mayor o menor control sobre las expectativas normativas, pero tendrá un control muy limitado sobre los otros dos. En algún momento del tiempo, el conocimiento de una empresa es el resultado indeterminado de los individuos que intentan dirigir las inevitables tensiones entre las expectativas normativas, las disposiciones y los contextos locales (Tsoukas, 1996).

Aún es raro encontrar ejemplos donde se considere el conocimiento como una materia prima (Al Subyani, en Amidon, 1999) en un proceso de producción, de manera que se pueda adquirir, desarrollar y vender. Un ejemplo lo encontramos en Wikström y Normann (1992; Sveiby, 1994), quienes ven la organización como un sistema de procesamiento de conocimiento. De forma similar, Chiarmonte (Amidon, 1999) opina que el conocimiento se comparte por toda la cadena de valor, utilizando a los miembros involucrados como fuente de conocimiento. Nuevo conocimiento es a menudo el resultado (o el producto) de la combinación de capacidades, por parte de una empresa, para generar nuevas aplicaciones a partir de los componentes de conocimiento existentes (Kogut y Zander, 1992; Van den Bosch *et al*., 1999).

Según Spender (1996a), una teoría de la empresa basada en el conocimiento puede producir perspectivas más allá de las teorías de función de producción y de teoría basada en los recursos de la empresa. Esta es una plataforma para una nueva visión de la empresa como un sistema de producción y aplicación de conocimiento dinámico, evolutivo, quasi-autónomo. Pero para construir una teoría de la empresa basada en el conocimiento debemos ir más allá de los conceptos de conocimiento que nuestro entrenamiento positivista nos ofrece. El conocimiento organizativo no se define de un modo positivista como un valor corporativo, sino que es un aspecto cualitativo del sistema de actividad diseñado por los directivos. Finalmente, conocer es ser capaz de tomar parte en el proceso que hace útil el conocimiento. Los cuatro heurísticos que le llevan a esta conclusión son:

1. Flexibilidad interpretativa
2. Frontera directiva
3. Identificación de las influencias institucionales
4. Distinción entre características sistémicas y componentes

Si no hay flexibilidad interpretativa el sistema de conocimiento es inactivo, asocial y puramente maquinal. Si el sistema es activo y evolutivo, debe existir una flexibilidad interpretativa que corresponda, por ejemplo, a la división del trabajo. El crecimiento puede ser acelerado dando pasos que aumenten la flexibilidad interpretativa, pero esto amenaza las fronteras del sistema desde dentro. La frontera directiva es innecesaria si el sistema es

inactivo porque dichas fronteras no son dinámicas. Tener una estrategia significa saber cuándo decir no a nuevas oportunidades, por lo que cada sistema de actividad requiere fronteras directivas, las cuales surgen, se hunden, reflotan, se adquieren, etc. para cambiar los compromisos de mercado de las empresas que pueden precipitar una flexibilidad interpretativa energizante. La empresa que es inactiva no es amenazada por influencias institucionales externas. Pero para la empresa activa, cada movimiento que tenga lugar en la frontera afecta a los otros más allá de las mismas. A menos que estas entidades externas y quasi-objetos sean identificados, la frontera del proceso directivo está incontrolada. Finalmente, existe una distinción entre los procesos y las características sistémicas y de componente de la empresa. La localización eficaz de los recursos requiere una identificación cuidadosa de los procesos de conocimiento interno. Pero su valor no puede estimarse hasta que su significado se haya establecido. Si favorecen a la contribución pública o privada de la empresa es crucial. Uno de los ejemplos más obvios es el proceso tecnológico de información de la empresa. Éste casi no se puede evaluar dentro de los criterios normales de inversión del capital de la empresa porque su contribución real es a las características públicas de la empresa. En contraste, una máquina de producción es esencialmente privada, privatizable y *outsourceable* (Spender, 1996a).

Los nuevos desarrollos en tecnología de ordenadores han dado lugar a las recientes teorías de la organización en las que el concepto del conocimiento se basa en la teoría de la información (Hammer y Champy, 1993; Dawidov y Malone, 1992; Sveiby, 1994). En esta línea, se define el conocimiento como el know-how útil o la información técnica, cultural o directiva de una organización (Appleyard, 1996; Levinson et al., 1995) cuya validez ha sido establecida a través de tests de exámenes o evidencias. En función del grado de estructuración a través del análisis, selección e interpretación, la información se convierte en hechos del conocimiento (Barabba y Zaltaman, 1990), permitiéndonos ser capaces de anticipar o incluso predecir algunos hechos (Umstätter, 1998; Bohn, 1994).

De este modo, el conocimiento puede distinguirse de la opinión, la especulación, las creencias u otros tipos de información no verificada. Esta definición incluye tanto el conocimiento codificado de los productos como los proyectos y documentos escritos, así como el conocimiento tácito no codificado en las rutinas (Liebeskink, 1996: 94). El

conocimiento es creado y organizado por muchos flujos de información (Nonaka, 1994) y se genera mediante un proceso mental adecuado (Dooley *et al*., 1998: 284), por lo que depende de la cognición humana y su conciencia. Es una combinación del contexto, memoria personal y procesos cognitivos (Skyrme, 1994; Davies, 1996).

La perspectiva cognitiva ha inspirado investigaciones sobre cómo los individuos adquieren conocimiento, aprenden y cómo el esquema cognitivo y las estructuras de valor funcionan como obstáculos o límites al aprendizaje. El enfoque está más en la toma de decisión. La intuición de la psicología cognitiva ha tenido mucha influencia en la teoría de la organización porque explica el comportamiento "irracional" en términos de las distintas percepciones individuales de la realidad. Los teóricos de la organización, desde 1950 y con la obra de Simon (1976) de la Racionalidad Limitada, han descrito a los tomadores de decisiones como buscadores de decisiones satisfactorias, más que óptimas. Muchas de las investigaciones de los teóricos de la organización sobre la toma de decisión la representan como un proceso confuso, desordenado (Mintzberg, 1980) e irracional (Brunsson, 1985), en el que las decisiones son difíciles de distinguir (Mintzberg y Pettigrew, 1990) o incluso se toman rara vez (March y Olsen, 1972). En esta rareza, los actores individuales nunca intentan actuar "racionalmente", por lo que el proceso debe ser descrito como un proceso incremental caracterizado por los intentos de tomar decisiones racionales (Quinn, 1980).

Dos líneas importantes de investigación de organizaciones, basadas en teorías sociológicas y teorías de psicología cognitiva, respectivamente, han intentado explicar el comportamiento del individuo. Teorías sobre cómo se construye la realidad a través de modelos mentales o esquemas de nuestra mente y cómo los individuos representan su entorno (Weick, 1979; Sveiby, 1994) han sido utilizadas para explicar las anomalías y el comportamiento irracional en las organizaciones. Tales metáforas se utilizan para investigar cómo las organizaciones cambian y "aprenden" reaccionando a las fuerzas del entorno y se mueven a través de niveles de cambio.

La ventaja de las teorías cognitivas es que ofrecen a los individuos (al menos a los altos directivos) el escenario y explican la irracionalidad en las organizaciones en términos de realidad percibida o construida de los individuos. El conocimiento es visto principalmente como algo del individuo y se formula en términos de reglas, valores y creencias. Sin

embargo hay problemas al tratar de explicar el comportamiento y cambio organizativo trasladando perspectivas basadas en el comportamiento y aprendizaje individual. Las teorías cognitivas tienden a concentrarse en el entorno interno o, como mucho, al nexo entre el entorno interno y externo, de manera que consideran el entorno externo como una variable independiente (Sveiby, 1994).

Otros autores definen el conocimiento a partir de sus componentes en función de la perspectiva que se adopte. De esta forma, desde la perspectiva del coste de transacción incluiría patentes, know-how técnico, experiencia financiera, personal directivo experimentado y acceso a los canales de marketing y distribución; desde la perspectiva del comportamiento estratégico comprendería el acceso al mercado y, en el contexto de los países desarrollados, relaciones privilegiadas con las agencias de gobierno; y, por último, desde la perspectiva del proceso de aprendizaje abarcaría las rutinas organizativas que no pueden ser transferidas eficientemente en el mercado, o no pueden ser especificadas en un acuerdo contractual, pero requieren una réplica de la organización misma (Shenkar *et al*., 1999: 135-136).

Por último, algunos autores definen el conocimiento como un proceso de creación mediante el cual se transforman los datos en información, al sugerirse lo que podrían significar para una persona dada y, en consecuencia, se crea conocimiento nuevo al relacionar la nueva información con la información previamente creada (Figura 1). De aquí que el conocimiento del directivo esté estrechamente relacionado con sus observaciones del entorno del negocio. No existe un mundo para ser representado, sino un punto de observación para seleccionar datos.

El entorno consiste en datos, no en información. El nuevo conocimiento sobre el entorno depende del conocimiento existente en la organización. Por tanto, el conocimiento depende de la historia.

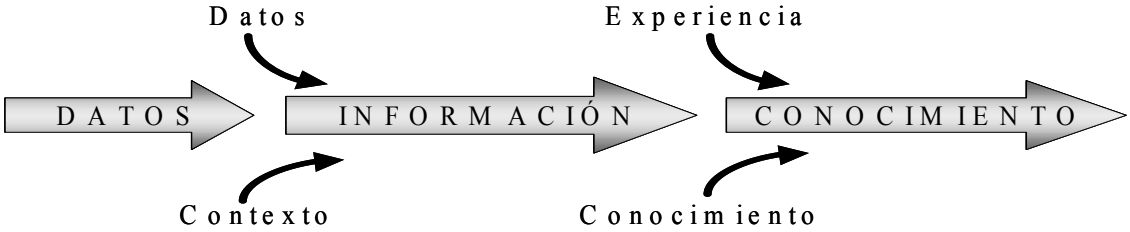
El conocimiento de la organización no se iguala a la información, sino que es más un resultado de combinar la experiencia previa con la nueva información. Este proceso podría obstaculizar la selección de nuevos datos y creación de información (von Krogh *et al*., 1996a). El valor del conocimiento crece cuando es compartido entre las personas como

consecuencia de las preguntas que se formulan, las cuales dan lugar a modificaciones sobre el concepto inicial que añaden más valor (Cabrera, 1999). La creación de conocimiento es el resultado de un doble proceso de combinación e intercambio. La creación gradual de conocimiento —o "single-loop learning" (Argyris *et al*., 1978) — requiere la combinación de piezas de conocimiento previamente desconectadas, mientras que las innovaciones radicales —"double-loop learning" o "paradigm shifts" (Kuhn, 1970) — se basan en la distinción de conceptos nuevos o caminos nuevos de combinar elementos que deberían haber sido asociados. En ambos casos, el intercambio de conocimiento es un requisito para la combinación y la creación de conocimiento colectivo (Nahapiet *et al.*, 1998; Cabrera, 1999: 8).

El conocimiento organizativo es el resultado de la historia particular de las interacciones internas y la adaptación externa experimentada por cada organización a medida que ésta participa de acciones, es decir, representa el punto que ha sido alcanzado por los procesos de aprendizaje de la organización en cualquier tiempo y es intrínsecamente valioso para su actividad. El conocimiento organizativo es "raro" (único) porque no hay dos organizaciones que experimenten exactamente la misma historia de experiencias de aprendizaje. Es "difícil de apropiar" por terceras partes debido a su carácter supra-individual (se tendría que alquilar la plantilla completa de la empresa para extraer ciertos tipos de conocimiento distribuido) y porque éste se ha construido a raíz de capacidades co-especializadas, es decir, capacidades cuyo valor depende de la presencia de otras capacidades organizativas específicas. Finalmente, el conocimiento organizativo es "difícil de imitar" debido a que a menudo se encuentra embebido en una red compleja de relaciones interpersonales formales e informales y en sistemas de normas y creencias compartidos y a menudo no hablados. Irónicamente, las mismas razones que hacen el conocimiento organizativo tan escurridizo para que los competidores lo imiten, también es escurridizo para controlarlo y dirigirlo sistemáticamente (Cabrera, 1999; von Krogh *et al*., 1996b).

Teniendo en cuenta todo lo anterior ya estamos en disposición de adoptar una definición para la palabra conocimiento. En el presente trabajo se va a considerar el conocimiento como un recurso poco tangible, resultante de un proceso de creación mediante el cual se

realiza un análisis y un seguimiento de las observaciones del entorno para averiguar la naturaleza, las cualidades y las relaciones de los entes, obteniéndose a partir de ellos unos datos, que se transforman en información, al sugerirse lo que podrían significar para una persona dada, y ésta, a su vez, en conocimiento, tras relacionar la nueva información con la información previamente creada a través de la experiencia (von Krogh *et al*., 1996). Nuestra experiencia se forma por la realidad y nuestro conocimiento de ésta se basa en las intuiciones a priori y, consecuentemente, están delimitadas por las categorías disponibles de la comprensión humana (Kant). Esto hace que el conocimiento tenga naturaleza predictiva, al proporcionar las bases para la predicción del futuro con un cierto grado de certeza, basada en la información sobre el pasado y el presente, permitiéndonos, además, realizar asociaciones causales o prescribir decisiones sobre qué hacer (Bohn, 1994).



Fuente: Earl & Scott, 1998

Figura 1. Creación de conocimiento.

1.2 Términos relacionados con el conocimiento

A veces se utilizan como sinónimos del conocimiento palabras como datos, información, competencia, etc. pero realmente no lo son. Existen pequeños matices que las diferencian.

1. Dato es aquello que proviene directamente de los sentidos, derivado del nivel de medida de alguna variable (Bohn, 1994). Es un portador del conocimiento y de la información, un medio a través del cual el conocimiento y la información pueden ser almacenados y transferidos. Tanto la información como el conocimiento se comunican a través de datos y por medio de los datos se almacena y transfiere en mecanismos y sistemas. En este sentido, una pieza de datos sólo se convierte en información o

conocimiento cuando éstos son interpretados por un receptor. De igual forma, la información y el conocimiento sostenido por una persona sólo puede ser comunicada a otra persona después de que se haya codificado en datos (Kock et al., 1997). Los datos son explícitos, en tanto que se pueden escribir, almacenar, repartir o discutir fácilmente (Davies, 1996). El papel impreso y los discos de ordenador son ejemplos de datos almacenados en mecanismos. Un correo electrónico empresarial y los sistemas de correo aéreo internacionales son ejemplos de almacenamiento de datos y sistemas de transferencia. Una de las razones de la confusión entre datos e información puede ser la amplia e intuitiva suposición de que la información es el principal componente de lo que se comunica a través de los datos.

2. La información son datos que han sido organizados o estructurados, es decir, situados en un contexto y con un significado determinado (Joia, 2000). La información nos cuenta el estado actual y pasado de algunas partes del sistema de producción (Bohn, 1994; Soto, 1998). Es claro que la información como tal no es suficiente para responder a un problema: se requiere análisis, seguimiento y creatividad para sacar el mayor provecho de esta información. Este es precisamente el proceso que se refiere al conocimiento (Soto, 1998). Es descriptiva, es decir, relaciona el pasado y el presente (Kock *et al*., 1997). La información es un flujo de mensajes (Nonaka, 1994); se adquiere a través de la observación directa, (Dooley *et al*., 1998).

3. Por último, la competencia es la capacidad de las personas para actuar en distintas situaciones. Incluye destrezas, educación, experiencia, valores y habilidades sociales (Sveiby, 1998). Es la suma del know-how y la habilidad para reflexionar. Es una relación entre el individuo y las reglas del sistema social. La persona tiene poder sobre su propio conocimiento, es decir, sobre el sistema de reglas que decide utilizar (Rolf, 1991). Sveiby (2000) considera que "el conocimiento es la habilidad para actuar". Posteriormente define un concepto relacionado con el conocimiento: la competencia. La competencia amplía la definición de conocimiento para incluir justo eso: la habilidad para actuar. La competencia de una persona, según Sveiby, está compuesta por cinco elementos interdependientes:

i. Conocimiento explícito: supone el conocimiento de hecho. Se adquiere esencialmente por medio de la información, generalmente en el marco de una formación particular.

ii. Aptitud: es el know-how, el talento o el arte de "saber cómo hacer las cosas". Supone una capacidad efectiva —física e intelectual— y se adquiere esencialmente a través de la formación y de la práctica. La aptitud supone el conocimiento de las reglas de procedimiento y de las capacidades para comunicarse.

iii. Experiencia: se adquiere principalmente reflexionando sobre los errores y los éxitos pasados.

iv. Juicios de valor: son percepciones de lo que la persona piensa que es justo. Funcionan como filtros conscientes e inconscientes en el aprendizaje de cada persona.

v. Red social: está formada por las relaciones del individuo con otros individuos en un entorno y una cultura transmitida por tradición.

Esta lista muestra que la información (un conocimiento explícito) sólo es uno de los elementos de la competencia. A esta lista, Hjertzén *et al*. (1999) añaden la motivación. En gran medida, la competencia depende del entorno. Esto es particularmente cierto para los elementos vinculados a la experiencia y para la red social de la competencia. Si una persona se encuentra situada en un nuevo entorno, pierde la competencia (Sveiby, 2000). Sin embargo, también podría ocurrir que la competencia sea más valiosa en un nuevo entorno si dicha competencia es relevante y escasa.

1.3 Conocimiento tácito y explícito

Cada vez adquiere más importancia el tema del conocimiento. Hoy en día se habla constantemente sobre la necesidad de las empresas de conocer cuáles son sus bienes intangibles y saber cómo gestionarlos para poder sobrevivir en el tiempo.

Muchos son los autores que coinciden en clasificar el conocimiento de dos formas: conocimiento tácito y explícito (Polanyi: 1966; Nelson y Winter: 1982; Cohen y Levinthal:

1990; Kogut *et al*., 1992; Nonaka *et al*., 1995; Grant, 1996; Lane *et al*., 1998; Stewart, 1997; Soto, 1998; Cabrera, 1999; Cook y Brown: 1999; Pfeffer *et al*., 1999; Edvinsson, 2000; Sveiby, 2000). Estos autores ofrecen distintos matices a estos términos pero, en definitiva, todos lo definen de forma similar.

Gran cantidad del conocimiento detallado de las rutinas y objetivos de la organización que permite funcionar a una empresa y sus laboratorios de I+D es tácito (Nelson y Winter, 1982). El conocimiento tácito es el conocimiento que no es fácilmente descrito o codificado (Dietrich, 1994; Shenkar *et al*., 1999), pero es esencial para realizar el trabajo (Pfeffer *et al*., 1999). Es el componente necesario de todo conocimiento. Es difícil de precisar debido a sus interconexiones con otros aspectos de la empresa, tales como las rutinas organizativas de que forma parte, sus procesos y contexto social (Shenkar *et al*., 1999; Lane *et al*., 1998; Inkpen *et al*., 1998). No se fabrica de una simple haba que puede enterrarse, perderse o reconstituirse (Tsoukas, 1996). Se asocia con las habilidades o el "know-how" y hace referencia a aquél conocimiento que está oculto o es inaccesible para propósitos prácticos (Cook *et al*., 1999: 381; Cabrera, 1999). De ahí que en las organizaciones se adquiera a través de la experiencia y el uso (Cohen y Levinthal, 1990; Inkpen *et al*., 1998) y se manifieste a través de su aplicación (Grant, 1996; Dooley *et al*., 1998). Polanyi (1966) observó que el conocimiento tácito debía ser experimentado, por ejemplo, para identificar cuáles eran las mejores prácticas en una organización, aun cuando las razones para su eficacia no fueran completamente entendidas.

El conocimiento tácito, a veces también llamado "conocimiento informal" (Michael Polanyi, en Barclay *et al*., (1997), es difícil de formalizar y no es fácilmente visible, lo que lo hace difícil, lenta y costosa de comunicar o de compartir con otros (Kogut *et al*., 1992). Algunos autores señalan que se transfiere a través de la socialización y la adaptación mutua, un proceso que requiere las interacciones cara a cara (Cabrera, 1999). En las organizaciones, el conocimiento tácito incluye factores intangibles enraizados en el contexto social específico —una profesión, una tecnología particular o un producto mercado, o las actividades de un grupo o equipo de trabajo—, las creencias personales, experiencias del individuo, perspectivas y valores personales, haciéndolo único, menos imitable y más valioso (Spender 1996b; Lane *et al*., 1998; Inkpen, 1998; Nelson y Winter

1982). Cuando se pide a los individuos de las organizaciones que describan cómo y porqué las cosas se realizan de cierta forma, a menudo contestan "No estoy seguro; es simplemente la forma en que las cosas se hacen aquí". La inhabilidad para articular o describir un proceso organizativo indica que el conocimiento que sostiene el proceso es altamente tácito (Inkpen, 1998). Las personas poseen más conocimiento del que ellos pueden mostrar debido a la cantidad de conocimiento tácito que poseen. Muchos expertos son capaces de articular porciones de su conocimiento experto, pero no así la riqueza de la estructura cognitiva, que es la esencia de ese conocimiento experto (Polanyi, 1966).

Siempre ha sido visto el conocimiento tácito como la clave para hacer cosas y crear nuevo valor en los negocios. Es por eso por lo que siempre vemos en el "aprendizaje de la organización" el interés por internalizar la información (a través de la experiencia y la acción) y por generar nuevo conocimiento a través de interacciones dirigidas (Barclay *et al*., 1997). El saber cómo está estrechamente relacionado con el conocimiento tácito (Grant, 1996; Sveiby, 1994).

Por otra parte, el conocimiento de una empresa también incluye el conocimiento articulado en un lenguaje, fácilmente de comunicar entre los individuos (Ortigueira, 1991) a través de datos o procedimientos codificados y encarnado en productos y procesos específicos. Dicho conocimiento es el conocimiento explícito, o "conocimiento formal" (Nickols en Barclay *et al*., 1997), el cual puede ser formalizado o explicado en detalle (Nonaka *et al*., 1995; Spender, 1996c; Barclay *et al*., 1997; Lane *et al*., 1998; Inkpen, 1998; Inkpen *et al*., 1998; Cook *et al*., 1999). El conocimiento explícito se refiere a las formas bien articuladas de conocimiento que pueden ser comunicadas a los demás, a través de reglas, hechos, conceptos, marcos teóricos.

Se caracteriza por ser un conocimiento objetivo, teórico, digital, formal y sistemático, fácilmente comunicado o compartido mediante especificaciones del producto o a través de una fórmula científica o un programa informático (Nonaka *et al*, 1995; Grant, 1996; Inkpen, 1998). Tiende a evolucionar con la experiencia y se manifiesta a través de su comunicación, pudiendo ser transferido a través de diseños de ingeniería, extraído de patentes, etc. (Grant, 1996; Dooley *et al*., 1998; Hamel, 1991). Se identifica este

conocimiento con el saber sobre hechos y teorías (Grant, 1996; Spender, 1996a; Sveiby, 1994).

El conocimiento explícito por sí sólo no tiene ningún valor productivo. Sólo cuando es interiorizado por los individuos y colectivizado a través de la socialización y la adaptación mutua puede ofrecer resultados tangibles (Cabrera, 1999). A menudo existe una fuerte dimensión tácita asociada con el cómo se utiliza e implanta el conocimiento explícito (Inkpen, 1998).

El conocimiento explícito sufre dos problemas claves en su asignación: en primer lugar, como un bien público o sin competencia, nadie que lo adquiera puede venderlo sin perderlo (Arrow, 1984) (es decir, nadie se puede deshacer de él al venderlo); en segundo lugar, el mero acto de negociar con el conocimiento lo hace disponible a compradores potenciales (Arrow, 1971). Su coste de imitación por parte de otras empresas u organizaciones es bastante bajo (Spender 1996b; Lane *et al*., 1998). Así, excepto para las patentes y los copyright, donde la propiedad del conocimiento está protegida legalmente, el conocimiento no puede ser apropiable mediante transacciones de mercado (Grant, 1996).

Un detalle a tener en cuenta en las definiciones anteriores es que el conocimiento explícito, por definición, sería objetivo. Por tanto, habría que preguntarse si es "un pensamiento subjetivo puesto sobre el papel" una verdad objetiva. La respuesta es negativa. Pero las palabras pueden también tener un significado diferente. El conocimiento tácito es parte de una persona, un sujeto, mientras que el conocimiento explícito existe como un objeto, una forma visible.

Una vez definido el conocimiento tácito y el conocimiento explícito no debemos olvidar la polémica que existe en torno a estos términos en relación a si ambos tipos de conocimiento son las dos caras de una misma moneda o, por el contrario, son dos tipos de conocimiento distintos.

Por una parte, Nonaka y Takeuchi son partidarios de la idea de que el conocimiento explícito es la externalización o exteriorización del conocimiento tácito, el cual se encuentra en la base de todo proceso de aprendizaje y, por tanto, se podrían considerar como las dos caras de una moneda (1994), Figura 2. En la misma línea, como dice

Prigogine (1989) 'el orden y el desorden se crean de forma simultánea', así también el conocimiento tácito y explícito son mutuamente constituidos, por lo que podrían no ser vistos como dos tipos separados de conocimiento. Todo conocimiento articulado se basa en antecedentes no articulados, es decir, en un conjunto de particulares subsidiarios que están tácitamente integrados por individuos. Esos particulares residen en las prácticas sociales, nuestras formas de vida, dentro de las cuales nosotros participamos. La habilidad de una persona para seguir las reglas se basa en unos antecedentes no articulados. Existen reglas para el procesamiento de conocimiento consciente e inconsciente que nos ayudan a actuar y ahorrar mucha energía cuando no necesitamos pensar antes de actuar. Los antecedentes no articulados los aprendemos a través de la "socialización", los cuales no sólo son aprendidos sino también establecidos (Tsoukas, 1996). En consecuencia, el conocimiento tácito es el componente necesario de todo conocimiento. Todo conocimiento tiene una dimensión tácita y es por ella su dificultad para ser explicada mediante palabras. El hecho de que sabemos más de lo que podemos contar (Sveiby, 1994) nos muestra que no todo el conocimiento tácito puede hacerse explícito. El compartir conocimiento tácito es una forma limitada de crear conocimiento porque, a menos que el conocimiento tácito se haga explícito, éste no puede ser fácilmente influenciado por la organización como un todo. La creación de conocimiento organizativo es una interacción continua y dinámica entre el conocimiento tácito y el explícito (Nonaka *et al*., 1995). El conocimiento está orientado a la acción a través de la forma en que nosotros generamos nuevo conocimiento, analizando las impresiones sensoriales que percibimos y porque estamos constantemente perdiendo conocimiento (Taylor, 1993; Sveiby, 1994; Tsoukas, 1996). Por tanto, la interacción entre lo explícito y lo tácito es evolutiva. La frontera entre los tipos de conocimiento explícito y tácito es porosa y flexible, por lo que hay un tráfico entre sus dominios (Nelson *et al*. 1982; Spender, 1996a). En esta frontera porosa, Inkpen *et al*. (1998) situarían los distintos tipos de conocimiento.

Por otra parte, existen autores (Cook y Brown, 1999) que coinciden con los anteriores en la existencia del conocimiento tácito y explícito, pero opinan que ambas formas de conocimiento son distintas, aunque complementarias entre sí. Es decir, uno no es una variante del otro, ni viceversa ya que uno actúa o trabaja cuando el otro no puede y,

además, uno no puede convertirse en el otro. Sin embargo, cada forma de conocimiento puede ser utilizada como ayuda para adquirir la otra. Afirman que el conocimiento explícito no es la externalización del conocimiento tácito, sino un tipo de conocimiento completamente distinto. Además, la generación de conocimiento explícito puede ser necesaria para la difusión del conocimiento tácito (o incluso para hacer el conocimiento tácito más "fácilmente influenciable por la organización como un todo"). Sin embargo, esto viene determinado por su utilidad como una herramienta en la cuestión productiva en una situación dada, no por las características generales del conocimiento explícito y tácito, como Nonaka y Takeuchi sugieren. Si se necesita conocimiento explícito, éste conocimiento explícito necesita ser generado y compartido; si el conocimiento tácito es necesitado, entonces es éste el que debe ser generado y compartido. Por último, destacan que la producción de nuevo conocimiento no descansa en "una interacción continua entre el conocimiento tácito y el explícito" sino en nuestra interacción con el mundo. Más concretamente en la utilización del conocimiento (explícito y/o tácito) como herramientas para la cuestión productiva como parte de nuestra interacción dinámica con los entes del mundo social y físico.

Figura 2. Continuum de conocimiento.

Nosotros, sin embargo, opinamos que gran parte del conocimiento tácito se puede hacer explícito, como se señala en la primera de las perspectivas comentadas anteriormente, aunque no todo el conocimiento tácito se puede hacer explícito, como se indica en la otra perspectiva. Es decir, existiría un continuum de conocimiento en el que, en un extremo, se situaría el conocimiento tácito puro, al que equipararíamos a la intuición personal y que nunca se podría convertir en conocimiento explícito, y en el otro extremo colocaríamos el conocimiento explícito puro. Cualquier nivel intermedio de conocimiento situado entre

ambos extremos tendría una mayor o menor proporción de conocimiento tácito en función de su grado de conversión en conocimiento explícito.

1.4 Conocimiento soft y conocimiento hard

A diferencia de la dicotomía de conocimiento tácito y explícito, Hildreth *et al*, (1999) definen el conocimiento 'soft' y el conocimiento 'hard' como una dualidad. El conocimiento *hard* se refiere a los conocimientos más formalizables mientras que el conocimiento *soft* se refiere a aquél que resulta menos cuantificable y más difícil de capturar, codificar y almacenar. Es el conocimiento implícito que se encuentra en las experiencias y acciones cotidianas de las personas. Este conocimiento incluye el conocimiento tácito que no puede ser articulado, las habilidades automatizadas y las experiencias interiorizadas, y el conocimiento cultural (Hildreth *et al*., 1999). La consideración como dualidad más que como una dicotomía significa que ambos son igualmente importantes y que deben ser tenidos en cuenta al gestionar el conocimiento. Muchos de los sistemas de gestión del conocimiento fracasan porque se concentran únicamente en los aspectos *hard* del conocimiento. Capturar y almacenar conocimiento *hard* son procesos sencillos hoy en día con los medios electrónicos disponibles. Pero si esta labor no va acompañada de una gestión *soft* del conocimiento que permita vincular ese conocimiento explícito a la experiencia y la participación, acaba siendo inútil. El capítulo siguiente, dedicado a las comunidades de práctica, muestra como éstas tienen en cuenta los aspectos *soft* y *hard* del conocimiento. Los procesos subyacentes en el desarrollo del conocimiento *soft* y *hard* se denominan participación y cosificación, respectivamente. La participación permite la transferencia del conocimiento contextualizado a la experiencia y permite su socialización al resto de la comunidad. La cosificación, entendida como dar forma concreta y explícita a algo abstracto, se refiere a los mecanismos que permiten almacenar y retener el conocimiento. Siempre debe existir un balance adecuado entre ambos procesos, Figura 3. Si prevalece la participación, no existirá un material explícito sobre el que coordinar a las personas y explicitar las ideas. Si prevalece la cosificación, no

existirá el debate suficiente que permita negociar un significado y compartir el conocimiento y la experiencia.

Figura 3. Dualidad de los aspectos *soft* y *hard* del conocimiento.

Capítulo 2. Comunidades de Práctica

2.1 Introducción a la Teoría del Aprendizaje Social

Hasta hace no mucho tiempo existía la suposición de que el aprendizaje era un proceso individual, con un principio y un final, claramente separado de otras actividades, y que siempre era el resultado de una enseñanza (Wenger, 1998). Todavía hoy en día el aprendizaje tiene lugar mayoritariamente en aulas mediante el uso de clases magistrales. No obstante, poco a poco están surgiendo nuevas tendencias hacia un aprendizaje basado más en la práctica y que garantice un aprendizaje durante toda la vida. Ambos elementos constituyen los pilares de inicio de la Teoría del Aprendizaje Social, de Wenger (1998), teoría que está recibiendo una gran atención en los últimos años. A continuación se abordará dicha teoría, tomando como punto de partida las tres teorías básicas del aprendizaje.

2.1.1 Teorías del aprendizaje

Básicamente, las teorías del aprendizaje son tres: conductista, cognoscitivista y constructivista.

La Teoría Conductista se centra en el principio de estímulo-respuesta y en el refuerzo selectivo. El aprendizaje se considera el resultado de estímulos y respuestas mediante el uso de recompensas. Un área de contenido se descompone en una serie de componentes y habilidades que son secuenciadas y transmitidas al receptor del aprendizaje, frecuentemente mediante una instrucción directa. Una vez absorbidas las partes específicas que componen un área de contenido, el usuario del aprendizaje sería capaz de combinar cada una de las partes como un todo y aplicarlas cuando fuere necesario. Esta teoría ubica al receptor del aprendizaje como un sujeto pasivo, que necesita de motivación externa y un refuerzo (Chen, 2003)

La Teoría Cognoscitiva se centra directamente en la estructura y en la forma de operar de la mente humana. Esta teoría trata de entender el procesamiento de la información de la mente, que tiene que ver con la forma en la que las personas recopilan, almacenan,

modifican e interpretan la información de su entorno, y en cómo esta información es recuperada y utilizada en sus actividades (Chen, 2003).

La Teoría Constructivista enfoca el aprendizaje como un proceso de construcción de conocimiento. Incluye cuestiones tales como la motivación, el autoaprendizaje y un enfoque hacia un contexto social del aprendizaje. Esto significa que el aprendizaje no es un sencillo asunto de transmisión, internalización y acumulación de conocimientos, sino "un proceso activo" por parte del alumno que ensambla, extiende, restaura e interpreta, y por lo tanto "construye" conocimientos partiendo de su experiencia e integrándola con la información que recibe.

Chen (2003) describe dos aspectos esenciales del constructivismo. El primero de ellos consiste en considerar el aprendizaje como un proceso de construcción de conocimiento y no de absorción, que es el concepto dominante de la Teoría Conductista. Dado que el conocimiento se construye a partir de las percepciones y concepciones individuales del entorno, cada uno construye un concepto o significado diferente. Esto significa que un aprendizaje no puede ser transmitido con palabras, sino que sólo tiene lugar cuando el receptor de ese aprendizaje se encuentra activamente involucrado en el proceso. El segundo se refiere a que el conocimiento se encuentra íntimamente relacionado con el entorno en el que se realiza el aprendizaje y se construye ese conocimiento.

2.2.2 Teoría del aprendizaje social

La Teoría del Aprendizaje Social, de Wenger (1998), se basa en la Teoría Constructivista y sitúa el aprendizaje en el contexto de la experiencia social del individuo. Las suposiciones subyacentes de esta teoría son:

- Los seres humanos somos seres sociales y esta característica es esencial para el aprendizaje.
- El conocimiento es una cuestión de competencia en relación con ciertas empresas consideradas valiosas, como por ejemplo, escribir poesía, arreglar máquinas, ser cordial etc.

- Conocer es cuestión de participar en la consecución de estas empresas, es decir, de comprometerse de una manera activa en el mundo.
- El significado –nuestra capacidad de experimentar el mundo y nuestro compromiso con él como algo significativo– es, en última instancia, lo que debe producir el aprendizaje.

De acuerdo a la Teoría del Aprendizaje Social, el aprendizaje hay que situarlo en la práctica y en los grupos sociales en los que dicho aprendizaje tiene lugar, que se denominan comunidades de práctica. Desde la perspectiva teórica del aprendizaje social, se considera el aprendizaje como "una construcción de versiones presentes de experiencias pasadas por parte de personas diversas actuando conjuntamente en la práctica diaria", una actividad situada en un contexto que la dota de inteligibilidad, según la cual la descontextualización del aprendizaje es imposible, puesto que toda adquisición de conocimiento está contextualizada en algún tipo de actividad social. El aprendizaje implica participación en una comunidad, dejando de ser considerado como la adquisición de conocimiento por individuos para ser reconocido como un proceso de participación social en el que la naturaleza de la situación impacta significativamente.

Asimismo, una práctica puede definirse como "la forma en la que las tareas se realizan espontánea e improvisadamente, respondiendo a un entorno cambiante e impredecible, y conducida por un conocimiento tácito" (Seely Brown y Duguid, 2000). El concepto de comunidades de práctica (CPs) también ha sido estudiado por Seely Brown y Duguid (1991), quienes ven estas comunidades como grupos de personas en los que se lleva a cabo un trabajo, un aprendizaje y una innovación. No obstante, en este trabajo se seguirá el concepto de comunidad de práctica introducido por Wenger (1998). A continuación se introducirá este concepto de comunidad de práctica ubicándolo en el contexto de la gestión del conocimiento.

2.2 Concepto de Comunidad de Práctica

La suposición básica subyacente de la Teoría de las Comunidades de Práctica es que el aprendizaje implica participación en una comunidad, dejando de ser considerado como la

adquisición de conocimiento por individuos para ser reconocido como un proceso de participación social en el que la naturaleza de la situación impacta significativamente.

Según Wenger *et al*. (2002), una *comunidad de práctica* es un grupo de personas que comparten una preocupación, un conjunto de problemas o un interés común acerca de un tema, y que profundizan su conocimiento y pericia en esta área a través de una interacción continuada.

A Etienne Wenger se le puede atribuir el hecho de acuñar el concepto de comunidad de práctica, que utilizó junto a Jane Lave (1991) en el libro publicado *Situated learning. Legitimate peripheral participation*. En este trabajo se refleja la idea de que el aprendizaje implica participación en comunidad y que la adquisición de conocimientos se considera un proceso de carácter social.

A quienes también se les atribuye la paternidad del término –según algunos autores– es a John Seely Brown y Paul Duguid (1991), quienes en su artículo *Organizational learning and communities of practice* apuntan el término a través del estudio del caso de la empresa Xerox. Según estos autores, el análisis del trabajo cotidiano de los reparadores de fotocopiadoras de Xerox tuvo una vital importancia en la aparición del término *comunidad de práctica*.

Posteriormente a estas dos publicaciones han sido muchos los autores que se han atrevido a definir el concepto de comunidad de práctica (CP). Pero fue otra vez Wenger (1998) quien en su libro *Communities of practice: Learning, meaning and identity*, fijó las tres premisas o dimensiones –como él las denomina– en las que se asienta una comunidad de práctica: el compromiso mutuo, la empresa conjunta y el repertorio compartido:

> "Una comunidad de práctica se define a sí misma a lo largo de tres dimensiones: su empresa conjunta es comprendida y continuamente renegociada por sus miembros, el compromiso mutuo que une a sus miembros juntos en una entidad social y el repertorio compartido de recursos comunes (rutinas, sensibilidades, artefactos, vocabulario, estilos...) que los miembros han desarrollado a lo largo del tiempo", Wenger (1998)

Veamos una por una las tres dimensiones:

- Empresa conjunta. La CP debe tener unos objetivos y necesidades que cubrir comunes, aunque no homogéneos. Cada uno de los miembros de la CP puede comprender ese objetivo de una manera distinta, pero aun así compartirlo. Los intereses y las necesidades pueden ser distintos y, por tanto, negociados, pero deben suponer una fuente de coordinación y de estímulo para la CP.
- Compromiso mutuo. El hecho de que cada miembro de la CP comparta su propio conocimiento y reciba el de los otros tiene más valor que el poder que, en otros círculos más clásicos, parece adquirir el que lo sabe todo. El conocimiento parcial de cada uno de los individuos es lo que le da valor dentro de la CP.
- Repertorio compartido. Con el tiempo la CP va adquiriendo rutinas, palabras, herramientas, maneras de hacer, símbolos o conceptos que ésta ha producido o adoptado en el curso de su existencia y que han formado parte de su práctica.

Las CPs son diferentes del resto de los equipos de trabajo de las organizaciones por diferentes razones. Consideremos, para empezar, porqué lo son de los grupos de trabajo convencionales: en primer lugar, el grupo-equipo de trabajo lo crea el director del departamento o del área para llevar a cabo un proyecto específico y los miembros del equipo son seleccionados a partir de las aptitudes y experiencias que pueden aportar a éste. En cambio, según Wenger y Snyder (2000), las CPs son informales y se organizan ellas mismas, lo que no quiere decir que éstas sean equipos sin estructura: la tienen y se basa en establecer sus propias agendas y elegir a sus propios líderes. Pero sí que es cierto que son mucho más flexibles. Las CPs consiguen superar la lenta jerarquía tradicional, pero al mismo tiempo mantienen una forma organizativa más duradera –fuera de las fronteras estructurales tradicionales– que los cambios que pueda imponer la propia organización. Las CPs tienen una habilidad que los equipos de trabajo convencionales no tienen y es la de poder establecer conexiones con personal de otros departamentos dentro de la misma organización (Lesser y Stork, 2001).

La Tabla 1 muestra una comparativa de las CPs con otras formas de comunidad. La tabla, tomada del trabajo de Wenger y Snyder (2000), distingue diversos tipos de grupos atendiendo a su propósito, pertenencia, estructura social y duración.

Tabla 1. Comparación de las características de las Comunidades de Práctica con otras formas de organización

	¿Cuál es el propósito?	¿Quién pertenece?	¿Qué los mantiene juntos?	¿Cuánto dura?
Comunidad de práctica	Desarrollar las capacidades de los miembros, construir e intercambiar conocimiento	Miembros que se seleccionan a sí mismos	Pasión, compromiso e identificación con los grupos de experiencia	Mientras exista interés en mantener al grupo
Grupo formal de trabajo	Para entregar un producto o servicio	Cualquiera que reporte al grupo del gerente	Requerimientos del trabajo y metas comunes	Hasta la siguiente reorganización
Equipo de proyecto	Para lograr una tarea específica	Empleados asignados por la alta dirección	Los puntos importantes del proyecto y las metas	Hasta que el proyecto se complete
Red informal	Recolectar y pasar la información de los negocios	Amigos y negocios conocidos	Necesidades mutuas	Mientras las personas tengan una razón para seguir conectados

Fuente: Wenger y Snyder (2000)

2.2.1 Tipologías de participantes

Entre los participantes en una comunidad existen personas con una vinculación especial a ella, como son las figuras del moderador y del gestor del conocimiento.

El *moderador* es el encargado de animar y dinamizar el enriquecimiento mutuo y el intercambio de experiencias dentro de la comunidad. Este animador debe ser un miembro respetado de la CP. Es fundamental que sea alguien perteneciente a la CP pues sólo un participante puede apreciar las cuestiones importantes que están en juego en ella, lo que es importante compartir, las ideas emergentes y, sobre todo, las personas que forman la CP y

las relaciones que se crean y se pueden crear entre ellas (Sanz, 2005). Generalmente, no es el experto líder en su campo. Es importante que no se confundan los papeles, porque si el moderador es el líder puede provocar limitaciones en el número de intervenciones de los miembros del grupo. Así mismo, el moderador debe disponer de libertad para poder gestionar bien las intervenciones, distinguir las aportaciones interesantes, guardar los documentos adjuntos que se vayan presentando, realizar resúmenes periódicos, etc.

Según Wenger *et al*. (2002), el moderador o coordinador (como él lo denomina) tiene las siguientes funciones clave:

- Identificar temas importantes que deben tratarse en el ámbito de la CP
- Planificar y facilitar las actividades de la CP. Éste es el aspecto más visible del papel del moderador
- Conectar informalmente a los miembros de la CP, superando los límites entre las unidades organizativas, y gestionar los activos del conocimiento
- Potenciar el desarrollo de los miembros de la CP
- Gestionar la frontera entre la CP y la organización formal, como por ejemplo los equipos y otras unidades organizativas
- Ayudar a construir la práctica, incluyendo el conocimiento base, la experiencia adquirida, las mejores prácticas, las herramientas y los métodos, y las actividades de aprendizaje
- Valorar la salud de la CP y evaluar las contribuciones de los miembros a la organización

Otra figura esencial en toda CP es la del *gestor del conocimiento*, cuya misión primordial es la de nutrir el debate mediante la aportación de fuentes de conocimiento complementarias a las de los participantes. El gestor de conocimiento tiene que ocuparse de que la práctica progrese y de que las relaciones con la organización sean adecuadas, y ha de evaluar su contribución tanto a los participantes como a la organización, para que nunca se pierda la sintonía con los objetivos estratégicos.

Además de estas dos figuras, se pueden distinguir tres categorías de participantes activos en la comunidad, en función de los mensajes enviados. Los *participantes únicos*, que envían solamente un mensaje; los *participantes múltiples*, que envían más de un mensaje y los *participantes vinculados* (habitualmente moderador y gestores), que envían muchos más mensajes.

2.3 Comunidades de práctica y gestión del conocimiento

Aunque inicialmente las CPs se situaron en el contexto del aprendizaje social, también se utilizan frecuentemente en el contexto de la gestión del conocimiento. La importancia de las CPs en la gestión del conocimiento ha sido descrita por numerosos autores (Kimble y Hildreth, 2004; Pan y Leidner, 2003; Hildreth y Kimble, 2002; Wenger *et al*., 2002; Seely Brown y Duguid, 2000; Wenger y Snyder, 2000). Según Wenger *et al*. (2002), la gestión del conocimiento, tradicionalmente centrada en el ámbito de las tecnologías y sistemas de información, debería enfocarse hacia la práctica en la que ese conocimiento se crea, considerando tanto las dimensiones tácita y explícita del conocimiento como su naturaleza dinámica. Este autor define la gestión del conocimiento como la coordinación de actividades de una variedad de actores que ayudan a descubrir, difundir y aplicar el conocimiento. Pan y Leidner, (2003) ilustran la importancia de las CPs en la gestión del conocimiento argumentando que las CPs no siguen los límites tradicionales de las organizaciones, sino que definen sus propios límites informales, más allá de los anteriores. Gracias a eso pueden explotar completamente todas sus competencias y habilidades, contribuyendo al éxito de su misión sin necesidad de cumplir con unas estructuras funcionales. De este modo, las CPs se convierten en aceleradores de las organizaciones (Pan y Leidner, 2003). Las organizaciones pueden beneficiarse de las CPs reconociendo que el conocimiento es inseparable de su contexto.

Para explicar la importancia del concepto de CP en la gestión del conocimiento habría que remontarse al desarrollo de la noción de gestión del conocimiento a lo largo de los años. Inicialmente, el conocimiento fue considerado como un objeto que podía ser capturado, codificado y almacenado. La gestión del conocimiento trataba esencialmente de optimizar

estos tres procesos (Wenger *et al*., 2002; Hildreth *et al*., 1999). La tecnología de soporte de la gestión del conocimiento también se centró en el almacenamiento y recuperación de los bienes de conocimiento (Boisot, 1998). Muchas de las iniciativas de gestión de conocimiento que se lanzaron en este tiempo fracasaron, bien porque el conocimiento almacenado no reflejaba en muchos casos prácticas reales, bien porque las organizaciones fueron incapaces de motivar a los empleados en el uso de estas bases de conocimiento (Wenger *et al*., 2002).

Poco a poco comenzó a estar clara la dificultad de capturar, codificar y almacenar conocimiento. Incluso, cabría pensar si el contenido almacenado en las bases de datos constituía realmente conocimiento o más bien era información (Hildreth y Kimble, 2002). El punto de atención comenzó entonces a estar en la creación, transferencia y aplicación del conocimiento (Alavi y Leidner, 2001), si bien nuevamente cabría preguntarse si resultaba sencillo crear conocimiento mediante los sistemas de gestión del conocimiento.

La dificultad de capturar, codificar y almacenar conocimiento, junto con el problema de creación de conocimiento mediante los sistemas de gestión de conocimiento, desplazaron el punto de atención hacia otros aspectos más humanos del conocimiento. Por ejemplo, Nonaka y Takeuchi (1995) definen dos tipos de conocimiento: tácito y explícito. El conocimiento tácito, como se dijo anteriormente, se basa en el hecho de que sabemos más de lo que podemos decir. Es el conocimiento implícito utilizado por los miembros de la organización para realizar sus tareas (Choo, 1998). El conocimiento tácito se encuentra imbricado en el contexto y las experiencias de sus propietarios, por lo que resulta muy difícil de transferir (Russel, Sambamurthy, y Zmud, 2001). Contempla conceptos tales como valores, creencias, experiencias, emociones y saber-hacer (Skok y Kalmanovitch, 2005). A diferencia del anterior, el conocimiento explícito puede ser codificado y es más fácil de transferir (Choo, 1998; Russel *et al*., 2001)

La dicotomía conocimiento tácito/explícito se encuentra ampliamente aceptada y reconoce el lado intangible y humano del conocimiento. Otros autores diferencian en cambio entre conocimiento hard y soft (Hildreth *et al*., 1999). Aunque existen muchas herramientas para gestionar el conocimiento hard, hay muy pocas que permitan hacer lo mismo con el conocimiento soft, mucho más difícil de gestionar. El carácter implícito de este último

hace que este tipo de conocimiento se pierda cuando los expertos abandonan la organización. Por tanto, existe la problemática de gestionar este conocimiento y las comunidades de práctica pueden representar una posible solución.

Las CPs constituyen una aproximación a la gestión del conocimiento desde la perspectiva del conocimiento en práctica (Lave *et al*., 1991; Brown y Duguid, 1991; Wenger, 1998). Las CPs proporcionan un entorno en el que los participantes pueden desarrollar su conocimiento en interacción con otros y donde el conocimiento se crea, se alimenta y se sostiene (Hildreth y Kimble, 2002). Su valor se encuentra precisamente en la participación en comunidad; en particular, en ser un miembro activo en las prácticas de la comunidad y en construir una identidad en su relación con ella (Fowler y Mayes, 1999). La creación y la difusión del conocimiento son dos de las actividades esenciales de las CPs (Brown y Duguid, 2000). Las CPs tienen en cuenta la gestión del conocimiento soft sin descuidar la gestión del conocimiento hard.

El proceso subyacente en la creación y desarrollo del conocimiento soft es la llamada Participación Periférica Legítima (PPL) (Lave *et al*., 1991). La *participación periférica legítima* es el proceso por el cual un individuo, situado dentro de un contexto determinado, aprende de este contexto mediante la participación en un "sistema de actividad con comprensión y significados compartidos". Es decir, mediante la participación en la actividad de la comunidad y la interacción durante este proceso con otros que pueden ser más o menos conocedores del campo, el individuo aprende progresivamente a "hablar el idioma" de la comunidad y a participar plenamente en su actividad. De esta manera progresa paulatinamente, pasando de la periferia hacia el interior de la comunidad (Kimble *et al*., 2000). Hildreth y Kimble (2002) sitúan la participación, que es un elemento clave en el proceso de negociación del significado, en la dimensión soft del conocimiento. No obstante, la participación no tendría sentido sin el otro elemento clave llamado *cosificación*. La cosificación significa dar una forma concreta a algo que es abstracto (convertir algo en cosa). Ambas, la participación y la cosificación, forman una dualidad que interviene en el proceso de negociación del significado. Hildreth and Kimble (2002) asocian la cosificación a la dimensión hard del conocimiento. Estas dos asociaciones señaladas, conocimiento hard y cosificación por un lado, y conocimiento soft y

participación por otro, muestran creciente importancia de las CPs en al gestión del conocimiento.

2.4 Comunidades virtuales

La aparición de Internet ha hecho emerger la posibilidad de comunidades de práctica distribuidas donde sus miembros no interaccionan de manera directa. A este tipo de comunidades se les denomina comunidades de práctica virtuales o redes de práctica (RP) (Brown y Duguid, 2000; Brown y Duguid, 2001). Las comunidades virtuales hacen uso de las tecnologías de la información y comunicación para superar barreras geográficas y horarias (Johnson, 2001). A diferencia de las comunidades tradicionales, las comunidades virtuales tienen menos formalismos y las normas no están tan presentes como en las tradicionales, pues sus miembros no tienen un contacto directo (Squire y Johnson, 2000). Johnson (2001) define una *comunidad virtual* como un grupo separado en el espacio y en el tiempo que hacen uso de las tecnologías de la información para comunicarse y colaborar. Considera a las comunidades virtuales dentro de la Teoría de las Comunidades de Práctica. Argumenta que la comunidad virtual es la comunidad diseñada, de donde emerge la auténtica comunidad de práctica. Es decir, la comunidad virtual es sólo el soporte de la comunidad de práctica. Para que ésta surja es necesario que sus integrantes lleguen a ser miembros plenos comenzando por una participación periférica y que permita la construcción de identidad y confianza (Hildreth *et al*., 1999). La ausencia de un contacto cara a cara hace que el desarrollo de comunidades de práctica distribuidas requiera de tiempo y una constante revisión de los objetivos y las prácticas para mantener la comunidad activa (Bradshaw, Powell, y Terrel, 2004).

Aunque las CPs pueden existir online, existen algunas limitaciones y riesgos. Por ejemplo, Dyer y Nobeoka (2000) argumentan que los participantes pasivos pueden llegar a constituir un serio problema. Estos participantes pasivos podrían definirse como aquellos miembros que se benefician de los resultados de la comunidad sin contribuir a su desarrollo. También se les conoce en inglés como *freeriders* o *lurkers* (Millen *et al*., 2005; MclureWasko y Faraj, 2005; Lueg, 2000). Las CPs toleran la presencia de participantes pasivos. Los

participantes periféricos pueden traer nuevas luces a la comunidad y negociar nuevos significados. Siguiendo una trayectoria desde la observación hacia la participación, los miembros periféricos llegarían a una participación plena, garantizando la supervivencia de la comunidad. La amenaza es que los participantes pasivos continúen su pasividad y nunca pasen a ser miembros plenos, dañando de ese modo el buen funcionamiento de la comunidad.

Otro dilema importante es motivar a los miembros de la comunidad a participar y compartir su conocimiento de forma abierta (Dyer y Nobeoka, 2000). Si no se comparte conocimiento es imposible que la comunidad pueda negociar un significado. Wasko y Faraj (2000) argumentan que se empieza a compartir conocimiento cuando se considera el conocimiento como un bien público, propiedad de la comunidad, en lugar de como un bien privado, propiedad de la organización. Si se considera como un bien público, los participantes comparten su conocimiento por un sentido de deber moral. La buena disposición a participar y mantener la comunidad y la reputación son otras motivaciones importantes que fomentan la participación (Wasko y Faraj, 2005).

2.4.1 Creación de conocimiento en comunidades distribuidas

La literatura inicial sobre creación de conocimiento se elaboró antes de que Internet revolucionase la colaboración entre equipos a través de sistemas electrónicos (Nonaka y Takeuchi, 1995; Argyris 1993, Prahalad y Hamel, 1990: Levitt y March, 1988; Winter, 1987; Teece, 1986; Nelson y Winter, 1982; Penrose, 1959). Los pioneros en la búsqueda de la creación del conocimiento trabajaron con tres suposiciones críticas:

- En primer lugar, supusieron que la unidad de análisis o el foco de atención debía situarse en una firma o en un conjunto de firmas. La firma constituía un repositorio de conocimiento y el lugar donde dicho conocimiento era creado y desplegado, pues suministra el contexto bajo el cual el conocimiento se construye. Incluso cuando el punto de atención está en el individuo (Argyris, 1993) lo está en el contexto del aprendizaje para una firma. Cuando una firma o un conjunto de firmas constituyen la unidad de análisis, se da por descontado que esa firma intentará aprovechar el

conocimiento creado por sus empleados como propiedad intelectual y como ventaja competitiva. Esto supone la protección de ese conocimiento o compartirlo bajo ciertas condiciones (Teece, 1986; Brown y Duguid, 2000).

- En segundo lugar, se asume que las interacciones cara a cara entre los desarrolladores de conocimiento compartiendo un mismo espacio facilita la construcción de confianza durante un largo período de tiempo (Nonaka y Takeuchi, 1995). En realidad, se trata de una extensión natural de considerar la firma como unidad de análisis, porque es la proximidad física de la firma la que soporta el desarrollo de una confianza a través de interacciones repetidas y de normas sociales compartidas. Aunque la confianza es una variable clave en la creación y gestión del conocimiento (Powell *et al*., 1996; Kramer y Tyler, 1996), el uso de medios de comunicación electrónicos comienza a ser dominante, por lo que también se puede conseguir la construcción de esa confianza en entornos virtuales, geográfica y organizativamente dispersos.
- Por último, se asume que dentro de la unidad de análisis que representa la firma, la creación de conocimiento tiene lugar bajo condiciones de autoridad y jerarquía, donde la producción de conocimiento complejo requiere de complejas divisiones del trabajo.

Estas suposiciones contrastan fuertemente con los modelos de creación de conocimiento basados en comunidades (Lee y Cole, 2003). Los principios organizativos de las comunidades marginan las suposiciones críticas de modelo basado en firmas. El modelo basado en comunidad pone especial atención en cómo solucionar problemas, conduce el proceso de creación de conocimiento y genera una estructura en una comunidad. La crítica y la corrección de errores juegan un papel primordial en el proceso de creación de conocimiento, de modo que las innovaciones son continuamente generadas, seleccionadas y retenidas. La Tabla 2 resume las diferencias más importantes en la creación de conocimiento del modelo basado en una organización y del modelo basado en comunidad. Las tres suposiciones explicadas anteriormente son violadas claramente en el modelo basado en comunidad:

- La cesión de los derechos de propiedad intelectual fomenta la confianza y el compartir conocimiento.

- La afiliación a la comunidad es abierta, consiguiendo de ese modo tamaños muy superiores a los de cualquier organización.
- Los incentivos y la motivación de los trabajadores se desplaza de los que necesitan los empleados a los que necesitan los voluntarios y, a diferencia de lo que ocurre en una organización, no existe una autoridad que regule el funcionamiento o el comportamiento de los miembros de la comunidad.
- Los individuos se encuentran organizativa y geográficamente dispersos.
- La plataforma de creación de conocimientos se basa en la comunicación a través de medios electrónicos y muchas veces dirigido a toda la comunidad.

Tabla 2. Creación de conocimiento en el modelo basado en la comunidad frente al modelo basado en la organización

Principios Organizativos	Modelo basado en la organización	Modelo basado en la comunidad
Propiedad intelectual	El conocimiento es privado y pertenece a la organización	El conocimiento es público aunque puede pertenecer a miembros que los comparten
Restricción de miembros	Los miembros son seleccionados y el tamaño de la organización está limitado por el número de empleados	La afiliación es abierta, por lo que el tamaño de la comunidad no está limitado
Autoridad e incentivos	Los miembros de la organización son empleados que reciben un salario por su trabajo	Los miembros de la comunidad son voluntarios que no reciben ningún salario por su trabajo
Distribución de conocimiento más allá de los límites geográficos y organizativos	La distribución está limitada a los límites de la organización	La distribución se extiende más allá de los límites de la firma
Modelo de comunicación dominante	Interacción cara a cara	Interacción mediante el uso de recursos electrónicos

Según Campbell (1960), todos los procesos que llevan a la expansión del conocimiento deben cumplir necesariamente tres condiciones: un mecanismo para introducir variaciones, un proceso de selección consistente, y un mecanismo para preservar y reproducir las variaciones seleccionadas. Básicamente, el proceso de creación de conocimiento implica un ciclo de iteraciones del tipo variación ciega y retención selectiva. En este sentido, los principios organizativos de una empresa son un vehículo eficiente para la acumulación de un conocimiento especializado, pero no para generar variedad en las innovaciones. Este hecho nos lleva a que es necesario buscar una mejor comprensión de la creación de conocimiento más allá de los límites de una organización (Lee y Cole, 2003).

2.5 Beneficios e inconvenientes de las comunidades de práctica

La principal ventaja de las CPs es que aumentan el nivel y el flujo de conocimiento. Pero dada la naturaleza intangible del hecho de compartir conocimiento, estos beneficios son difíciles de cuantificar.

Wenger *et al*. (2002) tratan de realizar esta cuantificación calculando el retorno de la inversión (ROI). Otros autores (Fontaine y Millen, 2004; Kimble y Hildreth, 2004; Lesser y Storck, 2001) se centran más en el capital social de las CPs. El capital social puede definirse en términos de una serie de conexiones entre individuos, el desarrollo de un sentido de confianza entre esas conexiones y la disponibilidad de un interés común o compartido (Lesser y Storck, 2001). Aunque los beneficios de las CPs pueden explicarse parcialmente desde la perspectiva social, los resultados son subjetivos por naturaleza y los gerentes buscan todavía resultados cuantificables.

Otro problema de las CPs reside en la transferencia libre de información y conocimiento. Para que esta transferencia pueda funcionar correctamente es necesario que el conocimiento sea considerado un bien público (Ardichvili *et al*., 2003). No obstante, en la mayoría de los casos el conocimiento se considera un bien privado (Wasko y Faraj, 2000). Esta visión propietaria del conocimiento es devastadora para el buen desarrollo de las CPs, paralizando el intercambio de conocimiento e información, y las habilidades de aprendizaje de la comunidad.

El hecho de que las CPs se autorregulen y se automotiven puede ser, al mismo tiempo, una ventaja y un inconveniente, ya que pueden desembocar en unos objetivos que no sean concurrentes con los de la organización (Gongla y Rizutto, 2004).

Finalmente, otros problemas que pueden presentar las CPs están relacionados con una falta de identidad común o una identidad demasiado fuerte, que impida compartir conocimiento con otras comunidades (Hislop, 2004)

Capítulo 3. Proyectos de Software de código abierto

3.1 Introducción

El término "código abierto" fue introducido por primera vez en 1998 (Raymond, 1998a) y desde entonces ha tenido un interés creciente. Habitualmente, el término "código abierto" se suele relacionar con Linux y sus aplicaciones (O'Reilly, 1999). Pero la realidad es que existen muchos ejemplos de software de código abierto, como por ejemplo el servidor web Apache, con una cuota de mercado de alrededor del 70%, el lenguaje PHP, el sistema de gestión de bases de datos MySQL o el paquete OpenOffice (Wheeler, 2007).

La característica fundamental del software de código abierto (OSS, Open Source Software) es que el código fuente se encuentra disponible a los usuarios (Weber, 2004). El término código abierto se encuentra definido por la Open Source Initiative (OSI) (OSI, 2002) y básicamente contempla los siguientes diez aspectos:

1. Libre redistribución: el software debe poder ser regalado o vendido libremente.
2. Código fuente: el código fuente debe estar incluido u obtenerse libremente.
3. Trabajos derivados: la redistribución de modificaciones debe estar permitida.
4. Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
5. Sin discriminación de personas o grupos: nadie puede dejarse fuera.
6. Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
7. Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
8. La licencia no debe ser específica de un producto: el programa no puede licenciarse sólo como parte de una distribución mayor.
9. La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.

10. La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

El software de código abierto se caracteriza por ofrecer el código fuente junto con el software, sin ningún cargo adicional, salvo costes de distribución. Esto permite que el usuario final pueda modificar las instrucciones y, por tanto, modificar el comportamiento del programa o añadir nuevas funcionalidades. Asimismo, ofrecen a cualquier usuario la posibilidad de colaborar en el desarrollo del proyecto software (Wheeler, 2007). Estos proyectos se desarrollan en la mayoría de los casos a través de Internet. Los sitios web de los proyectos de software de código abierto incluyen numerosa documentación: discusiones y debates, documentación, bases de datos de bugs, etc.

La mayoría de los proyectos de software de código abierto animan a los usuarios a participar en el proyecto, desde informar sobre bugs al desarrollo del propio código fuente. Si el usuario quiere modificar o añadir algo al software, es libre de hacerlo por sí mismo, pero trabajando con la comunidad y aportando el código fuente desarrollado. Éste código formará parte del código disponible para el resto, quienes se encargarán de mantenerlo y actualizarlo en futuras versiones. Los desarrolladores de software de código abierto trabajan juntos de manera voluntaria para crear y mejorar el proyecto. En este sentido, existen numerosos trabajos a cerca de la motivación en el desarrollo de software de código abierto (Hars y Ou, 2002; Hertel *et al*., 2003).

3.1.1 Software propietario, abierto y libre

Un software propietario es aquél que se encuentra sometido a una licencia por la que su propietario restringe su uso y limita su distribución. Típicamente, el software propietario reserva casi todos los derechos a su autor, excepto una licencia para ejecutar el software en el ordenador del usuario (Zittrain, 2004), y se suministra como un código binario ejecutable sin posibilidad de acceder al código fuente. Esta práctica mantiene el secreto del código fuente y, aunque presenta la ventaja para el usuario final de no tener que compilar el código, esto es a costa de no poder modificarlo. Un acuerdo de licencia de usuario final,

EULA (*End-User License Agreement*) es la forma más habitual de licencia software. Típicamente, su aceptación se incluye durante el proceso de instalación del software y como pre-condición para su instalación final, e incluye todas las restricciones asociadas a su uso y distribución. Aunque estas licencias rara vez son leídas por el usuario final (que se limita a hacer click en el botón de aceptar), sí que resultan importantes para los desarrolladores, pues afectan a posibles adaptaciones del software y a su uso por usuarios corporativos.

Respecto a las denominaciones de código abierto y software libre, a veces existe cierta confusión entre ellas. El término código abierto fue introducido por primera vez por Raymond, (1998a) para evitar la confusión con el término "software libre" que se venía usando hasta entonces. Cuando se habla de software libre hay que entender un asunto de liberad y no de precio. La confusión viene precisamente de que la palabra inglesa *free* significa tanto libre como gratis. Otra forma de describirlo es "libre" como en "libertad de expresión", no como en "cerveza gratis" (Weber, 2004). La otra acepción sería *free* como software disponible sin coste, pero sin el código fuente. Este tipo de software se denomina como "freeware".

La principal defensora del código libre frente al código abierto es la Fundación de Software Libre (FSF, Free Foundation Software, 2005). Según ella, software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Para poder explicar estas diferencias es necesario tener en cuenta que todo el movimiento de software de código abierto no es sólo un método de desarrollo del software, sino una cultura. Según Raymond (1998b), existen dos grupos clave en esta cultura, que serían los puristas y los pragmáticos. Los puristas son el grupo más fanático, de gran hostilidad al software comercial, y que ven el código abierto como un fin en sí mismo. Los pragmáticos son por el contrario moderadamente anti-comerciales y ven al código abierto más como un medio que como un fin.

El grupo purista se encuentra alrededor de la Free Software Foundation (FSF), fundada por Richard M. Stallman (Raymond, 1998b). La FSF tiene su origen en el grupo GNU. El proyecto GNU fue lanzado en 1984 para desarrollar un completo sistema operativo tipo UNIX, bajo la filosofía del software libre: el sistema GNU. Las variantes del sistema operativo GNU que utilizan un núcleo de Linux son utilizadas ampliamente en la actualidad. Aunque a menudo estos sistemas se refieren como "Linux", deberían ser llamados sistemas GNU/Linux. GNU es un acrónimo recursivo para «GNU No es Unix» y se pronuncia fonéticamente en español. La FSF introdujo la GNU *General Public License* (GPL) y se caracteriza por contemplar el código abierto o libre como un derecho (Stallman, 1992), a diferencia de los pragmáticos, quienes ven el desarrollo del software de código abierto como un medio de crear buen software.

Uno de los principios claves de la FSF, reflejado también en la licencia GPL que usan y promueven, es el "*copyleft*". Una licencia *copyleft* usa el *copyright* para proteger al código fuente así licenciado de ser incorporado en software de código cerrado o propietario. Cualquier software licenciado bajo una licencia *copyleft* debe permanecer bajo dicha licencia, incluyendo sus derivados (Weber, 2004).

En realidad, los principios de la FSF son coherentes con la definición realizada de código abierto, pero mientras que la FSF sólo cree en licencias del tipo *copyleft*, la definición de la OSI permite otras licencias no *copyleft* para el software de código abierto.

El grupo de los pragmáticos ha crecido mucho debido al movimiento Linux, encabezado por Linus Torvalds. Aunque sin oponerse a los principio de Richard M. Stallman, también respaldan el uso de software comercial de calidad. Bruce Perens, que lidera el proyecto

Debian (sistema operativo GNU/Linux) (Perens, 2005), también comparte esta visión pragmática. El principio básico del contrato social de Debian es la no discriminación contra ninguna persona, grupo de personas o campo de trabajo, incluyendo el software comercial. La exclusión expresa que hace FSF de cualquier elemento propietario se opone al principio del contrato social de Debian de promocionar el crecimiento del uso y desarrollo del software de código abierto.

En el otro extremo de todas estas tendencias e interpretaciones previas se encuentra el llamado software comercial o propietario. Realmente no son sinónimos, puesto que el software de código abierto puede ser comercial o ser usado comercialmente. El término propietario se suele usar para indicar que se trata de un software de código cerrado. El código fuente pertenece exclusivamente el vendedor y no se proporciona a los compradores.

3.1.2 Otras definiciones de términos

Propiedad Intelectual

La propiedad intelectual, desde el punto de vista de la tradición continental europea y de los principales países latinoamericanos, supone el reconocimiento de un derecho de propiedad especial en favor de un autor u otros titulares de derechos sobre las obras del intelecto humano. Generalmente, se entiende por derechos de autor aquellos que éste tiene para controlar el uso que se hace de su obra. Pero no existe una idea uniforme sobre el alcance y el contenido concreto de esos derechos. Las tradiciones anglosajona y europea conciben esos derechos de forma distinta. Mientras en el mundo anglosajón predomina una concepción utilitarista de los derechos de autor, en Europa se ha adoptado un enfoque que los concibe como derechos de la persona.

En la tradición estadounidense es el Congreso el que otorga unos derechos limitados a los autores para promover el bienestar social, permitiendo que aquéllos gocen de incentivos para desarrollar su creatividad y ponerla a disposición del público. No se trata, pues, como postula la tradición europea, de que exista un derecho natural de los autores a la propiedad de sus obras que la ley deba limitarse a reconocer. En su lugar, el copyright estadounidense

es, más bien, una negociación entre los autores y la sociedad, por la cual esta última concede a los primeros un monopolio temporal y limitado para controlar y explotar sus obras, con la esperanza de que así florezca la cultura y el arte.

La Ley de la Propiedad Intelectual (LPI) en España y leyes similares en otros países, desarrolladas sobre la base del Convenio de Berna para la protección de trabajos literarios y artísticos de 1886, regulan los derechos de autor. Estos derechos se dividen en derechos morales y patrimoniales. Los primeros garantizan al autor el control sobre la divulgación de su obra, con nombre o seudónimo, el reconocimiento de autoría, el respeto a la integridad de la obra y el derecho de modificación y retirada. Los segundos dan derecho a explotar económicamente la obra y pueden ser cedidos total o parcialmente, de forma exclusiva o no, a un tercero. Los derechos morales son vitalicios o indefinidos, mientras que los patrimoniales tienen una duración bastante larga (70 años después de la muerte del autor, si es una persona física, en el caso de la ley española).

Licencias

Los poseedores de una propiedad intelectual pueden vender o licenciar su propiedad para determinados usos. Los compradores de un libro compran por ejemplo el derecho a leerlo y poder revenderlo más adelante. Existen otros derechos como la copia de un determinado número de páginas o la cita del libro por motivos académicos. El caso del software es algo diferente. Una licencia de software es la autorización o permiso concedido por el titular del derecho de autor, en cualquier forma contractual, al usuario de un programa informático, para utilizar éste en una forma determinada y de conformidad con unas condiciones convenidas.

La licencia, que puede ser gratuita u onerosa, precisa los derechos (de uso, modificación o redistribución) concedidos a la persona autorizada y sus límites. Además, puede señalar el plazo de duración, el territorio de aplicación y todas las demás cláusulas que el titular del derecho de autor establezca.

Linux

Linus Torvals, estudiante de una universidad finalandesa, comenzó a desarrollar un núcleo o kernel compatible con UNIX en 1991, denominado Linux. Torvals distribuyó Linux bajo

una licencia *copyleft*, e invitó a cualquiera a contribuir a su desarrollo y mejora. A partir de ahí se constituyó una comunidad de desarrolladores que creció rápidamente y realizó grandes avances en poco tiempo. Realmente, Linux no es un sistema operativo sino el núcleo o kernel de un sistema operativo, independiente de la máquina en la que se ejecute. En 1992, la combinación del kernel de Linux con el sistema GNU dio lugar a un sistema operativo completamente libre. Si bien suele denominarse Linux al sistema operativo, su nombre correcto debería ser sistema operativo GNU/Linux. Desde entonces, las versiones mejoradas y extendidas del kernel de Linux y de las herramientas GNU han sido publicadas y millones de personas en el mundo se han unido a la comunidad GNU/Linux.

3.2 Licencias de software de código abierto

Un licencia software es un acuerdo entre un usuario y un desarrollador. A lo largo de los últimos años han ido surgiendo una gran variedad de licencias que podrían denominarse de código abierto, muchas de las cuales sirven de soporte de un determinado modelo de negocio, como son las licencias de código abierto de Sun, IBM y Netscape Corporation. Entre las más conocidas se encuentran:

- GNU General Public License (GPL)
- GNU Library or 'Lesser' Public License (LGPL)
- BSD license
- MIT license
- Artistic License
- Mozilla Public License (MPL)
- Q Public License (QPL)
- IBM Public License
- MITRE Collaborative Virtual Workspace License (CVW License)
- Ricoh Source Code Public License

- Python License
- zlib/libpng License

En la actualidad existen hasta 58 licencias de código abierto aprobadas por la OSI. A continuación se detallan las más importantes. Más información puede extraerse de los sitios web indicados en la Tabla 3.

Tabla 3. Licencias de software de código abierto más importantes.

Licencia	Web
GNU Public License – GPL	http://opensource.org/licenses/gpl-license.php
GNU Lesser General Public License – LGPL	http://opensource.org/licenses/lgpl-license.php
BSD License	http://opensource.org/licenses/bsd-license.php
Mozilla Public License	http://opensource.org/licenses/mozilla1.1.php

3.2.1 GNU Public License – GPL

La Licencia Pública General del proyecto GNU (más conocida por su acrónimo en inglés GPL) es, con diferencia, la licencia más popular y conocida de todas las del mundo del software libre. Su autoría corresponde a la FSF, promotora del proyecto GNU, y, en un principio, fue creada para ser la licencia de todo el software generado por la FSF. Sin embargo, su utilización ha ido más allá hasta convertirse en la licencia más utilizada, incluso por proyectos bandera del mundo del software libre, como el núcleo Linux.

La licencia GPL es interesante desde el punto de vista legal porque hace un uso muy creativo de la legislación de *copyright* consiguiendo efectos prácticamente contrarios a los que se suponen de la aplicación de esta legislación: en lugar de limitar los derechos de los usuarios, los garantiza. Por este motivo, en muchos casos se denomina a esta maniobra *copyleft* (juego de palabras en inglés que a veces se traduce como "izquierdos de autor").

En líneas básicas, la licencia GPL permite la redistribución binaria y la del código fuente, aunque en el caso de que redistribuya de manera binaria obliga a que también se pueda

acceder a las fuentes. Asimismo, está permitido realizar modificaciones sin restricciones. Sin embargo, sólo se puede redistribuir código licenciado bajo GPL de forma integrada (por ejemplo, mediante enlazado o linkado) con otro código si éste tiene una licencia compatible. Una licencia es incompatible con la GPL cuando restringe alguno de los derechos que la GPL garantiza, ya sea explícitamente, contradiciendo alguna cláusula, ya implícitamente, imponiendo alguna nueva. Por ejemplo, la licencia BSD actual es compatible, pero la de Apache, que exige que se mencione explícitamente en los materiales de propaganda que el trabajo combinado contiene código de todos y cada uno de los titulares de derechos, es incompatible. Esto no implica que no se puedan usar simultáneamente programas con ambas licencias, o incluso integrarlos. Sólo supone que esos programas integrados no se pueden distribuir, pues es imposible cumplir simultáneamente las condiciones de redistribución de ambas. Esto ha sido llamado *efecto viral* (aunque muchos consideran esta denominación como despectiva) de la GPL, ya que código publicado una vez con esas condiciones nunca puede cambiar de condiciones.

3.2.2 GNU Lesser General Public License – LGPL

La Licencia Pública General Menor del proyecto GNU (comúnmente conocida por sus iniciales en inglés LGPL) es la otra licencia de FSF. Pensada en sus inicios para su uso en bibliotecas (la L en sus comienzos venía de *library*, biblioteca) fue modificada recientemente para ser considerada la hermana menor (*lesser*, menor) de la GPL. La LGPL permite el uso de programas libres con software propietario. El programa en sí se redistribuye como si estuviera bajo la licencia GPL, pero se permite la integración con cualquier otro software sin prácticamente limitaciones.

3.2.3 BSD License

La licencia BSD es la licencia original de una distribución de Software, Berkeley Software Distribution, que acabó convirtiéndose en un derivativo de UNIX realizado por la conocida Universidad de California, Berkeley. La licencia BSD ha tenido dos formas

principalmente: la clásica (con la cláusula de publicidad) y la actual (sin esa cláusula, desde Julio del 99).

Es una licencia de software de código abierto que no impone restricciones significativas al uso posterior del código fuente por parte de cualquiera, los desarrolladores de software comercial incluidos. Se puede incorporar el código en otros productos y puede distribuirse en forma binaria, con o sin modificaciones, sometiéndose únicamente al aviso del *copyright*. De ahí que, al contrario que la licencia pública general GNU, la licencia BSD no es una licencia de software libre y no es "transmisible". A modo de ejemplo, el sistema operativo FreeBSD y el NetBSD se distribuyen con esta licencia, mientras que Apache, un paquete de servidor Web, se distribuye bajo una variante de la licencia BSD.

3.2.4 Mozilla Public License

Mozilla Public License (MPL) (Licencia Pública de Mozilla) es una licencia de código abierto y software libre. Fue desarrollada originalmente por Netscape Communications Corporation –una división de la compañía América Online– y más tarde su control fue traspasado a la "Fundación Mozilla". La licencia MPL cumple completamente con la definición de software de código abierto de Open Source Initiative (OSI) y con las cuatro libertades del software libre enunciadas por FSF. Sin embargo, MPL deja abierto el camino a una posible reutilización comercial no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni el relicenciamiento bajo la misma licencia. Muchas compañías han adoptado variaciones de la MPL para sus propios programas, como por ejemplo Netscape Public License (NPL), Internase License, Nokia Open Source License e IBM Public License.

3.3 Teorías sobre el software de código abierto

Existen muchas aproximaciones teóricas que tratan de explicar el fenómeno del software de código abierto. Eric Raymond describe la comunidad de código abierto y su forma de desarrollar software en el libro "The Cathedral & the Bazaar" (Raymond, 2001). El título

del libro representa una alegoría: la producción de software propietario es la cuidadosamente planificada construcción de una catedral, mientras que la producción de software de código abierto con las interacciones caóticas de las personas es un bazar oriental. Es decir, gestión fuerte y centralizada frente a desarrolladores distribuidos y organizados en cientos de proyectos aparentemente independientes.

Según Raymond (2001), parte de la respuesta a cómo funcionan los proyectos de software de código abierto reside en el hecho de que los desarrolladores no sólo necesitan soluciones sino que necesitan soluciones a tiempo, sin tener que esperar a que el proveedor de software proporcione una solución al problema. Encontrar un *bug* (error de software) en un producto como Internet Information Server (IIS) es una experiencia radicalmente diferente a encontrar un *bug* en un producto comparable de código abierto, como es Apache. La diferencia es que un *bug* en IIS requiere que Microsoft lo solucione, lo que puede llevar un tiempo indeterminado. En el caso del producto de código abierto, el problema probablemente ya se encuentre solucionado.

Otros argumentos sostienen que el código abierto es mejor porque existe un intenso espíritu de equipo dentro de la comunidad y porque sólo los individuos de más talento se unen a ella.

Existen también argumentaciones en contra de estas ideas. Por ejemplo, se asume que una compañía con software propietario no puede usar el mismo procedimiento de revisión masiva para localizar *bugs* en el software, extendiendo su red de desarrolladores. Además, en muchos proyectos de software de código abierto, muchas de las mejoras son realizadas por un pequeño grupo central, por lo que una compañía comercial podría sostener un grupo similar de desarrolladores.

Desde el punto de vista de la Teoría de la Organización, una perspectiva interesante es la aportada por Jonson (2006). En este trabajo se argumenta que el éxito del software de código libre se puede explicar parcialmente a la luz de la Teoría de los Costes de Transacción. La idea básica es que el coste de transacción es el coste de paliar la desconfianza mutua entre dos agentes económicos. El intercambio entre ellos exigirá, al final, la participación de una autoridad judicial con poder coactivo que represente un coste

social, necesario para paliar la falta de confianza, que excede al coste privado de una eventual transacción entre ellos. La organización de las comunidades de software de código abierto minimiza los costes de transacción asociados a la distribución de información entre trabajadores. Esto se manifiesta en la superioridad de estas comunidades en dos actividades: la colaboración o intercambio de ideas y la revisión entre iguales. Dentro de las compañías propietarias, estas actividades pueden languidecerse debido a problemas de agencia. En lo que respecta a la revisión entre iguales, está claro que la calidad de un proyecto será tanto mayor cuanto mayor sea esta revisión crítica. No obstante, dentro de una compañía comercial existen incentivos para que los trabajadores se confabulen contra el director o jefe y no se inspeccionen mutuamente los trabajos de una manera rigurosa. El incentivo para esa confabulación es retrasar la llegada de información acerca de la habilidad de los trabajadores, potencialmente peligroso para futuros salarios o incentivos. Esto no ocurre en las comunidades de código abierto, donde los participantes no reciben una remuneración y están más preocupados por la calidad del proyecto. Respecto al intercambio de ideas, los trabajadores de una compañía comercial pueden ser más reticentes a aportar mejoras, pues esto puede repercutir en mayores cargas de trabajo para ellos. Por el contrario, los participantes en comunidades de código abierto no pueden ser obligados a trabajar más allá de su deseo. Aunque esto supone que la comunidad de código abierto no siempre implementa las mejoras sugeridas de forma tan eficiente como podría hacerlo una compañía, los participantes en la comunidad siempre elegirán intercambiar ideas libremente puesto que siempre pueden elegir el esfuerzo que invertirán en desarrollarlas.

Existen muchos trabajos que tratan sobre la motivación de los programadores y que examinan los motivos por los cuales se deciden a participar en proyectos de software de código abierto (Bitzer *et al*., 2007; Subramanyam y Xia, 2008). Muchos de ellos analizan casos concretos de proyectos de software de código abierto famosos, como Apache, Perl, o Sendmail (Raymond, 1998b, 2001; Lerner y Tirole, 2002; Torvalds y Diamond, 2001). Otros analizan archivos web relacionados con Linux (Hertel *et al*., 2003), o bien una gran variedad de proyectos de software de código abierto diferentes (Lakhani y Wolf, 2005; Krishnamurthy, 2002; Hars y Ou, 2002; Lakhani y Hippel, 2003)

Básicamente, estos estudios identifican dos tipos de motivaciones: intrínsecas y extrínsecas. Una motivación intrínseca describe una situación en la que alguien hace algo porque es inherentemente interesante, divertida o estimulante. En el caso de la motivación extrínseca, alguien espera un beneficio aparte, normalmente monetaria, aunque también puede ser de otra índole (Deci y Ryan, 1985; Ryan y Deci, 2000; Amabile, 1996; Osterloh y Frey, 2000).

Las motivaciones intrínsecas se encuentran relacionadas con la diversión en la realización de una actividad o el deseo de hacer algo por la comunidad.

- Diversión: existe una evidencia clara de que para muchos programadores su trabajo es una recompensa en sí mismo. Muchos participantes en proyectos de software de código abierto sostienen que los motivos que les llevan a participar en ellos son por diversión, por aprender y por mostrar en público sus habilidades personales (Csikszentmihalyi, 1990, 1996).
- Motivación social: con frecuencia se habla de que las comunidades de código abierto poseen una cultura del regalo en lugar de una cultura del intercambio (Raymond, 2001). Muchos de los participantes en estas comunidades señalan que les gusta la sensación de ayudar a otros o de devolver algo a las personas con esa mentalidad (Faraj y Wasko, 2001; Frey y Meier, 2004; Shah, 2006). En ocasiones los participantes parece que responden más a toda la comunidad que a una cuestión personal (Wellman y Gulia, 1999).

En cuanto a las motivaciones extrínsecas tienen que ver con los beneficios que obtienen los programadores para satisfacer sus necesidades y su reputación.

- Beneficios de la mejora en la funcionalidad: los participantes en proyectos de software de código abierto pueden beneficiarse de la adaptación del software a sus propias necesidades (Von Hippel, 1988; Von Hippel y von Krogh, 2003). Publican sus correcciones para que otros usuarios puedan trabajar sobre ellas y mantenerlas o mejorarlas. Dado que el coste de publicar la corrección en Internet es muy bajo, merece la pena intentarlo aun cuando las expectativas de recibir ayuda de otros usuarios sea

baja. Los trabajos de Niedner *et al*. (2000) y Lakhani y Wolf (2005) señalan esta motivación como la más importante.

- Reputación: los participantes en comunidades de software de código abierto pueden estar motivados por destacar sus capacidades para, en un futuro, conseguir una compensación en forma de trabajo de una compañía de software o de una empresa de capital-riesgo. En proyectos de software de código abierto es más fácil conseguir una buena reputación que en proyectos de software propietario, dada la naturaleza pública de las listas de distribución (Lerner y Tirole, 2002; Moon y Sproull, 2000).

En contra de lo que podría pensarse a priori, el software de código abierto también tiene un claro interés comercial. Son muchas las compañías que han puesto sus ojos en los modelos innovadores característicos de los proyectos de software de código abierto (Osterloh y Rota, 2007). Aunque gran parte del esfuerzo de desarrollo de proyectos de software de código abierto sigue recayendo en programadores individuales, las empresas comerciales son vitales también para su éxito. En primer lugar, porque contribuyen a ellos y, en segundo, porque a diferencia de los programadores individuales, tienen interés en proporcionar servicios fiables y continuos incluso a usuarios no experimentados (Kogut y Metiu, 2000). Existen compañías que incluso basan su modelo de negocio en productos de software de código abierto. Por ejemplo, Red Hat vende soporte y servicios de Linux y otros componentes. Otras firmas decidieron abrir proyectos de software de código abierto allí donde antes habían desarrollado software propietario. Es el caso de Netscape, que fue la primera que en 1998 decidió abrir el código de su Netscape Communicator y empezar el proyecto Mozilla. Otras compañías deciden contribuir a proyectos de software de código abierto existentes. IBM contribuye al desarrollo de Linux y hace su hardware compatible con él. Hewlett Packard vende impresoras y otro hardware y contribuye en forma de *drivers* que hacen sus productos compatibles con software de código abierto.

Henkel (2004) define cinco categorías de motivos por los cuales compañías privadas se deciden a contribuir a proyectos de software de código abierto:

- Establecer un estándar que permita la compatibilidad
- Incrementar la demanda de bienes y servicios complementarios

- Beneficiarse del soporte proporcionado por desarrollos externos
- Mostrar la excelencia técnica y su compromiso con el software de código abierto
- Adaptar el software de código abierto existente a las necesidades de la empresa

La mayoría de los modelos de negocio que incorporan software de código abierto se benefician de la amplia difusión de su software. El mercado para estas compañías se extiende debido a que los usuarios pueden modificar el código para satisfacer sus necesidades específicas (Bessen, 2005; Franke y von Hippel, 2003). Lo habitual es que estas compañías adopten un modelo de negocio híbrido, mezclando productos y licencias propietarias y de código abierto (Bonaccorsi *et al*., 2006).

3.4 Proyectos de software de código abierto

Los proyectos de software de código abierto pueden observarse y analizarse a partir de toda la información públicamente disponible. Se puede decir que cualquier grupo de personas que desarrolle software y que suministre públicamente sus resultados bajo una licencia de código abierto constituyen un proyecto de software de código abierto. El recurso más importante de estos proyectos son los desarrolladores. El término desarrollador es bastante amplio e incluye a personas procedentes de diferentes ámbitos:

- Instituciones educativas: las Universidades y otras instituciones educativas producen una gran cantidad de software con propósitos educativos e investigadores, y una buena parte se distribuye conforme a la definición de código abierto. La Universidad produce continuamente una gran cantidad de programadores que todavía no se han incorporado al mundo laboral y que pueden dedicar una gran cantidad de esfuerzo al código libre. Algunos de los proyectos más famosos de código abierto comenzaron en ambientes universitarios, como Linux, BSD o Apache.
- Instituciones de investigación: muchas veces, resultados de la investigación se distribuyen bajo licencias más o menos permisivas.

- Distribuidores de software: los distribuidores de software de código abierto participan y contribuyen en proyectos de este tipo. Su motivación es extender su producto a nuevos usuarios, proporcionando nuevas funcionalidades o características no disponibles.
- Compañías comerciales: además de los distribuidores de software, grandes compañías comerciales como IBM o Hewlett Packard contribuyen en proyectos de código abierto.
- Usuarios corporativos: dado los enormes esfuerzos financieros que muchas compañías o administraciones públicas han de hacer a sus sistemas software, muchas se deciden a patrocinar proyectos de software de código abierto, ahorrándose así los cánones de licencias por el software propietario.
- Usuarios privados: los usuarios privados que utilizan software de código abierto se benefician de las mejoras que puedan aportar sobre el mismo. Muchos proyectos fueron iniciados por programadores de talento y son sostenidos por personas con habilidades de programación que dedican parte de su tiempo a mejorarlo.
- Gobiernos: muchos gobiernos han empezado a promover el uso de código abierto para infraestructuras críticas. Un buen ejemplo lo constituye el patrocinio que el Ministerio de Economía alemán realizó de GNU *Privacy Guard* (GPG) para reemplazar PGP (*Pretty Good Privacy*) en la autentificación de firmas digitales.

3.4.1 Ciclo de vida

Los proyectos de software de código abierto son como organismos vivos. No siguen unos patrones estrictos en sus distribuciones y pasan por diversos ciclos. A modo ilustrativo, un ciclo de vida típico sería el siguiente:

1. Alguien tiene una necesidad e intenta solucionarla.
2. Esta persona contacta con amigos y colegas para preguntarles sobre cómo solucionar el problema. Algunos tienen el mismo problema, pero sin solución.

3. Las personas interesadas comienzan a intercambiar información sobre el tema, creando así un boceto previo sobre el tema central del grupo.
4. Las personas interesadas dispuestas a gastar parte de su tiempo en solucionar el tema crean un proyecto informal.
5. Los miembros del proyecto continúan trabajando en el tema hasta alcanzar algunos resultados satisfactorios.
6. Los integrantes del proyecto deciden hacer el código accesible públicamente en un lugar accesible por el resto de personas. Incluso pueden anunciar su proyecto en listas de distribución o grupos de noticias. El proyecto Linux comenzó con un mensaje a una lista de distribución (http://groups.google.com/groups?selm=1991Oct5.054106.4647%40klaava.Helsinki.FI)
7. Otras personas reconocen algunas de sus necesidades en el proyecto y se interesan por una solución satisfactoria. Revisan los resultados logrados por el proyecto hasta el momento y, mirando el problema desde otra perspectiva, sugieren nuevas mejoras o deciden unirse a la comunidad.
8. El proyecto crece y la realimentación recibida ayuda a entender mejor el tema del grupo y a obtener estrategias para resolverlo.
9. Nuevas informaciones y recursos se integran en el proceso de desarrollo del proyecto. La solución crece y redirecciona el tema en líneas más exitosas.
10. El ciclo de investigación de soluciones se cierra y se retorna al punto 5.
11. La comunidad se encuentra ya establecida y reaccionará ante futuros cambios del mismo modo que emergió al principio.

Una clasificación habitual sobre las etapas de un proyecto de software de código abierto es: planificación, pre-alfa, alfa, beta, estabilidad y madurez.

- Planificación: todavía no se ha escrito nada de código y el ámbito del proyecto no está totalmente definido. Es sólo una idea. Tan pronto aparecen resultados tangibles en forma de código fuente, el proyecto entra en la siguiente etapa.

- Pre-alfa: se distribuye una versión muy preliminar del código, no demasiado legible. Todavía no está preparado para compilarse ni ejecutarse. Tan pronto como una cierta coherencia se hace visible en el código, el proyecto entra en la siguiente etapa.
- Alfa: el código distribuido toma forma y funciona parcialmente. Las primeras notas de desarrollo aparecen. El trabajo para expandir las prestaciones de la aplicación continúa. A medida que la cantidad de nuevas prestaciones disminuye, el proyecto entra en la siguiente etapa.
- Beta: el código se encuentra completo en cuanto a prestaciones, pero todavía contiene algunos fallos. Gradualmente se van eliminando, dando lugar a un software más fiable. Si el número de fallos es lo suficientemente bajo, el proyecto alcanza una versión estable, entrando en la siguiente etapa.
- Estabilidad: el software es suficientemente útil y fiable para su uso diario. Los cambios se aplican muy cuidadosamente y, habitualmente, para mejorar más la fiabilidad que su funcionalidad. Cuando no ocurren cambios significativos durante un largo período de tiempo, el proyecto entra en la siguiente etapa.
- Madurez: no hay apenas nuevos desarrollos y el software cumple con su misión de forma muy fiable. Los cambios se aplican con mucha precaución. Un proyecto permanece en su etapa de madurez hasta que se vuelve obsoleto o se sustituye por un software mejor.

Realmente son pocos los proyectos que alcanzan las etapas de estabilidad o madurez, bien porque no sobrepasan las anteriores, bien porque están en permanente actualización.

3.4.2 Modelos de negocio

Los proyectos de software de código abierto se han extendido ampliamente a lo largo de la última década y, tanto el software propietario como el de código abierto, siguen llamados a desempeñar un importante papel en el futuro. En cualquier caso, sí que puede afirmarse que existe un claro interés de las compañías de software por el paradigma que representa el software de código abierto. En particular, muchas de estas compañías tratan de incorporar

los mejores elementos de ambos modelos en sus estrategias de desarrollo. Aunque predecir el resultado final por la supremacía de uno de los modelos es difícil de aventurar, sí que resulta claro que el principal beneficiario será el consumidor en la forma de mayores posibilidades de elección y menores precios (Krogh *et al*., 2003).

A pesar de que las diferencias filosóficas entre el modelo propietario y el de código abierto son enormes, existen estrategias de negocio que tratan de integrar ambos esquemas. Existen varias posibilidades por medio de las cuales las empresas pueden conseguir beneficios bajo el paradigma del software de código abierto (Deek y McHugh, 2008).

Las licencias duales permiten al propietario de una licencia con *copyright* proporcionar distribuciones libres y abiertas a usuarios no comerciales, y de pago a usuarios comerciales. Este es el caso del sistema de bases de datos *MySQL*. La licencia no comercial y gratuita es una licencia GPL, que permite aprovechar la creatividad de una comunidad de soporte. La licencia comercial es una versión mejorada del software (*MySQL Pro*), más segura y fiable.

La consultoría en proyectos OSS constituye otra posibilidad de modelo de negocio. En este caso, se ofrece una amplia colección de soluciones de código abierto.

Proporcionar distribuciones y servicios de código abierto es el modelo de negocio de compañías como Red Hat, que construye y vende distribuciones software más que productos software. En particular, Red Hat crea y suministra sus propias distribuciones de Linux, así como formación, documentación y soporte para sus distribuciones.

Los modelos híbridos propietario y abierto también son posibles. La idea consiste en utilizar un software de código abierto, aplicarlo a la resolución de un problema y venderlo. Por ejemplo, el modelo de desarrollo de Google está orientado al uso del software de código abierto en general, si bien el producto final es propietario. En otros casos, las compañías patrocinan y dan soporte a desarrollos de código abierto.

Todas estas posibilidades de negocio tienen en común la necesidad de mantener una relación entre la compañía y la comunidad de soporte del software. En algunos casos, estas compañías se ven obligadas a alinear sus intereses con los de la comunidad de soporte. Es importante reseñar el hecho de que crear una comunidad de soporte no significa

necesariamente que desarrolladores se vean atraídos a formar parte de ellas, o que su interés permanezca en el tiempo. Por este motivo, los directivos y responsables software deben tener en cuenta de forma prioritaria cómo se va a gestionar la actividad de la comunidad de soporte. Por una parte es necesario atraer desarrolladores brillantes y nuevos talentos que contribuyan y mantengan el desarrollo de nuevas funcionalidades e ideas, pero son perder cierto grado de influencia y control sobre los futuros desarrollos. La capacidad de las compañías para gestionar adecuadamente la comunidad de soporte constituye por sí una importante ventaja competitiva (Deek y McHugh, 2008).

3.4.3 Ejemplos

Los siguientes ejemplos muestran algunos de los proyectos de software de código abierto más exitosos:

- **XFree86** (XFree86 Project, Inc., http://www.xfree86.org/): XFree86 es una implementación del sistema X Window System. Fue escrita originalmente para sistemas operativos Unix funcionando en ordenadores compatibles IBM PC. En la actualidad está disponible para muchos otros sistemas y plataformas. XFree86 es de código abierto y está publicado bajo la licencia XFree86 1.1.
- **KDE** (K Desktop Environment, http://www.kde.org): KDE es un entorno de escritorio gráfico e infraestructura de desarrollo para sistemas Unix y, en particular, Linux. La 'K' originariamente representaba la palabra “Kool”, pero su significado fue abandonado más tarde. Actualmente significa simplemente 'K', la letra inmediatamente anterior a la 'L' (inicial de Linux) en el alfabeto. Actualmente KDE es distribuido junto a muchas distribuciones Linux.
- **GNOME** (GNU Network Object Model Environment, http://www.gnome.org/): es un entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Window. Se encuentra disponible actualmente en más de 35 idiomas. Forma parte oficial del proyecto GNU. Surge en agosto de 1997 como proyecto liderado por los mexicanos Miguel de Icaza y Federico Mena para crear un entorno de escritorio completamente libre para sistemas operativos libres, en especial para GNU/Linux. Desde el principio,

el objetivo principal de GNOME ha sido proporcionar un conjunto de aplicaciones amigables y un escritorio fácil de utilizar. GNOME también es una palabra del idioma inglés que significa gnomo. En esos momentos existía otro proyecto anterior con los mismos objetivos, pero con diferentes medios: KDE. Los primeros desarrolladores de GNOME criticaban dicho proyecto por basarse en la biblioteca de controles gráficos Qt, cuya licencia (QPL), aunque libre, no era compatible con la licencia GPL de la FSF.

- **Apache** (http://www.apache.org/): El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en el código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a "*patchy server*" (un servidor "parcheado"). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de Apache Software Foundation. Apache presenta, entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayudase en su configuración. Apache tiene amplia aceptación en la red, siendo actualmente el servidor HTTP del 60% de los sitios web en el mundo

- **Linux** (http://kernel.org/): Linux es el núcleo o kernel del sistema operativo libre denominado GNU/Linux (también llamado Linux), que brinda una alternativa frente a sistemas operativos no libres como Unix y Windows. Este núcleo, escrito casi completamente en C con algunas extensiones GNU C, fue desarrollado por el hacker finlandés Linus Torvalds en un intento por obtener un sistema operativo libre similar a Unix que funcionara con microprocesadores Intel 80386. El proyecto nació en 1991 con un famoso mensaje en el grupo comp.os.minix de Usenet, que contenía lo siguiente: "Estoy haciendo un sistema operativo (gratuito) (sólo un hobby, no será nada grande y profesional como GNU) para clones AT 386(486)..." Muy pronto, los hackers de Minix aportaron ideas y código al núcleo Linux, y hasta hoy ha recibido

contribuciones de miles de programadores. Originalmente Linux era solamente el nombre del núcleo. El término "núcleo" (en inglés kernel) propiamente dicho se refiere al software de sistema de bajo nivel que provee una capa de abstracción sobre el hardware, control de discos y sistema de archivos, multitarea, balance de carga, comunicación en red y medidas de seguridad. Un núcleo no es un sistema operativo completo (tal y como se entiende el término normalmente). El sistema completo construido alrededor del núcleo Linux es conocido usualmente como el sistema operativo Linux, aunque hay quienes prefieren llamar GNU/Linux al sistema completo. La gente confunde a menudo núcleo con sistema operativo, llegando a ciertas inferencias incorrectas, como suponer por ejemplo, que Torvalds programa/coordina otros componentes del sistema, además del núcleo. Torvalds ha continuado liberando nuevas versiones del núcleo, consolidando aportes de otros programadores y haciendo cambios por su cuenta. Todas las versiones de Linux que tienen el número de sub-versión (el segundo número) par, pertenecen a la serie "estable", por ejemplo: 1.0.x, 1.2.x, 2.0.x, 2.2.x, 2.4.x y actualmente 2.6.x, mientras que las versiones con sub-versión impar, como la serie 2.5.x, son versiones de desarrollo, es decir que no son consideradas de producción. Mientras que Torvalds continúa liberando las últimas versiones de desarrollo, el mantenimiento de las ramas "estables", siempre algo más viejas, ha sido delegada a otros programadores. La rama estable actual es la 2.6.x.

- **Mozilla** (http://www.mozilla.org/): En el año 1998, Netscape liberó el código de Netscape Communicator y denominaron al nuevo proyecto Mozilla. Netscape, tras la estrategia de Microsoft de incrustar su navegador Internet Explorer a su sistema operativo Windows para dominar el mercado y ganar la guerra de navegadores, tuvo la idea de contraatacar a Microsoft liberando el código fuente de su navegador Netscape 4.7 y así convertirlo en un proyecto de software libre.

- **Moodle** (http://moodle.org/): Moodle es un sistema de gestión de cursos libre (Course Management System CMS) que ayuda a los educadores a crear comunidades de aprendizaje en línea. Moodle fue creado por Martin Dougiamas, quien era el administrador de WebCT en la Universidad Tecnológica de Curtin, y se basó en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye

en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo. Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer. La primera versión de la herramienta apareció el 20 de agosto de 2002 y, a partir de allí han aparecido nuevas versiones de forma regular. Hasta diciembre de 2006 la base de usuarios registrados incluye más de 19.000 sitios en todo el mundo y está traducido a más de 60 idiomas. El sitio más grande dice tener más de 170.000 estudiantes. La palabra Moodle era al principio un acrónimo de Modular Object-Oriented Dynamic Learning Environment (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular), lo que tiene algún significado para los programadores y teóricos de la educación, pero también se refiere al verbo anglosajón *noodle*, que describe el proceso de deambular perezosamente a través de algo y hacer las cosas cuando se antoja hacerlas, una placentera chapuza que a menudo lleva a la comprensión y la creatividad. Las dos acepciones se aplican a la manera en que se desarrolló Moodle y a la manera en que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso en línea. En términos de arquitectura, se trata de una aplicación web que puede funcionar en cualquier computador en el que se pueda ejecutar PHP. Opera con diversas bases de datos SQL, como por ejemplo MySQL y PostgreSQL. La licencia que utiliza Moodle es la GPL.

- **Latex** (http://www.latex-project.org/): es un procesador de textos formado por un gran conjunto de macros de TeX, escritas inicialmente por Leslie Lamport (LamportTeX) en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con LaTeX es comparable a la de una editorial científica de primera línea. LaTeX es software libre bajo licencia LPPL (Licencia Pública del Proyecto LaTeX), que no es compatible con GPL. LaTeX presupone una filosofía de trabajo diferente a la de los procesadores de texto habituales (conocidos como WYSIWYG, es decir, “lo que

ves es lo que obtienes") y se basa en comandos. Tradicionalmente, este aspecto se ha considerado una desventaja (probablemente la única). Sin embargo, LaTeX, a diferencia de los procesadores de texto de tipo WYSIWYG, permite a quien escribe un documento centrarse exclusivamente en el contenido, sin tener que preocuparse de los detalles del formato. Además de sus capacidades gráficas para representar ecuaciones, fórmulas complicadas, notación científica e incluso musical, permite estructurar fácilmente el documento (con capítulos, secciones, notas, bibliografía, índices analíticos, etc.), lo cual lo hace extremadamente cómodo y eficiente para artículos académicos y libros técnicos.

3.4.4 Fortalezas y debilidades de los proyectos de código abierto

Las principales fortalezas de los proyectos de software de código abierto se refieren a la frecuencia de publicación de nuevas versiones, la aportación de los usuarios y su escalabilidad.

- Frecuencia de lanzamiento de nuevas versiones: la publicación frecuente de nuevas versiones, combinado con políticas destinadas a mantener versiones estables y experimentales de manera concurrente, permite incrementar el potencial de realimentación de una gran variedad de usuarios. Normalmente, los usuarios de proyectos de software de código abierto suelen informar a cerca de los fallos que encuentran en las versiones experimentales. Para usuarios más interesados en una producción o desarrollo industrial se suministran las versiones estables del código. Algunos proyectos llegan al extremo de proporcionar versiones actualizadas al minuto.
- Realimentación de los usuarios: los proyectos de software de código abierto se caracterizan por un bucle de realimentación pequeño entre desarrolladores y usuarios. Desde que se informa de un *bug* desde la periferia de la comunidad hasta que se suministra un parche oficial que lo corrige pueden transcurrir tan sólo minutos u horas. Los Sistemas de Control de Versiones (CVS, Concurrent Versioning Systems) mantienen el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto, permitiendo que distintos desarrolladores

(potencialmente situados a gran distancia) colaboren ofreciendo versiones del código en permanente actualización. Estos ciclos de respuesta tan rápidos permiten que los miembros de la comunidad ayuden en los procesos de aseguramiento de la calidad del software, ya que ellos mismos son recompensados por los parches que corrigen los problemas que han detectado. Al ser el código abierto, los usuarios periféricos que encuentran los *bugs* pueden ayudar a resolverlos o, al menos, proporcionar casos y test que permitan a los desarrolladores del núcleo de la comunidad aislar los problemas rápidamente. La comunidad permite repartir en ella el esfuerzo de depuración del código, mejorando rápidamente la calidad del software (Schmidt, 2001).

- Escalabilidad: en proyectos de software se suele considerar un axioma: la llamada Ley de Brooks, que dice que añadir más programadores a un proyecto que se desarrolla con retraso lo retrasa aún más (Brooks, 1995). Es decir, que la productividad en el desarrollo de software no se incrementa con el aumento del número de desarrolladores. Esto se debe a que, al mismo tiempo, también crecen rápidamente los costes de coordinación y comunicación. En otras palabras, un equipo de 10 buenos desarrolladores puede producir un software de mayor calidad y con menos esfuerzo y coste que un equipo de 1000 desarrolladores. Por el contrario, la depuración del software mejora con el incremento de desarrolladores dedicados a comprobar el código. Es evidente que cuantas más personas se dediquen a testar el código, mayor será el número de *bugs* que son capaces de localizar. Los proyectos de software de código abierto son capaces de incumplir la ley de Brooks, adoptando una estructura tipo núcleo/periferia. En esta división del trabajo, un relativamente pequeño número de desarrolladores distribuidos por el mundo constituyen el núcleo humano del proyecto y son los responsables de garantizar la integridad del proyecto. Se encargan de revisar las contribuciones de los usuarios y las soluciones a los *bugs*, de añadir nuevas prestaciones y capacidades, y de vigilar día a día el progreso de las tareas y objetivos del proyecto. Por su parte, la periferia está formada por miles de miembros de la comunidad de usuarios, que ayudan en las labores de testeo y de depuración del software publicado periódicamente por el núcleo de la comunidad. Naturalmente, estas

divisiones son de carácter informal y los individuos pueden adoptar diferentes papeles en momentos diferentes a lo largo del ciclo de vida del proyecto.

En cuanto a las debilidades, están fundamentalmente relacionadas con la falta de organización formal y de responsabilidades:

- Comunicación: la comunicación es crucial en cualquier proyecto de software de código abierto y, al mismo tiempo, su mayor reto. Los mayores obstáculos para esta comunicación son las diferencias culturales, la ausencia de contacto personal y las diferencias horarias.
- Esfuerzos redundantes: la coordinación en proyectos de software de código abierto es baja y, a veces, ocurre que grupos de trabajo independientes desarrollan paralelamente las mismas tareas sin saber unos de otros. Si bien esta situación consume muchos recursos, tiene también la ventaja de proporcionar varias soluciones a un mismo problema. La elección entre varias alternativas ayuda a mejorar la calidad del software (Karels, 2003) y a mejorar el debate y la discusión.
- Falta de prioridades: dada la naturaleza distribuida de los proyectos de software de código abierto y su falta de liderazgo claro, las prioridades son inexistentes o fuertemente sesgadas hacia las apreciaciones de los colaboradores más influyentes. A veces, los proyectos son arrastrados a discusiones sin fin, si nadie puede imponer su opinión para acabar con la discusión. Muchos usuarios reclaman su derecho a participar en las decisiones técnicas aun cuando no tienen el conocimiento necesario.
- Falta de convenciones: los proyectos de software de código abierto no tiene reglas formales o convenciones escritas. Los recién llegados van gradualmente aprendiendo las reglas ocultas. Dado que es interesante contar con personas de talento, las dificultades de integración pueden ser muy perjudiciales para la buena marcha del proyecto.
- Dependencia de personas clave: en muchas ocasiones, el trabajo y las responsabilidades están excesivamente centralizados en un grupo de desarrolladores, lo que contrasta con la idea de comunidades muy distribuidas que se autogestionan. Esta dependencia crítica puede tener lugar por varios motivos. El más obvio es el nivel de

conocimiento requerido para entender todas las partes de un proyecto complejo. El esfuerzo para llegar a este nivel tan sólo lo realiza un grupo pequeño de desarrolladores. La falta de documentación contribuye a este hecho. Otra explicación puede ser el nivel de reconocimiento que los colaboradores individuales pueden llegar a recibir. Si sólo unos pocos colaboradores llegan a ser reconocidos por sus contribuciones, se favorece el papel dominante de este reducido grupo. La dependencia puede llegar a ser un serio problema si, por cualquier motivo, este reducido núcleo no puede continuar sus actividades.

- Liderazgo: el éxito de los proyectos de software de código abierto depende en gran medida de buenos líderes carismáticos. Deben dominar aspectos no sólo técnicos sino también relacionados con la comunicación, marketing y motivación. Gran parte del éxito de Linux se debe a las excelentes cualidades de liderazgo de su fundador Linus Torvalds. La escasez de buenos líderes es un factor limitante para el buen desarrollo de proyectos de código abierto.

3.5 Comunidades de software de código abierto

Los proyectos de software libre han ido creando a lo largo de los años sus propias herramientas y sistemas (también libres) para la ayuda en el proceso de desarrollo. Aunque cada proyecto sigue sus propias reglas y usa su propio conjunto de herramientas, hay ciertas prácticas, entornos y tecnologías que pueden considerarse como habituales en el mundo del desarrollo de software libre.

3.5.1 Participantes

Cada individuo que interactúa en un proyecto de software de código abierto podría considerarse un participante, aunque habría que distinguir entre usuarios y colaboradores. Los usuarios simplemente usan o se benefician de los resultados del proyecto, mientras que los colaboradores son los que invierten un esfuerzo en mejorar el producto.

- Los usuarios son aquellas personas que hacen uso de los servicios proporcionados por un proyecto, pero que no tienen intención de contribuir al mismo. A pesar de todo, constituyen una pieza clave para el buen desarrollo de los proyectos. Con el tiempo, muchos de estos usuarios pueden convertirse en colaboradores. Por ejemplo, cuando se lanza una nueva versión del software y deja de funcionar en su sistema. Este hecho motiva que el usuario envíe un mensaje del error a una lista de distribución para conseguir que alguien le solucione el problema, convirtiéndose de ese modo en colaborador. En otras ocasiones, el usuario puede tener conocimientos de programación y, a partir del código fuente, resolver él mismo el problema. Una vez solucionado puede optar por resolver el problema en su propio sistema o por enviar también la solución a una lista de distribución para la inclusión de su código en el código del proyecto. Aunque es más fácil para el usuario resolverlo sólo en su sistema y no interactuar con la lista de distribución, a largo plazo tendrá que resolver siempre el problema en próximas versiones de ese proyecto, con el inconveniente de que en la nueva versión no sirva la solución planteada anteriormente. Por tanto, existe un incentivo para incluir la solución en el código del proyecto. Además, estas contribuciones le proporcionan a su autor el reconocimiento de colegas, lo cual también es un premio en sí mismo.
- Colaboradores: los colaboradores son aquellas personas que tienen interés en el desarrollo del proyecto, siguen las discusiones en las listas de correo y hacen oír sus opiniones. Los colaboradores no son únicamente aquellos que contribuyen al código, sino que basta con participar en las discusiones y estimular el debate en la comunidad. Tan colaborador es el que responde a cuestiones generales del proyecto como los que envían preguntas sobre el proyecto, proporcionando información importante sobre su uso o su documentación. Ser un colaborador es participar en un proceso de aprendizaje. Distinguimos dos niveles:
    - Colaboradores principiantes: son personas que desean formar parte de un proyecto y participar en él. Llegan a ser colaboradores formando parte de la comunidad y actuando como aprendices. Para ellos es necesario que el resto de la comunidad los vea como colaboradores periféricos legítimos, permitiéndoles

participar a su nivel. Mediante la observación y la participación van adquiriendo conocimiento y habilidades y aprenden el contexto social del que formarán parte.

- Colaboradores veteranos: son miembros activos de la comunidad y que participan de sus actividades.

3.5.2 Tecnologías de desarrollo

La mayoría del software libre está realizado en lenguaje C, no sólo porque C es el lenguaje natural de toda variante de Unix (plataforma habitual del software libre) sino también su amplia difusión, tanto a nivel de programadores como a nivel de plataformas hardware (gcc es un compilador estándar instalado por defecto en casi todas las distribuciones de GNU/Linux).

Otros lenguajes que se le acercan bastante son C++, también soportado por gcc por defecto, y Java, con cierta semejanza y popular por permitir desarrollar para máquinas virtuales disponibles en gran variedad de plataformas.

A todo proyecto de desarrollo de programas le conviene tener archivada la historia del mismo. Por ejemplo, porque alguna modificación pudo producir un error oculto que se descubre tardíamente y hay que recuperar el original, al menos para analizar el problema. Si el proyecto lo desarrollan entre varias personas es necesario también registrar el autor de cada cambio por el mismo motivo. Si del proyecto van haciéndose entregas, es necesario saber exactamente qué versiones de cada módulo forman dichas entregas. Muchas veces, un proyecto mantiene una versión estable y otra experimental, como es el caso del kernel de Linux. Ambas hay que mantenerlas, corregir sus errores y transferir errores corregidos de una versión a la otra. Todo esto puede hacerse guardando y etiquetando convenientemente todas y cada una de las versiones de los ficheros. Lo que normalmente hace un sistema de control de fuentes, también llamado sistema de gestión de versiones, es registrar la historia de los ficheros como un conjunto de diferencias sobre un patrón, normalmente el más reciente, por eficiencia, etiquetando además cada diferencia con los meta-datos necesarios. Este tipo de sistemas se usa mucho en los proyectos de software de

código abierto, permitiendo que muchos programadores colaboren efectivamente, sin pisarse el trabajo, pero sin detener el avance de cada uno.

CVS (Concurrent Version Sytem) es un sistema de gestión de fuentes diseñado a finales de los 80, que es usado por abrumadora mayoría en los proyectos de código abierto (Fogel, 2001). Utiliza un repositorio central al que se accede según un sistema cliente/servidor. El administrador del sitio decide quienes tienen acceso al repositorio o a que partes del repositorio, aunque normalmente, una vez que un desarrollador ha sido admitido en el círculo de confianza, tiene acceso a todos los ficheros. Además puede permitirse un acceso anónimo sólo en lectura a todo el mundo. Este acceso anónimo es muy importante, ya que cualquier usuario ansioso de probar la última versión de un programa la puede extraer del CVS, descubrir errores y reportarlos, incluso en forma de parches con la corrección. También puede examinar la historia de todo el desarrollo.

Dado que el software de código abierto es un fenómeno que es posible debido a la colaboración de comunidades distribuidas, es necesario disponer de herramientas que hagan efectiva esa colaboración. Aunque inicialmente se utilizaron las noticias (news), hoy día existe la tendencia a preferir las listas de correo a los grupos de noticias. La razón principal ha sido el abuso con fines comerciales y la intrusión de gente despistada, que introduce ruido en las discusiones. Además las listas de correo ofrecen más control y pueden llegar a más gente. Los destinatarios han de suscribirse y cualquier dirección de correo es válida. El administrador de la lista puede elegir conocer quién se suscribe o dar de baja a alguien. Puede restringir las contribuciones a los miembros o puede elegir moderar los artículos, antes de que aparezcan. La administración de las listas tradicionalmente se ha hecho por correo electrónico, por medio de mensajes especiales con contraseña. Eso permite que el administrador no tenga acceso permanente a Internet, aunque cada vez eso es un fenómeno más raro, de modo que el gestor de listas de correo más popular hoy día (Mailman, the GNU mailing list manager) no puede ser administrado por correo electrónico y tiene que hacerse necesariamente vía Web. Las listas de correo juegan un papel crucial en el software libre, llegando en muchos casos a ser el único método de contribución.

Para facilitar el desarrollo de proyectos de código abierto existen sitios de soporte al desarrollo que proporcionan, de manera más o menos integrada, muchos de los servicios necesarios para llevar a cabo los proyectos de código abierto. Además, proporcionan servicios adicionales que permiten la búsqueda de proyectos por categorías y su clasificación según parámetros sencillos de actividad. Esto permite a los desarrolladores liberarse de la fatigosa tarea de montar una infraestructura de colaboración y centrarse en su proyecto. Uno de los más famosos es SourceForge (http://sourceforge.net/), que actualmente alberga más de 88.000 proyectos, aunque muchos de ellos no están vivos.

Los servicios que SourceForge proporciona a un proyecto son:

- Albergue para las páginas web del portal del proyecto, en la dirección proyecto.sourceforge.net donde se muestra el mismo al público.
- Opcionalmente, un servidor virtual que responda a direcciones de un dominio obtenido aparte.
- Tantos foros web y/o listas de correo como sean necesarios, según criterio de un administrador.
- Un servicio de noticias, donde los administradores anuncian novedades sobre el proyecto.
- Rastreadores (trackers), para informe y seguimiento de errores, peticiones de soporte, peticiones de mejoras o integración de parches. Los administradores dan una prioridad al asunto y asignan su solución a un desarrollador.
- Gestores de tareas. Similar a un rastreador, permite definir subproyectos con una serie de tareas. Estas tareas, además de una prioridad, tienen un plazo. Los desarrolladores a los que se les asignan pueden manifestar de vez en cuando un porcentaje de realización de la tarea.
- Un CVS con derechos iniciales de acceso para todos los desarrolladores.
- Servicio de subida y bajada de paquetes de software. Utilizándolo se tiene un registro de las versiones introducidas y se posibilita que los interesados reciban un aviso

cuando esto suceda. Además la subida implica la creación de varias réplicas en todo el mundo, lo que facilita la distribución.

- Servicio de publicación de documentos en HTML. Cualquiera puede registrarlos, pero sólo después de la aprobación por un administrador serán visibles.
- Copia de seguridad para recuperación de desastres, como rotura de disco, no de errores de usuario, como borrar un fichero accidentalmente.

Capítulo 4. Gestión del conocimiento soft y del conocimiento hard: participación y cosificación

4.1 Introducción

Tradicionalmente, el éxito de las comunidades virtuales se ha analizado desde la perspectiva de los modelos de éxito de sistemas de información, bajo la suposición de que una comunidad virtual no es más que una forma de sistema de información basado en Internet (Wachter *et al*., 2000). El modelo de éxito de un sistema de información desarrollado y posteriormente mejorado por DeLone y McLean (2003) postula que la calidad del sistema, la calidad de la información y la calidad del servicio afectan tanto a la satisfacción del usuario como a la intención de uso de un sistema de información, que a su vez son antecedentes directos del beneficio neto de un sistema de información (Figura 4).

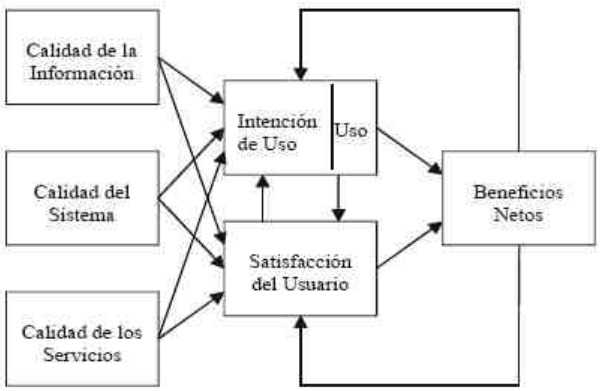


Figura 4. Modelo de éxito en sistemas de información (DeLone y McLean, 2003)

El modelo anterior se ha aplicado en el contexto de sistemas de información basados en Internet, como sistemas e-commerce (Molla y Licker, 2001), sistemas de toma de decisiones sobre web (Bharati y Chaudhury, 2004), sistemas de compra por internet (Ahn *et al*., 2004) o comunidades virtuales (Lin y Lee, 2006). El principal problema de este modelo estriba en que no considera las relaciones sociales entre los miembros de la comunidad ni la estructura de la misma. En este capítulo se pretende modelar la participación en comunidades virtuales desde la perspectiva del análisis de redes sociales y

determinar los principales antecedentes del éxito de la comunidad de desarrollo atendiendo a criterios de participación como herramienta de desarrollo del conocimiento *soft*.

4.2 Representación de las comunidades como redes sociales

Las redes sociales se definen como un conjunto finito de actores (individuos, grupos, organizaciones, comunidades, sociedades, etc) vinculados unos a otros a través de una relación o un conjunto de relaciones sociales. Las redes sociales se apoyan en el Análisis de Redes Sociales (ARS) el cual se centra en tomar las relaciones entre actores como el material sobre el cual se construye y se organiza el comportamiento social de los mismos. El punto de análisis deja de ser el individuo (egocéntrica) y pasan a serlo las relaciones (Wasserman *et al*., 1994), proporcionando un conjunto de métodos y técnicas para el estudio formal de las relaciones entre actores. Estas relaciones pueden basarse en las interacciones desarrolladas online, permitiendo el uso de técnicas de análisis de redes sociales para el estudio de comunidades virtuales (Garton *et al*., 1997). Numerosos investigadores reconocen que, en un sentido amplio, las redes sociales constituyen estructuras auto-organizativas de personas, información y comunidades (Kautz *et al*., 1997; Raghavan, 2002).

Las redes sociales son representadas mediante grafos y utilizan técnicas de la Teoría de Grafos para estudiar la estructura de las redes sociales (Yang y Chen, 2008). Un grafo está formado por nodos o vértices, que pueden representar individuos u organizaciones, y aristas o arcos, que representan las relaciones entre esos nodos. Si las relaciones entre los nodos no poseen un sentido se denominan aristas y dan lugar a grafos no dirigidos. Por el contrario, si existe un sentido definido en esas relaciones, entonces se denominan arcos y dan lugar a grafos dirigidos. Desde un punto de vista más formal, una red social puede representarse como un grafo G = (V, E), donde V denota un conjunto finito de vértices y E un conjunto finito de aristas o arcos tal que $E \subseteq V \times V$. Desde un punto de vista matricial, la red social se representa según la Ecuación (1)

$$M = (m_{i,j})_{n*n} \qquad where \quad n = |V| \quad , \quad m_{i,j} = \begin{cases} 1 & if\ (v_i, v_j) \in E \\ 0 & otherwise \end{cases} \tag{1}$$

En el caso de grafos valuados, existe un valor w(e) asociados a aristas o arcos tal que $w(e) = E x \Re$. Entonces la expresión matricial quedaría:

$$m_{i,j} = \begin{cases} w(e) & if\ (v_i, v_j) \in E \\ 0 & otherwise \end{cases} \qquad (2)$$

En el contexto de las comunidades virtuales, gran parte de la actividad tiene lugar a través de foros o listas de correo. Algunos autores utilizan el número de desarrolladores que envían mensajes a estos foros o el número de contribuciones como una medida del éxito de una comunidad (Preece, 2001). Normalmente, los mensajes almacenados en estos foros o listas de correo son almacenados y son públicamente accesibles en proyectos de software de código abierto. Pueden ordenarse por autores, mensajes, fechas o hilos de discusión (Figura 5, izquierda). La forma más interesante de analizar estos foros es ordenándolos por hilos de discusión, puesto que sigue la secuencia lógica de las interacciones entre usuarios.

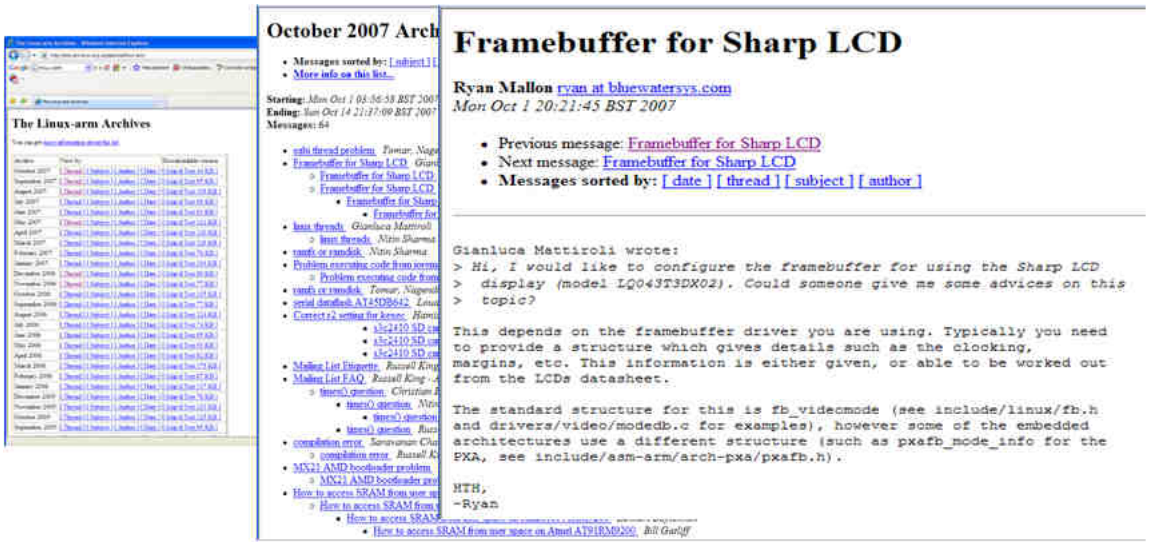


Figura 5. Foro de discusión del proyecto ARM Debian Linux en Octubre de 2007

Normalmente, un hilo de discusión se inicia con una persona que lanza un mensaje inicial a la comunidad pidiendo ayuda en un tema determinando, proponiendo una mejora o una idea nueva. A partir de aquí, otros miembros de la comunidad pueden decidir contestar a ese mensaje inicial aportando soluciones, alternativas, fuentes de información, o simplemente aportando nuevas consideraciones sobre el problema planteado. La Figura 5

muestra, en su parte central, cómo se organizan los sucesivos mensajes que forman parte del hilo de discusión y, a la derecha, la respuesta a uno de los mensajes de un hilo.

A través de los hilos de discusión los usuarios de la comunidad se enganchan a un proceso de conceptualización que puede dar lugar a una innovación colectiva y la creación de nuevo conocimiento. Asimismo, la participación en hilo permite que usuarios inexpertos vayan ganando conocimiento, habilidades y experiencia a través de un aprendizaje contextual. Por estos motivos, los hilos de discusión se utilizarán como la unidad básica a partir de la cual tienen lugar las interacciones entre los usuarios (Jones *et al*., 2004). Para clasificar los hilos de discusión se puede usar simplemente el tamaño (número de mensajes que contiene), el número de mensajes por hilo (Bonacci, 2004), o un conjunto de medidas más complejas como valor medio y desviación típica de mensajes por hilo, o el número de hilos sin respuesta (Toral *et al.*, 2009a).

Desde la perspectiva del análisis de redes sociales, V viene dado por el conjunto de usuarios y desarrolladores que participan dentro de la comunidad y E es el conjunto de interacciones que tienen lugar a través de los hilos de discusión, que es la unidad básica considerada (Jones *et al*., 2004). Una de las peculiaridades de los hilos de discusión es que contiene todo el contexto en el transcurre el debate, animando de este modo la participación (Kuk, 2006). Responder a un hilo de discusión es cognitivamente mucho más complejo que responder a un simple mensaje, ya que la respuesta tiene en cuenta normalmente todo el flujo de mensajes previos a la hora de elaborar una respuesta coherente con el hilo (Knock, 2001). Este es el motivo por el que se considerará que un usuario o miembro de la comunidad que escriba un mensaje en un hilo de discusión estará interaccionando con todos aquellos usuarios y miembros que previamente hubiesen escrito en ese hilo. En base a esto, el grafo resultante de representar la comunidad como una red social poseerá las siguientes características:

- Grafo dirigido. El sentido del arco que une dos vértices vendrá dado por el flujo de información entre dos miembros de la comunidad. Así pues, el sentido va desde aquel miembro que envía un nuevo mensaje a todos aquellos que previamente habían participado dentro del hilo de discusión.

- Grafo valuado. Un usuario de la comunidad puede participar varias veces dentro de un mismo hilo, o contestar a otros miembros varias veces en el mismo o en hilos diferentes. Por tanto, cada arco lleva un valor que indica el número de interacciones entre los dos vértices que une.

La Figura 6 ilustra la representación como grafo de la comunidad del proyecto ARM Debian Linux durante el año 2007. Por motivos de claridad, no se ha representado junto a cada nodo el alias que usa cada usuario, normalmente asociado a una dirección de correo electrónico. Las cabeceras de los mensajes deben ser procesadas para evitar duplicidad de alias o de correos (Bird *et al.*, 2006). Tampoco se explicitan los valores asociados a cada arco y que indican el número de interacciones entre dos miembros de la comunidad.

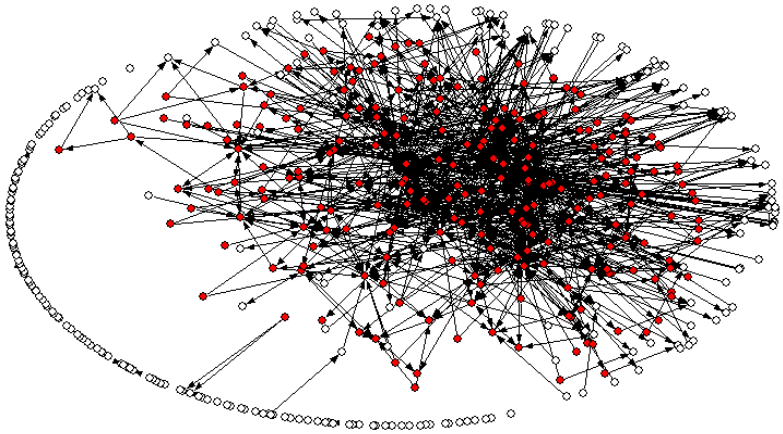


Figura 6. Red social de la comunidad de ARM Debian Linux durante 2007

Las principales características que pueden extraerse de una red social son:

- Tamaño: dado por el número de vértices o nodos de la red.
- Densidad: es una media del número de líneas en una red simple, expresada como una proporción del número máximo posible de líneas. El principal problema de esta definición es que no tiene en cuenta las líneas valuadas con valor superior a 1 y que depende del tamaño de la red. Una medida diferente de densidad se basa en la idea del grado de un nodo, que es el número de líneas que inciden (grado de entrada) o salen (grado de salida) de él (Toral *et al*., 2009b). Mayores grados de nodos producen redes

más densas, porque los nodos involucran más arcos, y el valor medio del grado de los nodos de una red no es una medida dependiente del tamaño de la red.

- Centralidad cercana (Closeness centralization): es un índice de centralidad basado en el concepto de distancia. La centralidad cercana de un nodo se calcula considerando el total de distancias entre un nodo y todos los demás nodos, donde la distancia más larga ofrece una menor puntuación de centralidad cercana. La centralidad cercana es un índice definido para toda la red y se calcula como la variación en la centralidad cercana de los vértices dividida por la variación máxima posible en la puntuación de centralidad cercana en una red del mismo tamaño (Toral *et al*., 2009b).
- Grado de Intermediación (Betweenness): es una medida de la centralidad que reside en la idea de que un nodo es más central en la medida en que actúe como intermediario en una red de comunicación (Nooy *et al*., 2005). Es decir, la centralidad de un nodo depende de la medida en la que es necesario como enlace para facilitar la conexión de otros nodos dentro de la red. Si se define una geodésica como el camino más corto entre dos nodos, la centralidad de intermediación de un vértice es la proporción de todas las geodésicas entre pares de nodos que incluyen este nodo, y la centralidad en la intermediación de una red es la variación en la centralidad de intermediación de los nodos dividida por la máxima variación posible en la centralidad de intermediación en una red del mismo tamaño. Desde la perspectiva del análisis de enlaces esta medida permite detectar pasarelas que conectan redes separadas (Faba-Pérez *et al*., 2005).

4.3 Características de la participación en comunidades virtuales

La estructura de las comunidades de soporte de proyectos de software de código abierto no es plana, tal y como se planteaba en el modelo tipo bazar de participación plena (Raymond, 1998). En su lugar, las comunidades de soporte muestran una estructura por capas, de modo que en la parte central se encuentran los miembros más activos y, a medida que nos vamos alejando, van apareciendo usuarios menos activos, usuarios periféricos, e incluso usuarios pasivos (Ye *et al*., 2005). Los miembros que constituyen el núcleo de la comunidad poseen una especial relevancia, ya que son los encargados de fomentar la

participación y de conseguir que miembros noveles vayan adquiriendo la destreza suficiente como para, con el tiempo, pasar a formar parte de ese núcleo. En este sentido, tiene especial relevancia su papel como intermediadores o 'brokers' de conocimiento.

4.3.1 Desigualdad participativa

Numerosos estudios han demostrado que en los proyectos de software de código abierto la mayor parte de las aportaciones son realizadas por un pequeño porcentaje de individuos, a pesar de que existan cientos o miles de usuarios de la comunidad de soporte (Toral *et al*., 2010a). Esta concentración es lo que se denomina desigualdad participativa (Kuk, 2006). La Figura 7 muestra el porcentaje de la comunidad frente al número de contribuciones generada. Más de 60% de la comunidad no realiza ninguna contribución y está constituida por usuarios pasivos. Este porcentaje es aún mayor en comunidades menos especializadas (Zhang y Stock, 2001).

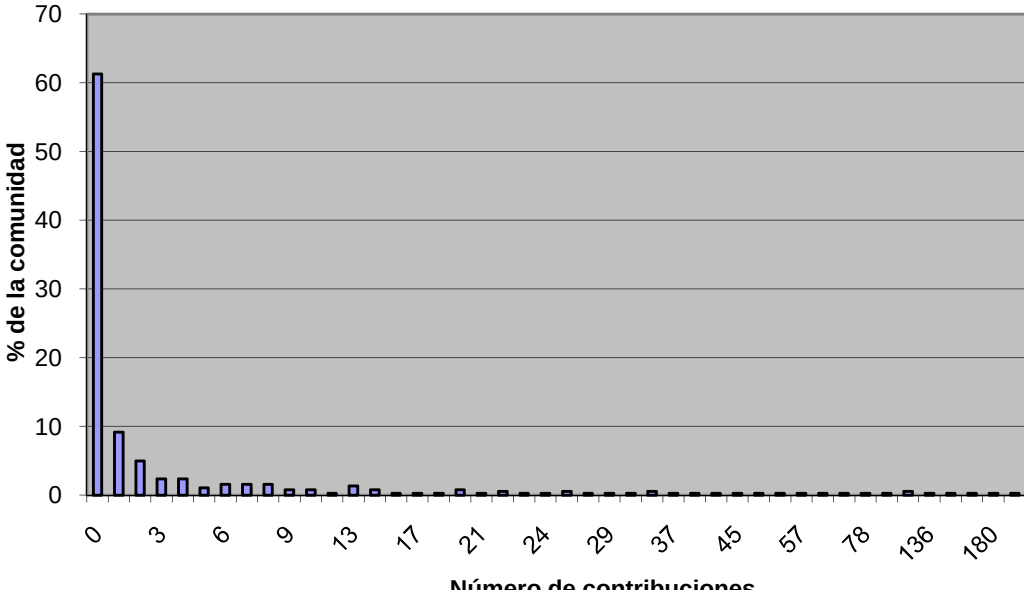


Figura 7. Distribución de las contribuciones para la comunidad ARM Debian Linux durante 2007

Una forma de establecer una medida del grado de desigualdad en las aportaciones al proyecto por parte de los desarrolladores es expresarlo mediante el coeficiente de Gini (1936). Este coeficiente varía entre 0 y 1: cuando hay poca concentración el coeficiente se

aproxima a cero, mientras que si la mayoría de las aportaciones se concentran en unos pocos desarrolladores, el coeficiente se aproxima a 1 (Dixon *et al*., 1987).

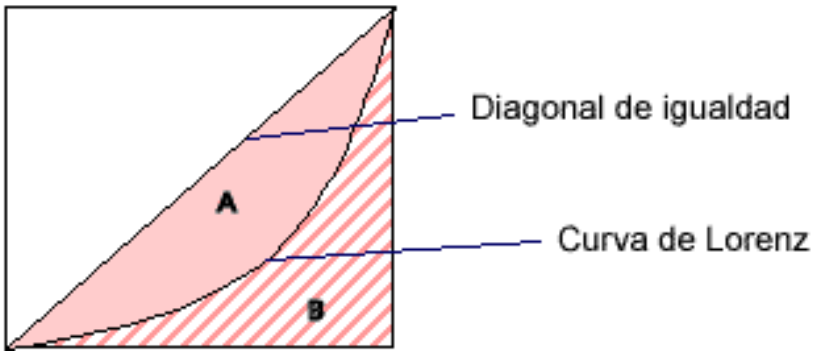


Figura 8. Áreas para calcular el coeficiente de Gini

El coeficiente de Gini se basa en la curva de Lorenz, que es una curva de frecuencia acumulada que compara la distribución empírica de una variable con la distribución uniforme (de igualdad) (Figura 8). Esta distribución uniforme está representada por una línea diagonal. Cuanto mayor es la distancia, o más propiamente, el área comprendida entre la curva de Lorenz y esta diagonal, mayor es la desigualdad.

Gráficamente, el coeficiente de Gini se define como el ratio entre la superficie encerrada por la diagonal de igualdad y la curva de Lorentz, y la superficie encerrada entre la diagonal de igualdad y la línea de perfecta desigualdad:

$$G = \frac{A}{A + B} \qquad (3)$$

Matemáticamente, se calcula como:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left| x_i - x_j \right|}{2n^2 \mu} \qquad (4)$$

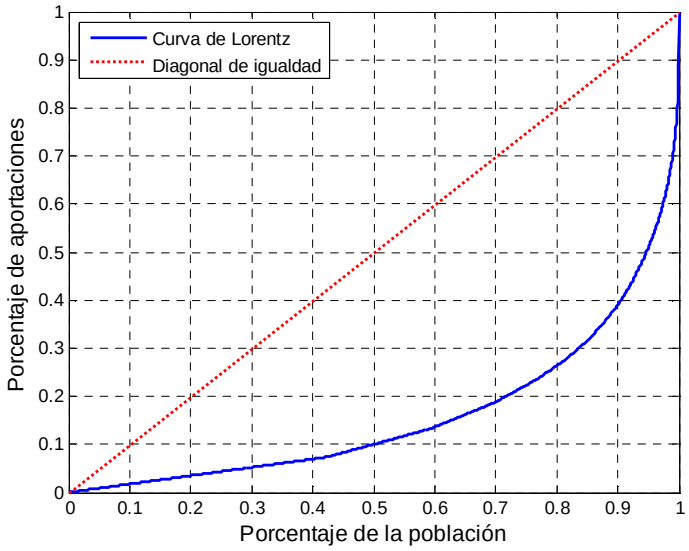


Figura 9. Curva de Lorentz y diagonal de igualdad para las aportaciones a la lista de distribución de la comunidad ARM Debian Linux

4.3.2 Estructura

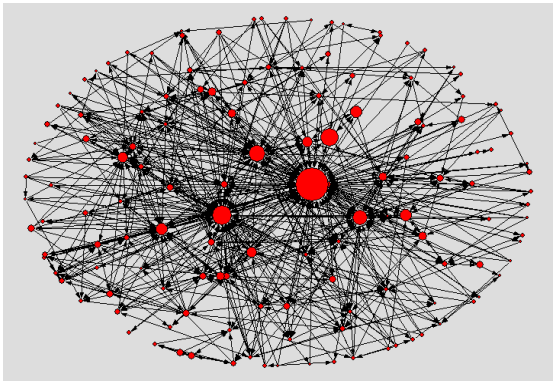
Como consecuencia de la desigualdad participativa, las comunidades de código abierto pueden dividirse en tres grupos (Mockus *et al*., 2002; Xu *et al*., 2005):

- Núcleo de la comunidad: Son los responsables de guiar y coordinar el desarrollo del proyecto de software de código abierto. Normalmente se encuentran involucrados en el proyecto durante un largo período de tiempo y llevan a cabo contribuciones significativas. Dentro de este grupo se incluyen líderes, moderadores y gestores de conocimiento.
- Miembros activos. Realizan contribuciones de forma periódica.
- Miembros periféricos. Sólo contribuyen ocasionalmente. Sus contribuciones son irregulares y su tiempo de involucración en el proyecto es corto y esporádico.

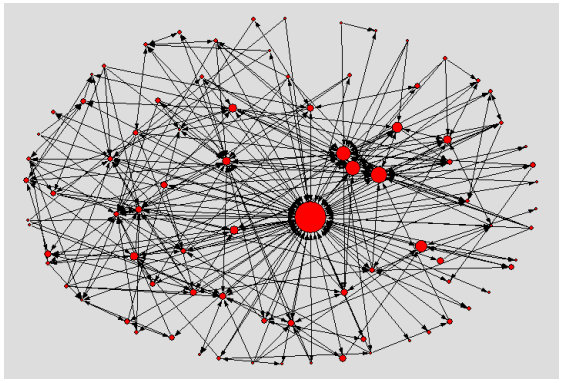
En las comunidades de soporte de proyectos software de código abierto, la información es pública y la participación está abierta a todo el mundo. No se exige ningún nivel predeterminado de experiencia para enviar un mensaje, que es la forma en la que los recién

llegados suelen comenzar su participación. No obstante, sí que es recomendable un cierto nivel de experiencia para responder cuestiones y participar en los temas de discusión. La cantidad de tiempo requerida para pasar de ser un miembro novel a un usuario experto depende del grado de involucración de cada individuo, si bien algún estudio revela que ese tiempo guarda relación directa con la cantidad de tiempo dedicado a leer y navegar por los recursos de que dispone la comunidad (Maybury, 2001)

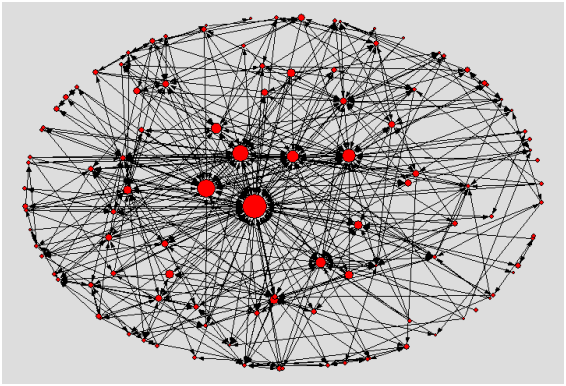
Tampoco es obligatoria la participación. De hecho, los denominados usuarios parásitos (conocidos como *free riders*) son tolerados (Wasko y Faraj, 2005). Los usuarios parásitos pueden definirse como aquellos miembros de la comunidad que disfrutan y hacen uso de los bienes de la comunidad, pero que no contribuyen ni a su desarrollo ni a su mejora. A pesar de ser tolerados, una presencia excesiva de usuarios parásitos puede constituir una amenaza para la comunidad. Para que la comunidad pueda sobrevivir en el futuro es necesario que un porcentaje de usuarios periféricos alcancen poco a poco el núcleo de la comunidad. El proceso por el cual tiene lugar ese tránsito se denomina, como anteriormente se comentó, participación periférica legítima (Lave *et al*., 1991; Fox, 2000). En este proceso, los usuarios noveles aprenden el funcionamiento de la comunidad a través de la participación, adquiriendo el lenguaje, valores y normas de la comunidad (Ducheneaut, 2005). Se trata además de un aprendizaje contextualizado, ya que se produce dentro de una discusión o problema concreto. El usuario novel va de ese modo aprendiendo de la experiencia de los más expertos. Para que el proceso descrito funcione, es imprescindible que miembros del núcleo de la comunidad desarrollen una labor de intermediación o de gestión de conocimiento respecto a otros miembros de la comunidad (Sowe *et al*., 2006). Actúan como intermediarios entre miembros expertos y usuarios periféricos, permitiendo que el conocimiento se distribuya entre los usuarios interesados. Una excesiva polarización de la comunidad hacia los miembros expertos puede conducir a que el debate se centre únicamente en las cuestiones que interesan a este reducido grupo, impidiendo la integración de nuevos miembros (Toral *et al*., 2009a).



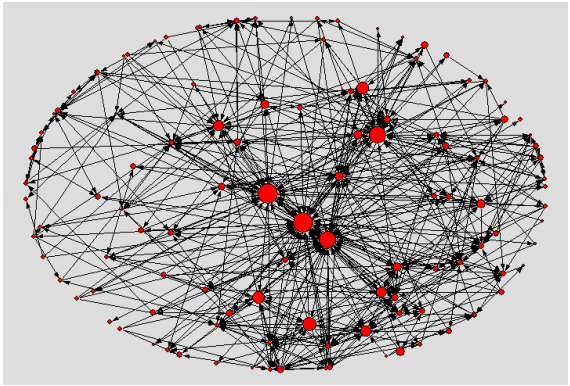
(a) 2002



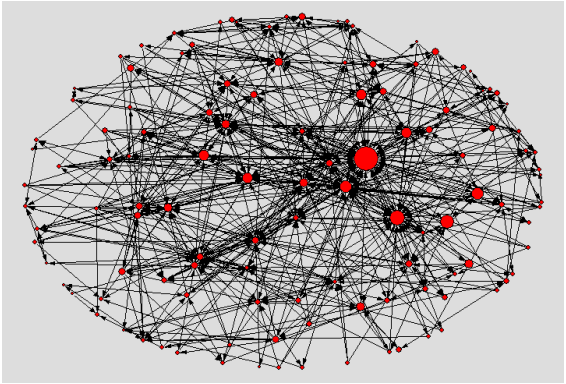
(b) 2003



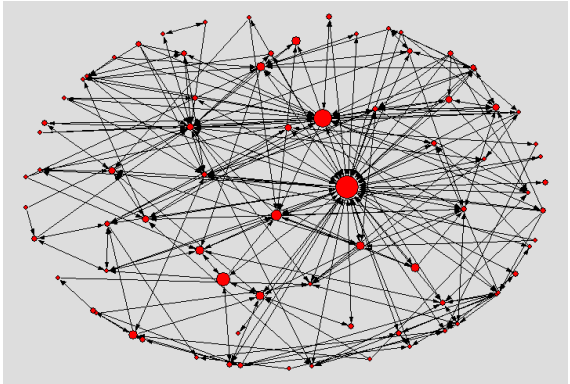
(c) 2004



(d) 2005



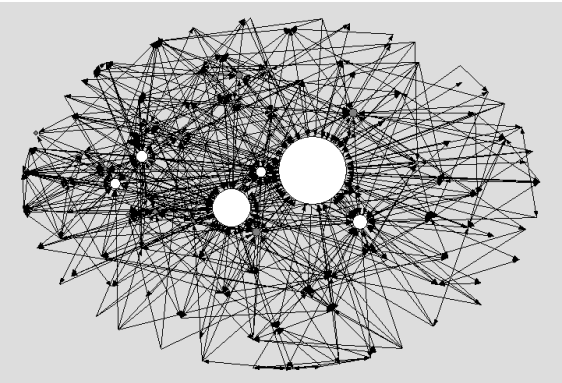
(e) 2006



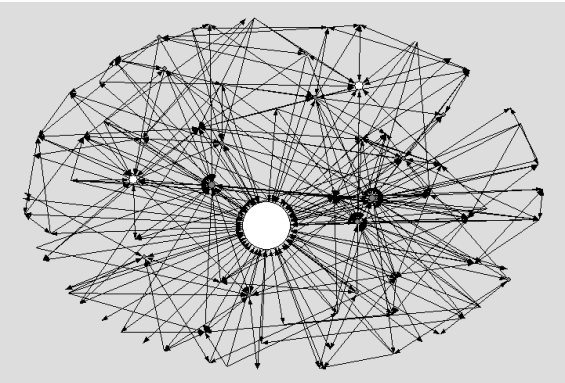
(f) 2007

Figura 10. Evolución de los miembros activos de la comunidad ARM Linux entre 2002 y 2007

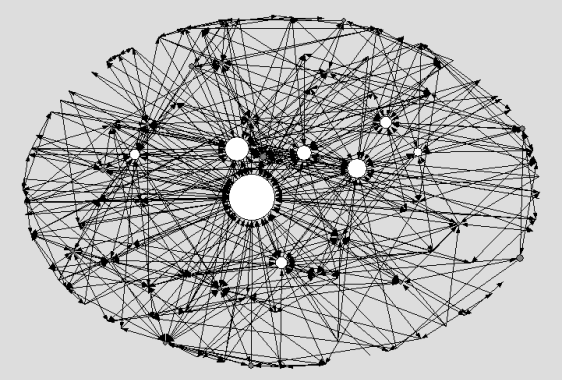
Desde un punto de vista macro-estructural, las redes pueden particionarse en sub-redes atendiendo a varias propiedades de los nodos. Por ejemplo, utilizando el grado de salida de los nodos pueden diferenciarse los usuarios activos de los miembros periféricos. La Figura 10 ilustra la evolución de los usuarios activos entre los años 2002 y 2007. Se han considerado miembros activos todos aquellos nodos cuyo grado de salida es mayor que el valor medio del grado de salida del conjunto de nodos de la red original.



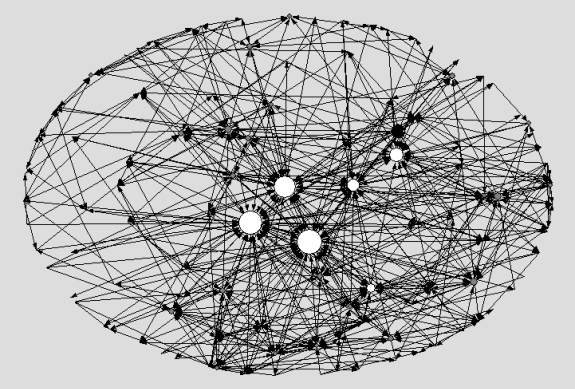
(a) 2002



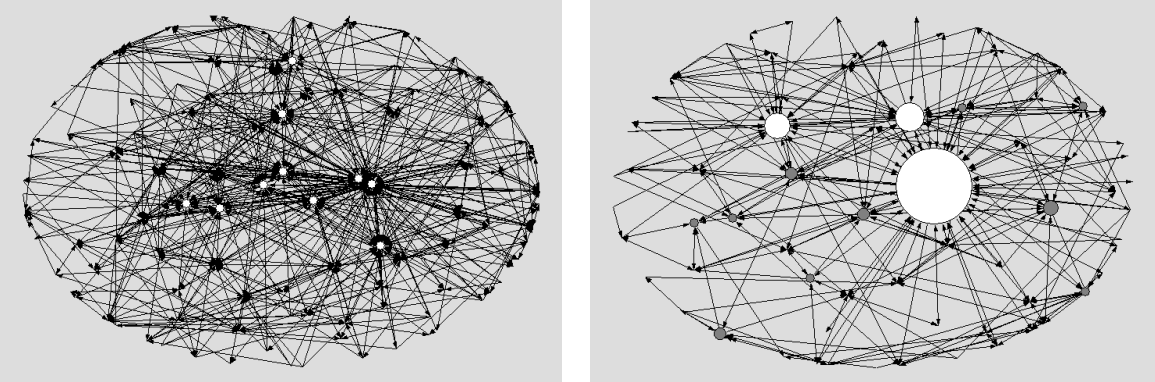
(b) 2003



(c) 2004



(d) 2005



(e) 2006 **(f) 2007**

Figura 11. Evolución de los gestores de conocimiento de la comunidad ARM Linux entre 2002 y 2007

Otra propiedad que puede utilizar para particionar la red original es el grado en el que un nodo actúa como mediador entre otros nodos de la red. Un gestor o broker se define como el nodo intermedio de una tríada, siendo una tríada un conjunto de tres vértices interconectados. En el contexto de las comunidades de código abierto, serían aquellos miembros que actúan como mediadores entre expertos y usuarios noveles. La Figura 11 ilustra la evolución de los gestores de conocimiento para la misma comunidad anterior entre los años 2002 y 2007, tomando con brokeres aquellos vértices que desempeñan la labor de intermediación más de 4 veces (Toral *et al*., 2010a).

Desde una perspectiva micro-estructural, el punto de interés se encuentra en la relevancia en el papel desempeñado por determinados nodos atendiendo a algún criterio. En el caso de la Figura 10 y la Figura 11 el tamaño de los nodos es proporcional al grado de salida y a la labor de intermediación, respectivamente. Este análisis micro-estructural permite identificar las personas clave dentro de la red.

4.4 Hipótesis y modelo de participación

El modelo propuesto trata de identificar los principales antecedentes del éxito de las comunidades de soporte de software de código abierto desde la perspectiva de la participación y el análisis de redes sociales. La propia medida del éxito de una comunidad constituye de por sí una cuestión abierta. Desde el punto de vista de los sistemas de

información, el modelo más ampliamente utilizado es el propuesto por DeLone y McLean (2003), que considera el éxito en función de los beneficios aportados a los usuarios. En el ámbito de las comunidades virtuales suelen usarse medidas como el nivel de actividad (contribuciones de usuarios y desarrolladores), número de miembros y efectividad del equipo gestor (Preece, 2001; Crowston *et al*., 2003).

Una de las principales característica medibles en una red social es su cohesión, que suele medirse a partir de la idea de densidad. La densidad expresa el porcentaje de conexiones de la red respecto al total posible con el mismo número de nodos y totalmente conectada, aunque también puede medirse a partir del grado de salida de los nodos de la red (Nooy *et al*., 2005). La cohesión de una red favorece la diseminación del conocimiento y atrae nuevos miembros a la comunidad (Hagedoorn y Duysters, 2002). Por otro lado, también favorece la involucración del núcleo de la comunidad y la incorporación de miembros noveles. En base a estos razonamientos, se postulan las siguientes hipótesis:

> H1: La cohesión de la red tiene un impacto positivo sobre el núcleo de la comunidad
>
> H2: La cohesión de la red tiene un impacto positivo sobre el éxito de la comunidad

La estructura de la comunidad de los proyectos de código abierto es una estructura en capas, donde la desigualdad participativa supone que entre un 45% y un 90 % de los usuarios no realizan contribuciones (Nonnecke y Preece, 2000). Las proporciones entre núcleo, miembros activos y miembros periféricos determinan la estructura a nivel de participación. Las estructuras basadas en la desigualdad participativa favorecen la coordinación de las actividades del proyecto software subyacente y el establecimiento de un núcleo fuerte y coordinado, demostrando resultar beneficiosas para su desarrollo futuro (Kuk, 2006).

En base a estos razonamientos, se postulan las siguientes hipótesis:

> H3: Las estructuras basadas en la desigualdad participativa tienen un impacto positivo sobre el núcleo de la comunidad

H4: Las estructuras basadas en la desigualdad participativa tienen un impacto positivo sobre el éxito de la comunidad

Las estructuras anteriores conducen a redes más o menos centralizadas. Desde una perspectiva ego-céntrica, los individuos tienen un mayor acceso a la información y mejores oportunidades en la medida en que ocupen posiciones más centrales. En el contexto de las comunidades de práctica, la idea de centralidad de cercanía (*closeness centrality*) presenta el inconveniente de basarse en la idea de distancia respecto al resto de nodos de la red. Pero como se ha indicado anteriormente, no todos los nodos desempeñan el mismo papel. Por este motivo se ha recurrido a la centralidad de intermediación (*betweeness centrality*), que define la centralidad de un individuo en la medida en que actúe como intermediario dentro de la red. Es decir, la medida en la que resulta necesario como nexo en la cadena de contactos que facilite la distribución de la información en la red. En este contexto, se postula que un proyecto con una elevada centralidad de intermediación consigue que un elevado número de miembros activos trabaje activamente con el núcleo de la comunidad y beneficie el intercambio de información (Rowley *et al*., 2000), dando lugar a la siguiente hipótesis:

H5: La centralidad de intermediación de la red tiene un impacto positivo sobre el éxito de la comunidad

Finalmente, el núcleo de la comunidad es crítico para garantizar las relaciones y guiar el desarrollo y el debate entre los miembros de la comunidad (Toral *et al*., 2010b). Son los engargados asimismo de fermentar la participación (Koh, 2007). Esto significa que no sólo son los miembros más activos, sino también los que realizan una labor más intensa como broeres o gestores de conocimiento (Sowe *et al*., 2006). En consecuencia, se propone la hipótesis:

H6: La presencia de un núcleo activo de miembros tiene un impacto positivo sobre el éxito de la comunidad

La Figura 12 detalla el modelo propuesto de éxito basado en la participación, incluyendo las hipótesis propuestas.

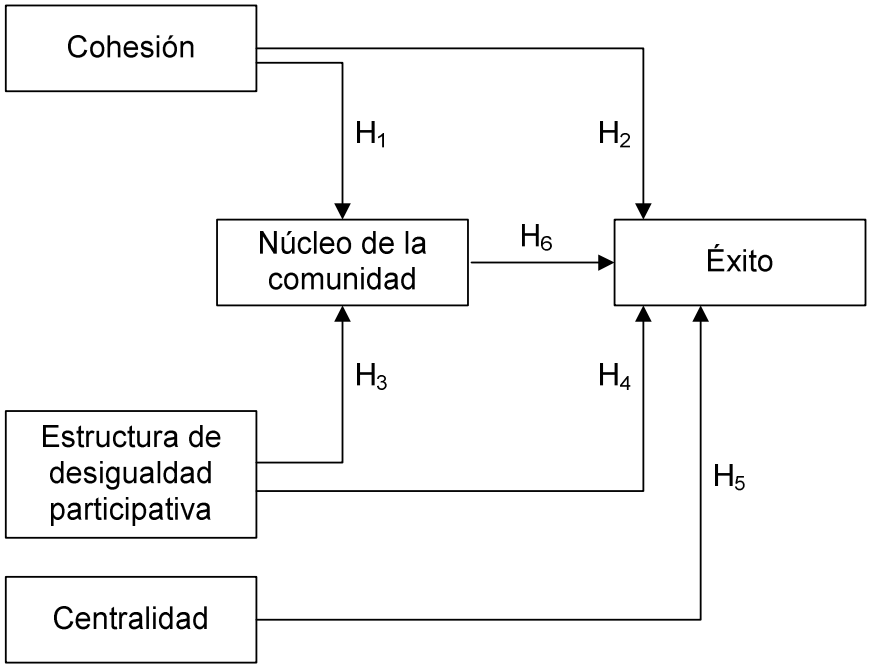


Figura 12. Modelo de éxito de las comunidades basado en la participación

4.5 Análisis del proceso de cosificación

Las comunidades de práctica constituyen una herramienta adecuada para gestionar la dualidad existente entre el conocimiento *soft* y el conocimiento *hard*. En particular, permiten crear y compartir conocimiento de una manera efectiva cuando se trata con problemas no estructurados (Kankanhalli *et al*., 2003). Compartir conocimiento significa transformar el conocimiento individual en un conocimiento colectivo y que además se almacene dentro de la organización (Liebowitz, 2001). Las interacciones sociales favorecen que el conocimiento pueda surgir de la práctica (van den Hooff y Huysman, en prensa), pero es necesario que también quede almacenado como bien de la comunidad. El análisis de los aspectos *hard* del conocimiento gestionado por las comunidades de soporte de proyectos de software de código abierto requiere el uso de técnicas de análisis semántico sobre los contenidos almacenados y públicamente disponibles. Debido a la ausencia de estudios en este sentido, se propone un análisis factorial exploratorio que

permite identificar las principales dimensiones subyacentes en el conocimiento explicitado y almacenados por las comunidades.

Capítulo 5. Metodología

5.1 Análisis factor

El Análisis Factorial es el nombre genérico que se da a una clase de métodos estadísticos multivariantes cuyo propósito principal es sacar a la luz la estructura subyacente en una matriz de datos. Analiza la estructura de las interrelaciones entre un gran número de variables, no exigiendo ninguna distinción entre variables dependientes e independientes. Utilizando esta información, calcula un conjunto de dimensiones latentes, conocidas como factores, que buscan explicar dichas interrelaciones. Es, por lo tanto, una técnica de reducción de datos, dado que si se cumplen sus hipótesis, la información contenida en la matriz de datos puede expresarse, sin mucha distorsión, en un número menor de dimensiones representadas por dichos factores.

El Análisis Factorial puede ser exploratorio o confirmatorio. El análisis exploratorio se caracteriza porque no se conocen a priori el número de factores y es en la aplicación empírica donde se determina este número. Por el contrario, en el análisis de tipo confirmatorio los factores están fijados a priori, utilizándose contrastes de hipótesis para su corroboración (Martínez Torres y Toral, 2010; Toral y Martínez Torres, 2010c).

Matemáticamente, sean X_1, X_2,…, X_p las p variables objeto de análisis, que supondremos tipificadas, medidas sobre n individuos, lo que nos proporciona la siguiente matriz de datos:

$$X = \begin{bmatrix} X_{11} X_{12} \cdots\cdots X_{1p} \\ X_{21} X_{22} \cdots\cdots X_{2p} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ X_{n1} X_{n2} \cdots\cdots X_{np} \end{bmatrix} \qquad (5)$$

El modelo del Análisis Factorial viene dado habitualmente por las ecuaciones:

$$X_1 = a_{11}F_1 + a_{12}F_2 + \ldots + a_{1k}F_k + u_1$$

$$X_2 = a_{21}F_1 + a_{22}F_2 + \ldots + a_{2k}F_k + u_2$$

…… (6)

$$X_p = a_{p1}F_1 + a_{p2}F_2 + \ldots + a_{pk}F_k + u_p$$

donde F_1, … , F_k (k<<p) son los factores comunes, u_1, …, u_p los factores únicos o específicos y los coeficientes $\{a_{ij}; i=1, \ldots, p; j=1, ..., k\}$ las cargas factoriales. Los factores únicos o específicos representan la parte aleatoria que es independiente de los factores.

Se supone, además, que los factores comunes están a su vez estandarizados [$E(F_i) = 0$; $Var(F_i) = 1$], los factores específicos tienen media 0 y están incorrelados [$E(u_i) = 0$; $Cov(u_i,u_j) = 0$ si $i \neq j$; j, i=1, …, p] y que ambos tipos de factores están incorrelados [$Cov(F_{i,uj}) = 0$, $\forall i=1,..,k$; j=1, …, p].

Si, además, los factores están incorrelados [$Cov(F_i,F_j) = 0$ si $i \neq j$; j, i=1, …, k] estamos ante un modelo con factores ortogonales. En caso contrario el modelo se dice que es de factores oblicuos.

Expresado en forma matricial

$$x = Af + u \quad X = FA' + U \qquad (7)$$

donde:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots\ldots \\ \ldots\ldots \\ x_p \end{bmatrix} \qquad f = \begin{bmatrix} F_1 \\ F_2 \\ \ldots\ldots \\ \ldots\ldots \\ F_k \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \ldots\ldots \\ \ldots\ldots \\ u_p \end{bmatrix}$$

X la matriz de datos,

$$A = \begin{bmatrix} a_{11}\, a_{12} \cdots\cdots a_{1k} \\ a_{21}\, a_{22} \cdots\cdots a_{2k} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{p1}\, a_{p2} \cdots\cdots a_{pk} \end{bmatrix}$$

es la matriz de cargas factoriales y

$$F = \begin{bmatrix} f_{11} f_{12} \ldots\ldots f_{1k} \\ f_{21} f_{22} \ldots\ldots f_{2k} \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ f_{p1} f_{p2} \ldots\ldots f_{pk} \end{bmatrix}$$ es la matriz de puntuaciones factoriales.

Utilizando las hipótesis anteriores se tiene que:

$$Var(X_i) = \sum_{j=1}^{k} a_{ij}^2 + \psi_i = h_i^2 + \psi_i \ , \ i = 1,...p \qquad (8)$$

donde $h_i^2 = Var\left(\sum_{j=1}^{k} a_{ij} F_j\right)$ y $\psi_i = Var(u_i)$ reciben el nombre de *comunalidad* y *especificidad* de la variable X_i, respectivamente.

Por lo tanto, la varianza de cada una de las variables analizadas puede descomponerse en dos partes: una, la comunalidad h_i^2, que representa la varianza explicada por los factores comunes, y otra la especificidad ψ_i que representa la parte de la varianza específica de cada variable que escapa a los factores comunes. Además se tiene que

$$Cov(X_i, X_l) = Cov\left(\sum_{j=1}^{k} a_{ij} F_j, \sum_{j=1}^{k} a_{il} F_l\right) = \sum_{j=1}^{k} a_{ij} a_{lj} \ , \forall i \neq l \qquad (9)$$

por lo que son los factores comunes los que explican las relaciones existentes entre las variables del problema. Es por esta razón que los factores que tienen interés y son susceptibles de interpretación experimental son los factores comunes. Los factores únicos se incluyen en el modelo dado la imposibilidad de expresar, en general, p variables en función de un número más reducido k de factores (Rencher, 2002).

Uno de los requisitos que debe cumplirse para que el Análisis Factorial tenga sentido es que las variables estén altamente intercorrelacionadas. Por tanto, si las correlaciones entre todas las variables son bajas, el Análisis Factorial tal vez no sea apropiado. Además, también se espera que las variables que tienen correlación muy alta entre sí la tengan con el

mismo factor o factores (Sharma, 1998). La finalidad de este análisis es comprobar si sus características son las más adecuadas para realizar un Análisis Factorial. Para ello, se parte de la matriz de correlaciones muestrales R = (r_{ij}), donde r_{ij} es la correlación muestral observada entre las variables X_i y X_j. Existen diferentes indicadores del grado de asociación entre las variables:

- **Test de esfericidad de Barlett:** Una posible forma de examinar la matriz de correlaciones es mediante el test de esfericidad de Bartlett que contrasta, bajo la hipótesis de normalidad multivariante, si la matriz de correlación de las variables observadas, R, es la identidad. Si una matriz de correlación es la identidad significa que las intercorrelaciones entre las variables son cero. La confirmación de esta hipótesis nula supone que las variables no están intercorrelacionadas. El test de esfericidad de Bartlett se obtiene a partir de una transformación del determinante de la matriz de correlación. El estadístico de dicho test viene dado por:

$$d_R = -\left[m - 1 - \frac{1}{6}(2p+5) \right] \log|R| = -\left[n - \frac{2p+11}{6} \right] \sum_{j=1}^{p} \log(\lambda_j) \qquad (10)$$

  donde n es el número de individuos de la muestra, p las variables observadas y λ_j (j = 1, …, p) los valores propios de R. Bajo la hipótesis nula este estadístico se distribuye asíntóticamente según una distribución χ^2 con p(p-1)/2 grados de libertad.

  Si la hipótesis nula es cierta, los valores propios valdrían uno, o equivalentemente, su logaritmo sería nulo y, por tanto, el estadístico del test valdría cero. Por el contrario, si con el test de Bartlett se obtienen valores altos de χ^2, o equivalentemente, un determinante bajo, esto significa que hay variables con correlaciones altas (un determinante próximo a cero indica que una o más variables podrían ser expresadas como una combinación lineal de otras variables). Así pues, si el estadístico del test toma valores grandes se rechaza la hipótesis nula con un cierto grado de significación. En caso de no rechazarse la hipótesis nula significaría que las variables no están intercorrelacionadas y en este supuesto debería reconsiderarse la aplicación de un Análisis Factorial.

- **Medidas de adecuación de la muestra:** El coeficiente de correlación parcial es un indicador de la fuerza de las relaciones entre dos variables eliminando la influencia del resto. Si las variables comparten factores comunes, el coeficiente de correlación parcial entre pares de variables deberá ser bajo, puesto que se eliminan los efectos lineales de las otras variables. Las correlaciones parciales son estimaciones de las correlaciones entre los factores únicos y deberían ser próximos a cero cuando el Análisis Factorial es adecuado, ya que estos factores se suponen que están incorrelados entre sí. Por lo tanto si existe un número elevado de coeficientes de este tipo distintos de cero es señal de que las hipótesis del modelo factorial no son compatibles con los datos. Una forma de evaluar este hecho es mediante la Medida de Adecuación de la Muestra KMO propuesta por Kaiser, Meyer y Olkin. Dicha medida viene dada por

$$KMO = \frac{\sum_{j \neq i} \sum_{i \neq j} r_{ij}^2}{\sum_{j \neq i} \sum_{i \neq j} r_{ij}^2 + \sum_{j \neq i} \sum_{i \neq j} r_{ij(p)}^2} \qquad (11)$$

  donde $r_{ij(p)}$ es el coeficiente de correlación parcial entre las variables X_i y X_j eliminando la influencia del resto de las variables.

  KMO es un índice que toma valores entre 0 y 1 y que se utiliza para comparar las magnitudes de los coeficientes de correlación observados con las magnitudes de los coeficientes de correlación parcial de forma que, cuanto más pequeño sea su valor, mayor es el valor de los coeficientes de correlación parciales $r_{ij(p)}$ y, por lo tanto, menos deseable es realizar un Análisis Factorial.

  Kaise, Meyer y Olkin aconsejan que si KMO $\geq$ 0,75 la idea de realizar un análisis factorial es buena, si 0,75 > KMO $\geq$ 0,5 la idea es aceptable y si KMO < 0,5 es inaceptable.

Una vez que se ha determinado que el Análisis Factorial es una técnica apropiada para analizar los datos, debe seleccionarse el método adecuado para la extracción de los factores. Existen diversos métodos, cada uno de ellos con sus ventajas e inconvenientes.

El modelo factorial en forma matricial viene dado por X = FA' + U. El problema consiste en cuantificar la matriz A de cargas factoriales que explica X en función de los factores. A partir de esta expresión se deduce la llamada identidad fundamental del Análisis Factorial:

$$R = AA' + \psi \qquad (13)$$

donde R es la matriz de correlación poblacional de las variables X_1, ..., X_p y $\Psi = \text{diag}(\psi_i)$ es la matriz diagonal de las especificidades.

Igualando cada elemento de la matriz R con la combinación lineal correspondiente al segundo miembro de esta ecuación resultan pxp ecuaciones, que es el número de elementos de R. Ahora bien, la matriz R es simétrica y, consecuentemente, está integrada por p(p+1)/2 elementos distintos, que es el número real de ecuaciones de que disponemos. En el segundo miembro los parámetros a estimar son los pxk elementos de la matriz A y los p elementos de la matriz Ψ. En consecuencia, para que el proceso de estimación pueda efectuarse se requiere que el número de ecuaciones sea mayor o igual que el número de parámetros a estimar [$p(p+1)/2 \geq p(k+1)$] o equivalentemente $k \leq (p-1)/2$.

Existen muchos métodos para obtener los factores comunes. Los distintos métodos difieren tanto en el algoritmo de cálculo como en la matriz que será analizada. Algunos de los más habituales son:

- **Componentes principales:** Método de extracción en el que los factores obtenidos son los autovectores de la matriz de correlaciones reescalados.
- **Mínimos cuadrados no ponderados:** Método de extracción que minimiza la suma de los cuadrados de las diferencias entre las matrices de correlaciones observada y reproducida, ignorando los elementos de la diagonal.
- **Mínimos cuadrados generalizados:** Método de extracción que minimiza la suma de los cuadrados de las diferencias entre las matrices de correlaciones observada y reproducida. Las correlaciones se ponderan por el inverso de su unicidad, de manera que las variables cuya unicidad es alta reciben un peso menor que aquellas cuyo valor es bajo. Este método genera un estadístico de bondad de ajuste chi-cuadrado que permite contrastar la hipótesis nula de que la matriz residual es una matriz nula.

- **Máxima verosimilitud:** Método de extracción que proporciona las estimaciones de los parámetros que, con mayor probabilidad, han producido la matriz de correlaciones observada, asumiendo que la muestra procede de una distribución normal multivariante. Las correlaciones se ponderan por el inverso de la unicidad de las variables y se emplea un algoritmo iterativo. Este método genera un estadístico de bondad de ajuste chi-cuadrado que permite contrastar la bondad del modelo para explicar la matriz de correlaciones.
- **Ejes principales:** Método de extracción iterativo en el que, como estimación inicial de la comunalidad, la matriz de correlaciones original se reduce, sustituyendo los unos de su diagonal por las estimaciones de la correlación múltiple al cuadrado entre cada variable y todas las demás. La matriz reducida se auto-descompone y se corrigen las estimaciones iniciales de la comunalidad por las nuevas estimaciones resultantes. El proceso continúa hasta que no existe diferencia entra las estimaciones de las comunalidades entre dos pasos sucesivos o se alcanza alguno de los criterios de parada.

Los criterios para elegir entre estos métodos se resumen en los siguientes puntos:

- Cuando las comunalidades son altas (mayores que 0.6) todos los procedimientos tienden a dar la misma solución.
- Cuando las comunalidades son bajas para algunas de las variables, el método de componentes principales tiende a dar soluciones muy diferentes del resto de los métodos, con cargas factoriales mayores.
- Si el número de variables es alto (mayor que 30), las estimaciones de la comunalidad tienen menos influencia en la solución obtenida y todos los métodos tienden a dar el mismo resultado.
- Si el número de variables es bajo todo depende del método utilizado para estimar las comunalidades y de si éstas son altas, más que del método utilizado para estimarlas.

La matriz factorial puede presentar un número de factores superior al necesario para explicar la estructura de los datos originales. Generalmente, hay un conjunto reducido de

factores, los primeros, que contienen casi toda la información. Los otros factores suelen contribuir relativamente poco. Uno de los problemas que se plantean consiste en determinar el número de factores que conviene conservar, puesto que de lo que se trata es de cumplir el principio de parsimonia. Se han dado diversas reglas y criterios orientativos para determinar el número de factores a conservar:

- **Determinación "a priori":** Este es el criterio más fiable si los datos y las variables están bien elegidos y el investigador conoce a fondo el terreno que pisa, puesto que lo ideal es plantear el Análisis Factorial con una idea previa de cuántos factores hay y cuáles son.
- **Regla de Kaiser:** Consiste en calcular los valores propios de la matriz de correlaciones R y tomar como número de factores el número de valores propios superiores a la unidad. Este criterio es una reminiscencia del Análisis de Componentes Principales y se ha comprobado en simulaciones que, generalmente, tiende a infraestimar el número de factores por lo que se recomienda su uso para establecer un límite inferior. Un límite superior se calcularía aplicando este mismo criterio pero tomando como límite 0.7.
- **Criterio del porcentaje de la varianza:** También es una reminiscencia del Análisis de Componentes Principales y consiste en tomar como número de factores el número mínimo necesario para que el porcentaje acumulado de la varianza explicado alcance un nivel satisfactorio que suele ser del 75% o el 80%. Tiene la ventaja de poderse aplicar también cuando la matriz analizada es la de varianzas y covarianzas, pero no tiene ninguna justificación teórica ni práctica.

La interpretación de los factores se basa en las correlaciones estimadas de los mismos con las variables originales del problema. Observar que, si el modelo de Análisis Factorial es cierto, se tiene que:

$$Corr(X_i, F_l) = Cov(X_i, F_l) = \sum_{j=1}^{k} a_{ij} Cov(F_j, F_l), \forall i = 1,..., p; l = 1,..., k \qquad (14)$$

En particular, si los factores son ortogonales

$$Corr(X_i, F_l) = a_{il}, \forall i = 1,..., p; l = 1,..., k \qquad (15)$$

Vemos, por lo tanto, que la matriz de cargas factoriales, A, juega un papel fundamental en dicha interpretación. Además, las cargas factoriales al cuadrado a_{il}^2 indican, si los factores son ortogonales, qué porcentaje de la varianza de la variable original X_i es explicado por el factor F_l.

Sin embargo, rara vez los métodos de extracción de factores vistos anteriormente proporcionan matrices de cargas factoriales adecuadas para la interpretación. Para resolver este problema están los procedimientos de Rotación de Factores, que se basan en la no unicidad de la solución al Análisis Factorial. En efecto, las soluciones dadas para la matriz A no son únicas, puesto que cualquier transformación ortogonal de A es también una solución. Si T es una matriz ortogonal, entonces TT' = T'T = I, al aplicar una transformación ortogonal a A se obtiene una solución distinta al sistema anterior. Esta es la base de los métodos de rotación de factores. Por tanto, si T es una matriz ortogonal, entonces A* = AT es solución y definimos otros factores F* = FT (F* es el vector F rotado por la matriz ortogonal T). Se comprueba que X y R siguen verificando las ecuaciones del modelo, es decir:

$$R = A^*A^{*\prime} + \Psi = (AT)(T'A') + \Psi = AA' + \Psi \qquad (16)$$

$$X = F^*A^{*\prime} + U = (FT)(T'A') + U = FA' + U \qquad (17)$$

Por lo tanto, el modelo es único salvo rotaciones ortogonales, es decir, se pueden realizar rotaciones de la matriz de ponderaciones o cargas factoriales sin alterar el modelo.

Los diferentes métodos de rotación de factores tratan de cumplir los siguientes objetivos:

- Cada factor debe tener unos pocos pesos altos y los otros próximos a cero
- Cada variable no debe estar saturada más que en un factor
- No deben existir factores con la misma distribución, es decir, dos factores distintos deben presentar distribuciones diferentes de cargas altas y bajas

Existen dos formas básicas de realizar la rotación de factores: la Rotación Ortogonal y la Rotación Oblicua según que los factores rotados sigan siendo ortogonales o no. Conviene

advertir que, tanto en la rotación ortogonal como en la rotación oblicua, la comunalidad de cada variable no se modifica, es decir, la rotación no afecta a la bondad de ajuste de la solución factorial: aunque cambie la matriz factorial, las especificidades no cambian y, por tanto, las comunalidades permanecen inalteradas. Sin embargo, cambia la varianza explicada por cada factor, luego los nuevos factores rotados no están ordenados de acuerdo con la información que contienen, cuantificada a través de su varianza.

Rotación Ortogonal: En la rotación ortogonal los ejes se rotan de forma que quede preservada la incorrelación entre los factores. Dicho de otra forma, los nuevos ejes, o ejes rotados, son perpendiculares de igual forma que lo son los factores sin rotar.

Si T es una matriz ortogonal con TT' = T'T = I, entonces:

$$X = FA' + U = FTT'A' + U = GB' + U \qquad (18)$$

La matriz G geométricamente es una rotación de F y verifica las mismas hipótesis que ésta. Lo que realmente se realiza es un giro de ejes, de manera que cambian las cargas factoriales y los factores. Se trata de buscar una matriz T tal que la nueva matriz de cargas factoriales B tenga muchos valores nulos o casi nulos y unos pocos valores cercanos a la unidad

- **Método Varimax:** Se trata de un método de rotación que minimiza el número de variables con cargas altas en un factor, mejorando así la capacidad de interpretación de factores. Este método considera que si se logra aumentar la varianza de las cargas factoriales al cuadrado de cada factor consiguiendo que algunas de sus cargas factoriales tiendan a acercarse a uno mientras que otras se acerquen a cero, lo que se obtiene es una pertenencia más clara e inteligible de cada variable a ese factor. Los nuevos ejes se obtienen maximizando la suma para los k factores retenidos de las varianzas de las cargas factoriales al cuadrado dentro de cada factor. Para evitar que las variables con mayores comunalidades tengan más peso en la solución final, suele efectuarse la normalización de Kaiser, consistente en dividir cada carga factorial al cuadrado por la comunalidad de la variable correspondiente. En consecuencia, el

Método Varimax determina la matriz B de forma que se maximice la suma de las varianzas:

$$V = p\sum_{i=1}^{k}\sum_{j=1}^{p}\left(\frac{b_{ij}}{h_j}\right)^4 - \sum_{i=1}^{k}\left(\sum_{j=1}^{p}\frac{b_{ij}^2}{h_j^2}\right)^2 \qquad (19)$$

- **Método Quartimax:** El objetivo de este método es que cada variable tenga correlaciones elevadas con un pequeño número de factores. Para ello busca maximizar la varianza de las cargas factoriales al cuadrado de cada variable en los factores, es decir, el método trata de maximizar la función:

$$V = p\sum_{i=1}^{p}\sum_{j=1}^{k}\left(b_{ij}^2 - \bar{b}_i^2\right)^2 \text{ , donde } \bar{b}_i^2 = \frac{1}{k}\sum_{j=1}^{k} b_{ij}^2 \qquad (20)$$

  Con ello se logra que cada variable concentre su pertenencia en un determinado factor, es decir, presente una carga factorial alta mientras que, en los demás factores, sus cargas factoriales tiendan a ser bajas. La interpretación así gana en claridad por cuanto la comunalidad total de cada variable permanece constante, quedando más evidente de este modo hacia qué factor se inclina con más fuerza cada variable. El método es tanto más clarificador cuanto mayor número de factores se hayan calculado.

  Este método tiende a producir un primer factor general, que se le suele dar el nombre de tamaño, y el resto de factores presentan ponderaciones menores que las dadas por el método Varimax.

- **Método Equamax:** Este método busca maximizar la media de los criterios anteriores. Suele tener un comportamiento similar a uno de los dos métodos anteriores.

Rotación oblicua: Se diferencia de la rotación ortogonal en que a la matriz T de rotación no se le exige ser ortogonal, sino únicamente no singular. De esta forma los factores rotados no tienen porqué ser ortogonales y tener, por lo tanto, correlaciones distintas de cero entre sí. La rotación oblicua puede utilizarse cuando es probable que los factores en la población tengan una correlación muy fuerte. Insistimos en que hay que ir con mucho cuidado en la interpretación de las rotaciones oblicuas, ya que la superposición de factores

puede confundir la significación de los mismos. De esta forma el análisis gana más flexibilidad y realismo pero a riesgo de perder robustez, por lo que conviene aplicar estos métodos si el número de observaciones por factor es elevado.

Una vez determinados los factores rotados el siguiente paso es calcular las matrices de puntuaciones factoriales F. Las posibilidades de analizar las puntuaciones factoriales de los sujetos son muy variadas según lo que se pretenda:

- Conocer qué elementos de la muestra son los más raros o extremos
- Conocer dónde se ubican ciertos grupos o subcolectivos de la muestra
- Conocer en qué factor sobresalen unos elementos y en qué factor no
- Explicar, analizando las informaciones anteriores, por qué han aparecido dichos factores en el análisis realizado

Existen diversos métodos de estimación de la matriz F. Las propiedades que serían deseable cumpliesen los factores estimados son:

- Cada factor estimado debe tener una correlación alta con el verdadero factor
- Cada factor estimado debe tener una correlación nula con los demás factores verdaderos
- Los factores estimados deben ser incorrelados dos a dos, es decir, mutuamente ortogonales si son ortogonales
- Los factores estimados deben ser estimadores insesgados de los verdaderos factores

Sin embargo, por la propia naturaleza de los factores comunes, el problema de su estimación es complejo. Se puede demostrar que los factores no son, en general, combinación lineal de las variables originales. Además, en la mayoría de las situaciones, no existirá una solución exacta y ni siquiera será única. Todos los métodos de obtención de puntaciones factoriales parten de la expresión:

$$X = FA' + U \text{ con } E[U]=0, \text{Var}[U] = \Psi \qquad (21)$$

a partir de la cual buscan estimar el valor de F.

Tres de los métodos de estimación más utilizados son los siguientes:

- **Método de regresión:** Estima F mediante el método de los mínimos cuadrados

$$\hat{F} = (A'A)^{-1} A'X \qquad (22)$$

- **Método de Barlett:** Utiliza el método de los mínimos cuadrados generalizados estimando las puntuaciones factoriales mediante:

$$\hat{F} = (A'\Psi^{-1}A)^{-1} A'\Psi^{-1}X \qquad (23)$$

- **Método de Anderson-Rubin:** Estima F mediante el método de los mínimos cuadrados generalizados pero imponiendo la condición adicional F'F = I

$$\hat{F} = (A'\Psi^{-1}R\Psi^{-1}A)^{-1} A'\Psi^{-1}X \qquad (24)$$

Si se utilizan las puntuaciones factoriales para clasificar los elementos de la muestra, es preciso efectuar a continuación pruebas que permitan rechazar la hipótesis nula de igualdad de medias, de modo que se confirme si los subgrupos dentro de la muestra poseen características o comportamientos diferentes.

El análisis de la varianza (ANOVA) de un factor sirve para comparar varios grupos en una variable cuantitativa. A la variable categórica (nominal u ordinal) que define los grupos que deseamos comparar la llamamos independiente y las variables cuantitativas sobre las que deseamos comparar los grupos las llamamos dependientes. La hipótesis que se pone a prueba en el ANOVA de un factor es que las medias poblacionales (las medias de cada variable dependiente en cada nivel de la variable independiente) son iguales. Si las medias poblacionales son iguales, eso significa que los grupos identificados por la variable dependiente no difieren en esa variable dependiente. Si se rechaza la hipótesis es que existen diferencias entre las medias.

El estadístico F refleja el grado de parecido entre las medias. F se define como la varianza entre las medias de los grupos dividido por las varianzas dentro de cada grupo:

$$F = \frac{\sigma_1^2}{\sigma_2^2} = \frac{n\sigma_Y^2}{S_j^2} \qquad (25)$$

Cuando la estimación de la varianza a partir de las diferencias entre medias sea similar a la estimación de la varianza basada en los valores individuales, el cociente será próximo a 1. Si las diferencias entre medias son grandes, el cociente será mayor que 1. Cuando las poblaciones son normales y sus varianzas iguales, el estadístico F se distribuye según el modelo de probabilidad F de Fisher-Snedecor, donde los grados de libertad del numerador es el número de grupos menos 1 y el del denominador el número total de observaciones menos el número de grupos. El nivel crítico que proporciona el programa es la probabilidad de obtener un valor de F igual al obtenido o mayor. Si la probabilidad es menor que 0.05 rechazamos la hipótesis de igualdad de medias. Los requisitos del ANOVA de un factor son, por tanto, normalidad e igualdad de varianzas (homocedasticidad).

En ocasiones, estos requisitos son demasiado exigentes por lo que entonces se acude a las denominadas pruebas no paramétricas. En particular, la prueba de Kruskal y Wallis (1952) se aplica al caso de varias muestras independientes. La única exigencia versa sobre la aleatoriedad en la extracción de las muestras, no haciendo referencia a ninguna de las otras condiciones adicionales de homocedasticidad y normalidad, necesarias para la aplicación del test paramétrico ANOVA.

5.2 PLS

Los Modelos de Ecuaciones Estructurales (MEE) se han convertido en uno de los desarrollos recientes más importantes del análisis multivariante y su uso se ha extendido entre las ciencias sociales. En particular, esta difusión se ha observado en el campo de la economía y la dirección de empresas. Surgen como fruto de la unión de dos tradiciones:

1. La Perspectiva Econométrica, que se enfoca en la predicción

2. El Enfoque Psicométrico, que modela conceptos como variables latentes (no observadas) que son indirectamente inferidas de múltiples medidas observadas (indicadores o variables manifiestas).

Es por ello que los Modelos de Ecuaciones Estructurales (MEE) han permitido a los científicos sociales la modelización analítica de caminos (paths) con variables latentes.

Se define este modelo como una técnica multivariante que combina aspectos de la regresión múltiple, examinando relaciones de dependencia, y Análisis Factorial, que representa conceptos inmedibles —factores— con variables múltiples, para estimar una serie de relaciones de dependencia interrelacionadas simultáneamente.

Los MEE valoran en un único análisis, sistemático e integrador tanto el modelo de medida como el modelo estructural. Es decir, se valoran tanto las cargas factoriales de las variables observables (indicadores o medidas) con relación a sus correspondientes variables latentes (constructos), valorándose la fiabilidad y validez de las medidas de los constructos teóricos, como las relaciones de causalidad hipotetizadas entre un conjunto de constructos independientes y dependientes.

De forma general, los MEE permiten:

1. Modelizar el error de medida, es decir, el grado en el que las variables que podemos medir (indicadores) no describen perfectamente la(s) variable(s) latente(s) de interés. Esto se realiza mediante la modelización explícita y el aislamiento de las fuentes de error, permitiendo que las relaciones sean ajustadas a estos errores
2. Incorporar constructos abstractos e inobservables, es decir, variables latentes o variables teóricas no observables
3. Modelizar relaciones entre múltiples variables predictoras (independientes, exógenas) y criterios (dependientes o endógenas)
4. Combinar y confrontar conocimiento a priori e hipótesis con datos empíricos, siendo más confirmatorios que exploratorios

El análisis holístico que los MEE desarrollan puede ser llevado a cabo por medio de dos tipos de técnicas estadísticas: los métodos basados en el análisis de las covarianzas, como

por ejemplo, Lisrel, EQS, Amos, Sepath, Ramona, MX y Calis; o los análisis basados en componentes o Partial Least Squares, como por ejemplo LV-PLS y PLS-Graph. Ambos enfoques difieren en los objetivos de sus análisis, las suposiciones estadísticas en las que se basan y la naturaleza de los estadísticos de ajuste que proporcionan.

El objetivo de los Métodos Basados en Covarianzas es estimar los parámetros del modelo, es decir, las cargas y valores, de tal modo que se minimicen las discrepancias entre la matriz empírica inicial de datos de covarianzas y la matriz de covarianzas deducida a partir del modelo y de los parámetros estimados. Se trata de usar el modelo para explicar la covariación de todos los indicadores. Proporciona medidas de bondad de ajustes globales que informan a cerca del grado con el que el modelo hipotetizado se ajusta a los datos disponibles. En estos métodos se coloca el énfasis sobre el ajuste del modelo completo, es decir, testar en conjunto una teoría sólida, adaptándose mejor a la investigación confirmatoria.

El objetivo de los Análisis Basados en Componentes o PLS es la predicción de las variables dependientes, tanto latentes como manifiestas. Esta meta se traduce en un intento por maximizar la varianza explicada (R^2) de las variables dependientes, lo que nos lleva a que las estimaciones de los parámetros estén basadas en la capacidad de minimizar las varianzas residuales de las variables endógenas. En comparación con los Métodos Basados en Covarianzas, PLS se adapta mejor a aplicaciones predictivas y el desarrollo de la teoría (análisis exploratorio), aunque también puede ser usado para la confirmación de la teoría (análisis confirmatorio).

En situaciones donde la teoría previa es sólida y se tiene como meta un mayor desarrollo y evaluación de la teoría, los métodos de estimación basados en covarianzas (por ejemplo, máxima verosimilitud —ML— o mínimos cuadrados generalizados —GLS—) son más adecuados. Sin embargo, PLS puede ser más adecuado para fines predictivos, ya que se orienta principalmente para el análisis causal predictivo en situaciones de alta complejidad pero con un conocimiento teórico escaso. Finalmente, hay que subrayar que ambos procedimientos deben ser entendidos como de naturaleza complementaria (Chin *et al*., 1996).

La elección entre los métodos de MEE depende de factores tales como el objetivo del estudio, la naturaleza de los datos y las suposiciones asociadas con el método, como anteriormente se dijo. Sin embargo, existen también consideraciones de carácter general que es preciso tener presente:

1. **La naturaleza de los constructos teóricos:** Dependiendo de la forma en la que el error es tratado, los constructos pueden ser calificados en dos categorías:

    a. Constructo indeterminado: es una combinación de sus indicadores y un término de error.

    b. Constructo definido: es un compuesto (frecuentemente llamado componente o variable derivada) de sus indicadores, es decir, una agregación lineal ponderada de sus indicadores. Los constructos definidos sacrifican la aspiración teórica de tener en cuenta medidas imprecisas por la ventaja práctica de la estimación del constructo y el cálculo directo de las puntuaciones de los componentes, y está completamente determinado por sus indicadores, asumiendo que el efecto combinado de los indicadores se encuentra libre del error de medida.

2. **La naturaleza de las relaciones entre los constructos**, de manera que la ortogonalidad implica una correlación nula, la simetría sugiere que no existe diferencia en la dirección de la relación, las relaciones direccionales nos dicen en cuánto una variable dependiente cambiará dada una transformación en una variable independiente, pudiendo ser unidireccional (recursiva) y bidireccional (no recursiva), y la causalidad es asumida por el investigador, ya que las leyes causales no pueden ser comprobadas.

3. **La naturaleza de las relaciones epistemológicas:** Una relación epistemológica describe el vínculo entre la teoría y los datos, es decir, entre los constructos teóricos y los datos empíricos. Existen diversos tipos de relaciones epistemológicas básicas que influyen sobre el método de análisis:

    a. Indicadores reflectivos, en los que el constructo no observado da lugar a lo que se observa

b. Indicadores formativos dan lugar al constructo teórico latente

c. Indicadores simétricos, en los que no se hacen suposiciones acerca de la direccionalidad o causalidad entre constructos e indicadores empíricos

d. Múltiple-indicator multiple-cause (MIMIC) model: esta representación hace una combinación de indicadores reflectivos y formativos

e. Modelo de alto nivel: se produce una combinación de indicadores reflectivos y formativos en dos niveles

El enfoque MEE basado en covarianzas (especialmente bajo la aplicación ML) persigue proporcionar una afirmación de causalidad, una descripción de los mecanismos causales. El problema que se suscita al intentar alcanzar tal tipo de conocimiento con estas técnicas son las suposiciones restrictivas que se requieren con respecto a la teoría subyacente, las distribuciones de los datos y los niveles de medida de las variables; estas demandas se pueden encontrar dentro de lo que se define como un sistema cerrado (Falk *et al*., 1992) de *modelización firme o rígida*. Sin embargo, dado estos limitativos requerimientos, parece difícil la aplicación estricta de este tipo de modelización en el campo de las ciencias sociales. En esta situación surge PLS, técnica que fue diseñada para reflejar las condiciones teóricas y empíricas de las ciencias sociales y del comportamiento, donde son habituales las situaciones con teorías no suficientemente asentadas y escasa información disponible (Wold, 1979). A esta forma de modelización se la conoce como *modelizacion flexible* (Wold, 1980). Los procedimientos matemáticos y estadísticos subyacentes en el sistema son rigurosos y robustos (Wold, 1979); sin embargo, el modelo matemático es flexible en el sentido de que no realiza suposiciones relativas a niveles de medida, distribuciones de los datos y tamaño muestral. La meta que se persigue es más moderada que la modelización firme, abandonándose la idea de causalidad (presente en la modelización firme) y se reemplaza por el concepto de predictibilidad. Mientras que la causalidad garantiza la capacidad de controlar los acontecimientos, la predictibilidad permite sólo un limitado grado de control (Falk *et al*., 1992).

En este sentido, la modelización flexible (PLS) podría ser usada apropiadamente incluso aunque concurran una o más de las condiciones y circunstancias siguientes (Falk *et al*., 1992):

1. Condiciones teóricas:
    a. Las hipótesis se derivan de una teoría de nivel macro en la que no se conocen todas las variables relevantes o destacadas
    b. Las relaciones entre constructos teóricos y sus manifestaciones son vagas
    c. Las relaciones entre constructos son conjeturales
2. Condiciones de medida:
    a. Alguna o todas las variables manifiestas son categóricas o presentan diferentes niveles de medida
    b. Las variables manifiestas tienen cierto grado de no fiabilidad
    c. Los residuos de las variables latentes y manifiestas se encuentran correlacionados (heterocedasticidad)
3. Condiciones de distribución:
    a. Los datos provienen de distribuciones desconocidas o no normales
4. Condiciones prácticas:
    a. Se emplean diseños de investigación no experimentales (por ejemplo, encuestas, datos secundarios, diseños de investigación cuasi experimentales, etc.)
    b. Se modelan un gran número de variables latentes y manifiestas
    c. Se disponen, bien de demasiados casos, bien de un número escaso

5.2.1 El modelo PLS

"El núcleo conceptual de PLS es una combinación iterativa de análisis de componentes principales, que vincula medidas con constructos, y de análisis path, que permite la

construcción de un sistema de constructos. Las relaciones hipotetizadas entre medidas (indicadores) y constructos, y entre constructos y otros constructos son guiadas por la teoría. La estimación de los parámetros que representan las medidas y las relaciones path es llevada a cabo empleando técnicas de Mínimos Cuadrados Ordinarios (OLS). PLS puede ser entendido con una sólida comprensión de análisis de componentes principales, análisis path y regresión OLS" (Barclay *et al*., 1995).

El objetivo de PLS es ayudar al investigador a obtener valores determinados para variables latentes con el fin de realizar predicciones. El modelo formal define explícitamente las variables latentes como combinaciones lineales de sus indicadores observados. Las estimaciones de pesos para crear las puntuaciones componentes de las variables latentes se obtienen en base a cómo se especifica el modelo estructural y el modelo de medición. Como resultado, las varianzas residuales de las variables dependientes, tanto latentes como observadas, se minimizan.

El modelo PLS es un sistema que incorpora técnicas multivariantes de primera generación; tiene presente el papel de guía que tiene la teoría en la descripción de relaciones, lo cual subraya lo expuesto por Fornell (1982; 1987) al señalar que las metodologías de análisis multivariante de segunda generación enfatizan los aspectos acumulativos del desarrollo de la teoría, por el que el conocimiento es incorporado a priori dentro del análisis empírico; y la técnica de estimación que sigue es Mínimos Cuadrados Ordinarios.

Gráficamente, el modelo PLS se podría describir según la Figura 13.

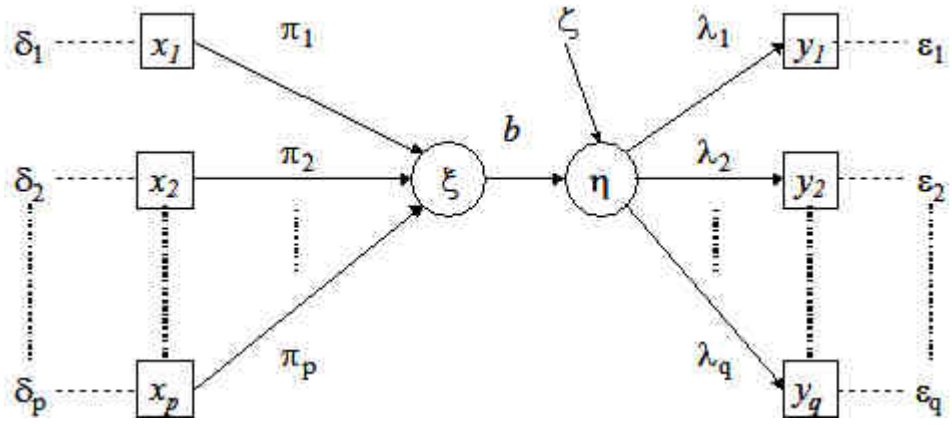


Figura 13. Modelo PLS genérico

donde:

ξ : constructo exógeno

η : constructo endógeno

x_t, $t = 1, ..., p$: variables x, medidas o indicadores

y_i, $i = 1, ..., q$: variables y, medidas o indicadores

π_j, $j = 1, ..., p$: pesos de regresión

δ_l, $l = 1, ..., p$: residuos provenientes de las regresiones

λ_m, $m = 1, ..., q$: cargas

ε_n, $n = 1, ..., q$: términos de error ($1-\lambda_m^2$)

ζ : residuo en el modelo estructural

b : coeficiente de regresión simple entre ξ y η

El constructo teórico, variable latente o no observable gráficamente se representa por un círculo. Dentro de los constructos podemos distinguir los constructos exógenos (ξ) que actúan como variables predictoras o "causales" de constructos endógenos (η).

Los indicadores, medidas, variables manifiestas u observables se simbolizan gráficamente por medio de cuadrados. Existen dos tipos básicos de indicadores:

1. **Indicadores reflectivos:** las variables observables son expresadas como una función del constructo, de tal modo que éstas reflejan o son manifestaciones del constructo. Por tanto, la variable latente precede a los indicadores en un sentido "causal".
2. **Indicadores formativos:** implican que el constructo es expresado como una función de las variables manifiestas; los indicadores forman, causan o preceden al constructo

La perspectiva de medición basada en el uso de indicadores formativos o causales refleja la idea de que "en muchos casos, los indicadores podrían ser vistos como causas y no efectos de la variable latente que dichos indicadores pretenden medir" (MacCallum *et al*., 1993).

Tabla 4. Principales diferencias entre indicadores reflectivos y formativos

Indicadores reflectivos		**Indicadores formativos**
Selección aleatoria a partir de un conjunto de indicadores relacionados con el concepto	Especificación de indicadores ⟷ que forman la medida	Representación de indicadores que recogen todo el significado y contenido del concepto
Intercambiables entre ellos. La eliminación de uno no cambia la naturaleza y contenido del concepto.	⟷ Naturaleza de los indicadores	No intercambiables entre ellos. La eliminación de uno cambia la naturaleza y contenido del concepto
Las correlaciones entre indicadores vienen explicadas por el modelo de medición en la medida en que cada indicador tiene una asociación con la variable latente	⟷ Consistencia interna	Las correlaciones entre indicadores no vienen explicadas por el modelo de medida, pues los indicadores están determinados exógenamente y no por el mismo concepto
Es recomendable una elevada correlación de los indicadores como indicio de que tras ellos subyace el mismo concepto y que, por tanto, son medidas válidas de éste	⟷ Validez de la medida	No es recomendable una elevada correlación entre indicadores, pues eso conlleva un problema de multicolinealidad. No se puede deducir la validez de los indicadores mediante dichas correlaciones.

Por ejemplo, la medición del estatus socioeconómico de una persona puede adoptar una perspectiva formativa, dado que dicho estatus está formado por una combinación de aspectos tales como el nivel de formación, el nivel de ingresos, la categoría profesional y el lugar de residencia del individuo. Por tanto, es razonable esperar que un aumento de cualquiera de estos aspectos afecte al estatus socioeconómico, lo cual es distinto del hecho de que ese estatus determine, por ejemplo, el grado de formación o la categoría profesional de un individuo. Consecuentemente, la elección entre una especificación reflectiva o formativa tiene que basarse en consideraciones teóricas sobre la prioridad causal entre los indicadores y la variable latente. Dicha elección no es una decisión trivial por cuanto que, frente a la perspectiva tradicional de desarrollo de escalas, la perspectiva de medición formativa presenta algunas diferencias que hacen inadecuado el uso de los procedimientos que tradicionalmente han sido utilizados para estimar la validez y la fiabilidad de las escalas compuestas por indicadores reflectivos (Bollen y Lennox, 1991; Bagozzi, 1994).

Las relaciones asimétricas o relaciones unidireccionales entre variables pueden ser interpretadas como relaciones "causales" o predictivas, siendo representadas gráficamente por medio de flechas con una única dirección. El esquema de flechas especifica las relaciones internas (modelo interno) entre constructos y las relaciones externas (modelo externo) entre cada variable latente y sus indicadores.

5.2.2 Procedimiento a seguir para la construcción de una medida con indicadores formativos

Tomando como referencia el trabajo de Diamantopoulos *et al*. (2001) hay tres fases a seguir que son críticas para la elaboración exitosa de un índice de medición:

1. Especificación del contenido
2. Especificación de los indicadores
3. Validez externa

La primera fase tiene que ver con la especificación del contenido del concepto que se pretende medir ya que, bajo la perspectiva de la medición formativa, la variable latente está

determinada por los indicadores de medición. Consecuentemente, es importante determinar el alcance del significado del concepto como paso previo a la identificación de los indicadores, pues la no-inclusión de todas las facetas del concepto puede resultar en la exclusión de indicadores relevantes y, por tanto, de parte del propio concepto.

Seguidamente, se pasa a identificar el conjunto de indicadores que recogen todo el significado del concepto en los términos descritos en la fase anterior. Una vez delimitado el contenido del concepto así como el conjunto de indicadores, en la tercera etapa se valora la idoneidad de dichos indicadores, y que por su naturaleza es inapropiado realizar dicha valoración en términos de consistencia interna. Ante esta situación, Diamantopoulos *et al*. (2001) recomiendan la estimación de un modelo MIMIC (*Multiple Indicators and Multiple Causes*) en el que se combinan múltiples indicadores y causas del concepto objeto de análisis. Específicamente, en dicho modelo los indicadores que componen el índice, X_i, actúan como causas directas de la variable latente η, que a su vez es indicada o aproximada por uno o más indicadores reflectivos, y_j, cuya inclusión es necesaria para la identificación del modelo (Bollen, 1991).

Finalmente, a efectos de validación nomológica, Diamantopoulos *et al*. (2001) recomiendan relacionar el índice resultante en la fase anterior con otros conceptos con los que teóricamente estaría relacionado (por ejemplo, antecedentes y/o consecuencias).

5.2.3 Funcionamiento del modelo

PLS genera tres categorías diferentes de estimaciones de parámetros. La primera categoría integra las estimaciones de pesos —*weight estimates*—, que se emplean para crear las puntuaciones de las variables latentes. La segunda categoría muestra las estimaciones de las relaciones o *paths* que conectan las variables latentes, así como las variables latentes y sus respectivos bloques de indicadores —las cargas o *loadings*—. Por último, la tercera categoría recoge las medias y parámetros de localización —constantes de regresión— de los indicadores y las variables latentes.

Los parámetros estructurales y de medida de un modelo causal PLS son estimados de forma iterativa usando Mínimos Cuadrados Ordinarios y regresiones simples y múltiples. El proceso puede ser descrito del siguiente modo:

1. En la primera iteración de PLS, un valor inicial para η es obtenido sumando simplemente los valores $y_1 \ldots y_q$ (es decir, las cargas $\lambda_1 \ldots \lambda_q$ son fijadas en 1)
2. Para estimar los pesos de regresión $\pi_1 \ldots \pi_p$ se lleva a cabo una regresión con η como variable dependiente y $x_1 \ldots x_p$ como variables independientes
3. Estas estimaciones son entonces usadas como pesos o ponderaciones en una combinación lineal de $x_1 \ldots x_p$ dando lugar a un valor inicial para ξ
4. Las cargas $\lambda_1 \ldots \lambda_q$ son estimadas entonces por una serie de regresiones simples de $y_1 \ldots y_q$ sobre ξ
5. El paso siguiente emplea las cargas estimadas, transformadas en pesos o ponderaciones, para establecer una combinación lineal de $y_1 \ldots y_q$ como nueva estimación de valor de η

Este procedimiento continúa hasta que la diferencia entre iteraciones consecutivas sea extremadamente pequeña. Por ejemplo, el procedimiento podría pararse una vez que la diferencia en la media de las R^2 de todos los constructos de una iteración a la siguiente es insignificante (por ejemplo, 0.001). Como paso final, se calcula el coeficiente de regresión simple *b* entre las puntuaciones de los componentes de ξ y η.

Para determinar cuál es la muestra requerida, PLS sigue un tratamiento de segmentación de modelos complejos; por ello puede trabajar con tamaños muestrales pequeños. Al consistir el proceso de estimación de los subconjuntos en regresiones simples y múltiples, la muestra requerida será aquélla que sirva de base a la regresión múltiple más compleja que se pueda encontrar (Barclay *et al*., 1995). Con relación al nomograma, se ha de encontrar cuál de las dos posibilidades siguientes es la mayor (lo que nos ofrecerá la mayor regresión múltiple):

1. El número de indicadores en el constructo formativo (dirigidos internamente) más complejo, es decir, aquella variable latente con el mayor número de variables manifiestas formativas

2. El mayor número de constructos antecedentes que conducen a un constructo endógeno como predictores en una regresión OLS, es decir, el mayor número de caminos estructurales que se dirigen a un constructo endógeno particular en el modelo estructural

Si se va a emplear una regresión heurística de 10 casos por predictor, los requisitos para el tamaño muestral serían el resultado de multiplicar por 10 la cifra mayor obtenida, bien en el apartado (1) anterior o en el (2). PLS no implica ningún modelo estadístico y, por tanto, evita la necesidad de realizar suposiciones con respecto a las escalas de medida. Por consiguiente, las variables pueden estar medidas por diversos niveles de medida (por ejemplo, escalas categóricas, ordinales, de intervalo o ratios). Además, PLS no precisa que los datos provengan de distribuciones normales o conocidas.

Aunque los parámetros de medida y estructurales son estimados a la vez, un modelo PLS es analizado e interpretado en dos etapas. En la primera se valora la validez y fiabilidad del modelo de medida, el cual trata de analizar si los conceptos teóricos están medidos correctamente a través de las variables observadas. Este análisis se realiza respecto a los atributos validez (mide realmente lo que se desea medir) y fiabilidad (mide de una forma estable y consistente): fiabilidad individual del ítem, la consistencia interna o fiabilidad de una escala, la validez convergente y la validez discriminante.

En la segunda etapa se valora el modelo estructural, evaluando el peso y la magnitud de las relaciones entre las distintas variables: varianza explicada de las variables endógenas (R^2) y los coeficientes *path* o pesos de regresión estandarizados (β). Esta secuencia asegura que tengamos medidas válidas y fiables antes de intentar extraer conclusiones referentes a las relaciones existentes entre los constructos.

5.2.4 Análisis de la validez y la fiabilidad

Fiabilidad individual del ítem

La fiabilidad individual del ítem es valorada examinando las cargas (λ) o correlaciones simples de las medidas o indicadores con su respectivo constructo. La *comunalidad de una*

variable (λ^2) manifiesta es aquella parte de su varianza que es explicada por el factor o constructo (Bollen, 1991). Para aceptar un indicador como integrante de un constructo, aquél ha de poseer una carga superior o igual a 0.707, lo que implica que la varianza compartida entre el constructo y sus indicadores es mayor que la varianza del error (Camines *et al*., 1979), aunque diversos investigadores opinan que esta regla empírica ($\lambda \geq 0.707$) no debería ser tan rígida en las etapas iniciales del desarrollo de escalas. Diversos estudios empíricos que han empleado modelos de ecuaciones estructurales con algoritmo PLS incluyen ítems con cargas o loadings inferiores a 0.7. Por ejemplo, Birkinshaw *et al*. (1995: 647) sostienen que "solamente los ítems con cargas superiores a 0.6 se incluyeron en el análisis"; Cool *et al*. (1989) incluyeron un ítem con carga inferior a 0.7 por razones teóricas; Fornell *et al*. (1990) aceptaron 4 ítems con cargas inferiores a 0.4. Los constructos con indicadores formativos deben ser interpretados en función de los pesos (similar análisis de correlación canónica) y no de las cargas. Aquellos indicadores que no satisfagan el criterio expuesto pueden ser eliminados en lo que se denomina "depuración de ítems".

Fiabilidad de un constructo

La fiabilidad de un constructo permite comprobar la consistencia interna de todos los indicadores al medir el concepto, es decir, se evalúa con qué rigurosidad la misma variable latente está midiendo las variables manifiestas (Roldán, 2000). Para ella, las medidas que se pueden utilizar son el coeficiente alfa de Cronbach y la fiabilidad compuesta (ρ_c) del constructo. Siguiendo las indicaciones de Barclay *et al*. (1995) y Fornell *et al*. (1981), se utilizará la fiabilidad compuesta, ya que presenta una serie de ventajas, como no estar influenciadas por el número de ítems existentes en las escalas y utilizar las cargas de los ítems tal y como existen en el modelo causal. Esta fiabilidad compuesta se mediría (Werts *et al*., 1974):

$$\rho_c = \frac{\left(\sum \lambda_i\right)^2}{\left(\sum \lambda_i\right)^2 + \sum_i \mathrm{var}(\varepsilon_i)} \qquad (26)$$

donde,

λ_i = carga estandarizada del indicador *i*

ε_i = error de medida del indicador *i*

$var(\varepsilon_i) = 1 - \lambda_i^2$

Nunnally (1978) sugiere 0.7 como un nivel adecuado para una fiabilidad "modesta" en etapas tempranas de investigación y un más estricto 0.8 para investigación básica. Ambas medidas son sólo aplicables a variables latentes con indicadores reflectivos. Sin embargo, en un constructo con indicadores formativos no se puede asumir que las medidas formativas covaríen, por lo que queda claro que estos indicadores no van a estar correlacionados.

Validez convergente

La validez convergente se valora por medio de la medida denominada "varianza extraída media (AVE)" (Fornell *et al*., 1981), la cual proporciona la cantidad de varianza que un constructo obtiene de sus indicadores con relación a la cantidad de varianza debida al error de medida

$$AVE = \frac{\sum \lambda_I^2}{\sum \lambda_I^2 + \sum_I \text{var}(\varepsilon_i)} \qquad (27)$$

donde,

λ_i = carga estandarizada del indicador *i*

ε_i = error de medida del indicador *i*

$var(\varepsilon_i) = 1 - \lambda_i^2$

Fornell y Larcker (1981) recomiendan que la varianza extraída media sea superior a 0.5, con lo que se establece que más del 50% de la varianza del constructo es debida a sus indicadores. Esta medida sólo puede ser aplicada en constructos con indicadores reflectivos.

Validez discriminante

La validez discriminante indica en qué medida un constructo dado es diferente de otros constructos. Han de existir correlaciones débiles entre éste y otras variables latentes que

midan fenómenos diferentes. Un constructo debería compartir más varianza con sus medidas o indicadores que con otros constructos en un modelo determinado. Se utiliza la varianza extraída media (AVE), es decir, la varianza media compartida entre un constructo y sus medidas. Esta medida debería ser mayor que la varianza compartida entre el constructo con los otros constructos del modelo (la correlación al cuadrado entre dos constructos).

Para evaluar el modelo estructural se ha de responder a las dos siguientes cuestiones:

1. Qué cantidad de la varianza de las variables endógenas es explicada por los constructos que las predicen. Para ello miramos el valor de R^2
2. En qué medida las variables predictoras contribuyen a la varianza explicada de las variables endógenas. Para ellos miramos el valor de β

Para responder a la primera pregunta, una medida del poder predictivo de un modelo es el valor R^2 para las variables latentes dependientes e indica la cantidad de varianza del constructo que es explicada por el modelo. Un modelo anidado debería ser rechazado si no produce un f^2 significativo. f^2 determina si la influencia de una variable latente particular sobre un constructo dependiente tiene un impacto sustantivo, siendo los niveles de f^2 de 0.02, 0.15 y 0.35.

$$f^2 = \frac{R^2_{incluida} - R^2_{excluida}}{1 - R^2_{incluida}} \qquad (28)$$

En cuanto a la segunda pregunta, ésta representa los coeficientes *path* o pesos de regresión estandarizados, indicando la fuerza relativa de las relaciones estadísticas. La varianza explicada en un constructo endógeno por otra variable latente viene dado por el valor absoluto del resultado de multiplicar el coeficiente *path* (β) por el correspondiente coeficiente de correlación entre ambas variables. Por ejemplo, en la relación entre A y B ($A \rightarrow B$), $\beta = 0.5$ y la correlación existente entre ambos es de 0.56, por lo que tendríamos como resultado 0.5 x 0.56 = 0.28, es decir, el 28% de la varianza de B es explicado por la variable latente A.

Bondad del ajuste

Las medidas existentes de bondad del ajuste están relacionadas con la capacidad del modelo para explicar las covarianzas de la muestra y asumir, por tanto, que todos los indicadores son reflectivos. En el modelo PLS no existen porque dicho modelo tiene una función objetivo distinta, no presupone ningún tipo de distribución de los datos y permite el empleo de variables manifiestas formativas. No obstante, es posible el empleo de técnicas no paramétricas de remuestreo para examinar la estabilidad de las estimaciones ofrecidas por el modelo PLS, como *Jackknife* y *Bootstrap* (preferible). Ambas ofrecen los errores estándar y los valores *t*. Los coeficientes *path* y, por extensión, las hipótesis planteadas aceptadas serán aquellas que sean significativas. Se utiliza una distribución *t de Student* de dos colas con *n-1* grados de libertad, donde *n* es el número de submuestras.

5.3 Análisis semántico

Las técnicas de análisis semántico se basan en representar los documentos como vectores de palabras (Salto y McGill, 1983), siendo las palabras un corpus lingüístico representativo de un tema.

El uso de los corpus textuales se ve motivado por dos motivos fundamentales:

1 La terminografía pretende la identificación y recopilación de los términos que los especialistas utilizan en realidad. Las dos opciones que existen para determinar el corpus lingüístico son: la consulta directa con los especialistas y el estudio detallado de las producciones lingüísticas que los especialistas crean para comunicarse entre ellos o con el resto de la sociedad. La primera de las opciones, la consulta directa con los especialistas, es insustituible y muy valiosa, pero puede presentar problemas prácticos por dos motivos: primero, no siempre es posible tenerlos a la disposición y segundo, los especialistas pueden tener dificultades en explicar el significado y el uso de la lengua que usan, al fin y al cabo, de forma intuitiva. Estos inconvenientes justifican el uso de medios informáticos.

2 El segundo de los motivos que hace imprescindible el uso del corpus en terminografía se refiere a la dimensión conceptual de los términos. Para poder identificar y recopilar los términos que los especialistas usan en realidad, los terminógrafos necesitan estudiar las estructuras de conocimiento (los conceptos y sus relaciones) que los términos representan y comunican. Es decir, los terminógrafos deben adquirir conocimiento sobre el dominio de especialidad, para poder sistematizar su terminología. Al igual que en el caso anterior, poseen dos opciones: la consulta con los especialistas y los medios informáticos. La consulta con especialistas, como única fuente de información, puede presentar problemas similares a los mencionados anteriormente. En este caso no se trata de que éstos no sean capaces de detallar aspectos lingüísticos o contextuales de los términos, ya que nos referimos a conocimiento del área en la que son especialistas. Sin embargo, puede darse el caso de que por muy bien que, obviamente, conozcan y dominen el ámbito de especialidad en el que trabajan, tengan dificultades en explicarlo, y aún más, en hacerlo de forma útil para el terminógrafo, es decir, de forma clara, completa y consistente (Ahmad, 1996). El estudio de la documentación especializada puede servir, por otra parte, para facilitar la comunicación entre el especialista y el terminógrafo (Meyer y Mackintosh, 1996). En la mayoría de las ocasiones, será más fácil para el especialista entender las preguntas del terminógrafo si éstas se refieren a un uso, significado, definición, etc. específico de un término, que el especialista puede ver en un texto.

Formalmente, en el modelo vectorial se intenta recoger la relación de cada documento D_i, de una colección de N documentos, con el conjunto de las m características de la colección. Un documento puede considerarse como un vector que expresa la relación del documento con cada una de esas características.

$$D_i \rightarrow \overline{d}(c_{i1}, c_{i2}, \ldots c_{in}) \qquad (29)$$

Observamos que el vector identifica en qué grado el documento D_i satisface cada una de las m características. En otras palabras el vector, c_{ik} es un valor numérico que expresa en qué grado el documento D_i posee la característica k. La noción de "característica" suele

concretarse en la ocurrencia de determinadas palabras o términos en el documento, aunque nada impide tomar en consideración otros aspectos (Landauer, 2002). Si se consideran las palabras como características definitorias del documento, el proceso que debe seguir el sistema de clasificación se inicia con la selección de aquellas palabras útiles que permitan discriminar unos documentos de otros. En este punto, debemos señalar que no todas las palabras contribuyen con la misma importancia en la caracterización del documento. Desde el punto de vista lingüístico aplicado a la recuperación o clasificación de documentos, existen lexemas casi vacíos de contenido semántico, como los artículos, las preposiciones o las conjunciones. Estos lexemas son conocidos como palabras funcionales en la tradición lingüística y como *stop words* en el procesamiento del lenguaje natural. Estas palabras, que en español comúnmente son entre 100 y 200, son poco útiles para el proceso de clasificación. También son poco importantes aquellas palabras que por su frecuencia de aparición en toda la colección de documentos pierden su poder de discriminación, es por ello que, o son eliminadas, o son ponderadas con muy bajo peso estadístico.

Una vez seleccionado el conjunto de términos caracterizadores de la colección de documentos, es necesario calcular el valor de cada elemento del vector del documento. El caso más simple es utilizar una aproximación binaria, de forma que si en el documento D_i aparece el término k, el valor c_{ik} sería 1, y en caso contrario sería 0.

El principal problema del modelo vectorial es el elevado número de dimensiones del espacio de características (una por palabra). Por este motivo es deseable proyectar los documentos en un sub-espacio de menor número de dimensiones donde la estructura semántica de los documentos se clarifique (Cai *et al*., 2005). En este espacio semántico de menor dimensión se pueden aplicar algoritmos tradicionales de clustering, como *spectral clustering* (Shi y Malik, 2000; Ng *et al*., 2001), *Latent Semantic Indexing* (Zha *et al*., 2001), y clustering basado en factorizaciones no negativas (Xu *et al*., 2003; Xu y Gong, 2004). En particular, la técnica conocida como *Latent Semantic Indexing* (LSI) descompone la matriz de términos-documentos usando una descomposición en valores singulares, Figura 14, reduciendo la alta dimensionalidad del problema y construyendo nuevas características como combinación de las originales (Deerwester *et al*., 1990; Abedin and Sohrabi, 2009).

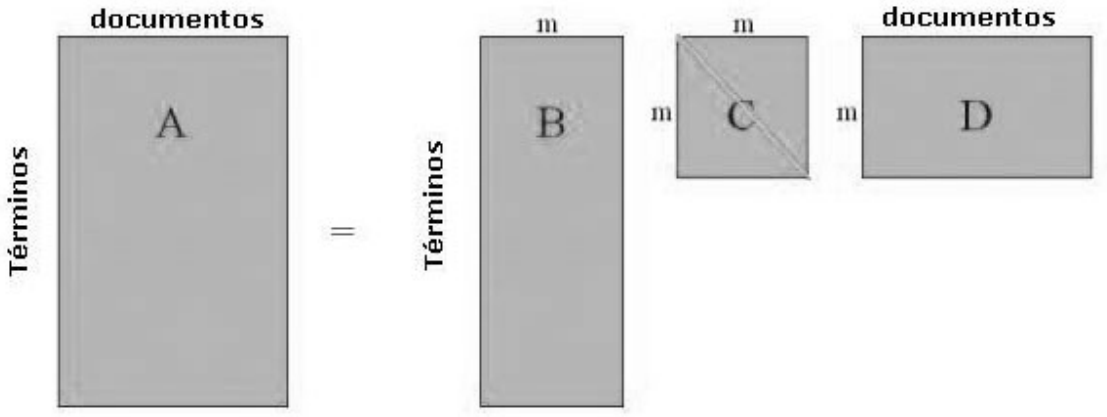


Figura 14. Descomposición de la matriz principal en las dos matrices de vectores singulares y una matriz diagonal de valores singulares

Sea X la matriz de términos por documentos (nxm) que representa las frecuencias de aparición de los términos de indexación en los documentos de una colección (la columna *i* de X contiene el vector de frecuencia del documento d_i). La matriz X puede descomponerse como el producto de tres matrices:

$$X = T_0 S_0 D_0^T \qquad (30)$$

De tal forma que S_0 es una matriz diagonal y las columnas de T_0 y D_0 son ortonormales, es decir, se verifica $T_0^T T_0 = I$ y $D_0^T D_0 = I$, siendo I la matriz identidad. La ecuación (30) es la descomposición en valores singulares: T_0 y D_0 son las matrices de los vectores singulares y S_0 contiene los valores singulares. Intercambiando filas en T_0 y D_0 se pueden ordenar los valores de S_0 por su magnitud, de tal forma que el mayor valor esté en la fila y columna primeras. La ventaja de esta descomposición es que permite una estrategia para encontrar aproximaciones óptimas de X usando matrices más pequeñas (con rangos menos que X). Si de S_0 se mantienen únicamente los k valores de mayor magnitud, asignando a cero el resto de los valores, entonces el producto de las matrices resultantes es otra matriz $\hat{X}$, de rango k, que es además la matriz de rango k que mejor aproxima X en el sentido de mínimos cuadrados. Eliminando de S_0 las filas y columnas que contienen solamente ceros y

eliminando de T_0 y D_0 las columnas correspondientes, se obtiene el modelo reducido de la ecuación (31) utilizado por LSI:

$$X \approx \hat{X} = TSD^T \tag{31}$$

En el modelo vectorial clásico se representan los documentos como vectores en el espacio de los términos y se pueden calcular las similitudes entre documentos. La similitud entre dos documentos d_i y d_j se define por el producto escalar de las columnas *i* y *j* de la matriz $\hat{X}$. En otras palabras, la matriz que se obtiene del producto $\hat{X}^T\hat{X}$ es la matriz que contiene las similitudes para cada par de documentos. Siguiendo la definición de $\hat{X}$:

$$\hat{X}^T\hat{X} = (TSD^T)^T TSD^T = DS^T T^T TSD^T = DS^T SD^T = DSS^T D^T = DS(DS)^T \tag{32}$$

Es decir, la similitud entre dos documento d_i y d_j está definida por el producto escalar entre las filas i y j de la matriz DS:

$$sim(d_i, d_j) = (DS)_{i,\cdot}[(DS)_{j,\cdot}]^T \tag{33}$$

En la que $(DS)_{i,\cdot}$ representa la fila *i* de la matriz *DS*. Por tanto, las filas de la matriz *DS* pueden considerarse como vectores que representan los documentos de la colección. Aquellos documentos que tienen muchas palabras en común se encuentran semánticamente próximos, mientras que los que tienen pocas en común están semánticamente lejanos. Dado que *DS* es una matriz *mxk* (pues *D* tienen dimensiones *mxk* y *S* dimensiones *kxk*), estos vectores tienen *k* elementos. Por tanto, el espacio vectorial considerado tiene una base de *k* vectores y se le puede considerar como un espacio de factores o conceptos en vez de términos. Dado que normalmente se elige un valor para *k* mucho menor que el número de términos de indexación ($k<<n$), la LSI realiza una reducción de las dimensiones del espacio original. No obstante, los vectores de factores son mucho más densos que los vectores usados en el modelo vectorial clásico, por lo que generalmente LSI no es más eficiente en cuanto a los requisitos de memoria y tiempo de cálculo. Además, la indexación LSI es mucho más costosa por la necesidad de realizar la descomposición en valores singulares de las matrices iniciales. Otro de los problemas está relacionado con la selección del número de factores o dimensiones a usar (el parámetro *k*). La elección óptima sería un

valor que fuera lo suficientemente grande como para reflejar la estructura real de los datos (el contenido conceptual de los documentos) y que, al mismo tiempo, fuese relativamente pequeño para obtener los efectos deseados de reducción del número de dimensiones e introducción de la conceptualidad. En la práctica se suele elegir de manera experimental.

LSI hace tres suposiciones básicas:

1 La información semántica puede obtenerse a partir de una matriz de co-ocurrencia de palabras y documentos

2 La información semántica se obtiene tras una reducción de número de dimensiones

3 Las palabras y documentos pueden representarse en un espacio Euclídeo

Otros modelos denominados generativos representan factores latentes o tópicos como una nueva variable que tiene relaciones de probabilidad condicionada con los términos y los documentos. El modelo de la indexación probabilística por semántica latente (*probabilistic Latent Semantic Indexing*, pLSI) fue introducido por Hofmann (1999). También en este modelo se representan los documentos en un espacio reducido que es el espacio semántico latente. pLSI se basa en un modelo general de variables latentes para datos de co-apariciones que asocia una variable latente de clase no observada a cada observación. Estas variables pueden considerarse como factores que reflejan conceptos semánticos que están asociados a las apariciones de las palabras en los documentos. Cada uno de ellas actúa como indicador de varias palabras, aquellas que describen su significado conceptual. Dado que le número de factores (variables de clase) es mucho menor que el número de términos, también la pLSI realiza una reducción de las dimensiones del espacio vectorial.

Sea $D=\{d_1, d_2, ..., d_m\}$ una colección de documentos y $T=\{t_1, t_2, ..., t_n\}$ el conjunto de términos de indexación de D. Como observaciones se consideran los pares (d_i,t_j) que reflejan la asignación de los términos a los documentos. $P(d,t)$ denota la probabilidad de que se observe un término en un documento. Asignando una variable z_k de un conjunto de variables latentes de clase $Z=\{z_1, z_2, ..., z_r\}$ a cada par de observaciones (d_i,t_j) y aplicando la regla de la probabilidad condicional se obtiene:

$$P(d,t)=\sum_{z_k\in Z} P(t,z_k,d)=\sum_{z_k\in Z} P(d)\,P(z_k\mid d)\,P(t\mid z_k,d) \tag{34}$$

El modelo utiliza dos suposiciones de independencia i) las observaciones (d_i,t_j) se generan de forma independiente y ii) condicionado a una clase latente z_k, los términos se generan de forma independiente de los documentos. Con la segunda suposición, la ecuación (34) se transforma en:

$$P(d,t)=\sum_{z_k\in Z} P(d)\,P(z_k\mid d)\,P(t\mid z_k) \tag{35}$$

Utilizando esta ecuación, la probabilidad de que un término aparezca en un documento dado puede calcularse de la siguiente forma:

$$P(d,t)=\frac{P(d,t)}{P(d)}=\sum_{z_k\in Z} P(z_k\mid d)\,P(t\mid z_k) \tag{36}$$

es decir, en función de las probabilidades $P(t|z)$ y $P(z|d)$.

Aprovechando este hecho, la idea del modelo consiste en representar cada documento por un vector de sus factores latentes (las probabilidades $P(z|d)$), en vez de por sus términos. Utilizando estos vectores, se puede calcular la similitud entre documentos de igual forma que en el modelo vectorial clásico. Así pues, el objetivo del proceso de indexación consiste en estimar las probabilidades $P(z_k|d_i)$ para cada $z_k\in Z$ y cada documento $d_i\in D$.

Para estimar estas probabilidades se maximiza la función de similitud logarítmica (*log-likelihood function*):

$$L=\sum_{d_i\in D}\sum_{t_j\in T} g_j\left(dtf_i\right)\log P(d_i,t_j) \tag{37}$$

Siendo $dtf_i=\{tf_{i1},\ldots,tf_{in}\}$ un vector sobre el conjunto de los términos de indexación y tf_i las frecuencias de aparición del término t_j en el documento d_i. Por su parte, g_j, con $j=1\ldots n$, es una familia de funciones que devuelve el valor asignado al término tj (por ejemplo, $g_j(dtf_i)=tf_{ij}$).

Invirtiendo la probabilidad *P(z|d)* con la ayuda de la regla de Bayes, se obtiene una versión simétrica del modelo descrito por la ecuación (35).

$$P(d,t)=\sum_{z_k\in Z} P(z_k)\,P(d\mid z_k)\,P(t\mid z_k) \tag{38}$$

Aplicando (38) en (37), la función a maximizar es:

$$L=\sum_{d_i\in D}\sum_{t_j\in T} g_j\left(dtf_i\right)\log\sum_{z_k\in Z} P(z_k)P(d_i\mid z_k)P(t_j\mid z_k) \tag{39}$$

Para estimar las probabilidades *P(t|z)*, *P(d|z)* y *P(z)*, se aplica el Algoritmo de Maximización de la Expectación, AME (*Expectation Maximization Algorithm*). El algoritmo tiene dos pasos que se alternan: i) la computación de las probabilidades "a posteriori", y ii) un paso de maximización.

De la Regla de Bayes se obtiene la ecuación para el primer paso:

$$P(z_k\mid d_i,t_j)=\frac{P(z_k,d_i,t_j)}{P(d_i,t_j)}=\frac{P(z_k)P(d_i\mid z_k)\,P(t_j\mid z_k)}{\sum_{z_v\in Z} P(z_v)P(d_i\mid z_v)\,P(t_j\mid z_v)} \tag{40}$$

en la que *P(z|d,t)* es la probabilidad de que un término *t* en un documento *d* esté explicado por el factor *z*.

En el paso de maximización se estiman las probabilidades $P(d_i,t_j)$ en función del conjunto de observaciones:

$$P(d_i,t_j)=\frac{g_j\left(dtf_i\right)}{\sum_{t_u\in T}\sum_{d_v\in D} g_u\left(dtf_v\right)} \tag{41}$$

Con ello, las ecuaciones para estimar $P(z_k)$, $P(t_j|z_k)$ y $P(d_i|z_k)$ son:

$$P(t_j\mid z_k)=\frac{P(t_j,z_k)}{P(z_k)}=\frac{\sum_{d_i\in D} P(d_i\mid t_j)\,P(z_k\mid d_i,t_j)}{\sum_{d_v\in D}\sum_{t_u\in T} P(d_v\mid t_u)\,P(z_k\mid d_v,t_u)}=\frac{\sum_{d_i\in D} g_j\left(dtf_i\right)P(z_k\mid d_i,t_j)}{\sum_{d_v\in D}\sum_{t_u\in T} g_u\left(dtf_v\right)P(z_k\mid d_v,t_u)} \tag{42}$$

$$P(d_i \mid z_k) = \frac{P(d_i, z_k)}{P(z_k)} = \frac{\sum_{t_j \in T} P(d_i \mid t_j)\, P(z_k \mid d_i, t_j)}{\sum_{d_v \in D} \sum_{t_u \in T} P(d_v \mid t_u)\, P(z_k \mid d_v, t_u)} = \frac{\sum_{t_j \in T} g_j(dtf_i) P(z_k \mid d_i, t_j)}{\sum_{d_v \in D} \sum_{t_u \in T} g_u(dtf_v) P(z_k \mid d_v, t_u)} \quad (43)$$

$$P(z_k) = \sum_{d_v \in D} \sum_{t_u \in T} P(d_v, t_u) P(z_k \mid d_v, t_u) = \frac{\sum_{d_v \in D} \sum_{t_u \in T} g_u(dtf_v) P(z_k \mid d_v, t_u)}{\sum_{d_v \in D} \sum_{t_u \in T} g_u(dtf_v)} \quad (44)$$

Alternando estos dos pasos, el algoritmo converge hacia un máximo local de la función (39) y estima los valores de las probabilidades *P(z)*, *P(t|z)* y *P(d|z)*.

El proceso de recuperación en el modelo pLSI está basado en una representación de los documentos por vectores sobre los cuales se calculan las relevancias mediante una función de similitud. Existen dos métodos para obtener la representación vectorial:

1. **pLSI-U:** se representa cada documento d_i por un vector sobre los términos de indexación $d_i=(w_{i1}, ..., w_{in})$ que aproxima el vector de frecuencias dtf_i y cuyos elementos se obtienen mediante la siguiente ecuación:

$$w_{ij} = P(t_j \mid d_i) = \sum_{z_k \in Z} P(t_j \mid z_k)\, P(z_k, d_i) = \sum_{z_k \in Z} \frac{P(t_j \mid z_k)\, P(d_i \mid z_k) P(z_k)}{P(d_i)} \quad (45)$$

2. **pLSI-Q:** se representa cada documento d_i por un vector en el espacio de las variables latentes $d_i=(w_{i1}, ..., w_{in})$, siendo:

$$w_{ik} = P(z_k \mid d_i) = \frac{P(d_i \mid z_k) P(z_k)}{P(d_i)} \quad (46)$$

Las probabilidades $P(d_i)$ se estiman a partir de las observaciones por la fórmula:

$$P(d_i) = \sum_{t_j \in T} P(d_i, t_j) = \frac{\sum_{t_j \in T} g_j(dtf_i)}{\sum_{t_u \in T} \sum_{d_v \in D} g_u(dtf_v)} \quad (47)$$

Los principales problemas de la pLSI son:

- La indexación es costosa debido al uso del *AME*. Normalmente son necesarias 40-60 iteraciones y cada iteración requiere del orden de *R*r* operaciones, siendo *r* el número

de variables latentes y *R* la suma sobre todos los documentos del número de términos distintos que aparecen en ellos (Hofmann, 1999).

- Los requerimientos de memoria en el método pLSI-U son importantes debido a que los vectores tienen una dimensión grande (correspondiente al número de términos de indexación) y normalmente son densos. En pLSI-Q, los requerimientos son menores y dependen del número de dimensiones del espacio semántico latente, es decir, de *r*.
- El modelo requiere la selección del parámetro r (el número de variables latentes utilizados). La selección se realiza normalmente de forma empírica.

Desde un punto de vista más intuitivo, el funcionamiento del modelo se detalla en la Figura 15. Los tópicos o variables latentes 1 y 2 se ilustran como bolsas que contienen palabras con diferentes distribuciones. Los documentos que figuran a la derecha se obtienen a partir de las palabras que hay en cada una de las bolsas, siendo su uso proporcional al peso asignado a cada palabra. Por ejemplo, los documentos 1 y 3 se generan muestreando las bolsas correspondientes a los tópicos 1 y 2, respectivamente, en tanto que el documento 2 se genera como una mezcla equitativa de ambos tópicos.

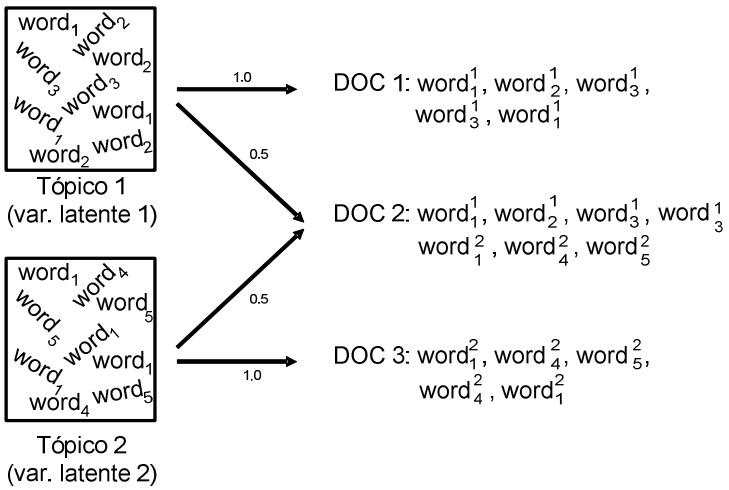


Figura 15. Proceso subyacente en los modelos probabilísticos de semántica latente

Obsérvese que cada palabra lleva asociado un superíndice que indica de qué tópico fueron extraídas. Esto significa que una misma palabra puede estar presente en varios tópicos,

permitiendo de este modo tratar con palabras polisémicas, es decir, palabras que pueden tener varios significados diferentes.

El modelo pLSI se mejoró introduciendo una variable aleatoria de Dirichlet para modelar los documentos, denominándose al modelo resultante modelo de asignación latente de Dirichltet (LDA, *Latent Dirichlet Allocation*), (Blei *et al*., 2003). A diferencia del pLSI, cada documento no es una mezcla fija de tópicos sino una mezcla aleatoria. No obstante, igual que pLSI, asume la existencia de un conjunto de variables latentes o tópicos predefinidos. Cada variable latente se representa como una distribución multinomial sobre el conjunto de palabras del vocabulario.

El modelo queda descrito por la siguiente expresión:

$$P(d) = \int_{\theta} \left[\prod_n \sum_z P(w_n \mid z_n; \phi) P(z_n \mid \theta) \right] P(\theta; \alpha)\, \delta\theta \qquad (48)$$

siendo $P(\theta;\alpha)$ una distribución de Dirichlet, α un hiper-parámetro, $P(z_n;\theta)$ una multinomial que indica el grado en que el tema z_n es tratado en un documento y ϕ una matriz de clases por palabras del vocabulario, con un hiper-parámetro β. De esta manera, la probabilidad de un documento depende de las probabilidades de que sus palabras denoten ciertas categorías dentro de una distribución de Dirichlet. Para aprender e inferir en el modelo usan un algoritmo EM análogo al modelo pLSI. Para facilitar el cálculo, Griffiths y Steyvers (2004) proponen el uso del algoritmo Gibbs sampling

$$p(z_i = j | z_{i-1}, w) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i}^{(d_i)} + T\alpha} \qquad (49)$$

Esta distribución representa la probabilidad de que la palabra w_i sea asiganada al tópico j dada todas las demás asignaciones z_{i-1}. Las cantidades $n_{-i,j}^{(w_i)}$ y $n_{-i,j}^{(\cdot)}$ representan el número de veces que la palabra w_i ha sido asignada al tópico j y el número total de palabras asignadas al tópico j, respectivamente. Las cantidades $n_{-i,j}^{(d_i)}$ y $n_{-i}^{(d_i)}$ representan el número de veces que la palabra w_i se ha asignado al tópico j en el documento d_i y el número de palabras en el documento d_i asignadas al tópico j. Los hiper-parámetros α y β se

computan usando el método descrito por Griffiths y Steyvers (2004), esto es, $\beta = 0.01$ y $\alpha = 50/T$.

Considerando T tópicos, la probabilidad de que la palabra w_i aparezca en un documento es:

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j) \quad (50)$$

$P(z_i=j)$ proporciona la probabilidad de elegir una palabra del tópico j en un documento, que varía entre diferentes documentos. Intuitivamente, $P(w|z)$ indica qué palabras son importantes en un tópico y $P(z)$ la importancia de esos tópicos en un documento.

Capítulo 6. Resultados: modelo de participación en comunidades de software de código abierto

6.1 Caso de estudio

Como caso de estudio se abordarán las comunidades de soporte de Linux para sistemas embebidos. Un sistema embebido es un dispositivo que lleva un procesador dentro, aunque el usuario final no sea necesariamente consciente de que existe. Ejemplos de sistemas embebidos se encuentran permanentemente a nuestro alrededor, desde dispositivos de electrónica de consumo (PDA, teléfonos móviles, reproductores MP3, cámaras digitales, etc.), pasando por la electrónica de vehículos de automoción y sistemas de seguridad y vigilancia, hasta equipos de comunicación e instrumentación o equipos aeroespaciales (Abbott, 2003).

De los 6.2 billones de procesadores fabricados en 2002, menos del 2% fueron a parar a PCs, Macs o estaciones de trabajo Unix. El otro 98% formó parte de sistemas embebidos (Ganssle, 2003) en aplicaciones que van desde juguetes y semáforos hasta equipos de consumo o aplicaciones domóticas o satelitales. Al comienzo de su desarrollo, los sistemas embebidos no incorporaban sistemas operativos. El desarrollo del software se realizaba actuando directamente sobre el hardware del sistema, sin incluir prácticamente capacidad de procesamiento multitarea ni de acceso en red. A medida que los sistemas embebidos fueron aumentando su complejidad y su capacidad de procesamiento, también se incrementaron los requerimientos sobre ellos: capacidad de procesamiento multitarea, gestión de memoria y de procesos, comunicación entre tareas y procesos, acceso a redes de comunicación, etc. Un sistema embebido puede, por ejemplo, incluir un servidor web que permita la configuración remota de un equipo mediante una página accesible por Internet, así como accesos remotos de mantenimiento y actualización (Raghavan *et al*., 2006; Yaghmour, 2003). Todos estos requerimientos obligaron el uso de sistemas operativos en los sistemas embebidos. Aunque actualmente existen varios sistemas operativos embebidos disponibles (Wind River's VxWorks, Microsoft Windows CE, QNX Neutrino, etc), Linux es claramente el líder en este tipo de sistemas. La Figura 16 muestra que en el año 2006 Linux embebido rozó el 50% de la cuota de mercado.

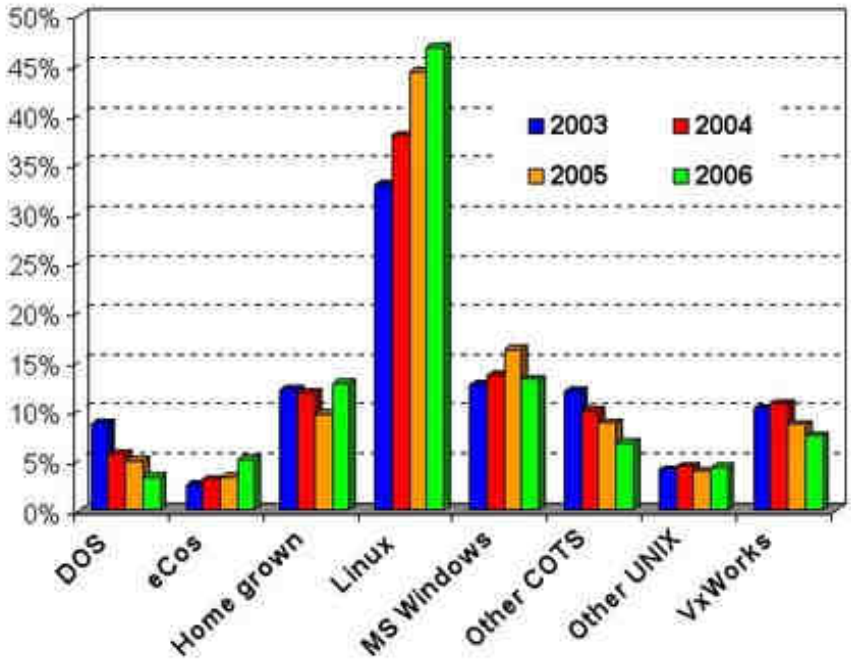


Figura 16. Evolución de los sistemas operativos para sistemas embebidos **(fuente: Linuxdevices.com, Snapshot of the embedded Linux market -- Mayo, 2006)**

Las principales ventajas de Linux embebido frente a otros sistemas operativos propietario son independencia del proveedor, plazo de comercialización y bajo coste. No obstante, hay que tener en cuenta que la licencia pública general (GPL), bajo la que Linux se distribuye, estipula que la distribución del software original o de versiones modificadas debe realizarse bajo las mismas condiciones definidas por GPL. En particular, el comprador o cliente tiene derecho a recibir el código fuente así como los derechos de modificar y redistribuir el software (Henkel, 2003; Free Software Foundation: GNU General Public License, 2006). De todos modos, GPL no obliga a hacer público el software modificado y no excluye poder vender el software (eso sí, sin royalties por unidad vendida).

Debido a la gran heterogeneidad de los sistemas embebidos, no existe una versión estándar de Linux, sino que hay que hablar de muchas distribuciones de Linux que cubren una o varias arquitecturas procesadoras. Todas ellas tienen en común los mismos componentes básicos, incluyendo el kernel de Linux, drivers, comandos, entornos de ventanas, utilidades básicas y librerías. Las diferencias entre las distribuciones se centran normalmente en cuál de los cientos de utilidades existentes se incluyen en los módulos añadidos (tanto en código

fuente como propietario), en las modificaciones del kernel y en la gestión de los procesos de instalación, configuración, mantenimiento y actualización. La Figura 17 muestra las distribuciones de Linux más utilizadas.

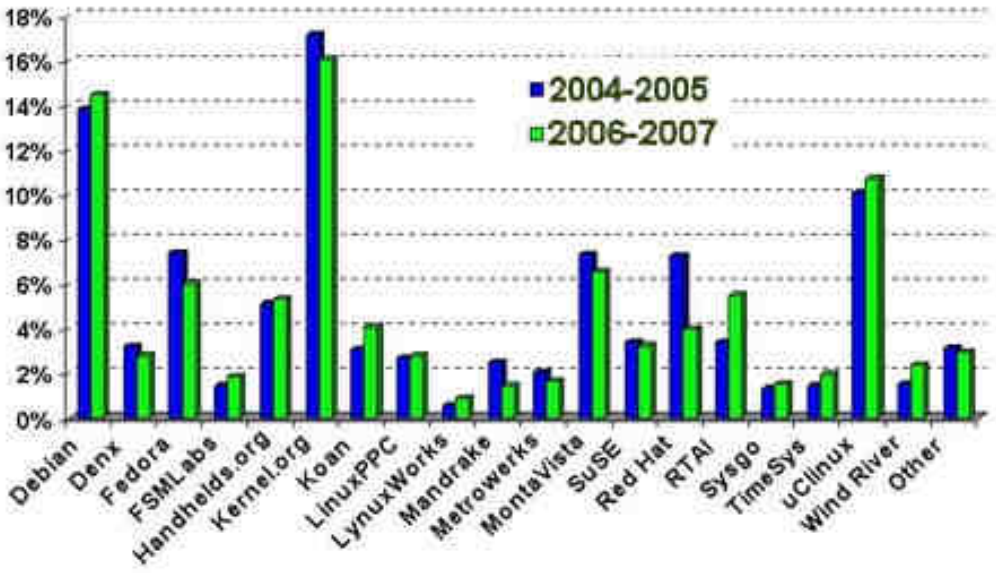


Figura 17. Distribuciones de Linux más utilizadas **(fuente: Linuxdevices.com, Snapshot of the embedded Linux market -- Mayo, 2006)**

El Linux embebido muestra algunas diferencias significativas con respecto al Linux tradicional para PC. En primer lugar, los sistemas embebidos incorporan una variedad más amplia de dispositivos de entrada y salida que los PCs. Eso supone que los programadores de sistemas embebidos tienen que trabajar frecuentemente con el hardware del sistema. En segundo lugar, y en contraste con otros proyectos de software de código abierto, la mayoría de las contribuciones en este campo no vienen de voluntarios o aficionados sino de firmas comerciales, muchas de las cuales son empresas dedicadas a Linux embebido. Por último, el hecho de que Linux se distribuya con licencia GPL hace que las empresas consideren revelar sus propios desarrollos software, a diferencia del caso de usar un software propietario. Todas estas características fomentan el trabajo colaborativo y justifican la necesidad de analizar las posibilidades de soporte mediante medios electrónicos.

La Figura 17 muestra que Debian es una de las distribuciones más populares y es la que se utilizará como caso de estudio, debido fundamentalmente a que posee una gran cantidad de

distribuciones para diferentes entornos y arquitecturas procesadoras. Hablando con propiedad, el proyecto Debian es una distribución GNU/Linux que tiene su origen en Agosto de 1993 (Murdock, 1993). La distribución Debian está formada por paquetes (más de 18000 en la versión 4.0) que contienen no sólo el sistema operativo sino otras muchas herramientas y aplicaciones (entorno de escritorio, navegadores, gestión de bases de datos, procesadores de textos, etc.). El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, siendo llevado adelante por los propios usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

El proyecto Debian cuenta con tres documentos fundadores (Mateos-García y Stein, 2008):

- El Contrato Social de Debian, que define un sistema de base por los cuales el proyecto y sus desarrolladores tratan los asuntos.
- Las Directrices de software libre de Debian, que definen los criterios del "software libre" y dictan qué software se aceptable para la distribución, según lo referido al contrato social. Estas pautas también se han adoptado como la base de la definición del Open Source.
- La Constitución de Debian, que describe la estructura de la organización para la toma de decisiones de manera formal dentro del proyecto. Enumera el poder y las responsabilidades del líder de proyecto Debian, de la secretaría y de los desarrolladores en general.

Actualmente, el proyecto incluye más de mil desarrolladores. Cada uno de ellos posee algún lugar en el proyecto ya sea relacionado con paquetes: mantenimiento, documentación, control de calidad o relacionado con la infraestructura del proyecto: coordinación de lanzamientos, traducciones de web, etc. Los mantenedores de los paquetes tienen un excedente de la jurisdicción sus propios paquetes. El proyecto mantiene asimismo listas de correo y utiliza sistemas de seguimiento de *bugs* para informar a toda la comunidad.

Anualmente se elige un líder del proyecto mediante una elección en la que pueden participar todos los desarrolladores. El líder del proyecto Debian tiene varias atribuciones especiales, pero estas atribuciones están lejos de ser una decisión absoluta y se utiliza raramente. Bajo resolución general, los desarrolladores pueden, entre otras cosas, reelegir al líder, invertir una decisión de éste o de sus delegados, o enmendar la constitución y otros documentos fundacionales.

Como caso de estudio se utilizarán las doce comunidades detalladas en la Tabla 5, que se refieren a distribuciones Debian para diferentes entornos y arquitecturas procesadoras embebidas. El período de análisis depende de cada comunidad (véase última columna de la Tabla 5). Como criterio se ha incluido el período en el que la comunidad ha tenido una actividad destacable y continua en cada año. La actividad será analizada anualmente, lo que proporciona un total de 110 casos válidos.

Tabla 5. Listado de comunidades consideradas en el caso de estudio propuesto

	URL	**Descripción**	**Periodo**
Debian port a m68k (D-68k)	http://lists.debian.org/debian-68k/	Adaptación de Debian GNU/Linux para Motorola 68k. Actualmente funciona en los procesadores 68020, 68030, 68040 y 68060.	98-08
Debian port a Alpha (D-Alpha)	http://lists.debian.org/debian-alpha/	Adaptación de Debian GNU/Linux a la familia de procesadores Alpha. Es una de las adaptaciones más veteranas y estable.	98-08
Debian port to AMD64 (D-AMD64)	http://lists.debian.org/debian-amd64/	Adaptación a los procesadores de 64 bits AMD64. El objetivo es soportar espacios de usuario tanto de 32 como de 64 bits en esta arquitectura. Esta adaptación permite usar los Opteron de 64 bits de AMD, los procesadores Athlon y Sempron, y lo procesadores de Intel con soporte EM64T, incluyendo Pentium D y varias series de Xeon y Core2.	04-08

	URL	Descripción	Periodo
Debian port to ARM (D-ARM)	http://lists.debian.org/debian-arm/	Adaptación a procesadores ARM, muy utilizados en sistemas embebidos.	99-08
Debian port to BSD (D-BSD)	http://lists.debian.org/debian-bsd/	Adaptación del sistema GNU Debian al núcleo FreeBSD.	01-08
Debian port to HPPA (D-HPPA)	http://lists.debian.org/debian-hppa/	Se trata de la adaptación a la arquitectura PA-RISC de Hewlett-Packard.	01-08
Debian port to Hurd (D-HURD)	http://lists.debian.org/debian-hurd/	El GNU Hurd es un sistema operativo totalmente nuevo puesto en marcha por el grupo GNU.	99-08
Debian port to IA64 (D-IA64)	http://lists.debian.org/debian-ia64/	Es la adaptación a la primera arquitectura de 64 bits de Intel.	01-08
Debian port to MIPS (D-MIPS)	http://lists.debian.org/debian-mips/	Adaptación de Debian GNU/Linux a la arquitectura MIPS, usada en máquinas SGI y DECstations de Digital.	99-08
Debian port to PowerPC (D-PPC)	http://lists.debian.org/debian-powerpc/	Adaptación de Debian GNU/Linux a la arquitectura PowerPC, permitiendo implementaciones tanto de 64-bit como 32-bit.	99-08
Debian port to S390 (D-S390)	http://lists.debian.org/debian-s390/	Adaptación de Debian GNU/Linux para IBM S/390	01-08
Debian port to SPARC (D-SPARC)	http://lists.debian.org/debian-sparc/	Esta adaptación funciona sobre la gama de estaciones de trabajo Sun SPARCstation, así como sobre alguna de	98-08

	URL	Descripción	Periodo
		sus sucesoras en la arquitectura sun4.	

6.2 Modelo de participación

El primer paso para la elaboración de un modelo de participación consiste en la extracción de la información de las comunidades consideradas en la Tabla 5. Todas ellas son accesibles desde la URL http://lists.debian.org/ports.html, desde donde puede accederse a los mensajes archivados por año y mes. La Figura 18 detalla el diagrama de flujo de la extracción de información.

Por cada de uno de los meses se hace un doble procesamiento:

- En primer lugar, se extraen los usuarios que han mandado mensajes, estableciendo las asociaciones alias – correo electrónico con el fin de agrupar aquellos usuarios que usan correos electrónicos con ligeras variantes, o que cambian su alias. Como resultado final se obtendrá una lista definitiva de usuarios, que conforman los nodos de la red social correspondiente a esa comunidad y ese año.
- En una segunda pasada, y utilizando la lista de usuarios anterior, se analizan cada uno de los hilos de discusión para definir los arcos que unen los nodos de la red, siguiendo el criterio establecido de que cada usuario que envía un mensaje a un hilo de discusión contesta realmente a todos los demás usuarios que han intervenido previamente.

Este procesamiento se ha realizado desde el entorno MATLAB, elaborando un programa capaz de procesar los fuentes html descargados por el propio programa desde las páginas web de cada comunidad. Los usuarios se extraen de las cabeceras de los mensajes. La Figura 19 muestra un ejemplo de cabecera. La línea que comienza con "*From*" contiene tanto el alias del usuario como su correo electrónico. De este modo, y según se indica en el diagrama de flujo de la Figura 18, se obtiene como resultado final la lista de usuarios de un año. A continuación se procesan los hilos de discusión para definir las conexiones entre los nodos (usuarios) de la red social.

Una vez generada la información de la red social relativa a una comunidad y un año, se genera un archivo en formato Pajek (Nooy *et al*., 2005) para su posterior procesamiento con este software. El proyecto Pajek, traducción del slovenio 'Araña', fue creado en 1996 por Vladimir Batagelj y Andrej Mrvar, de la Universidad de Ljubljana, Slovenia. Este software funciona bajo la plataforma de Windows y es de libre distribución para uso no comercial, permitiendo el análisis y visualización de redes sociales.

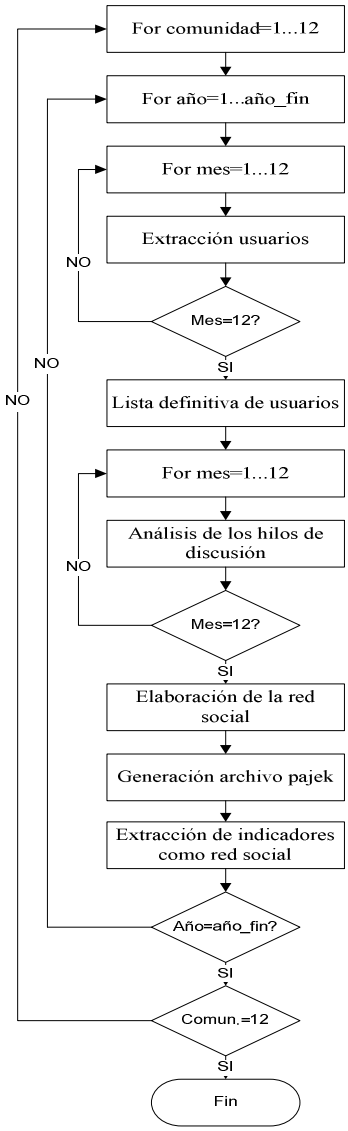
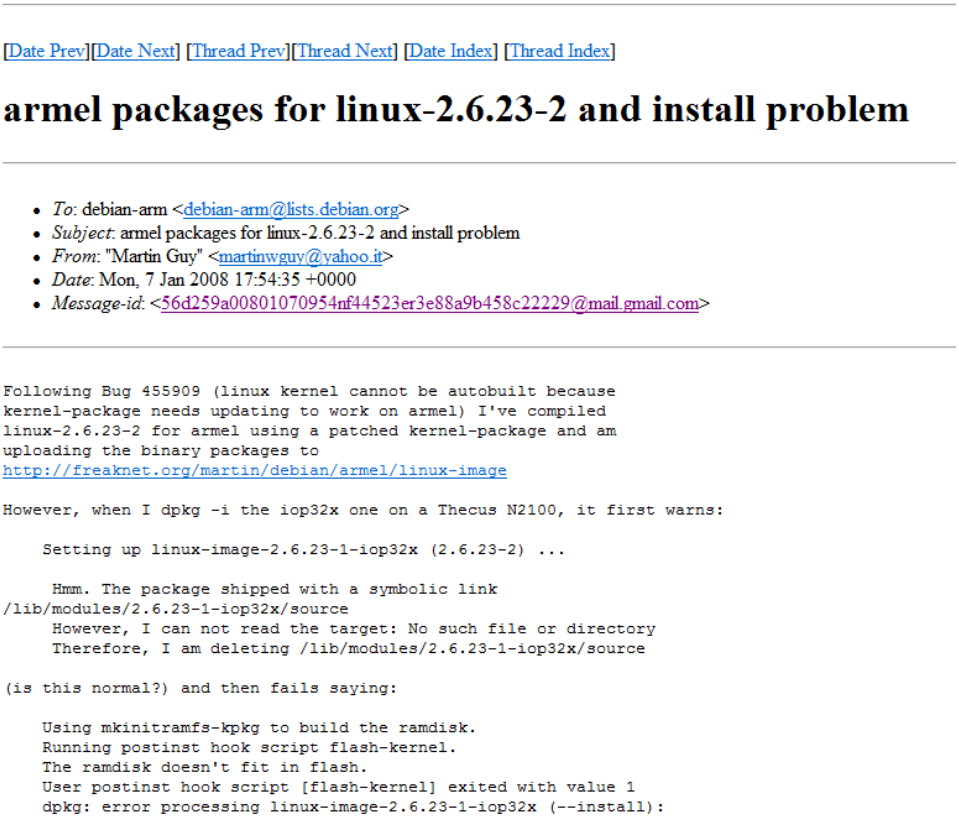


Figura 18. Diagrama de flujo de la extracción de información de participación



[Date Prev][Date Next] [Thread Prev][Thread Next] [Date Index] [Thread Index]

armel packages for linux-2.6.23-2 and install problem

- *To*: debian-arm <debian-arm@lists.debian.org>
- *Subject*: armel packages for linux-2.6.23-2 and install problem
- *From*: "Martin Guy" <martinwguy@yahoo.it>
- *Date*: Mon, 7 Jan 2008 17:54:35 +0000
- *Message-id*: <56d259a00801070954nf44523er3e88a9b458c22229@mail.gmail.com>

```
Following Bug 455909 (linux kernel cannot be autobuilt because
kernel-package needs updating to work on armel) I've compiled
linux-2.6.23-2 for armel using a patched kernel-package and am
uploading the binary packages to
http://freaknet.org/martin/debian/armel/linux-image

However, when I dpkg -i the iop32x one on a Thecus N2100, it first warns:

    Setting up linux-image-2.6.23-1-iop32x (2.6.23-2) ...

     Hmm. The package shipped with a symbolic link
/lib/modules/2.6.23-1-iop32x/source
     However, I can not read the target: No such file or directory
     Therefore, I am deleting /lib/modules/2.6.23-1-iop32x/source

(is this normal?) and then fails saying:

    Using mkinitramfs-kpkg to build the ramdisk.
    Running postinst hook script flash-kernel.
    The ramdisk doesn't fit in flash.
    User postinst hook script [flash-kernel] exited with value 1
    dpkg: error processing linux-image-2.6.23-1-iop32x (--install):
```

Figura 19. Cabecera de un mensaje de la comunidad Debian-arm (Enero 2008).

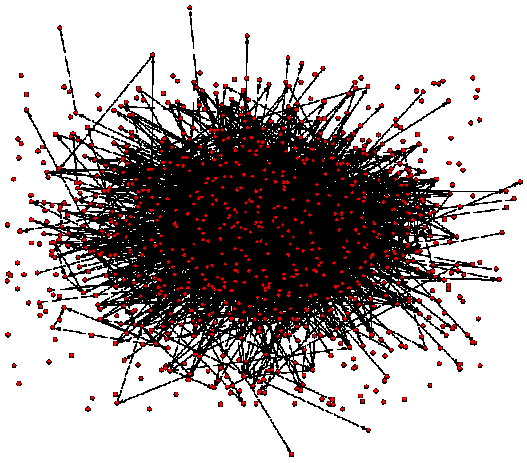
El proceso se repite para cada uno de los años considerados para cada comunidad, según se detalla en la Tabla 5.

6.2.1 Indicadores

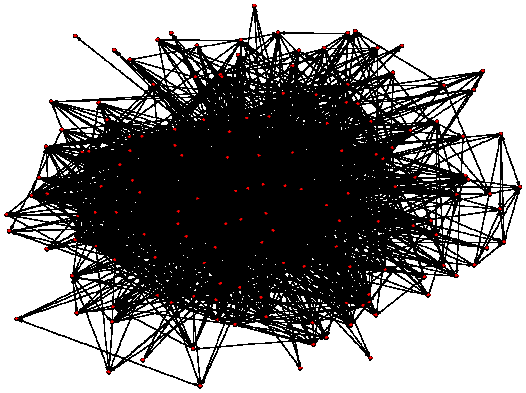
Seguidamente se detallan los indicadores utilizados para medir cada uno de los constructos entre los que se establecen las hipótesis del modelo de participación.

La cohesión puede medirse utilizando la idea de grado de salida de un nodo, esto es, el número de líneas que envía a otros nodos de la red. El grado de salida muestra los flujos de información entre los miembros de la comunidad (Toral *et al*., 2009c). Cuanto mayor sea el grado de salida de un vértice, mayor es su contribución a la comunidad. En consecuencia, el grado de salida puede utilizarse para distinguir entre las distintas tipologías de miembros que conforman la comunidad. En particular, el valor medio del

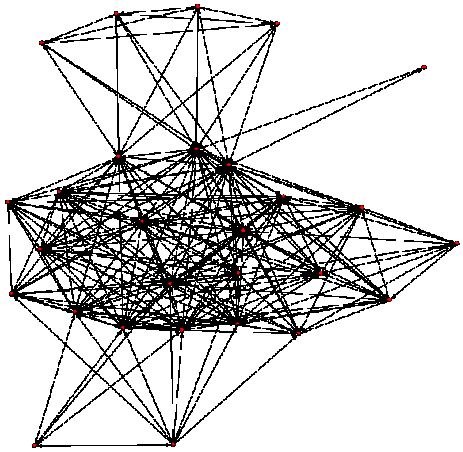
grado de salida de un nodo se ha utilizado como umbral para distinguir entre miembros activos y miembros periféricos, mientras que el valor medio más la desviación típica se ha utilizado como umbral para los miembros del núcleo de la comunidad (Toral *et al*., 2009b).



(a) Representación del total de miembros de la comunidad



(b) Representación de los miembros activos de la comunidad



(c) Representación de los miembros del núcleo comunidad

Figura 20. Comunidad Debian PPC durante el año 2005

La Figura 20 detalla la red de usuarios original [Figura 20 (a)] y las particiones extraídas de la red original mediante los valores umbrales propuestos que incluyen a los miembros activos y del núcleo de la comunidad [Figura 20 (b) y Figura 20 (c), respectivamente] para la distribución Debian PPC durante el año 2005. Los valores medios de los grados de salida de cada uno de estos grupos se utilizarán como indicadores de cohesión de la red, indicadores I1, I2 e I3 de la Tabla 6.

La estructura de la comunidad se medirá por medio de los ratios entre núcleo y miembros activos (indicador I4), miembros activos y miembros periféricos (indicador I5) y núcleo y total de miembros de la comunidad (I6).

La actividad del núcleo de la comunidad se medirá utilizado el tamaño de la comunidad (indicador I7), el número de miembros del núcleo que desarrollan una labor repetida de broker (indicador I8) y el porcentaje del grado de salida total debido al núcleo de la comunidad (indicador I9). Los indicadores utilizados muestran que no sólo es importante su participación, sino también su grado de intermediación con otros miembros de la comunidad (Sowe *et al*., 2006).

La centralidad de la red se ha medido mediante la idea de centralidad de intermediación, considerando la comunidad completa (indicador I10), los miembros activos (indicador I11)

y el núcleo de la comunidad (indicador I12). Finalmente, los indicadores que miden el éxito de la comunidad tienen que ver con su tamaño (indicadores I13 e I14) y su actividad (indicador I15), tal y como se apunta en trabajos previos (Crowston *et al*., 2003).

Tabla 6. Indicadores de medida de los constructos del modelo de participación

	Indicador	**Descripción**
I1	AvNucleoGS	Valor medio del grado de salida para los miembros que forman el núcleo de la comunidad
I2	AvActivoGS	Valor medio del grado de salida para los miembros activos de la comunidad
I3	AvTotalGS	Valor medio del grado de salida para los miembros de la comunidad
I4	Núcleo/Activos	Ratio entre núcleo y miembros activos de la comunidad
I5	Activos/Periféricos	Ratio entre miembros activos y periféricos
I6	Núcleo/Total	Ratio entre núcleo y total de miembros de la comunidad
I7	Tamaño-núcleo	Tamaño del núcleo de la comunidad
I8	Núcleo-Broker	Número de miembros del núcleo de la comunidad que desarrolla labor de broker
I9	%GS-núcleo	Porcentaje del grado de salida total debido al núcleo de la comunidad
I10	Intermed-Total	Grado de intermediación de la comunidad completa
I11	Intermed-Activos	Grado de intermediación de los miembros activos de la comunidad
I12	Intermed-Núcleo	Grado de intermediación del núcleo de la comunidad
I13	TamañoCom	Tamaño de la comunidad
I14	Activos	Número de miembros activos de la comunidad
I15	Hilos	Número de hilos de dicsusión

6.2.2 Modelo estructural y de medida

Para la correcta aplicación de PLS, el tamaño de la muestra necesario es, al menos, diez veces el número mayor de constructos antecedentes que conducen a un constructo endógeno (Barclay *et al*., 1995). Dado que el tamaño de la muestra es de 110, se cumple holgadamente este requisito.

En primer lugar, se evaluará el modelo de medida. Para ello, se va a analizar si los conceptos teóricos están medidos correctamente a través de las variables observadas, para lo cual se analiza la validez y la fiabilidad. Estas propiedades son indispensables cuando se miden actitudes, predisposiciones o respuestas emocionales, sometidas a una elevada subjetividad, por lo que las medidas realizadas no son exactamente reproducibles, ya que no se obtienen siempre los mismos resultados utilizando el mismo instrumento. Este hecho viene provocado por la presencia de errores de medición en las escalas de medidas. Estos son los errores que afectan a la fiabilidad y los sistemáticos, que afectan a la validez del instrumento de medida. En definitiva, la validez hace referencia a la bondad con que las medidas definen el concepto, mientras que la fiabilidad se relaciona con la coherencia de las medidas.

La fiabilidad mide el grado en el que las medidas están libres de errores aleatorios, es decir, proporcionan resultados consistentes. Los ítems que miden un constructo altamente fiable están fuertemente correlacionados, indicando que todos ellos miden el mismo concepto. La fiabilidad individual del ítem se valora examinando las cargas o correlaciones simples de las medidas o indicadores con sus respectivos constructos. Para que sea aceptado el indicador, éste debe tener una carga (loading) superior a 0.707 si es reflectivo (como en nuestro caso), aunque algunos autores opinan que esta regla no debe ser tan rígida en las etapas iniciales de desarrollo de escalas. Esto implica que la varianza compartida entre el constructo y sus indicadores es mayor que la varianza debida al error. Desde que las cargas son correlaciones, un nivel igual o superior a 0.707 implica que más del 50% de la varianza de las variables observadas es compartida por el constructo (Carmines *et al*., 1979). La Tabla 7 muestra que las cargas se encuentran por encima de este valor, salvo un único caso: el del ítem I4, que se encuentra justo en el límite.

Con la fiabilidad del constructo se pretende comprobar la consistencia interna de todos los indicadores al medir el concepto, es decir, se evalúa con qué rigurosidad están midiendo las variables manifiestas la misma variable latente (Roldán, 2000). Esta fiabilidad del constructo se calcula estudiando la fiabilidad compuesta del constructo (ρ_C), la cual se considera adecuada cuando adquiere el nivel de 0.7 (Nunnally, 1978). Un valor de la fiabilidad compuesta por encima del umbral de 0.7 puede apreciarse en la Tabla 7 junto a cada constructo.

Tabla 7. Fiabilidad de ítems y constructos

Constructos e indicadores	Peso	Carga	*t*-statistic
Éxito (ρc=0.932, AVE=0.821)			
Hilos	0.3675	0.9104	59.75
TamañoCom	0.3294	0.8876	22.48
Activos	0.4056	0.9199	69.57
Cohesión (ρc=0.944, AVE=0.894)			
AvNúcleoGS	0.4812	0.9355	27.21
AvActivoGS	0.5755	0.9554	50.55
AvTotalGS	0.2746	0.9167	7.96
Núcleo (ρc=0.981, AVE=0.945)			
Tamaño-núcleo	0.5177	0.9810	245.03
Núcleo-bróker	0.5022	0.9798	224.69
%GS-núcleo	0.4768	0.9552	18.84
Estructura (ρc=0.779, AVE=0.645)			
Núcleo/Activos	-0.5995	-0.6965	9.93
Activos/Periféricos	0.7241	0.8044	15.34
Núcleo/Total	-0.6036	-0.7002	9.94
Intermediación (ρc=0.903, AVE=0.824)			
Intermed-Total	1.1113	0.9975	4.05
Intermed-Activos	-0.1343	0.8084	8.46
Intermed-Núcleo	0.2153	0.8885	14.33

La validez convergente trata de asegurar que los ítems que miden un concepto miden realmente lo mismo. Por tanto, nos interesa que los ítems de un mismo constructo estén correlacionados. Se valora por medio de la varianza extraída media (AVE). Proporciona la cantidad de varianza que un constructo obtiene de sus indicadores con relación a la

cantidad de varianza debida al error de medida. Algunos autores (Fornell y Larcker, 1981) recomiendan que ésta sea superior a 0.50, con lo que se establece que más del 50% de la varianza del constructo es debida a sus indicadores. Esta medida sólo se aplica en constructos con indicadores reflectivos. En nuestro caso, la Tabla 7 muestra que se cumple la validez convergente para todos los constructos.

Finalmente, la validez discriminante indica en qué medida un constructo dado es diferente de otros constructos. Han de existir correlaciones débiles entre éste y otros constructos que midan fenómenos diferentes. Un constructo debe compartir más varianza con sus medidas o indicadores que con otros constructos en un modelo determinado. Se utiliza la varianza media compartida (AVE) entre un constructo y sus medidas, la cual debe ser mayor que la varianza compartida entre el constructo con los otros constructos del modelo (la correlación al cuadrado entre dos constructos). La Tabla 8 muestra las correlaciones entre los constructos y, en la diagonal, el valor de AVE que supera en todos los casos las correlaciones con otros constructos.

Tabla 8. Validez discriminante

	Éxito	Cohesión	Núcleo	Estructura	Intermediación
Éxito	**0.821**				
Cohesión	0.329	**0.894**			
Núcleo	0.603	0.293	**0.945**		
Estructura	0.609	0.176	0.613	**0.645**	
Intermediación	0.165	0.226	0.092	0.018	**0.824**

El modelo estructural obtenido con PLS queda recogido en la Figura 21. Modelo de participación en comunidades de software de código abierto

(Chin, 2003). Para evaluar el modelo estructural hay que estudiar dos cuestiones. En primer lugar, qué cantidad de la varianza de las variables endógenas es explicada por los constructos que las predicen, es decir, el poder predictivo del modelo, y, en segundo lugar, en qué medida las variables predictoras contribuyen a la varianza explicada de las variables endógenas.

Para responder a la primera pregunta, una medida del poder predictivo de un modelo es el valor R^2 para las variables latentes dependientes e indica la cantidad de varianza del constructo que es explicada por el modelo. Falk *et al*. (1992) establecen como valores adecuados de la varianza explicada aquellos que son iguales o mayores a 0.1; valores inferiores indican un bajo nivel predictivo de la variable latente dependiente. Esta condición se verifica tanto para el éxito como para el constructo relativo a la actividad del núcleo de la comunidad. En particular, la varianza explicada para el éxito de las comunidades supera el 50%.

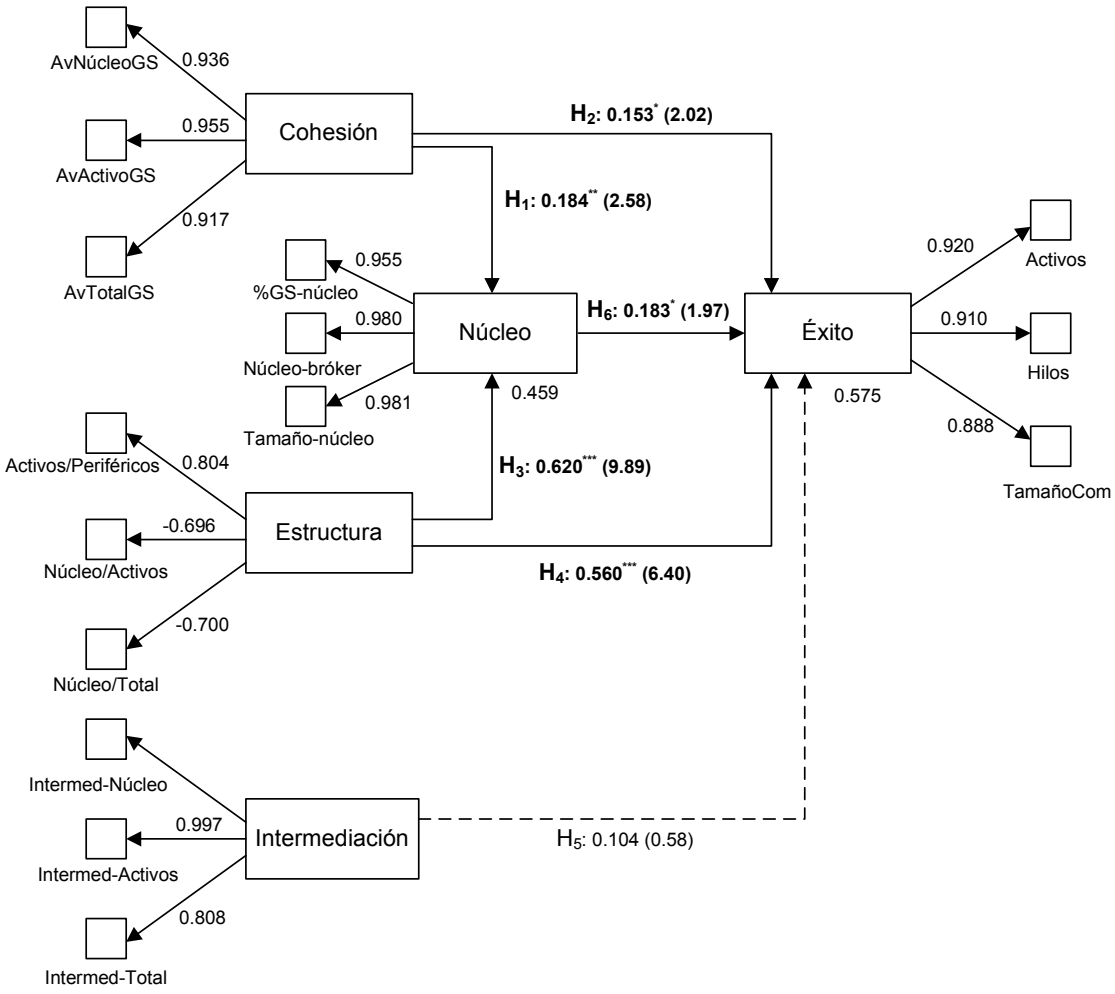


Figura 21. Modelo de participación en comunidades de software de código abierto

En cuanto a la segunda cuestión, se debe centrar la atención en los coeficientes de regresión o pesos de regresión estandarizados, así como en las correlaciones entre los

constructos o variables latentes. En el modelo de la Figura 21. Modelo de participación en comunidades de software de código abierto
todos los valores superan el umbral de 0.1 (Chin, 1998). Para poder contrastar las hipótesis planteadas debemos valorar la precisión y estabilidad de las estimaciones obtenidas, para lo cual recurrimos a la técnica Bootstrap que nos ofrece el error estándar y los valores t de los parámetros. Siguiendo a Chin (1998), para calcular la significación de los coeficientes path, se genera una prueba Bootstrap de 500 submuestras y una distribución t de Student de dos colas y con n-1 grados de libertad, donde n es el número de submuestras. Los resultados se muestran en la Figura 21. Modelo de participación en comunidades de software de código abierto

, con el valor del estadístico t entre paréntesis junto al valor de cada coeficiente de regresión. Todas las hipótesis quedan contrastadas a excepción de la hipótesis H5, relativa a la influencia del grado de intermediación. Las principales conclusiones que pueden extraerse del modelo son:

- La estructura de la comunidad es el constructo con una mayor incidencia sobre su éxito, tanto por su influencia directa como mediada a través de la actividad del núcleo de la comunidad. Esta estructura responde al principio de desigualdad participativa reconocido por muchos autores (Kuk, 2006; Sowe *et al*., 2008; Kuk, 2004). En consecuencia, es necesaria una estructura claramente definida, con una especial preponderancia del núcleo de la comunidad encargado de dirigir los esfuerzos de los miembros activos.

- La cohesión de la red integrada por los miembros de la comunidad también posee un efecto directo e indirecto apreciable sobre el éxito de la comunidad. En definitiva, la cohesión medida a partir del grado de salida de los distintos tipos de miembros es la base del mecanismo de participación, que lleva a un incremento del debate a través de los hilos de discusión. La cohesión también contempla un núcleo de la comunidad cohesionado y en la medida en que este núcleo funcione correctamente, la cohesión contribuirá de forma indirecta al éxito final del proyecto subyacente.

- El núcleo de la comunidad constituye un elemento esencial para el correcto desarrollo del proyecto subyacente, ocupando una posición central en el modelo propuesto. Tanto la cohesión como la estructura de la comunidad ven incrementado su efecto si el núcleo de la comunidad trabaja de manera eficiente.

- Finalmente, el grado de intermediación tiene un efecto más débil y no significativo sobre el éxito de la comunidad. Como posible causa de este resultado puede señalarse que la intermediación se realiza fundamentalmente por los miembros del núcleo de la comunidad, por lo que el efecto de intermediación del conjunto de la comunidad resulta poco significativo.

6.3 Análisis semántico: cosificación

El análisis del proceso de cosificación supone el análisis del contenido públicamente disponible en las comunidades de software de código abierto objeto de estudio. Para ello se utilizarán técnicas de análisis semántico basado en modelos generativos ya descritos en el capítulo dedicado a la metodología. A diferencia del proceso de participación, es muy difícil medir a posteriori el éxito de los procesos de cosificación, pues esto supondría determinar la calidad de las respuestas proporcionadas en los hilos de discusión, lo cual resulta difícilmente evaluable trabajando con un gran volumen de información. Por este motivo se ha optado por realizar un estudio exploratorio que identifique las principales dimensiones relativas a los procesos de cosificación.

La aplicación de las técnicas LDA supone realizar para cada comunidad los pasos descritos en la Figura 22.

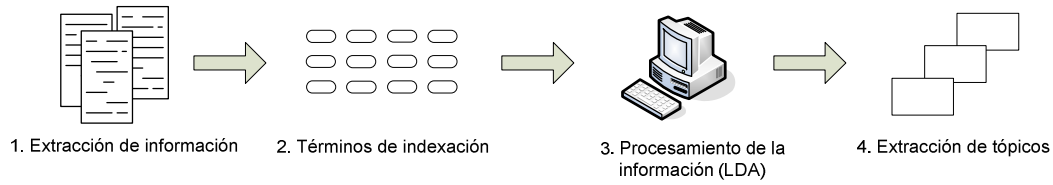


Figura 22. Diagrama de flujo de la aplicación de técnicas LDA

El primer paso consiste en la extracción de información. Para ello se ha elaborado un programa en MATLAB (Palm, 2003; Register, 2007) que accede automáticamente a los mensajes enviados por los miembros de cada comunidad, de modo que cada mensaje se trata como un documento diferente (incluyendo el tema y el cuerpo del mensaje). Cada mensaje se descarga como un documento HTML, por lo que es preciso un post-

procesamiento a nivel de cadenas de caracteres para extraer el tema y el cuerpo del mensaje, eliminar los *tags* de HTML, los signos de puntuación y transformar todo el texto a letras minúsculas. Por último, se elabora una lista de términos en función de su frecuencia de aparición, que servirá para crear la lista de términos de indexación de la Figura 22. Esta lista de términos pretende establecer el cuerpo de conocimiento dentro de cada comunidad. Para una correcta elaboración es preciso eliminar de la lista los términos de uso habitual en el lenguaje y discriminar aquellos que tienen una significación en el ámbito de los sistemas operativos en general y de Linux embebido en particular. El cuerpo de conocimiento definido por la lista resultante se utiliza como entrada al algoritmo LDA, que define los tópicos o factores latentes como último paso.

6.3.1 Aplicación a la comunidad Debian-ARM

En este apartado se ilustrará la técnica LDA, aplicándola a una comunidad concreta antes de pasar a aplicarla al conjunto de comunidades que definen el caso de estudio. La comunidad elegida es Debian-ARM, pues ARM es una familia de procesadores embebidos con un uso muy extendido en la electrónica de consumo (Toral *et al*., 2005; Barrero *et al*., 2008) y es representativa del conjunto de comunidades elegido.

La comunidad Debian ARM se ha analizado entre los años 1999 y 2007 (http://lists.debian.org/debian-arm/). A diferencia del método propuesto, una identificación manual de los tópicos requeriría la participación de expertos durante un período de tiempo bastante prolongado analizando los contenidos explicitados en la comunidad. Estos contenidos constan de miles de mensaje que, en ocasiones, alcanzan un nivel de complejidad elevado.

Tabla 9. Parámetros del algoritmo LDA

Parámetro	Descripción
D	Número de documentos (mensajes)
W	Número de términos de indexación
N	Número total de palabras
L	Longitud media de documentos (L = N/D)
T	Número de tópicos
ITER	Número de iteraciones

Tabla 10. Parámetros del algoritmo LDA para la comunidad Debian ARM

Año	D	W	N	L
1999	552	410	100339	181,77
2000	667	410	115869	173,72
2001	956	410	193313	202,21
2002	848	410	218035	257,12
2003	377	410	82904	219,90
2004	684	410	113440	165,85
2005	625	410	108523	173,64
2006	1098	410	218565	199,06
2007	1675	410	356481	212,82

El método LDA utilizado es una técnica de aprendizaje no supervisada, que evita el uso de expertos y reduce los requerimientos de tiempo para extraer los factores latentes. El hecho de que sea no supervisado también significa que puede aplicarse sin necesidad de que el analista sea experto en la materia (Newman *et al*., 2006).

El algoritmo LDA requiere la elección de una serie de parámetros detallados en la Tabla 9. Al igual que en el modelo de participación, cada año se analizará de forma individualizada y se considerará un caso diferente. La

Tabla 10 detalla los parámetros del algoritmo LDA para la comunidad Debian-ARM y cada uno de los años considerados. En total se han analizado 7482 mensajes, que es la suma de la segunda columna de la

Tabla 10. Aunque el número de documentos varía de un año a otro, la longitud media de los mensajes se mantiene bastante constante a lo largo de los años. El número de términos de indexación se ha fijado en 410 en función de la frecuencia con la que aparecen esos términos (Ng *et al*., 2001).

Hay dos parámetros importantes no detallados en la

Tabla 10. El primero es el número de iteraciones del algoritmo Gibbs sampling, que se ha elegido igual a 200. Se trata de un valor típico lo suficientemente elevado como para

garantizar la convergencia del algoritmo LDA (Blei *et al*., 2003). El segundo es el número de tópicos, que es un parámetro a pre fijar en el algoritmo LDA. Este parámetro se fijará de forma que se minimice la perplejidad, que es un parámetro frecuentemente utilizado a la hora de evaluar modelos de lenguaje (Manning y Schutze, 1999). Se basa en la probabilidad media que el modelo asigna a cada palabra dentro del cuerpo de prueba y se mide según la ecuación (51).

$$pplex = \exp\left(-\frac{1}{W}\sum_{n=1}^{W}\log P(w_n \mid d_n)\right) \tag{51}$$

La perplejidad es un valor que varía entre 1 y W, que es el número total de términos de indexación. El valor máximo W se alcanza cuando todos los términos de indexación tienen la misma probabilidad. Para obtener el número de tópicos, se ejecuta el algoritmo LDA para valores de los tópicos comprendidos entre 1 y 50, eligiéndose aquel valor que minimiza la perplejidad. La Figura 23 ilustra la variación de la perplejidad con el número de tópicos para un año concreto (2003) de la comunidad Debian ARM. Para cada valor del número de tópicos se ha ejecutado el algoritmo LDA, midiéndose el valor resultante de la perplejidad. En el caso de la comunidad Debian ARM analizada, el valor mínimo se alcanza para 19 tópicos.

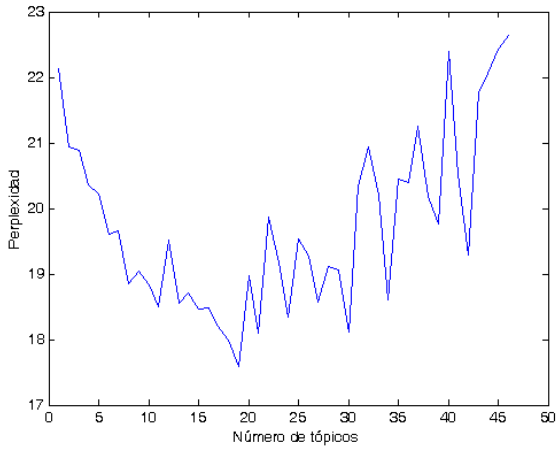


Figura 23. Variación de la perplejidad con el número de tópicos para la comunidad Debian ARM

La Tabla 11 muestra los tópicos obtenidos para cada año de la comunidad Debian ARM, incluyendo las 5 palabras más relevantes por tópico. Se pueden extraer las sigueinetes conclusiones:

Tabla 11. Distribución de tópicos por año para la comunidad Debian ARM

	Tópico1	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**			
1999	kernel	debian	problem	struct	ram	build	time	king	machine	arm	files			
	linux	image	gcc	instruction	system	start	source	config	compile	binary	server			
	gnu	ftp	built	support	program	running	patch	version	main	write	standard			
	cross	install	error	status	install	script	pass	glibc	fix	Documentation	basic			
	compiler	stuff	direct	memory	archive	trace	share	even	upload	value	debian			
2000	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**					
	kernel	install	arm	image	config	build	compile	fix	debian					
	problem	time	machine	ftp	ram	gcc	patch	main	arm					
	hosting	group	linux	linux	server	error	version	source	binary					
	start	support	compiled	system	built	software	king	gnu	embedded					
	share	arm	memory	files	off	direct	glibc	power	gdb					
2001	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**			
	kernel	build	debian	time	install	linux	compile	ram	arm	error	problem			
	king	arm	ftp	start	version	include	arm	source	files	report	machine			
	config	glibc	main	king	process	fix	patch	program	system	direct	gcc			
	support	cpu	open	write	status	stable	built	stuff	float	ports	gnu			
	image	default	basic	server	rmk	power	compiler	block	flash	struct	even			
2002	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**	**Tópico12**	**Tópico13**	**Tópico14**
	block	kernel	king	arm	ram	debian	build	system	machine	report	off	problem	linux	debian
	server	config	patch	fix	include	install	built	files	family	gnu	time	start	arm	main
	flash	source	even	debian	program	running	stable	image	support	software	write	error	gcc	ftp
	stuff	script	available	reference	device	project	target	user	king	ram	number	open	compile	source
	install	version	struct	traffic	module	process	upload	random	pass	system	memory	debug	cross	stable
2003	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**	**Tópico12**	**Tópico13**	**Tópico14**
	stuff	gcc	debian	address	arm	support	script	arm	block	build	error	time	compile	problem
	king	linux	commercial	result	ram	fpu	even	module	receive	binutils	fix	user	system	install
	return	include	tudelft	open	main	cpu	report	built	start	running	memory	files	king	version
	software	error	flash	number	build	debug	upload	patch	include	pass	function	console	compiler	toolchain
	direct	arm	mailing	family	registered	struct	problem	driver	client	executable	main	failure	install	off
	Tópico15	**Tópico16**	**Tópico17**	**Tópico18**	**Tópico19**									

	machine	image	arm	linux	kernel								
	cross	config	debian	build	available								
	source	process	Development	embedded	install								
	power	stable	project	main	ftp								
	issue	gnu	device	abi	off								
2004	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**				
	build	include	debian	install	problem	linux	usb	arm	off				
	arm	version	main	ram	built	arm	system	compile	support				
	upload	config	stable	machine	time	gcc	board	struct	king				
	available	error	kernel	patch	king	gnu	time	issue	even				
	machine	report	ftp	driver	fix	program	ram	instruction	family				
2005	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**	**Tópico12**	**Tópico13**
	gcc	install	system	debian	fix	machine	arm	time	kernel	arm	build	main	arm
	compile	debian	ram	arm	struct	off	block	king	family	support	built	source	linux
	patch	problem	board	report	software	access	stable	start	usb	user	upload	version	gnu
	error	running	embedded	network	abi	available	issue	pass	image	big-endian	archive	process	config
	cross	platform	files	cpu	result	hardware	flash	isp	script	float	accept	open	direct
2006	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**	**Tópico12**	
	arm	kernel	debian	family	system	arm	build	ram	block	king	config	kernel	
	machine	partition	install	user	process	linux	problem	flash	abi	time	device	image	
	main	loader	toolchain	address	compile	source	upload	direct	arm	start	usb	support	
	ports	off	switch	archive	even	gnu	issue	driver	struct	script	support	version	
	target	ramdisk	Developers	server	cross	gcc	fix	memory	float	available	sound	install	
2007	**Tópico1**	**Tópico2**	**Tópico3**	**Tópico4**	**Tópico5**	**Tópico6**	**Tópico7**	**Tópico8**	**Tópico9**	**Tópico10**	**Tópico11**	**Tópico12**	
	arm	error	gcc	patch	install	system	kernel	build	block	time	debian	config	
	abi	king	gnu	king	main	mount	image	source	struct	off	issue	usb	
	ports	server	arm	glibc	ftp	running	linux	compile	partition	problem	report	device	
	applied	process	linux	debian	debian	write	flash	machine	instruction	open	compiled	family	
	available	share	abi	support	problem	user	ram	cpu	download	version	archive	register	

- El número de tópicos varía de un año a otro. El valor mínimo es 9 y corresponde a los años 2000 y 2004, mientras que el valor máximo es 19 y corresponde al año 2003. El número de tópicos es una medida de la variedad de las discusiones tratadas en la comunidad. De la Tabla 11 puede observarse que, si bien existen palabras comunes en tópicos diferentes, no se repiten exactamente las mismas 5 palabras. Es decir, no se produce una repetición exacta de tópicos de un año a otro. Esto significa que cada tópico revela una nueva experiencia que se pone de relieve dentro de la comunidad.
- El hecho de que no exista repetición de tópicos pero sí palabras en común entre ellos resalta una de las ventajas del método LDA utilizado, que es su capacidad para tratar con palabras que pueden usarse en contextos diferentes. Así por ejemplo, la palabra *kernel* aparece muchas veces, pero en contextos relativos a compilación, configuración o disponibilidad.

6.3.2 Identificación de las dimensiones latentes

El algoritmo LDA se ha aplicado a continuación a las comunidades consideradas para el modelo de participación y detalladas en la Tabla 5.

Tabla 12. Indicadores extraídos del análisis semántico de las comunidades consideradas

Indicador	Descripción
I1	Número de tópicos
I2	Polisemia
I3	Polisemia ponderada
I4	Tamaño medio de los mensajes (caracteres)
I5	Número medio de mensajes por tópico
I6	Distribución de mensajes por tópico
I7	Tamaño medio de los mensajes (palabras)
I8	Número de hilos con al menos una respuesta
I9	Tamaño de los hilos (palabras)
I10	Número medio de hilos por tópico
I11	Distribución de hilos por tópico

Igual que se ha hecho anteriormente para la comunidad Debian ARM, cada comunidad de la Tabla 5 se analizará semánticamente de forma anual, lo que da un total de 110 casos.

La Tabla 12 resume la lista de indicadores extraídos de la aplicación del algoritmo LDA a cada uno de los años capturados para las comunidades de la Tabla 5. El primer indicador se refiere al número de tópicos elegido según el criterio de minimizar la perplejidad. Los indicadores I2 e I3 se refieren a la polisemia, o palabras con varios significados. En particular, I2 mide la polisemia como el número de veces que una palabra w_i aparece más de una vez en tópicos diferentes, mientras que I3 considera este valor pero ponderado por la probabilidad $P(w_i|z_i=j)$ de la palabra w_i bajo el tópico j_{th}. Los siguientes cuatro indicadores se refieren a los mensajes compartidos en la comunidad. En particular, miden el tamaño medio de los mensajes en caracteres y palabras, el número medio de mensajes incluidos en cada tópico y su distribución sobre los tópicos. Finalmente, los últimos tres indicadores se refieren a los hilos de discusión, y miden los hilos con al menos una respuesta y su tamaño en palabras, y el número medio y la distribución de hilos por tópico.

Para obtener las principales dimensiones relacionadas con los procesos de cosificación se ha llevado a cabo un análisis factor exploratorio, utilizando los indicadores descritos. La Tabla 13 muestra la medida de adecuación de Kaiser-Meyer—Olkin (KMO) y la prueba de esfericidad de Bartlett. El valor próximo a 0.75 de la primera y el elevado valor de la Chi-cuadrado confirman que la aplicación de esta técnica estadística es adecuada y que las variables poseen el suficiente grado de correlación como para que la extracción de factores comunes sea aceptable.

Tabla 13. KMO y prueba de esfericidad de Bartlett

Medida de adecuación muestral de Kaiser-Meyer-Olkin		,751
Prueba de esfericidad de Bartlett	Chi-Cuadrado Aprox.	1837,709
	gl	55
	Sig.	,000

Los auto-valores de la matriz de covarianza de la muestra se detallan en la Tabla 14.

Tabla 14. Varianza total explicada

Factor	Autovalores		
	Total	% de Varianza	% de Varianza acumulada
1	4,852	48,524	48,524
2	2,821	28,214	76,738
3	1,861	18,610	95,347
4	,157	1,569	96,916
5	,099	,985	97,901
6	,081	,812	98,713
7	,051	,513	99,227
8	,036	,364	99,591
9	,024	,236	99,828
10	,017	,172	100,000

En ella se detallan los autovalores obtenidos en orden descendente, la varianza explicada por cada uno de ellos y la varianza acumulada. De acuerdo a los criterios habituales de elección del número de factores, se han elegido un total de tres factores. La varianza explicada llega a superar el 90% de la varianza total.

Tabla 15. Cargas de los factores rotadas por el método Varimax

	Componente		
	1	2	3
I1	-,093	,959	-,070
I2	,209	,951	,066
I3	-,069	,982	,033
I4	-,117	-,016	,973
I5	,984	-,052	-,052
I6	,945	,066	-,011
I7	-,065	,036	,980
I8	,976	,120	-,078
I9	-,006	,103	,912
I10	,977	-,081	-,069
I11	,962	-,008	-,124

Para encontrar un significado de los factores comunes es necesario el cálculo de las cargas de los factores. Utilizando el método de extracción de componentes principales,

las cargas de los factores se estiman a partir de los autovectores asociados. Normalmente, suele resultar complicado realizar una correcta interpretación de los factores a partir de las cargas así extraídas. Afortunadamente, las cargas de los factores se pueden rotar multiplicándolos por una matriz ortogonal. Las cargas rotadas preservan las propiedades de las originales y además facilitan la interpretación de los factores.

El método Varimax es un método de rotación ortogonal que minimiza el número de variables con una carga elevada en cada factor. Este método simplifica la interpretación de los factores. La Tabla 15 muestra las cargas de los factores rotadas mediante el método Varimax. Para extraer el significado de cada factor, nos movemos horizontalmente de izquierda a derecha observando las cargas de cada una de las doce variables e identificando para qué factor alcanza su valor máximo. Las cargas de los factores se pueden considerar significativas a partir del umbral de 0.7 (Rencher, 2002). El resultado es una serie de descriptores asociados a cada factor que conducen a los siguientes factores comunes latentes:

- El primer factor se refiere a la actividad en torno a un tópico, como muestran las elevadas cargas de los indicadores I5, I8 e I10. En cualquier caso, los elevados valores de las desviaciones típicas en las distribuciones de mensajes e hilos por tópicos sugieren que no todos ellos se tratan con la misma profundidad. Así pues, las comunidades tienden a la especialización en determinados temas que despiertan más interés.
- El segundo factor se refiere a la capacidad de las comunidades de código abierto para crear y reutilizar conocimiento. El número de tópicos y la polisemia constituyen una medida del conocimiento creado y reutilizado. El hecho de que los tópicos sufran ligeras variaciones en su descripción significa que van evolucionando y que el conocimiento previo se mezcla y combina para generar nuevo conocimiento.
- El tercer factor se refiere a la cantidad de información suministrada. Los indicadores I4, I7 e I9 están relacionados con los tamaños medios de mensajes e hilos. La disponibilidad y la profundidad de tratamiento de los temas que surgen en el seno de la comunidad constituyen un factor determinante para el desarrollo y evolución de la comunidad.

Los dos primeros factores se encuentran relacionados con las categorías del conocimiento compartido descritas en la literatura. La primera consiste en el conocimiento y la experiencia personal revelada al resto de los miembros de la comunidad. Una vez que el conocimiento es revelado, puede ser reutilizado en otros campos o replicado en contextos diferentes (Wai, 2008), evitando la duplicación de esfuerzos. Finalmente, el conocimiento se puede recombinar para generar nuevo conocimiento (Kuk, 2006). Otros estudios también concluyen una relación positiva entre las actividades de compartir conocimiento y el desarrollo de la comunidad (Koh y Kim, 2004). Sin embargo, este estudio se basa fundamentalmente en indicadores de participación en lugar de analizar el contenido de los mensajes puestos a disposición de la comunidad.

Las observaciones se pueden asignar a cada uno de los factores extraídos usando las cargas factoriales. Las cargas factoriales son estimaciones de los factores latentes subyacentes y poseen un valor para cada una de las observaciones que constituyen la muestra. La asignación se ha efectuado atendiendo al factor en que se alcanza un valor máximo, siempre y cuando este valor resulte superior a 0.1 (Rencher, 2002). Siguiendo este criterio, las observaciones pueden asignarse a cada uno de los factores extraídos, lo que supone una categorización de la muestra original. Para validar el análisis factorial, la hipótesis de igualdad de medias en las tres categorías en que puede dividirse la población debería ser rechazada. Para ello se ha aplicado el test de Kruscal-Wallis, que es el equivalente no paramétrico del análisis ANOVA y que no requiere la hipótesis de normalidad en la muestra. Los resultados se detallan en la Tabla 16. La hipótesis nula se rechaza en todos los casos, lo que significa que las categorías definidas por los factores son claramente diferenciables.

Tabla 16. Test de Kruskal-Wallis

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
Chi-2	45,882	26,142	46,932	38,972	50,312	39,332	33,532	50,132	29,202	51,352	52,402
gl	2	2	2	2	2	2	2	2	2	2	2
Asymp. Sig.	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000

Capítulo 7. Conclusiones, limitaciones y futuros trabajos

7.1 Conclusiones

Hay dos aspectos centrales en la creación del conocimiento que tienen que ver con cómo maximizar la habilidad de la gente para crear nuevo conocimiento y cómo construir un ambiente que facilite compartir el conocimiento generado (Sveiby, 2000). En su propuesta sobre la organización creadora de conocimiento, Nonaka y Takeuchi (1995) dividen la creación de conocimiento en dos dimensiones: la ontológica en donde señalan que "el conocimiento es creado sólo por los individuos. Una compañía no puede crear conocimiento sin individuos", y la epistemológica, donde establecen las diferencias entre conocimiento tácito y explícito. El tácito es personal y de contexto específico, difícil de formalizar y de comunicar. El explícito o codificado es aquél que puede transmitirse utilizando el lenguaje formal y sistemático. Nonaka y Takeuchi consideran que el conocimiento se crea y desarrolla a través de la interacción social del conocimiento tácito y conocimiento explícito, y definen cuatro formas de conversión de conocimiento: socialización, que es la conversión de conocimiento tácito en conocimiento tácito compartiendo experiencias; exteriorización, donde el conocimiento tácito se vuelve explícito; combinación, que es la conversión de conocimiento explícito en conocimiento explícito; e interiorización, que implica la conversión de conocimiento explícito en tácito.

Frente a este esquema, las comunidades de práctica introducen un cambio de paradigma que se produce al concentrarse el conocimiento en un grupo y no en una persona, permitiendo eliminar los egos que producen un bloqueo en el aprendizaje. La información y el conocimiento pertenecen a la comunidad y los individuos son reconocidos por su participación y liderazgo. Las comunidades de prácticas parten de una concepción social del aprendizaje, que dependerá de las interacciones de las personas y de la construcción conjunta de significados. En este sentido las comunidades de práctica se ajustan mejor a la dualidad definida por Hildreth *et al*. (1999) sobre conocimiento *soft* y conocimiento *hard*. Esta concepción parte de la base de que el aprendizaje es una actividad situada en un contexto que la dota de inteligibilidad, según la cual la descontextualización del aprendizaje es imposible, puesto que toda adquisición de conocimiento está contextualizada en algún tipo de actividad social (Wenger, 1998). Este cambio en la unidad de análisis, desde el contexto de los

individuos al contexto de la comunidad, conduce a un cambio en el que se entiende el aprendizaje como "el desarrollo de una identidad como miembro de una comunidad y llegar a tener habilidades de conocimiento como parte del mismo proceso" (Lave y Wenger, 1991).

El proceso subyacente en la generación de conocimiento *soft* es la participación, puesto que el aprendizaje supone la participación en una comunidad y se reconoce como un proceso de participación social. En general, la participación favorece la transferencia informal de conocimiento dentro de redes y grupos sociales. No obstante, el proceso que tiene lugar dentro de las comunidades de práctica es el de participación periférica legítima, por medio del cual un miembro nuevo puede irse moviendo desde la periferia hasta el centro de la comunidad, incrementando su aprendizaje basado en la experiencia y siendo más activo y comprometido con el resto de la comunidad.

El proceso subyacente en la generación del conocimiento *hard* es la cosificación, consistente en explicitar y almacenar la experiencia compartida, dejándola accesible al resto de miembros de la comunidad.

A diferencia de la concepción de Nonaka y Takeuchi, ambos tipos de conocimiento, *soft* y *hard*, constituyen una dualidad que se complementan y realimentan el uno al otro. Es decir, los procesos de participación requieren de la explicitación del conocimiento, y se explicita aquello sobre lo que se discute, de cómo que es la propia comunidad la que genera nuevo conocimiento a través de sus debates y discusiones como un proceso de invención colectiva.

El auge actual de las tecnologías de información y comunicación ha hecho emerger nuevas formas de comunidades de práctica que hacen un uso intensivo de los medios electrónicos, que son las llamadas comunidades virtuales. Las comunidades virtuales, si bien sustituyen en gran medida el contacto cara a cara por un contacto virtual, amplifican muchas de las ventajas de las comunidades tradicionales, aprovechando las enormes economías de escala que proporciona Internet para acceder a un numerosísimo número de potenciales miembros y expertos, superando barreras físicas y los límites tradicionales de las organizaciones.

En este sentido, los proyectos de software de código abierto constituyen un ejemplo claro de un nuevo paradigma de elaboración de software basado en desarrollos en comunidad. A diferencia del software propietario, basados en esquemas tradicionales de desarrollo software, los proyectos de software de código abierto se basan en una

comunidad de usuarios subyacente que realiza aportaciones y contribuye al desarrollo y mejora continua del proyecto. Este tipo de comunidades hace un uso de intenso de Internet y los medios electrónicos, y es a través de ellos como se organizan y se ponen en contacto cientos e incluso miles de usuarios unidos a cada comunidad. El conocimiento nace y se comparte en el seno de la comunidad, de modo que constituye un bien que no pertenece a ningún individuo, sino a la comunidad en su conjunto. El conocimiento *soft* se desarrolla mediante la participación en forma de mensajes a listas de correo o a sistemas de seguimiento de errores o versiones. El conocimiento *hard* se desarrolla a través de las herramientas electrónicas que suelen formar parte de los proyectos de software de código abierto, como los sistemas de versiones o CVS (Concurrent Version Sytem), los rastreadores (trackers) o las listas de correo.

El elemento más importante y el bien más preciado de un proyecto de software de código abierto es su comunidad de soporte. A pesar de que idealmente las comunidades pretenden conseguir una participación completa y activa de todos sus miembros, lo cierto es que en todas ellas se encuentra presente la llamada desigualdad participativa, de modo que la mayoría de las aportaciones las realiza un reducido número de miembros de la comunidad. Esto se debe a que las comunidades presentan una estructura por capas, de modo que en el centro se encuentran los moderadores y usuarios más expertos, y a medida que nos alejamos de este núcleo más activo, la actividad va decreciendo hasta llegar a los usuarios pasivos que se encuentran en la periferia. Estos usuarios se caracterizan porque hacen uso del resultado software y de las aportaciones y mejoras realizadas en el seno de la comunidad, pero no realizan contribuciones para mejorarlo. En cualquier caso, y dada la naturaleza abierta de estos proyectos, los usuarios pasivos son tolerados dentro de la comunidad.

En este trabajo se han analizado los mecanismos de participación y cosificación que intervienen en las comunidades de soporte de los proyectos de software de código abierto.

Para el análisis de los mecanismos de participación se han utilizado técnicas de análisis de redes sociales, modelando la comunidad como un grafo cuyos vértices son los miembros de la comunidad y donde las flechas representan las interacciones entre esos miembros. A partir de estos grafos se ha medido una serie de indicadores de la red con los que se ha elaborado un modelo estructural y de medida del éxito de las comunidades

de software de código abierto. Este modelo define una serie de antecedentes causales que determinan cómo deben gestionarse los procesos de participación:

- Cohesión. La cohesión es una medida de la densidad de conexiones en la red social. Es importante que cuando los usuarios de la comunidad acuden a los foros encuentren respuesta. De otro modo no lograrán resolver el problema que tienen planteado y acudirán a otros foros o incluso puede que cambien de software. Frente a los foros que a veces plantean los proyectos de software propietario, los proyectos de software de código abierto se caracterizan por una mayor participación y una solución más exacta al problema planteado. Si la comunidad es lo suficientemente extensa, es muy posible que la duda o problema planteado haya sido resuelta con anterioridad, y algún miembro informe al respecto. Redes no cohesionadas provoca que el número de miembros activos y periféricos vaya descendiendo, impidiendo que a través de los procesos de participación mejoren su aprendizaje. Nuestro modelo valida la hipótesis de la influencia de la cohesión sobre el éxito final de la comunidad.

- Estructura. A pesar de que las interacciones en el seno de la comunidad poseen un carácter informal, éstas requieren de una estructura bien definida para que la actividad se mantenga y continúe, y para guiar futuros desarrollos. Las comunidades poseen una estructura por capas, pero sus integrantes no son fijos: es posible una movilidad de los miembros de la comunidad entre las distintas capas a medidas que aumentan su experiencia y su aprendizaje. Los procesos de participación periférica legítima permiten que los miembros, si lo desean, puedan moverse de la periferia hacia el centro de la comunidad. Las estructuras por capas, atendiendo al principio de desigualdad participativa, muestran una influencia positiva sobre el éxito de la comunidad, tal y como se deduce del modelo planteado. Esta estructura resuelve los problemas de coordinación derivados de manejar un elevado número de personas vinculadas al proyecto software.

- Núcleo. El núcleo de la comunidad es la capa más central y reducida de la comunidad, pero es la que tiene la misión más importante, ocupando un lugar de especial relevancia en el modelo de participación propuesto. El núcleo de la comunidad no sólo es responsable de la mayor parte de las contribuciones, sino que además debe mediar entre el resto de miembros para garantizar que se

mantiene la actividad y que los desarrollos y el debate van en la dirección adecuada. Su actividad no sólo tiene una incidencia positiva sobre el éxito de la comunidad, sino que también potencia la influencia de la cohesión y la estructura de la comunidad.

Para los procesos de cosificación se han utilizado técnicas de análisis semántico, que permiten analizar el contenido de la información explicitada en la comunidad. En particular, el algoritmo LDA utilizado es un modelo generativo que se basa en unas variables latentes o tópicos en base a los cuales se construyen los documentos de información. Como resultado se han obtenido las principales dimensiones que intervienen en los procesos de cosificación mediante un estudio exploratorio. Estos resultados identifican varias dimensiones típicas en lo que respecta a la explicitación de contenidos. El primero de los factores identificados muestra que las comunidades tienen una tendencia a la especialización en determinados temas; el segundo, su capacidad para crear y reutilizar conocimiento; y el tercero se refiere a la cantidad de información suministrada en cada tema. En general, lo importante es mantener un equilibrio entre estas dimensiones detectadas. Es importante crear y reutilizar conocimiento, pero una excesiva abundancia de temas puede conducir a un tratamiento leve de determinados temas, o que la información suministrada no sea lo profunda que los usuarios requieren.

El análisis conjunto de los procesos de participación y cosificación nos lleva a las siguientes consideraciones:

- Las comunidades de práctica virtuales son fenómenos sociales y como tales las relaciones entre sus miembros juega un papel importante. No obstante, y dado que existe un tema subyacente (en nuestro caso, el proyecto software), es importante también considerar el contenido y la calidad de la información suministrada.
- Las comunidades requieren de una estructura, sin alcanzar los organigramas y las estructuras más formales y estáticas de las organizaciones tradicionales. Esta estructura viene determinada por el grado de participación y nivel de experiencia de los miembros de la comunidad que, según su implicación, adquieren diversos roles.

- Los proyectos de software de código abierto no alcanzan la visión utópica de una participación en masa y global, sino que se rigen por el principio de desigualdad participativa. Lo importante es que el proceso de participación periférica legítima permita que usuarios con poca experiencia puedan alcanzar el núcleo de la comunidad.
- Sobre los miembros del núcleo de la comunidad recae la labor más importante, poniendo a disposición de la comunidad sus conocimientos, facilitando que nuevos usuarios continúen con la labor de desarrollo y participación, y animando la participación y guiando los futuros desarrollos software.

Se puede afirmar que este esquema de desarrollo por medio de comunidades de soporte ha trasformado y está revolucionando la industria del software, y tiene implicaciones importantes sobre el futuro. En particular, se puede observar lo siguiente:

- Se está acelerando el proceso de "*commoditización*" de ciertos elementos de la industria del software, con la aparición de soluciones de bajo coste en la base de la plataforma en las que la diferenciación que pueden ofrecer las soluciones propietarias es muy reducida.
- Es creciente el uso de esos elementos básicos en las propias soluciones propietarias, lo que supone, de hecho, introducir paradigmas relacionados con el software de código abierto en entornos propietarios. Las propias empresas de software propietario están incluyendo fragmentos de software desarrollado en código abierto en sus soluciones finales y patrocinando proyectos de software de código abierto. En estos proyectos, y también en proyectos de software propietario, los principales actores del software mundial están adoptando asimismo el modelo de la comunidad para probar con clientes y con comunidades sus nuevos productos antes del lanzamiento, para mejor ajustar los mismos a las necesidades de los clientes y reducir el tiempo de lanzamiento al mercado de sus productos.
- La penetración del software de código abierto está obligando a los proveedores tradicionales a reducir sus precios en productos o soluciones básicas, o a evolucionar sus modelos de licenciamiento, para poder competir con éxito y de manera sostenida con las opciones de proveedores de software de código abierto.

De entre todas las consideraciones anteriores es el proceso de "commoditización" el fenómeno que tendrá un mayor impacto en el desarrollo del software de código abierto (O'Reilly, 2005). Este proceso ha tenido lugar de la siguiente forma: inicialmente, los productos software ofrecidos se encontraban por debajo de las expectativas de los consumidores, de modo que los vendedores incrementaron las prestaciones exigidas a través de soluciones propietarias. A medida que los productos van mejorando, llega un momento en que empiezan a situarse por encima de las expectativas de la mayoría de los clientes. Es entonces cuando los clientes comienzan a valorar otras cualidades como el tiempo de salida al mercado, o el lanzamiento de nuevas versiones, o el coste. Como consecuencia, la arquitectura de los productos comienza a modularizarse y estandarizarse para satisfacer estas nuevas demandas. En ese momento, el producto en cuestión se convierte en una "commodity", y cada vez resulta más difícil a los compañías diferenciar sus productos de sus competidores (Ven y Mannaert, 2008).

7.2 Limitaciones

El proceso de investigación no es totalmente objetivo, no está guiado por un procedimiento preciso que marque cada uno de los pasos a seguir para no desviarse del camino correcto. Más bien es un proceso complejo, sistémico y con constantes pasos adelante y atrás. En su desarrollo es fundamental la presencia del investigador y las decisiones que se van tomando a lo largo del proceso de estudio. La adecuada interpretación de los resultados requiere que se hagan explícitas las limitaciones provenientes de dichas elecciones, de manera que se puede evaluar convenientemente el trabajo que se ha realizado y las formas alternativas que se podían haber utilizado. En el actual epígrafe se presentan las limitaciones que se han encontrado en la investigación desarrollada.

La primera limitación es la naturaleza exploratoria del estudio, tanto del análisis PLS del modelo de participación como del análisis factorial relativo a los procesos de cosificación. Esta naturaleza exploratoria se justifica por la ausencia de estudios previos sobre los procesos subyacentes en la gestión de conocimiento *soft* y *hard* en proyectos de software de código abierto. No obstante, sería recomendable la realización de estudios posteriores confirmatorios.

Una segunda limitación se encuentra en el número de indicadores utilizados. Tanto del análisis de redes sociales como del análisis semántico podría derivarse un mayor número de indicadores que permitiría contrastar el modelo obtenido e incluso ampliarlo. El principal problema de disponer de un elevado número de indicadores es la dificultad de discriminar los más relevantes para evitar problemas de sobre-ajuste en los modelos. No obstante, en las futuras líneas se proponen algunas soluciones al respecto.

Como tercera limitación, señalar que el caso de estudio propuesto se centra en las distribuciones Linux para sistemas embebidos. Si bien esta elección nos lleva a comunidades más pequeñas y profesionalizadas, el estudio podría extenderse a otros proyectos de software de código abierto no relacionado con el ámbito de los sistemas operativos, donde también existen comunidades altamente especializadas. Asimismo, se han analizado fundamentalmente el ámbito de los foros de discusión y las listas de distribución, aunque existen otras herramientas habitualmente empleadas en los proyectos de software de código abierto como los sistemas de seguimiento de versiones o de seguimiento de errores.

Es necesario resaltar también limitaciones referidas al método utilizado para la validación del modelo de medida y estructural. El objetivo de la modelización PLS es la predicción de las variables dependientes, tanto latentes como manifiestas, lo cual se traduce en un intento por maximizar la varianza explicada (R^2) de las variables dependientes, lo que lleva a que las estimaciones de los parámetros estén basadas en la capacidad de minimizar las varianzas residuales de las variables endógenas. PLS se adapta mejor a aplicaciones predictivas y el desarrollo de la teoría (análisis exploratorio), aunque también puede ser usado para la confirmación de la teoría (análisis confirmatorio). El análisis factorial también posee limitaciones en el sentido de que la recurrencia de la estructura factorial puede ser atribuible a la similitud de las variables empleadas, más que a una estructura de verdad subyacente (Block, 1995)

Otra limitación se relaciona con la noción de causalidad. Aunque se proporcionan evidencias sobre la causalidad del modelo, ésta en sí misma no ha sido probada. De hecho, dada la modelización empleada (modelización flexible), hemos abandonado conscientemente la idea de causalidad, apoyándonos en el concepto de predictiblidad. Como apuntan Falk *et al*. (1992), "mientras que la causalidad garantiza la capacidad de controlar los acontecimientos, la predictibilidad permite sólo un limitado grado de control". De hecho, al comentar las relaciones existentes entre constructos, Fornell

(1982) sostiene que las denominadas relaciones causales entre variables no pueden ser comprobadas, sino que son siempre asumidas por el investigador. En este sentido, hemos de reconocer que pueden existir distintos modelos alternativos.

7.3 Futuras líneas de investigación

La primera de las futuras líneas consistiría en elaborar un modelo global que incluya los procesos de participación y cosificación. La principal dificultad estriba en medir el éxito del proceso de cosificación, pues supondría evaluar la calidad de las respuestas planteadas en los foros y listas de distribución. Por ejemplo, se podría medir la satisfacción de los usuarios ante las respuestas recibidas, o analizar mediante expertos hasta qué punto las respuestas crean nuevo conocimiento y responden a las expectativas de los miembros de la comunidad.

Respecto a la metodología de análisis de redes sociales, se podría ampliar el número de características medibles sobre los grafos que modelan las comunidades virtuales. Entre ellos cabe mencionar:

- Densidad: la medida de densidad se ha realizado atendiendo fundamentalmente al grado (de salida) de los nodos. No obstante, la densidad puede ser también medida usando un punto de vista egocéntrico. La densidad egocéntrica de un nodo es la densidad de sus conexiones entre sus vecinos (Nooy *et al*. 2005).
- Componentes: Un componente es una subred fuertemente conectada de tamaño máximo. Los componentes permitirían la identificación de subestructuras dentro de la comunidad.
- Núcleos (*k-cor*es): un *k-core* es una subred en la que cada nodo tiene k grados dentro de esa subred. También permitirían la detección de subredes dentro de la comunidad.
- Distancia: se define como el número de pasos en el camino más corto entre dos nodos de la red. Normalmente suele utilizarse cuando existe algún nodo de referencia en la red respecto al cual medir distancias. En el caso de las comunidades habría que señalar un conjunto de nodos de referencia que serían los miembros del núcleo de la comunidad.

- Correlación entre particiones. Tanto el grado de salida como la distancia generan particiones sobre la red social objeto de análisis. El grado de correlación puede medirse haciendo uso de dos tipos de índices de asociación referenciados en la literatura: la V de Cramer y el índice de información de Rajski (Nooy *et al*., 2005). La V de Cramer mide la dependencia estadística entre dos clasificaciones. El índice de Rajski mide el grado por el cual la información de una clasificación se preserva en la otra clasificación.

En el ámbito de las técnicas de análisis semántico se han usado como términos de indexación unigramas, pero podría ampliarse considerando también bigramas o trigramas.

Si bien el hecho de añadir un mayor número de indicadores aumenta la riqueza de información disponible, presenta un inconveniente a la hora de elaborar un modelo, que es la dificultad de discriminar los indicadores más útiles evitando situaciones de sobre-ajuste. En este sentido se propone la utilización de un sistema experto basado en computación evolutiva como método de discriminar variables. En particular, los algoritmos genéticos se han aplicado exitosamente a este fin (Martínez Torres y Toral, 2010; Martínez Torres *et al*. 2010), permitiendo realizar una búsqueda guiada sobre el conjunto de selecciones de indicadores posibles, que nos llevaría a un número prohibitivo de posibilidades.

Finalmente, respecto al estudio empírico se pueden apuntar las siguientes líneas:

- Extensión del estudio a otras comunidades virtuales. En primer lugar a la comunidades Linux para PC, para establecer las diferencias entre comunidades de acceso masivo frente a comunidades más especialistas y profesionalizadas. En segundo lugar a otros proyectos de software de código abierto en otros ámbitos distintos de los sistemas operativos.

- Ampliar el objeto de estudio a sistemas de seguimiento de versiones y sistemas de seguimiento de errores. En este caso, aplicando únicamente el análisis de redes sociales.

- Analizar en detalle los miembros del núcleo de la comunidad y su evolución en el tiempo. Midiendo las distancias respecto a los nodos que forman parte del núcleo de la comunidad se puede determinar el grado de influencia de cada uno. El estudio de la evolución temporal permitiría ver en detalle el proceso de

participación periférica legítima por el que miembros periféricos alcanzan el núcleo de la comunidad.

Bibliografía

- Abbott, D. (2003): *Linux for Embedded and Real Time Applications*. Newnes, Elsevier Science, USA.Abdolmohammadi, M. J. y Greenlay, L. (2001): "Accounting Methods for Measuring Intellectual Capital", Round Table Group. Linking Leaders with Scholars
- Abedin, B., Sohrabi, B. (2009): "Graph theory application and web page ranking for website link structure improvement", *Behaviour y Information Technology*, Vol. 28, no. 1, pp. 63 – 72.
- Ahmad, K. (1996): "A Terminology Dynamic and the Growth of Knowledge: A Case Study in Nuclear Physics and in the Philosophy of Science", en TKE'96. *Proceedings of the 4th Conference on Terminology and Knowledge Engineering. Viena*, pp. 26-28.
- Ahn, T., Ryu, S. and Han, I. (2004): "The impact of the online and offline features on the user acceptance of Internet shopping malls", *Electronic Commerce Research and Applications*, Vol. 3, pp. 405–420.
- Alavi, M. and Leidner, D. E. (2001): "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues", *Management Information Systems Quarterly*, Vol. 25, no. 1, pp. 107–136.
- Amabile, T.M., (1996): *Creativity in Context: Update to the Social Psychology of Creativity*. Westview Press, Boulder, CO.
- Appleyard, M. M. (1996): "How Does Knowledge Flow? Interfirm Patterns in the Semiconductor Industry", *Strategic Management Journal*, Vol. 17, Special Issue (Winter), pp. 137-154
- Ardichvili, A., Page, V., and Wentling, T. (2003): "Motivation and barriers to participation in virtual knowledge-sharing communities of practice", *Journal of Knowledge Management*, Vol. 7, no. 1, pp. 64–77.
- Argyris, C. y Schon, D. (1978): *Organizational Learning*. Addison-Wesley, Reading, M.A.
- Argyris, C. (1993): *Knowledge for Action: A Guide to Overcoming Barriers to Organizational Change*. Josse y-Bass, San Francisco, CA.

- Arrow, K. J. (1971): *Essays in the Theory of Risk Bearing*. Markham, Chicago, IL
- Arrow, K. J. (1984): "Information and economic behavior", in *Collected Papers of Kenneth J. Arrow*, Vol. 4. Belknap Press, Cambridge, MA.
- Bagozzi, R. P. (1994): "Structural Equation Models in Marketing Research: basic Principles", en R. Bagozzi (Ed.): *Principles of Marketing Research*, pp. 317-385, Oxford Blackwell.
- Barabba, V. y Zaltman, G. (1990): *Hearing the Voice of the Market*, Harvard Business School Press
- Barclay, D.; Higgings, C. y Thompson, R. (1995): "The Partial Least Squares (PLS) Approach to Casual Modeling: Personal Computer Adoption and Use as an Illustration", *Technology Studies*, Vol. 2, nº 2, pp. 285-309
- Barclay, R. O. y Murray, P. C. (1997): "What Is Knowledge Management?", *Knowledge Praxis*, http://www.media-access.com/whatis.html
- Barrero, F., Toral, S. L., Gallardo, S. (2008): "EDSPLAB: Remote Laboratory for Experiments on DSP Applications", *Internet Research*, Vol. 18, Iss. 1, pp. 79-92.
- Bessen, J., (2005): *Open source software: Free provision of complex public goods*. http://www.researchoninnovation.org/opensrc. pdf. Acceso Febreo 2007.
- Bharati, P. y Chaudhury, A. (2004): "An empirical investigation of decision-making satisfaction in web-based decision support systems", *Decision Support Systems*, Vol. 37, pp. 187 – 197.
- Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A. (2006): "Mining Email Social Networks", *Proceedings of the International Workshop on Mining Software Repositories, MSR'06*, Shanghai, China, pp. 137-143.
- Birkinshaw, J. M. (1995): "Entrepreneurship in multinational corporations: The initiative process in Canadian subsidiaries", tesis doctoral, Western Business School.

- Bitzer, J., Schrettl, W., Schröder, P. (2007): "Intrinsic motivation in open source software development", *Journal of Comparative Economics*, Vol. 35, no. 1, pp. 160-169.
- Blei, D. M., Ng, A. Y., y Jordan, M. I. (2003): "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, Vol. 3, pp. 993-1022.
- Block, J. (1995): "A contrarian view of the five-factor approach to personality description", *Psychological Bulletin*, Vol. 117, pp. 185-215.
- Bohn, R. E. (1994): "Measuring and Managing Technological Knowledge", *Sloan Management Review*, Fall 1994
- Boisot, Max H. (1998): *Knowledge Assets: Securing Competitive Advantage in the Information Economy*. Oxford University Press, Oxford, USA.
- Bollen, K., and Lennox, R. (1991): "Conventional Wisdom on Measurement – a Structural Equation Perspective", *Psychological Bulletin*, Vol. 110, no. 2, pp. 305-314.
- Bonacci, D. (2004): "Towards quantitative tools for analysing qualitative properties of virtual communities", *Interdisciplinary Description of Complex Systems*, Vol. 2, no. 2, pp. 126-135.
- Bonaccorsi, A., Giannangeli, S., Rossi, C. (2006): "Hybrid business models in the open source software industry", *Management Science*, Vol. 52, no. 7, pp. 1085–1098.
- Bradshaw, P., Powell, S., and Terrel, I. (2004): *Building a community of practice: Technological and social implications for a distributed team*. In Hildreth, Paul M. and Kimble, Chris, editors, Knowledge Networks: Innovation through Communities of Practice.
- Brooks, F. P. (1995): *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, Reading, MA.
- Brown, J. S., and Duguid, P. (2000): *The social life of information*. Boston, MA: Harvard Business School Press.
- Brown, J. S., and Duguid, P. (2001): Knowledge and organization: A social practice perspective. *Organization Science*, Vol. 12, no. 2, pp. 198-213.

- Brunsson (1985): *The Irrational Organization*. Ed. Wiley
- Cabrera, A. (1999): *The Knowledge Sharing Dilemma*, Instituto de Empresa María de Molina 12. E-28006 Madrid
- Cai, D., He, X., and Han, J. (2005): "Document Clustering Using Locality Preserving Indexing", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, no. 12, pp. 1624-1637.
- Campbell, D.T. (1960): "Blind variation and selective retention in creative thought as in other knowledge processes", *Psychological Review*, Vol. 67, pp. 380–400.
- Carmines E. G. y Zeller, R. A. (1979): "Reliability and Validity Assessment". Sage University Paper Series on Quantitative Applications in the Social Sciences, nº 7017, Beverly Hills, CA: Sage
- Chen, C. (2003): "A constructivist approach to teaching: Implications in teaching computer networking". *Information Technology, Learning, and Performance Journal*, Vol. 21, no. 2, pp. 17–27.
- Chin, W. W.; Marcolin, B. L. y Newsted, P. R. (1996): "A Partial Least Squares Latent Variable Modeling Approach for Measuring Interaction Effects: Results from a Monte Carlo Simulation Study and Voice Mail Emotion/Adoption Study". Proceedings of the Seventeenth International Conference on Information Systems, 16-18 December, Cleveland, Ohio.
- Chin, W. W. (1998): "The Partial Least Squares Approach to Structural Equation Modeling", en Marcoulides, G. A. (Ed.), Modern Methods for Business Research: pp. 295-336. Mahwah, NJ: Lawrence Erlbaum Associates, Publisher.
- Chin, W.W. (2003), PLS-Graph (Version 3.00, Build 1058), [Computer software]. University of Houston.
- Choo, C. W. (1998): *The knowing organization: How organizations use information to construct meaning, create knowledge and make decisions*. Oxford University Press, New York, USA.
- Cohen, W.M. y Levinthal, D.A. (1990): "Absorptive Capacity: A New Perspective on Learning and Innovation", *Administrative Science Quarterly*, Vol. 35 (March), pp. 128-152

- Cook, S. C. N. y Brown, J. S. (1999): "Bridging Epistemologies: The Generative Dance Between Organizational Knowledge and Organizational Knowing", *Organization Science*, Vol. 10, nº 4, pp. 381-400.
- Cool, D.E., Tonks, N.K., Charbonneau, H., Walsh, K.A., Fischer, E.H., and Krebs, E.G. (1989): "cDNA isolated from a human T-cell library encoded a member of the protein-tyrosine-phosphatase family", In: (2nd ed.),*Proc. Natl. Acad. Sci. USA*, 86, pp. 5257–5261
- Crowston, K., Annabi, H., y Howison, J. (2003): "Defining Open Source Software Project Success", *International Conference on Information Systems*, pp. 327-340.
- Csikszentmihalyi, M. (1990): *Flow: The Psychology of Optimal Experience*. Harper y Row, New York.
- Csikszentmihalyi, M. (1996): *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Collins, New York.
- Davies, R. (1996): "Reflections on Knowledge. An idiosyncratic summary of the two 1996 Knowledge Conferences prepared by Robert Davies", http://www.mce.be/article/knowledge.htm
- Deci, E.L., Ryan, R.M. (1985): *Intrinsic Motivation and Self-Determination in Human Behavior*. Plenum, New York/London.
- Deek, F. P. And McHugh, J. A. M. (2008): *Open Source: Technology and Policy*, Cambridge University Press, NY.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990): "Indexing by latent semantic analysis", *J. Amer. Soc. Inf. Sci.*, Vol. 41, no. 6, pp. 391–407.
- DeLone, W.H. and McLean, E.R. (2003): "The DeLone and McLean model of information systems success: a ten-year update", *Journal of Management Information Systems*, Vol. 19, pp. 9–30.
- Diamantopoulos, A. y Winklhofer, H. M. (2001): "Index Construction with Formative Indicators: an Alternative to Scale Development", *Journal of Marketing Research*, 38, pp. 269-277

- Dietrich, M. (1994): *Transaction Cost Economics and Beyond: Towards a New Economics of the Firm*, Routledge, London
- Dixon, P. M., J. Weiner, T. Mitchell-Olds, R. Woodley. (1987): "Bootstrapping the Gini coefficient of inequality", *Ecology*, Vol. 68, pp. 1548–1551.
- Dooley, K. J.; Skilton, P. F. y Anderson, J. C. (1998): "Process Knowledge Bases: Understanding Processes through Cause and Effect Thinking", *Human Systems Management,* Vol. 17, nº 4, pp. 281-296
- Ducheneau, N. (2005): "Socialization in an Open Source Software Community: A Socio-Technical Analysis", *Computer Supported Cooperative Work*, Vol. 14, no. 4, pp. 323-368.
- Dunn, W. N. (1982): "Reforms as argument", *Knowledge: Creation, Diffusion, Utilization,* Vol. 3, pp. 293-326.
- Dunn, W. N. y Holzner, B. (1982): *Methodological Research on Knowledge Use and School Improvement. Informe final*. Departamento de Educación de EE.UU., Washington, DC.
- Dyer, J. H. and Nobeoka, K. (2000): "Creating and managing a high-performance knowledge-sharing network: the toyota case", Strategic *Management Journal*, Vol. 21, pp. 345–367.
- Edvinsson, L. y Malone, M. S. (2000): *El Capital Intelectual. Cómo Identificar y Calcular el Valor de los Recursos Intangibles de su Empresa*. Ed. Gestión 2000
- Faba-Pérez, C., Zapico-Alonso, F., Guerrero-Bote, V. P., y de Moya-Anegón, F. (2005): "Comparative analysis of webometric measurements in thematic environments", *Journal of the American Society for Information Science and Technology*, Vol. 56, no. 8, pp. 779–785.
- Falk, R. F. y Miller, N. B. (1992): "A Primer for Soft Modeling" Akron, Ohio: The University of Akrom
- Faraj, S., Wasko, M.M. (2001): *The web of knowledge: An investigation of knowledge exchange in networks of practice*, Florida State University Working Paper, Tallahassee, FL.

- Ferrater Mora, J. (1985): Diccionario de Filosofía de Bolsillo. Ed. Alianza Editorial. Compilado por Priscilla Cohn.
- Fogel, K. and Bar, M. (2001): *Open Source Code Development with CVS*. 2nd Edition. Paragliph Press.
- Fontaine, M. An. and Millen, D. R. (2004): *Understanding the benefits and impact of communities of practice.* In Hildreth, Paul M. and Kimble, Chris, editors, Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, pp. 1–13.
- Fornell, C. (1982): "A Second Generation of Multivariate Analysis" Vol. 1, *Methods*. New York. Praeger
- Fornell, C. y Larcker, D. F. (1981): "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error", *Journal of Marketing Research*, nº 18, pp. 39-50
- Fornell, C., Lorange, P., and Roos, J. (1990): "The cooperative venture formation process: A latent variable structural modeling approach", *Management Science*, Vol. 36, no. 10, pp. 1246-1255.
- Fowler, C.J.H. and Mayes, J.T. (1999): "Learning relationships: from theory to design", *Association for Learning Technology Journal*, Vol. 7, no. 3, pp. 6–16.
- Fox, S. (2000): "Communities of Practice, Foucault And Actor-Network Theory", *Journal of Management Studies*, Vol. 37, no. 6, pp. 853-868.
- Franke, N., von Hippel, E., (2003): *Satisfying heterogeneous user needs via innovation toolkits: the case of apache security software*. http://opensource.mit.edu/papers/rp-vonhippelfranke.pdf. Acceso Febrero 2007.
- Free Software Foundation: GNU General Public License (2001): http://www.fsf.org/licenses/gpl.html. Acceso Febrero 2007.
- Freidson, E. (1970): *Profession of Medicine: A Study of the Sociology of Applied Knowledge*. Doddd, Meade, Nueva York
- Frey, B.S., Meier, S. (2004): "Pro-social behavior in a natural setting", *Journal of Economic Behavior and Organization*, Vol. 54, pp. 65–88.

- FSF. (2005): *Why "Free Software" is better than "Open Source"*, URL http://www.fsf.org/licensing/essays/free-software-for-freedom.html. Acceso Febrero 2007.
- Ganssle, J. and Barr, M. (2003): *Embedded Systems Dictionary*. Lawrence, KS: CMP Books.
- Garton, L., Haythornthwaite, C., and Wellman, B. (1997): "Studying Online Social Networks", *Journal of Computer-Mediated Communication*, Vol. 3, no. 1, http://207.201.161.120/jcmc/vol3/issue1/garton.html.
- Gini, C. (1936): *On the Measure of Concentration with Espacial Reference to Income and Wealth*. Cowles Comission, 1936
- Glaser, R. (1984): "Education and Thinking: The Role of Knowledge", *American Psychology*, Vol. 39, nº 2, pp. 93-104
- Gongla, P. and Rizutto, C.R. (2004): *Where did the community go? communities of practice that dissapear*. In Hildreth, Paul M. and Kimble, Chris, editors, Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, pp. 295–307.
- Grant, R. M. (1996): "Toward a Knowledge-based Theory of the Firm", *Strategic Management Journal*, Vol. 17, Special Issue (Winter), pp. 109-122.
- Griffiths, T. L., and Steyvers, M. (2004): "Finding scientific Tópicos", Proceedings of the National Academy of Sciences, Vol. 101, pp. 5228-5235.
- Hagedoorn, J., y Duysters, G. (2002): "Learning in Dynamic Inter-Firm Networks: The Efficacy of Multiple Contacts", *Organization Studies*, Vol. 23, no. 4, pp. 525-548.
- Hamel, G. (1991): "Competition for Competence and Interpartner Learning Within International Strategic Alliances", *Strategic Management Journal*, Vol. 12, pp. 83-103
- Hammer y Champy (1993): "Reeningeering the Corporation", Free Press
- Hars, A., and Ou, S. (2002): "Working for Free? – Motivations of Participating in Open Source Projects", *International Journal of Electronic Commerce*, Vol. 6, pp. 25–39.

- Henkel, J. (2003): "Software development in embedded Linux - informal collaboration of competing firms", *Proceedings der 6. Internationalen Tagung Wirtschaftsinformatik*, Dresden, pp. 1-20.
- Henkel, J. (2004): *Patterns of free revealing – balancing code sharing and protection in commercial open source development*. http://opensource.mit.edu/papers/henkel2.pdf. Acceso Febrero 2007.
- Hertel, G., Nieder, S., Herrmann, S. (2003): "Motivation of software developers in Open source projects: an Internet based survey of contributors to the Linux kernel", *Research Policy*, Vol. 32, no. 7, pp. 1159–1177.
- Hildreth, P., Wright, P., and Kimble, C. (1999): *Knowledge management: Are we missing something?* In Brooks, L. and C., Kimble, editors, Information Systems - The Next Generation, York, pp. 347–356.
- Hildreth, P.M. and Kimble, C. (2002): "The duality of knowledge", *Information Research*, Vol. 8, No. 1, pp. 1–17.
- Hislop, D. (2004): *The paradox of communities of practice: Knowledge sharing between communities*. In Hildreth, Paul M. and Kimble, Chris, editors, Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, pp. 36–46.
- Hofmann, T. (1999): "Probabilistic latent semantic indexing", Procs. Of the 22nd *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'99), Berkeley, CA, ACM Press, pp. 50-57.
- Hjertzén, E. y Toll, J. (1999): *Measuring Knowledge Management at Capt Gemini AB,* Tesis doctoral. Linköpings Universitet
- Holzner, B. y Marx, J. (1979): *Knowledge application: The Knowledge System in Society*. Allyn and Bacon, Boston, Massachussets
- Inkpen, A. (1998): "Learning, Knowledge Adquisition through International, Strategic Alliances", *The Academy of Management Executive*, Vol. 12, nº 4, pp. 69-80
- Inkpen, A. C. y Dinur A. (1998): "Knowledge Mangement Processes and International Joint Ventures", *Organization Science*, Vol. 9, nº 4, pp. 454-468

- Johnson, C. M. (2001): A survey of current research on online communities of practice. *Internet and Higher Education*, Vol. 4, pp. 45–60.
- Jones, G., G. Ravid, S. Rafaeli. (2004): "Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration", *Information Systems Research*, Vol. 15, pp. 194–210.
- Jonson, J.P. (2006): "Collaboration, peer review and open source software", *Information Economics and Policy*, Vol. 18, pp. 477–497.
- Kankanhalli, A., Tanudidjaja, F., Sutano, J., Tan, B. (2003): "The role of IT in successful knowledge management initiatives", *Communications of the ACM*, Vol. 46, no. 9, pp. 69–73.
- Karels, M.J. (2003): "Commercializing Open Source Software", *Queue*, Vol. 1, no. 5, pp. 46-55.
- Kautz, H., Selman, B., Shah, M., (1997): "Referral Web: combining social networks and collaborative filtering", *Communications of ACM*, Vol. 40, no. 3, pp. 27–36.
- Khanna, T.; Gulati, R. y Nohria, N. (1998): "The Dynamics of Learning Alliances: Competition, Cooperation and Relative Scope", *Strategic Management Journal*, Vol. 19, nº 3 (March), pp.193-210
- Kimble, C., Hildreth, P., and Wright, P. (2000): *Communities of practice: Going virtual*. In Hildreth, Paul M. and Kimble, Chris, editors, Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, pp. 220–234.
- Kimble, C. and Hildreth, P. (2004): "Communities of practice: Going one step too far?", *AIM*, pp. 1–7.
- Knock, N. (2001): "Compensatory adaptation to a lean medium: An action research investigation of electronic communication in process involvement groups", *IEEE Trans. on Professional Communication*, Vol. 44, no. 4, pp. 267–285.
- Kock, N. F., Jr., McQueen; R. J. y Corner, J. L. (1997): "The Nature of Data, Information and Knowledge Exchanges in Business Processes: Implications for

process Improvement and Organizational Learning", *The Learning Organization*, Vol. 4, nº 2, pp. 70-80

- Kogut, B. y Zander, U. (1992): "Knowledge of the Firm, Combinative Capabilities and the Replication of Technology", *Organization Science*, Vol. 3, nº 3, pp. 383-397
- Kogut, B., Metiu, A. (2000). *The emergence of E-Innovation: insights from open source software development*. Reginald H. Jones Center Working Paper, Philadelphia, PA.
- Koh, J., Kim, Y. G. (2004): "Knowledge sharing in virtual communities: an e-business perspective", *Expert Systems with Applications*, Vol. 26, Iss. 2, pp. 155-166.
- Koh, J., Kim, Y.-G., Butler, B., Bock, G.-W. (2007): "Encouraging Participation in Virtual Communities", *Communications of the ACM*, vol. 50, no. 2, pp. 69-73.
- Kramer, R.M., Tyler, T.R. (1996). *Trust in Organizations: Frontiers of Theory and Research*. Sage Publications, Thousand Oaks, CA.
- Krishnamurthy, S. (2002): *Cave or community? An empirical examination of 100 mature open source projects*. First Monday 7. URL http://www.firstmonday.org, Acceso Febrero 2007.
- Krogh, G., Spaeth, S., Lakhani, K. (2003): "Community, joining, and specialisation in open source software innovation: a case study", *Research Policy*, Vol. 32, pp. 1217–1241.
- Kruskal, W.H., Wallis, W.A. (1952): "Use of ranks on one-criterion variance analysis", *Journal of the American Statistical Association*, Vol. 47, pp. 583-621.
- Kuhn, T. S. (1970): *The Structure of Scientific Revolutions* (2ª ed.) Chicago: Chicago University Press
- Kuk, G. (2004): "Selection, cliques and knowledge sharing in open source software development communities", *Proc. IADIS Internat. Conf. Web-Based Communities*, IADIS, Lisbon, Portugal.
- Kuk, G. (2006): "Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List", *Management Science*, Vol. 52, No. 7, pp. 1031–1042.

- Lakhani, K., Hippel, E. (2003): "How open source software works: 'free' user to user assistance", *Research Policy*, Vol. 32, pp. 923–943.
- Lakhani, K.R., Wolf, R.G. (2005): *Why hackers do what they do: Understanding motivation effort in free/open source software projects*. In: Feller, Joseph, Fitzgerald, Brian, Hissam, Scott A., Lakhani, Karim R. (Eds.), Perspectives on Free and Open Source Software. MIT Press, Cambridge.
- Landauer, T. (2002): "On the computational basis of learning and cognition: Arguments from LSA", *The psychology of learning and motivation*, Vol. 41, pp. 43-84.
- Lane, P. J. y Lubatkin, M. (1998): "Relative Absortive Capacity and Interorganizational Learning", *Strategic Management Journal*, Vol. 19, nº 5 (May), pp.461-477
- Lave, J; Wenger, E. and Pea, R. (1991): *Situated learning: legitime peripheral participation*. Cambridge: Cambridge University Press.
- Lee, G.K., Cole, R.E. (2003): "From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development", *Organization Science*, Vol. 14, no. 6, pp. 633-649.
- Lerner, J., Tirole, J. (2002): "Some simple economics of open source", *Journal of Industrial Economics*, Vol. 50, pp. 197–234.
- Lesser, E.L. and Storck, J. (2001): "Communities of practice and organizational performance", *IBM Systems Journal*, Vol. 40, no. 4, pp. 831–841.
- Levine, J. M.; Resnick, L. B. y Higgins, E. T. (1993): "Social Foundations of Cognition", *Annual Review Psychology*, Vol. 44, pp. 585-612
- Levinson, N. S. y Asahi, M. (1995): "Cross-National Alliances and Interorganizational Learning", *Organizational Dynamics*, Vol. 24, nº 2 (Autumn), pp. 50-63
- LeVitt, B., March, J. (1988), "Organizational learning", *Annual Review of Sociology*, Vol. 14, pp. 319–340.
- Liebowitz, J. (2001): "Knowledge management and its link to artificial intelligence", *Expert Systems with Applications*, Vol. 20, Iss. 1, 2001, pp. 1-6.

- Lin, H.-F. y Lee, G.-G. (2006): "Determinants of success for online communities: an empirical study", *Behaviour y Information Technology*, Vol. 25, No. 6, pp. 479-488.
- Lueg, C. (2000): "Where is the action in virtual communities of practice", *Proceedings of the Workshop Communication and Cooperation in Knowledge Communities* at the D-CSCW 2000 German Computer-Supported Cooperative Work Conference 'Verteiltes Arbeiten - Arbeit der Zukunft', pp. 1–7.
- MacCallum, R. C. and Browne, M. W. (1993): "The use of Causal Indicators in Covariance Structure Models – Some Practical Issues", *Psychological Bulletin*, Vol. 114, no 3, pp. 533-541.
- Manning, C. D., y Schutze, H. (1999): *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA.
- March J. y Olsen (1976):*Ambiguity and Choice in Organizations*. Ed. Bergen.
- Martínez-Torres, M. R., Toral, S. L. (2010): "Strategic Group Identification using Evolutionary Computation", Experts Systems with Applications, Vol. 37, Iss. 7, pp. 4948-4954.
- Martínez-Torres, M. R., Palacios, B., Toral, S. L., Barrero, F. (2010): "Application of Genetic Algorithms to the Identification of Website Link Structure", The 2010 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2010, pp. 1-8.
- Mateos-García, J. y Steinmueller, W. E. (2008): "The institutions of open source software: Examining the Debian community", *Information Economics and Policy*, Vol. 20, pp. 333–344
- Maybury, M. (2001): Collaborative Virtual Environments for Analysis and Decision Support, *Communications of the ACM*, Vol. 44, no. 12, pp. 51-54.
- Mclure Wasko, M. and Faraj, S. (2000): "It is what one does: why people participate and help others in electronic communities of practice", *Journal of Strategic Information System*s, Vol. 9, pp. 155–173.
- Mclure Wasko, M. and Faraj, S. (2005): "Why should i share? examining social capital and knowledge contribution in electronic networks of practice", *Management Information Systems Quarterly*, Vol. 29, no. 1, pp. 35–57.

- Meyer, I. y K. Mackintosh (1996): "The Corpus from a Terminographer's Viewpoint", *International Journal of Corpus Linguistics*, Vol. 1, no. 2, pp. 257-285.
- Millen, D., Feinberg, J., and Kerr, B. (2005): "Social bookmarking in the enterprise", *ACM Queue*, pp. 28–35.
- Mintzberg, H. (1980): *La Naturaleza del Trabajo Directivo*. Ed. Prentice Hall
- Mintzberg, H. y Pettigrew, A. M. (1990): "Studying Deciding", *Organization Studies*, Vol. 11, nº 1
- Mockus, A., Fielding, R.T., Herbsleb, J.D. (2002): "Two Case Studies of Open Source Software Development: Apache and Mozilla", *ACM Transactions on Software Engineering and Methodology*, Vol. 11, No. 3, pp. 309–346.
- Molla, A. y Licker, P.S. (2001): "E-commerce system success: an attempt to extend and respecify the DeLone and McLean model of IS success", *Journal of Electronic Commerce Success*, Vol. 2, pp. 1–11.
- Moon, J.Y., Sproull, L. (2000): "Essence of distributed work: the case of the Linux kernel", *Firstmonday*, 5, 11.
- Murdock, I. (1993): *New release under development, suggestions requested*. Disponible en http://ianmurdock.com [1/05/2010]
- Nahapiet, J. y Ghoshal, S. (1998): "Social Capital, Intellectual Capital and the Organizational Advantage", *Academy of Management Review*, Vol., 23, pp. 242-267
- Nelson, R. y Winter, S. (1982): *An Evolutionary Theory of Economic Change*. Belknap Press, Cambridge, MA
- Newman, D., Chemudugunta, C., Smyth, P., Steyvers, M. (2006): *Analyzing Entities and Tópicos in News Articles using Statistical Tópico Models*, LNCS 3975, Intelligence and Security Informatics, Springer.
- Ng, A.Y., Jordan, M., and Weiss, Y. (2001): *On Spectral Clustering: Analysis and an Algorithm*, Advances in Neural Information Processing Systems 14, Cambridge, Mass., MIT Press, 849-856.

- Niedner, S., Hertel, G., Hermann, S. (2000): *Motivation in Open Source Projects: An Empirical Study Among Linux Developers*. Summarized study results URL http:/www.psychologie.uni-kiel.de/linux-study. Acceso Febrero 2007.
- Nonaka, I. (1994): "A Dynamic Theroy of Organizational Knowledge Creation", *Organizational Science*, Vol. 5, nº 1, pp. 14-37
- Nonaka, I. y Takeuchi, H. (1995): *The Knowledge-Creating Company*. Oxford University Press, New York.
- Nonnecke, B., y Preece, J. (2000): "Lurker demographics: Counting the silent", *Proceedings of Human Factors in Computing Systems* (CHI 2000), ACM Press, pp. 73–80.
- Nooy, W., Mrvar, A., y Batagelj, V. (2005): *Exploratory Network Analysis with Pajek*, Cambridge University Press, New York.
- Nunnally, J. C. (1978): *Psychometric Theory*. 2nd Ed. New York, McGraw Hill
- O'Reilly, T. (1999): *Ten Myths about Open Source Software*. Published on O'Reilly.
- O'Reilly, T. (2005): *The open source paradigm shift*, in: J. Feller, B. Fitzgerald, S.A. Hissam, K.R. Lakhani (Eds.), Perspectives on Free and Open Source Software, MIT Press, Cambridge, MA, pp. 461–481.
- Ortigueira Bouzada, M. (1991): "La Comunicación en las Corporaciones Locales", *Temas de Administración Local*, nº 42, pp. 253-294
- Osterloh, M., Frey, B.S. (2000): "Motivation, knowledge transfer, and organizational firms", *Organization Science*, Vol. 11, pp. 538–550.
- Osterloh, M., Rota, S. (2007): "Open source software development—Just another case of collective invention?", *Research Policy*, Vol. 36, no. 2, pp. 157-171.
- Palm, W. J. (2003): Introduction to MATLAB 7 for Engineers, McGraw-Hill Science/Engineering/Math; 2 edition, NY.

- Pan, S. L. and Leidner, D. E. (2003): "Bridging communities of practice with informaton technology in pursuit of global knowledge sharing", *Journal of Strategic Information Systems*, Vol. 12, pp. 71–88.
- Penrose, E.T. (1959): *The Theory of the Growth of the Firm*. Blackwell, Oxford, England.
- Perens, B. (2005): *Bruce Perens - Biographical Notes and Resume*. URL http://perens.com/Articles/Bio.html. Acceso Febrero 2007.
- Pfeffer, J. y Sutton, R. I. (1999): *The Knowing-Doing Gap. How Smart Companies Turn Knowledge into Action*. Ed. Harvard Business School Press
- Polanyi, M. (1966): "The Tacit Dimension", Anchor Day Books, New York.
- Porter, M. (1985): "Competitive Advantage: Creating and Sustaining Superior Performance", Free Press, New York
- Powell, W.W., Koput, K.W., Smith-Doerr, L. (1996): "Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology", *Administrative Science Quarterly*, Vol. 41, no. 1, pp. 116–145.
- Prahalad, C. K. y Hamel, G. (1990): "The Core Competence and the Corporation", *Harvard Business Review*, May-June, pp. 71-91.
- Preece, J. (2001): "Sociability and usability: twenty years of chatting online", *Behaviour and Information Technology*, Vol. 20, no. 5, pp. 347–356.
- Prigogine, I. (1989): "The philosophy of instability", *Futures*, Vol. 21, pp. 396-400
- Raghavan, P., (2002): "Social networks: from the Web to the enterprise", *IEEE Internet Computing*, Vol. 6, no. 1, pp. 91–94.
- Raghavan, P., Lad, A., Neelakandan, S. (2006): *Embedded Linux System Design and Development*. Auerbach Publications, Taylor and Francis Group.
- Raymond, E.S. (1998a). *Goodbye, "free software"; hello, "open source"*. URL http://www.catb.org/~esr/open-source.html. Acceso Febrero 2007.
- Raymond, E.S. (1998b). *Homesteading the Noosphere*. First Monday, 3, 10. URL http://www.firstmonday.org/issues/issue3_10/raymond/index.html. Acceso Febrero 2007.

- Raymond, Eric S. (2001): *The Cathedral and the Bazaar*. Cambridge: O'Reilly.
- Register, A. H. (2007): *A Guide to MATLAB Object-Oriented Programming*. Chapman y Hall/CRC Press, NW.
- Rencher, A.C. (2002): *Methods of Multivariate Analysis*. Second Edition, A John Wiley y Sons, Inc. Publication.
- Resnick, L. B.; Levine, J. M. y Teasley, S. D. (1991): *Perspectives on Socially Shared Cognition. American Psychological Association*, Washington, DC
- Resnick, L.B. y Collins, A. (1996): "Cognition and Learning", en el libro *International Encyclopedia of Educational Technology*, 2ª edición, editado por Plomp, T. y Ely, D. P. Editorial Pergamon., pp. 48-51
- Robertson, R. y Holzner (1980): *Identity and Authority: Explorations in the Theory of Society*. St. Martin's Press, Nueva York
- Rogoff, B. (1990): *Apprenticeship in Thinking: Cognitive Development in Social Context*. Oxford University Press, New York.
- Roldán, J. L. (2000): "Sistemas de Información Ejecutivos (EIS): Génesis, Implantación y Repercusiones Organizativas". Tesis doctoral. Universidad de Sevilla.
- Rolf, B. (1991): Profession tradition och tyst kunskap. Doxa
- Rowley, T., Behrens, D., y Krackhardt, D. (2000): "Redundant Governance Structures: An Analysis of Structural and Relational Embeddedness in the Steel and Semiconductor Industries", *Strategic Management Journal*, Vol. 21, no. 3, pp. 369-386.
- Russel, L. P., Sambamurthy, V., and Zmud, R. W. (2001): The assimilation of knowledge platforms in organizations: An empirical investigation. *Organization Science*, Vol. 12, no. 2, pp. 117–135.
- Ryan, R. M., Deci, E. L. (2000): "Intrinsic and extrinsic motivations: Classic definitions and new directions", *Contemporary Educational Psychology*, Vol. 25, pp. 54–67.
- Salto, G., and McGill, M.J. (1983): *An Introduction to Modern Information Retrieval*, New York: McGraw-Hill.

- Sánchez Cerezo, S. (1983): Diccionario de las Ciencias de la Educación. Publicaciones DIAGONAL SANTILLANA para profesoresVol. 1 A-H, pp. 308-309
- Sanz, S. (2005): Comunidades de práctica virtuales: acceso y uso de contenidos. *Revista de Universidad y Sociedad del Conocimiento*, Vol. 2, no. 2, pp. 26-35.
- Schmidt, Douglas C. and Porter, Adam. (2001): *Leveraging Open-Source Communities To Improve the Quality y Performance of Open-Source Software*. Position Paper, First Workshop on Open-Source Software Engineering, 23rd International Conference on Software Engineering, Toronto, Canada.
- Seely Brown, J. and Duguid, P. (2000): "Balancing act: How to capture knowledge without killing it", *Harvard Business Review*, pp. 73–80.
- Seely Brown, J. and Duguid, P. (1991): "Organizational learning and communitities-of-practice: Toward a unified view of working, learning and innovation", *Organization Science*, Vol. 2, no. 1, pp. 40–57.
- Selznick, P. (1957): "Leadership in Administration"
- Shah, S. (2006): "Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development", *Management Science*, Vol. 52, No. 7, pp. 1000–1014.
- Sharma, S. (1998): *Applied Multivariate Techiques*. John Wiley and Sons
- Shenkar, O. y Li, J. (1999): "Knowledge Search in International Cooperative Ventures", *Organization Science*, Vol. 10, nº 2, pp. 134-143
- Shi, J., and Malik, J. (2000): "Normalized Cuts and Image Segmentation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, no. 8, pp. 888-905.
- Simon, H. (1976): *Administrative Behavior*. (3ª edición) Free Press, New York
- Skok, W. and Kalmanovitch, C. (2005): "Evaluating the role and effectiveness of an intranet in facilitating knowledge management: a case study at surrey county council", *Information y Management*, Vol. 42, pp. 731-744.
- Skyrme, D. J. (1994): "The Knowledge Asset", *Management Insight*, nº 11

- Snow, C. C. y Hrebiniak, L. G. (1980): "Strategy, distinctive Competence and Organisational Performance", *Administrative Science Quarterly*, Vol. 25, pp. 322-334
- Soto, L. D. (1998): "Un reporte de Administración del Conocimiento", http://www.luisdans.com
- Sowe, S., Stamelos, I., Angelis,L. (2006): "Identifying knowledge brokers that yield software engineering knowledge in OSS projects", *Information and Software Technology*, Vol. 48, pp. 1025–1033.
- Spender, J. C. (1996a): "Making Knowledge the Basis of a Dynamic Theory of the Firm", *Strategic Management Journal*, Vol. 17, Special Issue (Winter), pp. 45-62
- Spender, J. C. y Grant, R. M. (1996): "Knowledge and the Firm: Overview", *Strategic Management Journal,* Vol. 17, Special Issue (Winter), pp. 5-9
- Spender, J.-C. (1996b): "Organizational Knowledge, Learning and Memory: Three Concepts in Search of a Theroy", *Journal of Organizational Change Management*, Vol. 9, pp. 63-78
- Spender, J-C. (1996c): "Competitive Advantage from Tacit Knowledge? Unpacking the Concept and Its Strategic Implications", B. Moingeon, A. Edmondson, eds. *Organisational Learning and Competitive Advantage*, Sage, LondonVol. 14
- Squire, K. and Johnson, C. (2000): Supporting distributed communities of practice with interactive television. *Educational Technology Research and Development*, Vol. 20, no. 3, pp. 23–43.
- Stallman, R. (1992): *Why Software Should Be Free*. URL http://www.gnu.org/ philosophy/shouldbefree.html. Acceso Febrero 2007.
- Stewart, T.A. (1997): *Intellectual Capital: The New Wealth of Organizations*, Doubleday/Currency, New York
- Subramanyam, R., Xia, M. (2008): "Free/Libre Open Source Software development in developing and developed countries: A conceptual framework with an exploratory study", *Decision Support Systems*, Vol. 46, Iss. 1, pp. 173-186.

- Sveiby, K-E. (1994): "Towards a Knowledge Perspective on Organisation", Tesis Doctoral. Department of Business Administration. University of Stockholm
- Sveiby, K-E. (1998): "Measuring Intangibles and Intellectual Capital. An Emerging First Standard", Internet version, August 5, 1998
- Sveiby, K-E. (2000): *Capital Intelectual. La nueva riqueza de las empresas*. Ed. Gestión 2000
- Szulanski, G. (1996): "Exploring Internal Stickiness: Impediments to the Transfer of Best Practice Within the Firm", *Strategic Management Journal*, Vol. 17, Special Issue (Winter), pp. 27-43
- Taylor, C. (1993): "To Follow a Rule ...". en el libro de C. Calhoun, E. Lipuma and M. Portone (eds.), *Bourdieu: Critical Perspectives*. Polity Press, Cambridge, U.K., pp. 45-59
- Taylor, R. M. (1996): "Knowledge Management", http://ourworld.compuserve.com/homepages/roberttaylor/km.htm
- Teece, D. J. (1981): "The Multinational Enterprise: Market Failure and Market Power Considerations", *Sloan Management Review*, Vol. 22, nº 3, pp. 3-17
- Teece, D. (1986): "Profiting from technological innovation: Implications for integration, collaboration licensing and public policy", *Research Policy*, Vol. 15, pp. 285–305.
- Toral, S.L., Barrero, F., Martínez-Torres, M.R., Gallardo, S., Lillo, J. (2005): "Implementation of a Web-Based Educational Tool for Digital Signal Processing Teaching Using the Technological Acceptance Model", *IEEE Transactions on Education*, Vol. 48, Iss. 4, pp. 632-641.
- Toral, S. L., Martínez-Torres, M. R., Barrero, F. (2009a): "Modelling Mailing List Behaviour in Open Source Projects: the Case of ARM Embedded Linux", *Journal of Universal Computer Science*, Vol. 15, no. 3, pp. 648-664.
- Toral, S. L., Martínez-Torres, M. R. y Barrero, F. (2009b): "Virtual Communities as a resource for the development of OSS projects: the case of Linux ports to embedded processors", *Behavior and Information Technology*, Vol. 28, no. 5, pp. 405-419.

- Toral, S. L., Martínez-Torres, M. R. y Barrero, F., Cortés, F. (2009c): "An empirical study of the driving forces behind online communities", *Internet Research,* Vol. 19, no. 4, pp. 378-392.
- Toral, S. L., Martínez-Torres, M. R., Barrero, F. (2010a): "Analysis of virtual communities supporting OSS projects using social network analysis", *Information and Software Technology*, Vol. 52, no. 3, pp. 296-303.
- Toral, S. L., Martínez-Torres, M. R. y Barrero, F., Cortés, F. (2010b): "The role of Internet in the development of Future Software Projects", *Internet Research*, Vol. 20, no. 1, pp. 72-80.
- Toral, S. L., Martínez-Torres, M. R. (2010c): "International Comparison of RyD Investment by European, US and Japanese Companies", *International Journal of Technology Management*, Vol. 49, no. 1/2/3, pp. 107-122.
- Torvalds, L., Diamond, D. (2001): *Just for Fun: The Story of an Accidental Revolutionary*. Harper Business.
- Tsoukas, H. (1996): "The Firm as a Distributed Knowledge System: A Constructionist Approach", *Strategic Management Journal*, Vol. 17, Special Issue (Winter), pp. 11-25
- Umstätter, W. (1998): "Knowledge Measurement", September, http://www.ib.hu-berlin.de/wumsta/dhb3e.html
- Van Den Bosch, F. A. J.; Volberda, H.W. y de Boer, M. (1999): "Coevolution of Firm Absorptive Capacity and Knowledge Environment: Organizational Forms and Combinative Capabilities", *Organization Science*, Vol. 10, nº 5, pp. 551-568.
- Van den Hooff, B., Huysman, M. (2010): Managing knowledge sharing: Emergent and engineering approaches, *Information and Management*, en prensa.
- Ven, K., Mannaert, H. (2008): "Challenges and strategies in the use of Open Source Software by Independent Software Vendors", *Information and Software Technology*, Vol. 50, Iss. 9-10, pp. 991–1002.
- Viedma, J. M. (2001): "ICBS - Intellectual Capital Benchmarking System", *Journal of Intellectual Capital*, Vol. 2, nº 2

- Von Hippel, E. (1988): *The Sources of Innovation*. Oxford University Press, New York.
- Von Hippel, E., von Krogh, G. (2003): "Open source software and the "private-collective" innovation model: issues for organization science", *Organization Science*, Vol. 14, no. 2, pp. 209–223.
- von Krogh, G. y Roos, J. (1996a): "A Tale of the Unfinished", *Strategic Management Journal,* Vol. 17, pp. 729-737.
- von Krogh, G. y Roos, J. (1996b): *Managing Knowledged: Perspectives on Coopertation and Competition*. SAGE Publications, London.
- von Krogh, G. y Roos, J. (1996c): "Five Claims on Lnowing", *European Management Journal*, Vol. 13, nº 4, pp. 423-426.
- Wachter, R.M., Gupta, J.N.D. and Quaddus, M.A. (2000): "IT takes a village: virtual communities in support of education", *International Journal of Information Management*, Vol. 20, pp. 473 – 489.
- Wai, F.B. (2008): "Reuse of knowledge assets from repositories: A mixed methods study", *Information y Management*, Vol. 45, Iss. 6, pp. 365-375
- Wasko, M. and Faraj, S. (2005): "Why should i share? examining social capital and knowledge contribution in electronic networks of practice", *Management Information Systems Quarterly*, Vol. 29, no. 1, pp. 35–57.
- Wasserman, S., Faust, K., y Iacobucci, D. (1994): *Social Network Analysis: Methods and Applications* (Structural Analysis in the Social Sciences). Cambridge University Press.
- Weber, S. (2004): *The Success of Open Source*. Harvard University Press.
- Wellman, B., Gulia, M. (1999): *Virtual communities as communities*. In: Smith, M.A., Kollock, P. (Eds.), Communities in Cyberspace. Routledge, New York, NY.
- Weick, K. (1979): *The Social Psychology of Organizing*. Addison-Wesley, Reading, M.A.
- Wenger, E. (1998): *Communities of Practice: Learning, Meaning and Identity*. Cambridge University Press, New York, USA.

- Wenger, E.; Snyder, W. (2000): Communities of practice: the organizational frontier. *Harvard Business Review*, Vol. 78, no 1, pp. 139-145.
- Wenger, E., McDermott, R., and Snyder, W. M. (2002): *Cultivating Communities of Practice*. Harvard Business School Press, Boston, USA.
- Werts, C.E., Linn, R.L. and Jöreskog, K.G. (1974): *Quantifying unmeasured variables*. In: Blalock, H.M., Editor, , 1974. Measurement in the social sciences, Aldine, Chicago.
- Wheeler, D. (2007): *How to evaluate Open Source / Free Software (OSS/FS) Programs*. URL http://www.dwheeler.com/oss_fs_eval.html. Acceso Febrero 2007.
- Wiig, K. M. (1997): "Integrating Intellectual Capital and Knowledge Management", *Long Range Planning*, Vol. 30, nº 3, pp. 399-405
- Wiig, K. M. (1998): "On the Management of Knowledge", http://www.km-forum.org/wiig.htm
- Wikström y Norman (1992): "Kunskap och Värde FA Radet"
- Winter, S. (1995): "Four Rs of Profitability: Rents, Resources, Routines and Replication", en el libro C. A. Montgomery (ed.), *Resource-based and Evolutionary Theories of the Firm: Towards a Synnthesis*. Kluwer, Norwell, MA, pp. 147-178
- Winter, S. G. (1987): "Knowledge and Competence as Strategic Assets", en el libro de D. Teece (eds.), *The Competitive Challenge: Strategies for Industrial Innovation and Renewal*. Ballinger, Cambridge, MA, pp. 159-184
- Wold, H. (1979): "Model Construction and Evaluation when Theorical Knowledge is Scarce: Theory and Application of Partial Least Squares", Cahiers du Département D'Économétrie. Géneve: Faculté des Sciences Economiques et Sociales, Universitè de Gèneve
- Xu, J., Gao, Y., Christley, S. and Madey, G. (2005): "A Topological Analysis of the Open Source Software Development Community", *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. HICSS '05.,188-198.

- Xu, W., Liu, X., and Gong, Y. (2003): Document Clustering Based on Non-Negative Matrix Factorization, *Proc. Int'l Conf. Research and Development in Information Retrieval*, pp. 267-273.
- Xu, W., and Gong, Y. (2004): Document Clustering by Concept Factorization, *Proc. Int'l Conf. Research and Development in Information Retrieval*, pp. 202-209.
- Yaghmour, K. (2003): *Building Embedded Linux Systems*. O'Reilly.
- Yang, S. J. H., Chen, I. Y.L. (2008): "A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network", *Int. J. Human-Computer Studies*, Vol. 66, pp. 36–50.
- Ye, Y., Nakakoji, K., Yamamoto, Y., and Kishida, K. (2005): *The co-evolution of systems and communities in free and open source software development*. Free/open source software development. S. Koch. Hershey, PA, Idea Group Inc (IGI), 59-82.
- Zha, H., Ding, C., Gu, M., He, X., and Simon, H. (2001): *Spectral Relaxation for k-Means Clustering*, Advances in Neural Information Processing Systems 14, Cambridge, Mass.: MIT Press, 1057-1064.
- Zhang, W. and Storck, J. (2001): *Peripheral Members in Online Communities*. In: AMCIS, Boston, MA, [online]. Source. Available from: URL http://opensource.mit.edu/papers/zhang.pdf
- Zittrain, J. (2004): Normative Principles for Evaluating Free and Proprietary Software. University of Chicago Law Review, Vol. 71, no. 1, pp. 265.