



Master in Computing

Master of Science Thesis

**ESTUDIO SOBRE LA CALIDAD DE LA
INFORMACION PROVISTA POR
COMUNIDADES OPEN SOURCE**

Antonio Pérez Jiménez

Advisors: Claudia P. Ayala

08/09/2010

Contenido

1.	Introducción	8
1.1.	Motivación del proyecto	8
1.2.	Objetivos	9
1.3.	Justificación	9
1.4.	Estructura de la tesis	10
2.	Estado del arte y Trabajos Relacionados.....	11
2.1.	Open Source Software	11
2.1.1.	Qué es OSS	11
2.1.2.	Roles en OSS.....	13
2.1.2.1.	Roles en la comunidad OSS.....	13
2.1.2.2.	Roles Industriales	14
2.1.2.3.	Roles en el mercado OSS	15
2.1.3.	Modelos de negocio y Licencias	16
2.1.3.1.	Modelos de negocio en el mercado.....	16
2.1.3.2.	Licencias OSS	18
2.2.	La importancia de la Calidad en el Desarrollo de Software	20
2.2.1.	¿Qué es un modelo de calidad en software?	21
2.2.2.	Estructura y métricas en un modelo de calidad.....	22
2.2.3.	Modelo de calidad: ISO/IEC 9126.....	23
2.2.4.	Individual Quality Model Construction (IQMC)	26
3.	Metodología.....	27
3.1.	Métodos de Investigación e Ingeniería de Software Empírica.....	27
3.2.	Metodología seguida	28
4.	Construcción del Framework	30
4.1.	Paso 1: Analizar y clasificar las respuestas obtenidas del estudio	30
4.2.	Paso 2: Construcción del Modelo de Calidad.....	34
4.2.1.	Determinación de Características de Alto Nivel	35
4.2.2.	Determinación de Sub-características y Atributos.....	36

4.2.3.	Definición de Métricas	39
5.	Survey	44
5.1.	Mozilla Firefox	45
5.1.1.	Resultados de la investigación	45
5.1.2.	Conclusiones de la investigación del proyecto.....	51
5.2.	Agilo for Scrum	53
5.2.1.	Resultados de la investigación	53
5.2.2.	Conclusiones de la investigación del proyecto.....	58
5.3.	Fluent NHibernate.....	60
5.3.1.	Resultados de la investigación	60
5.3.2.	Conclusiones de la investigación del proyecto.....	65
5.4.	Gimp.....	66
5.4.1.	Resultados de la investigación	66
5.4.2.	Conclusiones de la investigación del proyecto.....	71
5.5.	GNU Grub	72
5.5.1.	Resultados de la investigación	72
5.5.2.	Conclusiones de la investigación del proyecto.....	77
5.6.	Iptables.....	79
5.6.1.	Resultados de la investigación	79
5.6.2.	Conclusiones de la investigación del proyecto.....	84
5.7.	Joomla!.....	85
5.7.1.	Resultados de la investigación	85
5.7.2.	Conclusiones de la investigación del proyecto.....	90
5.8.	MediaCoder	92
5.8.1.	Resultados de la investigación	92
5.8.2.	Conclusiones de la investigación del proyecto.....	96
5.9.	StatusNet.....	97
5.9.1.	Resultados de la investigación	97
5.9.2.	Conclusiones de la investigación del proyecto.....	102

5.10.	Subclipse	104
5.10.1.	Resultados de la investigación	104
5.10.2.	Conclusiones de la investigación del proyecto.....	109
6.	Análisis de los resultados obtenidos	110
6.1.	Resumen del survey	110
6.2.	Análisis del estudio realizado.....	113
7.	Validez y limitaciones del estudio	121
7.1.	Validez de la construcción	121
7.2.	Validez interna.....	121
7.3.	Validez externa.....	122
8.	Conclusiones Finales y Trabajo Futuro	123
9.	Bibliografía.....	124

Lista de Tablas

Tabla 1: Diferencias entre FOSS y OSS 2.0	17
Tabla 2: El modelo de calidad interno/externo ISO/IEC 9126-1	25
Tabla 3: Organizaciones participantes en el estudio	30
Tabla 4: Vista de proyectos y componentes OTS utilizados por los entrevistados	32
Tabla 5: Modelo de calidad desarrollado.....	42
Tabla 6: Modelo de calidad del proyecto Mozilla Firefox	48
Tabla 7: Modelo de calidad del proyecto Agilo	55
Tabla 8: Modelo de calidad del proyecto Fluent NHibernate	62
Tabla 9: Modelo de calidad del proyecto Gimp	68
Tabla 10: Modelo de calidad del proyecto Grub.....	74
Tabla 11: Modelo de calidad del proyecto Iptables	81
Tabla 12: Modelo de calidad del proyecto Joomla!	87
Tabla 13: Modelo de calidad del proyecto MediaCoder.....	94
Tabla 14: Modelo de calidad del proyecto StatusNet	99
Tabla 15: Modelo de calidad del proyecto Subclipse	106
Tabla 16 Resumen de los resultados obtenidos en el estudio realizado	112

Lista de ilustraciones

Ilustración 1: Roles des de la perspectiva interna de la comunidad OS	14
Ilustración 2: Roles des de la perspectiva de mercado OS	15
Ilustración 3: Vistas del estándar ISO/IEC 9126.....	24
Ilustración 4: Metodología seguida.....	29
Ilustración 5: Frecuencia de respuestas obtenidas en cada criterio	34
Ilustración 6: Página principal Firefox	45
Ilustración 7: Página principal Agilo	53
Ilustración 8: Servicios de personalización de Agilo.....	57
Ilustración 9: Página principal Fluent Nhibernate	60
Ilustración 10: Página principal del proyecto Gimp.....	66
Ilustración 11: Página principal del proyecto Grub	72
Ilustración 12: Información acerca de Grub.....	76
Ilustración 13: Página principal del proyecto lptables.....	79
Ilustración 14: Página principal del proyecto Joomla!.....	85
Ilustración 15: Página principal del proyecto MediaCoder.....	92
Ilustración 16: Página principal del proyecto Status.net	97
Ilustración 17: Página principal del proyecto Subclipse.....	104
Ilustración 18: Resultados de los atributos pertenecientes a la característica <i>Estabilidad de la información del componente</i>	113
Ilustración 19: Resultados de los atributos pertenecientes a la característica <i>Evidencia de éxito en el uso del componente</i>	114
Ilustración 20: Resultados de los atributos pertenecientes a la característica <i>Disponibilidad de mecanismos de servicios de soporte</i>	115
Ilustración 21: Resultados de los atributos pertenecientes a la característica <i>Facilidad de integración del componente</i> a excepción del atributo <i>Disponibilidad del código fuente y comentarios</i>	116
Ilustración 22: Resultados del atributo <i>Disponibilidad del código fuente y comentarios</i>	117
Ilustración 23: Resultados del atributo <i>Documentación del usuario final</i>	118
Ilustración 24: Resultados del atributo <i>Documentación de integradores</i>	118

Ilustración 25: Resultados del atributo <i>FAQs</i>	119
Ilustración 26: Resultados del atributo <i>Ficheros de ayuda</i>	119
Ilustración 27: Resultados del atributo <i>Ayuda online</i>	120
Ilustración 28: Resultados del atributo <i>Idiomas de la documentación</i>	120

1. Introducción

En este capítulo se procederá con la introducción de la tesis. Inicialmente se comentará la motivación del proyecto, seguido de los objetivos que se desean conseguir con la realización de esta tesis. El siguiente punto a tratar será la justificación del trabajo a realizar y finalmente se presentará la estructura que seguirá la tesis.

1.1. Motivación del proyecto

En los últimos años, el desarrollo de software basado en componentes creados por terceros, de aquí en adelante DSBC, se ha convertido en una parte vital de la industria del software. Especialmente, el uso de componentes de código abierto (OSS) está revolucionando la industria en los últimos años. Cabe mencionar que cuando nos referimos a OSS, también nos referimos a Free Software.

Dado que la definición de componente OSS puede ser definida de diversas maneras, en el presente trabajo y siempre que hagamos referencia a los componentes de código abierto, nos referimos a: *“una pieza de software que otro proyecto de software puede reutilizar e integrar dentro de su propio producto”* [torchiano2004a].

El éxito del DSBC, especialmente los basados en componentes OSS, de aquí en adelante DBSC-OSS, depende en gran medida de la capacidad de los integradores para seleccionar el componente adecuado [Boeg2006].

Sin embargo, la selección de componentes no es una tarea sencilla de realizar, ya que los integradores han de enfrentarse a la tarea de identificar los componentes y reunir la información sobre ellos que les permitan evaluar su adecuación al sistema. Esta tarea es especialmente compleja teniendo en cuenta la existencia de un gran mercado de componentes OSS en el que generalmente la información acerca de los componentes se encuentra extendida a través de Internet, mostrando un formato desestructurado u ofreciendo información incompleta.

En este contexto, el grupo de Ingeniería del Software de Sistemas de Información (GESSI) de la Universidad Politécnica de Catalunya (UPC) y la Universidad Noruega de Ciencia y Tecnología (NTNU) realizaron un estudio cuyo objetivo fue investigar la práctica industrial en la selección de componentes OSS [Ayalaetal2010].

El estudio se fundamentó en un enfoque empírico cualitativo, con entrevistas semi-estructuradas a 23 empleados de 20 organizaciones que llevaban a cabo procesos de selección de componentes OSS en España, Noruega y Luxemburgo. Los resultados mostraron que la selección de componentes es bastante informal y evidenciaron algunos hechos y problemas prácticos relacionados con los procesos y recursos utilizados para seleccionar los componentes [Ayalaetal2010].

Aunque la comunidad científica ha volcado esfuerzos considerables en el área de selección de componentes, estos esfuerzos se han encauzado principalmente en el establecimiento de criterios de evaluación y métricas para hacer un ranking de componentes que permita elegir aquel o aquellos componentes que mejor satisfagan

los requerimientos del sistema a integrar. Sin embargo, los resultados del estudio citado muestran que estos criterios no son utilizados habitualmente, principalmente porque la información requerida para evaluar componentes usando esos criterios de evaluación y métricas propuestas no es generalmente proporcionada por las comunidades OSS.

Así pues, uno de los principales problemas que dificulta la selección de componentes según los encuestados en el estudio es la falta de información adecuada y disponible por el proveedor del componente [Ayalaetal2010].

Debido a esta problemática, los seleccionadores de componentes utilizan una lista de criterios informales y no documentados. El uso de estos criterios informales depende de la disponibilidad de la información que es necesaria para poder evaluarlos, principalmente por la dificultad para reunir la información necesaria ya que esta información no se encuentra disponible o se presenta de una forma desestructurada y de difícil localización.

1.2. Objetivos

Basado en los estudios realizados por el grupo GESSI y NTNU [Ayalaetal2010], los dos objetivos principales de esta tesis son los siguientes:

- Proveer un framework para facilitar la identificación y consideración de las características que un integrador espera obtener cuando consulta información relacionada con un componente.
- Realizar un estudio para indagar en qué medida la información facilitada por las comunidades OSS cubre los criterios que se utilizan en la práctica industrial de selección de componentes.

En base a estos objetivos la pretensión es comprobar la medida en qué coinciden la “información requerida” por los integradores y la “información facilitada” por los proveedores para la selección de componentes, ya que es vital que coincidan ambos puntos de vista para mejorar la práctica industrial en la selección de componentes.

1.3. Justificación

Actualmente existen una gran cantidad de componentes OSS distribuidos principalmente a través de Internet. Muchos desarrolladores de software adquieren algunos de estos componentes con la finalidad de conseguir ciertas funcionalidades ya implementadas que puedan integrar en su software con la finalidad de reducir costes. Por esto, seleccionar el componente adecuado en un dominio específico resulta una parte vital para asegurar el cumplimiento de los requisitos del cliente.

A partir de esta necesidad, se han desarrollado muchos estudios tratando de evaluar la selección de componentes. Estos estudios se centran principalmente en definir las características funcionales y no funcionales del componente (pero sobretodo se centran en las funcionales) dejando a un lado un factor importante que es la calidad y disponibilidad de la información acerca del componente.

Por este motivo, proporcionar un framework con la información realmente útil para la selección de componentes es una necesidad prioritaria para proporcionar una guía aplicable en el mundo real tanto para integradores como para proveedores de componentes.

De los resultados del estudio de esta tesis podrán beneficiarse:

1. Las comunidades OSS ya que el framework propuesto les ayudará a comprobar si están facilitando la información necesaria por integradores a la hora de seleccionar un componente. De esta manera, conseguirán transmitir mayor confianza a usuarios e integradores que quieran elegir su componente y así conseguir una expansión de su uso.
2. Los investigadores podrán centrarse en buscar métricas más adecuadas para conseguir medir componentes en dominios específicos, ya que nuestro objetivo es proporcionar un framework a un alto nivel, que pueda ser usado en cualquier dominio y que pueda refinarse para conseguir una mayor precisión en un dominio en concreto.
3. Los integradores encontrarán la información necesaria para aplicar las métricas necesarias para la selección de componentes.

1.4. Estructura de la tesis

En el capítulo 1 se ha definido la motivación, los objetivos y la justificación para el desarrollo de la tesis de máster.

En el capítulo 2 se presenta el estado del arte y los trabajos relacionados que proveen los conceptos necesarios para la realización de ésta tesis.

El capítulo 3 describe la metodología utilizada, así como los instrumentos de investigación empleados para la elaboración de este trabajo.

El capítulo 4 describe la elaboración del framework realizado para agrupar la información que necesitan los integradores de manera prioritaria para realizar sus tareas de selección de componentes.

El capítulo 5 presentan los detalles del estudio realizado para comprobar en qué medida los proyectos OSS proveen la información requerida por los integradores para realizar su labor.

En el capítulo 6 se procederá a realizar un análisis de los resultados obtenidos en el estudio realizado.

El capítulo 7 describe la validez y limitaciones del estudio realizado.

Finalmente en el capítulo 8 se presentan las conclusiones finales sobre la tesis y el futuro trabajo.

2. Estado del arte y Trabajos Relacionados

Este capítulo se centra en la explicación del estado del arte y trabajos relacionados que proporcionan la base necesaria para abordar los objetivos de este trabajo.

Este capítulo se divide en los siguientes puntos:

- En la sección 2.1 se presentan los aspectos más relevantes de Open Source Software (OSS), y se aborda el concepto y la historia del movimiento OSS. Seguidamente se abordan los roles existentes en proyectos OSS y en el mercado de componentes, así como modelos de negocio y licencias en OSS.
- En la sección 2.2 se detallan temas relacionados con la calidad del software. En primer lugar se explica la importancia de la calidad en el software y a continuación se presenta una definición de modelo de calidad en software. Seguidamente se abordan los tipos de métricas aplicables a modelos de calidad, seguido de las características principales del modelo de calidad estándar ISO/IEC 9126 cuya estructura será utilizada como base para el modelo de calidad a construir en el contexto de esta tesis. Finalmente, se explica el método de construcción de modelos de calidad llamado IQMC que será utilizado en la construcción del modelo de calidad, producto de esta tesis.

2.1. Open Source Software

En este apartado se va a proceder a abordar el movimiento OSS, dado que el concepto Open Source aparecerá durante todo este trabajo.

2.1.1. Qué es OSS

Open Source Software (OSS) y Free Software (FS) son conceptos que parten de un movimiento inicial, pero que mantienen puntos de vista y objetivos distintos [OS+01]. Para comprender el concepto de OSS, es necesario conocer inicialmente el concepto de FS.

Cuando nos referimos a Free Software, nos referimos al software que respeta las libertades esenciales del usuario [OS+02]. Estas libertades son:

- Libertad de utilizar el software.
- Libertad de hacer cambios en el código.
- Libertad de estudiar el código.
- Libertad de distribuir copias con o sin cambios del software.

El movimiento FS se inició en 1983 por Richard Stallman, un antiguo programador del MIT, cuando anunció el proyecto GNU [OS+03]. Este proyecto se inició con la intención de crear un sistema operativo completo y libre compatible con el sistema operativo UNIX.

En la década de los 80 se desarrolló la Licencia Pública General de GNU (GPL) y se empezaron a distribuir componentes sujetos a esta licencia.

Pero no todos los usuarios y programadores de FS estaban de acuerdo con los objetivos del movimiento FS, esencialmente por la confusión causada por la ambigüedad en el significado del término *free* dado que se puede interpretar como *gratuito* o *libre*.

Fue en 1998 cuando se produjo un hecho importante que dividiría el movimiento FS. En ese año, se anunció la liberación del código fuente del navegador Netscape, produciendo una división entre los seguidores del FS que culminó con la creación de otro movimiento llamado Open Source Initiative (OSI) [OS+04].

Este movimiento de nueva creación decidió substituir el término “free” debido a las confusiones que originaba la interpretación de su significado tal y como se ha mencionado anteriormente.

Aunque una gran parte de OSS es también FS, una de las principales diferencias entre ambos movimientos radica en el hecho de que FS obliga a que el software permanezca “free” siempre, mientras que OSS no rechaza la comercialización del software.

Por último, cabe mencionar que la iniciativa OSI adoptó las mismas pautas que el proyecto GNU/LINUX. Por lo tanto, un componente OSS además de dar acceso al código fuente, cumple con la Open Source Definition [OS+05]. Esta definición establece diez condiciones que se deben cumplir para que un componente pueda ser considerado como OSS. Estas condiciones son las siguientes:

1. *Libre redistribución.* El software ha de poder ser regalado o vendido libremente.
2. *Código fuente.* El código fuente debe estar incluido o, en su defecto, debe poder ser obtenido 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 solo como parches.
5. *Sin discriminación de personas o grupos:* Ninguna persona puede ser rechazada.
6. *Sin discriminación de áreas de iniciativa:* Los usuarios comerciales no pueden ser excluidos.
7. *Distribución de la licencia:* Deben otorgarse 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 solo como parte de una distribución mayor.
9. *La licencia no debe restringir otro software:* La licencia no necesariamente obliga a que algún otro software que sea construido con software de código abierto deba también ser de código abierto.
10. *La licencia debe ser tecnológicamente neutral:* Ninguna disposición de la licencia puede ser basada en una tecnología específica o en un estilo de interfaz. Es decir, no debe requerirse la aceptación de la licencia por medio de un clic del ratón o de otra forma específica del medio de soporte del software.

En el contexto de este trabajo, cuando hablamos de componentes OSS incluimos en este grupo los componentes FS.

2.1.2. Roles en OSS

Para la ejecución de este trabajo, es importante conocer los roles que existen en proyectos OSS debido a la trascendencia de conocer los stakeholders que participan en el contexto del OSS. Por ello en este apartado se describen algunos de estos roles.

Dependiendo del enfoque de análisis, los roles relacionados con el uso y desarrollo de OSS pueden ser diversos. Por ello se van a analizar los roles desde tres perspectivas importantes en el contexto de este trabajo.

Los roles que se presentan son los siguientes:

- Los roles internos desde la perspectiva de las comunidades de desarrollo OSS.
- Los roles industriales, que surgen a partir del uso industrial de componentes OSS o prácticas de desarrollo comúnmente asociadas a comunidades OSS.
- Los roles relacionados con el mercado existente de componentes OSS.

2.1.2.1. Roles en la comunidad OSS

Inicialmente se van a detallar los roles en OSS desde la perspectiva interna en las comunidades OSS. Esta perspectiva recoge todas las personas que participan dentro de la comunidad clasificándolas según la tarea que desempeñan en la comunidad.

Según [Berquist and Ljungberg2001] los roles en OSS desde una perspectiva interna en la comunidad son los siguientes:

- *Propietario de un OSS*: es la persona (o el grupo) que empieza el proyecto y tiene los derechos de este proyecto.
- *Desarrollador principal*: es el grupo de desarrolladores principales que escriben la mayor parte del código fuente y toman las decisiones más importantes en el proyecto open source.
- *Desarrolladores*: es un grupo más amplio que los desarrolladores principales que reparan defectos en el componente.
- *Reportadores de Problemas*: es un grupo muy amplio que se centra en la detección y reporte de problemas en el componente.
- *Compradores del sistema*: son los encargados de realizar pruebas de funcionamiento en el componente.
- *Soporte a usuarios*: esta tarea se realiza principalmente por algunos usuarios voluntarios del producto proporcionando respuestas a las preguntas realizadas por otros usuarios.
- *Usuarios*: son las personas que utilizan el componente. Además de los usuarios individuales, también se consideran las organizaciones o empresas que usan el producto como usuarios. Los integradores comúnmente son parte de este conjunto ya que según un estudio publicado por [bonaccorsi2006], la mayor parte de integradores (ver 2.1.2.2) suelen utilizar componentes OSS sin participar activamente como desarrolladores u otros roles en la comunidad OSS.

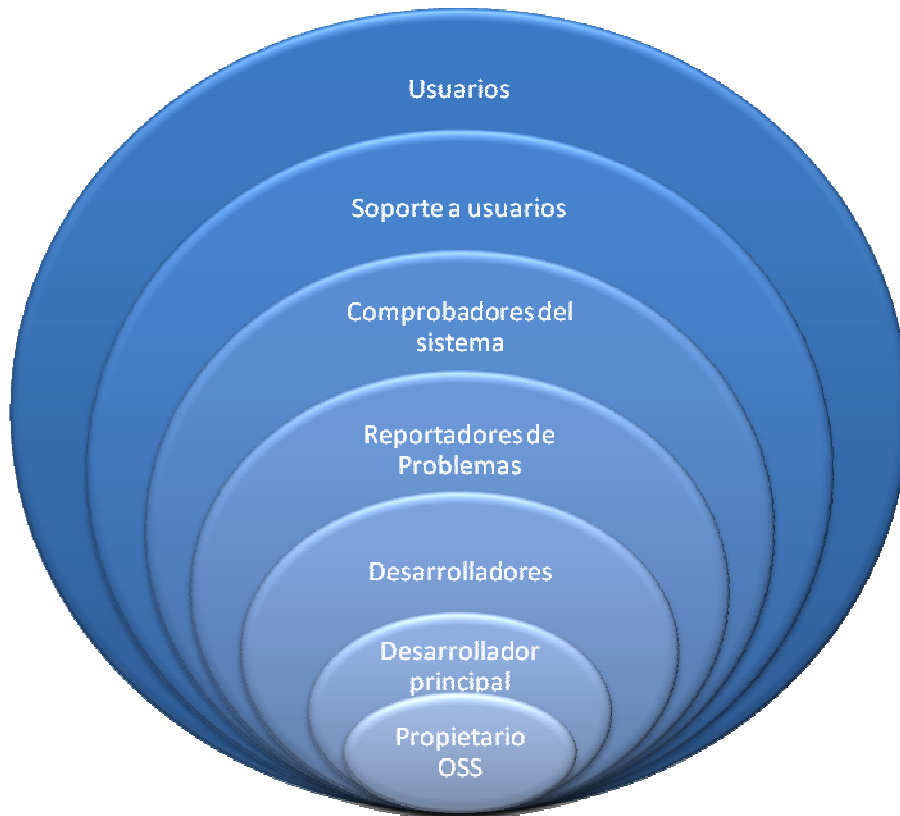


Ilustración 1: Roles des de la perspectiva interna de la comunidad OS

2.1.2.2. Roles Industriales

Si miramos los roles desde una perspectiva industrial, estos roles suelen ser considerablemente distintos. La perspectiva industrial aparece en el momento en el que algunas empresas empiezan a considerar el hecho de re-utilizar componentes OSS con diversos objetivos estratégicos, generalmente para reducir costes en el desarrollo de software.

Según [Haugeetal2007] los roles actuales en OSS desde una perspectiva industrial son los siguientes:

- *Proveedores de OSS*: es la compañía que controla la base del código del producto OSS. Ejemplos de proveedores son: *MySQL*, *Trolltech* y *Sun Microsystems*.
- *Integradores de OSS*: es una compañía que usa el componente en OSS en sus productos, o que construye sus productos encima de una infraestructura OSS.
- *Participantes en comunidades OSS*: es una compañía que interactúa activamente con uno o más proyectos OSS. Ejemplos serían: IBM y SUN que participan el desarrollo de Apache DB.
- *Participantes en comunidades OSS internas*: es una compañía o conjunto de compañías que colaboran en las tareas de desarrollo de software usando prácticas de desarrollo comúnmente asociadas a comunidades OSS.

2.1.2.3. Roles en el mercado OSS

Para finalizar, la última clasificación de roles que se va a explicar son los roles relacionados con el mercado de OSS. En el contexto de este trabajo, nos referimos al mercado OSS como el conjunto de organizaciones virtuales existentes en Internet que incluyen las interacciones de intercambio entre re-usadores y proveedores de componentes OSS, así como las acciones de otros actores que facilitan o promueven tales transacciones [Ayalaetal2009]. Partiendo de esta forma de entender el mercado, según [Ayalaetal2009] los roles desde esta perspectiva son los siguientes:

- *Integrador*: Se refiere a desarrolladores que buscan en el mercado de componentes aquellos que pueden cubrir ciertas funcionalidades requeridas para la integración de un sistema de software.
- *Proveedor*: se refiere a la comunidad OSS o a compañías que desarrollan componentes OSS, como por ejemplo: Openmrs.org, eclipse.org, etc.
- *Mecanismos de búsqueda*: se refieren a mecanismos que permiten la navegación a través del lugar de mercado. Existen herramientas especializadas como *Google Code Search* que desarrollan esta función.
- *Promotores*: se refiere a personas individuales o a organizaciones cuyo principal objetivo es fomentar el movimiento OSS. Como ejemplos tenemos la *Free Software Foundation (FSF)*, *Apache Foundation*, etc.
- *Intermediarios*: se refiere a organizaciones que distribuyen componentes OSS, con o sin ánimo de lucro, como por ejemplo el portal *SourceForge*.

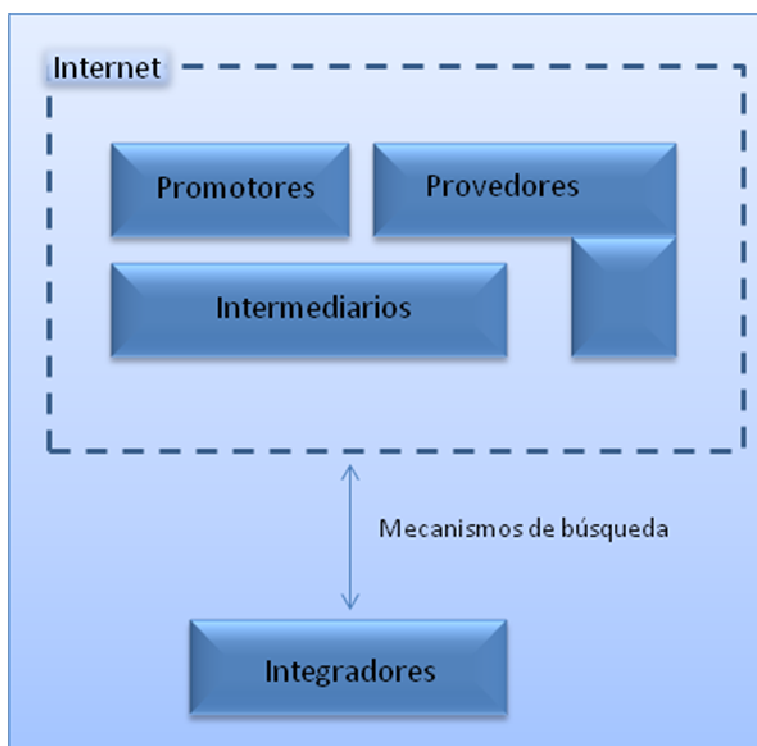


Ilustración 2: Roles des de la perspectiva de mercado OS

Como se puede observar, algunos de los roles son comunes en diferentes puntos de vista, pero en cambio, otros solo aparecen en un único punto de vista. Por ejemplo, el

rol de integrador está relacionado con los tres puntos de vista mencionados anteriormente.

Aunque se ha creído conveniente evidenciar la existencia de otros puntos de vista para clasificar los roles en OSS, en este trabajo nos centramos en el análisis de los roles desde el punto de vista de mercado, ya que nuestro propósito es proporcionar un framework que evidencie las necesidades de los integradores, que deberían ser satisfechas por los proveedores para facilitar el proceso de selección de componentes OSS.

2.1.3. Modelos de negocio y Licencias

Entender el modelo de negocio y de las licencias en OSS es indispensable para entender dicho movimiento y su evolución, así como las diversas interacciones en el mercado actual de OSS. También aprovecharemos la relación existente entre las licencias con el modelo de negocio para explicarlas en este apartado. Dicha relación es inherente dada la interdependencia de las licencias con el modelo de negocio, ya que cuando éste varía, tienden a aparecer nuevas licencias adaptadas a este cambio en el modelo de negocio.

2.1.3.1. Modelos de negocio en el mercado

No cabe duda, que el modelo de negocio seguido por el movimiento OSS y FS ha ido evolucionando a lo largo del tiempo.

Inicialmente, OSS y FS formaban un único movimiento, con la finalidad de proporcionar herramientas software de más calidad que las propietarias, aprovechando el valor que añade el hecho de poder compartir el código entre desarrolladores, sin ánimo de lucro.

Pero actualmente, tal y como explica Fitzgerald [Fitzgerald2006], el movimiento Free and Open Source Software (FOSS), como él lo denomina, ha ido evolucionando a un nuevo movimiento, el OSS 2.0.

En la siguiente tabla se puede observar las principales diferencias entre el FOSS y OSS 2.0 [Fitzgerald2006]:

Proceso	FOSS	OSS 2.0
Desarrollo del Ciclo de Vida	<ul style="list-style-type: none"> • Planificación • Análisis • Diseño- basado en principios de modularización • Implementación <ul style="list-style-type: none"> ○ Código ○ Revisión ○ Pre-commit test ○ Desarrollo de actualizaciones ○ Debugging Paralelo ○ Actualizaciones del producto <p>(normalmente las fases de planificación, análisis y diseño son realizadas por la persona o el grupo a seguir en el bazar)</p>	<ul style="list-style-type: none"> • Planificación- propuesta de estrategias por el mayor número de participantes posible intentando ganar ventaja competitiva • Análisis y diseño- más complejo en la extensión de dominios verticales donde los requisitos de negocio no son universalmente entendidos. • Sub-fases de implementación como en FOSS, pero el total del proceso de desarrollo se vuelve menos como un bazar • Cada vez más, los desarrolladores reciben compensación económica por trabajar en open source.
Dominios del Producto	<ul style="list-style-type: none"> • Infraestructuras horizontales (sistemas operativos, utilidades, compiladores, SGBD, servidores web) 	<ul style="list-style-type: none"> • Más aplicación de sistemas de información en dominios verticales
Estrategias de Negocio Primarias	<ul style="list-style-type: none"> • Añadir valor realizando servicios • Pérdida de líder/ creación de mercado 	<ul style="list-style-type: none"> • Añadir valor realizando servicios <ul style="list-style-type: none"> ○ Bootstrapping • Creación de mercado <ul style="list-style-type: none"> ○ Pérdida de líder ○ Producto y licencias duales ○ Reducción de costes ○ Accesorios • Aprovechar el desarrollo de la comunidad • Aprovechar la rama open source
Soporte al Producto	<ul style="list-style-type: none"> • Bastante irregular- la mayoría basados en listas de correo electrónico o el soporte se proporciona por empresas especializadas 	<ul style="list-style-type: none"> • Los clientes pagarán por soporte profesional
Licencias	<ul style="list-style-type: none"> • GPL, LGPL, Artistic License, BSD, y otras orientadas a MPL 	<ul style="list-style-type: none"> • Muchas más licencias • Uso de términos recíprocos en relación a las licencias

Tabla 1: Diferencias entre FOSS y OSS 2.0

En la Tabla 1, podemos observar la evolución que ha ido sufriendo el movimiento, pero para poder comprender mejor la diferencia entre FOSS y OSS 2.0 debemos comprender el modelo de negocio de ambos movimientos.

En el movimiento FOSS existen varias estrategias de negocio, siendo las dos principales estrategias *value-added-service-enabling* y *loss-leader/market-creating* [Hecker2000; Raymond1999].

Un ejemplo de la estrategia de negocio *value-added-service-enabling* sería un proyecto que proporciona soporte personalizado y no gratuito a usuarios, como realiza por ejemplo Red Hat. En cambio, un ejemplo de la estrategia *loss-leader/market-*

creating sería un proveedor que facilita una parte OSS, pero con el objetivo de que el usuario adquiera la versión Pro del componente, como ofrece por ejemplo *Agilo for Scrum*.

Si miramos estas estrategias de negocio en OSS 2.0, estas se mantienen pero han de ser mucho más matizadas. Por ejemplo la estrategia *value-added-service-enabling* se ha expandido de tal manera, que han llegado a aparecer pequeñas compañías con la finalidad de proveer soporte a un único componente OSS.

Un ejemplo de esta evolución, podemos verla en el caso de MySQL. La comunidad de este componente ofrece licencias anuales de soporte a clientes. Pero MySQL, vende el servicio de prestar dicho soporte a pequeñas empresas que se dedican únicamente a proporcionar soporte para este componente. Como el 90% de las incidencias de MySQL son fáciles de tratar, de esta manera MySQL solo recibe un 10% del total de incidencias. Con esta sub-contratación consiguen dar un mejor soporte a sus clientes, centrándose la comunidad en los casos realmente complicados [Fitzgerald2006].

En el caso de *loss-leader-market-creating*, en OSS 2.0 el énfasis se centra en la creación de mercado a través de un enfoque *loss-leader*. Actualmente, muchas empresas de código propietario están utilizando esta estrategia de negocio, como por ejemplo IBM, que pasó su Eclipse IDE a OSS, cuando el código de la aplicación estaba valorado en \$40 millones. Pero al realizar este cambio, IBM ha obtenido beneficios colaterales, además de mejorar su reputación dentro de la comunidad OSS [Fitzgerald2006].

Otro factor que ha experimentado un cambio en la estrategia de negocio en OSS 2.0 es la demanda por parte de clientes de soporte profesional para componentes OSS. Por este motivo han aparecido nuevos servicios de soporte como certificaciones y entrenamientos, ya que los clientes están dispuestos a pagar por ello.

Pero el cambio producido en la estrategia de negocio también ha afectado a las licencias de los componentes OSS.

2.1.3.2. Licencias OSS

Tanto como la Free Software Foundation y The Open Source Initiative disponen de listas extensas de licencias compatibles con la Free Software Definition [FSF2010] y la Open Souce Definition [OSlicenses2010]. Estas licencias han de cumplir con las reglas citadas en el punto 2.1.1 de este trabajo.

Las licencias en OSS tienen como finalidad proveer al usuario el derecho a realizar lo que desee con el software [Laurent2004].

Estas licencias OSS pueden ser divididas en dos categorías: las que no aplican restricciones en la distribución de trabajos derivados y las que las aplican. Según [Webbink2003], al primer tipo de licencias las denomina *Non-Protective Licenses*, ya que aunque el proveedor mantiene los derechos de autor, no protegen el código de ser utilizado en aplicaciones no OSS. Al segundo tipo de licencias las denomina *Protective Licenses* porque requieren que el código del componente siempre permanezca libre.

Algunos ejemplos de cada tipo son los siguientes:

- *Non-Protective Open Source licenses*: Academic Free License v.1.2; Apache Software License v.1.1; Artistic; Attribution Assurance license; BSD License; Eiffel Forum License; Intel Open Source License for CDSA/CSSM Implementation; MIT License; Open Group Test Suite License; Q Public License v.1.0; Sleepycat License; Sun Industry Standards Source License; University of Illinois/NCSA Open Source License; Vovida Software License v.1.0; W3C Software Notice and License; X.Net, Inc. License; zlib/libpng License; and Zope Public License v.2.0.
- *Protective Open Source licenses*: Apple Public Source License v.1.2; Artistic License; Common Public License v.1.0; GNU General Public License v.2.0; GNU Lesser General Public License v.2.1; IBM Public License v.1.0; Jabber Open Source License v.1.0; MITRE Collaborative Virtual Workspace License; Motosoto Open Source License v.0.9.1; Mozilla Public License v.1.0 and v.1.1; Nethack General Public License; Nokia Open Source License v.1.0a; OCLC Research Public License v.1.0; Open Software License v.1.1; Python License; Python Software Foundation License v.2.1.1; Ricoh Source Code Public License v.1.0; and Sun Public License v.1.0.

La primera licencia creada fue la GNU Public License (GPL) en la década de los 80. La mayoría de proyectos OSS se distribuyen bajo esta licencia, siendo Linux el mejor ejemplo. Esta licencia mantiene el concepto tradicional de restringir el acceso mediante los derechos de autor, garantizando un acceso completo y sin restricciones a cualquier componente OS y a cualquier tipo derivado.

Pero esta licencia requiere que cualquier aplicación que contenga software bajo la licencia GPL también ha de mantener la licencia GPL. Este hecho limitaba el uso de componentes OSS por parte de los desarrolladores de código propietario ya que se veían obligados a aplicar la licencia GPL en su componente. Por este motivo se desarrolló una nueva licencia más permisiva, la Lesser GPL (LGPL). LGPL difiere de la licencia GPL en dos partes: la primera es que está pensada para ser usada con librerías de software; la segunda, en que el software puede estar vinculado con código propietario que se ponga a GPL.

Poco después se creó una nueva licencia que todavía imponía menos restricciones, llamada Licencia BSD. La principal restricción de esta licencia es la obligación de reconocer el trabajo desarrollado por personas anteriormente.

Pero actualmente se ha producido una evolución en el tipo de licencias. [Fitzgerald2006] las clasifica en cuatro tipos:

- *Recíprocas*: es el mismo tipo de licencia que las *Non-protective*. Por ejemplo GPL o LGPL.
- *Estilo académico*: es el mismo tipo de licencias que las *Protective*. Como por ejemplo BSD.
- *Corporativas*: benefician a los intereses corporativos antes que a los de la comunidad desarrolladora de OSS. Normalmente, están basadas en la licencia

Mozilla Public License (MPL). Este tipo de licencia, permite la combinación de código abierto y código propietario, asegurando que los sponsors corporativos mantengan el control del trabajo derivado.

- *No aprobadas por la FSF u OSI*: este tipo de licencia llega a los límites del software propietario ya que intenta acomodar el modelo de código abierto. Dos ejemplos significativos son la *Sun Community Source License (SCSL)* y la familia de licencias *Microsoft Shared Source Initiative*.

La expansión del movimiento OSS está siendo tan grande, que hasta empresas desarrolladoras de código propietario se han visto obligadas a crear nuevos tipos de licencias. Un caso significativo es el de Microsoft, que ha desarrollado tres tipos de licencias:

- *Microsoft Reference License*: permite ver el código fuente, pero no copiarlo ni modificarlo.
- *Microsoft Community License*: basada en MPL, destinada para proyectos colaborativos.
- *Microsoft Permissive License*: similar a la BSD, permite revisar licencias, modificar, redistribuir y trabajar sobre el código sin pagar a Microsoft.

Con todo lo explicado, se observa que cuando la estrategia de negocio del movimiento OSS evoluciona, surge la necesidad de adaptar las licencias a esta nueva evolución.

2.2. La importancia de la Calidad en el Desarrollo de Software

El concepto de calidad resulta ser complejo, por lo cual no existe una definición universal. Calidad puede tener un significado diferente para cada persona, por lo que se trata de un concepto muy subjetivo y dependiente del contexto. En [Basili2004] se corrobora este hecho mediante la presentación del punto de vista de varios ingenieros con respecto del concepto de calidad en el software.

El valor de negocio de un producto de software resulta de la calidad percibida tanto por las personas que adquieren el software como por los usuarios finales. Por eso, la calidad es un atributo crítico del producto software, ya que sin ella se producirá una insatisfacción de los usuarios y pérdidas financieras [Suryan2003].

La calidad en software ha sido uno de los principales objetivos de la ingeniería del software en las últimas décadas. Sin embargo, a pesar del esfuerzo para desarrollar nuevas y mejores técnicas de mejora de procesos software y desarrollar mejores métricas y técnicas de validación del producto, este objetivo permanece difícil de alcanzar [Kitchenham1996].

Para la selección de componentes desarrollados por terceros (CDPT), un factor clave consiste en valorar la calidad del componente. Tradicionalmente los dos aspectos considerados para valorar la calidad han sido:

1. El proceso de desarrollo de software (por ejemplo cuando se adopta un proceso de mejora de software como CMM), que en algunos casos puede ser certificado por un tercero. Pero es bien sabido que dicha certificación en el

proceso de desarrollo de software no necesariamente conlleva a conseguir mejores productos [Voas1998], [Kitchenham1996].

2. La inspección del producto final para medir su conformidad con respecto a algunas características de calidad.

Sin embargo, en la selección de CDPT, estos aspectos tienen que ser cuidadosamente considerados ya que algunas características de calidad en algunos casos, son difíciles de comprobar. Como resultado, la selección de un componente erróneo puede ocasionar el fracaso del proyecto de desarrollo [Vitharana2003].

Por este motivo, en los últimos años varios trabajos se han centrado en la definición de diversos frameworks basados en modelos de calidad para evaluar la calidad en un CDPT.

Sin embargo, estos modelos de calidad no contemplan la problemática de la disponibilidad y calidad de la información requerida para poder ser utilizados. En este contexto, este trabajo se centra en el desarrollo de un modelo de calidad como un instrumento para tratar de definir la calidad de la información provista por los proveedores de componentes.

En los siguientes apartados se abordan aspectos relacionados con modelos de calidad en software. La sección 2.2.1 explica el concepto e importancia de un modelo de calidad en software. La sección 2.2.2 explica la estructura y métricas comúnmente utilizadas en los modelos de calidad. La sección 2.2.3 explica la estructura específica del modelo ISO/IEC 9126, ya que será la estructura usada para construir el modelo de calidad durante esta tesis. Finalmente, la sección 2.2.4 presenta el método IQMC que será utilizado en esta tesis como base para la construcción de un modelo de calidad.

2.2.1. ¿Qué es un modelo de calidad en software?

Una posible definición para modelo de calidad en software es: *“conjunto de características y sub-características y sus relaciones que han de satisfacer un producto software que pertenece a un dominio determinado”* [ISO/IEC14598].

Según [CarvalloThesis], un modelo de calidad en software proporciona los siguientes beneficios:

- Pueden ser usados como base para definir un framework de calidad común.
- Pueden ser adaptados a contextos específicos.
- Proporcionan una base medible para la evaluación de la calidad del software.

Existen varias propuestas de modelo de calidad en software, como por ejemplo Factor-Criteria-Metrics Models (FCM), Goal-Question-Metric (GQM) entre otros. Además, existen algunos estándares como por ejemplo la ISO/IEC 9126 que se explicará en el apartado 2.2.3.

2.2.2 Estructura y métricas en un modelo de calidad

Los modelos de calidad en software comparten una estructura en forma de árbol, compuesta por un conjunto de atributos de calidad de alto nivel que identifican y miden atributos de bajo nivel a los cuales están conectados.

Los modelos de calidad son creados para proveer las bases para la evaluación de software; por lo tanto a los atributos de calidad se les tiene que asignar métricas que permitan su medición.

La *International Vocabulary of Basic and General Terms in Metrology* (VIM) [VIM93] define métrica como “*el conjunto de operaciones que tiene como objetivo determinar un valor de una cantidad*”. En este estándar a una métrica de cantidad se le asigna una unidad, y una medida significa el proceso de medición. La cantidad es definida como el “*atributo de un fenómeno, cuerpo o sustancia que puede ser cualitativamente distinguido y determinado cuantitativamente*”; una unidad es una “*cantidad concreta, definida y aprobada por la convención, con la cual otras cantidades de la misma naturaleza se comparan con el fin de expresar sus magnitudes relativas a la cantidad*”. Finalmente el proceso de medida se define como el “*conjunto de operaciones, descritas específicamente, usadas en la realización de mediciones particulares de acuerdo con un método determinado*”.

Existen diferentes tipos de métricas tal y como se muestra a continuación [FP97]:

- *Nominal*. No hay noción de orden entre clases. Cualquier distintivo numérico o la representación de símbolos son medidas aceptables, pero no hay ninguna noción de una magnitud asociada con el número o símbolo.
- *Ordinal*. Las clases están ordenadas respecto al atributo. Cualquier mapeo que preserve este orden es aceptable.
- *Interval*. Una escala de intervalos preserva orden, como una escala ordinal. Un intervalo conserva las diferencias pero no relaciones. Suma y resta son aceptados en la escala intervalo, pero no la multiplicación y división.
- *Ratio*. Es un mapeo de medidas que preserve orden, el tamaño de los intervalos entre entidades y las relaciones entre entidades. Hay un elemento cero, que representa la ausencia total del atributo.
- *Absolute*. La medida para una escala absoluta se construye simplemente contando el número de elementos de un conjunto de entidades. El atributo siempre toma la forma “*número de apariciones de X en la entidad*”.

Además, existen otro tipo de métricas más simples:

- *Boolean*: para hacer notar la presencia o ausencia de una característica de calidad.
- *Numerical*: para dar un valor numérico al atributo. Por ejemplo puede ser un entero (Integer) o un decimal (Float).
- *Label*: para guardar un nombre en un atributo. Por ejemplo, el nombre de una empresa.

Pero algunos atributos requieren una representación más compleja y precisa, como por ejemplo:

- *Sets*: el atributo guarda una colección de valores.
- *Functions*: el valor del atributo no es absoluto, pero depende de algún otro valor. Un ejemplo de este tipo de métrica sería el tiempo de respuesta del sistema.

Pero tal y como dice Voas [Voas04]: “*no toda medición de software es numéricamente cuantificable. La medición de la calidad en software puede ser subjetiva y no absoluta.*”.

Por ello existen dos tipos de métricas, las *objetivas* y las *subjetivas*. Una *métrica objetiva* dispone de una escala y una unidad de medida bien definida mientras que una *métrica subjetiva* requiere de un criterio del evaluador.

2.2.3 Modelo de calidad: ISO/IEC 9126

El estándar ISO/IEC 9126, actualmente integrado en el estándar SQuaRE [SQU], contiene modelos de calidad y métricas. Algunos conceptos presentados necesitan ser refinados antes de usar el estándar en un proyecto real debido a que son demasiado genéricos.

La ISO/IEC 9126 presenta tres diferentes vistas de calidad. Seguidamente se define cada una de estas vistas:

- La *vista interna* se refiere principalmente a propiedades estáticas de las partes individuales de los productos de software, incluido el diseño, la estructura y la complejidad de los elementos del código.
- La *vista externa* se refiere al software terminado ejecutándose en el equipo de hardware con datos reales. En este punto de vista, los aspectos dinámicos del software juegan un papel importante.
- La *vista de calidad* en uso se refiere al desempeño de usuarios específicos desempeñando tareas específicas con el software en su entorno real. Esta vista mide la productividad y la eficiencia de los usuarios finales.

Estas tres vistas se apoyan mutuamente, la vista interna influye en la externa que a su vez influye en la calidad de uso.

Tal y como se puede observar en la figura siguiente, las vistas internas y externas comparten 6 características y 26 sub-características. En cambio, la vista de calidad en uso tiene sus propias 4 características.

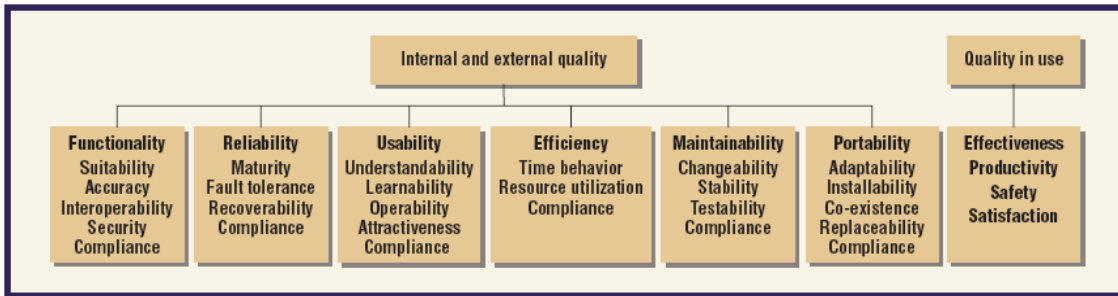


Ilustración 3: Vistas del estándar ISO/IEC 9126

Los principales conceptos y la principal estructura en la ISO/IEC 9126 son los siguientes:

- Una característica de calidad es un conjunto de propiedades de un producto de software por lo que su calidad puede ser descrito y evaluado.
- Una característica puede ser refinada en múltiples niveles de sub-características y no es medible.
- Una sub-característica puede ser subjetivamente medible cuando se requiere, y descompuesta en otras sub-característica o en atributos que ayuden a medirla.
- Un atributo es una propiedad física o abstracta que se puede medir de una entidad. Puede ser básico o derivado
- Un atributo básico es una característica de calidad objetivamente medible que no puede ser descompuesto.
- Un atributo derivado es una característica de calidad medible que puede ser descompuesto en uno o más atributos. A veces no es posible encontrar una métrica objetiva para ellos y se les ha de asignar una métrica subjetiva.
- Al hacer una medida, la medida se asigna a un atributo de una entidad, con una métrica.
- Una métrica es el método de medida definido por la escala de medida.

En la siguiente tabla se enumeran las seis características de calidad definidas en el modelo de calidad ISO/IEC 9621-1 y su descomposición en sub-características.

Characteristic/ Subcharacteristic		Description
1	Functionality	
1	Suitability	Presence and appropriateness of a set of functions for specified tasks
2	Accuracy	Provision of right or agreed results or effects.
3	Interoperability	Capability to the software products to interact with specified systems
4	Security	Prevention to (accidental or deliberated) unauthorized access to data
5	Functionality Compliance	Adherence to application of functionality related standards or conventions
2	Reliability	
1	Maturity	Capacity to avoid failures as a result of faults in the software
2	Fault Tolerance	Ability to maintain a specified level of performance in case of faults
3	Recoverability	Capability to re-establish level of performance after faults
4	Reliability Compliance	Adherence to application of reliability related standards or conventions
3	Usability	
1	Understandability	Effort for recognizing the logical concept and its applicability
2	Learnability	Effort for learning software application
3	Operability	Effort for operation and operation control
4	Attractiveness	Capability of the product to be attractive to the user
5	Usability Compliance	Adherence to application of usability related standards and conventions
4	Efficiency	
1	Time behaviour	Response and processing times; throughput rates
2	Resource Utilization	Amount of resources used and the duration of such use
3	Efficiency Compliance	Adherence to application of efficiency related standards or conventions
5	Maintainability	
1	Analyzability	Identification of deficiencies, failure causes, parts to be modified, etc.
2	Changeability	Capability to enable a specified modification to be implemented
3	Stability	Capability to avoid unexpected effects from modifications
4	Testability	Capability to enable for validating the modified software
5	Maintainability Compliance	Adherence to application of maintainability related standards or conventions
6	Portability	
1	Adaptability	Opportunity for adaptation to different environments
2	Installability	Effort needed to install the software in a specified environment
3	Coexistence	Capability to co-exist with other independent software in a common environment sharing common resources
4	Replaceability	Opportunity and effort for using software in the place of other software
5	Portability Compliance	Adherence to application of portability related standards or conventions

Tabla 2: El modelo de calidad interno/externo ISO/IEC 9126-1

2.2.4 Individual Quality Model Construction (IQMC)

Para la construcción del modelo de calidad producto de esta tesis, se utilizó el método IQMC descrito en este apartado.

El método IQMC ha sido concebido para ayudar en la definición de modelos de calidad de diversos dominios de software siguiendo la estructura del estándar ISO/IEC 9126 descrito anteriormente.

El método IQMC adopta un enfoque de modelo mixto; por una parte se debe escoger un framework de calidad o punto de vista para el análisis del cuál dependerá el catálogo resultante. El catálogo resultante tiene que cumplir con cuatros principios básicos:

- Sólo se fijan algunas características de calidad de alto nivel. Este es un punto crucial, ya que los modelos de calidad pueden variar drásticamente de un dominio a otro, por lo que su refinamiento en características de calidad de un nivel inferior también pueden ser notablemente diferentes.
- Se deberá permitir la creación de jerarquías de características de calidad, ya que es esencial para construir modelos de calidad estructurados.
- Estas jerarquías han de permitir solapamiento, puesto que las características de calidad pueden contribuir a otras de forma diferente.
- Debe ser generalizada. Esto descarta propuestas ad-hoc que puedan parecer atractivas y prometedoras pero que no están siendo utilizadas por la comunidad de ingeniería de software.

El método IQMC proporciona una serie de directrices y técnicas, con la intención de identificar las características de calidad apropiadas para ser incluidas en el modelo de calidad resultante.

El método consta de los siete pasos siguientes:

- Paso 0: Estudio del dominio.
- Paso 1: Determinar sub-características de calidad.
- Paso 2: Definir la jerarquía de sub-características.
- Paso 3: Descomponer sub-características en atributos.
- Paso 4: Descomponer atributos derivados en básicos.
- Paso 5: Establecer relaciones entre características de calidad.
- Paso 6: Determinar métricas para atributos básicos.

Más detalles acerca de este método pueden consultarse en [CarvalloThesis].

3. Metodología

En este capítulo se va a proceder a la explicación en general de la metodología de investigación seguida en esta tesis. En el apartado 3.1 se procede a la descripción de los tipos de investigación existentes seguido de una explicación general sobre la Ingeniería de Software Empírica. En el apartado 3.2 se procede a la explicación de la metodología seguida en esta tesis.

3.1. Métodos de Investigación e Ingeniería de Software Empírica

Existen dos tipos de paradigmas en investigación que utilizan diferentes enfoques empíricos en estudios:

- *La investigación cualitativa*: se ocupa de estudiar los objetos en su entorno natural. Se interesa en el descubrimiento de causas percibidas por los sujetos del estudio y en entender su punto de vista del problema en cuestión.
- *La investigación cuantitativa*: se interesa principalmente en la cuantificación de una relación o en comparar dos o más grupos. El objetivo es identificar una relación causa-efecto.

En el campo de la ingeniería del software empírica, hay tres tipos de estrategias mayoritarias [Robson93] [sjobergetal2007]:

- *Encuesta*: una encuesta se realiza a menudo en la investigación de forma retrospectiva, cuando por ejemplo, una herramienta o técnica ha sido usada durante un periodo de tiempo. El principal medio de obtención de datos cualitativos o cuantitativos son entrevistas o cuestionarios. Estos se realizan tomando muestras representativas de la población que es objeto de estudio. Los resultados de la encuesta se analizan para extraer conclusiones descriptivas o explicativas. Seguidamente se generalizan estas conclusiones hacia la población de donde fue tomada la muestra.
- *Experimentación*: los experimentos se realizan normalmente en un laboratorio que proporciona un alto nivel de control. El objetivo es manipular una o más variables y controlar todas las otras variables fijadas en niveles.
- *Caso de estudio*: un caso de estudio se realiza para investigar un fenómeno contemporáneo en un espacio de tiempo determinado. El investigador reúne información detallada sobre, por ejemplo, uno o varios proyectos durante un período de tiempo determinado. Mientras que la experimentación separa un fenómeno de su contexto y la capacidad de una encuesta para investigar un contexto es limitada, el caso de estudio pretende cubrir las condiciones contextuales.

En el contexto de este trabajo se va a seguir una metodología mixta cualitativa/cuantitativa, utilizando casos de estudios en los que se procederá a la comprobación de la calidad de la información provista por ciertos proveedores de componentes OSS con el fin de cuantificar el porcentaje de información requerida que está actualmente disponible.

3.2. Metodología seguida

Partiendo del estudio realizado por el grupo GESSI y NTNU conocemos qué información es más relevante para los integradores a la hora de seleccionar un componente OSS. Al ser la información que realmente los integradores buscan a la hora de seleccionar un componente OSS, esta información debería ser proporcionada por las comunidades OSS.

Con esta información, desarrollaremos un framework que recoja todas las características relevantes para los integradores a la hora de seleccionar un componente OSS.

Una vez creado este framework, podremos utilizarlo para comprobar si los proyectos existentes actualmente se adaptan realmente a las necesidades de los integradores a la hora de seleccionar un componente OSS. Para ello, seleccionaremos diez proyectos OSS al azar, e investigaremos el portal web del proveedor donde se distribuye el componente, con la finalidad de comprobar si dicha información está disponible en la mayoría de proyectos OSS.

La utilidad de este framework radica en que puede ser utilizado por las comunidades OSS para comprobar si la información demandada por los integradores es provista y decidir si es necesario adjuntarla en su portal para conseguir una mayor satisfacción en los usuarios (integradores). De esta manera, el uso de este framework, puede ayudar a la mejora del proyecto OSS en sí.

Resumiendo, los pasos a seguir en el desarrollo del estudio en esta tesis son:

- Paso 1: Análisis de respuestas de los Integradores. Este paso consiste en analizar y clasificar las respuestas obtenidas del estudio realizado por el grupo GESSI y NTNU.
- Paso 2: Construcción del modelo de calidad. Consiste en construir el modelo de calidad extrayendo características y atributos de las respuestas obtenidas y ya clasificadas. (detallado en el capítulo 4)
- Paso 3: Survey. Consiste en el estudio de diversas comunidades OSS elegidas al azar para analizar si la información requerida por los integradores (incluida en el framework) se encuentra presente en su portal. El objetivo es obtener una medida cuantitativa de la cantidad de información útil que está actualmente disponible.

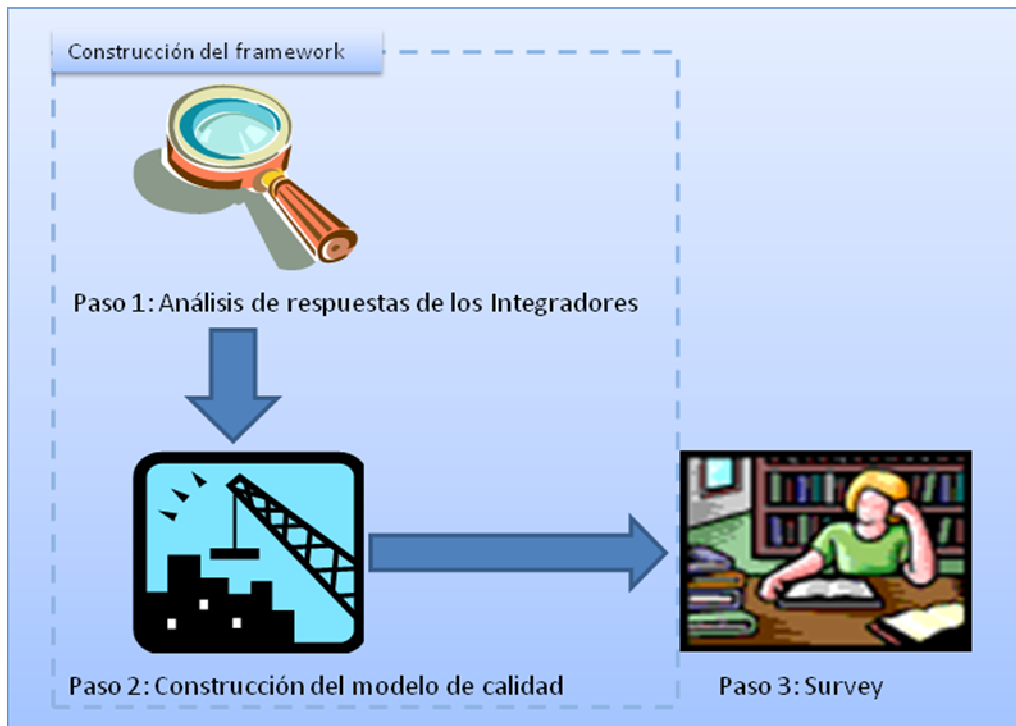


Ilustración 4: Metodología seguida

Los siguientes capítulos detallan el procedimiento y resultados en cada una de los pasos descritos anteriormente.

4. Construcción del Framework

En este capítulo se va a proceder a la explicación detallada de la construcción del framework. La explicación se realizará en tres pasos explicados en el capítulo anterior. En el apartado 4.1 se procede al análisis y clasificación de las respuestas obtenidas del estudio realizado por grupo GESSI y NTNU. En el apartado 4.2 se procederá a la construcción del modelo de calidad.

4.1. Paso 1: Analizar y clasificar las respuestas obtenidas del estudio

La tesis, como ya se ha mencionado anteriormente, se inicia partiendo de una serie de entrevistas realizadas por el grupo GESSI y el grupo NTNU, donde se realizan una serie de preguntas a integradores de CDPT.

Dado que el contenido de las respuestas obtenidas en el estudio es muy diverso, necesitamos analizarlas cuidadosamente y clasificarlas según el sector a que se refiera cada una de ellas.

En la tabla 3 puede observarse las organizaciones a las cuales pertenecían las personas entrevistadas que participaron en el estudio. En esta tabla, se ha reservado el nombre de la organización para mantener su privacidad.

Id Organización	Id Entrevistado	Tipo	Propiedad	Principal Área de Negocio	Personal
A	A	Independiente	Privada	SCC	≈20
B	B	Independiente	Privada	SCC	≈200
C	C	Independiente	Privada	SCC	11
D	D	Independiente	Privada	SCC	63
E	E	Independiente	Privada	SCC	21
F	F	Independiente	Privada	ITD	10
G	G	Independiente	Privada	SCC	≈200
H	H	Independiente	Privada	SCC	≈4
I	I	Filial	Publica	ITD	15
J	J	Independiente	Privada	SCC (principalmente Business Intelligence)	≈200
K	K1, K2	Filial	Publica	ITD	*
L	L	Filial	Privada	SH	*
M	M	Filial	Privada	SCC	≈ 150
N	N	Filial	Privada	SCC	80
O	O1, O2	Filial	Publica/ Privada	SCC	160
P	P	Independiente	Privada	SCC (principalmente Sistemas de Información Geográficos)	40
Q	Q1, Q2	Filial	Privada	SCC	900
R	R	Filial	Publica/ Privada	SCC	60
S	S	Filial	Privada	SH	35,000
T	T	Filial	Publica/ Privada	ESSS (Aplicaciones de negocio)	4

SCC – Software Consultancy, ESSS– Expert Support for Selecting Software Solutions; SH – Software House, ITD – IT Department.

*El personal estaba distribuido geográficamente por lo que el entrevistado no conocía el tamaño

Tabla 3: Organizaciones participantes en el estudio

Se puede observar en la tabla 3, que en tres de las empresas entrevistadas se pudo entrevistar a más de una persona perteneciente a la empresa.

En la tabla 4 se muestra información acerca de estas empresas, como el tipo de proyectos a los que se dedica, componentes utilizados en el proyecto que se estudia y el componente específico que es analizado en el estudio.

Entrevistado	OTS-Based System	Esfuerzo Total	COTS Utilizados	OSS utilizados	Componentes seleccionados
A	Aplicación web para la gestión de una base de datos de estudiantes	480	Oracle	Apache libraries, y Spring framework	Spring web service project
B	Aplicación web para un banco en internet	17,520	Servidor de aplicaciones Websphere	Spring framework, y diversos proyectos comunes	Spring framework
C	Aplicación Windows para asistir a personas disléxicas en la escritura	29,200		LGPL libraries for OpenOffice	Librerías para Open Office
D	Un gestor de contenidos para cubrir los requerimientos de la mayoría de sus clientes	21,900		Varios, tales como: Lucene, Jlog, Spring, Hibernate, readers for text, pdf, Word	Componentes Java Script
E	Un sistema que almacena horas de trabajo, calendarios y citas	14,600		Postgress, Hibernate, Spring, Java Server Faces (JSF), RichPhase, Jasper reports, Facelet, Tomhawk	RichPhase
F	Añadir estadísticas y gráficos a un sistema existente	60		Google Charts API	GoogleCharts
G	Sitio web corporativo que vende objetos en Internet	640		Mambo, OSCommerce, y varios más	Mambo
H	Herramienta web para la gestión de datos personales	1,920		Varios del entorno AJAX.	Hibernate
I	Desarrollo y mantenimiento de un sistema de gestión de recursos	5,840		Varios Php, MySQL, y componentes de Apache	MySQL
J	Sistema Business intelligence para la gestión de campañas de marketing	*	Business Objects, SQL Server 2005, .Net components		SQL Server 2005
K1	Framework para el desarrollo en J2EE	*		Componentes basados en J2EE (e.g., Spring, y Hibernate)	J2EE-based components
K2	Sistema para procesar y gestionar llamadas de emergencia	47,520	.Net components y Oracle		Oracle
L	Actualización de un sistema de gestión de registros	12,000	IBM products como Content Manager, Petition Server	Varios componentes como Formatting Objects Processor (FOP)	IBM Content Manager
M	Campus virtual colaborativo para una escuela de negocios	5,832	SharePoint, Windows Server, SQL server		SharePoint
N	Aplicación web para la gestión de incidencias	*		Varios Debian, MySQL 5, y componentes Apache 2	Debian
O1	Migración de un sistema propietario de gestión de contenidos a una solución OSS	2,280		Plone	Plone
O2	Migración de LotusNotes a una solución basada en OSS	11,520		OpenCoreBusiness	OpenCoreBusiness

P	Web de consultas para la visualización geográfica	6,000	OracleEnterprise solutions for GIS	Varios components del framework J"EE	J2EE-based components
Q1	Un sistema de información web para identificar la disponibilidad de servicios de telecomunicaciones por área	*	ArgiServer	Varios components Googlemaps, Google earth y Hibernate	GoogleMaps
Q2	Un sistema gestor de contenidos con diversas características	1,200	-	Diversos componentes Java	Componentes Java
R	Proyecto web para la colaboración de personas	33,620		MySQL, Java, Php o components Perl	Componentes Java
S	Sistema comercial Data warehouse	*	Commodities	*	*
T	Selección de una solución adecuada para la gestión de ventas, contabilidad y logística de una cadena de tiendas en el aeropuerto	*	Se evaluaron algunas soluciones comerciales	Se evaluaron algunas soluciones OSS	El cliente escogió una solución propietaria

*El entrevistado no conocía la respuesta o se ha mantenido la respuesta confidencial.

Tabla 4: Vista de proyectos y componentes OTS utilizados por los entrevistados

Las entrevistas se realizaron de forma semi-estructurada. Las preguntas realizadas durante la entrevista trataron de extraer información acerca de cómo las organizaciones entrevistadas realizan en la práctica la selección de componentes. De esta manera, el punto de partida de este trabajo son las respuestas que los integradores proporcionaron durante ese estudio.

La primera actividad realizada en esta tesis, consistió en comprender y analizar las respuestas obtenidas así como su contexto. Con tal fin agrupamos las respuestas relacionadas con la misma idea, acción o propiedad.

Para realizar dicha agrupación, utilizamos la técnica de "análisis de contenido" [Krippendorf1980] la cual soporta la creación de categorías a través de la agrupación de sentencias o frases que describen la misma idea, acción o propiedad. Por ejemplo una respuesta que hablaba de la funcionalidad del componente, la agrupamos con las demás respuestas que hablaban de la funcionalidad del componente. Además, se creyó oportuno identificar cada criterio con un código que identifique al criterio para podernos referir a éste sin necesidad de nombrarlo.

Finalmente, se obtuvieron once criterios diferentes que se describen seguidamente:

- *Crit-A:* engloba todas las respuestas de integradores que se centran principalmente en los requisitos del cliente, como por ejemplo la funcionalidad. Un ejemplo típico de las respuestas agrupadas en este criterio es: *"Nosotros comprobamos que el componente cumpla con nuestros requisitos funcionales"*.
- *Crit-B:* engloba todas las respuestas de integradores que se centran principalmente en la estabilidad que proporciona el proveedor al componente y el tiempo en el mercado de éste. Un ejemplo típico de las respuestas agrupadas en este criterio es: *"Siempre tratas de garantizar la estabilidad tecnológica, por lo que tratas de ver si el componente será mantenido por el proveedor. En otras palabras, si el proveedor va a seguir desarrollando el componente"*.

- *Crit-C:* engloba todas las respuestas de integradores que se centran principalmente en el éxito obtenido en el uso del componente en otros entornos, como por ejemplo si otras compañías lo han usado y si la experiencia ha sido positiva. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Si nosotros vemos que el componente funciona correctamente en otros entornos es un factor muy influyente para nosotros, especialmente si vemos o sabemos que el componente está siendo usado por un gran número de compañías conocidas.”*, *“Nosotros intentamos encontrar otra compañía que haya usado el componente y les preguntamos por su experiencia.”*
- *Crit-D:* engloba todas las respuestas de integradores que se centran principalmente en poder obtener soporte tanto durante el proceso de integración del componente como durante todo ciclo de vida del sistema. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Es necesario estar seguro que puedes obtener ayuda de personas durante la integración y durante el ciclo de vida completo del sistema”*.
- *Crit-E:* engloba todas las respuestas de integradores que se centran principalmente en la familiaridad que el grupo de integradores pueda tener con el componente, ya sea desde el lenguaje de programación como si alguno de los miembros del equipo de integradores ya conocen el software. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Nosotros elegimos componentes que tienen el mismo lenguaje de programación que conocemos”, “Normalmente reutilizamos componentes que hemos usado anteriormente.”*
- *Crit-F:* engloba todas las respuestas de integradores que se centran principalmente en que el componente sea fácil de integrar y que se pueda empezar a utilizar rápidamente. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Tiene que ser relativamente fácil empezar a utilizar el componente.”*
- *Crit-G:* engloba todas las respuestas de integradores que se centran principalmente en temas de rendimiento y escalabilidad del componente. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“El rendimiento es importante”*.
- *Crit-H:* engloba todas las respuestas de integradores que se centran principalmente en las licencias a las que está sujeto el componente. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Comprobamos el tipo de licencias y descartamos algunos componentes debido al tipo de licencia al que está sujeto”*.
- *Crit-I:* engloba todas las respuestas de integradores que se centran principalmente en el coste económico de la selección del componente. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“El precio es muy importante para nosotros ya que somos una compañía pequeña”*.
- *Crit-J:* engloba todas las respuestas de integradores que se centran principalmente en la calidad de la documentación ofrecida por los proveedores. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“Comprobamos entre otras cosas la documentación del proveedor”*.
- *Crit-K:* engloba todas las respuestas de integradores que se centran principalmente en la disponibilidad del código fuente. Un ejemplo típico de las respuestas agrupadas en este criterio es: *“La disponibilidad del código fuente*

es importante porque preferimos tener control sobre el código fuente para evitar el miedo de que las cosas salgan mal sin nosotros saber porqué i cual es la causa de ello”.

Una vez identificados los criterios y clasificadas las respuestas, en el gráfico siguiente se puede observar el número de respuestas obtenidas para cada criterio:

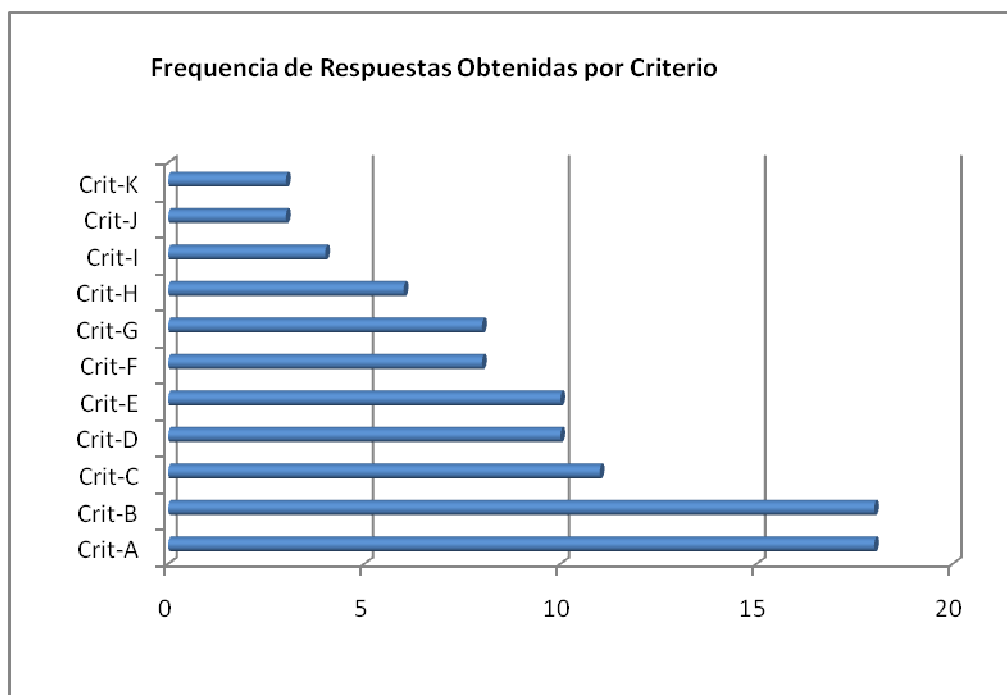


Ilustración 5: Frecuencia de respuestas obtenidas en cada criterio

4.2. Paso 2: Construcción del Modelo de Calidad

Partiendo de la clasificación de respuestas en criterios del punto anterior, hemos creído oportuno la elaboración de un modelo de calidad. Se ha elegido la elaboración de un modelo de calidad porque es un instrumento que nos permite clasificar la información obtenida en las entrevistas realizadas de una forma clara, además de ser una herramienta muy utilizada actualmente.

Nuestro modelo de calidad ha de representar las necesidades de información de los integradores de componentes. Por ello, dicho modelo de calidad pretende ser una herramienta efectiva para los proveedores de componentes para conocer el tipo de información que deberían proporcionar para posicionar su producto. Además, su uso podría favorecer las tareas de selección de componentes llevadas a cabo por los integradores.

Para el desarrollo de este modelo de calidad, se ha utilizado la estructura de la ISO/IEC 9126 ya que este estándar proporciona una serie de conceptos bien definidos y una estructura adecuada para clasificar la información en un modelo de calidad.

Además al ser uno de los estándares más conocidos en la práctica y teoría de la ingeniería del software, se facilita la comprensión del modelo por los stakeholders.

Para la construcción del modelo de calidad se ha seguido la metodología explicada por Carvallo en su tesis doctoral [CarvalloThesis], donde se explica la elaboración de un modelo calidad. Por lo tanto, seguiremos el método IQMC, explicado en el capítulo dos, que proporciona una serie de pautas y técnicas para la identificación de características de calidad adecuadas siguiendo la estructura de la ISO/IEC 9126. Este método, como ya se ha explicado en el apartado 2.2.2, consta de siete pasos:

- Paso 0: Estudio del dominio.
- Paso 1: Determinar sub-características de calidad.
- Paso 2: Definir la jerarquía de sub-características.
- Paso 3: Descomponer sub-características en atributos.
- Paso 4: Descomponer atributos derivados en básicos.
- Paso 5: Establecer relaciones entre características de calidad.
- Paso 6: Determinar métricas para atributos básicos.

En nuestro caso, el catálogo de partida para la construcción de nuestro modelo de calidad son los criterios obtenidos del estudio de las respuestas proporcionadas por los integradores.

Seguidamente aplicamos cada uno de los pasos pertenecientes al método IQMC para la elaboración del modelo de calidad. Inicialmente, en el apartado 4.2.1 se procede a la determinación de las características de alto nivel de nuestro modelo de calidad. En el apartado 4.2.2 se procede a la determinación de sub-características y atributos. Finalmente en el apartado 4.2.3 se procede a la determinación de métricas.

4.2.1. Determinación de Características de Alto Nivel

Nuestro punto de partida se basa en los criterios de información relevantes para que los integradores puedan llevar a cabo su labor de selección de componentes OSS. Es decir, aunque usaremos la estructura de la ISO/IEC 9126, no partiremos de las características de alto nivel que ella proporciona, sino que buscaremos las características de alto nivel propias de nuestro dominio de estudio.

Con esta finalidad, analizamos detenidamente cada criterio y los mapeamos directamente a características de alto nivel, ya que estos criterios son una forma de clasificar a alto nivel la información proporcionada por los integradores. Por lo tanto las características de alto nivel resultantes de nuestro modelo son:

- *Conformidad con los requisitos del cliente*, extraída del Crit-A.
- *Estabilidad de la información del componente*, extraída del Crit-B.
- *Evidencia de éxito de uso del componente*, extraída del Crit-C.
- *Disponibilidad de mecanismos de soporte*, extraída del Crit-D.
- *Experiencia previa con el componente*, extraída del Crit-E.
- *Facilidad de integración del componente*, extraída del Crit-F.
- *Rendimiento*, extraída del Crit-G.
- *Esquemas de licencias*, extraída del Crit-H.

- *Coste*, extraída del Crit-I.
- *Calidad de la documentación*, extraída del Crit-J.
- *Disponibilidad del código fuente*, extraída del Crit-K.

Sin embargo, al realizar un análisis de las características de alto nivel obtenidas, se ha considerado eliminar algunas de ellas por diferentes motivos. Finalmente han sido cuatro las características no consideradas como características de alto nivel del modelo. Los motivos por los cuales se ha tomado dicha decisión se detallan a continuación:

- *Conformidad con los requisitos del cliente*: Los requisitos funcionales son totalmente dependientes del dominio y son difíciles de rehusar en contextos diferentes.
- *Experiencia previa con el componente*: Esta característica describe primordialmente elementos contextuales del equipo integrador y dado que el objetivo del framework a realizar está centrado en la disponibilidad de información por parte del proveedor, esta característica ha sido descartada.
- *Coste*: Al tratarse de componentes OSS, el coste está relacionado principalmente (pero no exclusivamente) con el soporte. Por ello decidimos considerar esta característica a un nivel de abstracción más bajo (como se detallará más adelante) dentro de la característica *Disponibilidad de mecanismos de soporte*.
- *Disponibilidad del código fuente*: Hemos considerado que esta característica la podemos incluir dentro de la característica de *Facilidad de integración del componente* ya que información acerca del código fuente será necesaria en esta característica.

Finalmente, las características de alto nivel consideradas en el framework son las siete siguientes:

- *Estabilidad de la información del componente*.
- *Evidencia de éxito de uso del componente*.
- *Disponibilidad de mecanismos de soporte*.
- *Facilidad de integración del componente*
- *Rendimiento*.
- *Esquemas de licencias*
- *Calidad de la documentación*.

Una vez determinadas las características de alto nivel, debemos avanzar al siguiente paso que consiste en descomponer estas características en sub-características y atributos.

4.2.2. Determinación de Sub-características y Atributos

Analizando cuidadosamente las características de alto nivel obtenidas en el paso anterior, así como las respuestas específicas mencionadas durante las entrevistas, se han determinado ciertas sub-características.

Durante la observación de las respuestas, se buscaron palabras claves como por ejemplo: soporte gratuito, número de actualizaciones, entre otras. Estas palabras claves pueden desvelar sub-características y atributos. Por ejemplo, conocer si el soporte del componente es gratuito, nos desvelaría un atributo ya que a éste, se le puede especificar una métrica que nos permita medir si el soporte es gratuito o no lo es.

Al tratarse el método IQMC de un método iterativo, explicaremos el resultado de la aplicación del método para facilitar la comprensión del modelo de calidad desarrollado. Cabe mencionar que el modelo de calidad desarrollado puede ser modificado y ampliado si se desea obtener un mayor nivel de detalle u obtener mejores métricas.

Para cada característica de nuestro modelo de calidad, encontramos los siguientes sub-características y atributos:

- *Estabilidad de la información del componente:* en esta característica encontramos la sub-característica de *Historia del Producto* y el atributo *Compañías involucradas en la comunidad*. El atributo *Compañías involucradas en la comunidad* engloba a las compañías que contribuyen en la comunidad OSS. La sub-característica *Historia del Producto* engloba los aspectos relacionados con la evolución del producto desde su primera versión. De esta sub-característica, se extrajeron dos atributos: *Versiones del Proyecto* y *Tiempo del Producto en el Mercado*. El primero se centra en las versiones del proyecto y en el tiempo que ha estado cada versión en el mercado; el segundo se centra en el tiempo del componente en sí que ha estado en el mercado.
- *Estabilidad del proyecto:* en esta característica no hemos creído oportuno crear ninguna sub-característica ya que podíamos establecer atributos directamente. En total aparecieron cuatro atributos, de los cuales uno es derivado. El primer atributo encontrado fue el *Número de usuarios registrados* que recoge el número total de usuarios registrados en la comunidad. El segundo atributo encontrado fue *Empresas usando el componente*. Este atributo ha de recoger todas las empresas que usen actualmente el componente. El tercer atributo encontrado fue *Número de descargas*, que ha de recoger el número total de descargas del componente. Finalmente, el atributo derivado hallado fue *Valoraciones de usuarios*. Este atributo ha de recoger todas las valoraciones que los usuarios hayan hecho del producto. Por ello, se ha descompuesto el atributo derivado en un atributo básico, donde este último recoge la valoración personal de cada usuario.
- *Disponibilidad de mecanismos de soporte:* esta característica se ha descompuesto en dos sub-características: *Soporte y Servicios disponibles para usuarios finales* y *Soporte y Servicios disponibles para integradores*. De la primera sub-característica se extrajeron dos atributos: *Servicios de soporte gratuitos para usuarios* y *Servicios de soporte no gratuitos para usuarios*. La función de estos dos atributos es conocer los mecanismos de soporte tanto gratuitos como de pago para usuarios finales de la aplicación. De la segunda sub-característica también se extrajeron dos atributos: *Servicios de soporte*

gratuitos para integradores y Servicios de soporte no gratuitos para integradores. La función de estos atributos es dar a conocer los servicios de soporte tanto gratuitos como de pago para integradores.

- *Facilidad de integración del componente:* esta característica se ha descompuesto en una sub-característica y dos atributos. Los atributos que se extrajeron son *Documentación para integradores* y *Disponibilidad del código fuente y comentarios*, donde el primero nos ha de informar si existe documentación para integradores y el segundo informarnos de la disponibilidad del código fuente, y si este está comentado o no. La sub-característica que se extrajo es *Interoperabilidad*, donde ésta describe la necesidad y capacidad del producto para interactuar con otros sistemas. De esta sub-característica se extrajeron cuatro atributos: *Requisitos de Software*, *Requisitos de Hardware*, *Interoperabilidad con sistemas externos* y *Lenguajes de programación*. El primer atributo ha de proporcionar información acerca de los productos que son requeridos para la correcta funcionalidad del componente. El segundo atributo ha de proporcionar información acerca de los requisitos de Hardware para un funcionamiento correcto del componente. El tercer atributo ha de proporcionar información acerca de los sistemas externos que inter-operan correctamente con el componente. El último atributo ha de proporcionar información acerca de los lenguajes de programación utilizados en el desarrollo del componente.
- *Rendimiento:* de esta característica se extrajeron dos atributos: *Resultados de Rendimiento* y *Resultados de Rendimiento por terceros*. El primero ha de proporcionar información acerca de si se han realizado pruebas de rendimiento por el proveedor del sistema y el segundo de si se alguna otra persona u organización ha realizado alguna prueba de rendimiento del componente.
- *Esquemas de licencias:* de esta característica se han extraído dos atributos: *Licencias OSS* y *Licencias Comerciales*. El primer atributo ha de proporcionar información acerca del tipo de licencias OSS bajo las que se encuentra el componente OSS y el segundo atributo ha de proporcionar información acerca de las licencias comerciales que esté sujeto el componente, si es el caso.
- *Calidad de la documentación:* esta característica se ha descompuesto en dos sub-características y un atributo. El atributo extraído de la característica es *Idiomas de la documentación*, donde ha de proporcionar información acerca de los idiomas en los que se presenta la documentación proporcionada por el proveedor del componente. Las sub-características extraídas han sido: *Manuales* y *Documentación adicional*. La primera ha de proporcionar información acerca de la calidad en que se presentan los manuales por parte del proveedor; la segunda ha de proporcionar información acerca de otro tipo de documentación a parte de los manuales. De la primera sub-característica se han extraído dos atributos: *Documentación del usuario final* y *Documentación de integradores*. El primer atributo descompuesto de la sub-característica *Manuales*, ha de proporcionar información acerca de la calidad de la documentación para el usuario final; el segundo atributo, *Documentación para integradores*, ha de proporcionar información acerca de la calidad de la

información para integradores. De la segunda sub-característica extraída, *Documentación adicional*, se han extraído tres atributos: *FAQs*, *Archivos de ayuda* y *Ayuda online*. El primer atributo ha de proporcionar información sobre si el proveedor proporciona una lista de preguntas frecuentes; el segundo ha de proporcionar información acerca de si se proporcionan archivos de ayuda con el componente; el tercero ha de proporcionar información acerca de si el proveedor proporciona ayuda online.

Llegados a este punto, el último paso que resta para finalizar nuestro modelo de calidad consiste en la asignación de métricas.

4.2.3. Definición de Métricas

Llegados a este punto, el último paso en la construcción de nuestro modelo de calidad es establecer métricas a cada atributo. Este es un punto muy importante ya que hemos de ser capaces de encontrar una métrica adecuada para intentar medir si un proyecto OSS cumple con las expectativas que nosotros creemos importantes.

Las métricas las asignaremos utilizando los tipos de métricas explicadas en el apartado 2.2.1 de esta tesis.

Finalmente, las métricas establecidas para cada atributo se pueden observar en la siguiente tabla donde se muestra nuestro modelo de calidad desarrollado, incluyendo las características, sub-características, atributos y métricas del modelo de calidad que hemos desarrollado:

Característica/Sub-característica/Atributo	Métrica	Descripción
Estabilidad de la información del componente		
Compañías involucradas con la comunidad	Enterprises:Set(Labels: Nominal); Labels=(Nokia, Ericson, ...)	Describe la lista de empresas que participan con la comunidad OSS.
Historia del producto		Aspectos relacionados con la evolución del producto desde su primera versión
Versiones del Producto	Versions: List(<Version: Ordinal, TimeOnMarket: Ratio>); Version=(unknown), TimeOnMarket = Float[Year]	Describe una lista de versiones del producto y su tiempo en el mercado.
Tiempo del Producto en el Mercado	Period: Ratio; Period = Float[Years]	Describe el tiempo que el componente lleva en el mercado.
Evidencia de éxito en el uso del componente		
Usuarios registrados	Users : Integer[Number of registered users]	Describe el número de usuarios registrados en la comunidad.
Empresas utilizando el componente	Enterprises:Set(Labels: Nominal); Labels=(Nokia, Ericson, ...)	Lista de empresas utilizando el componente
Número de Descargas	Downloads = Integer[#downloads]	Número de descargas del componente
Valoraciones de usuarios	Atributo derivado ProductRating = Mean(ProductRatingMarks)	Describe la valoración global del producto basado en los comentarios de usuarios.
Puntuación individual y comentarios por usuario	Rating = List of (User, Comments, ProductRatingMarks =Function (String →4ValueOrder)) User = String, Comments: List of String 4ValueOrder = (Very High, High, Low, Very Low) Estas puntuaciones se obtienen directamente	Describe el resultado de la valoración que el producto ha recibido
Disponibilidad de mecanismos de Servicios de Soporte		
Soporte y Servicios disponibles para usuarios finales		Describe la disponibilidad de la información acerca el mecanismo de soporte ofrecido por el proveedor. Describe los mecanismos de soporte disponibles para usuarios finales

Característica/Sub-característica/Atributo	Métrica	Descripción
<i>Servicios de Soporte Gratuitos para usuarios</i>	<i>FreeUsersSupport: Set(Labels: Nominal); Labels=(FAQ, Forums, Wiki, Chat, Mail, RSS, demos ...)</i>	Describe los mecanismos de soporte disponibles sin coste.
<i>Servicio de Soporte No Gratuitos para usuarios</i>	<i>NonFreeUsersSupport: Set(Labels: Nominal); Labels=(Training, Tutorials ...)</i>	Describe los mecanismos de soporte disponibles con coste.
<i>Soporte y Servicios disponibles para Integradores</i>		Describe los mecanismos de soporte disponibles para integradores
<i>Servicios de soporte gratuitos para Integradores</i>	<i>FreeIntegratorSupport: Set(Labels: Nominal); Labels=(Forums, Wiki, Chat, Mail, RSS ...)</i>	Describe los mecanismos de soporte para integradores sin coste.
<i>Servicios de soporte no gratuitos para Integradores</i>	<i>NonFreeIntegratorsSupport: Set(Labels: Nominal); Labels=(Consulting Services, Wiki ...)</i>	Describe los servicios de soporte no gratuitos para integradores.
Facilidad de integración del componente		Describe aspectos relacionados con la facilidad de integración del componente.
<i>Interoperabilidad</i>		Describe la necesidad y capacidad del producto para interactuar con otros sistemas.
<i>Requisitos de Software</i>	<i>SoftwareRequirements = List of (SoftwareProduct, Comments) SoftwareProduct = String, Comments: List of String</i>	Describe una lista de otros productos Software que son requeridos para la funcionalidad correcta del producto.
<i>Requisitos de Hardware</i>	<i>HardwareRequirements = List of (SoftwareProduct, Comments) SoftwareProduct = String, Comments: List of String</i>	Describe una lista de requisitos de Hardware para un funcionamiento correcto del producto.
<i>Interoperabilidad con Sistemas de Software Externos</i>	<i>SoftwareInteroperability = List of (SoftwareProduct, Comments) SoftwareProduct = String, Comments: List of String</i>	Describe una lista de productos software que funcionan correctamente con el producto.
<i>Lenguajes de Programación</i>	<i>ProgrammingLanguages:Set(Labels: Nominal); Labels=(Java,C, ...)</i>	Lista de lenguajes de programación en uso en la implementación del producto.
<i>Documentación para integradores</i>	<i>Supported: Nominal; Supported = (True, False)</i>	Describe si la está disponible la documentación para integradores.
<i>Disponibilidad del código fuente y comentarios</i>	<i>Code: Nominal Code = (NotProvided, Suitably Commented, Non-suitably commented)</i>	Describe si el código fuente está disponible y comentado.

Característica/Sub-característica/Atributo	Métrica	Descripción
Rendimiento		
Describe información acerca de las propiedades del producto relacionadas con el rendimiento bajo condiciones determinadas		
Resultados de Rendimiento	Supported: Nominal; Supported = (True, False)	Describe la disponibilidad de resultados de realización de test de rendimiento.
Resultados de test de rendimiento por terceros	Supported: Nominal; Supported = (True, False)	Describe la disponibilidad de resultados de rendimiento realizado por terceros sobre el producto.
Esquemas de licencias		
Descripción de las opciones de licencias del componente		
Licencias OSS	OSSLicenses:Set(Labels: Nominal); Labels=(LGPL, GPL ...)	Lista de licencias OSS del producto.
Licencias comerciales	ComLicenses: List of(License, Comments) License =String	Lista de licencias comerciales del producto.
Calidad de la documentación		
Descripción de la calidad de la documentación proporcionada por el proveedor		
Manuales		
Describe la calidad de los manuales provistos por el proveedor.		
Documentación del usuario final	ManualsForFinalUser t: Nominal ManualsForFinalUsers = (NotProvided, Basic, Medium, Advanced)	Describe la calidad de la documentación del usuario final.
Documentación de integradores	ManualsForIntegrators: Nominal ManualsForIntegrators = (NotProvided, Basic, Medium, Advanced)	Describe la calidad de la documentación para integradores.
Documentación adicional		
Describe la calidad de documentación adicional proporcionada por el proveedor.		
FAQs	Content: Nominal; Content=(NotProvided, Basic, Medium, Advanced)things)	Describe si se proporciona una lista de FAQs.
Ficheros de ayuda.	Content: Nominal; Content=(NotProvided, Basic, Medium, Advanced)	Describe si se proporcionan Ficheros de ayuda..
Ayuda Online	Content: Nominal; Content=(NotProvided, Basic, Medium, Advanced)	Describe si se proporciona ayuda online.
Idiomas de la documentación	Languages:Set(Labels: Nominal); Labels=(English, Spanish ...)	Lisa de idiomas de la documentación disponible por el proveedor.

Tabla 5: Modelo de calidad desarrollado

Una vez construido nuestro modelo de calidad, iniciamos el estudio de los proyectos OSS detallado en el capítulo siguiente.

5. Survey

Una vez desarrollado el modelo de calidad, el siguiente paso en esta tesis consiste en realizar una serie de casos de estudio para comprobar la disponibilidad y calidad de la información proporcionada en algunos proyectos OSS. El aspecto más relevante de este estudio es que el framework (i.e., modelo de calidad) utilizado como base para el análisis ha sido construido tomando en cuenta la información relevante para integradores de componentes en la práctica de selección de componentes OSS en la industria.

Con el fin de realizar esta investigación, se seleccionó de forma aleatoria una muestra estratificada de 10 proyectos OSS a ser usados como casos de estudio en el portal *Ohloh.net*. El objetivo de la muestra estratificada fue seleccionar proyectos que cubriesen diferentes ámbitos y tamaños de comunidad. Así pues, se listaron ordenadamente los proyectos existentes en *Ohloh.net* con respecto al número de *usuarios* y *downloads* en Febrero 2010¹, cuyo número de proyectos existentes en dicha fecha ascendía a 437.982 proyectos. Se dividió el listado en 3 estratos que representaban: proyectos con mayor número de usuarios (esto es, proyectos cuya envergadura se puede considerar grande), proyectos con un número de usuarios medio (es decir, proyectos cuya envergadura podría considerarse media), y finalmente proyectos con menor número de usuarios (es decir, proyectos con menor envergadura que los anteriores). Finalmente, se eligieron al azar 4, 3 y 3 proyectos respectivamente en cada uno de los estratos. Los proyectos seleccionados se nombran a continuación:

- Mozilla Firefox
- Agilo
- Fluent NHibernate
- Gimp
- GNU Grub
- Iptables
- Joomla
- MediaCoder
- StatusNet
- Subclipse

La metodología a seguir en este apartado de la tesis será la siguiente:

- Comprobar, siguiendo nuestro modelo de calidad, la calidad y disponibilidad de la información que proporcionan los proveedores de cada uno de los componentes OSS escogidos explorando el portal web de la comunidad.
- Además, se buscará información sobre el componente en páginas webs dedicadas a alojar proyectos OSS como por ejemplo *Ohloh.net* y

¹ Los proyectos fueron seleccionados por un investigador externo (Dr. Daniela Cruzes) en Febrero 2010.

SourceForge.org, para contrastar la información provista en la comunidad del proyecto OSS, con la de estos sitios web.

El estudio de estas comunidades fue realizado durante el mes de Julio de 2010.

5.1. Mozilla Firefox

El primer componente investigado ha sido Mozilla Firefox. Este componente perteneciente a Mozilla Foundation tiene la función principal de proporcionar un navegador web con una calidad igual o superior a cualquier otro navegador web.

Firefox es uno de los navegadores web más utilizados junto con Internet Explorer. Algunas de las características que destacan este navegador son:

- Gran velocidad de navegación.
- Gran seguridad durante la navegación.
- Fácil personalización del componente.

Al tratarse de un gran proyecto ampliamente conocido, las expectativas son de encontrar mucha documentación y con una buena organización.



Ilustración 6: Página principal Firefox

5.1.1. Resultados de la investigación

Sobre la página inicial del componente [MF+01] podemos afirmar que dispone de una excelente presentación, con una interface agradable y con una organización de la información presentada excelente.

En cuanto al contenido, en esta página principal se describen desde las características principales del componente a otras más sofisticadas como el rendimiento, la seguridad, etc., aparte de ofrecer la descarga de la aplicación.

La página del componente se presenta en diversos idiomas, entre los cuales encontramos: español, catalán, inglés, alemán, francés, polaco, chino, entre otros. Además, la web dispone de muchos otros contenidos, como la opción de descargar plugins para Firefox, y de ésta manera personalizar el navegador a gusto del usuario.

A partir de la página inicial intentaremos acceder a la información específica del proyecto que nosotros buscamos. Seguidamente se presenta el modelo de calidad que resume la información obtenida de la exploración del portal.

Característica/Sub-característica/Atributo	Firefox	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	Mozilla Foundation, Mozilla Corporation, Mozilla Europe, Mozilla Japan, Mozilla Online, Mozilla Hispano, Mozilla Italia, Mozdev Community Organization	
<i>Historia del producto</i>		
<i>Versiones del Producto</i>	v1.5, 1 year v2.0, 2 year v3.0, 1 year v3.5, 0,5 year	Información extraída fuera de la comunidad
<i>Tiempo del Producto en el Mercado</i>	6 years	Información extraída fuera de la comunidad
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>		Información no hallada
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>	<i>Puntuación individual y comentarios por usuario</i>	Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Foro, Wiki, Chat	
<i>Servicio de Soporte No Gratuitos para usuarios</i>		Soporte gratuito totalmente
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Foro, Wiki, Chat, Listas de correo electrónico	
<i>Servicios de soporte no gratuitos para Integradores</i>		Soporte totalmente gratuito
Facilidad de integración del componente		
<i>Interoperabilidad</i>		

Característica/Sub-característica/Atributo	Firefox	Observaciones
<i>Requisitos de Software</i>	-Windows, XP Vista o 7 -Linux, no funciona con las librerías GTK+2.10 Glib 2.12 Pango 1.14 X.Org 1.0 o superiores -Mac, Mac OSX 10.4 y posteriores	
<i>Requisitos de Hardware</i>	-Pentium 233MHZ,64MB RAM 52MB espacio disco, Para Windows -Macintosh Intel x86 o PowerPCG3 G4 o G5, para Macintosh	
<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
<i>Lenguajes de Programación</i>	Java, Ajax, HTML, JavaScript, CSS, SVG, XHTML, DOM	
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably Commented	
Rendimiento		
<i>Resultados de Rendimiento</i>	False	
<i>Resultados de test de rendimiento por terceros</i>	False	
Esquemas de licencias		
<i>Licencias OSS</i>	GPL 2.0, LGPL v3, MPL 1.1	
<i>Licencias comerciales</i>		No existen licencias comerciales
Calidad de la documentación		
<i>Manuales</i>		
<i>Documentación del usuario final</i>	Advanced	
<i>Documentación de integradores</i>	Advanced	
<i>Documentación adicional</i>		
<i>FAQs</i>	Advanced	
<i>Ficheros de ayuda.</i>	Advanced	
<i>Ayuda Online</i>	Advanced	
<i>Idiomas de la documentación</i>	Inglés, español, catalán, alemán, polaco, italiano, entre otros.	

Tabla 6: Modelo de calidad del proyecto Mozilla Firefox

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Accediendo al link *Mozilla Foundation* situado en la parte inferior de la página principal [MF+01], siguiendo los links *About Mozilla* y *Organizations* encontramos información acerca del atributo *Compañías involucradas con la comunidad* [MF+02]. En esta página, encontramos que la principal organización que da soporte al proyecto se trata de *Mozilla Foundation*. Esta fundación se describe como híbrida y está dividida en varias organizaciones dando soporte a *Mozilla*:

- *Mozilla Foundation*: apoya la actual comunidad de Mozilla, la supervisión de la estructura de gobierno y gestión de recursos compartidos del grupo. También busca activamente nuevas formas para que la gente alrededor del mundo reconozcan Internet como un recurso público crítico.
- *Mozilla Corporation*: de total propiedad de la Fundación Mozilla, trabaja con la comunidad para desarrollar software que avance los principios de Mozilla. Incluye el navegador *Firefox*, que es ampliamente reconocido como líder del mercado en seguridad, privacidad y localización del lenguaje. Estas características hacen que Internet sea más seguro y accesible.
- *Mozilla Messaging*: desarrolla y promueve productos que la gente puede usar para comunicarse en línea de manera que se alinean con los principios de Mozilla. Esto incluye la creación del cliente de correo *Thunderbird*.
- Organizaciones afiliadas como *Mozilla Europe*, *Mozilla Japan* y *Mozilla Online*.
- Muchas otras organizaciones que no forman parte de *Mozilla Foundation* como *Mozilla Hispano*.

Referente a la sub-característica *Historia del Producto*, cabe destacar que la información referente al atributo *Versiones del Producto* no estaba disponible en la web del proveedor, y se ha extraído de [MF+03].

En cuanto al atributo *Tiempo del Producto en el Mercado*, la información también se ha extraído fuera de la comunidad [MF+03a] ya que en el portal del componente no se ha hallado dicha información. Cabe destacar que se ha tenido en cuenta el tiempo de vida del componente, desde que fue bautizado el componente con el nombre de *Firefox* en 2004.

- *Evidencia de éxito en el uso del componente*

Referente a todos los atributos pertenecientes a esta característica, no se ha hallado información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

En cuanto a los mecanismos de soporte, todo soporte se ofrece de forma gratuita y en formato online tanto para usuarios como para desarrolladores. Al soporte para usuarios es fácilmente accesible a través del menú principal situado en la parte

superior de la página web del componente, donde se puede utilizar un buscador y realizar consultas [MF+04]. Además, se presentan otras opciones de soporte, entre ellas encontramos un foro [MF+05], un chat [MF+06] y una wiki [MF+07]. A todos estos contenidos se puede acceder desde la sección de soporte del menú principal [MF+04].

En cuanto al soporte para integradores, el acceso se presenta de una forma menos visible, debiendo buscar el link *Mozilla Developer Center* [MF+08], situado en el menú inferior de la página web de la comunidad en la columna *Community*, desde el cuál puedes acceder a toda la documentación y soporte para integradores.

- *Facilidad de integración del componente*

Referente a los requisitos de software y hardware, el acceso a estos contenidos resulta más simple, accediéndose siguiendo el link de *Notas de la versión* en la página principal, y seguidamente accediendo al enlace *requisitos del sistema*. Finalmente se accede a [MF+09] donde se especifica la información perteneciente a este atributo.

Sobre el atributo *Interoperabilidad con Sistemas de Software Externos* no se ha hallado información.

En cambio, referente al atributo *Lenguajes de programación*, encontramos que los lenguajes de programación utilizados en el desarrollo del componente según el portal web del componente son: *JavaScript, Ajax, HTML, CSS, NSS, RDF, RSS, XBL, XForms, XML, XSTL, XPath, XUL, SVG, XHTML, DOM* [MF+10]. En cambio, en *Ohloh* [MF+11], los lenguajes de programación que se mencionan son los siguientes: *JavaScript, C++, CSS, XML, HTML, Perl, C, Shell script, Objective-C, Python y Make*. En este punto encontramos una contradicción referente a los lenguajes de programación utilizados en el desarrollo de este componente según el portal oficial del componente y la página en web del componente en *Ohloh*, mantenida por miembros pertenecientes a *Mozilla Firefox*. Esta contradicción puede llevar a una confusión al equipo de integradores a la hora de valorar este atributo.

En cuanto al valor del atributo *Documentación para integradores* se ha considerado "True" debido a que se puede acceder a la documentación para integradores a través de *Mozilla Developer Center* [MF+08], tal y como se ha explicado anteriormente.

Por último, a la información acerca del atributo *Disponibilidad del código fuente y comentarios* se ha accedido a través de la web de desarrolladores [MF+08], donde después de examinar el código se ha comprobado que el código se presenta muy bien comentado [MF+11].

- *Rendimiento*

Referente a esta característica no se ha encontrado ninguna información, por lo que se han considerado los valores de ambos atributos como "False". Únicamente en [MF+13] se habla acerca de que la última versión de *Firefox* (3.6) es más de dos veces más rápida que la versión 3.0 de éste componente. Para dicha medición se menciona el uso de un test *SunSpider* con Windows XP [MF+14].

- *Esquemas de Licencias*

En el caso de este componente, las licencias de este son *GNU GPL 2.0*, *GNU LGPL v2.1* y *MPL 1.1*. El acceso a esta información no se encuentra en la página inicial del componente, pero si puede consultarse en la página web de la comunidad *Mozilla* [MF+15],

- *Calidad de la documentación*

El acceso al contenido de la documentación resulta de fácil acceso tanto para usuarios como para integradores. Se puede acceder a esta documentación de usuarios e integradores desde [MF+04] y [MF+08]. Cabe resaltar que la calidad de la documentación que se proporciona es excelente tanto para usuarios e integradores por lo que se ha considerado el valor de ambos atributos como “*Advanced*”.

Acerca del atributo *FAQs*, se ha considerado como “*Advanced*” ya que se presentan diversas FAQ según el rol desempeñado en la comunidad con gran cantidad de información [MF+16]. El acceso a esta información debe realizarse a través de la página web de *Mozilla*, no desde la sección de *Mozilla Firefox*.

En cuanto al atributo *Ficheros de ayuda*, la información acerca de este atributo puede consultarse desde la sección de soporte para usuarios [MF+04].

Referente al atributo ayuda online, todo el soporte ofrecido por la comunidad se realiza de forma on-line, incluyendo canal de chat IRC, además de foros y una wiki tal y como se ha explicado anteriormente.

Por último, en cuanto al atributo *Idiomas de la documentación*, cabe resaltar que se ofrece documentación en varios idiomas, pudiéndose seleccionar el idioma deseado desde la página web donde se ofrece la documentación, a través de un botón cuya función es realizar dicho cambio de idioma, facilitando de esta forma el acceso a la información en el idioma deseado por el usuario.

5.1.2. Conclusiones de la investigación del proyecto

Firefox es un navegador web extremadamente popular. Dispone de un gran número de usuarios distribuidos por todo el planeta, siendo uno de los navegadores web más utilizados actualmente. Además, dispone de muchas personas participando en el desarrollo del componente, sobretodo en la parte de personalización del aspecto del navegador.

La investigación del portal, no ha sido una tarea complicada de realizar gracias a que la web presenta un formato y una estructura correcta, resultando la navegación fácil y cómoda. Para navegar por la web del componente, se dispone de muchos menús que ayudan a llegar fácilmente a la información que se desea.

El portal de la comunidad dispone de mucha información y documentación, sobre todo para desarrolladores, donde se explican desde las tecnologías utilizadas en la programación del navegador, a futuras modificaciones que van a desarrollarse en el componente. Todo ello redactado de una forma clara y detallada, conteniendo en

algunos casos ejemplos sencillos para facilitar su comprensión. Además, la documentación se presenta en varios idiomas, factor que permite a más personas animarse a utilizar y a contribuir en el componente, ya que el idioma deja de ser una limitación para acceder al componente.

Al proporcionar mucha cantidad de información y documentación, tanto para usuarios como para desarrolladores, cada vez más personas se animan a contribuir en este proyecto.

El nivel de detalle de la documentación presentada y del contenido de este componente, es totalmente correcto. Pero un 22% (6 de 27 atributos) de la información requerida no se ha hallado. Este sería un punto a analizar, ya que por ejemplo, el hecho de conocer que grandes compañías utilizan el componente, da más confianza a los usuarios y a los integradores a la hora de seleccionar un componente aunque ya por su nombre y su historia, *Mozilla Firefox* dé confianza de ser un proyecto serio y sólido.

Uno de los factores más sorprendentes en la comunidad es el hecho de no hallar una forma de colaboración económica con el proyecto, a diferencia de otros proyectos de la misma fundación donde sí resulta posible. La forma de colaborar en la comunidad de Firefox, es en el desarrollo del componente, realizando o traduciendo documentación, participando en foros, etc.

En el caso del soporte proporcionado se ha comprobado en los foros de los cuales dispone el componente que existe una gran actividad de usuarios proporcionándose ayuda entre ellos, ya que el número de publicaciones en el foro crece cada día. Además, el soporte se proporciona en distintos idiomas, facilitando el acceso a éste por personas que únicamente conozcan un idioma y que normalmente no utilizan algunos componentes por no comprender la información facilitada del componente.

La información acerca del rendimiento podría ser un factor clave en el caso de este componente debido a que la mayoría de navegadores proporcionan características similares a éste navegador, siendo esta característica la cual podría diferenciar a este componente del resto. *Mozilla Firefox* es uno de los navegadores más rápidos del mercado, y proporcionando datos y comparaciones sobre su rendimiento con otros navegadores, podría ayudar a diferenciar el componente de otros.

Por último, la valoración global de la estructura y contenido de la información proporcionada por la comunidad es buena, debiendo mejorar algunos aspectos que son importantes tanto para usuarios e integradores, como por ejemplo, proporcionar información sobre el número de descargas realizadas del componente, el número de usuarios registrados en la comunidad o citar a empresas que utilicen el componente, información que proporcionará más confianza a las personas que quieran utilizar el componente.

5.2. Agilo for Scrum

Agilo for Scrum es una herramienta sencilla, basada en la web para apoyar el proceso de *Scrum* [Agilo+01]. Está basado en *Trac*, otro componente OSS con gran éxito que consiste en un sistema generalizado *Ticket Track System*.

Agilo contiene lo básico que una herramienta necesita para apoyar el uso del proceso *Scrum*. Una característica importante del componente es la capacidad de vincular diferentes productos ya que el componente es altamente personalizable y por lo tanto se puede adaptar para cada proceso específico (por ejemplo, gestión multi-proyecto, etc.).

Agilo for Scrum es uno de los proyectos OSS que combina una parte OSS y una versión Pro, significando que una parte del componente se puede usar gratuitamente, pero para adquirir la versión Pro, se requiere el pago de una licencia. Además, proporciona otros servicios, como por ejemplo la realización de cursos para comprender el funcionamiento del componente, servicios de consultoría y de coaching, siendo estos cursos no gratuitos.



Ilustración 7: Página principal Agilo

5.2.1. Resultados de la investigación

Después de investigar profundamente la información proporcionada en la página web del componente, los resultados obtenidos no son los esperados. Siendo *Agilo* un proyecto OSS, el acceso a ciertos contenidos está muy restringido ya que para acceder a cierta información debes de ser un miembro registrado en la comunidad. Seguidamente se muestra la información recopilada según nuestro modelo de calidad:

Característica/Sub-característica/Atributo	Agilo for Scrum	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	Agile42	Es una compañía de consultoría, no se habla de ninguna otra compañía involucrada en la comunidad.
<i>Historia del producto</i>		
<i>Versiones del Producto</i>		Información no hallada
<i>Tiempo del Producto en el Mercado</i>	2 years	Información extraída fuera de la comunidad
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>	ASDIS, ebuddy, be2, Ericsson, DHD24, Hypoport, Princeton Financial Systems y RES Software	
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Blog, Google Group	
<i>Servicio de Soporte No Gratuitos para usuarios</i>	Soporte Profesional, Hosted Agilo, Servicio de Instalación y configuración	Se ocupan del correcto funcionamiento de la aplicación
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Google Group	
<i>Servicios de soporte no gratuitos para Integradores</i>	Servicio de personalización de la aplicación	Desarrollan nuevas funcionalidades a medida
Facilidad de integración del componente		
<i>Interoperabilidad</i>		

Característica/Sub-característica/Atributo	Agilo for Scrum	Observaciones	
<i>Requisitos de Software</i>	Windows	No se especifica ninguna versión en concreto excepto en Trac y Phyton.	
	Mac		
	Linux		
	Phyton 2.4+		
<i>Requisitos de Hardware</i>	Trac 0.11	Información no hallada	
<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada	
<i>Lenguajes de Programación</i>	Phyton, JavaScript	En la web del componente únicamente se habla de Phyton.	
<i>Documentación para integradores</i>	True	Escasa documentación	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably commented	El código fuente se encuentra bien comentado	
Rendimiento			
<i>Resultados de Rendimiento</i>	False		
<i>Resultados de test de rendimiento por terceros</i>	False		
Esquemas de licencias			
<i>Licencias OSS</i>	Apache Software License		
<i>Licencias comerciales</i>	End User License Agreement (EULA)		
Calidad de la documentación			
<i>Manuales</i>	<i>Documentación del usuario final</i>	Medium	Proporcionada dentro del componente
	<i>Documentación de integradores</i>	Basic	Escasa documentación para integradores proporcionada a través de una wiki
<i>Documentación adicional</i>	FAQs	Medium	Difícil acceso
	Ficheros de ayuda.	Medium	Incluidos en la aplicación
	Ayuda Online	Basic	Existe únicamente un grupo de Google para usuarios
<i>Idiomas de la documentación</i>	Inglés	La web si se presenta además en alemán y holandés	

Tabla 7: Modelo de calidad del proyecto Agilo

Seguidamente se procede a la explicación de la información recopilada en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Sobre esta característica únicamente se ha hallado información acerca del grupo *agilo42*, el promotor de este componente. Del resto de atributos no se ha hallado información. Para poder valorar el tiempo del producto del mercado se ha consultado información en [Agilo+02], fuera de la comunidad.

- Evidencia de éxito en el uso del componente

De todos los atributos pertenecientes a esta característica, únicamente se ha obtenido información sobre el atributo *Empresas utilizando el componente*. Se mencionan varias empresas que han adoptado el uso de este componente de forma exitosa. Entre estas compañías encontramos: *ASDIS, ebuddy, be2, Ericsson, DHD24, Hypoport, Princeton Financial Systems y RES Software*. Sobre cada una de estas empresas, se presenta un informe que explica cómo ha afectado positivamente el uso de *Agilo* en cada una de las compañías citadas. Dicha información puede consultarse en [Agilo+03]. El acceso a esta información se ha realizado accediendo al link *References* situado en el menú localizado en la zona superior de la página web del componente.

- Disponibilidad de mecanismos de Servicios de Soporte

Agilo presenta diversos mecanismos de soporte. Referente a los atributos pertenecientes a la sub-característica *Soporte y Servicios disponibles para usuarios finales*, cabe mencionar que se presentan servicios de soporte gratuitos, pero además se ofrecen otros servicios de soporte no gratuitos. Entre los servicios de soporte gratuitos para usuarios encontramos:

- Un blog [Agilo+04], donde se puede consultar información acerca de errores que se detectan en la aplicación.
- Un grupo de Google donde los usuarios pueden realizar cuestiones acerca del componente y obtener ayuda de otros usuarios o desarrolladores pertenecientes a la comunidad [Agilo+05].

El acceso a esta información se ha realizado desde la sección de soporte del portal [Agilo+06], accediéndose desde el menú situado en la parte inferior de la página, accediendo al link *Professional support* en la columna de *Agilo Services*.

En cuanto a los atributos pertenecientes a la sub-característica *Soporte y Servicios disponibles para integradores*, únicamente se presta un servicio de adaptación personalizada de *agilo*, incluyendo el desarrollo de nuevas funcionalidades siendo éste un servicio no gratuito. El acceso a esta información se ha realizado accediendo desde la sección de soporte del componente [Agilo+06], en la pestaña de *Customization Service* tal y como se puede observar en la ilustración 8.



Ilustración 8: Servicios de personalización de Agilo

- Facilidad de integración del componente

Sobre la información obtenida de la sub-característica *Interoperabilidad*, referente al atributo *Requisitos de Software*, en [Agilo+07] se menciona que para el correcto funcionamiento del componente es necesario disponer de *Python*, un lenguaje de programación OSS en su versión 2.4 o posterior, de *Trac* en su versión 0.11, además de disponer un sistema operativo, sin importar la versión de éste ya que *Python* es un lenguaje multiplataforma, permitiendo ser utilizado en cualquier sistema operativo. A esta información se accedió en la sección de descarga del componente. En cuanto al atributo *Lenguajes de Programación* se menciona en el portal que *Agilo* está programado básicamente en *Python*. Pero consultando la misma información en *Ohloh* [Agilo+08], se indican más lenguajes de programación utilizados en el desarrollo del componente. Estos lenguajes son: *JavaScript*, *HTML*, *CSS*, *ClearSilver*, *SQL*, *XML* y *Shell Script*. Referente a los atributos *Requisitos de Hardware* e *Interoperabilidad con Sistemas de Software Externos* no se proporciona ninguna información.

El atributo *Documentación para integradores*, se ha valorado como “*True*” aunque estando disponible dicha documentación, ésta es escasa e incompleta, proporcionando pocos recursos para personas que decidan participar en la comunidad.

Finalmente referente al atributo *Disponibilidad del código fuente y comentarios* el acceso a esta información puede realizarse una vez has realizado un registro en la comunidad. Desde [Agilo+09], accediendo a la descarga del componente, dispones de la opción de descargar el código fuente. Una vez se ha examinado el código, se ha observado que éste se encuentra convenientemente comentado y por ello se ha valorado como “*Suitably commented*”.

- Rendimiento

Referente a esta característica no se ha hallado información.

- Esquemas de licencias

Agilo está bajo la licencia *Apache Software License 2.0* [Agilo+10] en su versión OSS. Su versión Pro se encuentra bajo la *licencia End User License Agreement (EULA)*, la

cual se trata de un contrato entre *Agilo* y el usuario que contrata la versión Pro. A esta información se accede en el momento de realizar la descarga del componente.

- Calidad de la documentación

La información recopilada sobre la sub-característica *Manuales* muestra que la calidad de la información tanto en la documentación para usuarios finales como para integradores no dispone de una calidad adecuada. La documentación para el usuario final, únicamente se proporciona en los archivos de ayuda dentro del componente. En cambio, la documentación para integradores se presenta en [Agilo+11], una wiki donde el contenido de la información es muy incompleto, además de difícil acceso, ya que en la web del componente se presenta un único enlace a esta sección de desarrolladores en la última FAQ tal y como se puede observar en la última FAQ de [Agilo+12].

Sobre la sub-característica *Documentación adicional* si se presenta información de todos sus atributos:

- La información sobre el atributo FAQs puede consultarse en [Agilo+12], donde se puede haber observado que se trata de un contenido con una calidad aceptable. Se puede acceder a esta información desde la página inicial de *Agilo for Scrum*, siguiendo el enlace FAQ situado en la segunda columna *Why Choose Agilo Pro?*.
- Los ficheros de ayuda deben consultarse desde la aplicación, dado que los contenidos disponibles en su antigua web ya no son revisados y actualizados tal y como se menciona en [Agilo+13].
- En el atributo *Ayuda Online*, el único mecanismo de ayuda ofrecido a través de la red es el grupo de Google mencionado anteriormente [Agilo+05]. Al ser un mecanismo algo escaso, se ha valorado el atributo como “*Basic*”.

Finalmente en cuanto al atributo *Idiomas de la documentación*, toda la documentación se presenta únicamente en inglés. Pero el portal web de la comunidad si se presenta además en alemán y holandés.

5.2.2. Conclusiones de la investigación del proyecto

Investigar *Agilo* ha resultado una tarea algo complicada debido a la falta de información proporcionada en el portal de la comunidad, siendo una comunidad OSS más centrada en conseguir que el usuario acceda a su versión comercial.

La página web principal de la comunidad presenta principalmente su versión comercial, intentando conseguir que el usuario adquiera la versión Pro. Además *Agilo* es uno de los componentes OSS que ofrece soporte personalizado, junto con otros servicios como entrenamientos para conocer el funcionamiento de la aplicación, desarrollo personalizado o sesiones de coaching. Aunque estos métodos no son gratuitos si son aceptados dentro del movimiento OSS 2.0, ya que el cliente está dispuesto a pagar por estos servicios [Fitzgerald2006].

Recopilar información en el portal de la comunidad de *Agilo* ha sido una labor complicada ya que la navegación por el portal no resultó ser una tarea cómoda e intuitiva. Por ejemplo, el acceso a la sección de la comunidad dedicada a desarrolladores, fue encontrado por casualidad revisando las FAQ.

La comunidad de *Agilo* debería aportar más información acerca de su componente para conseguir que integradores de componentes OSS puedan evaluar si realmente *Agilo* es el componente que necesitan. Tal y como se presenta el producto, un integrador que no conozca *Agilo* con anterioridad, difícilmente se decidirá por seleccionar este componente debido a la falta de información sobre el producto.

Toda la información para integradores se presenta en una wiki, donde la navegación a través de ella resulta una tarea complicada. Además, la documentación ofrecida está muy incompleta.

Un hecho que evidencia la falta de mantenimiento de la información a través de su portal, es el hecho de dejar de ofrecer documentación de usuario actualizada en la wiki, adjuntando ésta información dentro de su aplicación. Podría darse el caso de necesitar consultar cierta información que no se encuentre presente en los archivos de ayuda de la aplicación, debiendo esperar a que ésta información se añada a una nueva versión del producto.

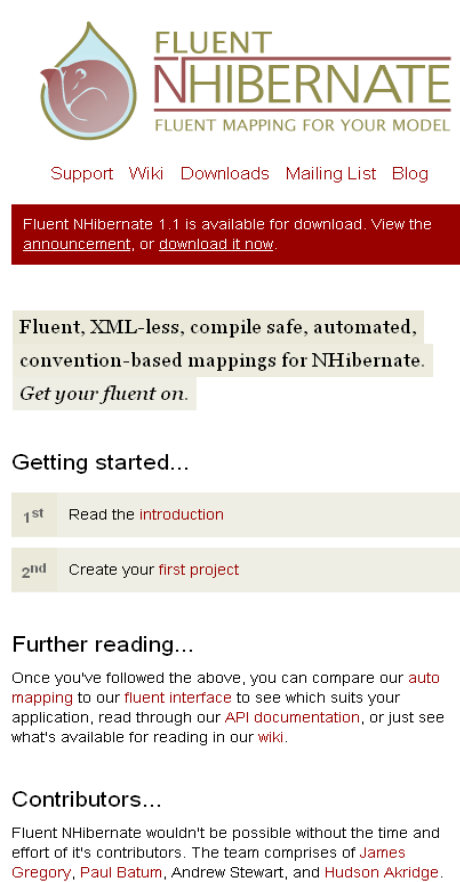
Con todas estas observaciones, *Agilo*, para potenciar su parte OSS debería replantearse su modelo de negocio para llegar a más usuarios. Debería añadir más información que sea relevante para integradores ya que un 22% de la información requerida no se proporciona (6 de 27 atributos) y mejorar la calidad de la información proporcionada.

En general, la valoración de la información proporcionada desde el punto de vista de integradores deja mucho que desear. No proporcionar información acerca de requisitos Hardware necesarios para el buen funcionamiento de un componente de este tipo, resulta una falta grave.

5.3. Fluent NHibernate

Fluent NHibernate es una alternativa al mapeo de archivos XML que proporciona *NHibernate*, pero es totalmente compatible con éste. A diferencia de escribir archivos XML como *NHibernate*, este componente permite escribir mapeos directamente en código C#.

El componente está destinado a desarrolladores ya que es una herramienta para ser utilizada en el desarrollo de software. Por lo tanto, los usuarios de este componente serán principalmente desarrolladores de software que desean utilizar las funcionalidades del componente para desarrollar su propio software.



The screenshot shows the main page of the Fluent NHibernate project. At the top, there is a logo consisting of a stylized water drop with a red and blue gradient, containing a white silhouette of a person. To the right of the logo, the text reads 'FLUENT NHIBERNATE' in a bold, sans-serif font, with 'FLUENT' in green and 'NHIBERNATE' in dark blue. Below this, in a smaller font, it says 'FLUENT MAPPING FOR YOUR MODEL'. Underneath the logo and text, there are navigation links: 'Support', 'Wiki', 'Downloads', 'Mailing List', and 'Blog'. A prominent red banner contains the text: 'Fluent NHibernate 1.1 is available for download. View the announcement, or download it now.' Below the banner, a light green box contains the text: 'Fluent, XML-less, compile safe, automated, convention-based mappings for NHibernate. Get your fluent on.' This is followed by a section titled 'Getting started...' which lists two steps: '1st Read the introduction' and '2nd Create your first project'. Below this is a section titled 'Further reading...' with text: 'Once you've followed the above, you can compare our auto mapping to our fluent interface to see which suits your application, read through our API documentation, or just see what's available for reading in our wiki.' The final section is 'Contributors...' with text: 'Fluent NHibernate wouldn't be possible without the time and effort of it's contributors. The team comprises of James Gregory, Paul Batum, Andrew Stewart, and Hudson Akridge.'

Ilustración 9: Página principal Fluent NHibernate

5.3.1. Resultados de la investigación

En el primer punto de la web se presenta una introducción al proyecto, sobre qué es y cómo funciona el componente. Seguidamente existe un enlace directamente con toda la documentación del componente y a su API. Finalmente menciona a los contribuidores del proyecto.

Veamos los resultados de la recopilación de información sobre el modelo de calidad propuesto en esta tesis:

Característica/Sub-característica/Atributo	Fluent NHibernate	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>		Información no hallada
<i>Historia del producto</i>		
<i>Versiones del Producto</i>	1.0, 2years	Existe la versión 1.0 y la 1.1. La 1.0 aparece en 2008, y la 1.1 en mayo de 2010. Se menciona que el código fuente se está modificando diariamente y se pueden consultar estas modificaciones en su repositorio
<i>Tiempo del Producto en el Mercado</i>	2 years	
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>		Información no hallada
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Wiki, foro, blog, listas de correo electrónico,	
<i>Servicio de Soporte No Gratuitos para usuarios</i>		Todo soporte es gratuito
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Wiki, foro, blog, listas de correo electrónico	
<i>Servicios de soporte no gratuitos para Integradores</i>		Todo soporte es gratuito
Facilidad de integración del componente		
<i>Interoperabilidad</i>		
<i>Requisitos de Software</i>	Ruby	Necesario para la instalación

Característica/Sub-característica/Atributo	Fluent NHibernate	Observaciones
<i>Requisitos de Hardware</i>		Información no hallada
<i>Interoperabilidad con Sistemas de Software Externos</i>	NHibernate 2.0	
<i>Lenguajes de Programación</i>	XML y C#	
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably Commented	
Rendimiento		
<i>Resultados de Rendimiento</i>	False	
<i>Resultados de test de rendimiento por terceros</i>	False	
Esquemas de licencias		
<i>Licencias OSS</i>	BSD License	
<i>Licencias comerciales</i>		No dispone de licencias comerciales
Calidad de la documentación		
<i>Manuales</i>		
<i>Documentación del usuario final</i>	Medium	
<i>Documentación de integradores</i>	Medium	
<i>Documentación adicional</i>		
FAQs	Basic	
Ficheros de ayuda.	Not-provided	
Ayuda Online	Medium	
<i>Idiomas de la documentación</i>	English	

Tabla 8: Modelo de calidad del proyecto Fluent NHibernate

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Referente a esta característica se ha hallado información acerca de las versiones del componente y del tiempo de mercado. Esta información no se proporciona directamente, pero se ha podido deducir ya que el proyecto comenzó en 2008 con la versión 1.0 y hasta mayo de 2010 no apareció la versión 1.1. En [Fluent+01] se menciona que el código fuente es modificado diariamente y para mantenerse actualizado debe ser consultado el repositorio donde se almacena el código periódicamente.

- *Evidencia de éxito en el uso del componente*

Sobre esta característica no se ha hallado ninguna información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

Existen varios servicios de soporte para usuarios proporcionados por la comunidad. Entre ellos encontramos:

- Un blog [Fluent+02] donde se presentan las características de la última versión del componente con ejemplos de nuevas funcionalidades del componente.
- Un foro [Fluent+03] donde existen temas de ayuda, dando la posibilidad de crear nuevos temas en el caso de no encontrar la información deseada en el foro.
- Una wiki, pudiéndose acceder en [Fluent+04], donde se puede observar que esta wiki dispone de poco contenido; en ella se incluye la documentación acerca del componente.
- Listas de correo electrónico dentro de un grupo de Google, pudiéndose acceder en [Fluent+05]; en estas listas de correos participan tanto usuarios finales del componente como desarrolladores.

Todos estos mecanismos de soporte son utilizados tanto por usuarios como integradores, sin haber una separación entre ambos. A esta información se puede acceder desde el menú de la página principal del componente [Fluent+06].

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad* la información proporcionada por el proveedor del componente es la siguiente:

- Sobre el atributo *Requisitos de Software*, la información se proporciona en [Fluent+07], donde se especifica que para generar un ejecutable es necesario disponer de *Ruby*, un lenguaje de programación, aunque se proporciona otra forma de instalación en el caso de no disponer de este lenguaje de programación.
- Referente al atributo *Requisitos de Hardware* no se proporciona información.

- Del atributo *Interoperabilidad con Sistemas de Software Externos*, en [Fluent+08], se especifica únicamente que el componente funciona correctamente con *NHibernate* en su versión 2.0.
- Acerca del atributo *Lenguajes de programación* en la web del proveedor se habla de que el lenguaje de programación es C# aunque en *Ohloh* [Fluent+09], también aparece que parte del código está en XML.

Sobre el atributo *Documentación para integradores*, la información se presenta dentro de la wiki citada anteriormente, por lo que se le ha valorado el atributo como “*True*”.

Por último, el acceso al código fuente es de fácil acceso, dando la opción de ser descargado [Fluent+01]. Comprobando el código se observa que está adecuadamente comentado.

- *Rendimiento*

Sobre esta característica no se ha hallado información.

- Esquemas de licencias

Fluent NHibernate está bajo la licencia BSD [Fluent+10] tal y como se puede observar en la página principal del componente [Fluent+06].

Referente al atributo *Licencias comerciales*, este componente no presenta ninguna versión comercial.

- *Calidad de la documentación*

Referente a la documentación final tanto para usuarios como para integradores, el contenido en si no es muy denso, siendo posible generar mucha más documentación. Toda la documentación de la web está distribuida en los mecanismos de soporte citados anteriormente, por lo cual el acceso a un tema concreto requiere mirar en varios de los mecanismos de soporte. Por ello se ha calificado como “*Medium*” tanto la calidad de la documentación para usuarios como para integradores.

En cuanto a la sub-característica *documentación adicional*, los mecanismos proporcionados son demasiado básicos. Los resultados obtenidos han sido:

- En la FAQ del proveedor [Fluent+11] el número de preguntas que se pueden consultar son escasas, por lo que se ha calificado como “*Basic*” este atributo. Se ha accedido a esta información desde la wiki de la página web del componente [Fluent+04].
- La *Ayuda Online*, aunque se proporciona mayoritariamente por las listas de correo electrónico, en ellas se registra un gran movimiento de usuarios participando en estas listas, por lo que se ha calificado como “*Medium*” este atributo.

Por último, toda la documentación se proporciona únicamente en inglés.

5.3.2. Conclusiones de la investigación del proyecto

Realizar la recopilación de la información en esta comunidad, ha resultado una tarea fluida. El acceso a la información es directo, aunque la cantidad y calidad de información proporcionada es uno de los factores a mejorar por esta comunidad.

Una vez concluida la investigación del proyecto, cabe destacar que el portal web del que dispone la comunidad no es muy extenso en cantidad de información como ya se ha dicho. Por una parte, este factor simplifica la navegación a través del portal pero por otro lado es posible que información relevante para integradores sobre el proyecto no esté incluida, como ha ocurrido en esta investigación donde un 26% de la información recopilada no se ha encontrado (7 de 27 atributos).

Estamos delante de un proyecto destinado a un sector determinado de usuarios, en concreto a usuarios pertenecientes al sector de desarrollo de software. Bien es cierto, que el tipo de personas que acceden a este proyecto son personas con altos conocimientos en informática y que se dedican al desarrollo de software. Por este motivo, el tipo de información proporcionada es más concreta, debido a que la mayoría de desarrolladores de software únicamente consultan información cuando les es necesario, como por ejemplo cuando necesitan solucionar un error de programación que no saben solucionar.

El tipo de documentación proporcionada por esta comunidad es insuficiente aunque para su contexto puede ser válido debido a que mayoritariamente los usuarios e integradores que acceden al portal lo realizan con la finalidad de hallar soporte para solventar un problema determinado. Para ello utilizarán algún mecanismo de soporte intentando obtener ayuda de alguna persona que les pueda proporcionar una solución.

Pero existe cierta información que sería interesante ser proporcionada del componente y que actualmente no se presenta en el portal de la comunidad. Por ejemplo, sería interesante conocer las ventajas respecto al rendimiento que presenta *Fluent NHibernate* respecto a *NHibernate*. Este último punto, podría ser una forma de convencer a más usuarios a elegir entre ambos componentes OSS.

Concluyendo, la información que proporcionen sobre su componente será vital para conseguir que más personas de su sector utilicen su componente y participen en la comunidad.

5.4. Gimp

GIMP (GNU Image Manipulation Program) es herramienta diseñada para trabajar con imágenes digitales con el fin de retocar, componer y crear imágenes.

Comenzó como un programa para *UNIX* desarrollado por *Peter Mattis* y *Spencer Kimball*, pero actualmente se presenta en otras plataformas como *Windows* y *Mac OSX*, entre otras.

Actualmente su interface se presenta en varios idiomas entre ellos: castellano, inglés, catalán, alemán, entre otros. Algunas personas ven a este componente como una alternativa a *Photoshop*, pero en OSS.

Al ser un proyecto que lleva muchos años funcionando, las expectativas en este componente son de recopilar una documentación completa y bien organizada, además de encontrar una gran cantidad de información referente al proyecto. En la ilustración 10 podemos ver la página principal del proyecto.



Ilustración 10: Página principal del proyecto Gimp

5.4.1. Resultados de la investigación

Después de investigar el portal de la comunidad, cabe destacar que se hace presente en la web de la comunidad que se trata de un proyecto sólido, donde el contenido está muy organizado y distribuido, separando el contenido para usuarios del de desarrolladores.

Pero como pasa en la mayoría de proyectos, hay cierta información relevante para integradores que no se proporciona.

Veamos la información recopilada sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	Gimp	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	GNU	
<i>Historia del producto</i>		
<i>Versiones del Producto</i>	v2.6.8, 0.5 years v2.6.9, 0.08 years v2.7.1, 0.05 years v2.6.10, 0.05 years	Duración basada desde la fecha de salida hasta la publicación de la siguiente versión.
<i>Tiempo del Producto en el Mercado</i>	15 years	Información extraída fuera de la web del componente.
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>		Información no hallada
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Listas de correo electrónico, chat	
<i>Servicio de Soporte No Gratuitos para usuarios</i>		Soporte totalmente gratuito
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Listas de correo electrónico, chat	
<i>Servicios de soporte no gratuitos para Integradores</i>		Soporte totalmente gratuito
Facilidad de integración del componente		

Característica/Sub-característica/Atributo	Gimp	Observaciones
Interoperabilidad		
<i>Requisitos de Software</i>	X windowing Layer, Glib 2.16, GTK+ 2.12, babl, GEGL, PangoFT 2.1.18, Fontcodig 2.2.0	Librerías para Mac Librerías para UNIX Es multiplataforma
<i>Requisitos de Hardware</i>	128Mb RAM	En el caso de Windows
<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
<i>Lenguajes de Programación</i>	C, Scheme, HTML, Automake, XML, Python, Perl, C++ , Autoconf, Shell scrip, XSL-Transformation y Modula-2	
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably commented	
Rendimiento		
<i>Resultados de Rendimiento</i>	False	
<i>Resultados de test de rendimiento por terceros</i>	False	
Esquemas de licencias		
<i>Licencias OSS</i>	GNU GPL	
<i>Licencias comerciales</i>		No existe versión comercial
Calidad de la documentación		
Manuales		
<i>Documentación del usuario final</i>	Advanced	
<i>Documentación de integradores</i>	Medium	
Documentación adicional		
<i>FAQs</i>	Advanced	
<i>Ficheros de ayuda.</i>	Advanced	
<i>Ayuda Online</i>	Advanced	
<i>Idiomas de la documentación</i>	Inglés, español, portugués, italiano, alemán, polaco, ruso, noruego y coreano	La documentación para desarrolladores solo en inglés.

Tabla 9: Modelo de calidad del proyecto Gimp

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Sobre esta característica se ha encontrado información acerca de las últimas versiones del producto, de su duración en el mercado y de la compañía involucrada con la comunidad. Para la estimación del tiempo en el mercado de cada versión se ha utilizado como referencia la diferencia entre la fecha de salida de cada versión y la fecha de salida de la versión siguiente. Esta información puede consultarse en la página principal del componente [Gimp+01]. La información acerca de la compañía involucrada en la comunidad, se ha deducido del nombre del componente *GNU Image Manipulation Program*.

La información sobre el tiempo de vida del componente ha sido extraída de [Gimp+02], ya que en la web del componente no se proporciona.

- Evidencia de éxito en el uso del componente

Sobre esta característica no se ha hallado información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

Los mecanismos de soporte tanto para usuarios finales como para integradores son compartidos entre ambos. Entre ellos encontramos listas de correo electrónico [Gimp+03] y un chat en IRC [Gimp+04]. Cabe destacar que existe documentación para desarrolladores y usuarios en secciones distintas de la comunidad, tal y como se explicará más adelante. El acceso a esta información puede realizarse desde el menú situado en la parte inferior de la página web principal de la comunidad [Gimp+01].

- *Facilidad de integración del componente*

Referente al atributo *Requisitos de Software*, nos encontramos delante de un componente multiplataforma, por lo que los únicos requisitos son ciertas librerías dependiendo del sistema operativo que se utilice. Esta información puede consultarse en el menú inferior del portal, escogiendo el sistema operativo en la columna *Downloads*.

Sobre el atributo *Requisitos de Hardware* únicamente en el caso de Windows se especifica que es necesario disponer de 128MB de RAM [Gimp+05]. Sobre el resto de sistemas operativos, únicamente se habla de librerías para su correcto funcionamiento.

Referente al atributo *Lenguajes de Programación*, la información perteneciente al atributo se ha extraído de *Ohloh* [Gimp+06] ya que en el portal del componente no se especifica claramente cuales lenguajes de programación son utilizados para el desarrollo del componente.

La información del atributo *Documentación para integradores* ha sido extraída de [Gimp+07], pudiéndose acceder directamente a esta información desde el enlace *Get Involved* del menú principal del portal del componente situado en la zona lateral derecha.

Por último el acceso al código fuente se ha realizado desde [Gimp+08], pudiéndose consultar el código de forma on-line a través de un FTP. Al observar el código se aprecia un código claro y comentado, lo que hace que resulte fácil de entender. Por ello se ha valorado el atributo como "*Suitably commented*".

- *Rendimiento*

Sobre esta característica no se ha hallado información.

- Esquemas de licencias

Este componente está bajo la licencia GNU General Public License tal y como puede consultarse en [Gimp+09]. Esta información es accesible fácilmente a través del enlace *History* situado en el menú de la parte inferior del portal de la comunidad.

- *Calidad de la documentación*

Los resultados de los atributos pertenecientes a la sub-característica *Manuales* se detallan a continuación:

- La documentación de usuario puede consultarse en [Gimp+10]. Esta documentación se presenta en varios idiomas entre ellos español e inglés. La documentación es muy extensa y explica todas las funcionalidades de la aplicación de una forma clara, relevante y completa. Por ello se ha valorado el atributo como "*Advanced*".
- La documentación para integradores puede consultarse en [Gimp+07], una sección únicamente donde se proporciona contenido para desarrolladores donde el contenido de información es bastante completo pero la estructura, la organización y la presentación de la información puede ser mejorable. Por este motivo se ha calificado el atributo referente a la calidad de la documentación para integradores como "*Medium*".

En cuanto a los resultados de la sub-característica *Documentación adicional* han sido:

- Referente al atributo *FAQs*, en el portal de la comunidad se presentan distintas FAQ para usuarios [Gimp+11] y desarrolladores [Gimp+12] facilitando de ésta forma su consulta dependiendo del rol dentro de la comunidad. Al disponer de varias FAQ y por la calidad en el contenido de éstas, se ha valorado como "*Advanced*" este atributo.
- Los archivos de ayuda pueden descargarse desde [Gimp+13], un servidor FTP. En estos ficheros puede comprobarse la gran cantidad de archivos de ayuda que existen con el componente y su gran calidad en el contenido. Por ello este atributo se ha calificado como "*Advanced*".

- En cuanto a la *Ayuda Online*, todos los mecanismos de soporte son proporcionados de forma online, utilizando listas de correos diferentes según el rol que se desempeña en la comunidad (desarrollador, usuario, entre otros) [Gimp+03] y también un canal de chat en IRC [Gimp+04] tal y como se ha mencionado anteriormente.

Por último, la documentación para usuarios se presenta en diversos idiomas mientras que la documentación para integradores únicamente se presenta en inglés.

5.4.2. Conclusiones de la investigación del proyecto

GIMP se trata de un proyecto sólido formado inicialmente por un grupo formado únicamente por dos personas desarrollando el componente y que actualmente, después de catorce años sigue presente en el mercado OSS presentando nuevas versiones con nuevas funcionalidades en cortos espacios de tiempo.

Un aspecto positivo en la comunidad es la presentación de la información ya que se separa la sección dedicada a la documentación para desarrolladores de la sección dedicada a la documentación para usuarios. De esta forma, la distribución de la información es más correcta y personalizada

En cuanto a los resultados de la investigación, un 26% de la información requerida para este estudio no estaba presente en la comunidad. Como ocurre en la mayoría de proveedores de componentes OSS, se centran mucho más recursos en los mecanismos de soporte que en el desarrollo de documentación. Este hecho provoca que se olviden aspectos relevantes que agrandarían la confianza de los usuarios en el uso del componente, siendo posible conseguir que más usuarios e integradores se decidan a escoger y utilizar su componente. Por ejemplo conocer el número de personas participando en la comunidad, el número de descargas del componente o si existen empresas utilizando el componente, son factores que los integradores valoran a la hora de seleccionar un componente OSS.

Bien es sabido que *GIMP* se trata de un componente pensado para ser utilizado como aplicación de escritorio. Pero la comunidad se olvida de la existencia de personas que desean integrar este componente en el software que desarrollan o desean utilizar ciertas funcionalidades del componente. En este caso, mucha de la información que los integradores consideran relevante no es proporcionada tal y como se puede observar en los resultados de la recopilación de información obtenidos sobre nuestro modelo de calidad.

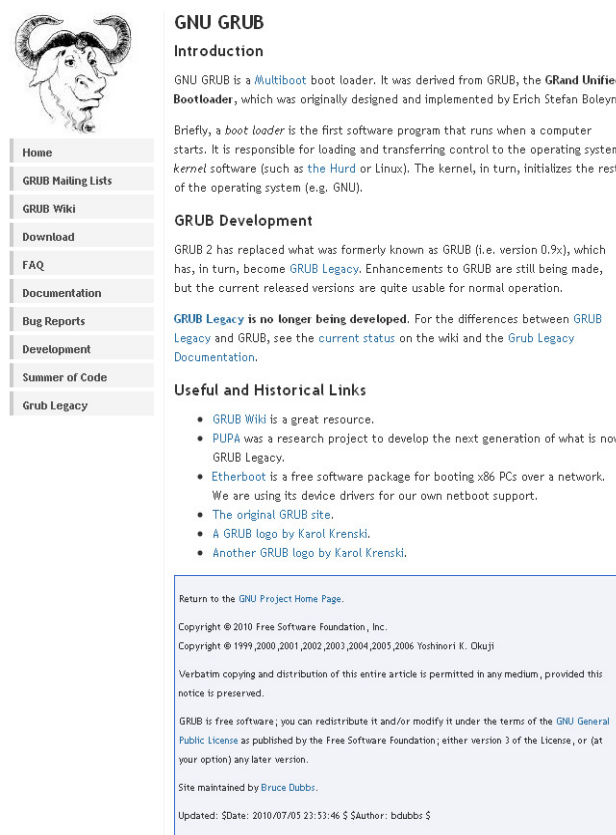
GIMP se trata de un componente con una comunidad importante respaldando el proyecto. Si la comunidad considerase añadir la información que en este trabajo se considera relevante para integradores de componentes OSS, abriría nuevas puertas de expansión al componente, ya que más integradores podrían valorar el componente para posiblemente seleccionarlo e integrarlo con el software que ellos desarrollan. De esta forma, más personas especializadas se unirían a la comunidad, dando al componente un empuje en su crecimiento como componente OSS.

5.5. GNU Grub

GNU Grub se trata de un gestor de arranque múltiple desarrollado por *GNU*, que está derivado de *Grand Unified Bootloader (GRUB)*.

Un gestor de arranque múltiple es el primer software que se ejecuta cuando se enciende un ordenador. Es el responsable de cargar y mantener el control de las operaciones del sistema.

Grub fue desarrollado por Erich Boleyn como una parte de la inicialización del sistema operativo *GNU Hurd* desarrollado por la *Free Software Foundation*. Pero no fue hasta 1999, cuando *Gordon Matzigkeit i Yoshinori K. Okuji* hicieron de *Grub* un paquete oficial de software en el proyecto *GNU*, abriendo el proceso de desarrollo al público.



GNU GRUB

Introduction

GNU GRUB is a **Multiboot** boot loader. It was derived from GRUB, the **Grand Unified Bootloader**, which was originally designed and implemented by Erich Stefan Boleyn.

Briefly, a *boot loader* is the first software program that runs when a computer starts. It is responsible for loading and transferring control to the operating system *kernel* software (such as [the Hurd](#) or Linux). The kernel, in turn, initializes the rest of the operating system (e.g. GNU).

GRUB Development

GRUB 2 has replaced what was formerly known as GRUB (i.e. version 0.9x), which has, in turn, become **GRUB Legacy**. Enhancements to GRUB are still being made, but the current released versions are quite usable for normal operation.

GRUB Legacy is no longer being developed. For the differences between **GRUB Legacy** and GRUB, see the [current status](#) on the wiki and the [Grub Legacy Documentation](#).

Useful and Historical Links

- [GRUB Wiki](#) is a great resource.
- [PIJPA](#) was a research project to develop the next generation of what is now GRUB Legacy.
- [Etherboot](#) is a free software package for booting x86 PCs over a network. We are using its device drivers for our own netboot support.
- [The original GRUB site](#).
- [A GRUB logo](#) by Karol Krenski.
- [Another GRUB logo](#) by Karol Krenski.

Return to the [GNU Project Home Page](#).

Copyright © 2010 Free Software Foundation, Inc.
Copyright © 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006 Yoshinori K. Okuji

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

GRUB is free software; you can redistribute it and/or modify it under the terms of the [GNU General Public License](#) as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Site maintained by Bruce Dubbs.

Updated: \$Date: 2010/07/05 23:53:46 \$ \$Author: bdubbs \$

Ilustración 11: Página principal del proyecto Grub

5.5.1. Resultados de la investigación

Después de investigar la web del proveedor del componente, cabe destacar que la información proporcionada por el proyecto es en parte incompleta. Veamos la información que proporciona del componente sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	GNU Grub	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	GNU	
<i>Historia del producto</i>		
<i>Versiones del Producto</i>	v1.95, 2 years v1.96, 3 years v1.97, 1.5 years	Existe un repositorio con todas las versiones. La estimación del tiempo de mercado de la versión se ha calculado respecto a la versión posterior.
<i>Tiempo del Producto en el Mercado</i>	11 years	
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>		Información no hallada
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Listas de correo electrónico, wiki	Todos los mecanismos de soporte son gratuitos
<i>Servicio de Soporte No Gratuitos para usuarios</i>		
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Listas de correo electrónico, wiki	Todos los mecanismos de soporte son gratuitos
<i>Servicios de soporte no gratuitos para Integradores</i>		
Facilidad de integración del componente		
<i>Interoperabilidad</i>		

Característica/Sub-característica/Atributo		GNU Grub	Observaciones
	<i>Requisitos de Software</i>	UNIX	Si se usa una versión UNIX-OS es necesario binutils-2.9.1.0.23 o posterior
	<i>Requisitos de Hardware</i>		Información no hallada
	<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
	<i>Lenguajes de Programación</i>	C, C++, Python, Ruby	
<i>Documentación para integradores</i>		True	
<i>Disponibilidad del código fuente y comentarios</i>		Suitably Commented	
Rendimiento			
<i>Resultados de Rendimiento</i>			Información no hallada
<i>Resultados de test de rendimiento por terceros</i>			Información no hallada
Esquemas de licencias			
<i>Licencias OSS</i>		GNU GPL	
<i>Licencias comerciales</i>			Se considera Free Software, no existe versión comercial
Calidad de la documentación			
<i>Manuales</i>	<i>Documentación del usuario final</i>	Advanced	
	<i>Documentación de integradores</i>	Advanced	
<i>Documentación adicional</i>	FAQs	Basic	
	Ficheros de ayuda.	Not-Provided	Información no hallada
	Ayuda Online	Medium	
<i>Idiomas de la documentación</i>		Inglés	

Tabla 10: Modelo de calidad del proyecto Grub

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos:

- *Estabilidad de la información del componente*

Sobre el atributo *Compañías involucradas en la comunidad* cabe destacar que el grupo GNU es quien da respaldo a este componente tal y como el nombre del componente indica.

En cuanto a la sub-característica *Historia del Producto*, encontramos información acerca de sus dos atributos:

- Por un lado, la información sobre el atributo *Versiones del Producto* se ha extraído de [Grub+02], siendo un FTP donde se presenta un listado de un gran número de versiones del componente. Para el cálculo del tiempo de mercado de cada versión se ha utilizado la diferencia entre la fecha de última modificación de una versión concreta y la fecha de última modificación de su predecesora.
- Por otro lado, sobre el atributo *Tiempo del Producto en el Mercado* se ha utilizado la información hallada en [Grub+03] donde se presenta la historia de *GNU Grub*, entre otra información, utilizando la fecha en la cual se integró *Grub* en el proyecto GNU como inicio de vida del componente.

El acceso a esta información no resulta complicado accediendo desde la sección *Documentation*, situado el enlace en el menú principal de la página principal del componente [Grub+01].

- *Evidencia de éxito en el uso del componente*

Referente a esta característica y a sus sub-características no se ha hallado información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

Entre los mecanismos de soporte, no se hace diferencia entre los mecanismos destinados a usuarios a los destinados a integradores. Los mecanismos de soporte proporcionados son totalmente gratuitos. Por un lado existen listas de correos [Grub+04], en la cual existen distintos tipos de listas de correos según si necesitas ayuda con *Grub*, si has detectado un error en el componente, o si deseas estar al día sobre la necesidad de nuevas funcionalidades que aparecen en el componente. Por otro lado, se proporciona una wiki [Grub+05], donde puede consultarse información proporcionada por otros miembros de la comunidad o crear una propia página dentro de la wiki. En esta wiki se incluyen contenidos de ayuda [Grub+06] donde se dispone de acceso a explicaciones sobre cómo funciona la wiki, entre otras.

El acceso a toda esta información se realiza fácilmente desde el menú principal de la página del componente donde se proporcionan enlaces directos a la wiki y a las listas de correo.

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad* únicamente se ha encontrado información sobre los dos atributos que se mencionan a continuación:

- *Requisitos de Software* del cual se ha obtenido en [Grub+03] información acerca del atributo, accediendo a esta información a través del enlace *Documentation* situado en el menú principal de la página web del componente. Esta información se encuentra en el apartado de instalación del componente, resultando estar bastante escondida.
- *Lenguajes de programación* del cual ha extraído información en *Ohloh* [Grub+07] ya que no se ha hallado información sobre este atributo en la página web del componente.

Referente al atributo *Documentación para integradores*, esta información se encuentra entre el contenido de la wiki del componente [Grub+05] y en el manual de usuario del componente [Grub+03] por lo que su calificación es “True”.

En cuanto al atributo *Disponibilidad del código fuente y comentarios*, éste puede descargarse desde un ftp [Grub+08]. El acceso al código resulta sencillo ya que se descarga con el componente. Si observamos el código se puede observar un código bien comentado que resulta fácil de leer y comprender por lo que se ha calificado el atributo como “*Suitably Commented*”.

- *Rendimiento*

Referente a esta característica y a sus atributos no se ha hallado información.

- *Esquemas de licencias*

Al ser un componente de *GNU*, únicamente se presenta bajo la licencia *GNU General Public License*, pudiéndose consultar esta información en el recuadro inferior de la página principal del componente como se muestra en la ilustración 12.

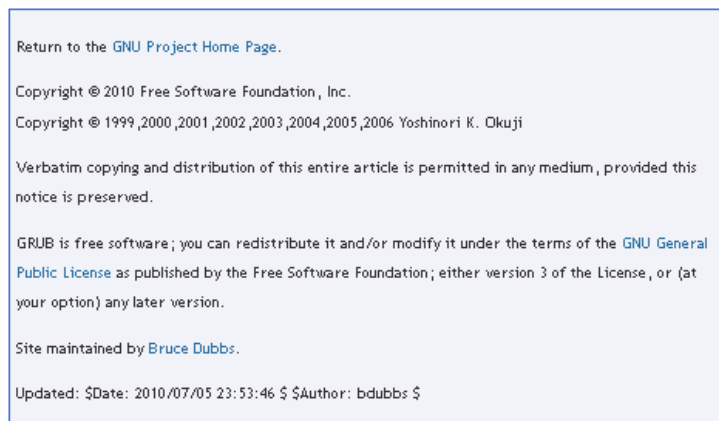


Ilustración 12: Información acerca de Grub

- *Calidad de la documentación*

Referente a los atributos *Documentación del usuario final* y *Documentación para integradores* pertenecientes a la sub-característica *Manuales*, como se ha dicho anteriormente, esta información se presenta tanto en la wiki [Grub+05] como en el manual donde se explica el funcionamiento de la aplicación [Grub+03]. En el caso del manual, éste se presenta en varios formatos con una buena estructura y una información completa sobre el funcionamiento del componente. En el caso de la wiki, el funcionamiento resulta algo incómodo, ya que la navegación resulta algo complicada, pero el contenido de la información es muy correcto. A destacar que el manual es más destinado al usuario final, aunque el integrador puede encontrar información interesante sobre el funcionamiento de la aplicación. Se ha considerado como “*Medium*” la documentación para integradores ya que la mayor parte de la información se presenta dentro de la wiki, resultando una tarea algo complicada acceder a ella. En cambio el manual de usuario está muy completo por lo que se ha considerado “*Advanced*”.

Sobre el atributo *FAQs* se ha considerado como “*Basic*” ya que solo se presentan dos preguntas tal y como se puede observar en [Grub+09].

Referente al atributo *Ayuda online*, se ha clasificado como “*Medium*” pese a que todo soporte proporcionado es de forma electrónica, éste podría ser mejorable añadiéndose algún otro tipo de soporte online además de las listas de correos.

Finalmente, en cuanto al atributo *Idiomas de la documentación* cabe destacar que únicamente se ofrece documentación en inglés.

5.5.2. Conclusiones de la investigación del proyecto

Grub se trata un proyecto con una vida superior a diez años, centrado en un sector software muy específico, los gestores de arranque.

La página web del componente presenta una estructura muy similar a otros componentes ya estudiados en este trabajo. Inicialmente se habla un poco de la funcionalidad de la aplicación y se da acceso a contenidos de la web como la documentación, la wiki, etc.

Centrándonos en los resultados obtenidos, un 33% de la información considerada relevante por integradores no se proporciona. Entre esta información está la referente a la característica *Evidencia de éxito en el uso del componente*, en la cual es posible valorar si el componente es usado. Por ello, una persona que desconozca el componente por completo, puede tener dudas acerca de si realmente el componente es el adecuado ya que los integradores dan mucha importancia a las opiniones realizadas por otros usuarios que hayan utilizado el componente. Además, al centrarse en un sector del software tan específico puede llevar a pensar a que el número de usuarios de la comunidad no sea muy elevado. Al no proporcionar esta información, el integrador puede desconfiar del componente, aunque sea *GNU* el grupo detrás del componente.

Grub se encuentra bajo la licencia *GPL*, obligando a mantener dicha licencia en cualquier nueva versión desarrollada con parte del componente sujeto a esta licencia. A los integradores normalmente no les interesan los componentes con este tipo de licencia.

En cuanto a la documentación ofrecida, como ya se ha dicho anteriormente la parte de usuario es bastante completa. En cuanto a la documentación para integradores, ésta es mejorable. Se podría incluir más información acerca del desarrollo del componente como por ejemplo adjuntar explicaciones acerca de las tecnologías utilizadas en el desarrollo del componente. Esta información puede ser de gran ayuda a personas que quieran contribuir con la comunidad, por ejemplo, en el momento de comenzar a modificar código, ya que posiblemente no dispongan de conocimientos previos con los lenguajes de programación utilizados. Pero normalmente, este aspecto se obvia en este tipo de componentes ya que el desarrollo de documentación es bastante costoso, y se llega a la conclusión de utilizar los recursos para desarrollar otras tareas como por ejemplo, proporcionar un mejor soporte. Normalmente son proyectos de gran envergadura que abarcan un gran número de usuarios los que proporcionan este tipo de documentación.

Además, proporcionar documentación únicamente en un idioma limita bastante el acceso a personas que desconozcan dicho idioma.

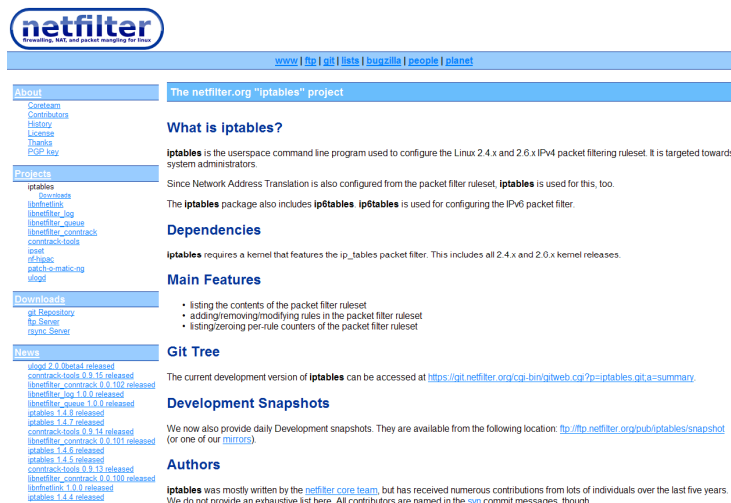
Como conclusión, *Grub* tiene varios aspectos que mejorar referente a la información proporcionada, ya que actualmente la información que proporciona es insuficiente para aplicar un método de selección de componentes.

5.6. Iptables

Iptables pertenece a *Netfilter*, un proyecto que se encarga de ofrecer herramientas OSS para cortafuegos basados en Linux. Actualmente *iptables* es el componente más popular construido sobre *Netfilter*

Este componente consiste en una herramienta de cortafuegos que permite filtrar paquetes además de realizar una traducción de direcciones de red (NAT) para IPv4 o mantener registros de log. Además es una herramienta de espacio de usuario mediante la cual el administrador puede definir políticas de filtrado del tráfico que circula por la red. Actualmente el componente está disponible en prácticamente todas las distribuciones de Linux.

Además, *iptables* también incluye *ip6tables*. *ip6tables* se usa para configurar los filtros de paquetes IPv6.



The screenshot shows the main page of the netfilter.org project for iptables. The page is titled "The netfilter.org 'iptables' project" and includes sections for "About", "Projects", "Dependencies", "Main Features", "Git Tree", "Development Snapshots", and "Authors". The "About" section describes iptables as a userspace command line program used to configure the Linux 2.4.x and 2.6.x IPv4 packet filtering ruleset. The "Main Features" section lists: listing the contents of the packet filter ruleset, adding/removing/modifying rules in the packet filter ruleset, and listing/zeroing per-rule counters of the packet filter ruleset. The "Authors" section mentions that iptables was mostly written by the netfilter core team.

Ilustración 13: Página principal del proyecto Iptables

5.6.1. Resultados de la investigación

Iptables es un componente sólido, hecho que se demuestra al llevar más de diez años en el mercado de OSS. Cabe mencionar que los resultados obtenidos son en gran parte positivos. Veamos la información recopilada acerca del componente sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	Iptables	Observaciones
Estabilidad de la información del componente		
Compañías involucradas con la comunidad	Netfilter	
Historia del producto		
Versiones del Producto	v1.4.5, 0.08 years	2 meses
	v1.4.6, 0.15 years	3 meses
	v1.4.7, 0.08 years	2 meses
Tiempo del Producto en el Mercado	11 years	
Evidencia de éxito en el uso del componente		
Usuarios registrados		Información no hallada
Empresas utilizando el componente		Información no hallada
Número de Descargas		Información no hallada
Valoraciones de usuarios		
Puntuación individual y comentarios por usuario		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
Soporte y Servicios disponibles para usuarios finales		
Servicios de Soporte Gratuitos para usuarios	Listas de correo electrónico	
Servicio de Soporte No Gratuitos para usuarios		Soporte totalmente gratuito
Soporte y Servicios disponibles para integradores		
Servicios de soporte gratuitos para Integradores	Listas de correo electrónico	
Servicios de soporte no gratuitos para Integradores		Soporte totalmente gratuito
Facilidad de integración del componente		
Interoperabilidad		
Requisitos de Software	Linux 2.4.x y 2.6.x	
Requisitos de Hardware		Información no hallada
Interoperabilidad con Sistemas de Software Externos		Información no hallada
Lenguajes de Programación	C, C++	
Documentación para integradores	True	
Disponibilidad del código fuente y comentarios	Suitably commented	

Característica/Sub-característica/Atributo	Iptables	Observaciones
Rendimiento		
<i>Resultados de Rendimiento</i>	False	
<i>Resultados de test de rendimiento por terceros</i>	False	
Esquemas de licencias		
<i>Licencias OSS</i>	GNU GPLv2	
<i>Licencias comerciales</i>		No existen licencias comerciales
Calidad de la documentación		
<i>Manuales</i>		
<i>Documentación del usuario final</i>	Advanced	
<i>Documentación de integradores</i>	Advanced	
<i>Documentación adicional</i>		
<i>FAQs</i>	Advanced	
<i>Ficheros de ayuda.</i>	Not-Provided	
<i>Ayuda Online</i>	Advanced	
<i>Idiomas de la documentación</i>	Inglés, español, suizo, francés, húngaro, italiano	

Tabla 11: Modelo de calidad del proyecto Iptables

Seguidamente se va a proceder a la explicación de la información recopilada en el modelo de calidad, describiendo los resultados obtenidos en el modelo de calidad:

- *Estabilidad de la información del componente*

Referente al atributo *Compañías involucradas con el componente*, *iptables* es un componente perteneciente al grupo *Netfilter*. Cabe resaltar que *iptables* es el componente más relacionado con el nombre del grupo *Netfilter*.

En cuanto a la sub-característica *Historia del Producto* los resultados han sido los siguientes:

- La información recopilada del atributo *Versiones del producto* ha sido extraída de la sección *Downloads* [Iptables+02], situado el enlace en el menú lateral izquierdo, donde aparece un histórico de versiones del componente, especificando su fecha de salida.
- En cuanto al atributo *Tiempo del Producto en el Mercado*, la información ha sido extraída de [Iptables+03], donde se encuentra información acerca de la historia del grupo *Netfilter*.

El acceso a la información recopilada en esta característica resulta de fácil acceso, utilizando el amplio menú situado en la parte izquierda de la página web del componente.

- *Evidencia de éxito en el uso del componente*

Referente a esta característica y a sus sub-características no se ha hallado información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

En el caso de los mecanismos de soporte, tanto para usuarios como para integradores únicamente se proporcionan listas de correo electrónico. La información recopilada en este apartado puede consultarse en [Iptables+04] donde se observa la existencia de diversas listas de correo, donde se proporciona soporte según el rol que se desempeña en la comunidad.

El acceso a la información recopilada en esta característica se ha realizado a través del menú situado en la parte izquierda de la página web del componente e accediendo al enlace *Mailing Lists*, donde además se especifican las normas y otra información acerca del funcionamiento de estas listas de correo electrónico.

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad* cabe destacar que sólo se ha encontrado información para los atributos *Requisitos de Software* y *Lenguajes de Programación*, donde:

- La información recopilada en el atributo *Requisitos de Software* ha sido extraída de [lptables+01], deduciéndose esta información ya que el componente trabaja con el sistema operativo *Linux* en sus versiones 2.4.X y 2.6.X, por lo tanto, el uso de uno de estos dos sistemas operativos es imprescindible para trabajar con el componente.
- La información acerca del atributo *Lenguajes de Programación* ha sido extraída de [lptables+05] pese a que en la web del componente si se habla del uso de los lenguajes C y C++, pero no se especificaba si eran los únicos lenguajes de programación utilizados.

En cuanto al valor del atributo *Documentación para integradores* se ha calificado como “*True*” ya que en [lptables+06] se puede acceder a dicha documentación.

Finalmente, la información del atributo *Disponibilidad del código fuente y comentarios* ha sido extraída accediendo al código fuente a través de un FTP proporcionado en la página principal del componente [lptables+07]. Accediendo al código fuente se puede comprobar que el código presenta comentarios que ayudan a comprender el significado de éste por lo que se ha calificado el atributo como “*Suitably commented*”.

- *Rendimiento*

Referente a esta característica y a sus atributos no se ha hallado información.

- Esquemas de licencias

La información acerca de la licencia sobre la que está sujeto el componente, puede consultarse en [lptables+08], accediéndose directamente desde el menú principal situado en la zona lateral izquierda y pulsando sobre el enlace *Licesing*.

- *Calidad de la documentación*

Desde el la página principal del componente puede accederse a la documentación en el apartado de *Documentation* que se encuentra en el menú principal de la derecha [lptables+06]. En esta página se proporciona acceso a la documentación para usuarios y para desarrolladores pudiéndose comprobar que toda la documentación proporcionada presenta una redacción correcta, estructurada y completa. Por este motivo se ha calificado como “*Advanced*” la documentación para usuarios y para integradores, debido a la gran cantidad de documentación y a su gran calidad.

En cuanto a la sub-característica *Documentación adicional*, cabe destacar que se ha encontrado información para todos sus atributos excepto del atributo *Ficheros de ayuda*. Seguidamente se detallan estos resultados:

- La información recopilada sobre el atributo *FAQs* puede consultarse desde la sección de documentación [lptables+06], presentándose en varios idiomas y con una gran cantidad de información. Por este motivo se ha calificado como “*Advanced*”.

- En cuanto al atributo *Ayuda online* se ha calificado también como “*Advanced*” debido a la buena estructuración de los mecanismos de soporte que se proporciona.
- Sobre los ficheros de ayuda no se ha hallado información por lo que se ha calificado como “*Not provided*”.

Finalmente, sobre el atributo *Idiomas de la documentación* en [lptables+06] puede observarse que dependiendo del contenido de la documentación, la información se presenta en diversos idiomas.

5.6.2. Conclusiones de la investigación del proyecto

Investigar la comunidad de *iptables* no ha resultado una tarea complicada mayoritariamente gracias a la correcta distribución de la información, simplificando la búsqueda de información en todo el portal. El acceso a cualquier información se realiza a través del menú situado en la zona lateral izquierda de la página web de la comunidad del componente.

Referente a los resultados obtenidos acerca de la información recopilada, cabe destacar que un 33% de la información requerida por integradores no se ha hallado. Entre esta información requerida por integradores que no se ha hallado se incluye toda la información perteneciente a la característica *Evidencia de éxito en el uso del componente*. Como se ha mencionado anteriormente en otros casos de estudios ya realizados en esta tesis, la falta de esta información dificulta la tarea de decisión a la hora de seleccionar un componente, ya que los integradores valoran mucho los comentarios de otros usuarios que ya hayan utilizado el componente. Otra información que no se proporciona es acerca de los requisitos hardware del componente, pudiéndose producir incompatibilidades con el hardware por parte del integrador al desconocer esta información.

Concluyendo, *iptables* proporciona información útil para desarrolladores, debiendo añadir información acerca de la característica *Evidencia de éxito en el uso del componente* para que los integradores puedan observar opiniones sobre el componente realizadas por usuarios que hayan utilizado el componente. Añadiendo esta información, conseguirán atraer a un mayor número de personas interesadas en el componente y quizás las personas que adquieran el componente también participen en la comunidad desarrollando o proponiendo funcionalidades nuevas, realizando documentación, etc. Además el resto de información que no se proporciona resulta vital para que integradores de componentes OSS puedan valorar la adecuación del componente en un proceso de selección de componentes.

5.7. Joomla!

Joomla! es un sistema de gestión de contenidos que permite crear y editar el contenido de un sitio web. Un sistema de gestión de contenido es un software que realiza un seguimiento de cada pieza de contenido en una página web. El contenido puede ser texto, fotos, música, vídeos, documentos, etc.

Entre las principales características del componente se encuentran:

- Mejorar el rendimiento web.
- Crear versiones imprimibles de páginas.
- Añadir blogs, foros, encuestas.
- Añadir la funcionalidad de búsqueda en el sitio web.



Ilustración 14: Página principal del proyecto Joomla!

5.7.1. Resultados de la investigación

Veamos la información que recopilada acerca del componente sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	Joomla!	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	Open Source Matters Joomlaa-Hosting	Soporte legal y económico Host oficial de Joomla!
<i>Historia del producto</i>		
<i>Versiones del Producto</i>		Únicamente se presenta la última versión
<i>Tiempo del Producto en el Mercado</i>	5 years	
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>	396957	En la web se menciona que existen más de 200,000 usuarios y contribuidores en la comunidad. En el foro hay registrados 154115 usuarios
<i>Empresas utilizando el componente</i>	Danone, MTV Networks Quizilla, IHOP, Harvard University, Citibank	Existen muchas más organizaciones utilizando el componente
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Foro	
<i>Servicio de Soporte No Gratuitos para usuarios</i>		Soporte totalmente gratuito
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Foro	
<i>Servicios de soporte no gratuitos para Integradores</i>		Soporte totalmente gratuito
Facilidad de integración del componente		
<i>Interoperabilidad</i>		
<i>Requisitos de Software</i>	Php 5.2+, MySQL 4..1.X+, Apache 2.x+, Microsoft ISS	Joomla! no es compatible con MySQL 6.x

Característica/Sub-característica/Atributo	Joomla!	Observaciones
<i>Requisitos de Hardware</i>		Información no hallada
<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
<i>Lenguajes de Programación</i>	Php, JavaScript, HTML, CSS, XML, SQL	En el portal web solo se especifica Php, En Ohloh aparecen más lenguajes de programación
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably commented	
Rendimiento		
<i>Resultados de Rendimiento</i>		Información no hallada
<i>Resultados de test de rendimiento por terceros</i>		Información no hallada
Esquemas de licencias		
<i>Licencias OSS</i>	GNU General Public License v2	
<i>Licencias comerciales</i>		No incluye versión comercial.
Calidad de la documentación		
<i>Manuales</i>		
<i>Documentación del usuario final</i>	Advanced	
<i>Documentación de integradores</i>	Advanced	
<i>Documentación adicional</i>		
<i>FAQs</i>	Advanced	
<i>Ficheros de ayuda.</i>	Advanced	
<i>Ayuda Online</i>	Advanced	
<i>Idiomas de la documentación</i>	Inglés	

Tabla 12: Modelo de calidad del proyecto Joomla!

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Sobre el atributo *Compañías involucradas con la comunidad*, la información ha sido extraída del pie de página situado en la página principal del portal web del componente [Joomla+01], donde se enlaza a dos compañías que son las que participan con la comunidad. Estas compañías son *Open Source Matters* [Joomla+02] y *Joomla-Hosting* [Joomla+03]. La primera compañía proporciona organización legal y soporte económico al proyecto *Joomla!*, mientras que la segunda compañía proporciona el host oficial del proyecto.

Sobre la sub-característica *Historia del producto* y a sus atributos, los resultados se explican a continuación:

- En [Joomla+04] se explican los motivos por los cuales seguirá siendo posible conseguir soporte en un futuro, justificando este hecho con el argumento de sus raíces, ya que el desarrollo del componente comenzó en el año 2000 y actualmente sigue evolucionando. Cabe mencionar que contrastando esta información con otras fuentes en la red [Joomla+05], se ha observado que *Joomla!* realmente nace en el año 2005, siendo esta la fecha que se ha considerado para valorar su tiempo en el mercado.
- Referente a las versiones del producto, únicamente se proporciona información en el portal de la comunidad sobre la última versión de *Joomla!* por lo cual, no se ha podido valorar este atributo.

- *Evidencia de éxito en el uso del componente*

Referente al atributo *Usuarios registrados* en [Joomla+04] se menciona que pertenecen a la comunidad más de 200.000 usuarios y contribuidores. Pero observando el foro [Joomla+06], se ha observado que existen 154115 usuarios registrados en el foro, y un total de 396.957 miembros en la comunidad, siendo esta última cifra la considerada como el total de usuarios registrados en la comunidad.

Acerca del atributo *Empresas utilizando el componente* en [Joomla+04] se mencionan varias empresas, aunque existe una lista completa de organizaciones utilizando el componente donde se presentan organizadas por categorías y pueden consultarse siguiendo el enlace "*Joomla Community Site Showcase*" situado en la parte inferior del listado de algunas de las empresas que colaboran con la comunidad.

- *Disponibilidad de mecanismos de Servicios de Soporte*

Referente a los mecanismos de soporte únicamente se proporciona un foro [Joomla+06] donde se separa el contenido según el rol que se desempeñe en la comunidad. Por ejemplo, el soporte de usuario y de integradores se presenta en apartados distintos.

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad* perteneciente a esta característica, se ha hallado información sobre dos de sus cuatro atributos. Los atributos de los cuales se ha hallado información han sido:

- *Requisitos de Software*, cuya información ha sido extraída de [Joomla+07]. En este apartado se menciona la incompatibilidad del componente con la versión 6 de *MySQL*.
- *Lenguajes de Programación*. Pese a que el componente está desarrollado mayoritariamente en el lenguaje *Php* tal y como se presenta en el portal del componente, en *Ohloh* [Joomla+08] se añaden los siguientes lenguajes de programación: *JavaScript*, *HTML*, *CSS*, *XML* y *SQL*.

Sobre el atributo *Documentación para integradores* se ha valorado como “*True*” ya que se puede acceder a la documentación de integradores en [Joomla+09] donde se presenta la documentación del componente separada por roles.

Por último, el valor del atributo *Disponibilidad del código fuente y comentarios* se ha calificado como “*Suitably commented*” después de haber accedido al código fuente descargando la aplicación y observando la presencia de comentarios en él que ayudan a comprender la funcionalidad de éste. Cabe mencionar que también se puede consultar el código en un repositorio online [Joomla+10].

- *Rendimiento*

Referente a esta característica y a sus atributos no se ha hallado información.

- Esquemas de licencias

Joomla! es un componente bajo la licencia *GNU General Public License v2*, [Joomla+11]. El acceso a esta información se ha realizado accediendo desde el menú situado en la zona inferior de la página web principal del componente, siguiendo el enlace *License* situado en la columna *About*.

- *Calidad de la documentación*

Referente a la sub-característica *Manuales* cabe mencionar que toda la documentación acerca del componente es accesible desde el menú inferior de la página web del componente en el apartado de documentación [Joomla+01]. La información se presenta en el contenido de una wiki, en la cual se separa el contenido de información dependiendo del rol que se desempeña en la comunidad. Además existe una zona en la web exclusiva para desarrolladores [Joomla+12]. Dada la claridad y completitud de contenidos en la información para usuarios e integradores se han calificado los atributos *Documentación del usuario final* y *Documentación de integradores* como “*Advanced*”.

Referente a la sub-característica *Documentación adicional* todos sus contenidos han sido considerados como “*Advanced*” debido a su gran calidad en los contenidos

ofrecidos. La información acerca del atributo *FAQs* se ha extraído de [Joomla+13] donde se observa la existencia de diversas FAQ clasificadas según el contenido. El acceso a esta información ha sido realizado a través del menú inferior de la página principal de la comunidad, accediéndose desde el enlace *FAQ* situado en la columna *Documentation For Everyone*. En cuanto al atributo *Ayuda Online* se han valorado los mecanismos de soporte y la documentación que se proporcionan de forma online, comprobando su gran calidad.

Por último, la documentación presentada en el portal web del componente únicamente se encuentra en un único idioma, el inglés.

5.7.2. Conclusiones de la investigación del proyecto

Investigar *Joomla!* ha resultado una tarea que ha requerido un tiempo considerable debido a la gran cantidad de información que se proporciona en su portal web. A pesar de este gran tamaño en el contenido de información, el acceso a la información no ha resultado una tarea complicada gracias al disponer de un menú situado en la parte inferior de la página web principal, donde se presentan enlaces a distintos contenidos de información relevantes acerca del componente.

Referente a los resultados obtenidos durante la investigación un 22% de la información requerida (6 de 27 atributos) no se ha hallado. Entre esta información se encuentran los atributos *Requisitos de hardware*, *Versiones del componente*, *Interoperabilidad con otros sistemas software* y la característica *Rendimiento*.

La versión actual estable del componente es la 1.5, dando la impresión de la no aparición de muchas otras versiones durante su periodo de vida de cinco años.

En cuanto al periodo de vida del componente, cabe mencionar que en el portal web se sitúan las raíces del componente en el año 2000, pero realmente no fue hasta el año 2005 cuando *Joomla!* comenzó su camino en solitario como resultado de la bifurcación de *Mambo*, un proyecto perteneciente a una compañía australiana. Por este motivo se ha escogido la fecha de creación del componente en el momento en el cual se bifurcó del proyecto *Mambo*.

Joomla! es el primer componente que proporciona información acerca de la característica *Evidencia de éxito en el uso del componente* en esta tesis. Como se ha ido mencionando a lo largo de la tesis, uno de los factores más valorados por los integradores de componentes es el hecho de conocer la opinión de otros usuarios que ya hayan utilizado el componente. Proporcionar información acerca del número de usuarios en la comunidad, en el caso de *Joomla!* un gran número de usuarios, da confianza a la hora de decidirse a seleccionar un componente. Además, otro factor importante, es el conocimiento de la existencia de empresas que actualmente utilizan el componente; en el caso de este componente existe un extenso listado de empresas y organizaciones categorizado según el sector de la empresa u organización. En cambio, no se proporciona información acerca del número de descargas del componente y tampoco hay presencia de valoraciones de usuarios, siendo otros factores importantes a la hora de seleccionar un componente.

Por lo general, la calidad de la información proporcionada por la comunidad de *Joomla!* es buena, proporcionando gran parte de la información que los integradores requieren en el momento de seleccionar un componente, pero no la suficiente para realizar una valoración global del componente para poder ser valorado por integradores de componentes OSS.

5.8. MediaCoder

MediaCoder es un transcodificador de audio/video. Dispone de una interfaz gráfica que funciona mediante un conjunto de líneas de comando y librerías dinámicas en el caso que se desee integrar en otro software. Además, puede convertir una gran variedad de formatos de vídeo y audio mediante códecs.

Las principales funcionalidades de este componente son:

- Compresión de vídeo/audio.
- Conversión de formatos.
- Extracción de audio de archivos de vídeo.
- Procesamiento de vídeo mediante GPU (unidad de procesamiento gráfico).
- Enviar archivos de audio o video mediante protocolos HTTP.



Ilustración 15: Página principal del proyecto MediaCoder

5.8.1. Resultados de la investigación

MediaCoder es un componente con una estrategia de negocio un tanto peculiar. Existe una gran falta de información en el portal web del componente. Veamos la información que se proporciona sobre el componente según nuestro modelo de calidad:

Característica/Sub-característica/Atributo	MediaCoder	Observaciones
Estabilidad de la información del componente		
Compañías involucradas con la comunidad		Información no hallada
Historia del producto		
Versiones del Producto		Información no hallada
Tiempo del Producto en el Mercado	5 years	Se proporciona la información una vez instalada la aplicación
Evidencia de éxito en el uso del componente		
Usuarios registrados		Información no hallada
Empresas utilizando el componente		Información no hallada
Número de Descargas		Información no hallada
Valoraciones de usuarios		
Puntuación individual y comentarios por usuario		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
Soporte y Servicios disponibles para usuarios finales		
Servicios de Soporte Gratuitos para usuarios	Foro	
Servicio de Soporte No Gratuitos para usuarios	E-mail, IM	IM= Instant Messenger
Soporte y Servicios disponibles para integradores		
Servicios de soporte gratuitos para Integradores		No se proporcionan
Servicios de soporte no gratuitos para Integradores		No se proporcionan
Facilidad de integración del componente		
Interoperabilidad		
Requisitos de Software	Windows XP/2003/2000 Windows XP/2003 x64 Edition Windows Vista x86/x64 Windows 7 x86/x64 Linux with Wine Mac OS X with Darwine	

Característica/Sub-característica/Atributo		MediaCoder	Observaciones
	<i>Requisitos de Hardware</i>	Sistema 86 o x64 con uno o más procesadores 512MB RAM	Se recomienda 1GB de RAM
	<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
	<i>Lenguajes de Programación</i>	C, C++, JavaScript, XML, CSS	
	<i>Documentación para integradores</i>	False	
	<i>Disponibilidad del código fuente y comentarios</i>	Not-Provided	
Rendimiento			
	<i>Resultados de Rendimiento</i>		Información no hallada
	<i>Resultados de test de rendimiento por terceros</i>		Información no hallada
Esquemas de licencias			
	<i>Licencias OSS</i>		Información no hallada
	<i>Licencias comerciales</i>		Existen pero no se especifican
Calidad de la documentación			
<i>Manuales</i>	<i>Documentación del usuario final</i>	Basic	Muy insuficiente
	<i>Documentación de integradores</i>	Not-Provided	
<i>Documentación adicional</i>	FAQs	Basic	
	Ficheros de ayuda.	Not-Provided	
	Ayuda Online	Basic	
	<i>Idiomas de la documentación</i>	Inglés	

Tabla 13: Modelo de calidad del proyecto MediaCoder

Seguidamente se va a proceder a la explicación de la información recopilada sobre el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Sobre esta característica y sus sub-características y atributos no se proporciona ninguna información excepto la información referente al inicio del proyecto. Esta información es accesible en [MediaCoder+02] una vez se ha instalado la aplicación y accedido a ella por primera vez.

- Evidencia de éxito en el uso del componente

Sobre esta característica y sus sub-características no se ha hallado información.

- *Disponibilidad de mecanismos de Servicios de Soporte*

Acerca de los mecanismos de soporte, únicamente se proporcionan para usuarios. Entre los mecanismos de soporte que se les proporcionan, existe un foro gratuito [MediaCoder+03] pudiéndose acceder a este desde el menú situado en la zona superior de la página principal del componente [MediaCoder+01]. Además se proporcionan mecanismos de soporte específicos previo pago, pero se basan en atención personalizada a través de correo electrónico o de chat [MediaCoder+04].

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad*, se ha encontrado información acerca de tres de sus cuatro atributos. La información de los atributos *Requisitos de Software* y *Requisitos de Hardware* ha sido extraída de [MediaCoder+05], donde se proporciona información acerca de los sistemas operativos compatibles con el componente además de los componentes hardware necesarios para el buen funcionamiento del componente.

Sobre el atributo *Interoperabilidad con Sistemas de Software Externos* no se proporciona ninguna información, pero sí se ha obtenido información acerca de los lenguajes de programación, donde la información ha sido extraída de *Ohloh* [MediaCoder+06] ya que en el portal no se da información acerca de los lenguajes de programación utilizados.

El atributo *Documentación para integradores* ha sido calificado “*False*” debido a la no presencia de documentación para integradores o desarrolladores del componente.

Por último, cabe destacar que el acceso al código fuente del componente no está disponible, por lo que no se ha podido valorar.

- *Rendimiento*

Sobre esta característica y sus atributos no se ha hallado ninguna información.

- Esquemas de licencias

Referente a las licencias, cabe mencionar que no se proporciona información acerca de ningún tipo de licencia, ni licencias OSS ni comerciales. Pero cabe destacar que al disponer de versiones comerciales, estas versiones han de disponer de alguna licencia comercial que seguramente será proporcionada en el momento de comprar su versión comercial.

- *Calidad de la documentación*

Referente a la sub-característica *Manuales* y a sus dos atributos, cabe mencionar que únicamente se provee documentación para el usuario final en [MediaCoder+07] pero este manual contiene un contenido demasiado básico, por lo que se ha calificado como “*Basic*”.

En cuanto a la sub-característica Documentación adicional, la información obtenida en la exploración del portal sobre sus atributos ha sido la siguiente:

- FAQ con un contenido demasiado poco denso, por lo que se ha valorado este atributo como “*Basic*” [MediaCoder+08].
- La ayuda en la aplicación te re-direcciona a los mecanismos de soporte de la web.
- La ayuda online se ha considerado “*Basic*” debido a que únicamente se proporciona a través de un foro y únicamente para usuarios.

Por último, sobre el atributo *Idiomas de la documentación* cabe destacar que toda la documentación del componente únicamente se presenta en inglés.

5.8.2. Conclusiones de la investigación del proyecto

MediaCoder es un componente que ha dejado de distribuir su código fuente, hecho que viola una de las normas de la OSI para considerarse OSS.

Aunque se trata de un componente gratuito, no podríamos catalogarlo actualmente como OSS o FS.

Este proyecto se sostiene de donaciones voluntarias que se realizan a través de la web. Además también obtienen beneficios de la venta de licencias comerciales y del soporte personalizado que proporcionan.

Pese a no cumplir con las normas de la OSI en cuanto componente OSS, el proyecto sigue anunciándose en páginas donde se alojan proyectos OSS como *Ohloh* [MediaCoder+09], donde varios usuarios ya han pedido que sea retirado el componente de la web.

5.9. StatusNet

La plataforma *StatusNet* permite a comunidades y organizaciones incorporar micro mensajes en su propio dominio. Entre sus funcionalidades destacan:

- *Creación de grupos* donde los miembros pueden seguir noticias de temas específicos y conectar con otras personas que tienen intereses comunes.
- *Informes en tiempo real* de los contenidos más populares en tu red.
- *Compartir archivos* como videos, imágenes, etc.
- *Compartir interface con otras redes sociales* como *Twitter* o *FaceBook*.
- *Acceso desde un dispositivo móvil*.
- *Incorporación o creación de plug-ins* para añadir nuevas funcionalidades.
- *Personalización de interfaces*.

The screenshot shows the StatusNet website homepage. At the top, there is a navigation bar with links for 'products', 'solutions', 'services', 'open source', 'About', 'Blog', 'Support', and 'Contact'. The StatusNet logo is prominently displayed on the left. Below the logo, there are two main sections: a blue box on the left with the headline 'Bring the conversation to your domain.' and a green box on the right with the headline 'Find out why thousands of Businesses, Media & Consumer Brands, Online Communities and Developers are using StatusNet to stay connected.' Below these sections, there are two large buttons: 'Download StatusNet now (It's free!)' and 'Sign Up for a Cloud Account (It's free too!)'. To the right of these buttons, there is a section titled 'Who's Using StatusNet?' featuring logos for Creative Commons, Motorola, Mozilla Foundation, and Canonical. Below this, there is a button that says 'Find out about the StatusNet Enterprise Network!'. At the bottom of the page, there are three columns: 'What's New at StatusNet?' with several news items, 'Latest Add-ons' with a list of plugins like FireStatus and Laconica Bar, and 'Popular Add-ons' with a numbered list of plugins like Tweenky and Distributed Microblogging for Drupal.

Ilustración 16: Página principal del proyecto Status.net

5.9.1. Resultados de la investigación

Veamos la información recopilada sobre el componente sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	StatusNet	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	StatusNet Inc.	
<i>Historia del producto</i>		
	<i>Versiones del Producto</i>	Información no hallada
	<i>Tiempo del Producto en el Mercado</i>	Información no hallada
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>		Información no hallada
<i>Empresas utilizando el componente</i>	Motorola, creative commons, Bite-Size Edits, DreamHost, wikiHow, Mozilla Foundation, Canonical, Kings, shitmydadsays	
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
	<i>Puntuación individual y comentarios por usuario</i>	Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
	<i>Servicios de Soporte Gratuitos para usuarios</i>	Foro, wiki, listas de correo electrónico, chat irc
	<i>Servicio de Soporte No Gratuitos para usuarios</i>	Email, asistencia telefónica
		Incluye instalación, mantenimiento y desarrollo personalizado
<i>Soporte y Servicios disponibles para integradores</i>		
	<i>Servicios de soporte gratuitos para Integradores</i>	Foro
	<i>Servicios de soporte no gratuitos para Integradores</i>	Email, asistencia telefónica
Facilidad de integración del componente		
<i>Interoperabilidad</i>		
	<i>Requisitos de Software</i>	PHP 5.2.3+, MySQL 5.x
		Para funcionalidades adicionales existen otros requisitos

Característica/Sub-característica/Atributo	StatusNet	Observaciones
<i>Requisitos de Hardware</i>		Información no hallada
<i>Interoperabilidad con Sistemas de Software Externos</i>		Información no hallada
<i>Lenguajes de Programación</i>	Php, Javascript, CSS	Extraído de Ohloh
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably commented	
Rendimiento		
<i>Resultados de Rendimiento</i>		Información no hallada
<i>Resultados de test de rendimiento por terceros</i>		Información no hallada
Esquemas de licencias		
<i>Licencias OSS</i>	GNU Affero General Public License	
<i>Licencias comerciales</i>		No dispone de versión comercial
Calidad de la documentación		
<i>Manuales</i>		
<i>Documentación del usuario final</i>	Medium	Formato inadecuado
<i>Documentación de integradores</i>	Advanced	
<i>Documentación adicional</i>		
FAQs	Medium	
Ficheros de ayuda.	Basic	
Ayuda Online	Medium	
<i>Idiomas de la documentación</i>	Inglés	

Tabla 14: Modelo de calidad del proyecto StatusNet

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

La información acerca del atributo *Compañías involucradas con la comunidad* ha sido extraída de [StatusNet+02] accediendo a esta información desde el link *About* situado en la parte superior derecha de la página principal del componente [StatusNet+01].

Referente a las versiones del producto y el tiempo del producto en el mercado no se ha hallado información.

- *Evidencia de éxito en el uso del componente*

Adentrándonos en la información sobre los atributos pertenecientes a esta característica, únicamente se ha encontrado en [StatusNet+03] información acerca del atributo *Empresas utilizando el componente* donde se proporcionan los nombres de compañías que actualmente usan *StatusNet*. Se ha accedido a esta información desde la página principal del componente, donde se resalta la sección *Who is using StatusNet?*

- *Disponibilidad de mecanismos de Servicios de Soporte*

Referente a los mecanismos de soporte, tanto los referidos a usuarios finales como para integradores se encuentran en el mismo lugar dado que son compartidos por ambos roles [StatusNet+04]. El acceso a la sección de soporte puede realizarse directamente siguiendo el link *Support* situado en la parte superior derecha de la página principal del componente. Existen mecanismos de soporte gratuitos y no gratuitos. Entre los mecanismos de soporte gratuitos se proporciona:

- Un foro donde cualquier persona puede acceder para consultar temas del foro ya existentes o para crear un tema nuevo [StatusNet+05].
- Listas de correo electrónico para realizar preguntas y obtener respuestas de otros usuarios que participan en la comunidad.
- Un chat en IRC.
- Una wiki con documentación [StatusNet+06].

Como mecanismos de soporte no gratuitos se proporciona:

- Un servicio de correo electrónico personalizado
- Asistencia telefónica.

Cabe destacar que existen tres perfiles distintos cuando se desea contratar soporte personalizado. Cada perfil proporciona un tipo de soporte más o menos amplio dependiendo del precio de éste. Cuanto mayor coste sea el del perfil contratado, se recibe una mejor atención personalizada. La información extraída acerca de los mecanismos de soporte no gratuitos se ha realizado desde [StatusNet+07].

- *Facilidad de integración del componente*

Referente a la sub-característica *Interoperabilidad* se ha encontrado información de dos de sus cuatro atributos. Se ha obtenido información sobre:

- *Requisitos de Software* en [StatusNet+08], concretamente en el apartado de requisitos se especifica el software necesario para que el componente funcione correctamente. El acceso a esta información no es directo; para acceder a esta información debes consultar la sección *Download*, donde se da la opción de acceder a la wiki. Una vez se ha accedido a la wiki se puede acceder a la guía de instalación del componente, siendo en esta guía donde se especifican los requisitos de software necesarios para el componente, entre otra información relevante como por ejemplo, la guía de usuario. Este archivo de texto presenta una estructura y un formato que dificultan su lectura.
- *Lenguajes de Programación*. Esta información se ha consultado en *Ohloh* [StatusNet+09] ya que no se ha encontrado información en la página web de la comunidad.

En cuanto a la calificación del atributo *Documentación para integradores*, se ha valorado como “*True*” dado que accediéndose al link *Open Source* del menú situado en la parte superior de la página web del componente, se accede a la documentación para desarrolladores donde se puede acceder a dicha información [StatusNet+10].

- *Rendimiento*

Referente a esta característica y atributos no se ha hallado información.

- Esquemas de licencias

El acceso a la información acerca de la licencia del componente, se encuentra en el mismo archivo donde se ha recopilado la información acerca de los requisitos de software [StatusNet+08]. En este archivo se especifica entre otra información que el componente está sujeto a la licencia *GNU Affero General Public License*.

- *Calidad de la documentación*

Referente a los atributos pertenecientes a la sub-característica de calidad *Manuales*, la valoración de la información acerca del atributo *Documentación del usuario final* se ha realizado comprobando dicha documentación en [StatusNet+11]. Debido a la calidad de dicha documentación, se ha calificado este atributo como “*Medium*” pese a la presencia de un contenido aceptable en cuanto a información proporcionada. Se ha decidido esta valoración dado que el formato y la estructura en el cual se presenta la información resulta demasiado simple, dado que únicamente es un fichero de texto donde se acumula demasiada información, disponiendo de un tamaño del texto demasiado pequeño. Sobre el atributo *Documentación de integradores*, se ha accedido a esta documentación desde la wiki que proporciona la página web del componente en [StatusNet+12]. A diferencia de la documentación de usuario, esta presenta un contenido completo además de disponer de una buena estructura de la información. Por este motivo se ha calificado “*Advanced*”.

En cuanto a los atributos pertenecientes a la sub-característica de calidad “*Documentación adicional*”, se ha podido valorar cada uno de los atributos pertenecientes a esta característica:

- El acceso a la información sobre atributo *FAQs* se ha realizado accediendo a la sección *Open Source* de la web de la comunidad. Desde ese punto se ha accedido al enlace *FAQ* situado en el recuadro *Get Help* en la zona inferior derecha de la página web [StatusNet+13]. Comprobando la información proporcionada en las *FAQ*, se ha calificado este atributo como “*Medium*” dado que el contenido presentado en ella es demasiado breve.
- Los ficheros de ayuda con la aplicación presentan también un contenido breve, motivo por el cual se ha valorado el atributo como “*Basic*”.
- El atributo *Ayuda online* se ha valorado como “*Advanced*” dado que proporciona buenas y completos mecanismos de ayuda online, aunque alguno de ellos no es gratuito.

Finalmente, toda la documentación se proporciona únicamente en inglés.

5.9.2. Conclusiones de la investigación del proyecto

StatusNet es un componente OSS que puede ser utilizado por cualquier tipo de usuario. Por este motivo muchas empresas, como por ejemplo *Motorola* utilizan el componente.

En el contexto de esta tesis, la información proporcionada desde el punto de vista de un integrador de componentes OSS, es incompleta. En primer lugar, no existe un histórico de versiones para poder valorar el intervalo de tiempo que existe entre versiones del componente, al igual que se desconoce la fecha de fundación del proyecto y su historia.

A la hora de seleccionar un componente OSS, uno de los factores más importantes para integradores es la madurez de la comunidad. Por ello sería necesario añadir información acerca de la historia del componente en la página web de la comunidad. Otro aspecto relevante que ya se ha comentado en otros casos de estudio de la tesis, es el hecho de poder conocer la opinión de otros usuarios que hayan utilizado el componente. Pero por otra parte, en la comunidad si se presenta información acerca de empresas que utilizan el componente en la actualidad.

En cuanto a los mecanismos de soporte proporcionados, *StatusNet* es uno de los proyectos OSS que proporcionan soporte personalizado, proporcionando nuevos mecanismos de soporte que se están expandiendo en muchos proyectos OSS actualmente. Pese a que los mecanismos de soporte gratuitos son aceptables, para obtener una mayor calidad en el soporte recibido es necesario comprar una licencia de soporte. Pero este punto no es un problema dado que los usuarios pueden estar dispuestos a pagar por recibir este soporte.

Sobre la facilidad de integración del componente, sería interesante conocer además de los requisitos software del componente, los requisitos de hardware ya que existen personas que disponen de hardware antiguo que quizás no sea compatible con el

componente. Además siempre es interesante conocer resultados de pruebas de rendimiento para conocer con que hardware y software se puede conseguir un mayor rendimiento en el uso del componente.

Acerca de la calidad de la información proporcionada por el componente, cabe resaltar la excelente documentación para integradores disponible en su wiki, aunque se debería mejorar la documentación del usuario final, dado que presentar un manual de usuario con un formato incorrecto, como es el caso en este componente, no proporciona una buena impresión acerca del componente. El resto de información proporcionada es aceptable, pero siempre se puede mejorar.

Concluyendo, el framework proporcionado en esta tesis puede ayudar a mejorar la calidad de la información proporcionada en esta comunidad OSS.

5.10. Subclipse

Subclipse es un plug-in desarrollado para *Eclipse* para integrar *Subversión* en *Eclipse*.

Subversion es un control de versiones que está reemplazando actualmente al CVS debido a algunas deficiencias en este último.

Eclipse es un framework OSS para el desarrollo de software utilizado por una gran cantidad de usuarios actualmente. *Eclipse* dispone de muchos plug-ins, entre ellos *Subclipse*.

Al ser mantenido *Subclipse* por *Subversion*, *Subclipse* siempre está sincronizado con las últimas actualizaciones de *Subversion*.

The screenshot shows the Subclipse project page on the CollabNet website. The page is titled "Subclipse" and includes a navigation menu on the left with sections like "Project pages", "Project tools", and "Search". The main content area features a "Project home" section with a description of Subclipse as an Eclipse Team Provider plug-in. Below this is a "News" section with several announcements, including "CollabNet Merge Client now included!" and "Subclipse 1.6.0 Support". A "Revision Graph" is displayed, showing a sequence of revisions (1, 2, 3, 17) and their relationships. The page also includes a "How do I..." section and a "Featured projects" table.

Ilustración 17: Página principal del proyecto Subclipse

5.10.1. Resultados de la investigación

Veamos la información recopilada acerca del componente sobre nuestro modelo de calidad:

Característica/Sub-característica/Atributo	Subclipse	Observaciones
Estabilidad de la información del componente		
<i>Compañías involucradas con la comunidad</i>	Eclipse IDE, Collabnet, Tigris.org	
<i>Historia del producto</i>		
<i>Versiones del Producto</i>	v1.6.9, 0.01 years v1.6.10, 0.15 years v1.6.12, 0.08 years	
<i>Tiempo del Producto en el Mercado</i>		Información no hallada
Evidencia de éxito en el uso del componente		
<i>Usuarios registrados</i>	600	
<i>Empresas utilizando el componente</i>		Información no hallada
<i>Número de Descargas</i>		Información no hallada
<i>Valoraciones de usuarios</i>		
<i>Puntuación individual y comentarios por usuario</i>		Información no hallada
Disponibilidad de mecanismos de Servicios de Soporte		
<i>Soporte y Servicios disponibles para usuarios finales</i>		
<i>Servicios de Soporte Gratuitos para usuarios</i>	Foro	
<i>Servicio de Soporte No Gratuitos para usuarios</i>	Soporte telefónico, ayuda en incidencias	
<i>Soporte y Servicios disponibles para integradores</i>		
<i>Servicios de soporte gratuitos para Integradores</i>	Foro	
<i>Servicios de soporte no gratuitos para Integradores</i>	Soporte telefónico, ayuda en incidencias	Para obtener soporte se ha de contratar soporte de subversión en CollabNet
Facilidad de integración del componente		
<i>Interoperabilidad</i>		
<i>Requisitos de Software</i>	Eclipse 3.x Subversion 1.6.x	Requisitos para la versión 1.6.x de Subclipse

Característica/Sub-característica/Atributo	Subclipse	Observaciones
<i>Requisitos de Hardware</i>		Información no hallada
<i>Interoperabilidad con Sistemas de Software Externos</i>	Eclipse	
<i>Lenguajes de Programación</i>	Subversion	
	Java	
<i>Documentación para integradores</i>	True	
<i>Disponibilidad del código fuente y comentarios</i>	Suitably commented	
Rendimiento		
<i>Resultados de Rendimiento</i>		Información no hallada
<i>Resultados de test de rendimiento por terceros</i>		Información no hallada
Esquemas de licencias		
<i>Licencias OSS</i>	Eclipse Public License 1.0	
<i>Licencias comerciales</i>		No dispone de versión comercial
Calidad de la documentación		
<i>Manuales</i>		
	<i>Documentación del usuario final</i>	Advanced
	<i>Documentación de integradores</i>	Advanced
<i>Documentación adicional</i>		
	FAQs	Advanced
	Ficheros de ayuda.	Not-provided
	Ayuda Online	Medium
<i>Idiomas de la documentación</i>	Inglés	

Tabla 15: Modelo de calidad del proyecto Subclipse

Seguidamente se va a proceder a la explicación de la información reunida en el modelo de calidad, describiendo los resultados obtenidos por cada característica:

- *Estabilidad de la información del componente*

Referente al atributo *Compañías involucradas en la comunidad* en la página principal del componente [Subclipse+01] encontramos que *Subclipse* es un equipo proveedor de plug-ins para *Eclipse*. El proyecto *Subclipse* está alojado en el grupo *Tigris*, una comunidad OSS centrada en construir herramientas para el desarrollo colaborativo de software. A su vez, *Tigris* está alojado en *CollabNet*, una compañía que vende software de gestión para equipos de desarrollo distribuidos que trabajan en empresas y en el desarrollo de OS. *CollabNet* proporciona soporte para muchos proyectos OSS, incluyendo *Subclipse*.

En cuanto al atributo *Historia del Producto*, no se ha encontrado información sobre el atributo *Tiempo del Producto en el Mercado*. En cambio, si se ha obtenido información acerca de las versiones del componente, donde seleccionando una versión del componente en la sección *Download and Install*, por ejemplo la 1.6, podemos ver todas las actualizaciones de esta versión [Subclipse+02]. Cabe mencionar que únicamente se presenta dicha información acerca de las últimas versiones del componente.

- *Evidencia de éxito en el uso del componente*

La información obtenida sobre esta característica, únicamente ha sido referente al atributo *Usuarios registrados*. En [Subclipse+03] podemos comprobar que existen 600 usuarios registrados actualmente en la comunidad. El acceso a esta información se ha realizado desde la página principal del componente [Subclipse+01], accediendo al enlace *Membership* situado en el menú de la zona izquierda en la página principal de la comunidad.

- *Disponibilidad de mecanismos de Servicios de Soporte*

En el caso de este componente, podemos ver a los integradores como usuarios finales dado que *Subclipse* se trata de un componente utilizado en el desarrollo de software, por lo que los usuarios finales son desarrolladores que utilizan este componente en el desarrollo de software. Por este motivo, los mecanismos de soporte sobre el modelo de calidad coinciden los referentes a usuarios finales y a los de integradores. Entre estos mecanismos de soporte, en la sección *Issues and Help* [Subclipse+04], pudiéndose acceder desde el menú lateral de la zona izquierda de la página principal del componente [Subclipse+01], encontramos la presencia de un foro [Subclipse+05] como mecanismo de soporte gratuito, además de ofrecer la opción de contratar soporte personalizado a través de *CollabNet*. Entre los mecanismos de soporte no gratuitos proporcionados por *CollabNet* encontramos soporte telefónico y ayuda en incidencias [Subclipse+06]. Pero estos mecanismos pueden variar según la modalidad contratada de soporte personalizado (Silver, Gold, Platinum).

- *Facilidad de integración del componente*

Referente a los atributos pertenecientes a la sub-característica *Requisitos de Software* se ha obtenido información acerca de dos de sus cuatro atributos. Se ha recopilado información de los atributos:

- *Requisitos de Software*. En la sección *Download and Install* [Subclipse+07] se especifica que para el uso de la última versión del componente es necesaria la versión 3.X de *Eclipse* y la 1.6.X de *Subclipse*.
- *Lenguajes de programación*, donde la información se ha extraído de *Ohloh* [Subclipse+08]. Aunque en el proyecto se menciona el lenguaje de programación, se ha decidido contrastar esta información en el portal *Ohloh*, dado que este portal realiza un estudio del código utilizado en el componente y publica esta información en su web.

El atributo *Documentación para integradores* se ha calificado “*True*” dado que accediendo a la wiki del componente desde el menú lateral situado en la zona izquierda de la página web del componente, puede accederse a la sección *Subclipse Development* [Subclipse+09] donde se puede acceder a la documentación para integradores.

Finalmente se ha valorado el atributo *Disponibilidad del código fuente y comentarios* como “*Suitably commented*” una vez que se ha descargado el componente y comprobado el código fuente de la aplicación, presentando éste comentarios que facilitan su comprensión.

- *Rendimiento*

Referente a esta característica y a sus atributos no se ha hallado información.

- *Esquemas de licencias*

Tal y como se menciona en la página principal del componente [Subclipse+01], el componente está bajo la licencia *Eclipse Public License 1.0*.

- *Calidad de la documentación*

Referente a la sub-característica *Manuales*, se han considerado los atributos *Documentación del usuario final* y *Documentación de integradores* como “*Advanced*”. Como se ha mencionado anteriormente, en este componente el usuario final y el integrador juegan el mismo rol. Observando la documentación proporcionada en la comunidad, por un lado el manual de instalación y uso del componente [Subclipse+10] y del otro la documentación más relacionada con el desarrollo de software [Subclipse+11], se ha observado que la estructura y contenidos de la documentación presentada son adecuados para los requisitos de los usuarios finales. A esta información se ha accedido desde las secciones *Documents & files* y *Wiki* situadas en el menú de la zona lateral izquierda de la web de la comunidad.

En cuanto a la sub-característica *Documentación adicional*, se ha calificado el atributo *FAQs* como “*Advanced*”, una vez consultada la información proporcionada en las *FAQ* [Subclipse+12]. Se ha accedido a esta sección desde la wiki del componente [Subclipse+13], donde se ha comprobado que contiene un amplio y completo contenido. Hay que resaltar que los archivos de ayuda de la aplicación se integran en *Eclipse*, el componente en el cual se integra *Subclipse*. El atributo *Ayuda Online* se ha calificado como “*Medium*” dado que únicamente se proporciona un foro como mecanismo de soporte online.

Finalmente, la documentación acerca del componente únicamente se proporciona en inglés.

5.10.2. Conclusiones de la investigación del proyecto

Subclipse como se ha explicado anteriormente, se trata de un componente desarrollado para ser integrado en *Eclipse* para ofrecer la funcionalidad de controlador de versiones. Por este motivo, la información y documentación ofrecida por la comunidad está diseñada para satisfacer las necesidades de los usuarios que vayan a instalar este componente en *Eclipse*.

Centrándonos en la recopilación de información requerida en este estudio, se ha obtenido un 74% del total de información que se requería (información sobre 20 atributos de un total de 27). Entre la información no proporcionada acerca del componente encontramos información sobre el tiempo en el mercado del componente, empresas que utilicen el componente actualmente, el número de descargas realizadas, la valoración de usuarios que lo hayan usado, los requisitos de hardware para un buen funcionamiento y pruebas de rendimiento.

Actualmente existen otros gestores de contenido para *Eclipse*, por lo que proporcionar información valorada por integradores ayudará a que los integradores se decidan a utilizar el componente.

Entre la información más importante que deberían proporcionar encontramos las valoraciones de usuarios, ya que estas valoraciones son muy valoradas a la hora de seleccionar un componente. Además, poder comprobar el rendimiento de la aplicación sería otro factor que ayudaría a diferenciarse de sus competidores.

Otro factor importante, como se ha dicho, son los requisitos hardware que necesita el componente para su correcto funcionamiento, siendo posible que el uso del componente en *Eclipse* dispare el consumo de recursos del ordenador pudiendo ocasionar problemas como por ejemplo, insuficiencia de memoria.

Como consideración final acerca de la información proporcionada por la comunidad del componente, existe una carencia de información importante que la comunidad debería estudiar de proporcionar para conseguir que usuarios o integradores que necesiten seleccionar un controlador de versiones se decidan por su componente en vez de alguno de sus competidores.

6. Análisis de los resultados obtenidos

En este capítulo se va a proceder a realizar un resumen de los resultados obtenidos en el capítulo 5 seguido de un análisis global de estos resultados.

6.1. Resumen del survey

En esta tesis se ha procedido a la construcción de un modelo de calidad que nos ha permitido valorar la calidad y disponibilidad de la información en diez proyectos OSS.

Seguidamente se va a proceder a la comprobación y análisis de los resultados obtenidos en la recopilación de información sobre los diez proyectos investigados en esta tesis. Para ello, se ha desarrollado la tabla 16 que se adjunta seguidamente donde:

- ✓= Información proporcionada por la comunidad.
- ☑= Información extraída fuera de la comunidad.
- ×=información no hallada.
- ✓☑= Conflicto entre la información hallada en la comunidad y con la hallada fuera de la comunidad.
- NP= Not Provided
- B= Basic
- M= Medium
- A= Advanced
- 1= Se proporciona documentación en un único idioma
- V= Se proporciona documentación en varios idiomas.
- SC= Suitably Commented

Característica/Sub-característica/Atributo	Agilo for Scrum	Firefox	Fluent NHibernate	Gimp	Grub	Iptables	Joomla!	MediaCoder	StatusNet	Subclipse
Facilidad de integración del componente										
<i>Interoperabilidad</i>										
Requisitos de Software	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Requisitos de Hardware	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗
Interoperabilidad con Sistemas de Software Externos	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Lenguajes de Programación	✓☑	✓☑	☑	☑	✓☑	☑	✓☑	☑	☑	☑
Documentación para integradores	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Disponibilidad del código fuente y comentarios	SC	SC	SC	SC	SC	SC	SC	NP	SC	SC
Rendimiento										
Resultados de Rendimiento	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Resultados de test de rendimiento por terceros	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Esquemas de licencias										
Licencias OSS	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Licencias comerciales	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Calidad de la documentación										
<i>Manuales</i>										
Documentación del usuario final	M	A	M	A	A	A	A	B	M	A
Documentación de integradores	B	A	M	A	A	A	A	NP	A	A
<i>Documentación adicional</i>										
FAQs	M	A	B	A	B	A	A	B	M	A
Ficheros de ayuda.	M	A	NP	A	NP	NP	A	NP	B	NP
Ayuda Online	B	A	M	A	M	A	A	B	M	M
Idiomas de la documentación	1	V	1	V	1	V	1	1	1	1

Tabla 16 Resumen de los resultados obtenidos en el estudio realizado

6.2. Análisis del estudio realizado

Seguidamente se van a analizar los resultados obtenidos comentando los resultados agrupados por características:

- *Estabilidad de la información del componente*

Los resultados sobre la información recopilada de los atributos pertenecientes a esta característica se muestran en la ilustración 18.

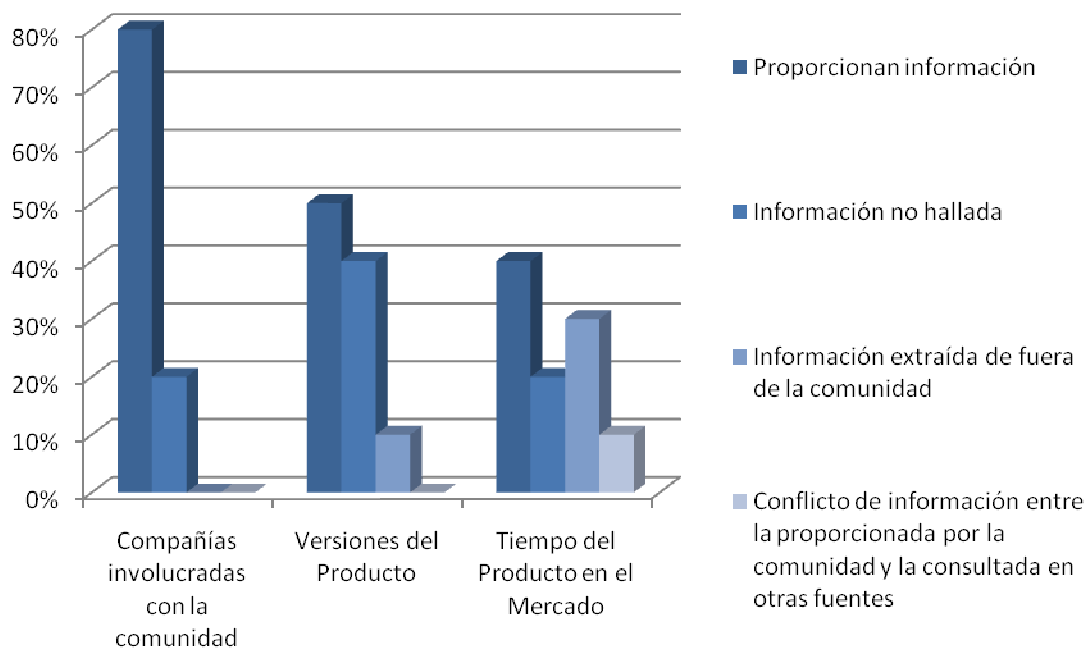


Ilustración 18: Resultados de los atributos pertenecientes a la característica *Estabilidad de la información del componente*

Sobre el atributo *Compañías involucradas con la comunidad*, ocho de las diez comunidades investigadas proporcionan información acerca de este atributo. La mayoría de estas comunidades no resalta esta información, encontrándose normalmente en letra pequeña al pie de página o en el nombre del componente, como por ejemplo en el caso de *Gimp*.

Acerca de los atributos pertenecientes a la sub-característica *Historia del producto*, los resultados han sido los siguientes:

- Únicamente cinco de las diez comunidades investigadas proporcionan información sobre el atributo *Versiones del Producto* hecho que evidencia que en muchos proyectos OSS se obvia proporcionar esta información. Hallar esta información cuando no es proporcionada por la comunidad del componente resulta una tarea complicada y difícilmente será posible hallar esta información fuera de comunidad. En el estudio realizado se ha hallado información acerca de las versiones del producto fuera de la comunidad únicamente en un componente, *Mozilla Firefox*.

- Cuatro de las diez comunidades investigadas proporcionan información acerca del atributo *Tiempo del Producto en el Mercado*. En una de estas comunidades, concretamente en el caso de *Joomla!* se ha hallado una información que al ser contrastada con otras fuentes externas, se ha determinado que la información proporcionada por la comunidad no era correcta. Del total de comunidades que no proporcionan información acerca este atributo, se ha conseguido reunir información de tres comunidades fuera de la comunidad. Para integradores de componentes OSS la información que proporciona este atributo indica la madurez del componente, siendo un factor muy valorado por la cual debería ser provista esta información por las comunidades OSS.

- *Evidencia de éxito en el uso del componente*

Los resultados de los atributos pertenecientes a esta característica se muestran en la ilustración 19.

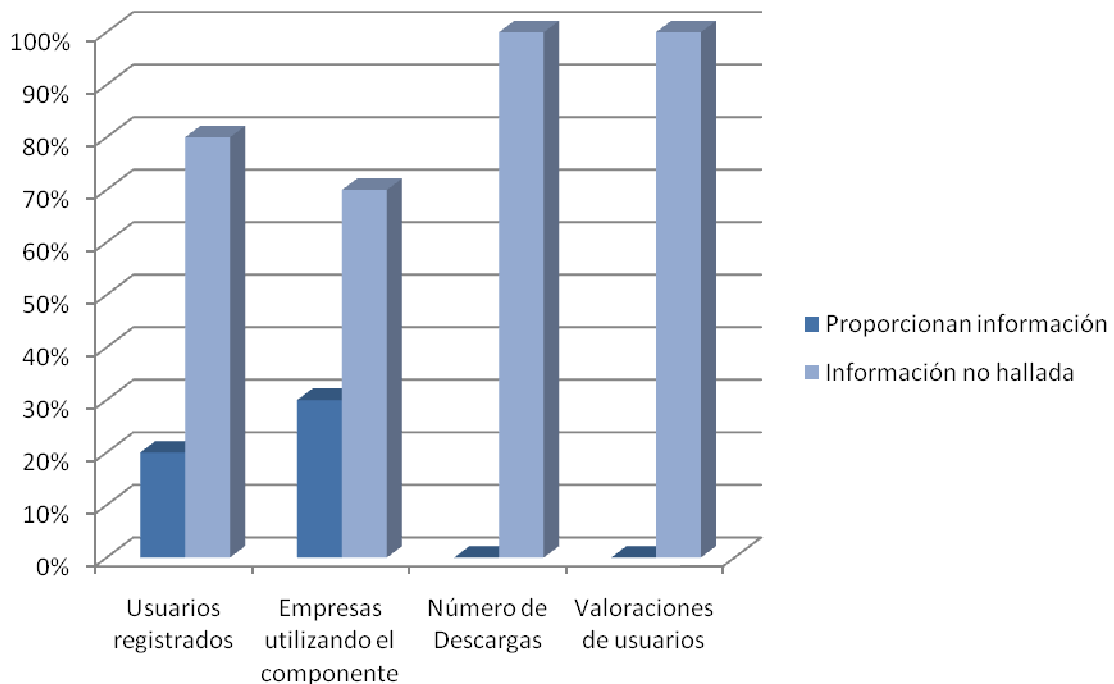


Ilustración 19: Resultados de los atributos pertenecientes a la característica *Evidencia de éxito en el uso del componente*

En los resultados de la ilustración 19 se observa que:

- Dos de las diez comunidades investigadas proporcionan información sobre los usuarios registrados. Cabe destacar que esta información no se resalta, por lo que en algunos casos resulta complicado obtener esta información. En el caso de nuestro estudio se evidencia la falta de esta información en los proyectos OSS, siendo uno de los factores que los integradores tienen en cuenta al realizar una selección de componentes OSS.

- Tres de las diez comunidades investigadas proporcionan información acerca del atributo *Empresas utilizando el componente*. El conocimiento de esta información añade valor y confianza al componente, pero tal y como muestran los resultados no es un factor que las comunidades OSS den relevancia ya que siete comunidades no proporcionan este tipo de información.
 - Ninguna de las comunidades ofrece información sobre el *Número de descargas* del componente. Aunque este atributo puede parecer confuso dado que el número de descargas no coincidirá con el número total de usuarios que han descargado el componente, esta información puede ser útil para obtener una idea acerca del uso del componente.
 - Ninguna de las comunidades investigadas ofrece valoraciones de usuarios. Uno de los factores más valorados por integradores a la hora de seleccionar componentes OSS en el estudio realizado en [Ayalaetal2010], son las valoraciones de usuarios. Estas valoraciones ofrecen información acerca del funcionamiento y agrado del componente. Como se evidencia en el estudio, ninguna comunidad ofrece esta información, por lo que los integradores de componentes deberán intentar hallar esta información fuera de la comunidad.
- *Disponibilidad de mecanismos de servicios de soporte*

Los resultados de la investigación sobre esta característica se pueden observar en la ilustración 20.

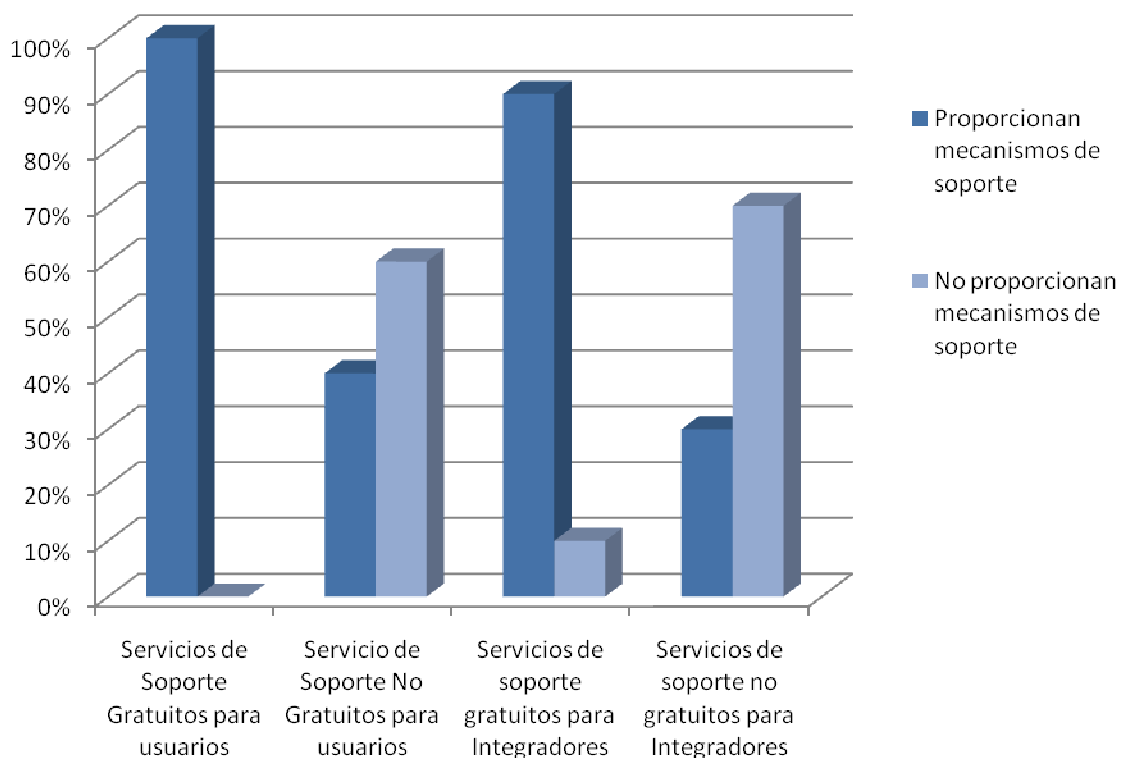


Ilustración 20: Resultados de los atributos pertenecientes a la característica *Disponibilidad de mecanismos de servicios de soporte*

Entre estos resultados se observa que:

- Todas las comunidades investigadas ofrecen algún tipo de soporte gratuito para usuarios. Entre los tipos de soporte más comunes encontramos la presencia de un foro o de listas de correo electrónico.
 - Cuatro de las comunidades ofrecen algún tipo de servicios de soporte no gratuitos. En el caso de *Subclipse* el soporte se ofrece a través de otra compañía tal y como [Fitzgerald2006] comenta que ocurre en las comunidades que siguen una estrategia de negocio *value-added-service-enabling* dentro del nuevo movimiento OSS 2.0.
 - Nueve de las diez comunidades investigadas ofrecen algún tipo de soporte gratuito para integradores. *MediaCoder* es la única compañía que no ofrece mecanismos de soporte gratuitos para usuarios debido a que actualmente no se trata de una comunidad OSS como ya se ha comentado anteriormente.
 - Tres de las comunidades investigadas proporcionan algún tipo de mecanismo de soporte no gratuito, evidenciando que no es una de las principales finalidades ofrecer soporte personalizado en comunidades OSS.
- *Facilidad de integración del componente*

Los resultados de los atributos pertenecientes a esta característica se muestran en la ilustración 21 a excepción del atributo *Disponibilidad del código fuente y comentarios*, el cual se detallará en la ilustración 22.

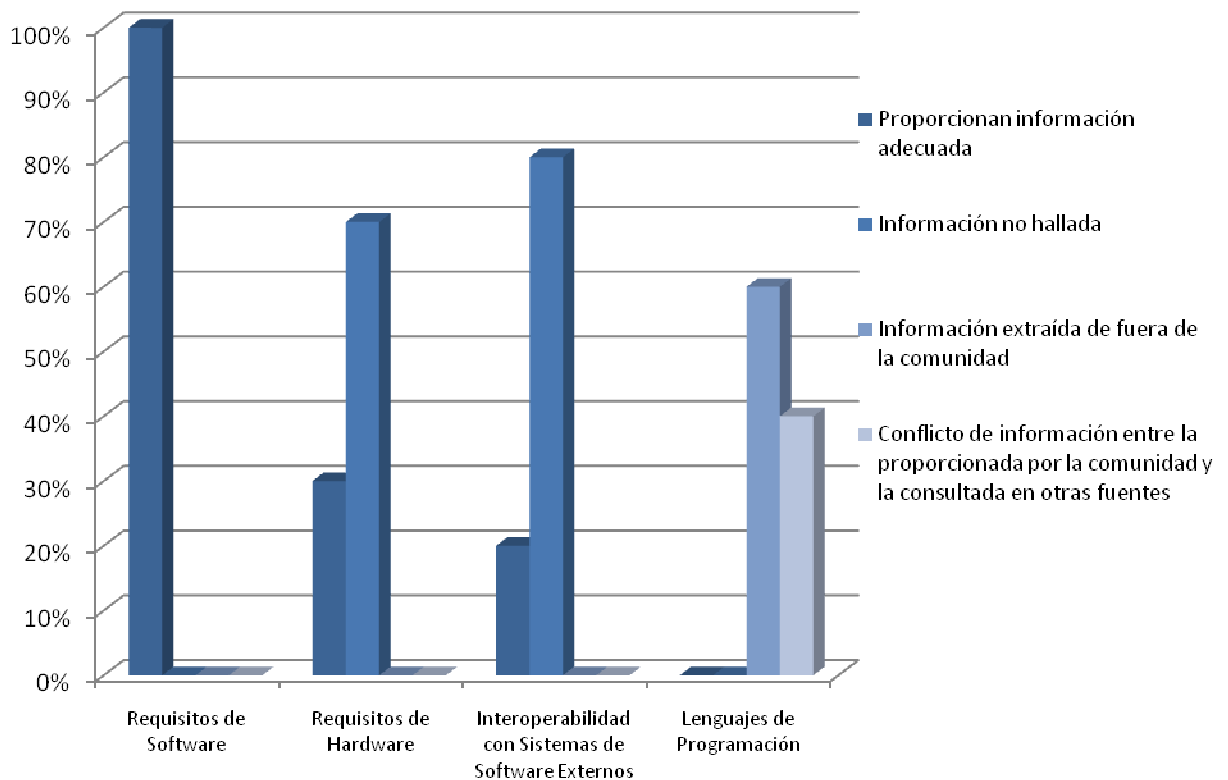


Ilustración 21: Resultados de los atributos pertenecientes a la característica *Facilidad de integración del componente* a excepción del atributo *Disponibilidad del código fuente y comentarios*

En estos resultados se observa que:

- Todos los proyectos investigados ofrecen algún tipo de información sobre el atributo *Requisitos de Software* aunque en la mayoría de casos, esta información es demasiada escasa y poco detallada.
- Tres del total de proyectos investigados ofrecen información acerca del atributo *Requisitos de Hardware*. En algunos esta información proporcionada resulta incompleta demostrando que las comunidades OSS no priorizan proporcionar este tipo de información.
- Dos del total de proyectos investigados ofrecen información acerca del atributo *Interoperabilidad con Sistemas de Software Externos*. En los proyectos que se ha hallado información acerca de este atributo, esta información no se presentaba clasificada, sino entremedio de otro tipo de información como por ejemplo los requisitos software.
- Sobre el atributo *Lenguajes de programación*, en la mayoría de proyectos investigados no se detalla el total de lenguajes de programación utilizados en el desarrollo de componentes. Por ello la información recopilada se ha realizado de forma externa a la comunidad, resaltando la carencia de esta información por parte de las comunidades OSS.
- Nueve del total de proyectos investigados ofrecen algún tipo de información para integradores, siendo *MediaCoder* el único caso especial dado que no proporciona acceso a su código fuente, por lo tanto tampoco ofrece documentación para desarrolladores e integradores.

Acercas del atributo *Disponibilidad del código fuente y comentarios* los resultados pueden observarse en la ilustración 22:

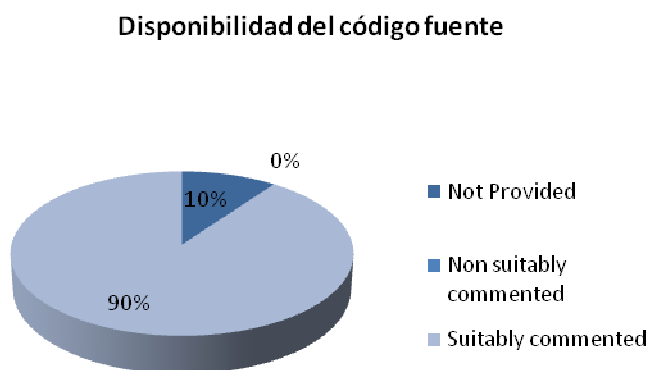


Ilustración 22: Resultados del atributo *Disponibilidad del código fuente y comentarios*

En el resultado de este atributo se observa que todos los proyectos investigados a excepción de uno (*MediaCoder*), ofrecen acceso al código fuente proporcionando un código comentado, destacando el interés por las comunidades OSS en ofrecer un código que resulta comprensible para cualquier persona que quiera colaborar con la comunidad.

- *Rendimiento*

Sobre esta característica ninguna de las comunidades investigadas ofrece información acerca del rendimiento de su componente. Con este resultado se observa la falta de interés que presentan las comunidades OSS en ofrecer datos acerca del rendimiento de los componentes que distribuyen.

- *Esquemas de licencias*

A excepción de *MediaCoder* todos los componentes presentan licencias OSS o FS. Únicamente *Agilo* ofrece la posibilidad de conseguir una versión comercial del componente, aunque *MediaCoder* también proporciona versiones comerciales pero no se ha podido acceder a las licencias de estas versiones comerciales.

- *Calidad de la documentación*

Referente a la sub-característica de manuales, los resultados obtenidos de cada atributo pueden observarse en las ilustraciones 23 y 24.

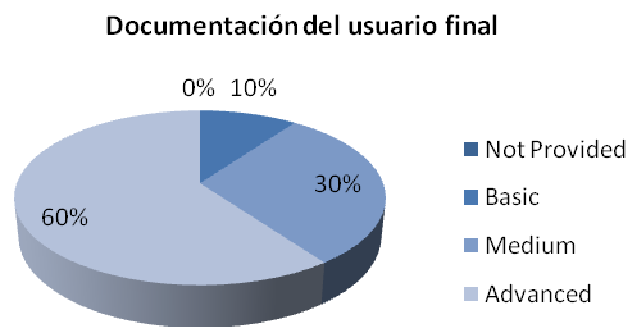


Ilustración 23: Resultados del atributo *Documentación del usuario final*

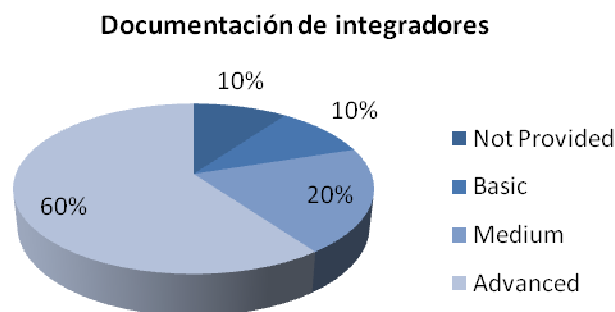


Ilustración 24: Resultados del atributo *Documentación de integradores*

Estos resultados evidencian la preocupación por las comunidades OSS en ofrecer una correcta documentación acerca de sus componentes. La mayoría de las comunidades ofrecen una documentación apropiada para su componente siendo esta complementada por los mecanismos de soporte que proporcionan. Normalmente las

comunidades más pequeñas tienden a proporcionar una documentación más escasa debido a que no disponen de recursos suficientes para crear una documentación completa y correcta, destinando sus recursos a otros sectores de la comunidad como por ejemplo, a proporcionar un correcto soporte a usuarios del componente.

Referente a la sub-característica *Documentación adicional* los resultados pueden observarse en las ilustraciones 25, 26 y 27. Entre estos resultados se ha observado:

- En el caso del atributo *FAQs*, los resultados obtenidos muestran que todas las comunidades investigadas proporcionan este tipo de información. En cuanto a la calidad de esta información únicamente cinco comunidades de las investigadas consiguen la calificación más alta.

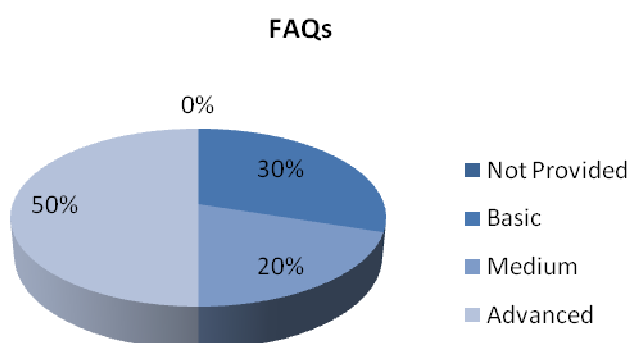


Ilustración 25: Resultados del atributo FAQs

- En cuanto al atributo *Ficheros de ayuda* los resultados muestran que la mitad de las comunidades investigadas no proporcionan ficheros de ayuda con la aplicación, por lo que el usuario o integrador deberá consultar el manual de usuario en el caso de tener una duda acerca del funcionamiento de la aplicación.

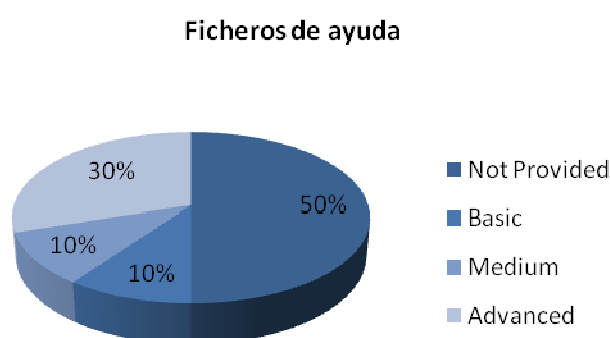


Ilustración 26: Resultados del atributo Ficheros de ayuda

- En el atributo *Ayuda Online* se ha observado que todas las comunidades investigadas ofrecen algún tipo de ayuda online. En cuanto a la calidad de esta información únicamente cuatro comunidades de las investigadas consiguen la calificación más alta.

Ayuda online

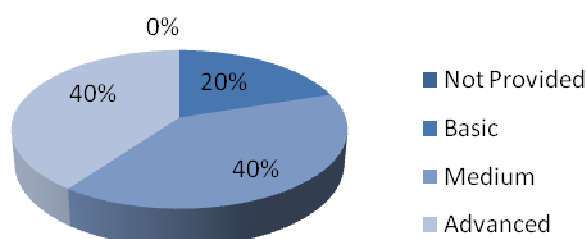


Ilustración 27: Resultados del atributo *Ayuda online*

Finalmente el único atributo restante a analizar es *Idiomas de la documentación*. Los resultados del análisis de este atributo pueden observarse en la ilustración 28.

Idiomas de la documentación

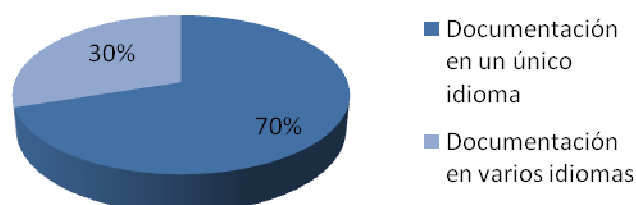


Ilustración 28: Resultados del atributo *Idiomas de la documentación*

En los resultados obtenidos se puede observar que siete de las comunidades investigadas proporcionan documentación en un único idioma, siendo el inglés el idioma seleccionado para proporcionar la documentación. Este hecho dificulta el acceso a información a personas que desconocen dicho idioma, por lo que difícilmente accederán a participar en la comunidad o a utilizar el componente.

7. Validez y limitaciones del estudio

El estudio realizado en esta tesis por su propia naturaleza presenta algunas amenazas en cuanto a la validez. En esta sección se analizan esas amenazas en términos de validez de construcción, validez interna y validez externa tal y como sugieren [Robson2002, Wohlinetal2000]. Además se enfatizan las estrategias correspondientes utilizadas sobre estas amenazas.

7.1. Validez de la construcción

Validez en la construcción de un estudio se refiere al grado en que la operacionalización de los elementos constitutivos del estudio (ej. test, cuestionarios, guiones de entrevista) miden realmente lo que debe medirse para proporcionar una respuesta coherente y significativa en el estudio [Robson2002]. En este sentido, en vez de utilizar necesidades de integradores extraídas simplemente de la literatura, esta tesis se ha basado en los resultados de un estudio empírico realizado por el grupo GESSI y NTNU [Ayalaetal2010] que demuestra las necesidades reales de los integradores de componentes OSS en el proceso de selección de estos componentes. Es decir, el framework construido y utilizado como base para la extracción y análisis de la información en cada caso de estudio, está basado en las necesidades reales de los integradores con respecto a la calidad y disponibilidad de la información provista por parte de las comunidades OSS, por lo que dicho framework se considera adecuado como estrategia para lidiar con varias posibles amenazas a la validez de construcción

7.2. Validez interna

La validez interna de un estudio se refiere al grado en que el estudio de casos es una investigación objetiva, es decir, refleja y explica la verdadera situación analizada. Las estrategias utilizadas o hechos relacionados con la investigación que fomentan la validez interna de nuestro estudio son: En primer lugar, destacar que nuestro interés en este estudio es totalmente imparcial, por lo que no existe la posibilidad de que la información de los diversos proyectos OSS analizados (i.e., los casos de estudio) hayan sido manipulados para mostrar un mejor cumplimiento de las demandas de los integradores en cualesquiera de los casos estudiados. Es decir, nuestro objetivo es meramente académico. Por otro lado, el framework utilizado y sus métricas asociadas nos permiten evaluar cada comunidad usando el mismo modelo, lo cual reduce desviaciones en el análisis y comparación de comunidades.

Somos conscientes otros aspectos que podrían afectar la validez interna de este trabajo, por ejemplo: la volatilidad de la información referente a las comunidades OSS, ya que la información proporcionada por una comunidad OSS puede ser ampliada, modificada o suprimida en cualquier momento, pudiendo variar el resultado del estudio según la fecha escogida para realizarse. Por ello, además de realizarse una investigación profunda de cada comunidad usando como protocolo de recopilación de información el framework construido y mencionado anteriormente, se han capturado y detallado las fechas determinadas en que se ha accedido y analizado cada caso de estudio con el fin de establecer la validez de la información encontrada y analizada.

7.3. Validez externa

La validez externa hace alusión a la capacidad de generalización de las conclusiones del estudio de casos. Es importante destacar que los estudios de casos, como el que se ha realizado en esta tesis, rara vez intentan hacer generalizaciones universales más allá del entorno del estudio. En cambio, si se centran en la descripción de los fenómenos contextuales del estudio [Robson2002].

Es obvio que lograr una completa validez externa requeriría el análisis del universo de proyectos OSS existentes, lo cual resulta prácticamente inviable. Para abordar este estudio en el contexto de una tesis de máster como la que se presenta, se decidió utilizar una estrategia que aborda una muestra de 10 proyectos o casos de estudio, la cual resulta bastante pequeña tomando en cuenta la cantidad de proyectos OSS existentes hoy en día. Sin embargo, para fortalecer el potencial de validez externa de nuestro estudio, la estrategia seguida consistió en buscar las comunidades OSS en el portal *www.ohloh.net*, (que es considerado el mayor directorio de proyectos OSS). En este portal, existía en la fecha de la selección un total de 437.982 proyectos OSS registrados en la web. Para la selección de las diez comunidades a investigar, se decidió ordenar el total de comunidades por el número de usuarios pertenecientes a *Ohloh* que utilizan el componente. Esta lista de componentes se decidió fragmentar en tres partes, realizando una división entre tres sobre el total de proyectos OSS registrados en *Ohloh*. De esta forma se obtuvieron tres grupos distintos:

- Un primer grupo con las comunidades OSS que disponen de más usuarios.
- Un segundo grupo con un tamaño medio en el número de usuarios.
- Un tercer grupo con las comunidades OSS que disponen de menos usuarios.

Una vez realizada esta fragmentación en tres grupos, se procedió a seleccionar al azar cuatro componentes del primer grupo, tres componentes del segundo grupo y tres componentes del tercer grupo. De esta manera, nuestro análisis aborda proyectos de diversas envergaduras, lo que nos permite una mayor fiabilidad en nuestras observaciones.

Sin embargo, se ha sido consciente de que los resultados podrían haber sido distintos si se hubiese utilizado cualquier otro criterio al a hora de seleccionar las comunidades OSS a investigar, por lo que las afirmaciones realizadas en esta tesis no deben tomarse como afirmaciones, sino como hipótesis que deben ser más validadas en profundidad.

8. Conclusiones Finales y Trabajo Futuro

En este capítulo se va a proceder a la exposición de las conclusiones finales obtenidas en esta tesis seguido del trabajo futuro.

Este trabajo presenta los resultados de un estudio un estudio orientado a investigar indagar en qué medida la información facilitada por las comunidades OSS cubre los criterios que se utilizan en la práctica industrial de selección de componentes. Con tal motivo, se desarrolla un framework para facilitar la identificación y consideración de las características que un integrador espera obtener cuando consulta información relacionada con un componente, seguidamente de un survey orientado a analizar en profundidad diversas comunidades OSS.

Centrándonos y observando los resultados obtenidos en el estudio realizado, se evidencia la existencia de una clara carencia en la calidad y disponibilidad de la información provista por las comunidades OSS. Las comunidades OSS ofrecen una información desestructurada de sus componentes sin valorar las necesidades de los distintos tipos de personas que pueden interesarse por su componente. Si observamos los resultados obtenidos en el caso de la característica *Rendimiento* de nuestro framework, podemos observar que ninguna de las comunidades investigadas ha proporcionado ningún tipo de información. Por otro lado, la característica que mejor se ha podido valorar ha sido *Calidad de la Documentación*, ya que la mayoría de comunidades investigadas proporcionaba la documentación que se buscaba con nuestro framework, aunque la calidad de esta documentación en general, no era la deseable.

En el caso de los integradores, se evidencia que en el proceso de selección de componentes OSS la información provista por las comunidades resulta insuficiente en la mayoría de comunidades para utilizar cualquier técnica de selección de componentes. Además, la presentación de la información se realiza de una forma desestructurada hecho que dificulta el proceso de obtención de información para una valoración correcta del componente. Por este motivo es comprensible que los integradores de componentes OSS empleen otros mecanismos para seleccionar componentes, como por ejemplo valorar la experiencia previa con el componente tal y como se refleja en el estudio [Ayalaetal2010].

Por ello, es de vital importancia dar a conocer a las comunidades OSS la información relevante que utilizan los integradores de componentes OSS a la hora de seleccionar un componente para conseguir que los integradores dispongan de herramientas que puedan utilizar en la práctica para valorar la adecuación de un componente en su dominio determinado. En este contexto, la relevancia de este trabajo reside en que intenta demostrar no solo las necesidades de información de los integradores, sino también cuantificar el porcentaje de información que se encuentra disponible con el fin de fomentar el interés de las comunidades OSS para proveer dicha información y satisfacer las demandas de los integradores.

Los contenidos del futuro trabajo a realizar son muy amplios debido a la escasa investigación realizada en el campo de la disponibilidad y calidad de la información proporcionada por proveedores de componentes.

9. Bibliografía

- [Ayalaetal2009] Ayala, C., Hauge, Ø, Conradi, R., Franch, X., Li, J., Sandanger-Velle, K., 2009. Challenges of the Open Source Component Marketplace in the Industry. Proceedings of the 5th IFIP Working Group 2.13 International Conference on Open Source Systems (OSS2009) - Open Source Ecosystems: Diverse Communities. Springer. Skövde, Sweden, pp. 213-224.
- [Ayalaetal2010] Claudia Ayala, PhD.; Øyvind Hauge, MsC; Reidar Conradi, PhD.; Xavier Franch, PhD.; Jingyue Li, PhD. Selection of Third Party Software in Off-The-Shelf-Based Software Development - An Interview Study with Industrial Practitioners
- [Basili2004] Basili, V.R., Boehm, B., Davis, A., Humphrey, W.S., Leveson, N., Mead, N.R., Musa, J.D., Parnas, D.L., Pfleger, S.L., Weyuker, E. : "New Year's Resolution for Software Quality", Quality Time, J. Hayes Eds. IEEE Software 21, 1 (January/February 2004) pp. 12-13.
- [Berquist and Ljungberg2001] The power of gifts: organizing social relationships in open source communities, Information System Journal 11, pp.305.320, 2001.
- [Boeg2006] Boeg, J., 2006. Certifying Software Component Attributes. IEEE Software, 23(3), 74-81.
- [bonaccorsi2006] Andrea Bonaccorsi and Cristina Rossi. Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business. Knowledge, Technology, & Policy / Winter 2006.
- [CarvalloThesis] Carvallo, J.P.: "Systematic Construction of Quality Models for COTS-Based Systems". PhD Thesis. Technical University of Catalunya (UPC). GESSI Group. 2005
- [Fitzgerald2006] Fitzgerald B, Lero- The Irish Software Engineering Research Centre, University of Limerick, Limerik, Ireland. The Transformation of the Open Source Software. MIS Quarterly Vol. 30 No. 3/September 2006, pag 587-598.
- [FP97] N.E. Fenton, S.L. Pfleeger. "Software Metrics: A Rigorous and Practical Approach". PWS Publishing, Boston, second edition edition, 1997
- [FSF2010] consultado 2010. [cited; Available from http://www.fsf.org/licensing/licenses/index_html]
- [Gerea2007] Marnela G., M. Selection of Open Source Components – A Qualitative Survey in Norwegian IT Industry, Masther Thesis, 2007
- [Haugeetal2007] Oyvind Hauge, Carl-Fredrik Sorensen, Andreas Rosdal. Surveying Industrial Roles in Open Source Software Development
- [Hecker2000] Hecker, F. "Setting Up Shop: The Business of Open-Source Software," June 2000 (available online at <http://www.hecker.org/writings/setting-up-shop>).

- [ISO/IEC14598] ISO/IEC 14598-1:1999
- [Kitchenham1996] Kitchenham, B., Pfleeger, S.L.: Software Quality: The Elusive Target. IEEE Software, Vol. 13, No. 1. January 1996, pp 12-21.
- [Krippendorff,1980] Krippendorff, A., 1980. Content Analysis. Sage Publications, London.
- [Laurent2004] Laurent, A. M. S. (2004). Understanding Open Source & Free Software Licensing. O'Reilly. ISBN: 0-596-00581-4.
- [OS+01] [Madanmohan, T.R., De, R., Open Source Reuse in Commercial Firms. IEEE Software. 21(6) p. 62-69.]
- [OS+02] Consultado 2010 [cited; Available from: <http://www.gnu.org/philosophy/open-source-misses-the-point.es.html>]
- [Os+03] Consultado 2010 [cited; Available from: <http://www.gnu.org/philosophy/open-source-misses-the-point.es.html>]
- [OSI+04] Open Source Initiative. Consultado 2010 [cited; Available from: <http://www.opensource.org/>].
- [OS+05] The Open Source Definition. Consultado 2010 [cited; Available from: <http://opensource.org/docs/osd>].
- [OSlicenses2010] Open Source Definition. Consultado 2010 [cited; available from: <http://opensource.org/licenses/>]
- [Raymond1999] Raymond, E. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly, Sebastapol, CA, 1999.
- [Robson93] Robson, C.: Experiment, Design and Statistics in Psychology, 3rd edition, Penguin Books, London, England, 1994.
- [Robson2010] Robson, C., 2002. Real World Research: A Resource for Social Scientists and Practitioner-researchers. Second Edition. Blackwell Publishers Inc.
- [sjobergetal2007] Sjoberg Dag I.K, Tore Dyba and Magne Jorgensen. The Future of Empirical Methods in Software Engineering Research. FOSE 2007, IEEE Computer Society.
- [SQU] ISO/IEC FCD 25000, Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, Geneva: International Organization for Standardization, 2004.
- [Suryn2003] Suryn, W.; Abran A: "ISO/IEC SQuaRE: The Second Generation of Standards for Software Product Quality," 7th IASTED International Conference on Software Engineering and Applications, California, USA. 2003.
- [torchiano2004a] Torchiano, M., Morisio, M., 2004. Overlooked Aspects of COTS-Based Development. IEEE Software, 21(2), 88-93.
- [VIM93] ISO "International vocabulary of basic and general terms in metrology (VIM)." 1993.

- [Vitharana2003] Vitharana, P., Zahedi, F., Jain, H.: "Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and Empirical Analysis". IEEE Transactions on Software Engineering. Vol. 29(7), 2003, pp 649-664.
- [Voas04] J. Voas, "Software's secret sauce: the "-ilities" [software quality]." Software, IEEE 21(6): 14. 2004.
- [Voas1998] Voas, J.: The Software Quality Certification Triangle. CrossTalk, Nov 1998, pp. 12-14.
- [Webbing2003] Webbink, M. H. (2003). Understanding Open Source Software. New South Wales Society For Computers and the Law, 51. Available online: http://www.nswscl.org.au/journal/51/Mark_H_Webbink.html accessed 2010-07-26.
- [Wohlinetal2000] Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A., 2000. Experimentation in Software Engineering - An Introduction. Kluwer Academic Publishers.

Mozilla Firefox

- [MF+01] Página principal Mozilla Firefox. Consultado 2010 [cited; Available from: <http://www.mozilla-europe.org/es/firefox>]
- [MF+02] Organizaciones de Mozilla. Consultado 2010 [cited; Available from: <http://www.mozilla.org/about/organizations.html>]
- [MF+03] Historia del Producto. Consultado 2010 [cited; Available from: http://es.wikipedia.org/wiki/Mozilla_Firefox#Historia]
- [MF+03a] Historial de versiones. Consultado 2010 [cited; Available from: http://es.wikipedia.org/wiki/Mozilla_Firefox#Historial_de_las_versiones]
- [MF+04] Soporte para usuarios. Consultado 2010 [cited; Available from: <http://support.mozilla.com/en-US/kb/>]
- [MF+05] Foro. . Consultado 2010 [cited; Available from: <http://support.mozilla.com/en-US/forum/>]
- [MF+06] Chat. . Consultado 2010 [cited; Available from: http://support.mozilla.com/en-US/kb/Live+Chat?new_chat]
- [MF+07] Wiki. . Consultado 2010 [cited; Available from: <https://wiki.mozilla.org/>]
- [MF+08] Mozilla Developer Center. Consultado 2010 [cited; Available from: <https://developer.mozilla.org/En>]
- [MF+09] Requisitos. Consultado 2010 [cited; Available from: <http://www.mozilla-europe.org/es/firefox/system-requirements/>]
- [MF+10] Lenguajes de programación utilizados según el portal web del componente. Consultado 2010 [cited; Available from: https://developer.mozilla.org/en/Web_Development]
- [MF+11] Lenguajes de programación utilizados según Ohloh. . Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/firefox/analyses/latest>]
- [MF+12] Código fuente. Consultado 2010 [cited; Available from: https://developer.mozilla.org/en/Download_Mozilla_Source_Code]
- [MF+13] Características de Firefox. Consultado 2010 [cited; Available from: <http://www.mozilla-europe.org/es/firefox/features/>]
- [MF+14] Test SunSpider. Consultado 2010 [cited; Available from: <http://www2.webkit.org/perf/sunspider-0.9/sunspider.html>]
- [MF+15] Licencia Firefox. Consultado 2010 [cited; Available from: <http://www.mozilla.org/MPL/>]
- [MF+16] FAQ. Consultado 2010 [cited; Available from: <http://www.mozilla.org/about/faq.html>]

Agilo

- [Agilo+01] Scrum. <http://es.wikipedia.org/wiki/Scrum>
- [Agilo+02] Información de Agilo. Consultado 2010 [cited; Available from: http://en.wikipedia.org/wiki/Agilo_for_Scrum]
- [Agilo+03] Empresas usando el componente. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/pages/references/>]
- [Agilo+04] Blog. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/blog/>]
- [Agilo+05] Grupo de Google. Consultado 2010 [cited; Available from: <http://groups.google.com/group/agilo/?pli=1>]
- [Agilo+06] Sección de soporte. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/pages/support/>]
- [Agilo+07] Requisitos de Software. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/pages/download-install/>]
- [Agilo+08] Lenguajes de programación en Ohloh. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/agilo-scrum/analyses/latest>]
- [Agilo+09] Descarga del código fuente. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/pages/agilo/>]
- [Agilo+10] Licencia Apache 2. Consultado 2010 [cited; Available from: http://www.ohloh.net/licenses/apache_2]
- [Agilo+11] Soporte para Integradores. Consultado 2010 [cited; Available from: <https://dev.agile42.com/>]
- [Agilo+12] FAQ. Consultado 2010 [cited; Available from: <http://www.agile42.com/cms/pages/faq/>]
- [Agilo+13] Ficheros de ayuda incluidos en el componente. Consultado 2010 [cited; Available from: <https://dev.agile42.com/agilo-help/>]

Fluent NHibernate

- [Fluent+01] Descarga del componente. Consultado 2010 [cited; Available from: <http://fluentnhibernate.org/downloads>]
- [Fluent+02] Blog. Consultado 2010 [cited; Available from: <http://fluentnhibernate.org/blog/index>]
- [Fluent+03] Foro. Consultado 2010 [cited; Available from: <http://support.fluentnhibernate.org/discussions>]
- [Fluent+04] Wiki. Consultado 2010 [cited; Available from: <http://support.fluentnhibernate.org/discussions>]
- [Fluent+05] Grupo de Google. Consultado 2010 [cited; Available from: <http://groups.google.com/group/fluent-nhibernate?pli=1>]
- [Fluent+06] Página principal. Consultado 2010 [cited; Available from: <http://fluentnhibernate.org/>]
- [Fluent+07] Instalación del componente. Consultado 2010 [cited; Available from: http://wiki.fluentnhibernate.org/Getting_started#Installation]
- [Fluent+08] Introducción al componente. Consultado 2010 [cited; Available from: http://wiki.fluentnhibernate.org/Getting_started]
- [Fluent+09] Lenguajes de programación en Ohloh. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/fluent-nhibernate/analyses/latest>]
- [Fluent+10] FAQ. Consultado 2010 [cited; Available from: <http://wiki.fluentnhibernate.org/FAQs>]

Gimp

- [Gimp+01] Página principal. Consultado 2010 [cited; Available from: <http://www.gimp.org/>]
- [Gimp+02] Información de Gimp. Consultado 2010 [cited; Available from: <http://es.wikipedia.org/wiki/GIMP>]
- [Gimp+03] Listas de correos. Consultado 2010 [cited; Available from: http://www.gimp.org/mail_lists.html]
- [Gimp+04] Chat IRC. Consultado 2010 [cited; Available from: <http://www.gimp.org/irc.html>]
- [Gimp+05] Requisitos Hardware para Windows. Consultado 2010 [cited; Available from: <http://www.gimp.org/windows/>]
- [Gimp+06] Lenguajes de programación en Ohloh. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/gimp/analyses/latest>]
- [Gimp+07] Documentación para integradores. Consultado 2010 [cited; Available from: <http://developer.gimp.org/>]
- [Gimp+08] Código fuente. Consultado 2010 [cited; Available from: <http://git.gnome.org/browse/gimp/>]
- [Gimp+09] Licencia. Consultado 2010 [cited; Available from: <http://www.gimp.org/about/COPYING>]
- [Gimp+10] Documentación para usuario. Consultado 2010 [cited; Available from: <http://docs.gimp.org/>]
- [Gimp+11] FAQ para usuarios. Consultado 2010 [cited; Available from: <http://www.gimp.org/docs/userfaq.html>]
- [Gimp+12] FAQ para desarrolladores. Consultado 2010 [cited; Available from: <http://developer.gimp.org/faq.html>]
- [Gimp+13] Descarga de archivos de ayuda. Consultado 2010 [cited; Available from: <ftp://ftp.gimp.org/pub/gimp/help/>]

Grub

- [Grub+01] Página principal. Consultado 2010 [cited: Available from: <http://www.gnu.org/software/grub/index.html>]
- [Grub+02] Versiones del componente. Consultado 2010 [cited; Available from: <ftp://alpha.gnu.org/gnu/grub/>]
- [Grub+03] Manual de usuario. Consultado 2010 [cited; Available from: <http://www.gnu.org/software/grub/manual/>]
- [Grub+04] Listas de correos. Consultado 2010 [cited; Available from: <http://www.gnu.org/software/grub/grub-mailinglist.en.html>]
- [Grub+05] Wiki. Consultado 2010 [cited; Available from: <http://grub.enbug.org/>]
- [Grub+06] Contenidos de ayuda. Consultado 2010 [cited; Available from: <http://grub.enbug.org/HelpContents>]
- [Grub+07] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/grub/analyses/latest>]
- [Grub+08] Acceso al código fuente. Consultado 2010 [cited; Available from: <ftp://alpha.gnu.org/gnu/grub/>]
- [Grub+09] FAQ. Consultado 2010 [cited; Available from: <http://www.gnu.org/software/grub/grub-faq.en.html>]

Iptables

- [Iptables+01] Página principal. Consultado 2010 [cited; Available from: <http://www.netfilter.org/projects/iptables/index.html>]
- [Iptables+02] Zona de descarga. Consultado 2010 [cited; Available from: <http://www.netfilter.org/projects/iptables/downloads.html>]
- [Iptables+03] Historia del grupo Netfilter. Consultado 2010 [cited; Available from: <http://www.netfilter.org/about.html#history>]
- [Iptables+04] Listas de correo. Consultado 2010 [cited; Available from: <http://www.netfilter.org/maillinglists.html>]
- [Iptables+05] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/iptables/analyses/latest>]
- [Iptables+06] Documentación. Consultado 2010 [cited; Available from: <http://www.netfilter.org/documentation/index.html>]
- [Iptables+07] Acceso online del código fuente. Consultado 2010 [cited; Available from: <ftp://ftp.netfilter.org/pub/iptables/snapshot/>]
- [Iptables+08] Licencia. Consultado 2010 [cited; Available from: <http://www.netfilter.org/licensing.html>]

Joomla!

- [Joomla+01] Página principal. Consultado 2010 [cited; Available from: <http://www.joomla.org>]
- [Joomla+02] Open Source Matters. Consultado 2010 [cited; Available from: <http://opensourcematters.org/index.php>]
- [Joomla+03] Joomla Hosting. Consultado 2010 [cited; Available from: <http://www.rochenhost.com/joomla-hosting>]
- [Joomla+04] Información acerca de Joomla!. Consultado 2010 [cited; Available from: <http://www.joomla.org/about-joomla.html>]
- [Joomla+05] Historia del proyecto. Consultado 2010 [cited; Available from: http://es.wikipedia.org/wiki/Joomla!#Historia_del_proyecto]
- [Joomla+06] Foro. Consultado 2010 [cited; Available from: <http://forum.joomla.org/>]
- [Joomla+07] Requisitos técnicos. Consultado 2010 [cited; Available from: <http://www.joomla.org/technical-requirements.html>]
- [Joomla+08] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/joomla/analyses/latest>]
- [Joomla+09] Documentación. Consultado 2010 [cited; Available from: <http://docs.joomla.org/>]
- [Joomla+10] Consulta online del código fuente. Consultado 2010 [cited; Available from: <http://joomlancode.org/gf/project/joomla/scmsvn/?action=browse&path=/development/releases/1.5/>]
- [Joomla+11] Licencia. Consultado 2010 [cited; Available from: http://opensourcematters.org/index.php?option=com_content&view=article&id=56&Itemid=155]
- [Joomla+12] Zona de desarrolladores. Consultado 2010 [cited; Available from: <http://developer.joomla.org/>]
- [Joomla+13] FAQ. Consultado 2010 [cited; Available from: <http://docs.joomla.org/FAQs>]

MediaCoder

- [MediaCoder+01] Página Principal. Consultado 2010 [cited; Available from: <http://www.mediacoderhq.com/>]
- [MediaCoder+02] Información acerca de MediaCoder. Consultado 2010 [cited; Available from: <http://www.mediacoderhq.com/start/welcome.htm?p=19819&build=4710>]
- [MediaCoder+03] Foro. Consultado 2010 [cited; Available from: <http://forum.mediacoderhq.com/>]
- [MediaCoder+04] Soporte no gratuito. Consultado 2010 [cited; Available from: <http://www.mediacoderhq.com/premium/>]
- [MediaCoder+05] Zona de descarga del componente. Consultado 2010 [cited; Available from: <http://www.mediacoderhq.com/download.htm>]
- [MediaCoder+06] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/mediacoder/analyses/latest>]
- [MediaCoder+07] Manual de usuario. Consultado 2010 [cited; Available from: http://wiki.mediacoderhq.com/index.php/Users_Guide]
- [MediaCoder+08] FAQ. Consultado 2010 [cited; Available from: http://wiki.mediacoderhq.com/index.php/MediaCoder_FAQ]
- [MediaCoder+09] MediaCoder en Ohloh. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/mediacoder>]

StatusNet

- [StatusNet+01] Página principal. Consultado 2010 [cited; Available from: <http://status.net/>]
- [StatusNet+02] Información acerca de la compañía. Consultado 2010 [cited; Available from: <http://status.net/company>]
- [StatusNet+03] Empresas utilizando el componente. Consultado 2010 [cited; Available from: <http://status.net/whos-using-statusnet>]
- [StatusNet+04] Sección de soporte. Consultado 2010 [cited; Available from: <http://status.net/support>]
- [StatusNet+05] Foro. Consultado 2010 [cited; Available from: <http://forum.status.net>]
- [StatusNet+06] Wiki. Consultado 2010 [cited; Available from: http://status.net/wiki/Main_Page]
- [StatusNet+07] Mecanismos de soporte no gratuitos . Consultado 2010 [cited; Available from: <http://status.net/enterprise-network>]
- [StatusNet+08] Información acerca del componente. Consultado 2010 [cited; Available from: <http://gitorious.org/statusnet/mainline/blobs/master/README>]
- [StatusNet+09] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://www.ohloh.net/p/statusnet/analyses/latest>]
- [StatusNet+10] Sección de desarrolladores. Consultado 2010 [cited; Available from: <http://status.net/open-source>]
- [StatusNet+11] Manual de usuario. Consultado 2010 [cited; Available from: <http://status.net/wiki/Installation>]
- [StatusNet+12] Documentación para desarrolladores. Consultado 2010 [cited; Available from: <http://status.net/wiki/Development>]
- [StatusNet+13] FAQ. Consultado 2010 [cited; Available from: <http://status.net/wiki/FAQ>]

Subclipse

- [Subclipse+01] Página principal. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org>]
- [Subclipse+02] Versiones de la versión 1.6.X. Consultado 2010 [cited; Available from: http://subclipse.tigris.org/subclipse_1.6.x/changes.html]
- [Subclipse+03] Miembros de la comunidad. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/servlets/ProjectMemberList>]
- [Subclipse+04] Cuestiones y ayuda. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/servlets/ProjectProcess?pageID=t65B1R>]
- [Subclipse+05] Foro. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/ds/viewForumSummary.do?dsForumId=1047>]
- [Subclipse+06] Mecanismos de soporte no gratuitos. Consultado 2010 [cited; Available from: <http://www.open.collab.net/support/support-programs/>]
- [Subclipse+07] Sección de descarga. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA>]
- [Subclipse+08] Lenguajes de programación. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA>]
- [Subclipse+09] Documentación para integradores. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/wiki/SubclipseDevelopment>]
- [Subclipse+10] Manual de instalación. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=1732&expandFolder=1732&folderID=0>]
- [Subclipse+11] Sección para desarrolladores. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/wiki/SubclipseDevelopment>]
- [Subclipse+12] FAQ. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/wiki/PluginFAQ>]
- [Subclipse+13] Wiki. Consultado 2010 [cited; Available from: <http://subclipse.tigris.org/wiki/>]