

**Solución de alta disponibilidad (HA) y balanceo de carga para el Servicio Web de la Red de Datos  
de la Universidad del Cauca**



**Jeimmy Viviana Cuellar Rivera**  
**José Raul Romero Mera**

**ANEXO F**

**Universidad del Cauca**  
**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Telecomunicaciones**  
**Grupo I+D Nuevas Tecnologías en Telecomunicaciones**  
**Popayán, Noviembre de 2012**

**Análisis de Modelos de Evaluación de Calidad de Software Libre**  
***Analysis of Quality assessment models of free software***

Alejandro Toledo Tovar MsC (C)  
Ingeniero en Electrónica y Telecomunicaciones

Jeimmy Viviana Cuellar Rivera  
José Raul Romero Mera  
Estudiantes de Ingeniería Electrónica y Telecomunicaciones

{atoledo, jcuellar, jrromero}@unicauca.edu.co

Grupo I+D Nuevas Tecnologías en Telecomunicaciones  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Universidad del Cauca  
Popayán - Colombia

**Resumen:** En los últimos años, el Software Libre ha ido ganando terreno en el mercado y muchas compañías muestran interés por su uso y ante la cantidad enorme de proyectos disponibles, se hace necesario contar con un método que permita hacer una evaluación de su calidad que facilite tomar decisiones sobre cual proyecto usar. El presente documento es el resultado de una revisión bibliográfica en el que se muestra algunos de los más importantes modelos existentes para la evaluación de proyectos de software libre a fin de determinar cuál de ellos puede ser utilizado en el proyecto en desarrollo: “Solución de Alta Disponibilidad (HA) y balanceo de carga para el Servicio Web de la Red de Datos de la Universidad del Cauca”.

**Abstract:** *In recent years, Free Software has been gaining ground in the market and many companies show interest in their use and given the huge amount of projects available, it is necessary to have a method to make a quality assessment to facilitate making decisions on which project to use. This document is the result of a literature review which shows some of the most important models for evaluation of existing free software projects to determine which of them can be used to the development project: “Troubleshooting High Availability (HA) and load balancing for Web Service Data Network at the University of Cauca”.*

**Palabras clave:** Software Libre, Código Abierto, Calidad, Modelo de Evaluación, OpenBRR, QSOS, QualOSS, QualiPso OMM.

**Key words:** *Free Software, Open Source, Quality, Assessment Framework, OpenBRR, QSOS, QualOSS, QualiPso OMM.*

## 1. INTRODUCCIÓN

El término "FLOSS" (Free/Libre and Open Source Software) es utilizado para referirse tanto al Software Libre, *Free Software*, definido por la *Free Software Definition* [1] de la *Free Software Foundation*, como al Software de Código Abierto, *Open Source Software*, definido por la "Open Source Definition" [2] de la *Open Source Initiative*, también FLOSS se define como las acciones que una persona puede realizar en una parte de software que ha recibido y que son respaldadas con los permisos otorgados por el titular de los derechos de autor mediante una licencia de distribución debidamente reconocida por *Free Software Foundation* y/o *Open Source Initiative* [3]. En el caso de las licencias del Software Libre, los usuarios tienen la libertad de ejecutar el programa para cualquier propósito, de estudiar y modificar el programa, y de distribuir copias, ya sea del programa original o modificado, gratis o por cobro de una tarifa por distribución [1]. El disponer del código fuente no implica la posibilidad de distribuir copias modificadas o la libertad de adaptarlo a nuestras necesidades (que sí posee el software libre) [4], esto puede ser restringido según el tipo de licencia que tenga el producto FLOSS [5],

En los últimos años, los proyectos FLOSS han ganado terreno en el mercado y muchas compañías muestran interés a pesar que aún se mantienen ciertos temores. Cada vez, las aplicaciones FLOSS se emplean más en el software de desarrollo y en los principales productos de las compañías [6]. Además, la cantidad disponible de proyectos FLOSS es enorme y, a menudo su calidad puede que no sea adecuada para la adopción en procesos de negocios reales, por tanto, las empresas deben analizar cuidadosamente las soluciones disponibles y elegir el software que se adapte a sus necesidades funcionales y de calidad [7], la cual es entendida como el "conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, portabilidad, usabilidad, seguridad e integridad" [8].

El desafío que se presenta es la elección de una determinada solución tecnológica software que cumpla con un requerimiento o necesidad en particular [7], en el caso del software libre [9] es aún más complejo, dados los mitos existentes en torno a él y a la dinámica en su desarrollo, que va ligada a una comunidad de desarrolladores y de usuarios, lo que hace imprescindible contar con un método que permita hacer una evaluación cualitativa mediante la integración de las características del software libre o de código abierto. Esto ha motivado que en los últimos años se realicen múltiples investigaciones que han permitido desarrollar unas iniciativas [6] en las que se han definido elementos más precisos para la toma de decisiones y provisión de mecanismos que permitan a las comunidades de desarrollo gestionar la calidad del desarrollo de sus proyectos FLOSS. Esto ha

ocurrido en razón a la imposibilidad que se presenta con los modelos tradicionales de calidad que se emplean en el desarrollo de software privativo (Normas ISO, modelo CMMI<sup>1</sup>, entre otras) [10]. Al menos se han identificado 20 iniciativas diferentes para la evaluación de proyectos FLOSS [6]. Tabla 1.

No.	Nombre	Año	Fuente	Orig.	Método
1	Capgemini Open Source Maturity Model	2003	[11]	I	Yes
2	Evaluation Framework for Open Source Software	2004	[12]	R	No
3	A Model for Comparative Assessment of Open Source Products	2004	[13], [14]	R	Yes
4	Navica Open Source Maturity Model	2004	[15]	I	Yes
5	Woods and Guliani's OSMM	2005	[16]	I	No
6	Open Business Readiness Rating (OpenBRR)	2005	[17], [18]	R/I	Yes
7	Atos Origin Method for Qualification and Selection of Open Source Software (QSOS)	2006	[19]	I	Yes
8	Evaluation Criteria for Free/Open Source Software Products	2006	[20]	R	No
9	A Quality Model for OSS Selection	2007	[21]	R	No
10	Selection Process of Open Source Software	2007	[22]	R	Yes
11	Observatory for Innovation and Technological transfer on Open Source software (OITOS)	2007	[23], [24]	R	Yes
12	Framework for OS Critical Systems Evaluation (FOCSE)	2007	[25]	R	No
13	Balanced Scorecards for OSS	2007	[26]	R	No
14	Open Business Quality Rating (OpenBQR)	2007	[27]	R	Yes
15	Evaluating OSS through Prototyping	2007	[28]	R	Yes
16	A Comprehensive Approach for Assessing Open Source Projects	2008	[29]	R	No
17	Software Quality Observatory for Open Source Software (SQO-OSS)	2008	[30]	R	Yes
18	An operational approach for selecting open source components in a software development project	2008	[31]	R	No
19	QualiPSo trustworthiness model	2008	[32], [33]	R	No
20	OpenSource Maturity Model (OMM)	2009	[34]	R	No

Tabla 1. Métodos, Modelos, Marcos de Referencia y/o enfoques de Evaluación de FLOSS. Tomado de [6].

<sup>1</sup> Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. [53]

En la Tabla 1, Las iniciativas se identifican en orden cronológico de su publicación, la columna “Fuente” refiere a los documentos que reportaron el método, la columna “Orig.” indica si la iniciativa proviene de la (I)ndustria o (R)esearch, Investigación; la columna “Método” indica si el método está bien definido con sus actividades necesarias, tareas, entradas y salidas o es un mero conjunto de criterios de evaluación. Se observa que en la Tabla 1 se incluyen metodologías, que son modelos y herramientas que permiten evaluar la calidad de un software o proyecto de FLOSS; algunas con un grado de desarrollo y aplicación importante y otras aún en fase de desarrollo [6].

## 2. DESCRIPCION DE ALGUNOS MODELOS DE EVALUACIÓN

De las iniciativas relacionadas en la Tabla 1, se describen los modelos de evaluación: OpenBRR, QSOS, SQO-OSS, QualiOSS y QualiPso OMM, dada la alta acogida que han generado dentro de la comunidad FLOSS, como a las reiteradas referencias evidenciadas en las distintas fuentes de consulta revisadas. También se tiene en cuenta los dos primeros modelos que surgieron y que fueron referentes para los demás. Estos modelos se agrupan en modelos de primera y segunda generación [35]:

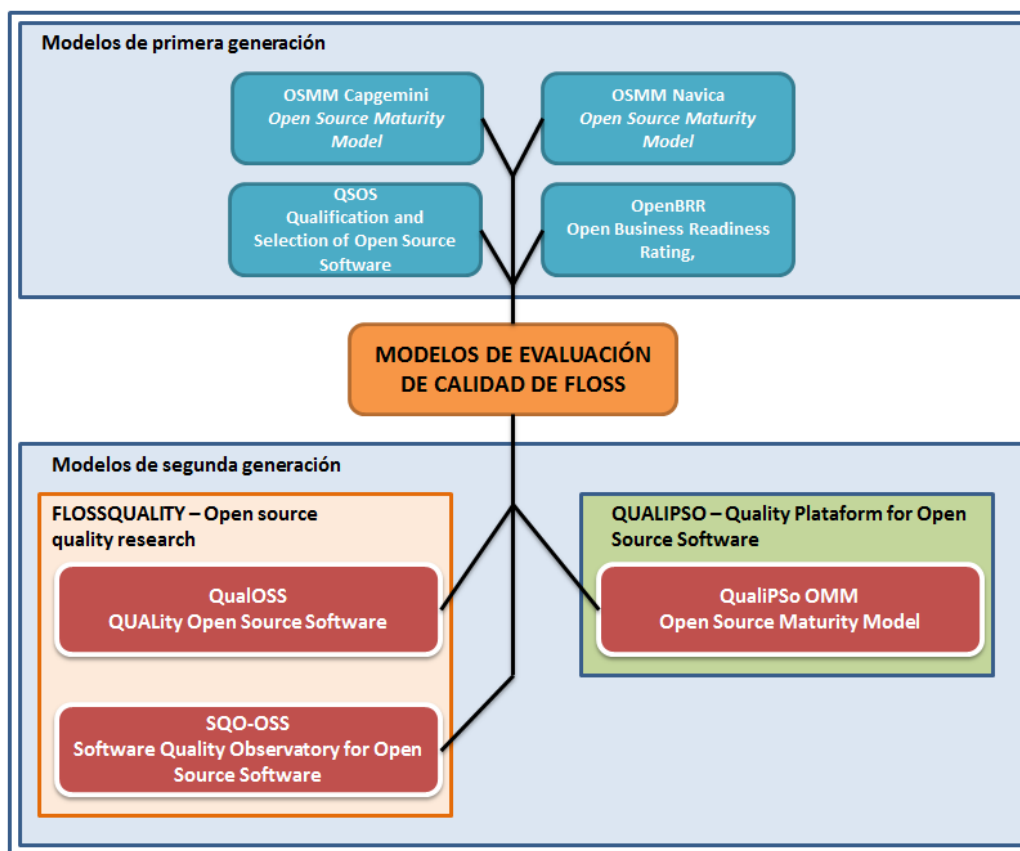


Figura 1. Algunos Modelos de Evaluación de Calidad de Software FLOSS

### 2.1. Modelos de Calidad de Software Libre de Primera Generación.

Entre los años 2003 y 2005 surgió la primera generación de modelos de evaluación de la calidad para software libre, estos se basaron sobre los modelos tradicionales aplicados en el software privativo con adaptaciones para el software libre, pertenecen a este conjunto:

### **2.1.1. *Open Source Maturity Model, OSMM Capgemini* [11]**

Este modelo es utilizado para asesoramiento independiente por la parte de la empresa Capgemini, el método tiene una licencia no-libre y se requiere una distribución autorizada [36]. Para la evaluación de un producto FLOSS, se emplean veintisiete indicadores, de los cuales, doce son los Indicadores de productos usados para determinar la madurez o adultez de un producto de FLOSS; se califican de 1, 3 y 5 [37] para determinar si un producto FLOSS es un producto inmaduro, o está en desarrollo hacia la madurez, o es suficiente maduro [38]. Estos se agrupan en cuatro diferentes grupos:

- **El producto** (Edad, licencias, las jerarquías humanas, puntos de venta, comunidad de desarrolladores).
- **Integración** (Modularidad, colaboración con otros productos, estándar).
- **Utilización** (Soporte, la facilidad de implementación).
- **Aceptación** (Comunidad de usuarios, penetración en el mercado).

Los quince restantes, son los Indicadores de Aplicaciones empleados para determinar la adaptación de las aplicaciones FLOSS a las necesidades de los clientes [38], como es: Usabilidad, Interfaz, Rendimiento, Fiabilidad, Seguridad, Tecnología probada, Independencia de proveedor, Independencia de la plataforma, Soporte, Reporte, Administración, Asesoramiento, Capacitación, Dotación de personal, Implementación. Se les califica en una escala de 1 a 5 donde 5 es "extremadamente importante" y 1 "no importante" [37].

Todos estos datos de calificación se combinan en una calificación 'final' que indica la idoneidad del producto para las demandas determinadas. Finalmente, determinar una única puntuación solo permite una fácil comparación entre productos de candidatos [37]

### **2.1.2. *Open Source Maturity Model, OSMM Navica***

El Modelo OSMM de Navica, determina la madurez de un producto FLOSS en tres fases [7]:

**Fase 1: Evaluación de la madurez del elemento de cada producto y asignación de una puntuación de madurez.** Se identifican los elementos claves del producto y evalúa su madurez, estos son: Productos de Software, Soporte, Documentación, Formación, Integración de productos y Servicios profesionales. Cada elemento es evaluado y se le asigna una calificación de madurez según el siguiente proceso:

- Definición del requerimiento organizacional

- Localización de los recursos
- Evaluación de la madurez de elemento
- Asignación de calificación puntuación de madurez en una escala de 1 a 10 al elemento.

**Fase 2: Definición de factores de ponderación para cada elemento según los requerimientos de la organización.** Se proporciona una lista, Tabla 2, de calificación con valores de ponderación propuestos por defecto por el modelo. Los valores por defecto pueden ser cambiados de acuerdo a las necesidades específicas pero debe mantenerse la suma de 10.

Categoría	Ponderación
Software	4
Soporte	2
Documentación	1
Capacitación	1
Integración	1
Servicios profesionales	1
<b>TOTAL</b>	<b>10</b>

Tabla 2. Asignación de valores ponderados a los factores en OSSM de Navica. Tomado de [39].

**Fase 3: Calculo de la calificación general de madurez del producto.** OSMM asigna una ponderación a la calificación de madurez de cada elemento permitiendo que cada elemento refleje su importancia a la madurez global del producto. La calificación ponderada de cada elemento se suma a proporcionar una calificación general de la madurez del producto. El modelo se suministra bajo la licencia *Academic Free License*. [55]

### 2.1.3. Qualification and Selection of Open Source Software QSOS [19]

Proporcionada por Atos Origin para calificar y seleccionar FLOSS bajo los términos de la licencia *GNU Free Documentation License* [56] [54]. Para el desarrollo de la evaluación se cuenta con una herramienta libre llamada O3S (*Open Source Software de Selección*)<sup>2</sup>. Se distinguen cuatro pasos:

- Definición:** Identifica los factores que puedan ser considerados en los siguientes pasos, se evalúan entre otros el tipo de licencia tanto actual como futura (En caso del desarrollo de producto nuevo), y tipos de comunidades desarrolladoras.
- Evaluación:** Recopila información desde la comunidad FLOSS, con el fin de: Construir la Tarjeta de Identidad (IC) para cada software y la hoja de evaluación del software con información general, disponibilidad de servicios,

<sup>2</sup> A disposición de la comunidad en el sitio Web <http://www.qsos.org> para crear, modificar y consultar documentos de identidad y hojas de evaluación.

especificaciones físicas y técnicas, entre otras. Los aspectos de calidad son ponderados en un rango de 0 a 2.

- C. Calificación:** Definición de los filtros de la traducción de las necesidades y limitaciones relacionadas con la selección de software de código libre o abierto en un contexto específico. Se logra calificar el contexto del usuario más adelante será usado, se establece parámetros como la necesidad del usuario de una familia o compatibilidad con un único sistema operativo, permitiendo filtrar lo que es realmente importante. La herramienta O3S permite la definición de estos filtros personalizados.
- D. Selección:** Identificación del software que cumplan los requisitos del usuario o comparación del software de la misma familia esto se logra aplicando el filtro establecido en el paso C, "Calificación" de los datos, facilitados por los dos primeros pasos, con el fin de continuar las consultas, comparaciones y selección de productos.

#### **2.1.4. Open Business Readiness Rating, OpenBRR**

Proporcionado por "Carnegie Mellon West Center for Open Source investigation" y patrocinado por O'Reilly CodeZoo, SpikeSource, e Intel, disponible bajo la licencia "Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License", destinado a ayudar a los administradores TI a determinar qué FLOSS sería el más adecuado para sus necesidades. Se proponen estandarizaciones y agrupación en categorías de los diferentes tipos de datos de evaluación. Consta de 4 fases [18]:

##### **Fase 1: Quick Assessment Filter - Evaluación Rápida**

- Identificación de una lista corta inicial de los productos de software a evaluar.
- Medición de cada producto de software contra los criterios de evaluación rápida.
- Eliminación de cualquier producto de software de la lista que no cumpla con el requerimiento de usuario.

**Fase 2: Target Usage Assessment - Clasificación y ponderación de los criterios de selección:** Clasificación de las 12 categorías de acuerdo a su importancia (1 – mayor, 12 – menor) y elección de 7 (o menos) categorías para luego asignar un porcentaje de importancia a cada una, de tal forma que se totalice 100% sobre las categorías elegidas. Para cada métrica dentro de una categoría, se sitúa la métrica según importancia a disposición de empresas. Por otra parte, ante la base de aportes de la comunidad y comentarios del público sobre el RFC, según referencia [17], se elaboró un borrador en el que refleja la retroalimentación, y en el que se reducía las categorías de 12 a 7. [18] [40]

**Fase 3: Data Collection & Processing - Recopilación y procesamiento de datos:** Para cada métrica utilizada en la calificación de cada categoría se calcula la ponderación aplicada para cada una. Incluye la normalización de las métricas, clasificación de categoría, factores de ponderación y mediciones de la funcionalidad de procesamiento.



**Fase 4: Data Translation - Cálculo y publicación de los resultados:** Clasificación de la categoría de uso de traducción de datos y la orientación funcional; y la ponderación de factores para calcular la puntuación de calificación de disposición de negocios.

## **2.2. Modelos de Calidad de Software Libre de Segunda Generación**

Recientemente, ha surgido una ola de segunda generación de modelos de calidad del software libre, los cuales están basados en las metodologías tanto en los modelos tradicionales de calidad como los de primera generación del software libre. La principal diferencia es la amplia cantidad de herramientas de soporte o apoyo. Los modelos de calidad de segunda generación son:

### **2.2.1. Modelo de Calidad “QUALity of Open Source Software – QualOSS”**

Es uno de los proyectos financiados por la Comisión Europea dentro del 6to programa marco [41] en relación con las métricas de software libre y de calidad, coordinado por el CETIC con participación de ocho socios de 5 países europeos: Bélgica, Francia, Alemania, España y los Países Bajos [42] para evaluar la robustez (capacidad de manejar los problemas) y la capacidad de evolución (la capacidad para seguir siendo viable en el largo plazo) de las diversas iniciativas FLOSS para mejorar la posición competitiva de la industria europea del software, proporcionando metodologías y herramientas para mejorar la productividad y la calidad de sus productos de software. El método de evaluación QualOSS toma ventaja de la amplia información disponible en repositorios de FLOSS como los datos del software y otros datos producidos por la comunidad de desarrolladores. [43].

QualOSS usa los GQM (Goal-Question-Metrics) [39], [43], [44] y la ejecución de trazabilidad garantizan que un método de evaluación QualOSS sea riguroso. Sin embargo, un método de evaluación QualOSS debe ser documentado y las evaluaciones deben mantener un seguimiento detallado de sus distintas acciones. Así, los resultados evaluación QualOSS son considerados solamente válidas si se remonta a la entrada original y si todo el proceso para obtener los resultados se encuentre documentado. El proceso de Evaluación estándar se compone de cinco pasos [43]:

- 1. Inicio:** identificación de la razón del negocio para la evaluación.
- 2. Configuración de la evaluación:** Identificación de datos, herramientas, personas, flujos de trabajos que se involucran en una evaluación.
- 3. Recolección y análisis de datos**
- 4. Interpretación de los datos.**
- 5. Supervisión de la evaluación.**

### **2.2.2. QualiPSo Open Source Maturity Model (OMM)**

Modelo perteneciente a QualiPSo, uno de los proyectos financiados por la Comisión Europea dentro del 6to programa marco [41] para desarrollar una plataforma de calidad de software de código abierto centrado en la confianza y en la calidad de los sistemas de código abierto. Esta metodología está basada en el modelo CMMI y esta liberado bajo la licencia Creative Commons. El objetivo del proyecto es permitir a las empresas de software, emplear software libre en la producción y en sus principales productos a fin de aumentar el interés y el número de contribuciones a los proyectos de software libre. [45], [46].

OMM contiene todos los elementos que han de ser evaluados, también tiene un conjunto de normas y directrices que describen cómo llevar a cabo los procesos de evaluación, por lo que se le denomina indistintamente modelo y metodología.

Los elementos de confianza (TWE) incluidos en OMM fueron recopilados o inspirado en dos fuentes: Los FLOSS-TWEs obtenidos de una amplia encuesta realizada a los desarrolladores, usuarios e integradores de FLOSS. [34] y Áreas de proceso CMMI.

OMM está organizado en niveles de madurez, cada nivel está en base al nivel inferior e incluye sus elementos de confianza (TWE) y son:

1. **Nivel Básico:** Que puede ser alcanzado fácilmente por la adopción de unas pocas prácticas necesarias en el proceso de desarrollo de FLOSS.
2. **Nivel intermedio:** Que puede lograrse mediante el cumplimiento de todos los elementos de confianza (TWE) del nivel básico y requiere elementos de confianza (TWE) del nivel intermedio.
3. **El nivel Avanzado,** es el nivel más alto que los proyectos de FLOSS pueden lograr por cumplimiento de todos los elementos de confianza (TWE) a partir de elementos confianza de los niveles básicos y media y los requeridos del nivel avanzado.

### **2.2.3. Software Quality Observatory for Open Source Software (SQO-OSS)** [47]

Proyecto financiado parcialmente por la Comisión Europea dentro del 6to programa marco [41]. Fue construido para su uso en el sistema Alitheia, una herramienta de evaluación de la calidad basada en OSGi, como un sistema de apoyo de decisión basado en mediciones, por lo que la automatización con respecto a la clasificación del software en la evaluación es una de las primeras prioridades. La plataforma de SQO-OSS tiene como objetivo combinar procesos y métricas de producto, con nuevos indicadores de calidad extraídos de base de datos que actualmente están infrautilizadas. En el proceso de construcción del modelo, se ha utilizado una versión simplificada del proceso Goal-Question-Metric (GQM) [30], [48], [49]. El resultado es una vista de árbol jerárquico de los atributos de calidad, cuyas hojas se analizan más en métricas medidas por el sistema y

utilizadas para la evaluación de los criterios seleccionados en una simple vista y que es agregar las mediciones. Se usa el perfil basado en procesos de evaluación. El perfil basado en la agregación permite categorizar la calidad de software en cuatro categorías: Excellent (E), Good (G), Fair (F) and Poor (P) (or as an ordinal scale E>G>F>P), para lo cual se construye perfiles con ciertos valores de medidas [50].

La implementación de SQO-OSS se basa en una arquitectura de plugin, que consta de componentes que se comunican a través de una bodega de datos común. Todo lo que puede ser un complemento será un plug-in, lo cual, significa que se tiene:

- Plugins para el cálculo de indicadores, procesamiento de datos y procesamiento de resultados
- Un plugin para el gestor de la inserción, eliminación, y las actualizaciones de plugins
- Una base de datos para hacer el seguimiento de los plugins, las mediciones y los datos en bruto.

SQO-OSS pretende correlacionar los datos de OSS de diversas fuentes de información con las características de calidad [47]: Funcionalidad, confiabilidad, usabilidad, eficiencia, mantenimiento y portabilidad. También tiene la intención de cooperar con otros proyectos relacionados, permitiendo que todas las partes involucradas obtengan resultados más rápidos, uniendo esfuerzos con el fin de aprovechar el crecimiento y la adopción de software libre [47].

### **3. ELECCIÓN DE LA METODOLOGÍA**

Para la elección de la metodología de evaluación de software libre se ha efectuado una revisión de las metodologías más importantes referenciadas en las diferentes fuentes de consulta desde 2005 al 2011, al encontrar una gran cantidad de ellas, se hace necesario contar con una forma que permita escoger un modelo o metodología, en la revisión de consulta se encontró una propuesta, referencia [6] - *OSS Framework fOr Comparing Open Source software Evaluation Methods* (FOCOSEM), según los proponentes, el objetivo de este marco no es hacer ningún juicio sobre la calidad de los métodos de evaluación de FLOSS existentes sino el de ofrecer ideas que ayuden a los profesionales a seleccionar un método adecuado de evaluación de FLOSS, además se ha mencionado que en la medida que se vaya avanzando en las evaluaciones se estará realizando ajustes necesarios como la adición o retiro de elementos al método propuesto. Esta iniciativa aún se encuentra en desarrollo y no han publicado avances o resultados obtenidos, que forman parte de un trabajo de Tesis Doctoral. Tomando este marco con algunas modificaciones y algunos estudios comparativos [51], [52] se efectúa un cuadro comparativo con las metodologías: QualiPSo OMM, QualOSS, Open

BRR y QSOS que se muestra en la Tabla 3., a fin de clarificar algunas de las ventajas que podría presentar para el proyecto “Solución de Alta Disponibilidad (HA) y balanceo de carga para el Servicio Web de la Red de Datos de la Universidad del Cauca”.

Componente	Elemento	Descripción Breve	QSOS	OpenBRR	QUALOSS	OMM
Contexto del Método	Objetivo específico	¿Cuál es el objetivo particular del método?	Evaluar, seleccionar y comparar FLOSS de forma objetiva, trazable con argumento sostenido.	Permitir a toda la comunidad (empresas, integradores y desarrolladores) ponderar el Software de manera abierta y estandarizada.	Permitir la comparación de productos de software libre de forma semi-automática, sencilla y rápida.	Proporcionar una base para el desarrollo de productos de manera eficiente y confiable para las comunidades FLOSS y una base para evaluar los procesos empujados por las comunidades desarrolladoras para las empresas integradoras o integradores
	Evaluación de la Funcionalidad	¿Es la política de funcionalidad parte del método de evaluación?	No	Si	Si	No
	Disponibilidad pública de resultados	¿Están las evaluaciones de los productos OSS almacenados en un repositorio de acceso público?	No, en repositorios. Si en la pagina web	No	No	Si, en repositorios.
	Relación con otros métodos	¿Cómo se relaciona el método con otros métodos? Es decir qué métodos se basan en este método?	CapGemisis OSMM y Navica OSMM	CapGemisis OSMM y Navica OSMM	QSOS, OpenBRR	CMMI, OpenBRR, QSOS
Usuario del Método	Habilidades requeridas	¿Qué habilidades necesita el usuario para emplear el método?	Conocimiento profundo sobre cada uno de los software a utilizar y sobre cada detalle necesario para la evaluación, además del conocimiento general del software libre para generar una adecuada puntuación del mismo	Requiere conocimiento profundo sobre cada uno de los software a utilizar y sobre cada detalle necesario para la evaluación, además del conocimiento general del software libre para generar una adecuada puntuación del mismo	Requiere conocimientos básicos de instalación de software libre y manejo de la herramienta propia del método	Entendimiento básico de los procesos de desarrollo y la calidad del software. Conocimiento en manejo de herramientas libres y open source.
	Usuarios potenciales	¿Quiénes son los usuarios potenciales del método?	Usuarios, desarrolladores e ingenieros de IT	Usuarios, desarrolladores e ingenieros de IT	Cualquier usuario alrededor del mundo	Usuarios, desarrolladores e ingenieros de IT, Compañías
Proceso del Método	Actividades del Método	¿Cuáles son las actividades del método evaluación y los pasos?	Son 4 pasos 1 - Definición y organización de lo que será evaluado 2 - Evaluación del software 3 - Calificación de su evaluación 4 - Selección	Fase 1 - Evaluación Rápida Fase 2 - Evaluación de uso objetivo Fase 3 - Recolección y procesamineto de datos Fase 4 - Traducción de datos	Son 5 pasos 1 - Inicio de la evaluación. 2 - Creación y planeación de la evaluación 3 - Recolección y procesamiento de datos 4 - Interpretación de resultados 5 - Supervisión de la evaluación	Pasos para escenario integración de los componentes de software libre: 1. Decisión a nivel de política corporativa/administración para incluir FLOSS en los productos de la empresa. 2. Proceso de selección. 3. Designación responsable para la evaluación y selección. 4. Documentación de los requisitos/funcionalidad a ser satisfecha por el componente de FLOSS propuesto.
	Número de criterios	¿Cuántos criterios se utilizan en la evaluación?		41	29	12
	Categoría de Evaluación	¿Cuáles son las categorías de criterios del método basados en las que se evalúa el producto OSS?	Las categorías de los criterios son: Durabilidad, Solucion Industrializada, Adaptabilidad Técnica, estrategia y proveedor del servicio.	Funcionalidad, Usabilidad, Calidad, Seguridad, Rendimiento, Escalabilidad, Arquitectura, Soporte, Documentacion, Adopcion, Comunidad y Profesionalismo	Las características de calidad son: Durabilidad, Solucion Industrializada, Adaptabilidad Técnica, estrategia y proveedor del servicio.	Tres (3) Niveles de Madurez: Básico, Intermedio, Avanzado
	Resultados	¿Cuáles son los resultados del método de evaluación?	Hojas de evaluación	Poderacion en Hojas electronicas	Poderacion en Hojas electronicas	Poderacion en Hojas electronicas
	Herramientas de Apoyo	¿Está el método de evaluación apoyado por una herramienta?	OSS (Open Source Selection Software)	Plantillas de evaluación	Se pretende contar con una herramienta final que automatize el proceso de evaluación	plantillas de evaluación online y offline. Herramienta de medición semiautomática está en la etapa de Prototipo.
Evaluación del Método	Validación	¿Ha sido validado el método evaluación?	El modelo está probado y validado en varios proyectos de software libre.	El modelo está probado y validado en varios proyectos de software libre.	El modelo está probado y validado en varios proyectos de software libre.	El modelo está probado y validado en varios proyectos de software libre.
	Etapas de madurez	¿Cuál es el estado de madurez del método de evaluación?	Llego a su madurez y fue abandonado.	Llego a su madurez y fue abandonado.	Inicial	Inicial

Tabla 2. Comparativo de algunos modelos de evaluación empleando el marco de comparación FOCOSEM

Al revisar el cuadro comparativo de la Tabla 3., se tiene que los modelos QualiPso OMM y QualOSS, los cuales, terminaron en el año 2010, no están maduros e incluso no existe una herramienta final como tal que obtenga automáticamente las métricas para el modelo de QualOSS, además, no todas las métricas pueden ser obtenidas automáticamente, como las relacionadas con la documentación por lo que hay que se debe buscar manualmente en el código fuente o en la página web del proyecto y otras métricas como las relacionadas con la comunidad que en su mayoría se han logrado obtenerse en forma automatizada deben efectuar una

inserción manual en la Base de Datos final. En el caso de QualiPso OMM, la herramienta de medición semiautomática aún está en la etapa de Prototipo, solo se dispone de un archivo de hoja de cálculo para hacer una evaluación manual del mismo modo que se hace para la evaluación CMMI; a pesar de que OMM se indica que es fácil y no se requiere de habilidades, se encuentra cierto grado de complejidad en algunas métricas de algunos TWE al carecer de un profundo conocimiento en ciertos temas, OpenBRR, a pesar de su abandono, ha sido la base de nuevas metodologías de evaluación y usado en varios proyectos desarrollados al interior de las compañías, las cuales, los consideran sensibles por lo que no están disponibles al público.

#### **4. RESULTADOS**

- Uno de los resultados del presente artículo es la tabla 3 en la cual se puede apreciar de manera resumida las diferentes virtudes de cada una de las principales metodologías de evaluación de calidad de software libre.
- Elección de una Metodología adecuada para la evaluación de Software libre en el marco del proyecto “Solución de Alta Disponibilidad (HA) y balanceo de carga para el Servicio Web de la Red de Datos de la Universidad del Cauca”, el cual se encuentra en la fase de implementación de la Solución.

#### **5. CONCLUSIONES**

Inicialmente, se encontró que la mejor opción es emplear la metodología OpenBRR, Tabla 3, para efectuar la evaluación de software libre, por lo siguiente:

- Adaptabilidad de la metodología: El hecho de permitir criterios propios para la evaluación de la funcionalidad permite adecuar y personalizar las necesidades específicas del proyecto a implementar.
- Cantidad de parámetros de evaluación mayor a 3: Hace que la evaluación sea más precisa dado que se puede hacer una ponderación más amplia.
- Facilidad de comprensión de los criterios a evaluar: Esto hace que la evaluación de las herramientas pueda desarrollarse de forma simple.
- Uso de evaluación rápida: Que permite desechar una herramienta antes de tener que realizar un proceso más largo, lo cual resulta útil para el ahorro de tiempo.
- Facilidad de manejo: Las herramientas entregadas son sencillas de usar y ponderar, haciendo que el trabajo sea mucho más ameno y sencillo

Por otra parte, cada vez es más amplio el uso de aplicaciones o proyectos FLOSS tanto en grandes, medianas y pequeñas industrias e incluso, aunque ciertamente en un menor grado, en el hogar; estas aplicaciones o proyectos se encuentran disponibles en gran cantidad de repositorios y diferentes sitios de internet de

donde pueden ser descargados. Ante la gran oferta presentada, existe la posibilidad de riesgos que dificultan la toma de decisiones favorables sobre cuál herramienta seleccionar, por lo que se debe tener una metodología de evaluación diferentes a los modelos tradicionales de calidad que se emplean en el desarrollo de software privativo (Normas ISO, modelo CMMI, entre otras), estos, tipos de software, están dirigidos a un modelo de desarrollo central presente en grandes empresas de desarrollo de software.

Finalmente, los modelos QualOSS y QualiPso OMM son recientes por lo que la información existente está relacionada con su uso a nivel interno de los proyectos que los concibieron y aún se están terminando de desarrollar ciertas herramientas que permitirían en gran porcentaje la medición de métricas de evaluación.

## 6. AGRADECIMIENTOS

Para culminar el presente artículo de la mejor manera, los autores expresan sus agradecimientos a la Universidad del Cauca.

## 7. REFERENCIAS

- [1] Free Software Foundation, "The Free Software Definition," GNU Project - Free Software Foundation (FSF). [En línea]. Disponible en: [HYPERLINK "http://www.gnu.org/philosophy/free-sw.html"](http://www.gnu.org/philosophy/free-sw.html). [Accesado: Abr. 30, 2011]
- [2] Open Source Initiative, "The Open Source Definition (Annotated)," Open Source Initiative. [En línea]. Disponible en: [HYPERLINK "http://www.opensource.org/osd.html"](http://www.opensource.org/osd.html). [Accesado: Abr. 30, 2011]
- [3] J. M. Gonzalez Barahona, Good practices for R&D projects producing FLOSS (Request for comments), GSyC/LibreSoft (URJC), Octubre 2010.
- [4] "¿Qué es el software libre?" [En línea]. Disponible en: [HYPERLINK "http://www.escuelapnud.org/files/pub\\_pages/about\\_us/attachments/software\\_libre\\_v2.pdf"](http://www.escuelapnud.org/files/pub_pages/about_us/attachments/software_libre_v2.pdf) [Accesado: Julio 9, 2011 ]
- [5] J. Gonzales Barahona, J. Seoane Pascual, y G. Robles, Introducción al software Libre, Universitat Oberta de Catalunya. [En línea]. Disponible en: [HYPERLINK "http://curso-sobre.berlios.de/introsobre/1.0/libre.pdf"](http://curso-sobre.berlios.de/introsobre/1.0/libre.pdf). [Accesado: Junio 22, 2011 ]
- [6] K. Stol y M. Ali Babar, "A comparison framework for open source software evaluation methods," en Open Source Software: New Horizons 6th International IFIP WG 2.13 Conference on Open Source Systems, OSS 2010, 1st ed., vol. 319. Notre Dame, IN, USA: Springer-Verlag, 2010, pp. 389-394.
- [7] B. Russo, M. Scotto, A. Sillitti, y G. Succi, Agile Technologies in open Source Development.: Hershey: IGI Global, 2010, p. 302.
- [8] O. Fernández, Carrasco, D. García, León, y A. Beltrán, Benavides, "Informes Técnicos: Un enfoque actual sobre la calidad del software". [Accesado: Abr. 15, 2011]
- [9] D. A. Wheeler, "How to Evaluate Open Source Software / Free Software (OSS/FS) Programs," David A. Wheeler's Personal Home Page. [En línea]. Disponible en: [HYPERLINK "http://www.dwheeler.com/oss\\_fs\\_eval.html"](http://www.dwheeler.com/oss_fs_eval.html). [Accesado: Abr. 30, 2011]

- [10] C. Montoya Becerra, "Evaluación de la Calidad de Proyectos de Software Libre," Blog de Hablemos de Software Libre, Abr. 30, 2009. [En línea]. Disponible en: [HYPERLINK "http://soflibreocmb.blogspot.com/2009/04/evaluacion-de-la-calidad-de-proyectos.html"](http://soflibreocmb.blogspot.com/2009/04/evaluacion-de-la-calidad-de-proyectos.html). [Accesado: Abr. 23, 2011]
- [11] F. Duijnhouwer y C. Widdows, "Open Source Maturity Model," Capgemini Expert Letter, 2003.
- [12] T. Koponen y V. Hotti, "Evaluation framework for open source software," en Proc. Software Engineering and Practice (SERP), 2004, Las Vegas, Nevada, USA.
- [13] G. Polančič y R.V. Horvat, "A Model for Comparative Assessment Of Open Source Products," en Proc. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, 2004, Orlando, USA.
- [14] G. Polančič, R.V. Horvat, y T. Rozman, "Comparative assessment of open source software using easy accessible data," en Proc. 26th International Conference on Information Technology Interfaces, Junio 7-10, 2004, Cavtat, Croatia, pp. 673 - 678.
- [15] B. Golden, Succeeding with Open Source. Addison-Wesley, 2004.
- [16] D. Woods y G. Guliani, Open Source for the Enterprise: Managing Risks Reaping Rewards: O'Reilly Media, Inc., 2005.
- [17] A. I. Wasserman, M. Pal, y C. Chan, "Business Readiness Rating Project", "BRR Whitepaper 2005 RFC 1". [En línea]. Disponible en: [HYPERLINK "http://pascal.case.unibz.it/retrieve/1095/BRR\\_whitepaper\\_2005RFC1.pdf"](http://pascal.case.unibz.it/retrieve/1095/BRR_whitepaper_2005RFC1.pdf). [Accesado: Abril 12, 2011 ]
- [18] A. I. Wasserman, M. Pal, y C. Chan, "The Business Readiness Rating Model: an Evaluation Framework for Open Source," en Proc. Of Workshop on Evaluation Techniques for Open Source Software, 2nd Int'l. Conf. on Open Source Systems, 2006, Como, Italia.
- [19] Atos, Origin. "Method for Qualification and Selection of Open Source software (QSOS) version 1.6," qsos.org, Oct. 23, 2006.
- [20] D. Cruz, T. Wieland, y A. Ziegler, "Evaluation criteria for free/open source software products based on project analysis," Software Process: Improvement and Practice, no. 2, Noviembre 2006.
- [21] W.J. Sung, J. H. Kim, y S. Y. Rhew, "A Quality Model for Open Source Software Selection," en Proc. Sixth International Conference on Advanced Language Processing and Web Information Technology, 2007, Luoyang, Henan, China, pp. 515-519.
- [22] Y. M. Lee, J. B. Kim, I. W. Choi, y S. Y. Rhew, "A Study on Selection Process of Open Source Software," en Proc. Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT), 2007, Luoyang, Henan, China.
- [23] M. Cabano, C. Monti, y G. Piancastelli, "Context-Dependent Evaluation Methodology for Open Source Software," en Proc. Third IFIP WG 2.13 International Conference on Open Source Systems (OSS 2007), 2007, Limerick, Ireland, pp. 301-306.
- [24] Assessment of the degree of maturity of Open Source open source software. [En línea]. Disponible en: [HYPERLINK "http://www.oitos.it/opencms/opencms/oitos/Valutazione\\_di\\_prodotti/Modello1.2.pdf"](http://www.oitos.it/opencms/opencms/oitos/Valutazione_di_prodotti/Modello1.2.pdf). [Accesado: Abril 12, 2011 ]
- [25] C. A. Ardagna, E. Damiani, y F. Frati, "FOCSE: An OWA-based Evaluation Framework for OS Adoption in Critical Environments," en Proc. Third IFIP WG 2.13 International Conference on Open Source Systems, 2007, Limerick, Ireland, pp. 3-16.
- [26] L. Lavazza, "Beyond Total Cost of Ownership: Applying Balanced Scorecards to Open-Source Software," en Proc. International Conference on Software Engineering Advances (ICSEA) Cap Esterel, 2007, French Riviera, France, p. 74.
- [27] D. Taibi, L. Lavazza, y S. Morasca, "OpenBQR: a framework for the assessment of OSS," en Proc. Third IFIP WG 2.13 International Conference on Open Source Systems (OSS 2007), 2007, Limerick, Ireland, pp. 173-186.

- [28] R. Carbon, M. Ciolkowski, J. Heidrich, I. John, y D. Muthig, "Evaluating Open Source Software through Prototyping," en Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives' (Information Science Reference, 2007), St. Amant y B. Still, Eds. Ed., 2007, pp. 269-281.
- [29] M. Ciolkowski y M. Soto, "Towards a Comprehensive Approach for Assessing Open Source Projects," en Software Process and Product Measurement.: Springer-Verlag, 2008.
- [30] I. Samoladas, G. Gousios, D. Spinellis, y I. Stamelos, "The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation," en Proc. Fourth IFIP WG 2.13 International Conference on Open Source Systems (OSS 2008), 2008, Milano, Italy.
- [31] A. Majchrowski y J. Deprez, "An operational approach for selecting open source components in a software development project," en Proc. 15th European Conference, Software Process Improvement (EuroSPI), Septiembre 3-5, 2008, Dublin, Ireland.
- [32] V. del Bianco, L. Lavazza, S. Morasca, y D. Taibi, "Quality of Open Source Software: The QualiPSo Trustworthiness Model," en Proc. Fifth IFIP WG 2.13 International Conference on Open Source Systems (OSS 2009), Junio 3-6, 2009, Skövde, Suecia.
- [33] V. del Bianco, L. Lavazza, S. Morasca, y D. Taibi, "The observed characteristics and relevant factors used for assessing the trustworthiness of OSS products and artefacts," Skövde, Suecia, Technical Report A5.D1.5.3, 2008.
- [34] E. Petrinja, R. Nambakam, y A. Sillitti, "Introducing the OpenSource Maturity Model," en Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009. FLOSS '09. ICSE Workshop on, 2009, Vancouver, Canadá, pp. 37 - 41.
- [35] K. Haaland, A. Groven, R. Glott, y A. Tannenber, "Free/Libre Open Source Quality Models- a comparison between two approaches," en 4th FLOSS International Workshop on Free/Libre Open Source Software, Julio 2010, Jena, Thuringia, Germany.
- [36] B. Russo, M. Scotto, A. Sillitti, y G. Succi, Agile Technologies in open Source Development. Hershey: IGI Global, 2010, p. 304.
- [37] National Resource Centre for FOSS, Open Source Maturity Model for a Department/Enterprise/Institution. [En línea]. Disponible en: [HYPERLINK "http://www.nrcfoss.aukbc.org.in/maturity/"](http://www.nrcfoss.aukbc.org.in/maturity/). [Accesado: Junio 21, 2011]
- [38] CapGemini, Open Source Maturity Model: een selectie voor een open wereld, Septiembre 2008. [En línea]. Disponible en: [HYPERLINK "http://www.nl.capgemini.com/expertise/publicaties/open-source-maturity-model-eeen-selectie-voor-eeen-open-wereld/"](http://www.nl.capgemini.com/expertise/publicaties/open-source-maturity-model-eeen-selectie-voor-eeen-open-wereld/). [Accesado: Junio 21, 2011]
- [39] J. A. J. Wilson, Open Source Maturity Model, OSS Watch, Enero 2011. [En línea]. Disponible en: [HYPERLINK "http://www.oss-watch.ac.uk/resources/osmm.xml?style=text"](http://www.oss-watch.ac.uk/resources/osmm.xml?style=text). [Accesado: Junio 21, 2011]
- [40] A. Das y A. I. Wasserman, "Using FLOSSmole Data in Determining Business Readiness Ratings," en Workshop on Public Data about Software Development - WoPDaSD 2007, 2007, Limerick, Irlanda.
- [41] European Commission, CORDIS FP6: What is FP6. [En línea]. Disponible en: [HYPERLINK "http://cordis.europa.eu/fp6/whatisfp6.htm"](http://cordis.europa.eu/fp6/whatisfp6.htm). [Accesado: Junio 20, 2011]
- [42] Centre d'Excellence en Technologies de l'Information et de la Communication - CETIC, QualOSS QUALity in Open Source Software. [En línea]. Disponible en: [HYPERLINK "http://www.cetic.be/article500.html"](http://www.cetic.be/article500.html). [Accesado: Junio 20, 2011]
- [43] J. Deprez, QualOSS Final Activity Report, Centre d'Excellence en Technologies de l'Information et de la de l'Information et de la, Enero 2010. [En línea]. Disponible en: [HYPERLINK "http://www.qualoss.org/deliverables/QualOSS\\_Final\\_Activity\\_Report\\_Submitted.pdf"](http://www.qualoss.org/deliverables/QualOSS_Final_Activity_Report_Submitted.pdf). [Accesado: Junio 21, 2011]



- [44] J. Tabilo, GMQ - Gestión de Software. [En línea]. Disponible en: [HYPERLINK "http://www.fing.edu.uy/inco/cursos/gestsoft/Presentaciones/GQM%20-%20G9/GQM.doc"](http://www.fing.edu.uy/inco/cursos/gestsoft/Presentaciones/GQM%20-%20G9/GQM.doc). [Accesado: Julio 2011, 18 ]
- [45] M. Wittmann y R. Nambakam, Deliverable A6.D1.6.3 CMM-like model for OSS, QualiPSO project web site, Noviembre 2008. [En línea]. Disponible en: [HYPERLINK "http://www.qualipso.org/sites/default/files/A6.D1.6.3CMM-LIKEMODELFOROSS.pdf"](http://www.qualipso.org/sites/default/files/A6.D1.6.3CMM-LIKEMODELFOROSS.pdf). [Accesado: Junio 21, 2011]
- [46] E. Petrinja, A. Sillitti, y G. Succi, "Overview on Trust in large FLOSS Communities," en IFIP International Federation for Information Processing, Septiembre 2008, Milan, Italy, pp. 47-56.
- [47] G. Gousios et al., Software Quality Assessment of Open Source, Athens University of Economics and Business. [En línea]. Disponible en: [HYPERLINK "http://istlab.dmst.aueb.gr/~george/pubs/2007-PCI-GKSLVS/paper.pdf"](http://istlab.dmst.aueb.gr/~george/pubs/2007-PCI-GKSLVS/paper.pdf). [Accesado: Julio 15, 2011 ]
- [48] R. V Solingen, "The goal/question/metric," en Encyclopedia of Software Engineering 2., 2002, pp. 578–583.
- [49] V. R. Basili, "Software Modeling and Measurement: The Goal/Question/Metric Paradigm," Computer Science Technical Report Series NR: CSTR2956/ NR: UMIACSTR9296, 1992.
- [50] D. Spinellis et al., "Evaluating the quality of open source software," en SQM 2008: Second International Workshop on Software Quality and Maintainability—12th European Conference on Software Maintenance and Reengineering (CSMR 2008) satellite event, Marzo 2008, Atenas, Grecia, pp. 5 - 28.
- [51] E. Petrinja, A. Sillitti, y J. Succi, "Comparing OpenBRR, QSOS, and OMM Assessment Models," en 6th International Conference on Open Source Systems (OSS2010), 2010, Notre Dame, Indiana, pp. 224-238.
- [52] J. Deprez y S. Alexandre, Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS, Procutor: Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC). [En línea]. Disponible en: [HYPERLINK "http://www.qualoss.org/dissemination/DEPREZ\\_CompareFIOSSAssessMethodo-Camera-02.pdf"](http://www.qualoss.org/dissemination/DEPREZ_CompareFIOSSAssessMethodo-Camera-02.pdf). [Accesado: Mayo 16, 2011]
- [53] (SEI), Carnegie Mellon Software Engineering Institute, "Capability Maturity Model Integration (CMMI): Overview", Julio 2011. [En línea]. Disponible en: [HYPERLINK "http://www.sei.cmu.edu/cmml/"](http://www.sei.cmu.edu/cmml/) <http://www.sei.cmu.edu/cmml/>. [Accesado: Mayo 18, 2011]
- [54] ¿Qué es licencia GNU o GLP? [En línea]. Disponible en: [HYPERLINK "http://www.dansocial.gov.co/index.php?option=com\\_remository&Itemid=444&func=fileinfo&id=119"](http://www.dansocial.gov.co/index.php?option=com_remository&Itemid=444&func=fileinfo&id=119). [Accesado: Mayo 25, 2011]
- [55] Open Source Initiative, Open Source Initiative OSI - The Academic Free License 3.0: Licensing, Open Source Initiative. [En línea]. Disponible en: [HYPERLINK "http://www.opensource.org/licenses/afl-3.0.php"](http://www.opensource.org/licenses/afl-3.0.php). [Accesado: Mayo 15, 2011]
- [56] B. Noguera, Qué es una licencia GNU. [En línea]. Disponible en: [HYPERLINK "http://culturacion.com/2010/12/que-es-una-licencia-gnu/"](http://culturacion.com/2010/12/que-es-una-licencia-gnu/). [Accesado: 25 Mayo, 2011]