



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

**REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA
IDENTIFICACIÓN DE FACTORES DE ÉXITO, FRACASO
Y RIESGOS EN LA ADOPCIÓN DE SOFTWARE LIBRE**

Trabajo de graduación previo a la obtención del título de
Ingeniero en Sistemas y Telemática

Autores:

Oscar David Armijos Mera
Luis Fernando Delgado Guerrero

Director:

Ing. Carvallo Vega Juan Pablo, PhD.

Cuenca – Ecuador

2018

DEDICATORIA

Dedico este trabajo principalmente a Dios, por ser mi sustento día a día, a mis padres por su ayuda incondicional en mi vida, a mi familia por apoyarme siempre y a todas las personas que de una u otra manera colaboraron a la realización de este proyecto.

Oscar David Armijos Mera

El camino que he recorrido ha sido largo y haber llegado a culminar esta tesis es algo que me llena de alegría; objetivo cumplido que deseo dedicar de manera muy especial en primera instancia a mis padres por ser personas trabajadoras y llenas de valores que me supieron formar e inculcar, así mismo no puedo dejar de lado todo el apoyo recibido de mis hermanos y demás familiares, quienes contribuyeron de una u otra forma al alcance de mi sueño con amor paciencia y esperanza.

Luis Fernando Delgado

AGRADECIMIENTO

Agradezco principalmente a Dios, por ser mi sustento día a día, a mi padre Darwin Armijos Gonzaga y madre Elisa Mera Brito por su ayuda incondicional y ser un motor de Fe en mi vida, a mi hija Arely Armijos por ser luz en mi vida, a mi hermana Isabel Armijos, y mis demás familiares por su ayuda incondicional. También agradezco a María por su apoyo en este proceso y al asesoramiento, experiencia y conocimiento compartido por parte de nuestro tutor Ing. Juan Pablo Carvallo Vega, PhD.

Del mismo modo agradezco a mi amigo y compañero de tesis Fernando Delgado y a todas las personas que de una u otra manera colaboraron a la realización este proyecto.

Oscar David Armijos Mera

Agradecer a Dios por colocar en mi vida a las personas y situaciones necesarias para mi desarrollo como estudiante con miras al profesionalismo y antes de nada como persona mi guía y luz en todo momento, por supuesto cumplir este objetivo no hubiera sido posible sin la guía, asesoramiento, experiencia y conocimiento por parte del Ing. Juan Pablo Carvallo Vega, PhD. A quien agradezco con total sinceridad por todo el apoyo brindado. Del mismo modo agradecer a la Msc. Rosalva Vintimilla, Ing. y los demás integrantes del proyecto “Adoptados”, a mis padres, hermanos, entre ellos a mi hermana Eunora y su familia por su ayuda incondicional, a mi amigo y compañero de tesis Oscar Armijos Mera.

Luis Fernando Delgado

RESUMEN

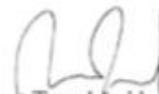
El presente trabajo tiene como finalidad realizar una revisión sistemática de literatura de la adopción de FOSS, con lo que se pretende identificar, evaluar e interpretar las investigaciones existentes relacionadas a los factores de éxito, fracaso, riesgos, barreras, dominios exitosos, documentación generada y opinión del usuario, en la adopción de software libre por las organizaciones a nivel mundial.

Mediante la guía de Kitchenham & Charters (2007), se elaboró un protocolo de revisión, se identificó y seleccionó los estudios primarios de las fuentes requeridas, se evaluó la calidad y se extrajo los datos de los estudios seleccionados en el formulario definido en el protocolo, para luego tabular y analizar los datos recopilados, finalizando con la discusión de los resultados obtenidos.

Palabras claves: software de código abierto, adopción, riesgos, software libre, organizaciones, beneficios, barreras.

ABSTRACT

The purpose of the present work was to perform a systematic literature review of the adoption of FOSS. The aim was to identify, evaluate and interpret existing research related to success factors, failure, risks, barriers, successful domains, generated documentation and user's opinion on the adoption of free software by organizations worldwide. Using the Kitchenham & Charters guide (2007), a review protocol was developed, primary studies were identified and selected from the required sources. The quality was evaluated and the information of the selected studies was extracted in the form defined in the protocol. Then, the collected data was tabulated and analyzed, ending with the discussion of the results obtained.



Translated by

Ing. Paul Arpi

ÍNDICE

DEDICATORIA	II
AGRADECIMIENTO	III
RESUMEN	IV
ABSTRACT	V
ÍNDICE	VI
Índice de Figuras	IX
Índice de Tablas	XII
Capítulo 1	1
1.1 Introducción	1
1.2 Antecedentes	2
1.3 Justificación	2
1.4 Objetivos	6
1.5 Metodología	6
1.6 Estructura general del documento	6
Capítulo 2. Metodología para una revisión sistemática de literatura	8
2.1. Introducción	8
2.2. En función del nivel de evidencia	9
2.2.1. Métodos primarios	9
2.2.2. Métodos secundarios	10
2.3. En función al propósito	12
2.4. Revisiones sistemáticas de la literatura	12
2.4.1. Proceso de la revisión sistemática de literatura	13
2.4.1.1. Planificar la revisión	14
2.4.1.1.1. Identificación de la necesidad de una revisión	14
2.4.1.1.2. Puesta en marcha de una revisión	15
2.4.1.1.3. Especificar las preguntas de investigación	15
2.4.1.1.4. Desarrollar el protocolo de revisión	16
2.4.1.1.5. Evaluar el protocolo de revisión	20
2.4.1.2. Realizar la revisión	20
2.4.1.2.1. Identificación de la investigación	20
2.4.1.2.2. Selección de estudios primarios	21
2.4.1.2.3. Evaluación de la calidad del estudio primario	21
2.4.1.2.4. Extracción de datos y monitoreo	21

2.4.1.2.5. Síntesis de datos.....	22
2.4.1.3. Reportar la revisión.....	22
2.5. Bibliotecas digitales.....	22
2.5.1. IEEE Xplore.....	23
2.5.2. SpringerLink.....	24
2.5.3. ACM Digital Library.....	25
Capítulo 3. Planeación del SLR.....	26
3.1. Introducción.....	26
3.2. Campo a estudiar.....	26
3.3. Identificación de la necesidad de una revisión.....	26
3.4. Preguntas de investigación.....	27
3.5. Protocolo.....	28
3.5.1. Estrategia de búsqueda de estudios primarios.....	28
3.5.1.1. Determinación de la cadena de búsqueda genérica.....	28
3.5.1.2. Seleccionar los recursos de búsqueda en los que se debe buscar (bibliotecas digitales, revistas específicas, y resúmenes de congresos).....	29
3.6. Criterios de selección de estudios.....	30
3.6.1. Criterios de inclusión:.....	30
3.6.2. Criterios de exclusión:.....	30
3.7. Procedimientos de selección de estudios.....	31
3.8. Listas de verificación y procedimientos de evaluación de calidad del estudio.....	31
3.8.1. Criterios de evaluación.....	31
3.8.2. Formulario para la evaluación de calidad.....	33
3.9. Estrategia de extracción de datos.....	33
3.10. Síntesis de los datos extraídos.....	40
CAPÍTULO 4. Desarrollo del SLR.....	41
4.1. Introducción.....	41
4.2. Identificación y selección de estudios.....	41
4.2.1. Identificación de estudios.....	41
4.2.1.1. Biblioteca Digital IEEE Xplore.....	42
4.2.1.1.1. Prueba piloto.....	43
4.2.1.1.2. Refinamiento 1.....	44
4.2.1.1.3. Refinamiento 2.....	44
4.2.1.2. Biblioteca ACM DIGITAL LIBRARY.....	45
4.2.1.2.1. Prueba piloto.....	46
4.2.1.2.2. Refinamiento 1.....	47

4.2.1.2.3.	Refinamiento 2.....	47
4.2.1.3.	Biblioteca Springer Link.....	48
4.2.1.3.1.	Prueba piloto.....	50
4.2.1.3.2.	Refinamiento 1.....	51
4.2.1.3.3.	Refinamiento 2.....	51
4.2.2.	Selección de estudios.....	52
4.2.2.1.	Preselección de estudios primarios.....	52
4.2.2.2.	Selección final de estudios primarios.....	55
4.2.2.3.	Identificación de estudios adicionales mediante la técnica bola de nieve (Snowballing).....	56
4.2.2.3.1.	Inicio:.....	56
4.2.2.3.2.	Iteraciones:.....	56
4.2.2.3.3.	Inclusión y exclusión.....	57
4.3.	Evaluación de la calidad de los estudios primarios.....	57
4.4.	Extracción de datos.....	58
Capítulo 5. Análisis de datos.....		60
5.1.	Introducción.....	60
5.2.	Interpretación de datos extraídos.....	60
5.2.1.	¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?.....	63
5.2.1.1.	Factores de éxito.....	63
5.2.1.2.	Factores de fracaso.....	87
5.2.2.	¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre a nivel mundial?.....	98
5.2.3.	¿Cuáles son los beneficios de la utilización de software libre?.....	119
5.2.4.	¿En qué dominios de software ha resultado exitosa la adopción de software libre a nivel mundial?.....	145
5.2.5.	¿Cuáles son las barreras que han identificado las organizaciones al adoptar software libre?.....	156
5.2.6.	¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?.....	177
5.2.7.	¿Se ha incluido la opinión del usuario del uso del software libre?.....	177
Capítulo 6. Discusión de resultados obtenidos.....		178
6.1.	Introducción.....	178
6.2.	Principales hallazgos.....	178
6.3.	Amenazas a la validez.....	182
6.3.1.	Validez de constructo.....	183

6.3.2.	Validez interna.....	183
6.3.3.	Validez externa.....	183
6.3.4.	Validez de conclusión.....	184
Capítulo 7. Conclusiones y recomendaciones		185
7.1.	Conclusiones	185
7.2.	Recomendaciones	187
Referencias		189

Índice de Figuras

Figura 2.1	Proceso para realizar una SLR (Kitchenham, 2007).....	14
Figura 2.2	Proceso de Snowballing (Wohlin, 2014)	19
Figura 4.1	Búsqueda avanzada en IEEE Xplore (Base de datos IEEE Xplore, 2017) ...	42
Figura 4.2	Pasos para la identificación de publicaciones en biblioteca IEEE Xplore (Autoría propia, 2017).....	43
Figura 4.3	Búsqueda avanzada en ACM Digital Library (ACM Digital Library, 2017)	45
Figura 4.4	Pasos para identificación de publicaciones en biblioteca ACM Digital Library (Autoría propia, 2017)	46
Figura 4.5	Búsqueda avanzada en Springer link (Springer Link, 2017)	48
Figura 4.6	Pasos para identificación de publicaciones en biblioteca Springer Link (Autoría propia, 2017).....	49
Figura 4.7	Asistente de búsquedas avanzadas en Springer Link (Springer Link, 2017)	50
Figura 4.8	Preselección de artículos obtenidos de IEEE Xplore en primera instancia (Autoría propia, 2017).....	53
Figura 4.9	Preselección de artículos obtenidos de ACM Digital Library en primera instancia (Autoría propia, 2017).....	53
Figura 4.10	Preselección de artículos obtenidos de Springer Link en primera instancia (Autoría propia, 2017).....	54
Figura 4.11	Preselección de artículos obtenidos de IEEE Xplore en segunda instancia (Autoría propia, 2017).....	54
Figura 4.12	Preselección de artículos obtenidos de ACM Digital Library en segunda instancia (Autoría propia, 2017).....	55
Figura 4.13	Preselección de artículos obtenidos de Springer Link en segunda instancia (Autoría propia, 2017).....	55
Figura 4.14	Evaluación de calidad de los estudios primarios (Autoría propia, 2017) ...	57
Figura 4.15	Resultados obtenidos de acuerdo con cada fase (Autoría propia, 2017).....	58
Figura 4.16	Extracción de datos (Autoría propia, 2017)	59
Figura 5.1	Artículos relevantes al estudio (Autoría propia, 2017)	60
Figura 5.2	Distribución por tipo de publicación (Autoría propia, 2017).....	61
Figura 5.3	Tendencias de publicaciones por año (Autoría propia, 2017).....	62
Figura 5.4	Categorías y subcategorías a los que pertenecen los factores de éxito (Autoría propia, 2017).....	64
Figura 5.5	Número de subcategorías por categoría (Autoría propia, 2017).....	65

Figura 5.6 Cantidad de factores encontrados por subcategoría (Autoría propia, 2017).	66
Figura 5.7 Porcentaje de factores de éxito que contiene cada categoría (Autoría propia, 2017).....	67
Figura 5.8 Representación gráfica de frecuencia de los factores de éxito (Autoría propia, 2017).....	72
Figura 5.9 Artículos que hacen mención a la categoría “Capacitación” y subcategoría “Capacitación formación” (Autoría propia, 2017)	74
Figura 5.10 Artículos que hacen mención a la categoría “Valores percibidos” y subcategoría “Ingresos” (Autoría propia, 2017).....	75
Figura 5.11 Artículos que hacen mención a la categoría “Gestión/manejo” y subcategoría “Administrar la correcta evolución del software” (Autoría propia, 2017)	77
Figura 5.12 Artículos que hacen mención a la categoría “Gestión/manejo” y subcategoría “Planificación” (Autoría propia, 2017).	79
Figura 5.13 Artículos que hacen mención a la categoría “Actividad asociada con la comunidad” y subcategoría “Actividad que produce la comunidad en apoyo al OSS” (Autoría propia, 2017).	80
Figura 5.14 Artículos que hacen mención a la categoría “Políticas” y subcategoría “Políticas corporativas de TIC y de adopción del OSS” (Autoría propia, 2017).	83
Figura 5.15 Categorías y subcategorías a los que pertenecen los factores de fracaso (Autoría propia, 2017).	88
Figura 5.16 Factores de fracaso: número de subcategorías por categoría (Autoría propia, 2017).....	89
Figura 5.17 Factores de fracaso: número de subcategorías por categoría (Autoría propia, 2017).....	90
Figura 5.18 Porcentaje de factores de fracaso que contiene cada categoría (Autoría propia, 2017).....	91
Figura 5.19 Representación gráfica de menciones de los factores de fracaso (Autoría propia, 2017).....	93
Figura 5.20 Artículos que hacen mención a la categoría “Competitividad” y en la subcategoría “Baja competitividad” (Autoría propia, 2017).	95
Figura 5.21 Artículos que hacen mención a categoría “Factor Humano” y en la subcategoría “Desarrollador” (Autoría propia, 2017).	96
Figura 5.22 Categorías y subcategorías a los que pertenecen los riesgos identificados en la adopción de OSS (Autoría propia, 2017).	99
Figura 5.23 Número de subcategorías por categoría (Autoría propia, 2017).	100
Figura 5.24 Cantidad de riesgos encontrados por subcategoría (Autoría propia, 2017).	101
Figura 5.25 Porcentaje de riesgos que contiene cada categoría (Autoría propia, 2017).	102
Figura 5.26 Representación gráfica de menciones de los riesgos de adopción de OSS (Autoría propia, 2017).	109
Figura 5.27 Artículos que hacen mención a la categoría “Costes” y subcategoría “Problema de costes” (Autoría propia, 2017).....	111
Figura 5.28 Artículos que hacen mención a la categoría “Riesgos Legales” y subcategoría “Licencias” (Autoría propia, 2017).	114
Figura 5.29 Artículos que hacen mención a categoría “Riesgos Legales” subcategoría “Propiedad intelectual” (Autoría propia, 2017).....	115

Figura 5.30 Artículos que hacen mención a categoría “Seguridad” y subcategoría “Riesgo de seguridad” (Autoría propia, 2017).	116
Figura 5.31 Artículos que hacen mención a categoría “Software” y subcategoría “Complejidad” (Autoría propia, 2017).	117
Figura 5.32 Categorías y subcategorías de los beneficios de adoptar software libre (Autoría propia, 2017).	120
Figura 5.33 Número de subcategorías por categoría (Autoría propia, 2017).	121
Figura 5.34 Cantidad de beneficios encontrados por subcategoría (Autoría propia, 2017).	122
Figura 5.35 Porcentaje de beneficios que contiene cada categoría (Autoría propia, 2017).	123
Figura 5.36 Representación gráfica de menciones de los beneficios de adopción de OSS (Autoría propia, 2017).	128
Figura 5.37 Artículos que hacen mención a la categoría “Conocimiento” y subcategoría “Aprendizaje” (Autoría propia, 2017).	130
Figura 5.38 Artículos que hacen mención a la categoría “Rentabilidad” y subcategoría “Inversión” (Autoría propia, 2017).	132
Figura 5.39 Artículos que hacen mención a la categoría “Desarrollo” y subcategoría “Desarrollo económico de un país” (Autoría propia, 2017).	133
Figura 5.40 Artículos que hacen mención a la categoría “Desarrollo” y subcategoría “Genera oportunidad de empleo” (Autoría propia, 2017).	134
Figura 5.41 Artículos que hacen mención a la categoría “Organización” y subcategoría “Empresa” (Autoría propia, 2017).	136
Figura 5.42 Artículos que hacen mención a la categoría “Innovación” y subcategoría “Impulso a la innovación” (Autoría propia, 2017).	138
Figura 5.43 Artículos que hacen mención a la categoría “Flexibilidad” y subcategoría “Código fuente” (Autoría propia, 2017).	139
Figura 5.44 Artículos que hacen mención a la categoría “Hardware” y subcategoría “Comunidad” (Autoría propia, 2017).	141
Figura 5.45 Artículos que hacen mención a la categoría “Software” y subcategoría “Control” (Autoría propia, 2017).	142
Figura 5.46 Cantidad de artículos que hacen mención a los beneficios en la categoría “Software” y subcategoría “Software desarrollado” (Autoría propia, 2017).	143
Figura 5.47 Cantidad de artículos que hacen mención a los beneficios en la categoría “Tiempo” y subcategoría “Ahorro de tiempo” (Autoría propia, 2017).	145
Figura 5.48 Categorías de dominios exitosos de adopción de OSS (Autoría propia, 2017).	146
Figura 5.49 Categorías y subcategorías a los que pertenecen dominios exitosos en la adopción de OSS (Autoría propia, 2017).	147
Figura 5.50 Número de subcategorías por categoría (Autoría propia, 2017).	148
Figura 5.51 Cantidad de dominios de éxito encontrados por subcategoría (Autoría propia, 2017).	149
Figura 5.52 Representación gráfica de menciones de los dominios exitosos de adopción de OSS (Autoría propia, 2017).	152
Figura 5.53 Artículos que hacen mención a la categoría “Software de sistema” y subcategoría “Servidores” (Autoría propia, 2017).	154

Figura 5.54 Categorías y subcategorías a los que pertenecen las barreras identificadas en la adopción de OSS (Autoría propia, 2017).	157
Figura 5.55 Número de subcategorías por categoría (Autoría propia, 2017).	158
Figura 5.56 Cantidad de Barreras encontradas por subcategoría (Autoría propia, 2017).	159
Figura 5.57 Porcentaje de barreras que contiene cada categoría (Autoría propia, 2017).	160
Figura 5.58 Representación gráfica de menciones de las barreras que impide adoptar software libre (Autoría propia, 2017).	165
Figura 5.59 Artículos que hacen mención a categoría “Costes” y subcategoría “Problemas económicos” (Autoría propia, 2017).	167
Figura 5.60 Artículos que hacen mención a la categoría “Factores Humanos” y subcategoría “Experiencia” (Autoría propia, 2017).	169
Figura 5.61 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Incertidumbre” (Autoría propia, 2017).	170
Figura 5.62 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Motivación” (Autoría propia, 2017).	171
Figura 5.63 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Tiempo del personal” (Autoría propia, 2017). .	172
Figura 5.64 Cantidad de artículos que hacen mención a las barreras en la categoría “Políticas” y subcategoría “Falta de políticas para implementar OSS” (Autoría propia, 2017).	174
Figura 5.65 Cantidad de artículos que hacen mención a las barreras en la categoría “Seguridad” y subcategoría “Preocupaciones” (Autoría propia, 2017).	175

Índice de Tablas.

Tabla 1.1 Trabajos relacionados (Autoría propia, 2018).....	5
Tabla 2.1 Diferencias entre Mapping y SLRs (Genero <i>et al.</i> , 2014) (Runeson <i>et al.</i> , 2012).....	12
Tabla 2.2 Representación general de las características de las estrategias empíricas de investigación (Genero <i>et al.</i> , 2014) (Runeson <i>et al.</i> , 2012).....	12
Tabla 2.3 Operadores utilizados en consulta (WIPO, 2008)	23
Tabla 3.1 Preguntas de investigación (Autoría propia, 2018)	28
Tabla 3.2 Términos relevantes y alternativos (Autoría propia, 2018).....	29
Tabla 3.3 Bibliotecas seleccionadas (Autoría propia, 2017)	30
Tabla 3.4 Formulario de evaluación de calidad (Autoría propia, 2018).....	33
Tabla 3.5 Factores de adopción de FOSS (Autoría propia, 2018).....	34
Tabla 3.6 Formulario de extracción de riesgos de adopción de FOSS (Autoría propia, 2017).....	35
Tabla 3.7 Formulario de extracción de beneficios de adopción de FOSS (Autoría propia, 2017).....	36
Tabla 3.8 Formulario de extracción de barreras para la adopción de FOSS (Autoría propia, 2017).....	37

Tabla 3.9 Formulario de extracción de dominios exitosos en la adopción de FOSS (Autoría propia, 2017)	39
Tabla 3.10 Formulario de extracción de la pregunta ¿La documentación generada en la adopción de OSS (Autoría propia, 2017)	39
Tabla 3.11 Formulario de extracción de la pregunta ¿Se ha incluido la opinión del usuario sobre el uso del software libre (Autoría propia, 2017)	39
Tabla 4.1 Primera cadena de búsqueda IEEE (Autoría propia, 2017).....	44
Tabla 4.2 Refinamiento 1 cadena de búsqueda IEEE (Autoría propia, 2017).....	44
Tabla 4.3 Refinamiento 2, cadena de búsqueda IEEE (Autoría propia, 2017).....	45
Tabla 4.4 Primera cadena de búsqueda ACM Digital Library (Autoría propia, 2017) ..	47
Tabla 4.5 Refinamiento 1 cadena búsqueda ACM Digital Library (Autoría propia, 2017)	47
Tabla 4.6 Refinamiento 2 cadena búsqueda ACM Digital Library (Autoría propia, 2017)	48
Tabla 4.7 Primera cadena de búsqueda Springer Link (Autoría propia, 2017).....	50
Tabla 4.8 Refinamiento 1 cadena búsqueda Springer Link (Autoría propia, 2017).....	51
Tabla 4.9 Refinamiento 2 cadena búsqueda Springer Link (Autoría propia, 2017).....	52
Tabla 5.1 Factores de éxito (Autoría propia, 2017).....	71
Tabla 5.2 Factores de fracaso identificados (Autoría propia, 2017).	93
Tabla 5.3 Riesgos de adopción de OSS (Autoría propia, 2017).....	109
Tabla 5.4 Beneficios de adoptar OSS (Autoría propia, 2017).....	127
Tabla 5.5 Dominios de éxito de adopción de OSS (Autoría propia, 2017).....	151
Tabla 5.6 Barreras que impiden adoptar software libre (Autoría propia, 2017).	164

Capítulo 1

1.1 Introducción

Según “Free Software Foundation (FSF)”, el Software Libre es todo programa que otorga su libre utilización y que permite estudiarlo y compartir los conocimientos adquiridos con la comunidad en general.

El uso del software libre data desde los años de 1984; fue creado por Richard M. Stallman con el proyecto denominado GNU para generar el reemplazo al sistema operativo UNIX. (Free Software Foundation, 2013); mientras que en febrero de 1998 se acuñó el término Open Source Software (OSS) , debido a la ambigüedad del término Free Software, ya que la filosofía del OSS, es resaltar los beneficios de compartir el código fuente, y con el fin de iniciar una campaña de software libre para llevarlo al ámbito empresarial y posteriormente a los gobiernos. (Feller, Fitzgerald, Hissam, & Lakhani, 2007) (Samoladas, Stamelos, Angelis, & Oikonomou, 2004) (Rodríguez, 2008) (Petrijevcanin Vuksanovic & Sudarevic, 2012).

Cuando se habla de Software Libre, es recomendable no utilizar expresiones tales como regalar o gratis, debido a que caemos en la errada interpretación de tomarlo como una cuestión de precio y no de libertad (Stallman, 2004).

Según Torres y Petrizzo (2015), Free Software Foundation (FSF) determina las cuatro características para que un software sea libre:

1. Utilización libre del software con cualquier fin.
2. Acceder libremente al código del software para analizarlo y adaptarlo a las necesidades.
3. Libertad de redistribuir copias de software.
4. Mejorar el software y compartirlo al público en general.

Hay una gran variedad de aplicaciones de Software libre que se ofrecen en el mercado. Entre las más famosas que se pueden citar, según Rodríguez (2008) están:

- El sistema operativo Linux.
- El servidor de web Apache.
- El manejador de bases de datos objeto-relacional PostgreSQL.
- El navegador Mozilla Firefox.

- La suite de aplicaciones de escritorio OpenOffice.
- El servidor de correo Sendmail.

1.2 Antecedentes

Con la predisposición de lograr soberanía y autonomía tecnológica, además de minimizar los costos en recursos públicos al Gobierno del Ecuador en el Decreto Ejecutivo No. 1014, firmado el 10 de abril de 2008, en su artículo 1, constituye como política pública la implementación de Software Libre en las Entidades de Administración Pública Central en sus sistemas y equipamientos informáticos (Correa, 2008).

La implementación de Software Libre (FOSS) es un aspecto relevante en la realidad ecuatoriana, y al no tener identificado, ni analizado a detalle los riesgos y las buenas prácticas se limita a ser explotado de manera óptima, en cuanto a los beneficios que tiene, para obtener una adopción adecuada de FOSS, en empresas públicas y privadas en el país.

Para atender esta necesidad, se plantea el proyecto “Adoptados”, que mantiene la Universidad del Azuay, sobre la adopción de Software Libre, dentro del cual se pretende realizar una revisión sistemática de literatura (SLR), sobre las experiencias en la implementación de FOSS en las diferentes organizaciones a nivel internacional, con el objeto de determinar el estado del arte en esta materia y contrastarlo con las prácticas que se identifiquen en la adopción de Software Libre en el país.

El SLR a realizarse pretende identificar los siguientes aspectos: factores de éxito y fracaso, riesgos, beneficios y barreras que conlleva la adopción de FOSS en las organizaciones; dominios en los que su implementación a resultado exitosa; valorar si la documentación generada para los usuarios finales es suficiente; y determinar si se toma en cuenta la opinión del usuario al adoptar FOSS.

1.3 Justificación

Se han identificado 11 trabajos de Revisiones Sistemáticas de Literatura en el campo del Software Libre, con la utilización de Google Académico, aplicando la cadena “SRL Open Source Software” y “Systematic Literature Review Open Source Software”, de los resultados se seleccionó los que más se relacionaban al estudio planteado; pero sus objetivos, preguntas de investigación y alcance son distintos a la investigación planteada. El objetivo de esta revisión es identificar el estado del arte sobre los factores de éxito,

fracaso, barreras, riesgos, beneficios, documentación generada por la comunidad y opinión del usuario en el uso de FOSS a nivel organizacional. La Tabla nro. 1.1, describe los trabajos relacionados, juntamente con su objetivo y las preguntas de investigación.

Partiendo de esta indagación previa, se puede corroborar que, si bien existen estudios relacionados al tema de investigación, ninguno de estos se relaciona en su totalidad con las preguntas de investigación planteadas en este estudio.

Utilizando la técnica de SLR se pretende comprender las ventajas y desventajas de la implementación de FOSS en las organizaciones, así como también los factores que deben ser tomados en cuenta para alcanzar el éxito, evitar el fracaso y disminuir los riesgos al adoptar FOSS.

Es importante la realización de este estudio, para conocer las prácticas de adopción de FOSS a nivel internacional, las cuales servirán como base de comparación con lo que se está realizando en el Ecuador.

La Tabla nro. 1.1, describe los trabajos relacionados, el objetivo y las preguntas de investigación de los trabajos:

	Referencia	Objetivo	Preguntas de investigación
1	(Alasbali & Benatallah, 2015)	Identificar las ventajas y desafíos en el uso del OSS en la educación de las ciencias de la computación	<p>P1. ¿Cuáles son las principales ventajas de utilizar Código abierto en la educación de informática en comparación con los enfoques educativos tradicionales?</p> <p>P2. ¿Cuáles son las principales dificultades y los principales desafíos del uso del código abierto en la enseñanza de las ciencias de la computación?</p>
2	(Hauge, Ayala, & Conradi, 2010),	Analizar el mecanismo de adopción de Software OSS en las organizaciones. Indagar sobre el enfoque, las características y las limitaciones de la investigación empírica sobre la adopción de OSS en las organizaciones.	<p>P1. ¿De qué manera las organizaciones de software intensivo adoptan OSS?</p> <p>P2. ¿Cuál ha sido el enfoque de la investigación empírica sobre la adopción de OSS en las organizaciones?</p> <p>P3. ¿Cuáles son las características y limitaciones de la investigación empírica actual sobre la adopción institucional de OSS?</p>
3	(Steinmacher, Graciotto Silva, Gerosa, & Redmiles, 2015)	Identificar las barreras que enfrentan los que recién llegan a contribuir con proyectos de software de código abierto	<p>P1. ¿Cuáles son las barreras que dificultan la contribución de los nuevos socios en los proyectos OSS?</p>

4	(Breivold, Chauhan, & Babar, 2010)	Obtener una visión general de los estudios existentes en la evolución del software de fuente abierta	<p>P1. ¿Cuáles son los principales temas de investigación que se tratan en la literatura científica sobre la evolución del software de fuente abierta, y el análisis y el logro de los atributos de calidad relacionados con la evolución?</p> <p>P2. ¿Cuáles son las métricas que se utilizan para la medición y el análisis de la evolución de OSS, y cuáles son las limitaciones en el uso de estas métricas, si las hay?</p>
5	(Sulaman, y otros, 2014),	El objetivo de este estudio es revisar la investigación llevada a cabo sobre el uso del código fuente abierto en el desarrollo de software y sistemas críticos para la seguridad.	<p>P1. ¿En qué dominios de sistemas críticos para la seguridad se han utilizado los sistemas OSS?</p> <p>P2. ¿Qué tipos de funcionalidad OSS se han incluido en los sistemas críticos para la seguridad?</p> <p>P3. ¿Cuáles son las características de las comunidades que desarrollan el OSS identificado en RQ2?</p>
6	(Wu, Shi, Chen, Wang, & Boehm, 2017);	El objetivo de este estudio es identificar el estado actual de los enfoques de estimación del esfuerzo de mantenimiento existentes para OSS.	<p>P1. ¿Qué evidencia hay para las técnicas / métodos de estimación del esfuerzo de mantenimiento para los proyectos OSS?</p> <p>P2. ¿Qué indicadores relacionados con los registros de desarrollo de OSS se extraen para la estimación del esfuerzo de mantenimiento y cómo se pueden clasificar?</p> <p>P3. ¿Cuáles son los proyectos comunes y el tamaño del conjunto de datos utilizados como casos de estudio en la estimación del esfuerzo de mantenimiento OSS, y cómo ha sido la frecuencia de los enfoques relacionados con el tamaño del conjunto de datos?</p> <p>P4. ¿Qué métodos / enfoques se utilizan para estimar el esfuerzo real de mantenimiento del proyecto?</p> <p>P5. ¿Cuál es la precisión de la estimación general de la estimación del esfuerzo de mantenimiento del OSS?</p>
7	(Steinmacher, Chaves, Conte, & Gorosa, 2014);	Informar los resultados de un estudio cualitativo que se basa en los datos obtenidos de una revisión sistemática de la literatura y de los profesionales.	<p>P1. ¿Cuáles son las barreras que influyen en la incorporación de los recién llegados a los proyectos de OSS?</p>
8	(Lee, Wei, & Ghani, 2016)	El objetivo de este documento es realizar una revisión sistemática de	<p>P1. ¿Qué enfoques se han empleado para estimar el esfuerzo inicial en proyectos de desarrollo de aplicaciones web?</p>

		la literatura (SLR) del desarrollo de aplicaciones web OSS.	<p>P2. ¿Cuáles son los enfoques más comunes utilizados en la estimación del esfuerzo inicial en proyectos de desarrollo de aplicaciones web?</p> <p>P3. ¿Cuál es el conjunto de datos disponible utilizado para entrenar el modelo de estimación de esfuerzo?</p> <p>P4. ¿Cuáles son las herramientas de desarrollo de fuentes más utilizadas en el desarrollo de aplicaciones web?</p> <p>P5. ¿Existe un modelo de estimación de esfuerzo temprano para estimar el esfuerzo en el campo relacionado?</p> <p>P6. ¿Es necesario desarrollar un modelo de estimación de esfuerzo inicial para los proyectos de desarrollo de aplicaciones web OSS?</p>
9	(Årdal, Alstadsæter, & Røttingen, 2011);	El objetivo es dar una reproducción imparcial de la evidencia actual que aborda la pregunta de investigación, ¿cuáles son las características y barreras comunes del desarrollo de software de código abierto?	<p>P1. ¿Cuáles son las características comunes para iniciar y mantener un proyecto de desarrollo de software de código abierto?</p> <p>P2. ¿Cuáles son las barreras para desarrollar un proyecto de desarrollo de software de código abierto?</p>
10	(Syeed, Kilamo, & Imed Hammouda, 2012);	Realizar una revisión sistemática de la literatura sobre el estado del arte en la predicción de proyectos de OSS considerando tanto el código como la dimensión de la comunidad	<p>P1. ¿Cuáles son los principales enfoques / propósitos del estudio?</p> <p>P2. ¿Cuáles son los conjuntos de datos de los proyectos OSS explotados en la predicción?</p> <p>P3. ¿Qué tipos de métodos se utilizan para predecir proyectos de OSS?</p> <p>P4. ¿Qué tipos de métricas se usan para predecir proyectos de OSS?</p>
11	(Adewumi, Misra, Omoregbe, Crawford, & Soto, 2016).	Realizar una revisión sistemática de la literatura para investigar las propiedades de los modelos existentes de evaluación de la calidad del OSS clasificándolos con respecto a sus características de calidad, la metodología que utilizan para la evaluación y su dominio de aplicación.	<p>P1. ¿Cuáles son las características de calidad clave que poseen los modelos de evaluación de calidad de OSS?</p> <p>P2. ¿Cuáles son los métodos utilizados por estos modelos de evaluación de calidad de OSS para las decisiones de selección?</p> <p>P3. ¿Cuál es el dominio de aplicación de los modelos de evaluación de calidad de OSS?</p>

Tabla 1.1 Trabajos relacionados (Autoría propia, 2018)

1.4 Objetivos

Objetivo General

- Realizar una revisión sistemática de literatura para identificar el estado del arte sobre los factores de éxito, fracaso, riesgos, barreras, beneficios, dominios exitosos, documentación generada y opinión del usuario en el uso del FOSS.

Objetivos específicos

- Elaborar un protocolo de investigación.
- Seleccionar los artículos de las fuentes requeridas.
- Realizar la revisión de la literatura.
- Realizar la tabulación de análisis de datos de los artículos.
- Analizar los resultados obtenidos.

1.5 Metodología

Para la consecución de los objetivos del proyecto, se ha establecido una metodología basada en los siguientes principios:

- Profundizar en las bases teóricas del proyecto con la utilización de bibliografía.
- Identificar el punto de partida del desarrollo del Software Libre.
- Aplicación de la guía para la elaboración de la Revisión Sistemática de Literatura (SLR) orientada a la aplicación de FOSS (Kitchenham & Charters, 2007).

1.6 Estructura general del documento

La presente tesis se encuentra estructurada con los siguientes capítulos:

- **Introducción.** - Se presentan la introducción, justificación, objetivos, antecedentes y fundamentos teóricos de la revisión sistemática a desarrollar.

- **Metodología para una revisión sistemática de literatura.** – Se explica los métodos de investigación en la ingeniería de software y la metodología a seguir para la realización de una revisión sistemática de literatura.
- **Planeación del SLR.** – Se realizan actividades como: identificar la necesidad de la revisión, plantear las preguntas de investigación y desarrollar el protocolo a seguir en la revisión.
- **Desarrollo del SLR.** - En este capítulo se identifica, selecciona y evalúa la calidad de los estudios potencialmente relevantes para la investigación, además se extraen los datos relevantes para responder las preguntas de investigación anteriormente planteadas.
- **Análisis de datos.** – Partiendo de la extracción de datos realizada en el desarrollo del SLR, se procede a clasificar y analizar la información obtenida.
- **Discusión de resultados obtenidos.** - Se presentan los principales hallazgos de los resultados obtenidos de la revisión sistemática de literatura.
- **Conclusiones y Recomendaciones.** – se presentan conclusiones en base a la discusión realizada y recomendaciones de los autores en base a la experiencia en la realización del SLR.

Capítulo 2. Metodología para una revisión sistemática de literatura

2.1. Introducción

La expresión “Ingeniería de software” se ha ido fortaleciendo con el transcurso del tiempo, según Genero et al. (2014), mencionan que hace más de dos décadas la ingeniería de software se la consideraba como una disciplina nueva y poco desarrollada; sin embargo, a comienzos del siglo XXI, surgen las primeras publicaciones que describían las pautas a seguir en el área de la ingeniería de software, al momento de hacer experimentos o encuestas. Posteriormente a lo antes mencionando, en los últimos años investigadores han proporcionado una gran variedad de técnicas y guías para realizar investigaciones exactas y precisas. También las organizaciones y los profesionales se han visto en la necesidad de comparar de manera experimental las creencias y nuevas técnicas en el área de ingeniería de software, otorgando mayor atención a la ingeniería de software basada en evidencias (EBSE: Evidence-Based Software Engineering) dicho termino podría considerarse una evolución de la ingeniería del software empírica (ESE: Empirical Software Engineering).

En la investigación en ingeniería de software existen algunos tipos de métodos de investigación que se clasifican en métodos primarios y secundarios; los primeros hacen referencian a los métodos que se emplean con la finalidad de recuperar evidencia empírica, de un tema de interés en concreto entre los métodos primarios están: los experimentos, encuestas, estudios de caso, entre otros. Por otro lado, los métodos secundarios se encargan de obtener de una forma sistemática y precisa los estudios primarios que tiene correspondencia con una pregunta de investigación concreta, cuyo propósito es realizar una síntesis de las pruebas obtenidas, con lo cual se podrá dar respuesta a la pregunta en cuestión, entre los métodos secundarios están: las revisiones y mapeos sistemáticos de literatura.

En el capítulo que se desarrolla a continuación se identifican los diversos tipos de metodologías aplicadas a la ingeniería de software; analizando los métodos primarios, así como, los secundarios.

2.2. En función del nivel de evidencia

En función del nivel de evidencia los métodos de investigación según Genero et al. (2014), pueden ser clasificados como, métodos primarios y métodos secundarios. Las siguientes subsecciones describen cada uno de ellos.

2.2.1. Métodos primarios

Se emplean para conseguir pruebas de un tema de interés en concreto, para ello se recurre a: experimentos, casi-experimentos, estudios de caso, encuestas, entre otros. Esos tipos de técnicas metodológicas se utilizan para realizar los llamados estudios primarios.

- **Encuestas**

Permite recopilar información estandarizada de una muestra específica de una población a estudiar; las encuestas proporcionan datos cualitativos o cuantitativos importantes para describir, comparar y explicar conocimiento, actitudes, o comportamientos de las personas, a través de un cuestionario desarrollado o una entrevista elaborada previamente (Genero Bocco, Cruz-Lemus, & Mario, 2014) (Wohlin, y otros, 2012).

- **Experimentos**

Es una investigación empírica que manipula una variable independiente sobre una variable dependiente del entorno estudiado, tomando en cuenta a los sujetos asignados al azar (Wohlin, y otros, 2012).

- **Casi-experimentos**

Esta investigación es muy parecida a los experimentos; sin embargo, no es posible la aleatorización de los sujetos objeto de estudio; la elección surge a partir de las habilidades y conocimientos de los sujetos seleccionados (Wohlin, y otros, 2012).

- **Estudios de caso**

Según Nohlen et al. (2012) y Genero et al. (2014), son métodos que utilizan varias fuentes de evidencia, para investigar un fenómeno o caso en su contexto real, especialmente cuando los límites entre el fenómeno y su contexto no están claramente definidos, y a partir de ello generar conclusiones tanto cualitativas y cuantitativas. Este método se utiliza cuando no existe una muestra representativa y no hay aleatoriedad de variables perturbadoras. Los datos son recopilados con una intención

específica, mientras se desarrolla el estudio y, a partir de los mismos es posible el análisis de tipo estadístico.

Hay diferencias entre el estudio de caso y el experimento, el primero es observacional, y el último es un estudio controlado.

- **Investigación-acción**

Este método de tipo cualitativo es el más utilizado en sistemas de información e ingeniería de software, debido a que cumple dos premisas: proporcionar un beneficio al interesado de la investigación y fundamentalmente desarrollar un conocimiento de la investigación (Genero Bocco, Cruz-Lemus, & Mario, 2014).

Dentro de los métodos primarios se encuentran los métodos inductivos y deductivos; entre otros, y dependiendo del tipo de datos a manejar se distinguen en cuantitativos y cualitativos, (Genero Bocco, Cruz-Lemus, & Mario, 2014). Aunque Genero et al. (2014), prefiere distinguir por un lado en métodos fijos donde se establecen desde el inicio de la investigación todos los datos imprescindibles, este método se basa en datos cuantitativos. Por otro lado, están los métodos flexibles en donde los parámetros establecidos pueden variar en el transcurso de la investigación; y estos se basan en datos cualitativos y si es necesario en datos cuantitativos.

2.2.2. Métodos secundarios

Facilitan la recopilación de forma sistemática y precisa de los estudios primarios que van de la mano con la pregunta de investigación concreta, para así poder realizar una síntesis de las pruebas obtenidas para responder a la pregunta en cuestión. Como métodos de investigación secundaria están las revisiones sistemáticas de literatura (SLR) y mapeos sistemáticos. Dichos métodos permiten generar estudios secundarios y estudios terciarios.

- **Revisiones sistemáticas de literatura (SLR)**

Las revisiones sistemáticas de literatura permiten responder a preguntas de investigación concretas; estas respuestas surgen de la evaluación e interpretación de la investigación desarrollada. La revisión se apoya en un protocolo preestablecido para minimizar las desviaciones (Kitchenham, 2007).

- **Mapeos sistemáticos (Mapping)**

Según Kitchenham y Charters (2007) y Genero et al. (2014), los Mapeos sistemáticos brindan una perspectiva general o global sobre un área de investigación, ya que posee un alcance más profundo que las SLRs, y son utilizados para identificar la evidencia existente. Las diferencias entre los mapeos sistemáticos y los SLRs, difieren en que los primeros requieren menos tiempo y las preguntas de investigación son más amplias y en las SLRs son preguntas específicas.

En la Tabla 2.1, se presentan las diferencias entre los mapas sistemáticos (Mapping) y los SLRs, según Genero et al. (2014):

Elementos	Mapping	SLRs
Objetivos	Clasificación y análisis temático de la literatura sobre un tema específico de la ingeniería de software.	Identificar las mejores prácticas con respecto a procedimientos, tecnologías, métodos o herramientas específicas mediante la agregación de la información específica obtenida a partir de estudios empíricos.
Pregunta de investigación	Genérica, relacionada con tendencias de investigación, como, por ejemplo: que investigadores, cuantos estudios, que tipo de estudios, etc.	Específica, relacionada con resultados de estudios empíricos, como, por ejemplo: ¿es mejor el método/tecnología A que la B?
Proceso de búsqueda	Definido por el área de estudio o por el interés.	Definido por la pregunta de investigación la cual identifica la tecnología específica que está siendo investigada.
Alcance	Amplio – se incluye todos los artículos sobre un área de interés, pero solo se extrae de ellos datos para clasificarlos.	Centrado – solo se incluye artículos que contengan estudios empíricos relacionados con las preguntas de investigación y se extrae de ellos información detallada sobre los resultados obtenidos en cada uno de ellos.
Requisitos de la estrategia de búsqueda	A menudo menos estricta si solo se busca tendencias; como por ejemplo se puede buscar solo en un conjunto específico de publicaciones, limitándolas a artículos de revistas, o limitándolas a una de dos bibliotecas digitales.	Extremadamente exigente se deben encontrar todos los artículos relevantes. Generalmente además de buscar en las fuentes establecidas, puede ser necesario buscar en las referencias de los estudios primarios seleccionados o consultar a los

		expertos para incluir el mayor número de artículos posibles.
Evaluación de la calidad	No es esencial. Al incluir tantos estudios teóricos como empíricos de cualquier tipo, suele ser muy difícil definir un mecanismo de evaluación.	Es importante asegurarse de que los resultados se basan en la evidencia de mejor calidad.
Resultados	Un conjunto de artículos relacionados con un área de intereses clasificados en una serie de dimensiones, especificando el número total de artículos en cada dimensión.	Se agregan resultados de los estudios empíricos para contestar a las preguntas de investigación.

Tabla 2.1 Diferencias entre Mapping y SLRs (Genero *et al.*, 2014) (Runeson *et al.*, 2012)

2.3. En función al propósito

Según Genero et al. (2014), en función del propósito se pueden identificar cuatro tipos de investigaciones:

- **Exploratoria:** Permite indagar que es lo que sucede y a partir de ello desarrollar ideas y formular hipótesis para nuevas investigaciones.
- **Descriptiva:** Tiene como objetivo describir el panorama del estado actual de una situación o fenómeno objeto de estudio.
- **Explicativa:** Pretende establecer la causa que se encuentra inmersa en una situación o problema en una manera de correlación causal (causa-efecto).
- **Mejora:** Permite mejorar algún aspecto en especial del fenómeno estudiado.

En la Tabla nro. 2.2 se presenta una comparación de manera general de las principales estrategias de investigación.

Factor	Encuesta	Experimento	Estudio de caso	Investigación en acción
Tipo de diseño	Fijo	Fijo	Flexible	Flexible
Objetivo principal	Descriptivo	Explicativo	Exploratorio	Mejora
Naturaleza de los datos	Cuantitativa Cualitativa	Cuantitativa	Cuantitativa Cualitativa	Cuantitativa Cualitativa

Tabla 2.2 Representación general de las características de las estrategias empíricas de investigación (Genero *et al.*, 2014) (Runeson *et al.*, 2012)

2.4. Revisiones sistemáticas de la literatura

Las Revisiones Sistemáticas de Literatura o Systematic Literature Review (SLR), permiten que toda la investigación que se encuentra presente y disponible sea identificada,

evaluada y entendida, la misma que satisface la pregunta de investigación propuesta (Kitchenham, 2007). Las SLR forman parte de los estudios secundarios y todo estudio inmerso dentro de una SLR se considera estudio primario (Genero Bocco, Cruz-Lemus, & Mario, 2014).

Lo que incentiva el desarrollo de una SLR, es que permite nuevos descubrimientos y el planteamiento de ideas renovadas para próximas investigaciones (Zhang & Ali Babar , 2013). Aunque las SLR requieran mucho tiempo y esfuerzo, el desafío más predominante es conseguir un equilibrio entre la severidad metodológica y el esfuerzo demandado (Genero Bocco, Cruz-Lemus, & Mario, 2014).

Según Genero et al. (2014), desde algún tiempo se han realizado una gran variedad de investigaciones (Zhang & Ali Babar , 2013), (Kitchenham, y otros, 2010), (MacDonell, Shepperd, Kitchenham, & Mendes, 2010), etc. Cuyo propósito es servir de guía para mejorar el proceso de búsqueda de evidencia que responda a las interrogantes planteadas en la investigación. Luego de haber realizado SLRs se han dado a conocer las lecciones aprendidas con la finalidad de compartir sus experiencias (Dybå, Dingsøyr , & Hanssen , 2007), (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007).

2.4.1. Proceso de la revisión sistemática de literatura

Para el proceso de revisión sistemática de literatura (Kitchenham, 2007), se plantean tres principales puntos: planificar la revisión, realizar la revisión y reportar la revisión sistemática de literatura.

En la Figura nro. 2.1, se presenta el proceso a seguir para realizar una revisión sistemática de literatura.

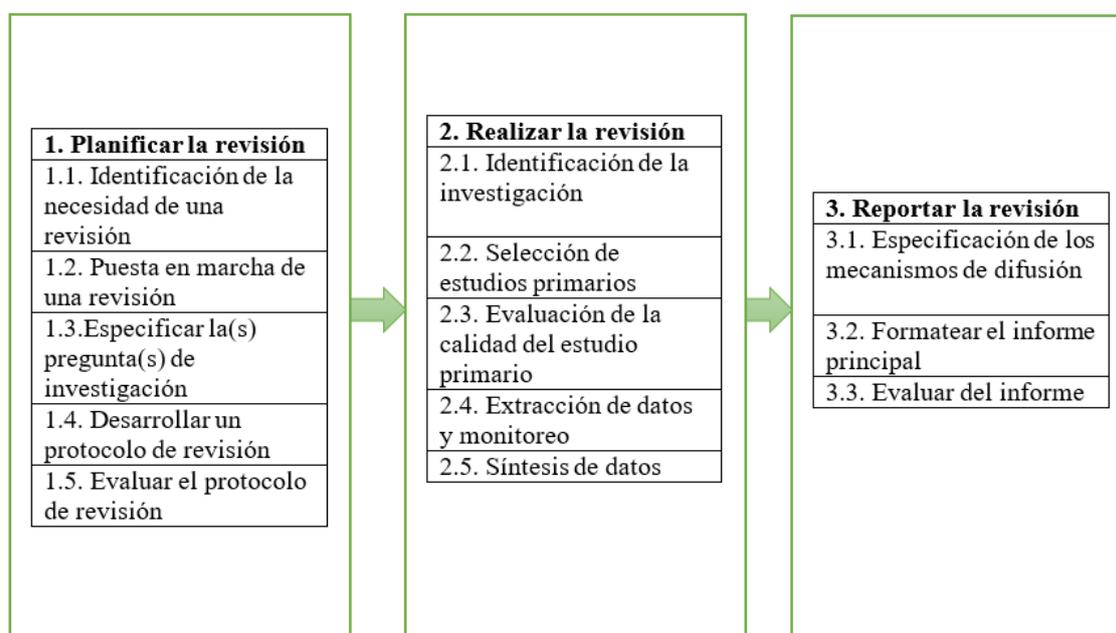


Figura 2.1 Proceso para realizar una SLR (Kitchenham, 2007)

A continuación, se da a conocer de manera más específica cada uno de los puntos que forman parte de las etapas de los procesos de la revisión sistemática de la literatura, las definiciones de las siguientes secciones se basan de acuerdo con la guía propuesta por Kitchenham (2007):

2.4.1.1. Planificar la revisión

Para la realización de una revisión sistemática de literatura se debe: identificar la necesidad de la revisión, especificar la pregunta de investigación, elaborar el protocolo, entre otros. Por consiguiente, se debe tener mayor atención a las preguntas de investigación, ya que las mismas acompañan a todo el proceso de la investigación. Todos estos aspectos se desarrollan a continuación:

2.4.1.1.1. Identificación de la necesidad de una revisión

En este punto es importante sintetizar toda la información que se ha obtenido sobre el tema de interés tratado; además, es totalmente válido empaparse de la existencia de otras revisiones sistemáticas relacionadas al tema; con el objetivo de examinar y argumentar otra revisión más; puesto que esta lectura minuciosa contribuirá con la especificación del protocolo de la nueva revisión sistemática que se pretende realizar.

2.4.1.1.2. Puesta en marcha de una revisión

Cuando una empresa demanda información sobre un tema concreto y se ven limitados a realizar una revisión sistemática de literatura por falta de tiempo o experiencia; delegan a un grupo de personas para que se encargue de este trabajo y la empresa genera un documento de puesta en marcha de la cuestión requerida que incluye:

- Título del Proyecto
- Antecedentes
- Preguntas de revisión
- Asesoría / Manejo de miembro de grupo (investigadores, profesionales, miembros, responsables de políticas, etc.)
- Métodos de la revisión
- Calendario del proyecto
- Estrategia de Difusión
- Infraestructura de Apoyo
- Presupuesto
- Referencia

La puesta en marcha de una revisión se puede obviar, e incluir la estrategia de difusión en el protocolo de revisión, si un investigador o un grupo de investigadores realizan una investigación que cubra sus propios intereses.

2.4.1.1.3. Especificar las preguntas de investigación

Este punto es el más importante de las revisiones sistemáticas, ya que las preguntas acompañan y orientan a todo el proceso de la investigación; con la finalidad de responder a las preguntas de investigación propuestas; por consiguiente, se deberán seleccionar los estudios primarios, extraer y sintetizar los datos precisos.

Para la redacción de las preguntas de investigación se debe considerar lo siguiente:

- **Población:** involucra a profesionales en la ingeniería con o sin experiencia, grupos industriales, pequeñas empresas entre otras; las cuales van a intervenir directamente en la investigación para recolectar la evidencia (Genero Bocco, Cruz-Lemus, & Mario, 2014) (Wohlin, y otros, 2012).

- **Intervención:** utilizada en el estudio empírico, en concreto que tecnología / instrumento / metodología se está estudiando.
- **Comparación:** que tecnología, instrumento o metodología permite comparar la intervención.
- **Resultados:** la fiabilidad o significancia, minimización de costes de producción y la reducción de tiempo de aparición en el mercado son factores de interés para los expertos, los cuales están ligados a los resultados. Por lo que se dice que los resultados deben ser significativos desde el punto de vista estadístico y práctico (Wohlin, y otros, 2012), (Kitchenham, 2007).
- **Contexto:** en este se desarrolla la comparación (la academia o la industria), las personas participes del estudio (practicantes, estudiantes, académicos, etc.), y las tareas que se realizan en el estudio.
- Se debe establecer el tipo de **diseño experimental**.

2.4.1.1.4. Desarrollar el protocolo de revisión

Con el afán de disminuir sesgos en la investigación es crucial definir un proceso formal y específico para desarrollar la revisión sistemática, por ello es necesario realizar pruebas durante la definición del protocolo en relación con la estrategia de búsqueda y extracción de datos (Genero Bocco, Cruz-Lemus, & Mario, 2014). Por lo cual Kitchenham (2007) propone los puntos que deben estar incluidos en el antes mencionado protocolo:

- **Antecedentes:** es primordial identificar la necesidad de la revisión, de tal manera que se pueda justificar porque se ha de realizar la investigación.
- **Preguntas de investigación:** una vez definidas las preguntas de investigación que la revisión deberá responder, estas formaran parte del protocolo.
- **Estrategia de búsqueda:** se refiera a la búsqueda de estudios primarios, para ello se necesita definir la cadena, período y fuentes de búsqueda como: bibliotecas digitales, revistas, y resúmenes de congresos, talleres, etc. En cuanto a la cadena de búsqueda Genero et al. (2014), aconseja que la misma debe ser definida en ingles debido a que es el lenguaje mundial para la investigación; además que, la cadena de búsqueda requerirá de un refinamiento repetitivo hasta conseguir que

la cadena satisfaga el tema de investigación; por lo general se da cuando el número de resultados de la búsqueda de los artículos es amplio. Para definir la cadena de búsqueda según Brereton et al. (2007), citado en (Genero Bocco, Cruz-Lemus, & Mario, 2014), se debe:

- Seleccionar los términos relevantes o principales.
- A partir de los términos relevantes, se debe identificar términos alternativos apoyándose en sinónimos o palabras relacionadas a los terminos principales.
- Ayudarse de los operadores lógicos como OR/AND, el primero para las palabras relacionadas, sinónimos y términos alternativos y el segundo para relacionar los términos relevantes.

Para establecer la cadena de búsqueda Genero *et al.* (2014), explica que es válido consultar a expertos en el tema de investigación, para que aporten ideas de cómo obtener términos relevantes que permitan armar la cadena de búsqueda, adicionalmente en ocasiones los términos suelen ser extraídos de los elementos de las preguntas de investigación (población, intervención, comparación, entre otros). Además, Genero *et al.* (2014), sugiere que se debe tomar en cuenta en que campos del artículo se va a indagar (título, resumen, o palabras claves, etc.).

En ciertas ocasiones al no contar con un número apropiado de artículos relevantes para el estudio, es necesario consultar en las referencias de los artículos, a esta técnica de la conoce como “bola de nieve “(Snowballing), (Genero Bocco, Cruz-Lemus, & Mario, 2014).

- **Snowballing**

De acuerdo con Wohlin (2014), el Snowballing es una técnica que consiste en revisar las referencias de uno o varios artículos de interés con el propósito de encontrar documentos adicionales relevantes que aporten con el estudio realizado. Además, el Snowballing permite verificar la fuente de donde fueron referenciados y citados los artículos. El término de usar las referencias del documento, se lo denomina bola de nieve hacía atrás (backward Snowballing) y el término usar las citas, se lo denomina bola de nieve hacia adelante (forward Snowballing).

Para realizar el Snowballing Wohlin (2014), menciona los siguientes pasos:

1. **Establecer el inicio:** Se establece un conjunto inicial tentativo de artículos de acuerdo con citas y referencias de las publicaciones encontradas en las bibliotecas digitales con la cadena de búsqueda generada; Wohlin (2014), recomienda para este paso identificar las citas y referencias de los artículos de Google Académico para evitar el sesgo a favor de cualquier editor específico.
2. **Iteraciones:** Una vez que se establece el conjunto de inicio se realiza una bola de nieve hacia atrás (backward Snowballing) o bola de nieve hacia adelante (forward Snowballing), para ello se examina el documento completo con la finalidad de determinar su inclusión o exclusión en la bola de nieve.
 - **Bola de nieve hacia atrás:** Se usa la lista de referencias para la selección de artículos nuevos a incluir.
 - **Bola de nieve hacia adelante:** Se seleccionan nuevos artículos basándose en las citas de los documentos previos.
 - **Inclusión y exclusión.** Wohlin (2014), menciona que se debe tener en cuenta la inclusión y la exclusión para poder continuar haciendo uso de un nuevo documento para la bola de nieve. una vez hecha la bola de nieve hacia adelante y atrás los documentos seleccionados estarán en una pila para proceder con otra iteración. Además solo los documentos encontrados que se derivan de documentos incluidos formarán parte del análisis.
3. **Autores:** Una vez terminada la iteración, al no encontrar documentos nuevos; Wohlin (2014), recomienda establecer contacto con algunos autores de los documentos incluidos, para una posible identificación de documentos adicionales.
4. **Extracción de datos:** Los documentos formarán parte de la extracción de datos, de acuerdo con las preguntas de investigación establecidas en el SLR.

En la Figura nro. 2.2, se presenta el proceso de Snowballing.

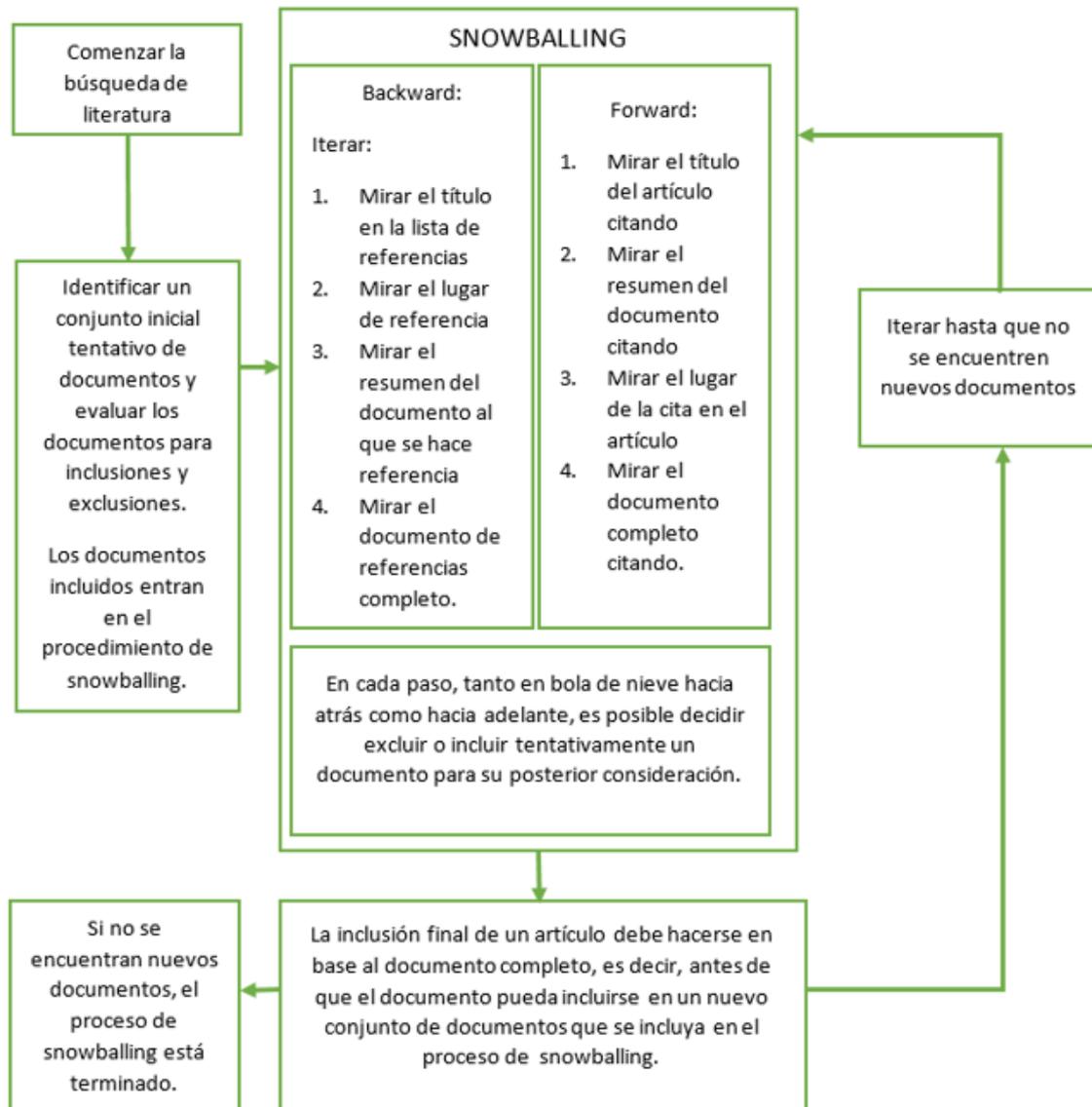


Figura 2.2 Proceso de Snowballing (Wohlin, 2014)

- **Criterios de selección de estudios:** se establecen a través de criterios de inclusión y exclusión, para verificar que artículos serán tomados en cuenta y cuáles no, en la revisión sistemática.
- **Procedimiento para la selección de estudios:** se detalla cómo se emplearán los criterios de selección de los estudios en la investigación, que acciones se tomaran en caso de haber desacuerdos entre los investigadores y de verificar quien se hará cargo de comprobar si la inclusión y exclusión se aplicó de manera adecuada.
- **Listas de verificación y procedimientos para la evaluación de la calidad de los estudios:** es necesario controlar la calidad de los

estudios primarios, para ver si se puede incluir en la investigación por lo que es necesario realizar listas de verificación.

- **Estrategia para la extracción de datos:** se debe definir como extraer los datos y un esquema de clasificación, con el fin de conseguir la información pertinente para dar respuestas oportunas y precisas a las preguntas de investigación, establecidas por los investigadores (Genero Bocco, Cruz-Lemus, & Mario, 2014).
- **Síntesis de los datos extraídos:** según (Genero Bocco, Cruz-Lemus, & Mario, 2014), para sintetizar los datos obtenidos, hay que tener en cuenta que si los estudios primarios permiten un meta de análisis formal, que emplea métodos estadísticos; caso contrario, al de no ser posible realizar una síntesis estadística, se debe realizar la síntesis con otros métodos como: síntesis narrativa, teoría fundamentada en los datos, análisis temático, meta-síntesis, entre otros.
- **Estrategia de difusión:** se debe definir como dar conocer los resultados obtenidos de la investigación.
- **Calendario del proyecto:** se debe definir un calendario que apoye la investigación, estableciendo actividades para intentar cumplir con el proyecto y evitar retrasos (Genero Bocco, Cruz-Lemus, & Mario, 2014).

2.4.1.1.5. Evaluar el protocolo de revisión

En la investigación sistemática de literatura es crucial definir un protocolo que permita establecer como evaluar el mismo. Se recomienda que sea evaluado por personas expertas en el tema, los cuales inclusive podrían validar el informe final de la SLR.

2.4.1.2. Realizar la revisión

Una vez realizada la planificación de la SLR, se procede a realizar la revisión la misma que contiene cinco tareas:

2.4.1.2.1. Identificación de la investigación

De acuerdo con la estrategia de búsqueda imparcial que se ha establecido en el protocolo; el objetivo de una revisión sistemática será encontrar publicaciones que

permitirán dar respuestas a las preguntas de investigación formuladas (Genero Bocco, Cruz-Lemus, & Mario, 2014).

Genero et al. (2014), menciona que los buscadores de las bibliotecas digitales suelen presentar limitaciones; además, de que existen circunstancias en las que se llevará a cabo un refinamiento de las cadenas de búsqueda, así como una modificación en los rangos de tiempo de búsqueda; todos estos cambios deben ser archivados. El autor recomienda que las publicaciones duplicadas que han sido descubiertas sean excluidas de la investigación.

2.4.1.2.2. Selección de estudios primarios

A través de la etapa de selección de estudios que está definido en el protocolo de revisión, se localizará todas las publicaciones relevantes que proporcionen evidencia, que permitan responder las preguntas de investigación, generando así una lista selecta de estudios primarios; la cual debe ser almacenada en algún sistema de gestión de referencias, conjuntamente con los artículos digitales completos, y la lista de publicaciones excluidas con su debida justificación (Genero Bocco, Cruz-Lemus, & Mario, 2014).

2.4.1.2.3. Evaluación de la calidad del estudio primario

No solo hay que evaluar los criterios de inclusión y exclusión; si no también la calidad de los estudios primarios que fueron escogidos en función a la lista de verificación y procedimiento de evaluación descrito en el protocolo (Kitchenham, 2007), para garantizar de esta manera, que los estudios que están incluidos en la SLR sean de calidad.

2.4.1.2.4. Extracción de datos y monitoreo

Una vez seleccionado los estudios que satisfacen la investigación, el siguiente paso es registrar con la información obtenida los formularios de extracción de datos que fueron establecidos en el protocolo, los mismos que fueron elaborados para responder las preguntas de investigación formuladas y los criterios de calidad del estudio, estos a su vez contienen información adicional como: nombre del investigador, título, autor, fecha de la extracción, revista, detalles de publicación (Kitchenham, 2007) .

2.4.1.2.5. Síntesis de datos

A partir de los datos recopilados se procede a sintetizar los resultados de los estudios primarios (Kitchenham, 2007), para ello se tendrán en cuenta el método establecido en el protocolo para esta actividad, contestando así las preguntas formuladas. Por lo general el resultado del resumen se ilustra con tablas o gráficos (Genero Bocco, Cruz-Lemus, & Mario, 2014).

2.4.1.3. Reportar la revisión

En este último paso se procede a elaborar el respectivo informe que según Genero et al. (2014), tiene como finalidad dar a conocer todo el proceso de revisión tomando en cuenta el medio de difusión elegido al momento de definir el protocolo; Kitchenham (2007), propone un formato para hacer reportes de la revisión.

Una vez realizado el informe es importante incluir:

- Amenazas a la validez, en el que se comenten las limitaciones del proceso de realización de la revisión sistemática de la literatura.
- Lecciones aprendidas para próximas investigaciones.

Finalmente, el informe debe ser enviado a un experto para que lo autentifique.

2.5. Bibliotecas digitales

Están diseñadas para admitir distintos tipos de búsquedas, dependiendo de las necesidades específicas del usuario, éstas varían una de otra, de acuerdo con el alcance geográfico e histórico, y por el tipo de contenido que estas disponen como son: revistas, libros, actas de conferencia, entre otros. También las búsquedas pueden ser limitadas en base a los campos accesibles de los documentos; además las bibliotecas digitales, cuentan con varias herramientas para recuperar resultados de investigaciones relevantes, visualizar y analizar los resultados obtenidos (WIPO, 2008).

Las búsquedas se pueden realizar directamente escribiendo la cadena a buscar y aplicando los siguientes operadores descritos en la Tabla nro. 2.3.

Operador	Descripción	Tipo
-----------------	--------------------	-------------

AND	Los términos detallados han de ser incluidos.	Booleanos
OR	Alguno de los términos definidos, tienen que ser incluidos.	Booleanos
NOT	La variable especificada debe ser excluido.	Booleanos
XOR	La búsqueda debe contener algún término especificado; pero no los dos a la vez.	Booleanos
+	La variable a continuación debe incluirse	Booleanos
-	El término siguiente debe excluirse	Booleanos
NEAR	Los términos introducidos deben estar cerca, sin importar su orden.	Proximidad
ADJ	Los términos introducidos deben estar juntos y en el orden definido.	Proximidad

Tabla 2.3 Operadores utilizados en consulta (WIPO, 2008)

2.5.1. IEEE Xplore

Es un repositorio para la investigación de tipo académico, facilita el ingreso total a literatura relacionada con las Ciencias de la Computación, Ingeniería Eléctrica y Electrónica; también se obtiene parcialmente el ingreso a diversos libros y libros electrónicos, mediante la opción de IEEE Xplore Books y en acuerdo con la editorial MIT Press y con Wiley e-book (Universidad Autónoma de Madrid, 2017).

Las publicaciones en IEEE Xplore están disponibles desde el año 1988 hasta la fecha, abarcan más de 3,7 millones de publicaciones en texto completo, facilitando el ingreso a: publicaciones periódicas (revistas), actas de congreso, libros electrónicos y otros. Además, alrededor de 20.000 publicaciones son incorporadas mensualmente. (Universidad Autónoma de Madrid, 2017).

La base de datos IEEE Xplore dispone de distintas interfaces de búsqueda tanto como para la investigación básica y la exploración por citas, las cuales están accesibles para el público en general. Así mismo dispone de una interfaz para emprender búsquedas: avanzadas estructuradas, por instrucciones, contenido completo, metadatos, autor, y búsqueda CrossRef, para registrados únicamente (WIPO, 2008).

Según WIPO (2008), indica que para realizar búsquedas básicas se puede emplear operadores booleanos básicos (AND, OR, y NOT); mientras que, en la búsqueda por citas, se puede emprender una exploración por diversos campos como son: DOI (Identificador de Objeto Digital), título de publicación, título del documento, volumen, año, entre otros. Las búsquedas avanzadas permiten hacer uso de palabras clave y operadores booleanos básicos en campos como: texto completo, metadatos, título del

documento o publicación, resumen, autor, entre otros. También las búsquedas avanzadas facilitan la aplicación de diversos filtros de acuerdo con el editor, tipo y año de publicación.

Las búsquedas en texto completo permiten hacer las siguientes operaciones de proximidad: NEAR, PARAGRAPH, SENTENCE, PHRASE, operadores relacionales, truncamientos y búsquedas de similitudes (WIPO, 2008).

La opción CrossRef admite búsquedas tanto en el contenido de IEEE, como en el de otros editores de manera paralela; dichos editores forman parte del programa CrossRef Search. Los afiliados a esta opción podrán almacenar y dar seguimiento a sus búsquedas (WIPO, 2008).

Los resultados de búsqueda pueden ser ordenados por fecha de publicación, relevancia, título; estos resultados incluyen referencias y citas que son solo accesibles para los que se encuentren suscritos.

2.5.2. SpringerLink

Es una fuente electrónica de información para áreas como: ciencias del comportamiento, matemáticas, informática, ingeniería, entre otras. (Facultad de Filosofía y Letras: UBA, 2005).

El repositorio ayuda a ingresar a más de 4 millones de publicaciones, cuenta con una interfaz para realizar búsquedas de contenido libre, así como para emprender búsquedas agrupadas. Para la introducción de una búsqueda, las interfaces admiten operadores como: AND, OR, NOT. También estas proveen de concordancia de frase, comodines, anidamiento y filtros que permiten reducir los resultados (WIPO, 2008).

Según WIPO (2008), indica que la diferencia entre la búsqueda estructurada y la búsqueda en contenido libre está en que la primera admite una búsqueda mixta en todo el texto o una búsqueda por campos como son: título del documento, autor, resumen del documento, entre otros. Además, la búsqueda estructurada permite establecer un rango de fechas para que se ejecute la búsqueda; sin embargo, aunque la búsqueda en texto libre permite realizar una exploración mediante una variedad de campos, es imposible buscar por rango de fechas.

Los resultados obtenidos de las búsquedas a más de poderlos ordenar, también pueden ser refinados reduciendo dichos resultados mediante la introducción de filtros

como: el tipo de contenido, lenguaje, disciplina, fecha de publicación, entre otros (WIPO, 2008).

2.5.3. ACM Digital Library

Es una plataforma en línea que contiene artículos completos de las publicaciones de ACM (Association for Computing Machinery), también proporciona acceso a las publicaciones de otras organizaciones miembros de la ACM (Biblioteca de Informática y Documentación “Enric V, 2014).

ACM permite realizar búsquedas simples, búsquedas avanzadas o navegar por los diferentes índices. La búsqueda simple permite ingresar una palabra o una frase, se busca cada palabra individualmente, se puede incluir comillas para buscar una frase exacta; mientras la búsqueda avanzada proporciona un formulario para realizar la búsqueda por campos determinados y filtrar por fechas de publicación. (Biblioteca de Informática y Documentación “Enric V, 2014).

Los resultados que se obtienen de las búsquedas se pueden ordenar por relevancia, título, publicación, fecha, editor entre otros, además ACM proporciona un apartado para refinar los resultados por nombre del autor, editores, nombre o año de la publicación o por tipo de conferencias (Biblioteca de Informática y Documentación “Enric V, 2014).

Capítulo 3. Planeación del SLR

3.1. Introducción

La revisión sistemática de literatura (SLR) consta de varias fases como se mencionó en el Capítulo nro. 2, según Kitchenham & Charters (2007), la primera fase en una revisión sistemática de literatura es la planificación de la revisión, dentro de ella se debe realizar varias actividades tales como: Identificar la necesidad de la revisión, detallar las preguntas de investigación y desarrollar el protocolo de revisión.

En el protocolo de revisión están establecidos los procedimientos y pasos que se debe seguir durante el proceso de revisión sistemática de literatura de un trabajo de investigación específico.

3.2. Campo a estudiar

El estudio se desarrolla dentro del campo del Open Source Software, específicamente; los factores de éxito y fracaso, riesgos, beneficios, barreras, documentación y opinión del usuario frente a la adopción de este.

3.3. Identificación de la necesidad de una revisión

La revisión sistemática de literatura se la realiza con la finalidad de obtener documentación relacionada a la adopción del Software Libre, de esta manera se pueda conocer los retos que afrontan las organizaciones al adoptar software libre y su influencia en la transición hacia una economía social del conocimiento común y abierto. Comprender las ventajas y desventajas de la implementación de software libre en las organizaciones, así como también, los pasos que deben ser aplicados para alcanzar el éxito, evitar el fracaso y evadir el riesgo.

Luego del análisis que se mencionó en el capítulo 1, en la sección de justificación en el apartado de artículos relacionados, se llegó a la conclusión que no existe ninguna revisión sistemática de literatura que este enfocada concretamente al objetivo planteado en este trabajo de titulación. Los resultados de la revisión sistemática de literatura

contribuirán a las diferentes organizaciones ya sean públicas o privadas; con una guía para la correcta adopción de software libre.

3.4. Preguntas de investigación

Una de las partes fundamentales que no deben pasar desapercibidas en un SLR, según Kitchenham & Charters (2007) es la formulación de las preguntas de investigación. Las preguntas son el sustento que acompaña todo el proceso de investigación, son la guía base de la revisión, ya que permite discernir y tomar decisiones sobre la misma. En base a las preguntas se debe proporcionar calidad a la investigación mediante la selección de artículos con contenido científico que apunten a la investigación planteada, además, disminuir relativamente los tiempos de búsqueda de los artículos antes mencionados.

Para determinar las preguntas de investigación se debe tener presente algunos puntos que se describieron en el Capítulo nro. 2.

- **Población:** Esta investigación está dirigida a la organización pública y privada que deseen adoptar software libre.
- **Intervención:** Se enfoca en la adopción del software libre, donde se incluyen: éxitos, fracasos, riesgos, beneficios, barreras, documentación y opinión de usuarios al adoptarlo.
- **Comparación:** En este caso se omite este componente y el formato se convierte en PIO.
- **Resultados:** Proporcionar calidad a la investigación mediante la selección de artículos con gran contenido científico que apunten a la investigación planteada, además disminuir relativamente los tiempos de búsqueda de antes mencionados artículos.

La Tabla nro. 3.1, presenta las preguntas de investigación propuestas para este trabajo.

Preguntas	Motivación
<p>PI. ¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?</p>	<p>Se desea conocer qué factores positivos son determinantes para una adopción exitosa de software libre en las organizaciones y entender que factores conllevan al fracaso de dicha implementación.</p>

P2. ¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre?	Establecer los riesgos que conlleva la adopción de software libre.
P3. ¿Cuáles son los beneficios de la utilización de software libre?	Encontrar qué beneficios se consigue al momento de adoptar software libre, en caso de haber ventajas que pruebas existe de ello.
P4. ¿Cuáles son las barreras que han identificado las organizaciones al adoptar software libre?	Obtener información de las barreras que impiden adoptar software libre a las organizaciones.
P5. ¿En qué dominios de software ha resultado exitosa la adopción de software libre?	Determinar y clasificar los dominios en el que el software libre está siendo usado y teniendo popularidad.
P6. ¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?	Conocer si existe suficiente documentación acerca del software libre para una implementación exitosa.
P7. ¿Se ha incluido la opinión del usuario del uso del software libre?	Obtener literatura relevante que dé a conocer, si se toma en cuenta la opinión de usuario, antes y después de la implementación de software libre en las organizaciones.

Tabla 3.1 Preguntas de investigación (Autoría propia, 2018)

3.5. Protocolo

3.5.1. Estrategia de búsqueda de estudios primarios.

Las siguientes subsecciones describen las estrategias utilizadas para la búsqueda de estudios primarios relevantes para este SLR:

3.5.1.1. Determinación de la cadena de búsqueda genérica

Mediante palabras concisas, términos relevantes y términos alternativos, que nacen de las preguntas de investigación. Se pretende formar la cadena de búsqueda, la cual será aplicada en las fuentes indicadas más adelante. La cadena de búsqueda dependiendo del buscador de cada biblioteca digital se lo hará por título y resumen, caso contrario por el texto completo. La cadena de búsqueda se pondrá a consideración del experto. En la Tabla nro. 3.2, se encuentran los términos relevantes y alternativos.

Términos relevantes	Términos alternativos
Success	
Failure	
Open Source Software	OSS OR FOSS OR FLOSS
Benefit	Advantage
Risk	
Adoption	

Organization	Company
Documentation	
Implementation	Utilization
Impediments	Barriers
User	
Opinion	

Tabla 3.2 Términos relevantes y alternativos (Autoría propia, 2018)

Para formar la cadena de búsqueda, se seleccionaron los términos relevantes, los cuales fueron identificados con la ayuda de la lectura de SLRs con relación al tema investigado, estos artículos son mencionados en el Capítulo nro. 1; partiendo de los términos relevantes se identificaron términos alternativos y con la ayuda de los operadores lógicos “AND” y “OR” se enlazaron los términos relevantes y alternativos. Además, se contó con la consultoría de expertos en temas de investigación, dando como resultado la siguiente cadena genérica resultante.

Title: (Open Source Software OR OSS OR FOSS OR FLOSS) **AND Abstract:** (Open Source Software OR OSS OR FOSS) **AND** (risk OR failure OR success OR barriers OR Advantage OR Opinion OR Impediments OR FLOSS)

Los términos: benefit, adoption, organization, company, documentation, implementation, utilization y user no se incluyeron en la cadena genérica al no presentar cambios relevantes en los resultados de la búsqueda, al incluirlos en la cadena de búsqueda, para el desarrollo del estudio.

3.5.1.2. Seleccionar los recursos de búsqueda en los que se debe buscar (bibliotecas digitales, revistas específicas, y resúmenes de congresos).

Las bibliotecas digitales seleccionadas para el estudio que se presentan en la Tabla nro. 3.3, fueron elegidas porque cubren un gran volumen de artículos relacionados a la ingeniería de software, son utilizadas por gran parte de revisiones sistemáticas relacionadas a la rama del estudio y son a las que normalmente tienen acceso las universidades del Ecuador. No se incluyeron otras bibliotecas digitales porque con la consulta del experto en el tema se consideró suficiente para el estudio.

Bibliotecas digitales	URL	Responsables
IEEE Xplore Digital Library.	http://ieeexplore.ieee.org/Xplore/home.jsp	Oscar Armijos, Fernando Delgado

Springer Link.	https://link.springer.com/	Oscar Armijos, Fernando Delgado
ACM Digital Library	https://dl.acm.org/	Oscar Armijos, Fernando Delgado

Tabla 3.3 Bibliotecas seleccionadas (Autoría propia, 2017)

Los estudios primarios que se obtengan de la búsqueda de estas bibliotecas se complementaran con estudios que obtengan de:

- Búsqueda manual en conferencias y talleres.
- Búsqueda manual en revistas y libros.

3.6. Criterios de selección de estudios.

A continuación, se indica los criterios de selección de estudios los cuales permiten determinar que documentos cumplen con los criterios para la inclusión y cuales no para la exclusión de estos en la investigación.

3.6.1. Criterios de inclusión:

- Todos los artículos que proporcionen información relevante a las preguntas de investigación, a partir del año 1998 donde fue acuñado el término Open Source Software (Petrijevcenin Vuksanovic & Sudarevic, 2012), (Fitzgerald, 2006).Tipos de publicaciones que corresponda a: artículos de revistas, actas de conferencia y capítulos de libro.
- Artículos en idiomas inglés y español.

3.6.2. Criterios de exclusión:

- Artículos duplicados.
- Artículos que no aporten a la investigación.
- Artículos con cero citas, que no aporten ampliamente a las preguntas de investigación.
- Artículos que no correspondan a los tipos de publicaciones definidos para el estudio; como también pósteres.
- Artículos que no estén en los idiomas definidos.

3.7. Procedimientos de selección de estudios

Cada estudio obtenido producto de la búsqueda manual o automática en las bibliotecas digitales: IEEE Xplore Digital Library, Springer Link, ACM. Tendrá el siguiente proceso:

- Descargar el listado de artículos a partir de la aplicación de la búsqueda automática.
- De los artículos obtenidos se leerá el título y el resumen, calificándoles con: “Si”, “No”, “Tal vez”, en base a los criterios de inclusión y exclusión, actividad ejecutada por un primer grupo de investigadores, en primera instancia.
- Un segundo grupo de investigadores revisara la calificación de los artículos seleccionados en primera instancia, para validar la calificación obtenida en la primera instancia.
- Los artículos que se marcados con “SI” o “Tal vez” en segunda instancia serán leídos en su totalidad.
- Finalmente se seleccionarán solo los estudios que cumplan los criterios de inclusión definidos en el protocolo y se rechazarán los artículos que se alineen con los criterios de exclusión.

3.8. Listas de verificación y procedimientos de evaluación de calidad del estudio.

A más de establecer criterios de inclusión/exclusión, en el protocolo se ha definido una lista de verificación a ser utilizada en el control de calidad para cada estudio individual. en la lista se establece una puntuación para cada criterio de inclusión.

3.8.1. Criterios de evaluación

Los artículos seleccionados se evaluarán siguiendo los siguientes criterios

- **Criterio 1:** ¿El estudio seleccionado cumple con los criterios de inclusión o no cumple con los de exclusión?

Las posibles respuestas para este criterio son:

- ‘1’, si los criterios de inclusión están presentes en el artículo.

- '0', si los criterios de inclusión no son claramente definidos.
- '-1', si los criterios de inclusión no están presentes en el artículo.
- **Criterio 2:** ¿El estudio responde a las preguntas de investigación?
Las posibles respuestas para este criterio son:
 - '1', si cada trabajo responde al menos a una pregunta de investigación amplia
 - '0', si el trabajo responde a una pregunta de investigación resumida
 - '-1', si no se responde a ninguna pregunta de investigación.
- **Criterio 3:** ¿El estudio abarca temas sobre éxito, fracaso, riesgo, barreras, dominios exitosos, documentación comprensible, opinión del usuario, en la adopción de software libre?
Las posibles respuestas para este criterio son:
 - '1', si los artículos corresponden a una investigación amplia de adopción de software libre en las organizaciones
 - '0', si corresponde a una investigación resumida de adopción de OSS
 - '-1', si el artículo no corresponde a una investigación de adopción de OSS.
- **Criterio 4:** ¿El estudio ha sido citado por otros autores?
Las posibles respuestas para este criterio son:
 - '1', si los artículos han sido citados por al menos seis autores
 - '0', si el artículo ha sido citado por uno hasta cinco autores
 - '-1', si el artículo no ha sido citado.

Para artículos los cuales responden ampliamente las preguntas de investigación, se omitirá el criterio 4.

3.8.2. Formulario para la evaluación de calidad

En la Tabla nro. 3.4, se presenta el formulario de evaluación de calidad que se aplicará a los artículos seleccionados.

Biblioteca:	
Título:	
Código:	
Fecha de publicación:	
DOI:	
Citaciones Google Académico:	
Criterio 1: ¿Los estudios seleccionados cumplen con los criterios de exclusión o inclusión?	1 / 0 / -1
Criterio 2: ¿En la indagación realizada, cuan probable es haber cubierto toda la literatura relevante?	1 / 0 / -1
Criterio 3: ¿Los estudios responden a las preguntas de investigación?	1 / 0 / -1
Criterio 4: ¿Los artículos abarcan temas sobre éxito, fracaso, riesgo en la adopción de software libre?	1 / 0 / -1
Criterio 5: ¿El estudio ha sido citado por otros autores?	1 / 0 / -1

Tabla 3.4 Formulario de evaluación de calidad (Autoría propia, 2018)

3.9. Estrategia de extracción de datos.

Con la finalidad de extraer datos para brindar respuestas a las preguntas de investigación antes formuladas para el estudio, se presenta la siguiente estrategia de extracción que consta de 4 partes: los metadatos del estudio y un formulario en base a las preguntas de investigación, objetivo, y observaciones:

1) Los datos extraídos (metadatos) de cada artículo serán:

- Nombre de la biblioteca digital
- Numero de articulo
- Título del documento
- El año de publicación del documento
- Identificador de objeto digital D.O.I.
- Autores
- Código:

- **IEEE:** IEEE-Iniciales_revisores+número
- **ACM:** ACM- Iniciales_ revisores+número
- **Springer:** SPR- Iniciales_ revisores+número
- **Snowballing:** SNO- Iniciales_ revisores+número

2) Formulario de extracción en base a las preguntas de e investigación:

✓ **P1. ¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?**

En la Tabla nro. 3.5, se muestran el formulario de extracción de factores de éxito y fracaso de la adopción de FOSS

			Descripción	Valores
¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?	Factores de éxito	Capacitación	Capacitación de las partes interesadas en la adopción de OSS.	Boolean 0/1
		Costes	Beneficios económicos que existan a partir de la adopción del OSS.	Boolean 0/1
		Gestión eficaz del SE	Administrar correctamente la evolución del software.	Boolean 0/1
		Actividad Asociada	Cantidad de actividad que produce la comunidad en apoyo al OSS	Boolean 0/1
		Soporte	Ayuda que se pueda administrar a la organización con respecto al OSS.	Boolean 0/1
		Políticas	Políticas corporativas, de TIC y de adopción del OSS.	Boolean 0/1
		Otros	Otros factores de éxito contemplados en la adopción del OSS que no figuran en las anteriores definiciones.	Conjunto de factores de éxito de adopción de OSS separados por filas.
	Factores de fracaso	Licencias	Problemas relacionados con las licencias del OSS.	Boolean 0/1
		Baja Competitividad	Poca competitividad al frente de ofertas comerciales.	Boolean 0/1
		Políticas	Ausencia de políticas de OSS, TIC u adopción.	Boolean 0/1
		Otros	Otros factores de fracaso contemplados en la adopción del OSS que no figuran en las anteriores definiciones.	Conjunto de factores de fracaso adopción de OSS separados por filas.

Tabla 3.5 Factores de adopción de FOSS (Autoría propia, 2018)

✓ **P2. ¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre a nivel mundial?**

En la Tabla nro. 3.6, se muestra el formulario de extracción de los riesgos de adopción de FOSS.

			Descripción	Valores
¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre a nivel mundial?	Riesgos en la adopción	Licencias	Infracciones ocurridas por la falta de conocimiento de las licencias del OSS	Boolean 0/1
		Escalabilidad, seguridad y rendimiento		Boolean 0/1
		Propiedad intelectual	Posible riesgo de pérdida de derechos intelectuales.	Boolean 0/1
		Falta de garantía	Problemas debido a que no existen responsables definidos para el soporte de OSS.	Boolean 0/1
		Costos	Problemas con valores reales generados por los costos de OSS.	Boolean 0/1
		Otros	Identificación de otros posibles riesgos que se pueden dar en la adopción de OSS.	Conjunto de posibles riesgos que se pueden dar en la adopción de OSS separados por filas.

Tabla 3.6 Formulario de extracción de riesgos de adopción de FOSS (Autoría propia, 2017)

✓ **P3. ¿Cuáles son los beneficios de la utilización de software libre?**

En la Tabla nro. 3.7, se muestra el formulario de extracción de beneficios de adopción de FOSS.

				Descripción	Valores
¿Cuáles son los beneficios de la utilización de software libre?	Beneficios de utilizar OSS	Costes	Rentabilidad	Beneficios con relación a valores monetarios	Boolean 0/1
			Ahorro		Boolean 0/1
			Bajo costo de adquisición		Boolean 0/1
			No produce gastos		Boolean 0/1

		Flexibilidad	Disponibilidad de código fuente	Resultado de la disponibilidad del código fuente y el derecho a modificarlo libremente y adaptarlo cuando las necesidades del usuario	Boolean 0/1
			Personalizable		Boolean 0/1
		Tiempo	Reduce tiempo en encontrar mejoras	Ahorro de tiempo con la implementación de OSS.	Boolean 0/1
			Reduce tiempo en detectar errores		Boolean 0/1
		Participación de la comunidad	Solucionar errores	Ayuda de la comunidad para resolver errores o contribuir en el OSS.	Boolean 0/1
			Contribuciones		Boolean 0/1
		Desarrollo	Fuente de desarrollo de un país	Desarrollo económico de un país	Boolean 0/1
			Genera empleo		Boolean 0/1
		Independencia de proveedores			Boolean 0/1
		Facilita la integración de componentes		Ventaja de integrar gran diversidad de componentes fácilmente	Boolean 0/1
		Otros		Identificación de otros beneficios encontrados que no se encuentran listados anteriormente	Conjunto de posibles beneficios que se pueden encontrar en la adopción de OSS separados por filas.

Tabla 3.7 Formulario de extracción de beneficios de adopción de FOSS (Autoría propia, 2017)

✓ **P4. ¿Cuáles son las barreras que han identificado las organizaciones al adoptar software libre?**

En la Tabla nro. 3.8, se muestra el formulario de extracción de barreras ante la adopción de FOSS

				Descripción	Valores
¿Qué impide a las organizaciones adoptar software libre?	Barreras al adoptar OSS	Factores humanos	Incertidumbre	Consiste en los problemas que se pueden dar por parte de los que conforman la organización	Boolean 0/1
			Experiencia		Boolean 0/1
			Capacitación		Boolean 0/1
			Rechazo del OSS		Boolean 0/1
			Insuficiencia de personal		Boolean 0/1
			Tiempo del personal		Boolean 0/1
		Seguridad	Preocupaciones	Problemas suscitados con la seguridad en la utilización de OSS	Boolean 0/1
			Mejoras		Boolean 0/1
		Políticas		Falta de políticas para implementar OSS	Boolean 0/1
		Compatibilidad		Problemas de compatibilidad con componentes externos.	Boolean 0/1
		Migración		Un proceso de cambio de un software a otro, o bien de un sistema operativo a otro.	Boolean 0/1
		Integración		Problemas con la transferencia de información de un sistema otro.	Boolean 0/1
		Licencias		Problemas con las licencias al no estar claras o sean mal utilizadas.	Boolean 0/1
		Soporte		Falta de soporte de la comunidad o desarrolladores.	Boolean 0/1
Otros		Otras barreras que encuentran las organizaciones al adoptar OSS que no fueron detalladas anteriormente.	Conjunto de barreras que impiden adoptar OSS separados por filas.		

Tabla 3.8 Formulario de extracción de barreras para la adopción de FOSS (Autoría propia, 2017)

✓ **P5. ¿En qué dominios de software ha resultado exitosa la adopción de software libre a nivel mundial?**

En la Tabla nro. 3.9, se muestra el formulario de extracción de dominios exitosos en la adopción de FOSS.

					Descripción	Valores
¿En qué dominios de software ha resultado exitosa la adopción de software libre a nivel mundial?	Dominios exitosos	Software de sistema	Operativo		Software principal o conjunto de programas de un sistema informático que gestiona hardware y provee servicios a los programas de ejecución de software.	Boolean 0/1
			Servidores	Web	Aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia	Boolean 0/1
				Correo		
				Archivos e impresión		
		Diagnostico	Seguridad	Sistema de información segura y confiable	Boolean 0/1	
		Software de aplicación	Ofimática		Aplicaciones informáticas utilizadas en oficinas, para realizar diferentes funciones sobre archivos o documentos.	Boolean 0/1
			Educativo		Programas avanzados vinculados a la educación.	Boolean 0/1
			Médico		Programas informáticos utilizados para fines médicos.	Boolean 0/1
			Base de Datos		Programas que permiten almacenar y acceder a los datos de forma rápida y estructurada.	Boolean 0/1
			Navegadores		Programas que permiten acceso a la web.	Boolean 0/1
		Otros			Otros dominios exitosos no detallados anteriormente (introducir nombres de dominios)	Conjunto de dominios éxitos en la adopción de OSS

					separados por comas)	separados por filas.
--	--	--	--	--	----------------------	----------------------

Tabla 3.9 Formulario de extracción de dominios exitosos en la adopción de FOSS (Autoría propia, 2017)

✓ **P6. ¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?**

En la Tabla nro. 3.10, se muestra el formulario de extracción de la pregunta ¿La documentación generada por la comunidad desarrolladora es suficientemente comprensible para implementarlo eficientemente?

		Descripción	Valores
¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?	Documentación necesaria		1 suficiente, 0 escasa, -1 no existe.

Tabla 3.10 Formulario de extracción de la pregunta ¿La documentación generada en la adopción de OSS (Autoría propia, 2017)

✓ **P7. ¿Se ha incluido la opinión del usuario sobre el uso del software libre?**

En la Tabla nro. 3.11, se muestra el formulario de extracción de la pregunta ¿Se ha incluido la opinión del usuario sobre el uso de software libre?

		Descripción	Valores
¿Se ha incluido la opinión del usuario del uso del software libre?	Opinión del usuario		1 Si, -1 No, 0 No habla

Tabla 3.11 Formulario de extracción de la pregunta ¿Se ha incluido la opinión del usuario sobre el uso del software libre (Autoría propia, 2017)

3) Objetivo y observaciones:

- ✓ En el objetivo, se deberá indicar de que trata el estudio, si aporta o no con la investigación.

- ✓ En el campo observaciones se deberá describir aspecto relevante encontrado en el artículo.

3.10. Síntesis de los datos extraídos.

Para la síntesis de los datos se utilizará una síntesis narrativa juntamente con tablas y gráficos, a partir de los resultados obtenidos en los estudios primarios.

CAPÍTULO 4. Desarrollo del SLR

4.1. Introducción

Este capítulo desarrolla el protocolo de investigación descrito en el capítulo anterior. Como punto de partida se identifican estudios primarios relevantes, mediante definición y la posterior aplicación de cadenas de búsqueda en las bases de datos descritas en el protocolo. Se evaluó la calidad de cada uno de los estudios identificados como potencialmente relevantes con el objetivo de determinar si debían o no formar parte de la investigación de acuerdo con el umbral establecido; posteriormente, se realizó una segunda búsqueda, aplicando la técnica de bola de nieve (Snowballing) a partir de las referencias importantes incluidas en los estudios primarios seleccionados. El formulario de extracción de datos fue completado con la información extraída de los artículos seleccionados.

4.2. Identificación y selección de estudios

4.2.1. Identificación de estudios

Al usar la estrategia de búsqueda definida en el protocolo, en el Capítulo nro. 3, se trató de encontrar un conjunto de publicaciones relevantes en las tres bibliotecas digitales que se determinaron para el estudio, éstas son: IEEE Xplore, ACM Digital Library, Springer Link. Con estos documentos se pretendía obtener respuesta a las preguntas de investigación.

Para la correcta identificación de las publicaciones se llevó a cabo una prueba piloto en tres librerías digitales; mediante la combinación de las palabras claves y términos relativos, con la ayuda de expertos en el tema y de los operadores lógicos “AND” y “OR” para determinar los términos a usar en la prueba piloto. Luego, se continuó con el refinamiento de las cadenas de búsqueda para cada biblioteca, que consta de dos fases, las cuales se explican en las secciones siguientes, teniendo la siguiente cadena resultante: **Title:** (Open Source Software OR OSS OR FOSS OR FLOSS) **AND** **Abstract:** (Open Source Software OR OSS OR FOSS) **AND** (risk OR failure OR success OR barriers OR Advantage OR Opinion OR Impediments OR FLOSS). La búsqueda por

lo general se realizó considerando el título y el resumen del documento; esto no fue realizado en la biblioteca de Springer Link que no permitía este tipo de búsqueda, en la cual se realizó la búsqueda por título y texto completo.

Todo lo antes mencionado se realiza debido a que las bibliotecas digitales presentan ciertos inconvenientes al momento de ejecutar la investigación; es por este motivo, que es necesario refinar la cadena de búsqueda en cada una de las librerías tal como se describe en (Genero Bocco, Cruz-Lemus, & Mario, 2014).

A continuación, se detalla las pruebas piloto, y los refinamientos de la cadena de búsqueda por cada biblioteca digital:

4.2.1.1. Biblioteca Digital IEEE Xplore

Como se explicó en la Sección nro. 2.5, las bibliotecas digitales admiten diferentes clases de búsqueda; entre ellas están las búsquedas avanzadas, que soportan distintos tipos de operadores (booleanos y filtros), que permiten ajustar la búsqueda de documentos relevantes para el estudio.

En la Figura nro. 4.1, se puede apreciar una búsqueda avanzada en IEEE Xplore. La búsqueda avanzada permitirá introducir la cadena de búsqueda definida.

IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites Cart (0) | Create Account | Personal Sign In

IEEE Xplore[®]
Digital Library Institutional Sign In **IEEE**

Browse ▾ My Settings ▾ Get Help ▾ Subscribe

Advanced Search Options

Advanced Keyword/Phrases **Command Search** Citation Search Preferences

ENTER KEYWORDS, PHRASES, OR A BOOLEAN EXPRESSION
Note: Use the drop down lists to generate the correct Operator and Data Field Codes.
This wizard will NOT build your expression. [View examples of how to write a boolean search string](#)

Search: Metadata Only Full Text & Metadata

Data Fields ▾ Operators ▾

```
((\"Abstract\". QT.Open Source Software QT.) OR (\"Abstract\". QT.OSS.QT.) OR (\"Abstract\". QT.FOSS.QT.)) AND (\"Abstract\".risk OR (\"Abstract\". QT.failure QT.) OR (\"Abstract\".success) OR (\"Abstract\".barriers) OR (\"Abstract\".Advantage) OR (\"Abstract\".Impediments) OR (\"Abstract\".FLOSS)) AND \"Document Title\". QT.Open Source Software.QT OR \"Document Title\". QT.OSS.QT. OR \"Document Title\". QT.FOSS.QT OR \"Document Title\". QT.FLOSS.QT
```

SEARCH GUIDELINES
Operators need to be in all caps
- i.e. AND/OR/NOT/NEAR.

Asterisk wildcards cannot be used within quotes or with the NEAR/ONEAR operators.

There is a maximum of 15 search terms.

[LEARN MORE ABOUT](#)
• Data Fields »
• Search Examples »
• Search Operators »
• Search Guidelines »

Figura 4.1 Búsqueda avanzada en IEEE Xplore (Base de datos IEEE Xplore, 2017)

La Figura nro. 4.2, presenta los pasos que se han seguido en la identificación de publicaciones relevantes en la biblioteca IEEE Xplore.

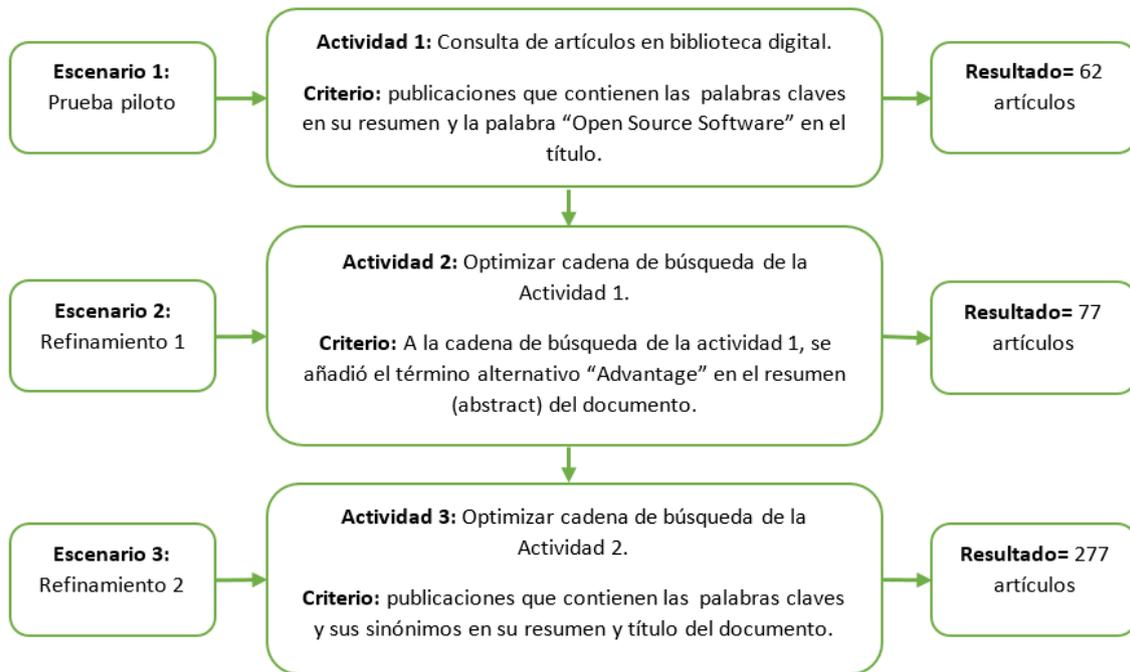


Figura 4.2 Pasos para la identificación de publicaciones en biblioteca IEEE Xplore (Autoría propia, 2017)

A continuación, se describen las cadenas que fueron usadas en IEEE Xplore, las cuales estuvieron construidas con los términos relevantes como “palabras claves”, operadores booleanos “AND”, “OR” y los campos de datos como: el resumen del artículo (Abstract) y el título del documento (Document Title). Para cada uno de los siguientes escenarios: prueba piloto, refinamiento 1 y refinamiento 2.

4.2.1.1.1. Prueba piloto

La primera cadena de búsqueda fue realizada con la utilización de las palabras claves del resumen (Abstract) de los artículos. Las palabras claves y alternativas consideradas fueron: “Open Source Software”, “OSS”, “FOSS”, “risk”, “failure”, “success” y “barriers”. Además, se consideró que el título del documento (Document Title) contenga la cadena “Open Source Software”. Los resultados se presentan en la Tabla nro. 4.1.

Cadena de búsqueda	Resultado	Descripción
(("Abstract":.QT.Open Source Software.QT.) OR ("Abstract":.QT.OSS.QT.) OR ("Abstract":.QT.FOSS.QT.)) AND ("Abstract":risk OR ("Abstract":.QT.failure.QT.) OR ("Abstract":success) OR ("Abstract":barriers)) AND "Document Title":.QT.Open Source Software.QT.	62	Para la primera cadena de búsqueda, se realizó la exploración en el resumen del documento y cuyo título contenga la cadena "Open Source Software".

Tabla 4.1 Primera cadena de búsqueda IEEE (Autoría propia, 2017)

4.2.1.1.2. Refinamiento 1

En esta fase se procedió a optimizar la cadena de búsqueda definida en la etapa anterior de la "prueba piloto". La cadena de búsqueda se describe en la Tabla nro. 4.2, esta tabla contiene los mismos términos relevantes y alternativos de la prueba piloto, excepto que en esta prueba se agregó la palabra "advantage" en el resumen (Abstract), para aumentar la cantidad de artículos relevantes para el estudio."

Cadena de búsqueda	Resultado	Descripción
(("Abstract":.QT.Open Source Software.QT.) OR ("Abstract":.QT.OSS.QT.) OR ("Abstract":.QT.FOSS.QT.)) AND ("Abstract":risk OR ("Abstract":.QT.failure.QT.) OR ("Abstract":success) OR ("Abstract":barriers) OR ("Abstract":Advantage)) AND "Document Title":.QT.Open Source Software.QT.	77	Para la segunda cadena de búsqueda, se realizó la exploración de las palabras claves y sus sinónimos en el resumen del documento y cuyo título contenga la cadena "Open Source Software".

Tabla 4.2 Refinamiento 1 cadena de búsqueda IEEE (Autoría propia, 2017)

4.2.1.1.3. Refinamiento 2

Luego de aplicar el refinamiento 1, se vio la necesidad de aplicar uno nuevo, porque los artículos resultantes eran insuficientes, para llevar a cabo el estudio; además, algunos términos relevantes y relativos no fueron tomados en cuenta en el resumen de los documentos en las etapas anteriores, como: "FLOSS", "Impediments". Se consideró importante que el título del documento (Document Title) contenga las cadenas relevantes y alternativas como: "Open Source Software", "FOSS", "OSS", "FLOSS".

La Tabla nro. 4.3, presenta el segundo refinamiento de la cadena de búsqueda aplicada sobre la base de datos IEEE Explore. En esta se hizo uso de los

operadores “OR” y “AND”, para buscar por cadenas exactas con el uso de las comillas en el resumen y título de los documentos; con lo que, se obtuvo un número importante de artículos relevantes para este estudio. Los artículos resultantes fueron 277; los cuáles serán tomados en cuenta para el proceso de selección que se explica en la Sección nro. 4.2.2.

Cadena de búsqueda	Resultado	Descripción
<pre>(("Abstract":.QT.Open Source Software.QT.) OR ("Abstract":.QT.OSS.QT.) OR ("Abstract":.QT.FOSS.QT.)) AND ("Abstract":risk OR ("Abstract":.QT.failure.QT.) OR ("Abstract":.success) OR ("Abstract":.barriers) OR ("Abstract":.Advantage) OR ("Abstract":.Impediments) OR ("Abstract":.FLOSS)) AND "Document Title":.QT.Open Source Software.QT OR "Document Title":.QT.OSS.QT. OR "Document Title":.QT.FOSS.QT OR "Document Title":.QT.FLOSS.QT</pre>	277	Se añadieron a la búsqueda palabras claves y sinónimos en resumen y título

Tabla 4.3 Refinamiento 2, cadena de búsqueda IEEE (Autoría propia, 2017)

4.2.1.2. Biblioteca ACM DIGITAL LIBRARY

Al igual que la biblioteca IEEE Xplore, la Biblioteca digital ACM permite realizar búsquedas avanzadas. La Figura nro. 4.3, presenta un ejemplo de búsqueda avanzada en ACM.

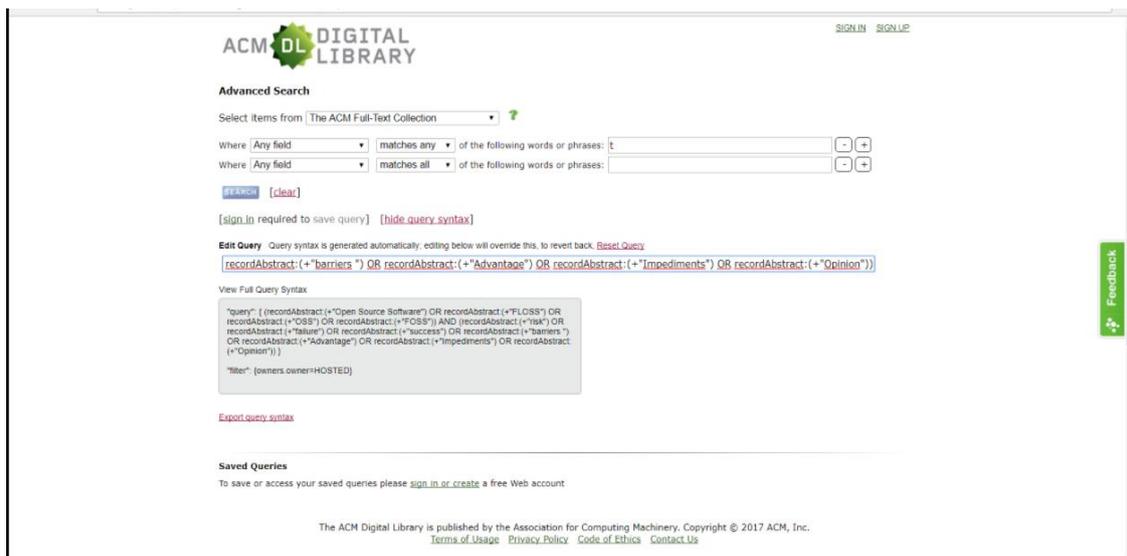


Figura 4.3 Búsqueda avanzada en ACM Digital Library (ACM Digital Library, 2017)

En la Figura nro. 4.4, se puede apreciar los pasos que se han seguido para la identificación de publicaciones relevantes en la biblioteca ACM Digital Library.

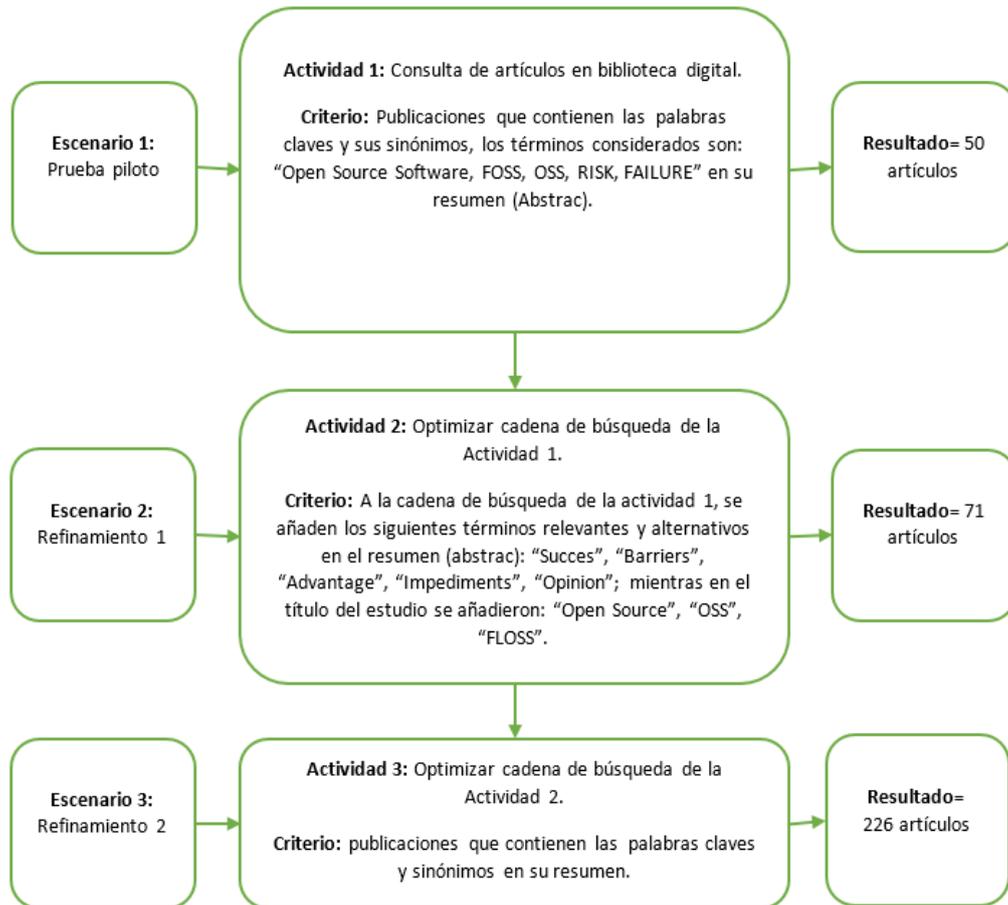


Figura 4.4 Pasos para identificación de publicaciones en biblioteca ACM Digital Library (Autoría propia, 2017)

A continuación, se detallan las cadenas de búsqueda, que se han empleado en ACM Digital Library, las cuales estuvieron construidas con los términos relevantes como “palabras claves”, y se han vinculados con los operadores booleanos “AND”, “OR” y los campos de datos como: resumen del artículo (recordAbstract), título del documento (acmdlTitle). Dichos operadores y campos utilizados han permitido ir refinando cada vez más la búsqueda. Como se puede observar más adelante, los campos de datos varían según los siguientes escenarios: prueba piloto, refinamiento 1 y refinamiento 2.

4.2.1.2.1. Prueba piloto

La primera cadena de búsqueda fue realizada con la utilización de las palabras claves en el resumen (Abstract) de los artículos. Las palabras claves y alternativas

consideradas para esta etapa son: “Open Source Software”, “OSS”, “FOSS”, “risk”, “failure”. Los resultados se presentan en la Tabla nro. 4.4.

Cadena de búsqueda	Resultado	Descripción
(recordAbstract:(+"Open Source Software") OR recordAbstract:(+"OSS") OR recordAbstract:(+"FOSS")) AND (recordAbstract:(+"risk") OR recordAbstract:(+"failure"))	50	Publicaciones que contengan las palabras claves y alternativas en el resumen

Tabla 4.4 Primera cadena de búsqueda ACM Digital Library (Autoría propia, 2017)

4.2.1.2.2. Refinamiento 1

En esta etapa, se menciona como fue optimizada la cadena de búsqueda definida en la fase anterior “prueba piloto”. La cadena de búsqueda que se describió en la Tabla nro. 4.4, contiene términos relevantes que se utilizaron anteriormente, también se ha agregado las palabras: “success”, “barriers”, “Advantage”, “Impediments”, “Opinion”. Al menos uno de estos términos debe estar incluido en el resumen de los artículos. Es importante recalcar que, para esta cadena de búsqueda, se tomó en cuenta que el título de los estudios contenga cualquiera de los siguientes términos: “Open Source Software”, “OSS”, “FLOSS”. Los resultados se presentan en la Tabla nro. 4.5.

Cadena de búsqueda	Resultado	Descripción
(acmdlTitle:(+"Open Source Software") OR acmdlTitle:(+"OSS") OR acmdlTitle:(+"FLOSS") OR acmdlTitle:(+"FLOSS")) AND (recordAbstract:(+"Open Source Software") OR recordAbstract:(+"OSS") OR recordAbstract:(+"FOSS")) AND (recordAbstract:(+"risk") OR recordAbstract:(+"failure") OR recordAbstract:(+"success") OR recordAbstract:(+"barriers ") OR recordAbstract:(+"Advantage") OR recordAbstract:(+"Impediments") OR recordAbstract:(+"Opinion"))	71	Publicaciones que contienen las palabras claves y sus sinónimos en el resumen y título del documento

Tabla 4.5 Refinamiento 1 cadena búsqueda ACM Digital Library (Autoría propia, 2017)

4.2.1.2.3. Refinamiento 2

A partir del primer refinamiento, y con el afán de obtener un número significativo de artículos para llevar a cabo el estudio, se optó por realizar un nuevo refinamiento a la cadena de búsqueda anterior, por esta razón se tomó en cuenta algunos términos relevantes y alternativos que deberán estar presentes de alguna manera en el resumen del

documento. Dichos términos antes mencionados son: “Open Source Software”, “FOSS”, “OSS”, “FLOSS” juntamente con cualquiera de las siguientes cadenas: “risk”, "failure", “success”, "barriers “, "Advantage”, “Impediments”, “Opinion”.

La Tabla nro. 4.6, presenta la cadena definitiva resultante de los dos refinamientos, en esta se han usado los operadores “OR” y “AND”; que ha permitido buscar, por cadenas exactas, usando las comillas, ya sea en el título o en el resumen de los documentos, lo que permitió identificar un total de 226 artículos relevantes para el estudio, los cuales serán utilizados como entrada al proceso de selección que se explica en la Sección nro. 4.2.2.

Cadena de búsqueda	Resultado	Descripción
(recordAbstract:(+"Open Source Software") OR recordAbstract:(+"FLOSS") OR recordAbstract:(+"OSS") OR recordAbstract:(+"FOSS")) AND (recordAbstract:(+"risk") OR recordAbstract:(+"failure") OR recordAbstract:(+"success") OR recordAbstract:(+"barriers ") OR recordAbstract:(+"Advantage") OR recordAbstract:(+"Impediments") OR recordAbstract:(+"Opinion"))	226	Publicaciones que contienen las palabras claves y sinónimos en su resumen.

Tabla 4.6 Refinamiento 2 cadena búsqueda ACM Digital Library (Autoría propia, 2017)

4.2.1.3. Biblioteca Springer Link

Al igual que la biblioteca IEEE Xplore y ACM, la Biblioteca digital Springer Link permite realizar búsquedas avanzadas. La Figura nro. 4.5, presenta un ejemplo de búsqueda avanzada en Springer Link.



Figura 4.5 Búsqueda avanzada en Springer link (Springer Link, 2017)

En la Figura nro. 4.6, se puede apreciar los pasos que se han seguido para la identificación de publicaciones relevantes en la biblioteca Springer Link.

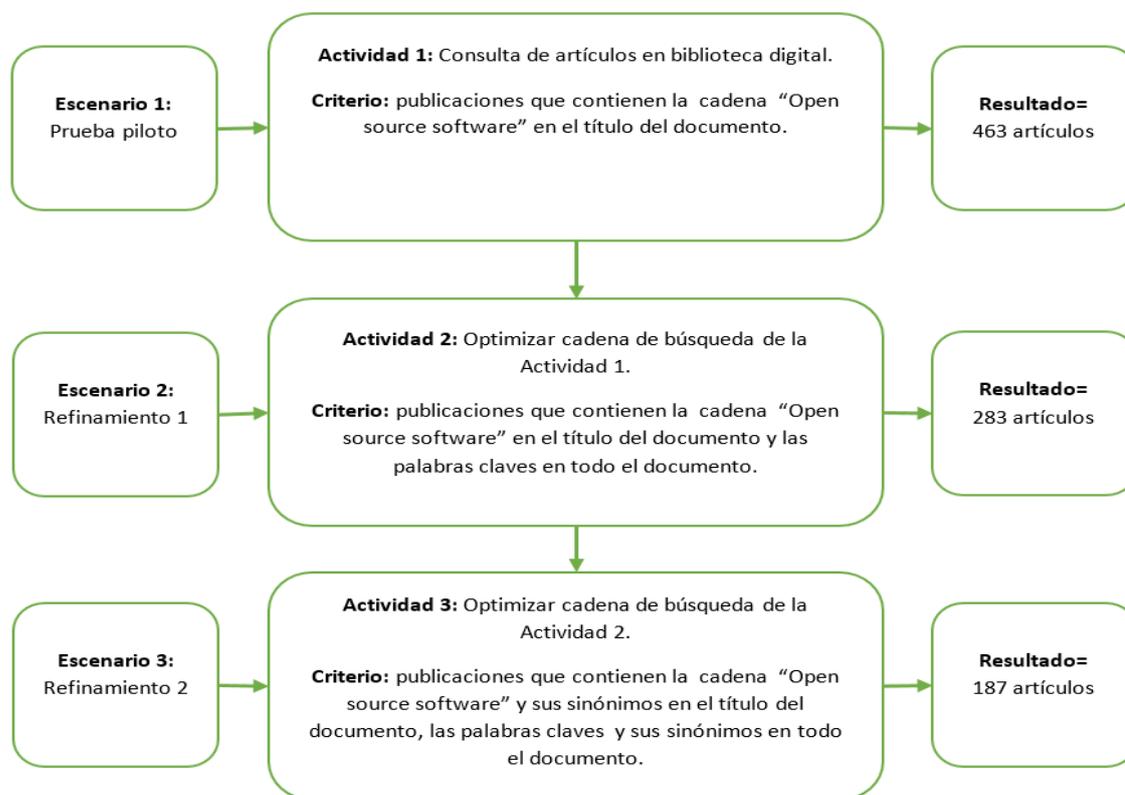


Figura 4.6 Pasos para identificación de publicaciones en biblioteca Springer Link (Autoría propia, 2017)

A continuación, se describen las cadenas que fueron empleadas en Springer Link, las cuales estuvieron construidas con los términos relevantes como “palabras claves”, que fueron vinculados a los operadores booleanos “AND” para incluir los artículos que contengan todas las palabras introducidas y “OR” para incorporar documentos que contengan al menos una de las palabras introducidas en la búsqueda. Al hacer referencia al Capítulo nro.2, Sección nro. 2.4.2; en esta librería las búsquedas avanzadas se las puede hacer en todo el texto o en los campos; sin embargo, se ha visto limitada para realizar una búsqueda en el resumen de los documentos, ya que no es soportado por esta biblioteca. Más adelante se describe la cadena de búsqueda obtenida en los siguientes escenarios: prueba piloto, refinamiento 1 y refinamiento 2.

Para armar las cadenas se ha utilizado el asistente de búsqueda avanzada de Springer Link que se presenta en la Figura nro. 4.7

Advanced Search

The screenshot shows the 'Advanced Search' interface with the following elements:

- Find Resources** section:

 - with all of the words**: AND
 - with the exact phrase**: FRASE
 - with at least one of the words**: OR (Example: risk failure success barriers Advantage Impediments Opinion "Open Source Softw.
 - without the words**: NOT
 - where the title contains**: CAMPO (Example: "open source software", e.g. "Cassini at Saturn" or Saturn)
 - where the author / editor is**: CAMPO (Example: "H.G. Kennedy" or Elvis Morrison)
 - Show documents published**: FECHA (between [] and [])
 - Include Preview-Only content**: ACCESO (checked)

- Search** button at the bottom.

Figura 4.7 Asistente de búsquedas avanzadas en Springer Link (Springer Link, 2017)

4.2.1.3.1. Prueba piloto

En la Tabla nro. 4.7, se presenta la primera cadena de búsqueda para encontrar artículos relevantes, en donde el título de estos documentos contenga la palabra clave: “Open Source software”.

Cadena de búsqueda	Resultado	Descripción
Título del documento; “Open Source Software”	463	Publicaciones que contengan “Open Source Software” en el título de documento

Tabla 4.7 Primera cadena de búsqueda Springer Link (Autoría propia, 2017)

4.2.1.3.2. Refinamiento 1

Luego de haber establecido la primera cadena de búsqueda definida en la fase anterior “prueba piloto”, se ha procedido a optimizar la misma, a más de buscar en el título de los artículos la palabra clave: “Open Source Software”, se han incluido los documentos donde esté presente cualquiera de los términos relevantes y alternativos como son: “risk”, “failure”, “success” “barriers”. La Tabla nro. 4.8, presenta la cadena de búsqueda generada.

Cadena de búsqueda	Resultado	Descripción
Título del documento; “Open Source Software” Todas las palabras: risk or failure or success or barriers	283	Publicaciones que contengan “Open Source Software” en el título de documento y las palabras claves en todo el documento.

Tabla 4.8 Refinamiento 1 cadena búsqueda Springer Link (Autoría propia, 2017)

4.2.1.3.3. Refinamiento 2

Luego de aplicar el refinamiento 1, se incluyó a la cadena de búsqueda de la etapa anterior más términos relevantes, como son, por ejemplo, los sinónimos, que deberán estar presentes en el título y en el texto del documento, estos términos fueron los siguientes: “Open source software”, “OSS”, “FLOSS”, “FOSS”, “risk”, “failure”, “success”, “barriers”, “Advantage”, “Impediments”, “Opinion”.

En La Tabla nro. 4.9, se presenta la cadena de búsqueda final, utilizada para la exploración de artículos que aporten con la investigación.

De igual forma, tuvimos la asesoría del tutor experto “Ing. Juan Pablo Carvallo. PhD.” que ha sido fundamental para este proceso. Es importante mencionar la dificultad en Springer Link para buscar en el resumen de los documentos, ya que esta biblioteca en su búsqueda avanzada solo ha permitido utilizar operadores booleanos como: “OR” y “AND”. También se ha podido realizar la búsqueda por título del documento, e introducir palabras y frases exactas, las cuales deberán contener el texto de los documentos. Obteniendo de esta manera una gran variedad de artículos resultantes, que fueron 187 artículos, que se consideran para la investigación.

Cadena de búsqueda	Resultado	Descripción
("Open source software" or "OSS" or "FLOSS" or "FOSS") and (risk OR failure OR success OR barriers OR Advantage OR Impediments OR Opinion* or FOSS or FLOSS or "Open Source Software" or OSS) and dc.title("Open Source Software" or "OSS" or "FLOSS" or "FOSS")	187	Publicaciones que contengan "Open Source Software" y sus sinónimos en el título de documento, las palabras claves y sus sinónimos en todo el documento.

Tabla 4.9 Refinamiento 2 cadena búsqueda Springer Link (Autoría propia, 2017)

Producto de la búsqueda de las tres bibliotecas digitales seleccionadas, se obtuvo como resultado un total de 690 artículos posibles, que pueden aportar con la investigación, de los cuales se ha importado los metadatos (título, año de publicación, autores, DOI y palabras claves) y el resumen. En este proceso no se encontró artículos redundantes.

4.2.2. Selección de estudios

Una vez culminada la identificación de los estudios potencialmente relevantes, en la fase de identificación antes mencionada, se ha procedido a descargar la información de título, autores, resumen, entre otros, esta información se almacenó en una tabla de Excel como se presenta en la Figura nro. 4.8, esto se realizó de los 690 artículos resultantes. Luego, en base a lo planificado en el protocolo que se menciona en el Capítulo nro. 3, se ha tomado en cuenta los criterios y el procedimiento de selección de estudios que se han mencionado en la Sección nro. 3.6 y nro. 3.7. Al no encontrar artículos duplicados, de los 690 artículos resultantes se continuó con el proceso de selección de estudios para identificar su importancia real. El proceso de selección está contemplado en dos etapas: preselección y selección final de los estudios primarios.

4.2.2.1. Preselección de estudios primarios

En primera instancia de los 690 documentos identificados, se ha procedido a hacer una primera lectura, basándose en el título y el resumen de cada artículo, actividad que fue realizada por un primer grupo de investigadores, el proceso fue llevado a cabo con una repartición del 50% de los artículos. A partir de ello, se calificó en un campo denominado "Preselección 1", que se agregó a la tabla donde fueron importados los metadatos y resúmenes de los documentos seleccionados como se puede apreciar en las

Figuras nro. 4.8, 4.9, y 4.10 respectivamente. En el campo que se añadió, cada artículo fue valorado como: “Si”, “No” y “Tal vez”.

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1
Shared Mental Models among Open Source Software Developers	2008	B. Scozzi and K. Crowston and U. V. Eseryel and Q. Li	10.1109/HICSS.2008.391	public domain software,software development management,Free/Libre Open Source Software development team,cognitive mapping,open source software developers,process analysis,shared mental model,software development process,Buildings,Cognitive science,Data analysis,Distributed computing,Human factors,Java,Open source software,Programming,Shape,Sustainable development	Shared understandings are important for software development as they guide effective individual contributions and coordination of the software development process. However, it is not clear if such understandings can be developed in highly distributed groups that do not regularly meet face-to-face. In this paper, we present the results of a preliminary analysis of shared mental models within a Free/Libre Open Source Software (FLOSS) development team. We analyzed mental models using cognitive mapping and process analysis and compared the models of four developers from the Apache Lucene Java project. Our analysis suggests that there is a high level of sharing among core developers but the sharing is not complete, with some differences related to tenure and role in the project. Finally, we suggest directions for further research on shared mental models in FLOSS teams.	No
An Exploratory Study on the Evolution of OSS Developer Communities	2008	K. Ngamkajornwiwat and D. Zhang and A. G. Koru and L. Zhou and a. R. Nolkar	10.1109/HICSS.2008.58	public domain software,resource allocation,software development management,software quality,OSS development,open source software,resource distribution,resource management,social networks,software development,software evolutionary patterns,software quality,Collaborative software,Complex networks,information systems,International collaboration,Open source software,Programming,Project management,Quality management,Social network services,Software systems	Software is developed in a dynamic context where team structure, requirements, and processes evolve together with the product. Although researchers have been studying the evolution of software systems since the early 70s, the study on the evolution of software development teams remains rare. Such evolutionary patterns and their impact on software quality are especially important in the context of OSS (Open Source Software) development, where a group of volunteer developers collaborate online for an extended period of time. By analyzing how social networks of developers evolve over time while building OSS products, we can gain knowledge and experience to improve the effectiveness and efficiency of resource management and distribution in future OSS projects. To this end, we studied the evolution of the developer communities using a suite of OSS products developed under the Koffice project. We found that in general, the social networks of OSS developer communities change over time in certain ways. Identifying such patterns can help OSS managers better understand the unique process of OSS development and improve their management and coordination of the projects.	Si
Free and Open Source Software Quality Assurance	2006	A. Tawileh and O. Rana	10.1109/ICTA.2006.1684867	public domain software,software quality,free software quality assurance,open source software quality assurance,Collaborative software,Computer industry,Computer science,Cultural differences,Global communication,Open source software,Protection,Quality assurance,Software quality,Unified modeling language	During the past two decades, the free and open source software community has been developing high quality software that proved to be a serious rival to proprietary software produced by commercial companies. However, in spite of its considerable success within the developers' community, F/OSS did not attract many enthusiasts from outside this community. Claims against the adoption of F/OSS are usually rooted in the apparent lack of quality assurance mechanisms required to guarantee the protection of users' interests. The objective of this study is to promote higher adoption of F/OSS artefacts outside the developers' community by exploring possibilities to provide appropriate assurances that F/OSS artefacts will meet the quality levels expected by users.	Tal vez
Network Management Knowledge Base - A Step towards OSS Intelligence	2007	R. Lovric and M. Dragan and M. Kunic	10.1109/CONTEL.2007.381863	database management systems,knowledge based systems,knowledge management,telecommunication computing,telecommunication network management,OSS intelligence,attributive language,knowledge management,network management,knowledge base,operations support system,telecommunication network management,databases,Automation,Conference management,Expert systems,Intelligent networks,Inventory management,Knowledge management,Quality of service,Resource management,Solids,Telecommunication network management	Even though introduction of OSS in daily telecom activities represents a significant improvement by itself, it still leaves an extra space for further improvements. Knowledge bases proved to be an effective way to fill that space, and even more, they represent a solid base for an expert system which can raise a system that implements such activities to a far more sophisticated level giving the latter a new dimension of automation. This paper describes basic concepts of telecommunication network management databases derived knowledge base, along with concepts of attributive language used as a method for describing knowledge base and possibility for introducing higher level improvements such as expert system.	No

Figura 4.8 Preselección de artículos obtenidos de IEEE Xplore en primera instancia (Autoría propia, 2017)

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1
Learning Through Open Source Participation	2011	Heidi J.C. Ellis and Mel Chua and Matthew C. Jadud and Gregory W. Hislop	10.1145/1953163.1953191	education, faculty development, open source software	Free and Open Source Software (FOSS) and documentation projects provide excellent learning opportunities for students. In the context of active learning, FOSS is particularly interesting in providing transparent meritocracies that allow students to observe and contribute as part of their learning. This panel will present four different perspectives on student involvement in Free and Open Source Software (FOSS) projects. These perspectives will cover: (a) different ways that students can contribute to FOSS projects beyond coding, (b) an industry perspective on student involvement in FOSS projects, (c) how Humanitarian FOSS can provide a welcoming environment for student learning, and (d) barriers to faculty involvement and how such barriers can be overcome.	No
An Open Source Field Trip for Faculty: Pre-conference Workshop	2011	Gregory W. Hislop and Cliff Kusmaul and Mel Chua and Sebastian Dziallas	No tiene	No contiene	Student participation in Free and Open Source Software (FOSS) projects and communities offers excellent opportunities for learning and for developing student interest in computing. This workshop will be a virtual field trip for faculty wanting to know more about FOSS communities, practices, and tools so they can take advantage of this opportunity. The workshop will provide an overview of FOSS from a development perspective, hands-on activities to introduce FOSS tools, and examples of student participation in FOSS projects. Participants will also be introduced to a variety of resources that support faculty and student participation in FOSS communities. The intended audience is computing educators at the college or high school level.	No
Foss 101: Engaging Introductory Students in the Open Source Movement	2009	Ralph Morelli and Trishan de Lanerolle	10.1145/1539024.1508977	curriculum development, open source movement, open source software	Can engaging students in free and open source software (FOSS) pique their interest in computer science? This paper describes an introductory computer science course that introduced students to using FOSS, contributing to a humanitarian FOSS project, and to studying the broader impact of FOSS on our society. Students learned basic webprogramming skills (PHP/MySQL) and made small but significant contributions to a global FOSS project. Mistakes were made and opportunities were missed. But overall the experiment was a success and the experience was enjoyable and educational for students and instructor alike. By building on what worked well, this course could serve as a model for incorporating study of FOSS into the introductory computing curriculum.	No
Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of sourceforge.Net Projects	2007	Robert English and Charles M. Schweik	10.1109/FLOSS.2007.9	No contiene	Free/Libre and Open Source Software (FLOSS) projects are a form of commons where individuals work collectively to produce software that is a public, rather than a private, good. The famous phrase "Tragedy of the Commons" describes a situation where a natural resource commons, such as a pasture, or a water supply, gets depleted because of overuse. The tragedy in FLOSS commons is distinctly different -- it occurs when collective action ceases before a software product is produced or reaches its full potential. This paper builds on previous work about defining success in FLOSS projects by taking a collective action perspective. We first report the results of interviews with FLOSS developers regarding our ideas about success and failure in FLOSS projects. Building on those interviews and previous work, we then describe our criteria for defining success/tragedy in FLOSS commons. Finally, we discuss the results of a preliminary classification of nearly all projects hosted on Sourceforge.net as of August 2006.	No
Open Source Software Development: Expectations and Experience from a Small Development Project	2008	Adanna Ezeala and Hyunji Kim and Loretta A. Moore	10.1145/1593105.1593168	open source software, software development process	Open Source Software (OSS) is software that provides access to its source code in order to allow users to improve and redistribute the software. The emergence of OSS has introduced new ways to develop, test, and maintain software. While several success stories about large-scale OSS projects are reported, little research has been done on how small-scale OSS projects are managed by OSS developers. This paper reports preliminary findings from our experience with a small OSS project, while discussing differences between software processes in large- and small-scale OSS developments.	Tal vez

Figura 4.9 Preselección de artículos obtenidos de ACM Digital Library en primera instancia (Autoría propia, 2017)

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1
Lessons Learned from Teaching Open Source Software Development	2014	Becca Morgan Carlos Jensen	10.1007/978-3-642-55128-4_18	No dispone	Free/Open Source Software allows students to learn valuable real world skills and experiences, as well as to create a portfolio to show future employers. However, the learning curve to joining FOSS can be daunting, often leading newcomers to walk away frustrated. Universities therefore need to find ways to provide a structured introduction to students, helping them overcome the barriers to entry. This paper describes two courses taught at two universities, built around a Communities of Practice model, and the lessons learned from these. Suggestions and insights are shared for how to structure and evaluate such courses for maximum effect.	Tal vez
Comparing Assessment Methodologies for Free/Open Source Software: OpenBIR and QOS	2008	Jean-Christophe Deprez Simon Alexandre	10.1007/978-3-540-69566-0_17	No dispone	Many organisations using Free/Open Source Software (FOSS) are dealing with the major problem of selecting the most appropriate software product corresponding to their needs. Most of these companies are currently selecting FOSS projects using ad-hoc techniques. However, in the last couple of years, two methodologies for assessing FOSS project have emerged, namely QOS and OpenBIR. The objective of this work is, through a detailed and rigorous assessment methodology comparison, to allow companies to have a better understanding of these two assessment methodologies content and limitation. This work compares both methodologies on several aspects, among others, their overall approaches, their scoring procedures and their evaluation criteria.	Si
Rethinking free, libre and open source software	2006	Ruben van Wendel de Jode Yusef Lindhay David Ph.D.candidate	10.1007/112130-006-1001-y	No dispone	This special issue includes seven articles that make significant contribution to the literature pertaining to knowledge and public policy around Free, Libre and Open Source Software (FLOSS). Focusing on questions in two themes (i) motivation and organisation and (ii) public policy, the articles in this volume develop new analytic models and report on new empirical findings, as an important step in bridging the wide gap that exists in public policy literature around FLOSS. Warning against rhetorical pitfalls that have been prevalent in FLOSS research, this introduction starts with a short history of FLOSS development, continues with a brief thematic literature review and review of the misconceptions surrounding FLOSS, and concludes with a first introduction of the articles that follow. He is part of the Dutch Institute of Government (NiG), the research school for public administration and political science. His research focuses on the organisation of open source communities. He received two grants from the Netherlands Organization for Scientific Research (NWO) for research related to open source communities. The first grant was to study the interplay between intellectual property rights and open source communities. The results are published in <i>Governing the Virtual Commons</i> (Cambridge University Press, 2003). He has written numerous articles on open source, which have appeared in journals like <i>Electronic Markets</i> , <i>Knowledge, Technology and Policy</i> , and the <i>International Journal of IT Standards & Standardisation Research</i> . She received her Ph.D. in sociology from the University of York (UK) in 2004. Her Ph.D. research investigated the heterogeneity and contingency in the Free, Libre Open Source Software (FLOSS) social worlds, which is based on a constellation of facing networks, from the sociological perspective. Her principal research interests center on FLOSS studies, Science and Technology Studies (STS), virtual communities and knowledge-sharing. Shay is also a fellow at The Information Society Project at the Yale Law School. Shay holds a B.Sc. in computer science and a B.A. in philosophy, magna cum laude, from Tel-Aviv University, and an M.A. from New York University where his interdisciplinary research thesis focused on the political economy of free and open source software and file sharing networks. Shay is an entrepreneur that co-founded two software start-up companies, and was involved for several years in cutting edge software research, combining open source and proprietary software.	Si

Figura 4.10 Preselección de artículos obtenidos de Springer Link en primera instancia (Autoría propia, 2017)

Luego de ello un segundo grupo de investigadores calificados revisaron la calificación de los artículos seleccionados, por el primer grupo de investigadores, para ello se realizó una segunda lectura del título y resumen de los artículos calificando a cada uno como: “Si”, “No” y “Tal vez”. Como se puede apreciar en las Figuras nro. 4.11, 4.12 y 4.13, respectivamente, la tabla completa será anexada a las evidencias del presente trabajo.

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1	Preselección 2
Shared Mental Models among Open Source Software Developers	2008	B. Scozzi and K. Crowston and U. Y. Eseryel and Q. Li	10.1109/HICSS.2008.391	public domain software,software development management,Free/Libre Open Source Software development team,cognitive mapping,open source software developers,process analysis,shared mental model,software development process,Buildings,Cognitive science,Data analysis,Distributed computing,Human factors,Java,Open source software,Programming,Shape,Sustainable development	Shared understandings are important for software development as they guide effective individual contributions and coordination of the software development process. However, it is not clear if such understandings can be developed in highly distributed groups that do not regularly meet face-to-face. In this paper, we present the results of a preliminary analysis of shared mental models within a Free/Libre Open Source Software (FLOSS) development team. We analyzed mental models using cognitive mapping and process analysis and compared the models of four developers from the Apache Lucene Java project. Our analysis suggests that there is a high level of sharing among core developers but the sharing is not complete, with some differences related to tenure and role in the project. Finally, we suggest directions for further research on shared mental models in FLOSS teams.	No	NO
An Exploratory Study on the Evolution of OSS Developer Communities	2008	K. Ngamkajornwatt and D. Zhong and A. G. Kuru and L. Zhou and a. R. Nolkar	10.1109/HICSS.2008.58	public domain software,resource allocation,software development management,software quality,OSS development,open source software,resource distribution,resource management,social networks,software development,software evolutionary patterns,software quality,Collaborative software,Complex networks,Information systems,international collaboration,Open source software,Programming,Project management,Quality management,Social network services,Software systems	Software is developed in a dynamic context where team structure, requirements, and processes evolve together with the product. Although researchers have been studying the evolution of software systems since the early 70s, the study on the evolution of software development teams remains rare. Such evolutionary patterns and their impact on software quality are especially important in the context of OSS (Open Source Software) development, where a group of volunteer developers collaborate online for an extended period of time. By analyzing how social networks of developers evolve over time while building OSS products, we can gain knowledge and experience to improve the effectiveness and efficiency of resource management and distribution in future OSS projects. To this end, we studied the evolution of the developer communities using a suite of OSS products developed under the KOffice project. We found that in general, the social networks of OSS developer communities change over time in certain ways. Identifying such patterns can help OSS managers better understand the unique process of OSS development and improve their management and coordination of the projects.	Si	Si
Free and Open Source Software Quality Assurance	2006	A. Tawileh and O. Rana	10.1109/ICTA.2006.1684867	public domain software,software quality,free software quality assurance,open source software quality assurance,Collaborative software,Computer industry,Computer science,Cultural differences,Global communication,Open source software,Protection,Quality assurance,Software quality,Unified modeling language	During the past two decades, the free and open source software community has been developing high quality software that proved to be a serious rival to proprietary software produced by commercial companies. However, in spite of its considerable success within the developers' community, F/OSS did not attract many enthusiasts from outside this community. Claims against the adoption of F/OSS are usually rooted in the apparent lack of quality assurance mechanisms required to guarantee the protection of users' interests. The objective of this study is to promote higher adoption of F/OSS artefacts outside the developers' community by exploring possibilities to provide appropriate assurances that F/OSS artefacts will meet the quality levels expected by users.	Tal vez	Si
Network Management Knowledge Base - A Step towards OSS Intelligence	2007	R. Lovric and N. Dragan and M. Kunstic	10.1109/CONTEL.2007.3818663	database management systems,Knowledge based systems,Knowledge management,telecommunication computing,telecommunication network management,OSS intelligence,attributive language,Knowledge management,network management,Knowledge base,operations support systems,international collaboration,Open source databases,Automation,Conference management,Expert systems,Intelligent networks,Inventory management,Knowledge management,Quality of service,Resource management,Solids,Telecommunication network management	Even though introduction of OSS in daily telecom activities represents a significant improvement by itself, it still leaves an extra space for further improvements. Knowledge bases proved to be an effective way to fill that space, and even more, they represent a solid base for an expert system which can raise a system that implements such activities to a far more sophisticated level giving the latter a new dimension of automation. This paper describes basic concepts of telecommunication network management databases derived knowledge base, along with concepts of attributive language used as a method for describing knowledge base and possibility for introducing higher level improvements such as expert system.	No	NO

Figura 4.11 Preselección de artículos obtenidos de IEEE Xplore en segunda instancia (Autoría propia, 2017)

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1	Preselección 2
Learning Through Open Source Participation	2011	Heidi J.C. Ellis and Mel Chua and Matthew C. Jajud and Gregory W. Hislop	10.1145/1953163.1953191	education, faculty development, open source software	Free and Open Source Software (FOSS) and documentation projects provide excellent learning opportunities for students. In the context of active learning, FOSS is particularly interesting in providing transparent meritocracies that allow students to observe and contribute as part of their learning. This panel will present four different perspectives on student involvement in Free and Open Source Software (FOSS) projects. These perspectives will cover: (a) different ways that students can contribute to FOSS projects beyond coding, (b) an industry perspective on student involvement in FOSS projects, (c) how Humanitarian FOSS can provide a welcoming environment for student learning, and (d) barriers to faculty involvement and how such barriers can be overcome.	No	No
An Open Source Field Trip for Faculty: Pre-conference Workshop	2011	Gregory W. Hislop and Cliff Kusumaul and Mel Chua and Sebastian Diazlas	No tiene	No contiene	Student participation in Free and Open Source Software (FOSS) projects and communities offers excellent opportunities for learning and for developing student interest in computing. This workshop will be a virtual field trip for faculty wanting to know more about FOSS communities, practices, and tools so they can take advantage of this opportunity. The workshop will provide an overview of FOSS from a development perspective, hands-on activities to introduce FOSS tools, and examples of student participation in FOSS projects. Participants will also be introduced to a variety of resources that support faculty and student participation in FOSS communities. The intended audience is computing educators at the college or high school level.	No	No
Foss 101: Engaging introductory Students in the Open Source Movement	2009	Ralph Morelli and Trishan de Lanerolle	10.1145/1598024.1598977	curriculum development, open source movement, open source software	Can engaging students in free and open source software (FOSS) pique their interest in computer science? This paper describes an introductory computer science course that introduced students to using FOSS, contributing to a humanitarian FOSS project, and studying the broader impact of FOSS on our society. Students learned basic webprogramming skills (PHP/MySQL) and made small but significant contributions to a global FOSS project. Mistakes were made and opportunities were missed. But overall the experiment was a success and the experience was enjoyable and educational for students and instructor alike. By building on what worked well, this course could serve as a model for incorporating study of FOSS into the introductory computing curriculum.	No	No
Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of Sourceforge.Net Projects	2007	Robert English and Charles M. Schweik	10.1109/FLOSS.2007.9	No contiene	Free/Libre and Open Source Software (FLOSS) projects are a form of commons where individuals work collectively to produce software that is a public, rather than a private, good. The famous phrase "Tragedy of the Commons" describes a situation where a natural resource commons, such as a pasture, or a water supply, gets depleted because of overuse. The tragedy in FLOSS commons is distinctly different – it occurs when collective action ceases before a software product is produced or reaches its full potential. This paper builds on previous work about defining success in FLOSS projects by taking a collective action perspective. We first report the results of interviews with FLOSS developers regarding our ideas about success and failure in FLOSS projects. Building on those interviews and previous work, we then describe our criteria for defining success/tragedy in FLOSS commons. Finally, we discuss the results of a preliminary classification of nearly all projects hosted on Sourceforge.net as of August 2006.	No	Si
Open Source Software Development: Expectations and Experience from a Small Development Project	2008	Adanna Ezeala and Hyunju Kim and Loretta A. Moore	10.1145/1593105.1593168	open source software, software development process	Open Source Software (OSS) is software that provides access to its source code in order to allow users to improve and redistribute the software. The emergence of OSS has introduced new ways to develop, test, and maintain software. While several success stories about large-scale OSS projects are reported, little research has been done on how small-scale OSS projects are managed by OSS developers. This paper reports preliminary findings from our experience with a small OSS project, while discussing differences between software processes in large- and small-scale OSS developments.	Tal vez	No

Figura 4.12 Preselección de artículos obtenidos de ACM Digital Library en segunda instancia (Autoría propia, 2017)

Título	Año de publicación	Autores	DOI	Palabras claves	Resumen	Preselección 1	Preselección 2
Lessons Learned from Teaching Open Source Software Development	2014	Becka Morgan/Carlos Jensen	10.1007/978-3-642-55128-4_18	No dispone	Free/Open Source Software allows students to learn valuable real world skills and experiences, as well as to create a portfolio to show future employers. However, the learning curve to joining FOSS can be daunting, often leading newcomers to walk away frustrated. Universities therefore need to find ways to provide a structured introduction to students, helping them overcome the barriers to entry. This paper describes two courses taught at two universities, built around a Communities of Practice model, and the lessons learned from these. Suggestions and insights are shared for how to structure and evaluate such courses for maximum effect.	Tal vez	Si
Comparing Assessment Methodologies for Free/Open Source Software: OpenBTR and OSOS	2008	Jean-Christophe Despre/Simon Alexandre	10.1007/978-3-540-69566-0_17	No dispone	Many organizations using Free/Open Source Software (FOSS) are dealing with the major problem of selecting the most appropriate software product corresponding to their needs. Most of these companies are currently selecting FOSS projects using ad-hoc techniques. However, in the last couple of years, two methodologies for assessing FOSS project have emerged, namely OSOS and OpenBTR. The objective of this work is, through a detailed and rigorous assessment methodology comparison, to allow companies to have a better understanding of these two assessment methodologies content and limitation. This work compares both methodologies on several aspects, among others, their overall approaches, their scoring procedures and their evaluation criteria.	Si	Si
Rethinking free, libre and open source software	2006	Ruben van Wendel de Jood/Luwei Lin/Shay David Ph.D candidate	10.1007/s11210-006-1001-y	No dispone	This special issue includes seven articles that make significant contribution to the literature pertaining to knowledge and public policy around Free, Libre and Open Source Software (FLOSS). Focusing on questions in two themes (i) motivation and organization and (ii) public policy, the articles in this volume develop new analytic models and report on new empirical findings, as an important step in bridging the wide gap that exists in public policy literature around FLOSS. Warning against rhetorical pitfalls that have been prevalent in FLOSS research, this introduction starts with a short history of FLOSS development, continues with a brief thematic literature review and review of the misconceptions surrounding FLOSS, and concludes with a first introduction of the articles that follow. He is part of the Dutch Institute of Government (NIG), the research school for public administration and political science. His research focuses on the organization of open source communities. He received two grants from the Netherlands Organization for Scientific Research (NWO) for research related to open source communities. The first grant was to study the interplay between intellectual property rights and open source communities. The results are published in <i>Governing the Virtual Commons</i> (Cambridge University Press, 2003). He has written numerous articles on open source, which have appeared in journals like <i>Electronic Markets</i> , <i>Knowledge, Technology and Policy</i> , and the <i>International Journal of IT Standards & Standardisation Research</i> . She received her Ph.D. in sociology from the University of York (UK) in 2004. Her Ph.D. research investigated the heterogeneity and contingency in the Free, Libre Open Source Software (FLOSS) social worlds, which is based on a constellation of hacking practices, from the sociological perspective. Her principal research interests center on FLOSS studies, Science and Technology Studies (STS), virtual communities and knowledge-sharing. Shay is also a fellow at the Information Society Project at the Yale Law School. Shay holds a B.Sc. in computer science and a B.A. in philosophy, magna cum laude, from Tel-Aviv University, and an M.A. from New York University where his interdisciplinary research thesis focused on the political economy of free and open source software and file sharing networks. Shay is an entrepreneur that co-founded two software start-up companies, and was involved for several years in cutting edge software research, combining open source and proprietary software.	Si	Si

Figura 4.13 Preselección de artículos obtenidos de Springer Link en segunda instancia (Autoría propia, 2017)

Este proceso, permitió identificar 165 artículos como potencialmente relevantes para el estudio.

4.2.2.2. Selección final de estudios primarios

Una vez identificados los artículos significativos en el proceso de preselección, se ha continuado con la selección de estudios, leyendo el contenido completo de dichos artículos. Para esto, se recuperó los artículos completos de las tres bibliotecas digitales antes mencionadas. De los 165 artículos preseleccionados los cuales aportaban con el estudio planteado, 76 respondieron al menos a una pregunta de investigación, los mismos que se someterán a una revisión de calidad, que se explica en la Sección nro. 4.3, de este

capítulo; el resultado de esta evaluación será la selección final de artículos que serán utilizados para rellenar el formulario de extracción de datos explicado más adelante en la Sección nro. 4.4, de este capítulo.

4.2.2.3. Identificación de estudios adicionales mediante la técnica bola de nieve (Snowballing)

Como se ha mencionado en el Capítulo nro. 2, la técnica de bola de nieve que permite buscar artículos potencialmente relevantes para incluirlos en la investigación en base a las referencias de los artículos encontrados. Esta técnica, contempla varios pasos requeridos para la segunda identificación de estudios adicionales, los cuales se explican a continuación.

4.2.2.3.1. Inicio:

En este paso se estableció el inicio de la bola de nieve con 165 artículos, producto de la selección de estudios, que se explicó en la sección anterior. De los cuales, mediante la lectura completa del artículo se identificó 33 referencias que podrían contribuir con el estudio de la investigación; excluyendo las que no aportan con lo planeado. Para identificar las citas de los artículos se usó el buscador académico de Google Académico.

4.2.2.3.2. Iteraciones:

La técnica de bola de nieve tiene dos variantes a considerar: bola de nieve hacia adelante (backward Snowballing) y bola de nieve hacia atrás (forward Snowballing). En esta revisión sistemática de literatura, se consideró solo la bola de nieve hacia atrás, donde se usa la lista de referencias posibles que fueron identificadas para seleccionar artículos nuevos a incluir, debido al tiempo que se le otorga a esta investigación.

- **Bola de nieve hacia atrás:**

De las 33 referencias identificadas, se procedió a realizar una búsqueda manual de estudios para identificar los documentos que puedan aportar con la investigación; de dichas referencias se logró recuperar solamente 28 documentos, debido a que las otras referencias pertenecían a links que redirigían a otras páginas, y cuyos documentos no se encontraban o eran links caídos.

4.2.2.3.3. Inclusión y exclusión

Luego de haber obtenido la posible literatura relevante para el estudio se procedió a aplicar los criterios de inclusión y exclusión definidos en el protocolo que se hace referencia en el Capítulo nro. 3, mediante la lectura completa de los documentos. Como resultado, un total de 17 artículos adicionales fueron incluidos, debido a que aportaron al menos con una pregunta de investigación. Estos documentos se agregaron a los que fueron identificados en la etapa de selección explicada en este capítulo, para ello referirse a la Sección nro. 4.2.2.

4.3. Evaluación de la calidad de los estudios primarios

Los 76 documentos identificados en la selección de estudios primarios y los 17 identificados utilizando la técnica de bola de nieve; fueron sometidos a un proceso de evaluación de calidad, el mismo que se basa en cinco criterios para evaluar la calidad, estos fueron definidos en el protocolo de revisión como referencia del Capítulo nro.3, Sección nro.3.8. Debido a que, según Kitchenham (2007), los datos de calidad pueden ser obtenidos de manera simultánea con la extracción de datos. Es por este motivo, que en esta etapa se leyeron en su totalidad, los artículos identificados. Para la selección de estudios, los cuales se apoyaron en la extracción de datos; ver Figura nro. 4.14; se pudo obtener información para determinar si estos documentos cumplían con el umbral de calidad. De esta forma, de los 93 artículos evaluados 8 se excluyeron por no superar el umbral definido, para tomarlos en cuenta como estudios pertinentes de calidad para este estudio. En la Figura nro. 4.14, se presenta un extracto del proceso que se ha seguido para la evaluación de calidad, la tabla completa será anexada a las evidencias del presente trabajo.

Biblioteca						
Título	A Case Study-Open Source Community and the Commercial Enterprise	A Model for Quality Assurance of OSS Architecture	A Qualitative Study of Open Source Software Development - the OpenEMR Project	A Three-Layer View Model of OSS - toward understanding of diversity of OSS	Adoption of Free Libre Open Source Software (FLOSS) - A risk management perspective	An assurance model for OSS adoption in next-generation telco environments-C-A
Código	IEEE-ODAM1	IEEE-ODAM2	IEEE-ODAM3	IEEE-ODAM4	IEEE-ODAM5	IEEE-ODAM6
Fecha de publicación	2009	2012	2011	2011	2014	2009
DOI	10.1109/ITNG.2009.313	10.1109/CONSEG.2012.6349474	10.1109/ESEM.2011.11	No tiene	10.1109/COMPSSAC.2014.25	10.1109/DEST.2009.5276735
Citaciones Google Scholar	12	0	19	3	8	3
Evaluación de calidad						
¿Los estudios seleccionados cumplen con los criterios de exclusión o inclusión?	1	1	0	-1	0	1
¿Los estudios responden a las preguntas de investigación?	0	0	0	-1	0	0
¿Los artículos abarcan temas sobre éxito, fracaso, riesgo en la adopción de software libre?	0	0	0	-1	0	0
¿El estudio ha sido citado por otros autores Google Scholar?	1	-1	1	0	1	0

Figura 4.14 Evaluación de calidad de los estudios primarios (Autoría propia, 2017)

En la Figura nro. 4.15, se presentan los resultados de los artículos obtenidos en cada fase: Identificación, selección (preselección y selección final) y evaluación de calidad de los estudios primarios.

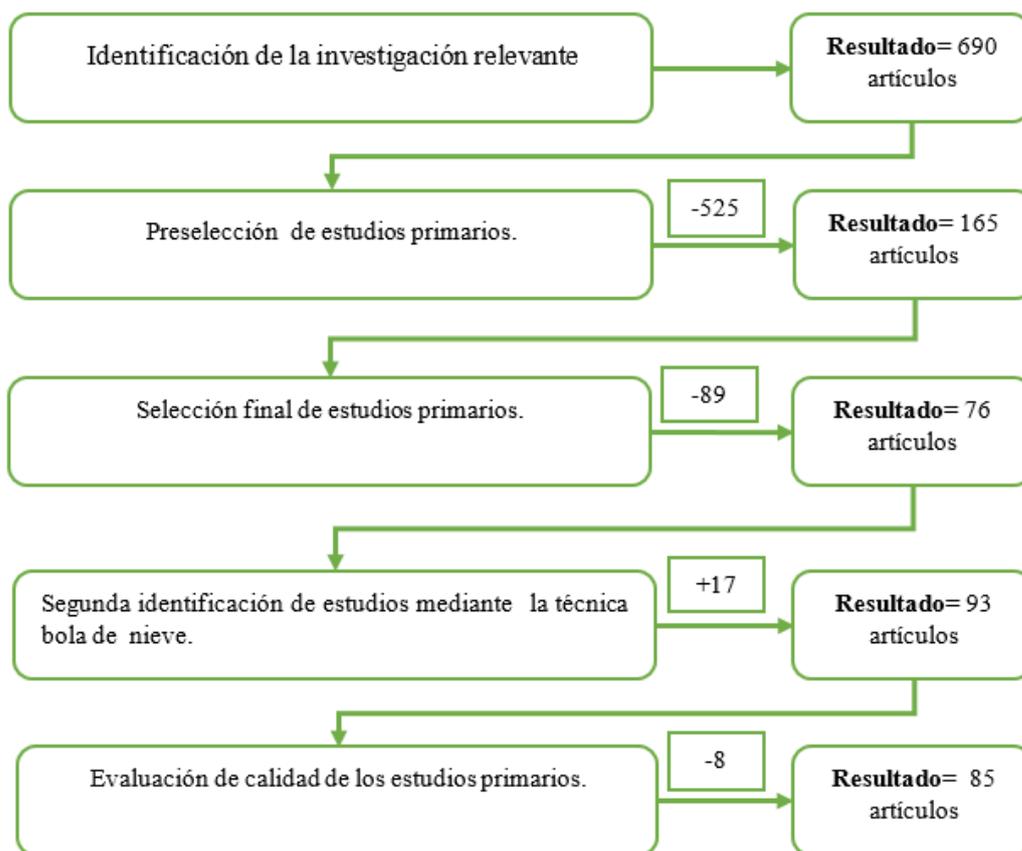


Figura 4.15 Resultados obtenidos de acuerdo con cada fase (Autoría propia, 2017)

4.4. Extracción de datos

Luego de haber seleccionado los estudios primarios que aportaron con la investigación y ser sometidos a una evaluación de calidad, se procede a extraer la información de cada uno de los artículos, en base a las preguntas de investigación, para así rellenar el formulario descrito en el protocolo del Capítulo nro. 3 para la extracción de datos. Para construir dicho formulario se utilizó el paquete de Microsoft Office, haciendo uso de la herramienta de Microsoft Excel.

Para esta fase, de cada artículo se extrajeron los metadatos (Título del documento, El año de publicación del documento, Autores, Resumen, DOI, Tipo de publicación); luego, se procedió a leer el contenido completo del documento y se extrajo la información

que respondían a las preguntas de investigación, completando el formulario de Excel antes mencionado. También, se escribió el objetivo de cada documento, si era importante o no para el estudio realizado y las observaciones encontradas tales como: referencias de documentos que deberían ser revisados por ser potencialmente relevantes para el estudio; además, si el documento daba una explicación amplia del tema investigado o solo se mencionaba algo relacionado, entre otros aspectos. Con el objeto de indicar la amplitud del aporte del documento a la investigación, en cada columna por artículo, se fue identificando por tres tipos de colores: verde (los que aportan ampliamente con la investigación), rojo (los que no aportan con la investigación), amarillo (los que aportan con la investigación, encontrando al menos una observación importante).

La selección final de estudios primarios, evaluación de calidad y bola de nieve se realizó en paralelo con la etapa de extracción de datos.

En la Figura nro. 4.16, se presenta la ejecución de esta etapa, la tabla completa será anexada a las evidencias del presente trabajo.

				1	2	3	4	5
Titulo				A Case Study-Open Source Community and the Commercial Enterprise	A Qualitative Study of Open Source Software Development - the OpenEMR Project	A Three-Layer-View Model of OSS - toward understanding of diversity of OSS - Yamakami - 2011	Adoption of Free, Open-Source Software (FOSS) in risk management perspective - Kenett - 2014	An assurance model for OSS adoption in next-generation software development - Ardagna - 2009
Año de publicación				2009	2011	2011	2014	2009
DOI				10.1109/ITNG.2009.313	10.1109/ESEM.2011.11	No tiene	10.1109/COMPASAC.2014.25	10.1109/DEST.2009.5276735
Autores				K_Gary	John Noll, Sarah Beecham, Dominik Seichter	Toshihiko Yamakami	Ron S. Kenett; Xavier Franch; Angelo Susi; Nikolas Galanis	Claudio A. Ardagna; Massimo Banzì; Ernesto Damiani; Fulvio Frati; Nabil El Ioïni
Codigo				IEEE-ODAM1	IEEE-ODAM3	IEEE-ODAM4	IEEE-ODAM5	IEEE-ODAM6
Capacitación		Capacitación de las partes interesadas en la adopción de OSS	0	0	0	0	0	0
Costos		Beneficios economicos que existan a partir de la adopción del OSS	0	0	0	0	0	0
Gestión eficaz del SE		Administrar correctamente la evolución del software	0	0	0	0	0	0
Actividad Asociada		Cantidad de actividad que produce la comunidad en apoyo al OSS	0	1	0	0	0	0
Soporte		Ayuda que se pueda administrar a la organización con respecto al OSS	0	0	0	0	0	0
Políticas		Políticas corporativas, de TIC y de adopción del OSS	0	0	0	0	0	0
		Disponibilidad del desarrollador		Contar con un grupo de desarrolladores externos al núcleo del proyecto que arregle la mayoría de errores				
		Plan de fechas y requisitos de lanzamiento del OSS						

Figura 4.16 Extracción de datos (Autoría propia, 2017)

Capítulo 5. Análisis de datos

5.1. Introducción

Este capítulo presenta la síntesis del análisis de los datos extraídos en este estudio. Para ello se utiliza una síntesis narrativa acompañada de tablas y gráficos, con el propósito de dar respuesta a cada pregunta de investigación planteada en esta tesis, y así, poder proporcionar una interpretación de lo que se ha encontrado.

5.2. Interpretación de datos extraídos.

De los 690 artículos resultantes de la búsqueda en las tres bibliotecas digitales elegidas, se obtuvo 85 artículos que aportan a la investigación planteada, estos son utilizados para la extracción de datos final; representando un 11% del total de artículos obtenidos. Esto se encuentra representado gráficamente en la Figura nro. 5.1.



Figura 5.1 Artículos relevantes al estudio (Autoría propia, 2017)

De los 85 artículos que aportan a la investigación, el 49% corresponde a actas de conferencia, un 32% son artículos de revista, el 18% capítulos de libro y un 1% reportes técnicos. En la Figura nro. 5.2, se presenta la distribución por tipo de publicación.

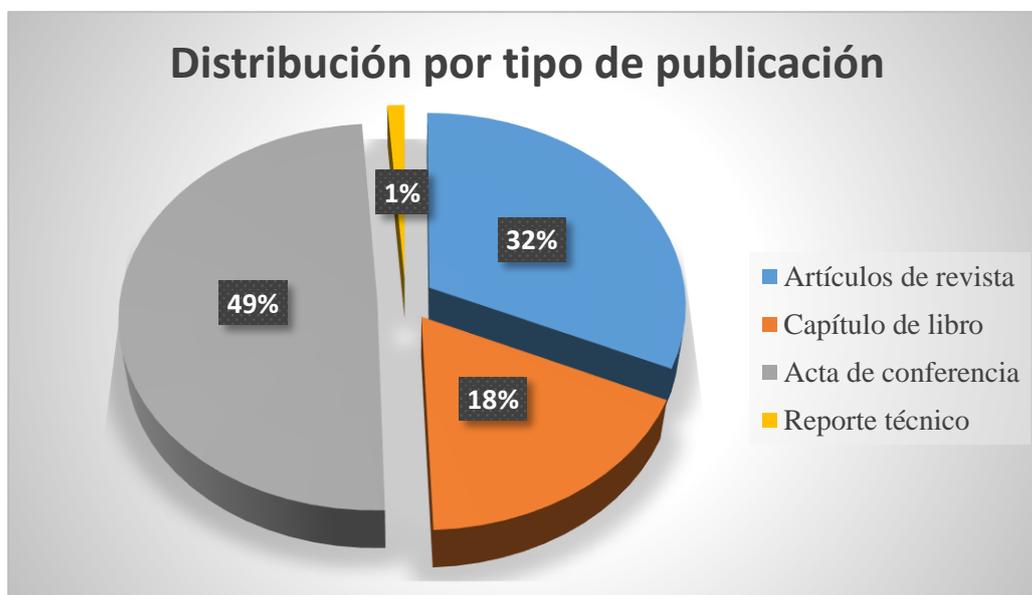


Figura 5.2 Distribución por tipo de publicación (Autoría propia, 2017)

En la Figura nro. 5.3, se presenta el número de artículos relevantes por año que han sido seleccionados para el estudio entre los años 1998 y 2016, sin considerar el año 2017, porque la realización del estudio fue a mediados del año, por lo cual no se puede considerar como una información completa.

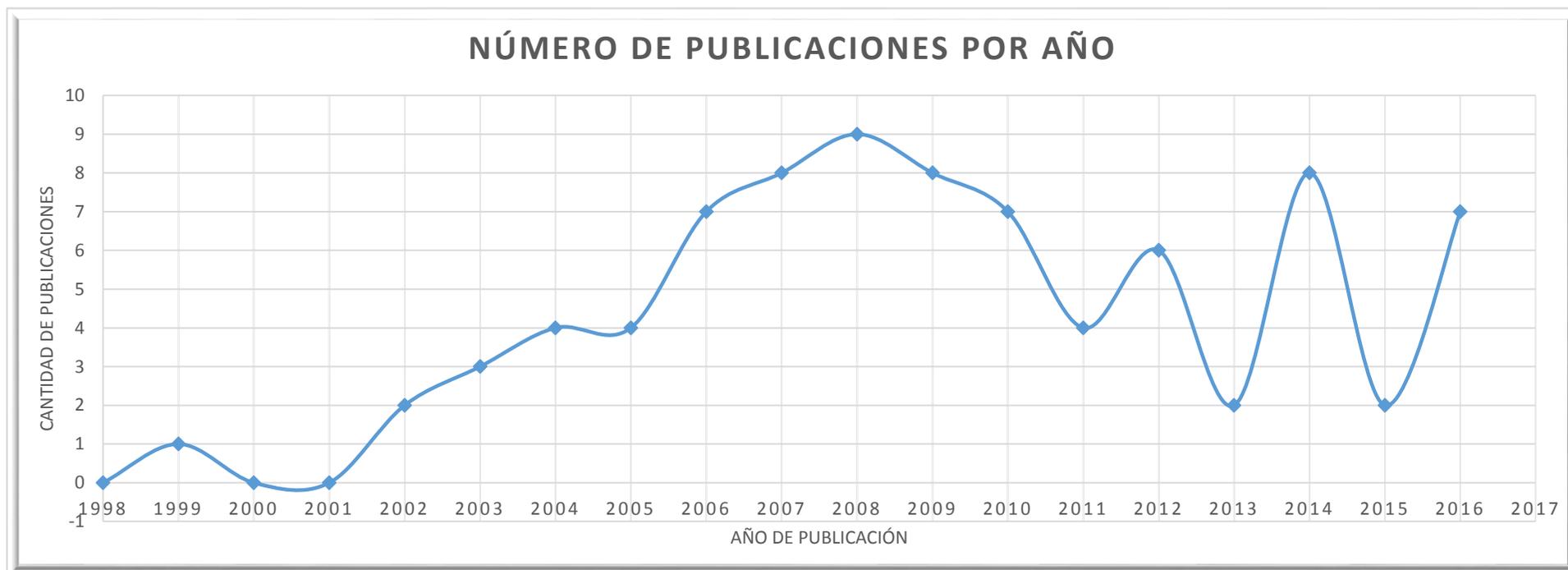


Figura 5.3 Tendencias de publicaciones por año (Autoría propia, 2017)

Los datos que se encuentran en el formulario de extracción, producto de las lecturas de los 85 artículos, han sido utilizados de tal manera que se pueda dar respuesta a las preguntas definidas previamente; estas preguntas se presentan a continuación; para el análisis de los resultados encontrados se ha realizado una clasificación en categorías y subcategorías, partiendo de la relación de las respuestas obtenidas para cada factor de éxito y fracaso, barrera, beneficio y dominios exitosos.

5.2.1. ¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?

5.2.1.1. Factores de éxito

En base a la lectura de los 85 artículos, se han encontrado 68 factores de éxito en la adopción de tecnologías de OSS, las mismas que se han agrupado en 13 categorías y 26 subcategorías. En la Figura nro. 5.4, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

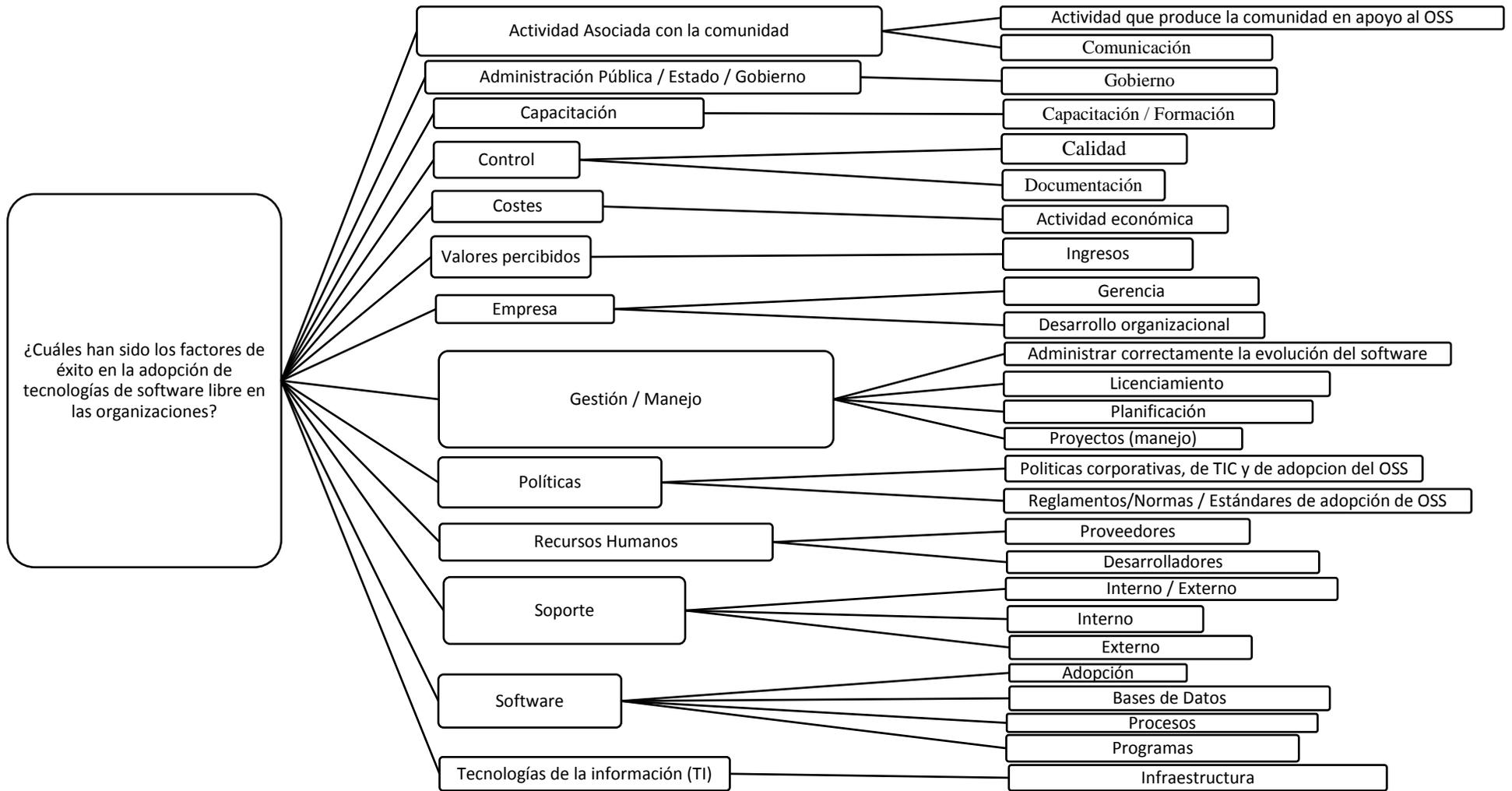


Figura 5.4 Categorías y subcategorías a los que pertenecen los factores de éxito (Autoría propia, 2017)

En la Figura nro. 5.5, se presenta el número de subcategorías que contiene cada categoría.

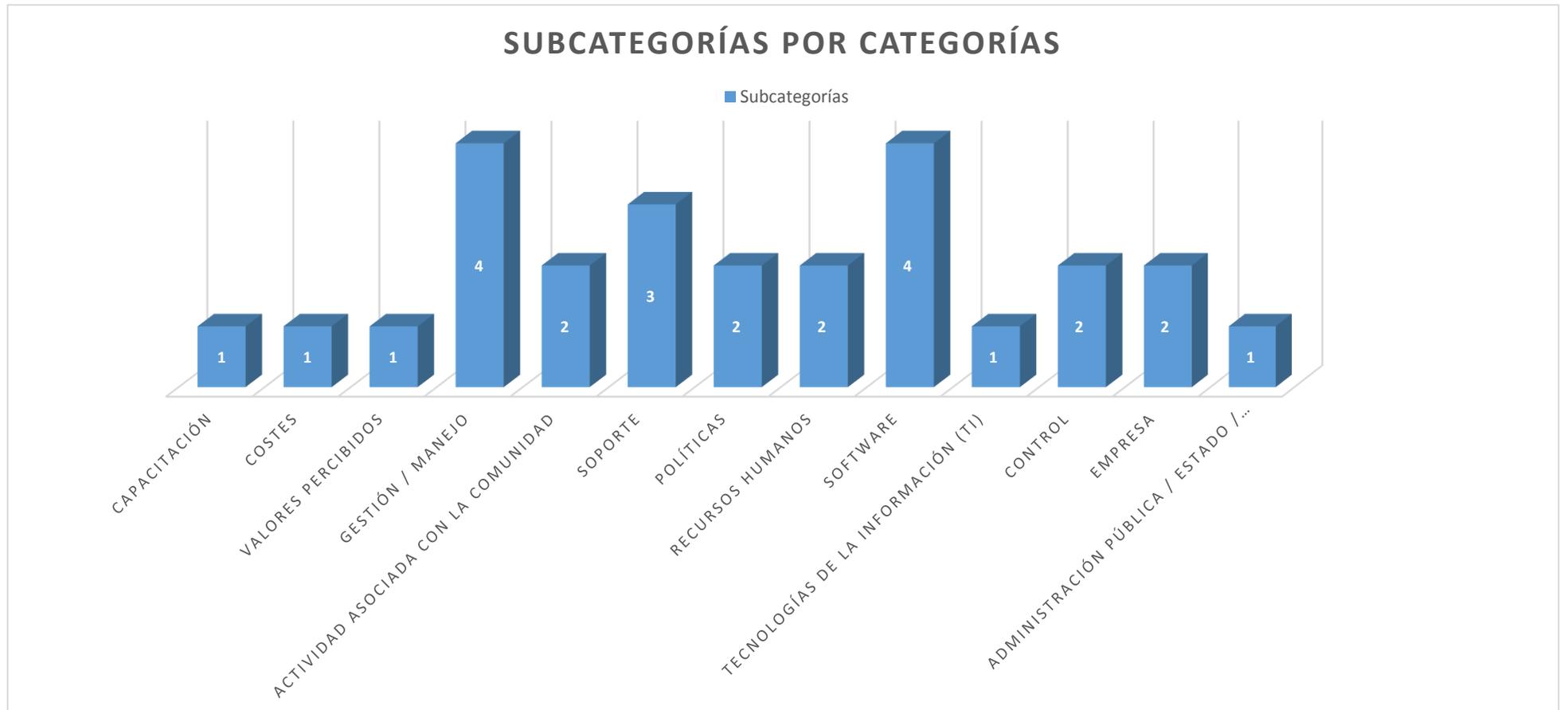


Figura 5.5 Número de subcategorías por categoría (Autoría propia, 2017)

En la Figura nro. 5.6, se presenta el número de factores de éxito encontrados que contiene cada subcategoría.

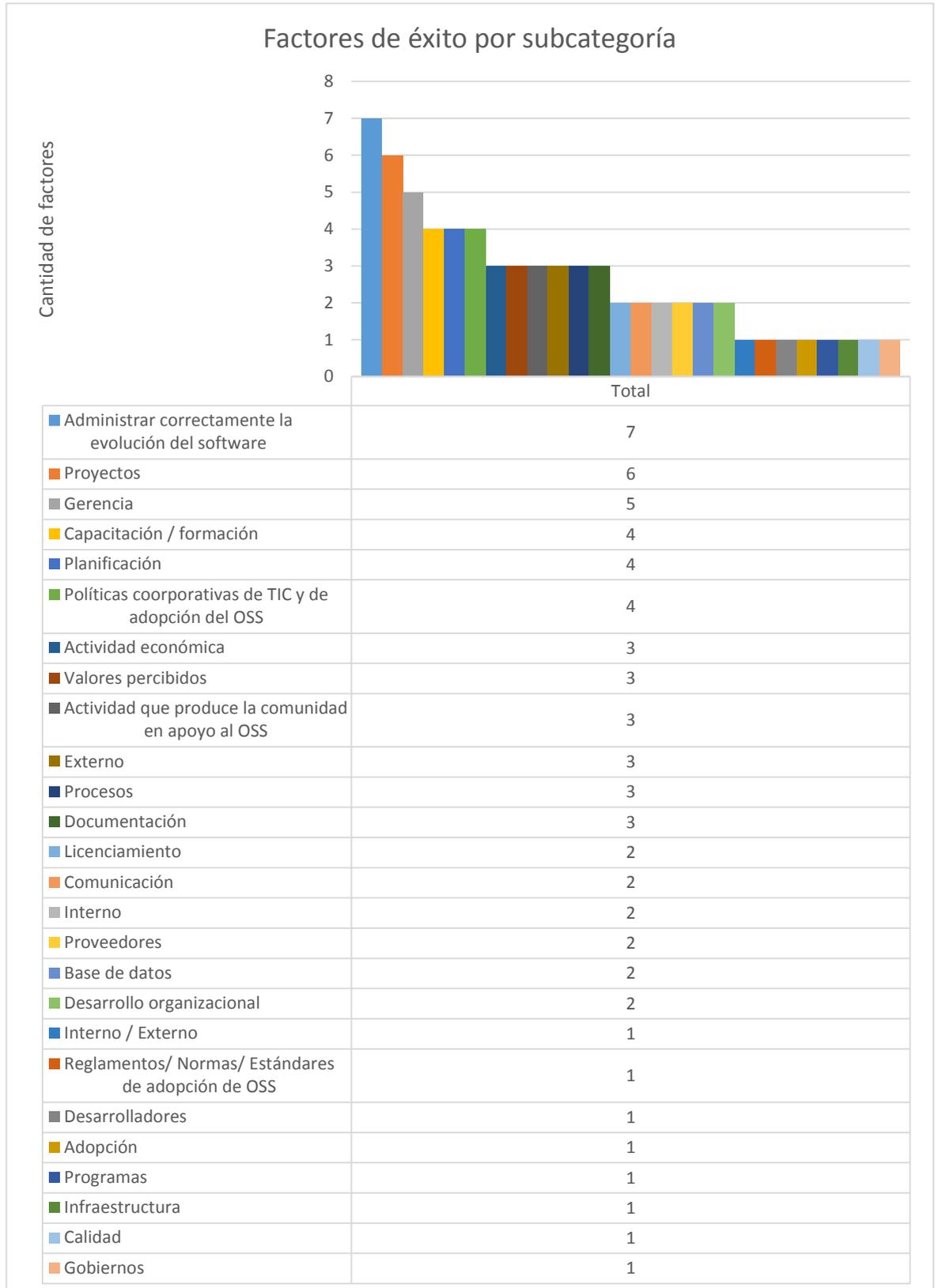


Figura 5.6 Cantidad de factores encontrados por subcategoría (Autoría propia, 2017)

En resumen, de los resultados obtenidos de la búsqueda de factores de éxito en la adopción de software libre, el 28% de factores han sido clasificados en la categoría “Gestión/manejo”, un 10% en la categoría “Software” y “Empresa”, con el 9%, se encuentran la categoría de “Soporte”, con el 7% “Políticas” y “Actividad asociada con la comunidad”, con un 6% “Control y “Capacitación”; mientras que el 5% pertenece a las categorías de “Costes”, “Valores percibidos”, con un 3% “Recursos Humanos”; con un 2% están las categorías “Administración Pública/ Estado/Gobiernos” y “Tecnologías de la información (TIC)”, como se presenta en la Figura nro. 5.7.

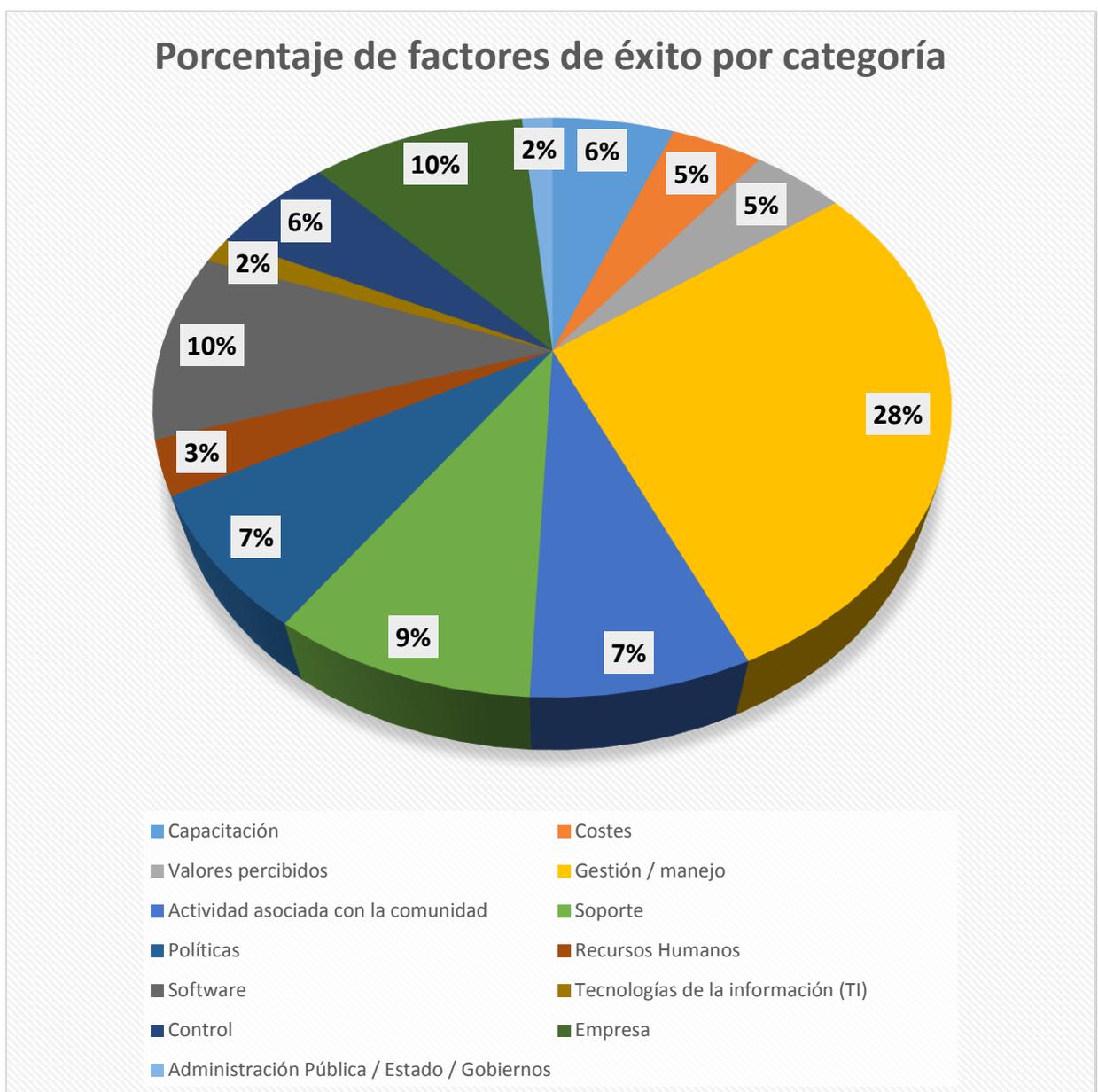


Figura 5.7 Porcentaje de factores de éxito que contiene cada categoría (Autoría propia, 2017)

Como se ha indicado anteriormente producto de la lectura de los artículos seleccionados, se ha encontrado 68 factores de éxito que han sido mencionados por los autores de dicha literatura, los mismos que se presentan en la Tabla nro. 5.1.

Código de Factor	Categoría	Subcategoría	Factores	Frecuencia de factores por artículo	Porcentaje de incidencia
F1	Capacitación (Actividades ejecutadas por una organización, con el fin de mejorar las destrezas del personal)	Capacitación / formación (Procurar tener personal preparado y con experiencia para implementar OSS)	Capacitación general (proceso mediante el cual se estimula al trabajador a incrementar sus conocimientos, destreza y habilidad)	6	5%
F2			Entrenamiento (preparación que se sigue para desempeñar una función)	2	2%
F3			Experiencia personal (Conocimientos previos sobre OSS)	2	2%
F4			Uso de metodologías ágiles	1	1%
F5	Costes (Gastos en la adopción generada por la adopción de FOSS.)	Actividad Económica (Proceso que se lleva a cabo para obtener un producto o servicio)	Identificación oportuna de costes ocultos	1	1%
F6			Asignación de recursos (recursos ocupados para el desarrollo de software)	1	1%
F7			Reducción de costes (por reutilización de código)	1	1%
F8	Valores percibidos (todo aquello que el cliente recibe al adoptar FOSS)	Ingresos (Todas las ganancias que ingresen a la organización)	Beneficios económicos existentes a partir de la adopción de OSS (Reducción de costes en hardware o software al implementar OSS)	5	4%
F9			Utilidad (percibida de la tecnología)	1	1%
F10			Ingresos suficientes (generados por la adopción de OSS)	1	1%
F11	Gestión/manejo (Conjunto de operaciones que se realizan para dirigir y administrar procesos)	Administrar correctamente la evolución del software (Planificar, organizar, integrar, dirigir y controlar la evolución del software)	Madurez del OSS (cómo ha evolucionado y se ha consolidado en el mercado del OSS)	1	1%
F12			Catapulta para nuevos negocios	1	1%
F13			Entender el papel de los contribuyentes individuales en un proyecto y sus patrones de comunicación y colaboración	1	1%
F14			Entender el modelo de negocio	1	1%
F15			Contribución laboral (mide el esfuerzo de la comunidad para avanzar en la funcionalidad del producto).	1	1%
F16			Gestión eficaz de la evolución del software (Aplicar un modelo que permita gestionar eficazmente la evolución del software)	3	3%
F17			Plan de fechas y requisitos de lanzamiento del OSS	1	1%
F18		Licenciamiento (Procedimiento que consiste en	Comprobar la licencia entregada antes de adoptar el OSS	1	1%

F19		verificar las condiciones de uso del FOSS)	Proceso de contribución de parches (que existan los parches necesarios para el proyecto)	1	1%
F20		Planificación (Esfuerzos que se realizan a fin de cumplir objetivos)	Agendamiento de migración de datos (Planes para la correcta transferencia de datos de la organización de un origen a otro)	2	2%
F21			Apoyo en la alta dirección (Es fundamental para la implementación de OSS, la decisión del cambio de la alta dirección)	2	2%
F22			Planificación minuciosa (planificación cuidando todos los detalles basado en todos los aspectos de software)	1	1%
F23			Principios de desarrollo (Estratégica y de evaluación del software)	1	1%
F24			Proyectos (Manejar, planear, organizar, y controlar proyectos de software exitosamente)	Atención al Desarrollador (sirve como un indicador del interés, que los desarrolladores de software tienen en el proyecto)	1
F25		Flexibilidad (personalizar código)		1	1%
F26		Gradualidad		1	1%
F27		Progreso a la meta		1	1%
F28		Satisfacción del cliente		1	1%
F29		Satisfacción del personal		1	1%
F30		Actividad que produce la comunidad en apoyo al OSS (Todo tipo de actividad realizada por la comunidad que contribuya al éxito del FOSS)	Cooperación (Actividad producida por una comunidad activa en el proyecto de OSS)	17	15%
F31	Actividad asociada a la comunidad (Vinculación de la comunidad al proyecto FOSS)		Dell IdeaStorm es una comunidad de innovación que integra a los usuarios finales para mejorar los productos y servicios existentes en la empresa Dell, así como ofrecer nuevas oportunidades de negocio. (Se basa en el modelo producto-cliente)	1	1%
F32			Implementación de las mejores prácticas, basadas en el conocimiento de una comunidad.	1	1%
F33			Comunicación (Intercambiar información entre las comunidades de FOSS)	1	1%
F34			Comunicación eficiente de la comunidad (aplicando herramientas)	1	1%
F35	Soporte (Ayuda que se pueda administrar a la organización con respecto al FOSS)	Interno / Externo (Ayuda que puede prestar personal interno o externo a la organización)	Disponibilidad de soporte (Apoyo o compromiso por parte de la comunidad desarrolladora o interesados dentro de la organización)	8	7%
F36		Interno (Apoyo que brinda personal interno a la organización)	Gestión eficaz de adopción de un proyecto OSS	1	1%
F37			Verificar y evaluar cuidadosamente el producto	1	1%
F38		Externo (Apoyo que brinda personal externo)	Grupo de desarrolladores externos al proyecto que arreglen errores	1	1%

F39		a la organización)	Reputación	1	1%
F40			Solución rápida a errores por parte del proveedor	1	1%
F41	Políticas (Proceso de tomar decisiones que se aplican a todos los miembros de un grupo)	Políticas corporativas de TIC y de adopción del OSS	Cuestiones legales (Aspectos legales)	2	2%
F42			Mantenerse legalmente protegido	1	1%
F43			Políticas empresariales (Aspectos que influyen en la organización y en la adquisición de servicios)	4	4%
F44			Políticas de implementación de OSS (Creación de políticas claras y comprensibles para la implementación de FOSS para la parte legal, económica y social)	3	3%
F45		Reglamentos/ Normas/ Estándares de adopción de OSS	Estándares	1	1%
F46	Recursos Humanos (Aporte de empleados o colaboradores de una organización)	Proveedores (Aporte que pueden brindar los proveedores de FOSS a la organización)	Disponibilidad del contratista	1	1%
F47		Apertura hacia el exterior (para que el proveedor pueda entregar documentación final, con políticas y reglamentos sobre la adopción de OSS)	1	1%	
F48		Desarrolladores (Aporte que pueden brindar los desarrolladores de FOSS a la organización)	Disponibilidad del desarrollador (Aportes de los desarrolladores de FOSS)	1	1%
F49	Software (Conjunto de programas informáticos que permiten ejecutar distintas tareas en un dispositivo)	Adopción (Implementación de FOSS)	Adopción incremental (Implementación gradual del proceso de gestión de FOSS)	2	2%
F50		Base de datos (Conjunto de datos pertenecientes a un mismo contexto)	Base de datos de repositorio con libre acceso	1	1%
F51		Disponibilidad de datos	1	1%	
F52		Procesos (Ejecución de acciones realizadas con un orden)	Automatización de procesos	1	1%
F53		Reutilización de componentes	1	1%	
F54		Velocidad (en desarrollo, solución de errores)	1	1%	
F55		Programas (Aplicaciones de FOSS)	Facilidad de instalación	1	1%
F56	Tecnologías de la información (TI)	Infraestructura (Conjunto de dispositivos físicos y aplicaciones de software requeridas para manejar una organización)	Flexibilidad de la infraestructura de TI	1	1%
F57	Control (Evaluación del rendimiento de sistemas de código abierto)	Calidad (Conjunto de los mecanismos, acciones y herramientas realizadas para detectar la	Calidad del sistema OSS (calidad en el desarrollo del producto)	1	1%

		presencia de errores.)					
F58		Documentación (Evaluación del estado de la documentación de FOSS)	Calidad del trabajo	1	1%		
F59			Conversión de documentos sin problema	1	1%		
F60			Mejora la usabilidad de la documentación	1	1%		
F61	Empresa (Institución dedicada a actividades o persecución de fines económicos o comerciales para satisfacer las necesidades de bienes o servicios de los solicitantes)	Gerencia (Persona o conjunto de personas que se encargan de dirigir, gestionar o administrar una sociedad, empresa u otra entidad.)	Estrategias para seguir al líder	1	1%		
F62				Estructura organizativa (estructura jerárquica plana)	1	1%	
F63				Liderazgo	1	1%	
F64				Toma de decisiones críticas	1	1%	
F65				Tomar el tiempo necesario para analizar los sistemas existentes y pensar a largo plazo en la migración de datos en organizaciones con ambientes complejos	1	1%	
F66				Desarrollo organizacional (Trata sobre el desarrollo, funcionamiento y efectividad en las relaciones humanas dentro de una organización con un fin en común)	Crecimiento	1	1%
F67					Filantropía (Suficiente dinero, progreso y propósito de trabajo dentro de la empresa)	1	1%
F68	Administración Pública / Estado / Gobiernos (Conjunto de organismos y personas que se dedican a la administración o el gobierno de los asuntos de un estado.)	Gobiernos (Conjunto de personas y organismos que gobiernan o dirigen una división político-administrativa)	Fuerte apoyo de gobierno	1	1%		
Totales				113	100%		

Tabla 5.1 Factores de éxito (Autoría propia, 2017)

La Figura nro. 5.8, hace referencia a la Tabla nro. 5.1, mencionada anteriormente; en donde, en la gráfica se presentan los nombres de los factores con más de una incidencia y los factores con 1 incidencia son denominados como “*Otros*”.

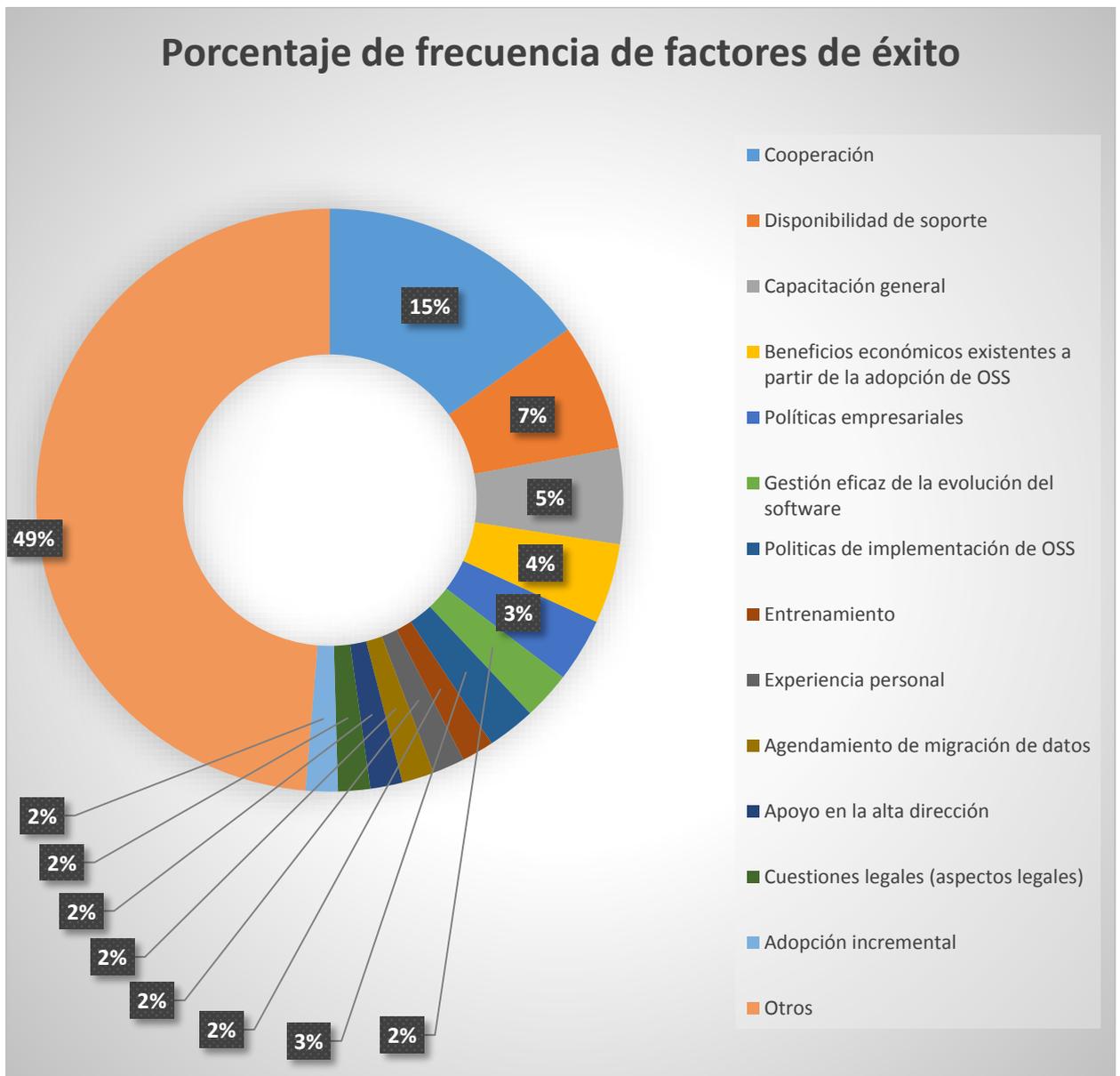


Figura 5.8 Representación gráfica de frecuencia de los factores de éxito (Autoría propia, 2017)

Partiendo de la información expuesta anteriormente en la Tabla nro. 5.1, y Figura nro. 5.8, se puede decir que el factor “*cooperación*” es el que más se menciona con un 15% de incidencias; seguido por: “*disponibilidad de soporte*” con un 7%, “*capacitación general*” con un 5%, “*beneficios económicos*” y “*políticas empresariales*” con un 4%, “*gestión eficaz de la evolución del software*” y “*políticas de implementación de OSS*”,

con un 3%; con un 2% se encuentran “*Entrenamiento*”, “*Experiencia personal*”, “*Agendamiento de migración de datos*”, “*Apoyo de alta dirección*”, “*Cuestiones legales*”, “*Adopción incremental*”, y con un 1% de incidencias se encuentran otros factores dando un total de 49% la sumatoria de ellos; estos factores serán descritos a continuación con más profundidad. Se recalca que la información obtenida de los factores de éxito, en la mayoría de las publicaciones se hace una breve referencia sobre el tema, o solo se menciona y no se presenta evidencia amplia de lo que se expone en el artículo, de esta manera, se obtiene lo siguiente:

- **Categoría “*Capacitación*”:** Subcategoría “*Capacitación/formación*”, de los 11 artículos obtenidos en esta subcategoría, 6 mencionan que un factor de éxito es la “*Capacitación general*”; según Petrijevcenin & Sudarevic (2012), señalan que uno de los factores cruciales para el éxito de la implementación de software de código abierto, es tener personal capacitado, y más aún si no se cuenta con apoyo externo de empresas especializadas en el tema, que puedan brinden soporte externo. Fitzgerald & Kenny (2003) exponen que, el mayor aprendizaje para el Hospital Beaumont, ha sido capacitar a su personal para que brinde soporte en soluciones OSS.

De la misma manera dentro de los 12 artículos antes mencionados en la categoría “*Capacitación*” y dentro de la subcategoría “*Capacitación/formación*”, 2 artículos indican que un factor de éxito es el “*Entrenamiento del personal*”, y otros 2 mencionan la “*Experiencia personal*”; en donde Lennerholt *et al.* (2008) resalta, que es de suma importancia almacenar las experiencias pasadas, que han tenido los empleados con el uso del OSS, no solo dentro de una organización, sino también de fuera; de esta manera es posible obtener conocimiento de sucesos anteriores de lo que se debe y no se debe hacer. Por otra parte; 1 artículo menciona que un factor de éxito es “*El uso de metodologías ágiles*” para la capacitación/formación del personal.

En la Figura nro. 5.9, se presenta el número de artículos que hacen mención a cada factor en la categoría “Capacitación” y en la subcategoría de “Capacitación/formación”.

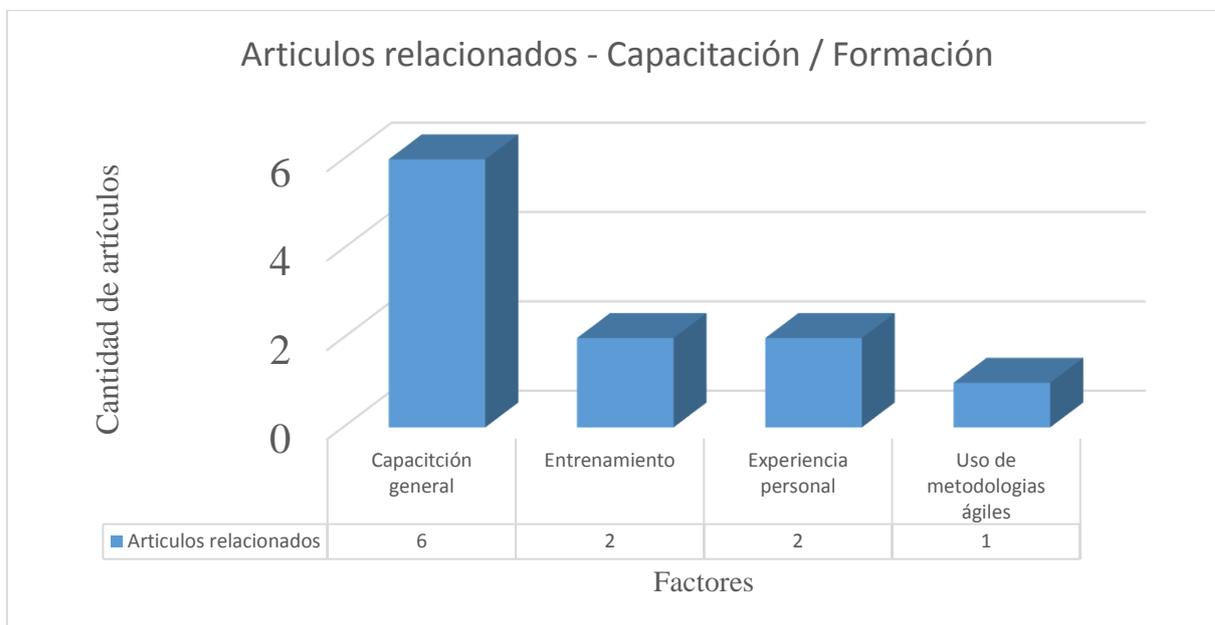


Figura 5.9 Artículos que hacen mención a la categoría “Capacitación” y subcategoría “Capacitación formación”
(Autoría propia, 2017)

- **Categoría “Costes”:** Subcategoría “*Actividad económica*”, de los 3 artículos encontrados que hacen referencia a esta subcategoría, 1 artículo revela que un factor de éxito es “*Identificación oportuna de costes ocultos*”, dicho factor es señalado por Silic & Back (2017), en donde mencionan que, la adopción del OSS puede generar costes ocultos, debido a que ningún software es gratuito.

Adicionalmente en la categoría “Costes” y subcategoría “*Actividad económica*”, de los 3 artículos encontrados que hacen referencia a esta subcategoría, 1 artículo menciona que un factor de éxito es la “*Asignación de recursos para el desarrollo de software libre*”, en otro artículo se encuentra, la “*Reducción de costes por reutilización de código*”.

- **Categoría “Valores percibidos”:** Subcategoría “*Ingresos*”, de los 7 artículos encontrados que hacen referencia a esta subcategoría, 5 artículos mencionan que un factor de éxito son “*Beneficios económicos existentes a partir de la adopción*”.

de OSS”, donde Ven *et al.* (2008), indica que con en el estudio realizado, pudieron descubrir que una gran cantidad de organizaciones, señalan que el costo reducido fue el impulso para adoptar OSS; mientras que otro artículo hace referencia a la “*Utilidad percibida de la tecnología*” como factor de éxito.

Adicionalmente en la categoría “*Valores percibidos*” y subcategoría “*Ingresos*”, de los 7 artículos encontrados que hacen referencia a esta subcategoría, 1 artículo señala a otro factor de éxito como los “*Ingresos suficientes (generados por la adopción del OSS)*”, según Barcomb (2015), el uso de FOSS le permite a la empresa tener ingresos sostenibles, manteniendo la organización y la estabilidad laboral.

En la Figura nro. 5.10, se presenta el número artículos que hacen mención a cada uno de los factores en la categoría “*Valores percibidos*” y subcategoría “*Ingresos*”.

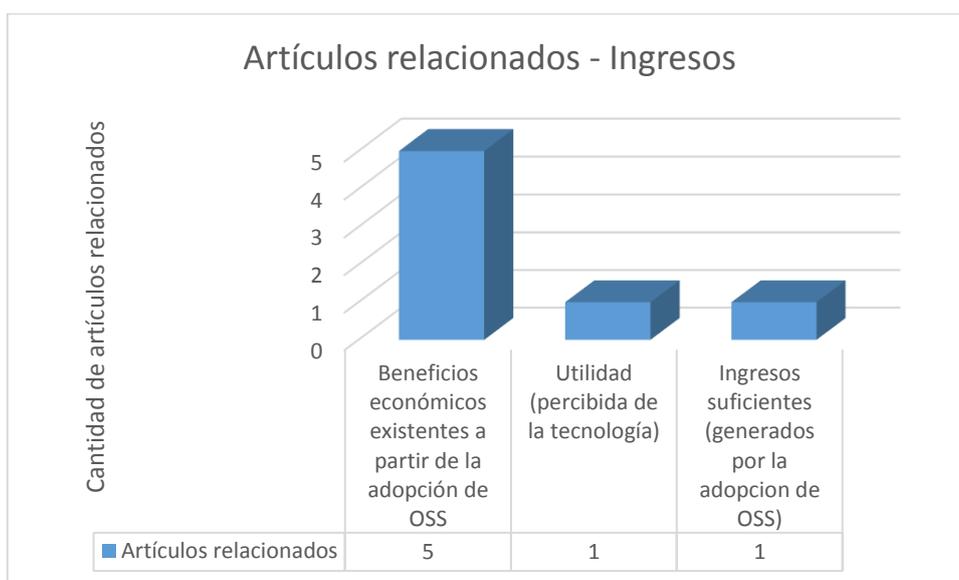


Figura 5.10 Artículos que hacen mención a la categoría “Valores percibidos” y subcategoría “Ingresos” (Autoría propia, 2017)

- **Categoría “Gestión/manejo”:** Subcategoría “*Administrar la correcta evolución del software*”, de los 9 artículos encontrados que hacen referencia a esta subcategoría, 1 artículo menciona que un factor de éxito es la “*Madurez del OSS*”, según Ven *et al.* (2008), señala que varias de las empresas entrevistadas mencionaron que la evolución del OSS, ha generado un alto índice de confiabilidad.

Dentro de la misma categoría y subcategoría, se menciona para cada artículo diferente, a los siguientes, como factores de éxito a la “*Catapulta para los nuevos negocios*”, el “*Entender el papel de los contribuyentes individuales en un proyecto y sus patrones de comunicación y colaboración en un tiempo de vida de un proyecto*” y el “*Entender el modelo de negocio*”; ya que según Petrijevcenin y Sudarevic (2012), mencionan que es importante comprender el modelo de negocio que existe detrás del desarrollo de software de código abierto. Antes de que sea implementado en las Instituciones Públicas.

Además, dentro de la misma categoría y subcategoría se encontraron en otros artículos, factores como la “*Contribución laboral*”, según Tsay *et al.* (2012), este permite medir el esfuerzo que realiza la comunidad para avanzar en la funcionalidad del producto; y en otros 3 artículos distintos se consideran como factores a la “*Gestión eficaz de la evolución del software*” y 1 a la “*Planificación de fechas y requisitos de lanzamientos*”, siendo los mismos factores de éxito.

En la Figura nro. 5.11, se presenta el número artículos que hacen mención a cada factor en la categoría “Gestión/manejo” y subcategoría “Administrar la correcta evolución del software”.

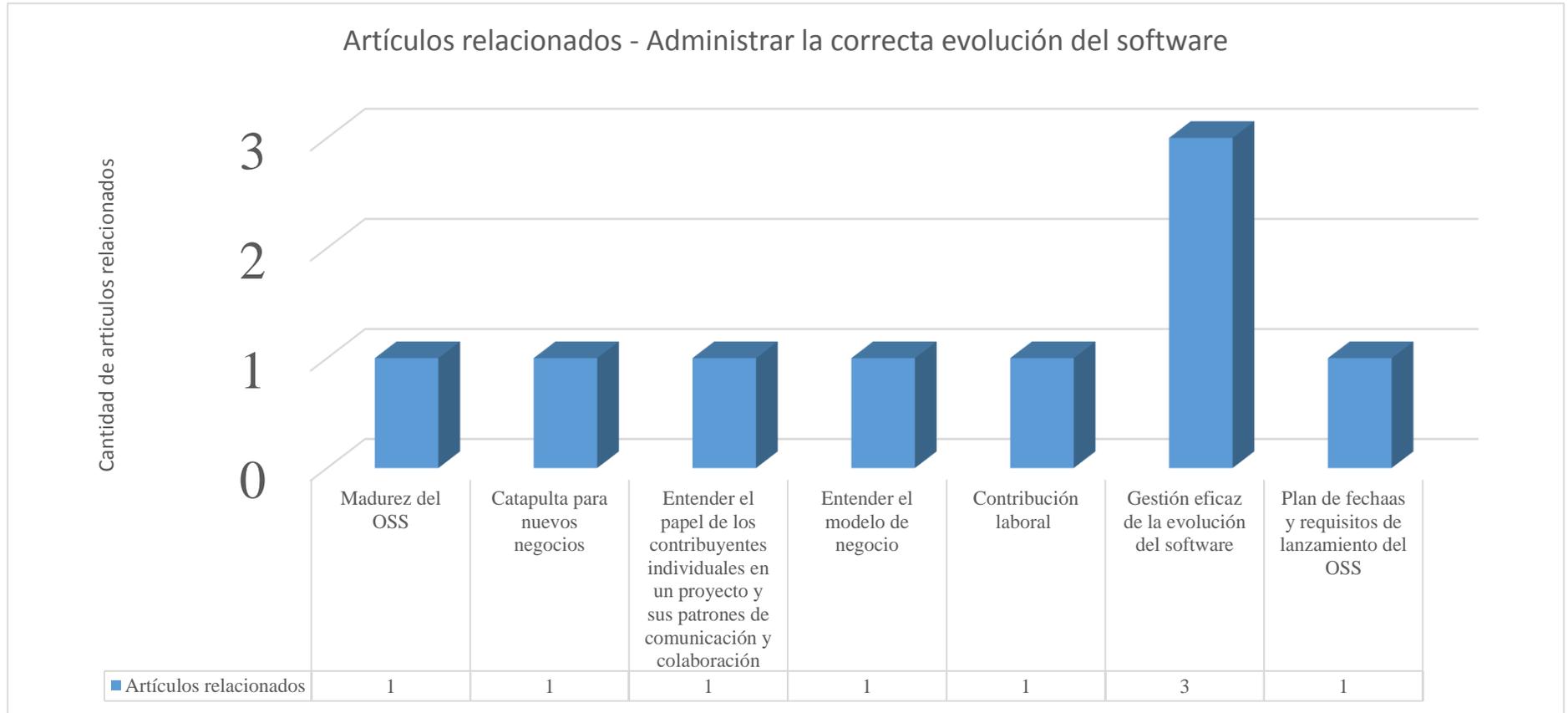


Figura 5.11 Artículos que hacen mención a la categoría “Gestión/manejo” y subcategoría “Administrar la correcta evolución del software” (Autoría propia, 2017)

Dentro de la categoría “*Gestión/manejo*” también se encuentra la subcategoría “*Licenciamiento*” de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es “*Comprobar la licencia entregada antes de adoptar software libre*”, ya que según Petrijevcenin & Sudarevic (2012), existen varios tipos de licencias en el código abierto que tienen diversas características y condiciones para los usuarios. Otro artículo indica que un factor es el “*Proceso de contribución de parches (que existan los parches necesarios para el proyecto)*”.

En la categoría “*Gestión/manejo*” existe otra subcategoría, llamada “*Planificación*”; de los 6 artículos encontrados que hacen referencia a esta subcategoría, 2 artículos mencionan que un factor de éxito es el “*Agendamiento de la migración de datos*”, otros 2 artículos consideran el “*Apoyo en alta dirección*”, donde Fitzgerald & Kenny (2003), indican el caso del Hospital Beaumont, donde la adopción del OSS recibió apoyo incondicional del CEO (director ejecutivo), concluyendo que el apoyo de la alta dirección es crítico para el éxito de la adopción del OSS; 1 solo artículo menciona que la “*Planificación minuciosa*” es un factor de éxito, cuidando todos los aspectos relacionados al software y en 1 artículo más, se indica que un factor de éxito es el “*Principio de desarrollo*”, que de acuerdo a Lennerholt *et al.* (2008), es importante reconocer los principios rectores a ser adoptados, en relación con la estrategia y el proceso de evaluación, al momento de desarrollar software.

En la Figura nro. 5.12, se presenta el número artículos que hacen mención a cada factor en la categoría “*Gestión/manejo*” y subcategoría “*Planificación*”.

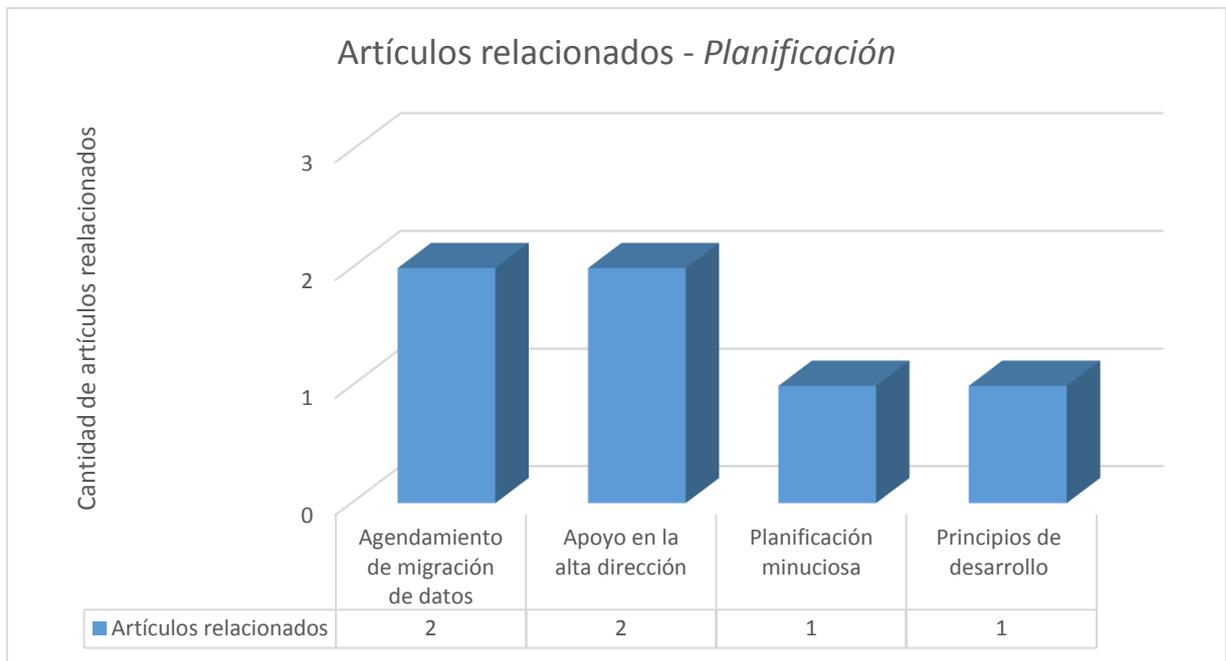


Figura 5.12 Artículos que hacen mención a la categoría “*Gestión/manejo*” y subcategoría “*Planificación*” (Autoría propia, 2017).

En la categoría “*Gestión/manejo*” y subcategoría “*Proyectos (manejo)*”, de los 6 artículos encontrados que hacen referencia a esta subcategoría, se menciona, que los factores de éxito son: 1 artículo con “*Atención al desarrollador*” ya que según, Tsay *et al.* (2012), el factor sirve como un indicador de interés que los desarrolladores de software tienen en el proyecto; otro artículo menciona la “*Flexibilidad (personalizar código)*”, 1 artículo con la “*Gradualidad (implementación de manera paulatina de todos los aspectos del software)*”, 1 artículo con el “*Progreso a la meta*”, 1 artículo con la “*Satisfacción del cliente*” y un último artículo con la “*satisfacción personal*”.

- **Categoría “*Actividad asociada con la comunidad*”:** Subcategoría “*Actividad que produce la comunidad en apoyo al OSS*”, de los 19 artículos que hacen referencia a esta subcategoría, 17 de ellos mencionan que un factor de éxito es la “*cooperación de la comunidad (actividad producida por una comunidad activa en el proyecto de OSS)*”, según Deen Sethanandha *et al.* (2010) comentan, que es

importante tener en cuenta la contribución de una comunidad activa ya que esta afecta directamente tanto la calidad, como a la evolución del OSS; Fitzgerald & Kenny (2003), mencionan que la adopción de OSS obliga a dar seguimiento continuo para analizar, cómo está la cooperación de la comunidad, determinando de esta manera si un proyecto es activo o está a punto de expirar; de igual forma 1 artículo indica que un factor de éxito es la *“Dell IdeaStorm es una comunidad de innovación que integra a los usuarios finales para mejorar los productos y servicios existentes en la empresa Dell, así como ofrecer nuevas oportunidades de negocio. (Se basa en el modelo producto-cliente)”* como factor de éxito, además 1 artículo menciona como factor de éxito la *“Implementación de las mejores prácticas, basadas en el conocimiento de una comunidad”*.

En la Figura nro. 5.13, se presenta el número de artículos que hacen mención a la categoría *“Actividad asociada con la comunidad”* y subcategoría *“Actividad que produce la comunidad en apoyo al OSS”*.

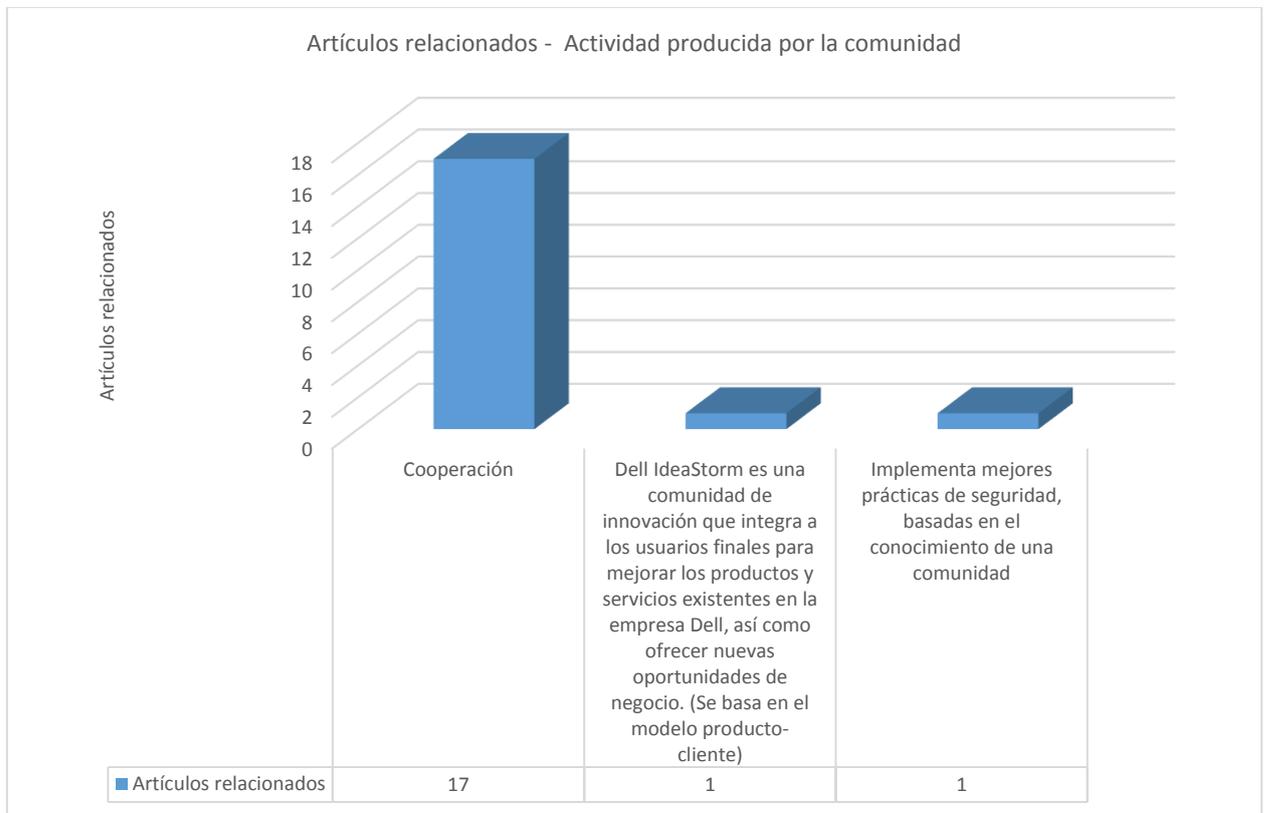


Figura 5.13 Artículos que hacen mención a la categoría *“Actividad asociada con la comunidad”* y subcategoría *“Actividad que produce la comunidad en apoyo al OSS”* (Autoría propia, 2017).

Dentro de la misma categoría “*Actividad asociada con la comunidad*”, también pertenece la subcategoría “*Comunicación*”, de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es “*La comunicación abierta con la comunidad de software libre (en términos de licencia)*”, según Fendt *et al.* (2016), señalan que es importante participar activamente con la comunidad de software libre, con una cercana colaboración; ya que esto disminuye los esfuerzos de analizar el estado de la licencia de los componentes de FOSS, del mismo modo minimiza el esfuerzo para la obtención de requerimientos que ayudan a avalar el cumplimiento de la licencia; en otro artículo se indica que la “*Comunicación eficiente de la comunidad (mediante herramientas)*”, es otro factor de éxito.

- **Categoría “*Soporte (Ayuda que se pueda administrar a la organización con respecto al OSS)*”:** Subcategoría “*Interno/Externo*”, existen 8 artículos que mencionan que un factor de éxito es la “*Disponibilidad de soporte*”, Ven *et al.* (2008), recalca que son varias las organizaciones que utilizan algún tipo de soporte externo de OSS, por razones diversas; por ejemplo en el estudio que realizan los antes mencionados autores encontraron que algunas organizaciones requerían de un consultor externo, ya que no contaban con personal especializado para instalación y soporte dentro de la empresa; otras organizaciones utilizan soporte como un seguro para que sus problemas se resuelvan en la brevedad posible, si algo sale mal. Así mismo otros autores como, Silic & Back (2017), indican que el fracaso es inevitable en la adopción de OSS, si no se tiene presente el factor de soporte, en diferentes niveles y formas, ya sea de tipo: política organizacional, departamento de TI (tecnología de la información), entre otros. Fitzgerald & Kenny (2003), mencionan que en el hospital “Beaumont Hospital”, todavía mantienen una estrategia de soporte para OSS como una opción, ya que la libertad para descargar aplicaciones no significa que se pueda usar de manera efectiva.

Dentro de la categoría “*Soporte (Ayuda que se pueda administrar a la organización con respecto al OSS)*”, existe otra subcategoría, llamada “*Interno*”, en donde de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es “*La Gestión eficaz de adopción de*

un proyecto OSS”, otro artículo indica que “*Verificar y evaluar cuidadosamente el producto*”, es otro factor de éxito.

Dentro de la categoría “*Soporte (Ayuda que se pueda administrar a la organización con respecto al OSS)*”: Subcategoría “*Externo*”, de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es el “*Grupo de desarrolladores externos al proyecto que arreglen errores*”, otro artículo menciona como factor a la “*Reputación*”, y finalmente 1 artículo menciona que un factor de éxito es la “*Solución rápida de errores por parte del proveedor*”.

- **Categoría “*Políticas*”:** Subcategoría “*Políticas corporativas de TIC y de adopción del OSS*”, de los 10 artículos encontrados que hacen referencia a esta subcategoría; en 2 de ellos se menciona que un factor de éxito son las “*Cuestiones legales (Comprender el aspecto legal del OSS)*”; según Petrijevcenin & Sudarevic (2012), el aspecto legal es algo que se debe tener en cuenta debido a que, cuando se adquiere software el usuario está comprometido a seguir los reglamentos dispuestos para la adopción del software obtenido; de igual manera 1 artículo indica que un factor de éxito es “*Mantenerse legalmente protegido*”. De acuerdo con Silic & Back (2017), recalcan que existe una variedad de herramientas y componentes OSS, que pueden contener código que no está permitido utilizar, por tal motivo los responsables de TI (Tecnología de la Información) de la empresa, deben prestar mucha atención y tener cautela con el cumplimiento de estas políticas, para no tener problemas legales, y así tratar de minimizar en lo posible el riesgo.

De la misma manera dentro de los 10 artículos antes mencionados en la categoría “*Políticas*” y en la subcategoría “*Políticas corporativas de TIC y de adopción del OSS*”, 4 artículos indican que un factor de éxito son las “*Políticas empresariales*”, Fendt *et al.* (2016), señala que se debería tener una política y una estrategia corporativa en general, no solo para la adopción del FOSS, sino también para aspectos que están relacionados con la actividad empresarial; otros 3 artículos hacen referencia a un factor de éxito, como las “*Políticas de implementación de OSS*”, los autores, Lennerholt *et al.* (2008) y Petrijevcenin & Sudarevic (2012);

resaltan que es importante y trascendente tener un plan estratégico que impulse arquitecturas abiertas, como también políticas de adopción flexible claras y completas, y que dicho plan debería sustentarse en el entendimiento de todos los aspectos referentes al software: licencias de código abierto, políticas para la selección de software, entre otros.

En la Figura nro. 5.14, se presenta el número de artículos que hacen mención a la categoría “Políticas” y subcategoría “Políticas corporativas de TIC y de adopción del OSS”.



Figura 5.14 Artículos que hacen mención a la categoría “Políticas” y subcategoría “Políticas corporativas de TIC y de adopción del OSS” (Autoría propia, 2017).

Dentro de la categoría “Políticas”: Subcategoría “Reglamentos/Normas/Estándares de adopción de OSS”, 1 artículo menciona que el factor de éxito de adopción de software libre son mantener “Estándares”, Al-Ajlán (2009), señala que el acceso a código abierto facilita la legibilidad y facilidad de uso del sistema en un tiempo prolongado, ya que el enfoque abierto, que está en estrecha relación con los estándares permite asegurar que las necesidades del desarrollador sean cubiertas a largo plazo. Además, menciona que el enfoque abierto, que se base en estándares internacionales facilita la independencia de proveedores.

- **Categoría “Recursos Humanos”:** Subcategoría “Proveedores”, de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es la “Disponibilidad del contratista” y 1 artículo considera a la “Apertura hacia el exterior (para que el proveedor pueda entregar documentación final, con políticas y reglamentos sobre la adopción de OSS)” como un factor de éxito, según Fendt *et al.* (2016), resalta, que se debe obtener declaraciones de los proveedores de OSS, donde indiquen los componentes de FOSS a ser implementados en el software desarrollado, conjuntamente con la evidencia de la manera que cumplan con los términos de licencia.

Dentro de la categoría “Recursos Humanos”: subcategoría “Desarrolladores”, 1 artículo menciona que el factor de éxito en la adopción de software libre es la “Disponibilidad del desarrollador”.

- **Categoría “Software”:** subcategoría “Adopción”, 2 artículos mencionan que el factor de éxito de adopción de software libre es la “Adopción incremental”, de acuerdo Fendt *et al.* (2016), indican que la implementación del FOSS en una organización debe ser gradual, hasta adquirir experiencia, ya que esto aumenta las posibilidades de éxito en la empresa.

Dentro de la categoría “Software”: subcategoría “Base de datos”, de los 2 artículos encontrados que hacen referencia a esta subcategoría; 1 artículo menciona que un factor de éxito es tener una “Base de datos de repositorio con libre acceso” que contenga la experiencia de adopción de software libre dentro o fuera de la organización, según Ruffin & Ebert (2004), mencionan que Alcatel posee una base de datos con todas las experiencias que se hayan obtenido del software de código abierto, a través de estudios o por medio de la práctica. La base de datos de libre acceso contiene información del OSS: de quien ha usado, de los distribuidores ideales y confiables, juntamente con su forma de comportarse; entre otros aspectos. Para ello los autores mencionados anteriormente en este estudio, indican que las empresas pequeñas deberían tener este tipo de base de conocimiento, con lo que pueden responder a las inquietudes del software de

código abierto, evitando de esta manera pagar por experiencia externa. Otro artículo menciona como factor de éxito a la “*Disponibilidad de datos*”.

Dentro de la categoría “*Software*”: subcategoría “*Procesos*”, de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es la “*Automatización de procesos*”, Fendt *et al.* (2016), expone que se necesitan herramientas que estén bien integradas para automatizar el proceso de gestión de FOSS; de esta manera el esfuerzo será mínimo para el análisis de FOSS. Otro artículo menciona que un factor de éxito es la “*Reutilización de componentes de software libre*”, ya que según Ajila & Wu (2007), la reutilización de componentes aporta a la productividad, porque disminuye tiempos de comercialización; y finalmente 1 artículo da a conocer que un factor de éxito es la “*Velocidad (en desarrollo y solución de errores)*”.

En la categoría “*Software*”: subcategoría “*Programas*”, 1 artículo encontrado menciona que el factor de éxito es la “*Facilidad de instalación*” en la adopción de software libre, en donde Kovacs *et al.* (2004), resalta que hubo una migración exitosa de diez pequeñas empresas de administración pública a herramientas de Open Office; en donde una de las experiencias más características, fue la fácil instalación de dicha herramienta.

- **Categoría “*Tecnologías de la información (TI)*”:** Subcategoría “*Infraestructura*”, en 1 artículo encontrado se menciona que el factor de éxito de adopción de software libre es la “*Flexibilidad de la infraestructura de TI*”.
- **Categoría “*Control*”:** Subcategoría “*Calidad*”, 1 artículo encontrado menciona que el factor de éxito de adopción de software libre es la “*Calidad del sistema OSS (calidad en el desarrollo del producto)*”.

En la categoría “*Control*”: subcategoría “*Documentación*”, de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un factor de éxito es la “*Calidad de trabajo*”, otro artículo menciona que es la

“*Conversión de documentos sin problema*”, ya que según Kovacs *et al.* (2004), menciona que otra experiencia de migrar a OpenOffice fue la conversión de documentos de Microsoft Office a OpenOffice sin dificultad, añade, además, que los documentos disminuyeron también de tamaño con OpenOffice. Finalmente 1 artículo da a conocer que un factor de éxito es la “*Mejora la usabilidad de la documentación*”.

- **Categoría “*Empresa*”:** Subcategoría “*Gerencia*”, de los 5 artículos encontrados que hacen referencia a esta subcategoría; se menciona, que los factores de éxito son: 1 artículo con “*Estrategia para seguir al líder*”, otros artículos indican que se debe tener una “*Estructura organizativa (estructura organizativa plana)*”, “*Liderazgo*”, se debe considerar como factor a la “*Toma de decisiones críticas*” y finalmente 1 artículo menciona que un factor de éxito es “*Tomar el tiempo necesario para analizar los sistemas existentes y pensar a largo plazo en la migración de datos en organizaciones con ambientes complejos*”, según Silic & Back (2017), resaltan que una migración a OSS puede llevar mucho tiempo y que se debería pensar a largo plazo, debido a que puede haber problemas con archivos existentes en la organización; como es el caso de Múnich (proyecto iniciado por la ciudad de Múnich para migrar a OSS), y la manera de afrontar esos problemas en la migración será revisar los sistemas que posee la empresa y planificar cómo será la migración.

En la categoría “*Empresa*”: subcategoría “*Desarrollo organizacional*”, 1 artículo encontrado menciona que el factor de éxito de adopción de software libre es el “*Crecimiento*” y otro artículo indica que es “*Filantropía (Suficiente dinero, progreso y propósito de trabajo dentro de la empresa)*”.

- **Categoría “*Administración Pública/Estado/Gobiernos*”:** subcategoría “*Gobiernos*”, en 1 artículo encontrado se menciona que el factor de éxito de adopción de software libre es el “*Fuerte apoyo del gobierno*”.

5.2.1.2. Factores de fracaso

De la lectura de los 85 artículos, se han encontrado 26 factores de fracaso en la adopción de tecnologías de OSS, que se han agrupado en 8 categorías y 17 subcategorías. En la Figura nro. 5.15, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

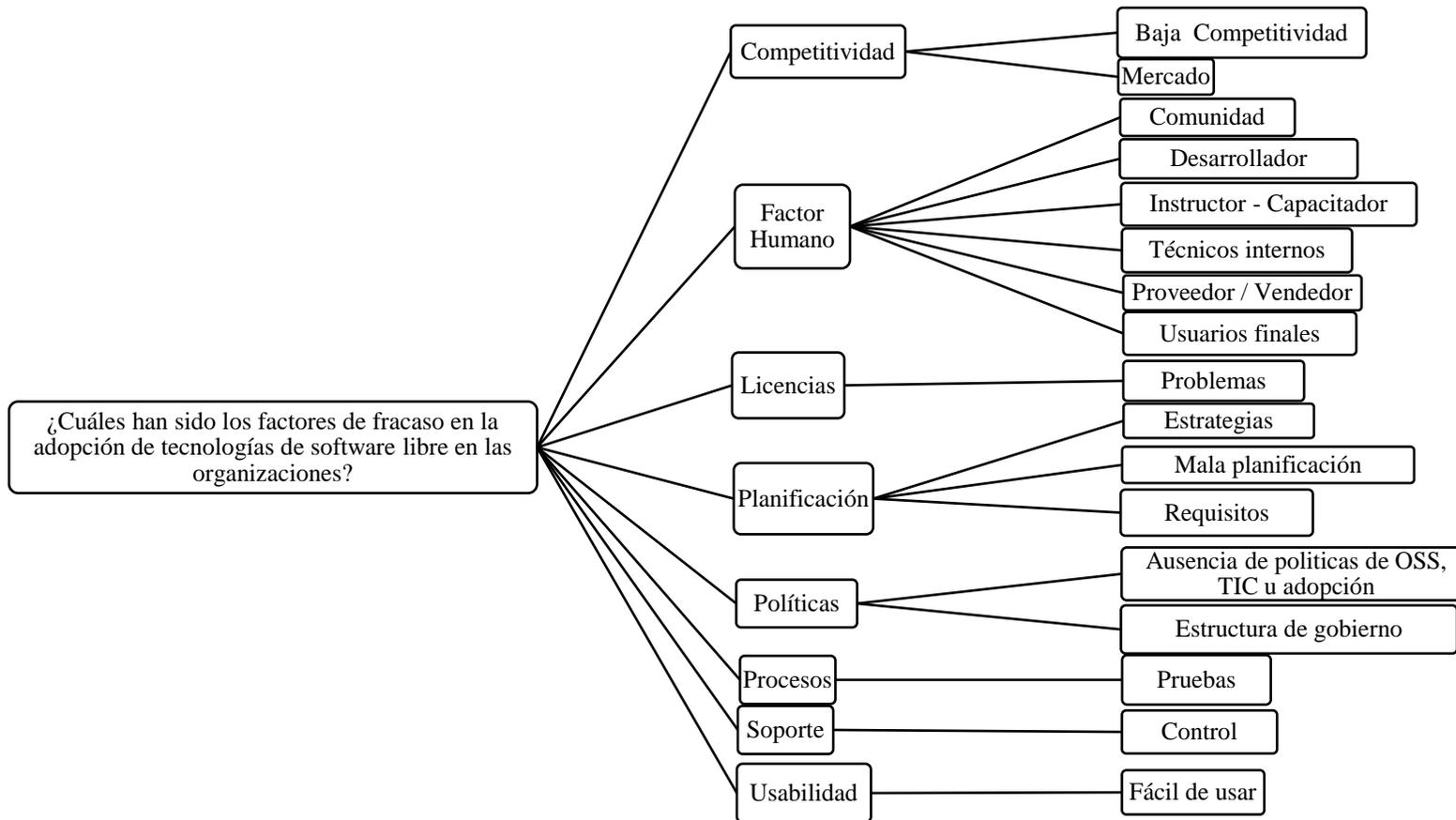


Figura 5.15 Categorías y subcategorías a los que pertenecen los factores de fracaso (Autoría propia, 2017).

En la Figura nro. 5.16, se presenta el número de subcategorías que contiene cada categoría en factores de fracaso.

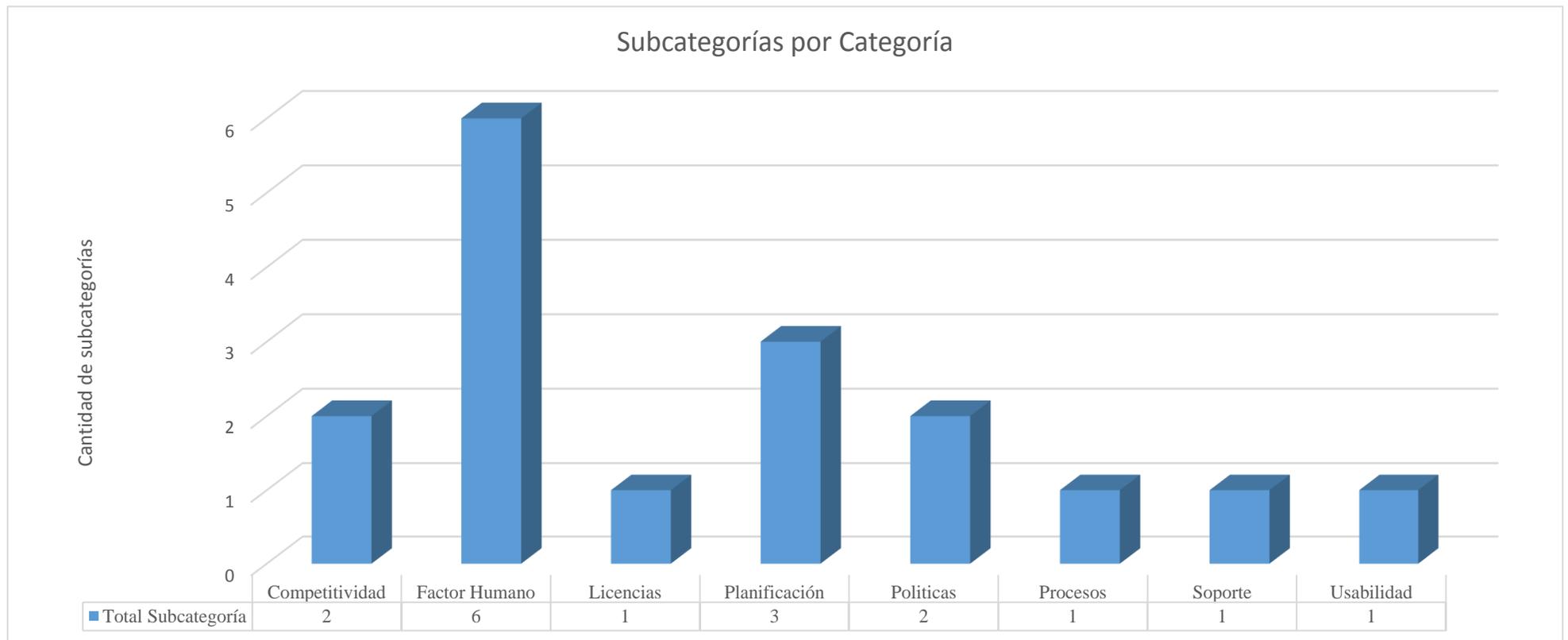


Figura 5.16 Factores de fracaso: número de subcategorías por categoría (Autoría propia, 2017).

En la Figura nro. 5.17, se presenta el número de factores de fracaso encontrados, que contiene cada subcategoría.

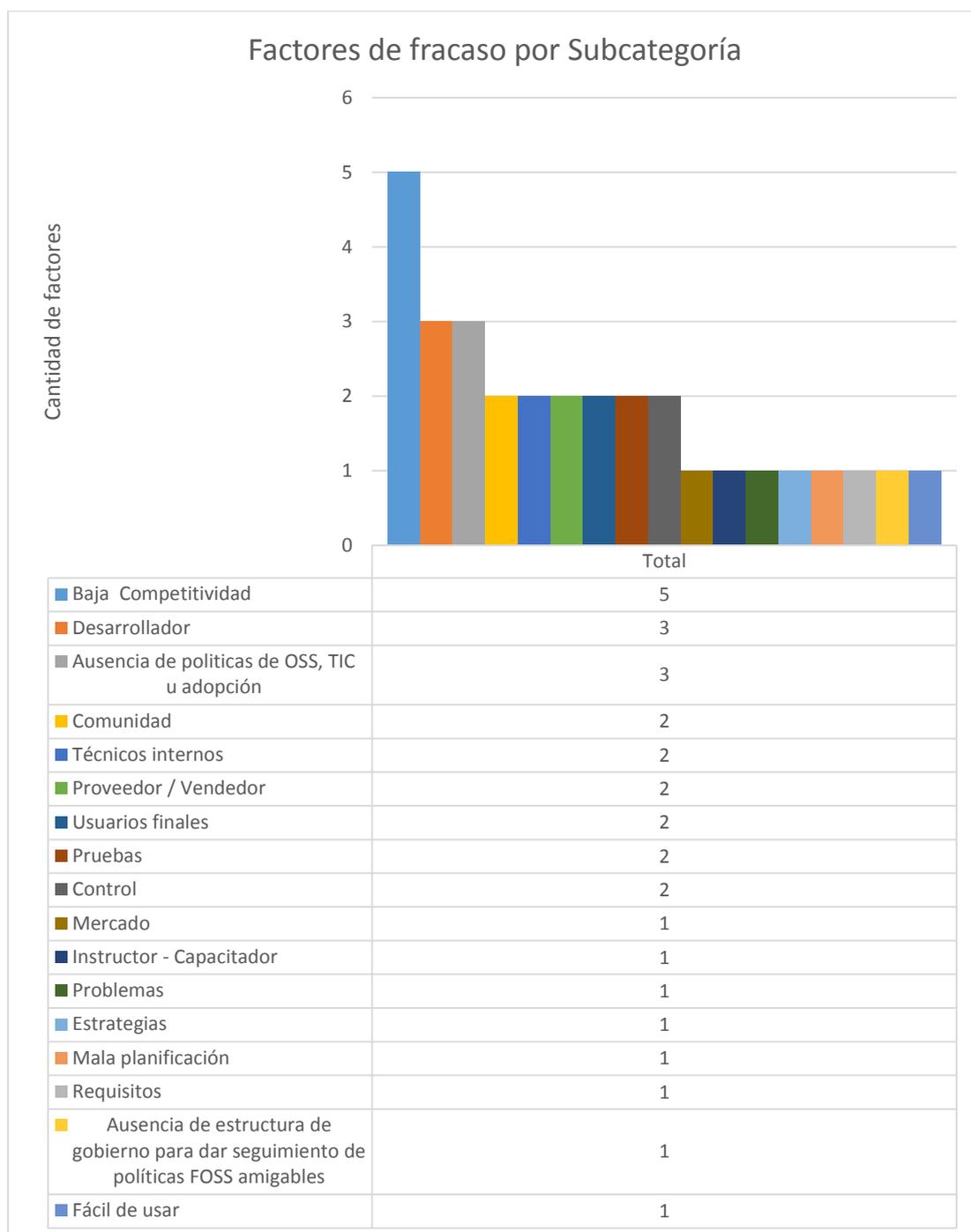


Figura 5.17 Factores de fracaso por subcategorías (Autoría propia, 2017).

En resumen, de los resultados obtenidos de la búsqueda de factores de fracaso en la adopción de software libre, el 39% de los factores han sido clasificados en la categoría “Factor humano”, un 19% en la categoría “Competitividad”, el 13% están en la categoría

“Políticas, y un 10% de factores están dentro de la categoría “*Planificación*”; por otro lado el 7% en “*procesos*”; mientras que el 6% pertenece a la categoría “*Soporte*”, un 3% está en la categoría “*Usabilidad*” y el 3 % restante de los factores se encuentran en la categoría “*Licencias*”, como se presenta en la Figura nro. 5.18.

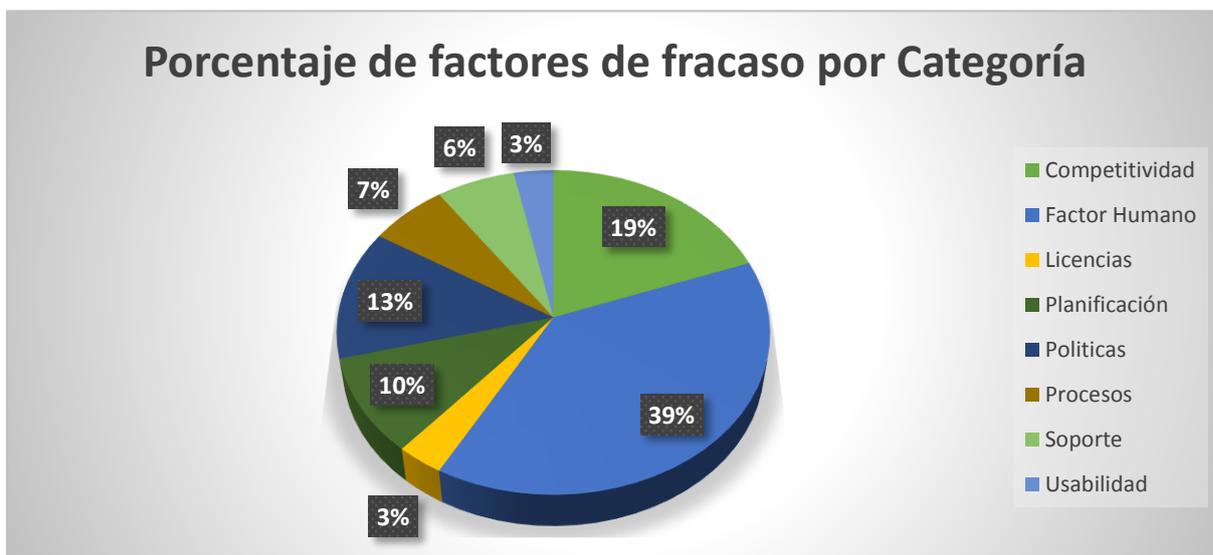


Figura 5.18 Porcentaje de factores de fracaso que contiene cada categoría (Autoría propia, 2017).

Como se ha indicado anteriormente, producto de la lectura de los artículos seleccionados, se han encontrado 26 factores de fracaso que han sido mencionados por los autores de dicha literatura, los mismos que se presentan en la Tabla nro. 5.2.

Número de factor	Categoría	Subcategoría	Factor	Frecuencia de factores por artículo	Porcentaje de incidencia
F1	Competitividad (Capacidad de generar la mayor satisfacción de los consumidores)	Baja Competitividad (Satisfacción baja de los consumidores)	Poca competitividad al frente de ofertas comerciales.	4	12%
F2			Falta de credibilidad en el FOSS	1	3%
F3		Mercado (Exceso o competencia fuerte por parte de desarrolladores)	Mercado Competitivo	1	3%
F4	Factor Humano (Comportamiento del personal desde el punto de vista de recursos humanos)	Comunidad (Grupo de individuos con un objetivo en común)	Falta de liderazgo	1	3%
F5			Inclusión de miembros en la comunidad FOSS, como fanáticos militantes	1	3%

F6			Conflictos entre desarrolladores	1	3%
F7		Desarrollador (Programador que se dedica a desarrollar software)	Falta de tiempo para dedicarse al desarrollo de software	1	3%
F8			Pérdida de interés por parte del desarrollador (Pérdida de interés por continuar el proyecto)	2	6%
F9			Instructor – Capacitador (Personal que capacita a las partes interesadas)	Insatisfacción por parte de los instructores con la inclusión de FOSS	1
F10		Técnicos internos (Profesionales que pertenecen a la organización)	Falta de conocimientos técnicos internos en OSS	1	3%
F11			Falta de experiencia	1	3%
F12		Proveedor / Vendedor	Difusión de incertidumbre, miedo y duda que infunden los vendedores propietarios	1	3%
F13			Pérdida del patrocinador	1	3%
F14		Usuarios finales (Persona que usa el producto)	Falta de investigación por parte del usuario	1	3%
F15			Pérdida de interés por parte del usuario final	1	3%
F16	Licencias (Autorización concedida para realizar una determinada actividad)	Problemas (Obstáculos encontrados con el licenciamiento)	Dificultades relacionadas con las licencias del OSS	2	6%
F17		Estrategias (Planificación de acciones para alcanzar los objetivos propuestos)	Prácticas comerciales sin estrategias	1	3%
F18	Planificación (Definir un plan de trabajo)	Mala planificación (Plan de trabajo inadecuado)	Mala planificación y precipitación	1	3%
F19		Requisitos (Condición necesaria para implementar FOSS)	No se comprenden los requisitos por parte del desarrollador	1	3%
F20	Políticas (Aspectos relacionados a un esquema de gobierno)	Ausencia de políticas (Falta de políticas en la organización)	Ausencia de políticas (Falta de políticas OSS, TIC o de adopción)	3	9%
F21		Estructura de gobierno (Modo de organización de un gobierno)	Ausencia de estructura de gobierno para dar seguimiento de políticas FOSS amigables	1	3%
F22	Procesos (Actividad de ejecutar una secuencia de instrucciones)	Pruebas (Identificar propiedades de software)	No existe un riguroso proceso de pruebas	1	3%

F23			Proyectos obsoletos frente a nuevos sistemas operativos	1	3%
F24	Soporte (Asistencia que se proporciona al proyecto)	Control (Medición de desempeño)	No contar con un grupo que dé solución a errores	1	3%
F25			Número de idiomas que tiene el proyecto	1	3%
F26	Usabilidad (Medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema)	Fácil de usar (Facilidad de uso de un sistema)	Falta de usabilidad	1	3%
Totales				33	1,00

Tabla 5.2 Factores de fracaso identificados (Autoría propia, 2017).

La Figura nro. 5.19, hace referencia a la Tabla nro. 5.2, mencionada anteriormente; en donde, en la gráfica se presenta el nombre de los factores con más de una incidencia y los factores con 1 incidencia son denominados “*Otros*”.

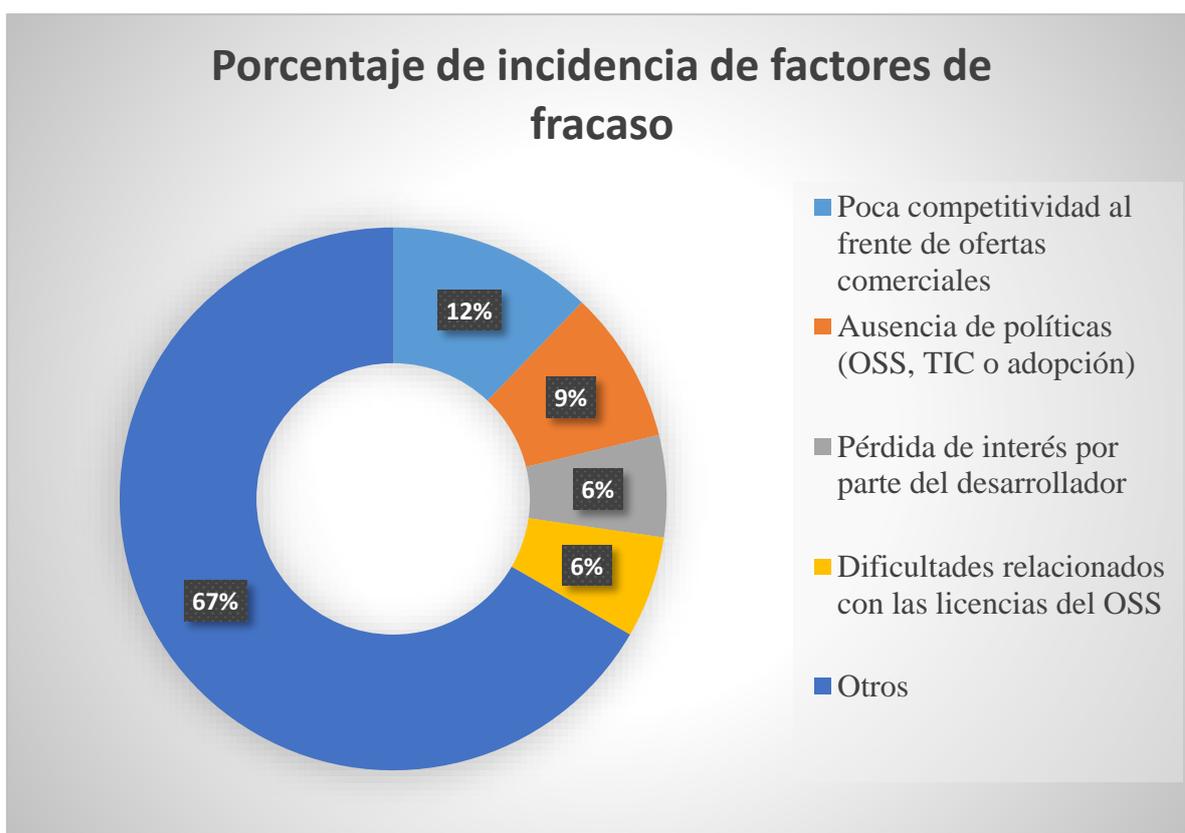


Figura 5.19 Representación gráfica de menciones de los factores de fracaso (Autoría propia, 2017).

A partir de la información que se presentó en la Tabla nro. 5.2, y Figura nro. 5.19, se puede decir que el factor de fracaso más mencionado en los artículos es la “*Poca competitividad al frente de ofertas comerciales*”, con un 12%; seguido de “*Ausencia de políticas (OSS, TIC o adopción)*”, con un 9%; “*Pérdida de interés por parte del desarrollador*” y “*Dificultades relacionados con las licencias del OSS*” con un 6%; mientras los otros factores de fracaso son mencionados por los artículos cada uno en un 3%, dando un total de 67%, siendo nominados como “*Otros*”.

De la información obtenida de los factores de fracaso, se identifica que la mayoría de los autores que se refieren al tema o solo mencionan los factores en base a la experiencia, al igual que sucedió con los factores de éxito.

Para el caso de los factores de fracaso, no se muestra evidencia amplia de lo que se expone; además, se debe indicar que son escasos los artículos que hablen de fracaso en la adopción de software libre. Es importante indicar que ciertos factores de fracaso encontrados están relacionados al desarrollo de proyectos de OSS; sin embargo, se ha tomado en cuenta el desarrollo de proyectos, por tener cierto grado de importancia con lo investigado, de esta manera, se presentan las siguientes categorías y subcategorías:

- **Categoría “*Competitividad*”:** Subcategoría “*Baja competitividad*”, de los 5 artículos obtenidos que hacen referencia a la subcategoría, en 4 artículos se menciona que un factor de fracaso en la adopción de software libre es la “*Poca competitividad al frente de ofertas comerciales*”; Sahraoui (2009), menciona en su artículo que la poca competitividad del proyecto de código abierto en GNU, que se llevaba a cabo en la Universidad del Golfo, frente a las ofertas comerciales, fue un caso de fracaso.

En la categoría “*Competitividad*” y en la subcategoría “*Baja competitividad*”, de los 5 artículos obtenidos que hacen referencia a la subcategoría, 1 artículo menciona a la “*Falta de credibilidad en el FOSS*” como factor de fracaso, ya que según Sahraoui (2009), uno de los acontecimientos que detuvieron el desarrollo y el éxito indiscutible del proyecto de adopción de código abierto en GNU (Sistema operativo tipo Unix) en la Universidad del Golfo, fue la falta de aceptación, acompañado con el fuerte rechazo por parte del personal universitario; se conoce

que ciertos individuos comprendían que el software de código abierto no tenía credibilidad y que había poca probabilidad de progresar con el tiempo.

En la Figura nro. 5.20, se presenta el número de artículos que hacen mención a la categoría “Competitividad” y en la subcategoría “Baja competitividad”.

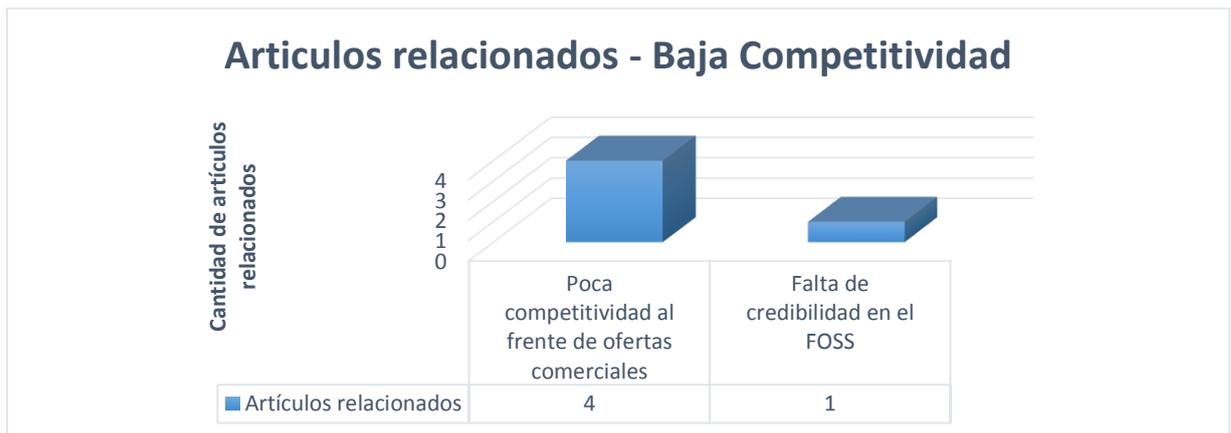


Figura 5.20 Artículos que hacen mención a la categoría “Competitividad” y en la subcategoría “Baja competitividad” (Autoría propia, 2017).

Dentro de la categoría “Competitividad”: subcategoría “Mercado”, se menciona 1 artículo que es “Mercado competitivo”, como factor de fracaso, razón por la cual afecta al proyecto, haciendo que estos sean abandonados por no poder competir dentro del mercado.

- **Categoría “Factor Humano”:** Subcategoría “Comunidad”, de los 2 artículos encontrados que hacen referencia a la subcategoría: 1 artículo menciona que un factor de fracaso es la “Falta de liderazgo”; mientras que otro artículo indica que es la “Inclusión de miembros en la comunidad FOSS, como fanáticos militantes”.

Como parte de la categoría “Factor Humano”: subcategoría “Desarrollador”, de los 4 artículos encontrados que hacen referencia a la subcategoría: 1 artículo menciona que un factor de fracaso son los “Conflictos entre desarrolladores”, o entre desarrolladores con propietarios de proyectos por diversas razones, como la falta de experiencia del desarrollador usando tecnologías del proyecto, problemas legales, entre otros (Coelho & Valente, 2017); en otro artículo se menciona que

la “*Falta de tiempo para dedicarse al desarrollo de software*”, es un factor de fracaso determinante y finalmente en 2 artículos se señala que la “*Pérdida de interés por parte del desarrollador*”, es un factor grave de fracaso.

En la Figura nro. 5.21, se presenta el número de artículos que hacen mención a la categoría “*Factor Humano*” y subcategoría “*Desarrollador*”.

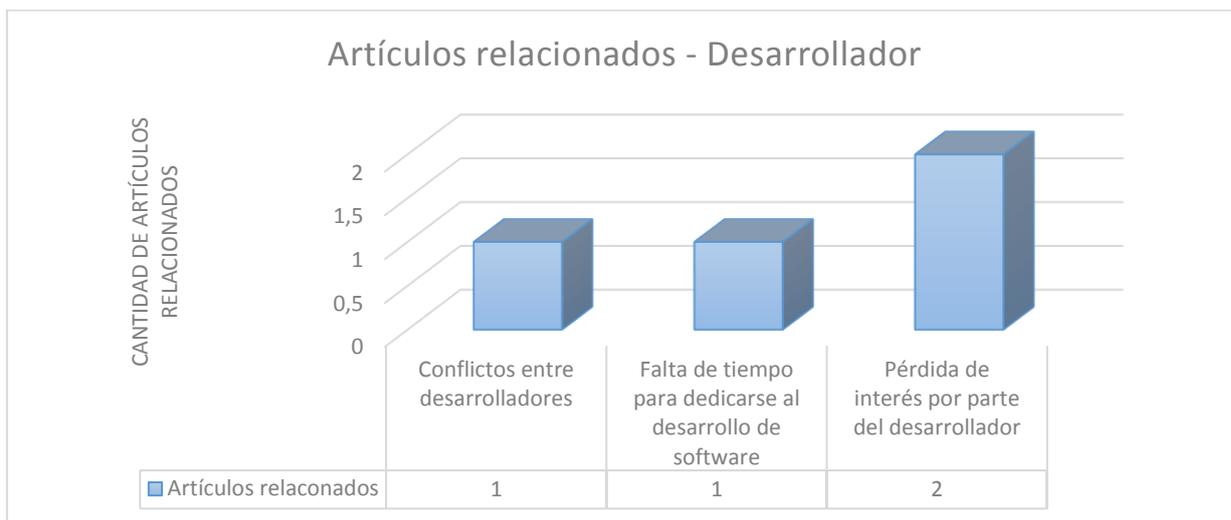


Figura 5.21 Artículos que hacen mención a categoría “*Factor Humano*” y en la subcategoría “*Desarrollador*” (Autoría propia, 2017).

En la categoría “*Factor Humano*”: subcategoría “*Instructor-Capacitador*”, 1 artículo encontrado menciona que un factor de fracaso es la “*Insatisfacción por parte de los instructores con la inclusión del FOSS*”.

En la categoría “*Factor Humano*”: subcategoría “*Técnicos internos*”, indica 1 artículo que menciona como factores de fracaso a la “*Falta de conocimientos técnicos internos en OSS*” y la “*Falta de experiencia*”, este se relaciona con la experiencia como organización.

Dentro de la categoría “*Factor Humano*”: subcategoría “*Proveedor/Vendedor*”, se ha encontrado 1 artículo, que da a conocer que entre los factores de fracaso están: la “*Difusión de incertidumbre, miedo y duda que infunden los vendedores propietarios*” interrumpiendo el avance de los proyectos de código abierto (Sahraoui, 2009), y la “*Pérdida del patrocinador que impulsa el proyecto*”.

En la categoría “*Factor Humano*”: subcategoría “*Usuarios finales*”, de los 2 artículos encontrados que hacen referencia a la subcategoría: en 1 artículo se menciona que un factor de fracaso es la “*Falta de investigación por parte del usuario*”, y en otro artículo se consideran que es la “*Perdida de interés por parte del usuario final*”, son factores de fracaso.

- **Categoría “licencias”:** Subcategoría “*Problemas*” se ha encontrado 1 artículo que menciona que un factor de fracaso son las “*Dificultades relacionadas con las licencias del OSS*”.
- **Categoría “Planificación”:** subcategoría “*Estrategias*”, se ha encontrado 1 artículo que menciona que un factor de fracaso son las “*Prácticas comerciales sin estrategias*”.

Dentro de la categoría “*Planificación*” y subcategoría “*Mala planificación*”, se menciona 1 solo artículo y como factor de fracaso considera a la “*Mala planificación y precipitación*”.

Dentro de la categoría “*Planificación*” y subcategoría “*Requisitos*”, 1 artículo encontrado menciona que un factor de fracaso de adopción de software libre es que “*No se comprenden los requisitos por parte del desarrollador*”.

- **Categoría “Políticas”:** subcategoría “*Ausencia de políticas*”, 3 artículos mencionan que el factor de fracaso en la adopción de software libre es la “*Ausencia de políticas de OSS, TIC o adopción*”.

Dentro de la categoría “*Políticas*”: subcategoría “*Estructura de gobierno*”, 1 artículo menciona que el factor de fracaso en la adopción de software libre es la “*Ausencia de estructura de gobierno para dar seguimiento de política de FOSS amigables*”.

- **Categoría “Procesos”:** Subcategoría “*Pruebas*”, de los 2 artículos encontrados que hacen referencias a esta subcategoría: 1 artículo menciona que un factor de fracaso es la “*Inexistencia de un riguroso proceso de pruebas*” y otro señala que

son “*Proyectos obsoletos frente a nuevos sistemas operativos*”, debido a que esos proyectos discontinuados tienen características inútiles y no aplicables.

- **Categoría “*SopORTE*”:** subcategoría “*Control*”, de los 2 artículos encontrados que hacen referencias a esta subcategoría: 1 artículo indica que un factor de fracaso es “*No contar con un grupo que dé solución a errores*”; sin embargo, otro señala que un factor de fracaso es el “*Número de idiomas que tiene el proyecto*”.
- **Categoría “*Usabilidad*”:** Subcategoría “*Fácil de usar*”, 1 artículo encontrado menciona que el factor de fracaso en la adopción de software libre es la “*Falta de usabilidad*”.

5.2.2. ¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre a nivel mundial?

Producto de la lectura de los 85 artículos, se han encontrado 41 riesgos en la adopción de tecnologías de OSS, que se han agrupado en 10 categorías y 23 subcategorías. En la Figura nro. 5.22, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

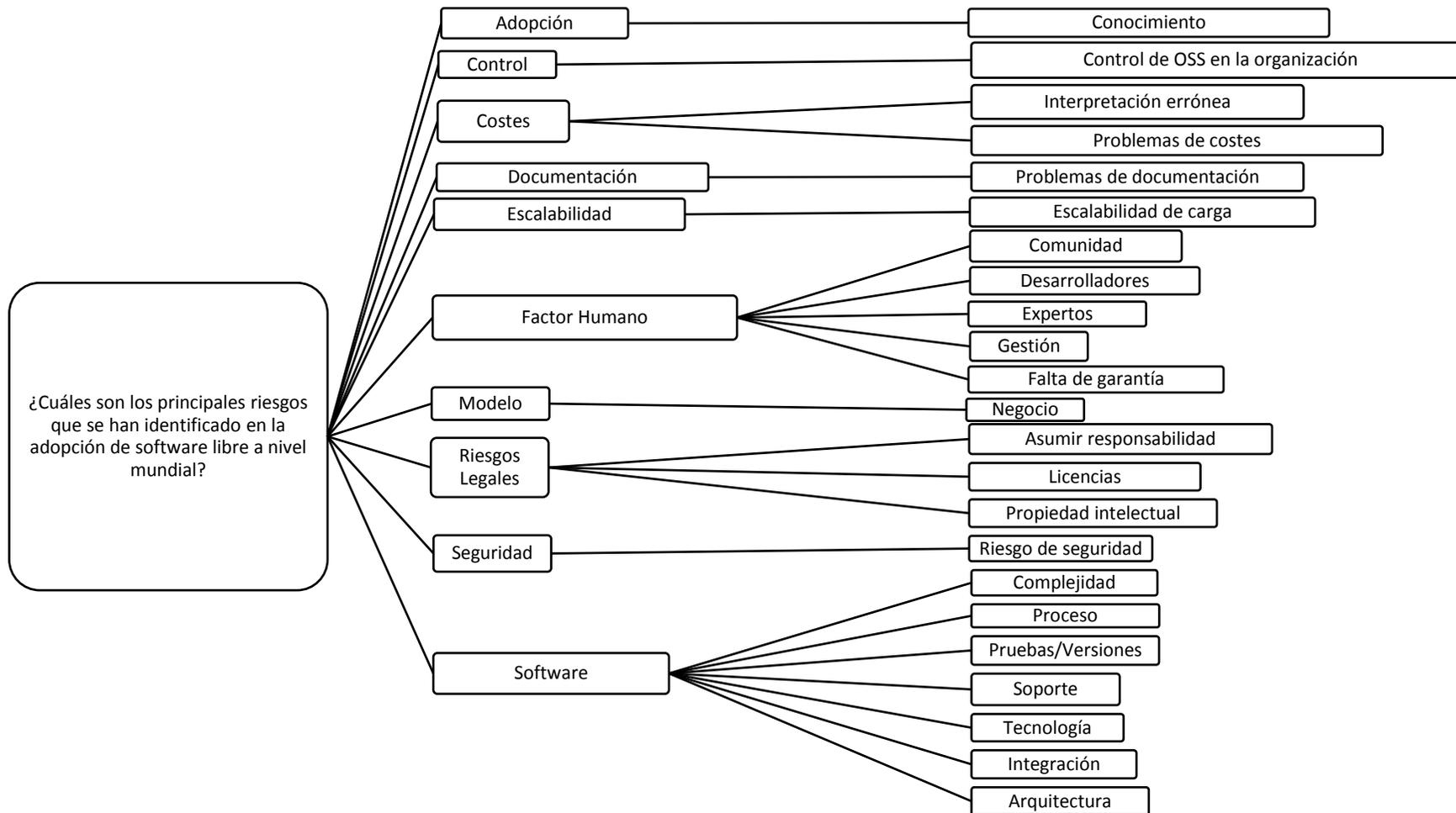


Figura 5.22 Categorías y subcategorías a los que pertenecen los riesgos identificados en la adopción de OSS (Autoría propia, 2017).

En la Figura nro. 5.23, se presenta el número de subcategorías que contiene cada categoría.

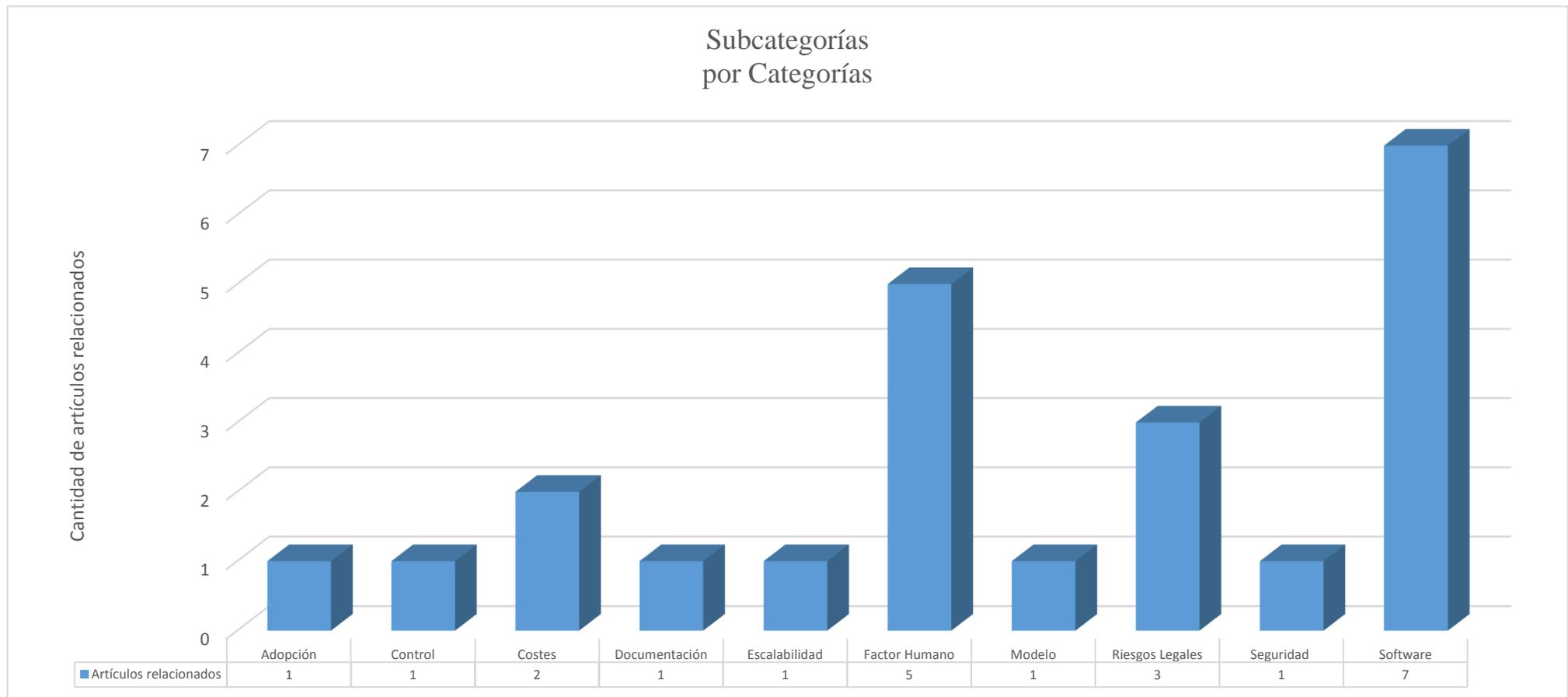


Figura 5.23 Número de subcategorías por categoría (Autoría propia, 2017).

En la Figura nro. 5.24, se presenta el número de riesgos identificados, que contiene cada subcategoría.

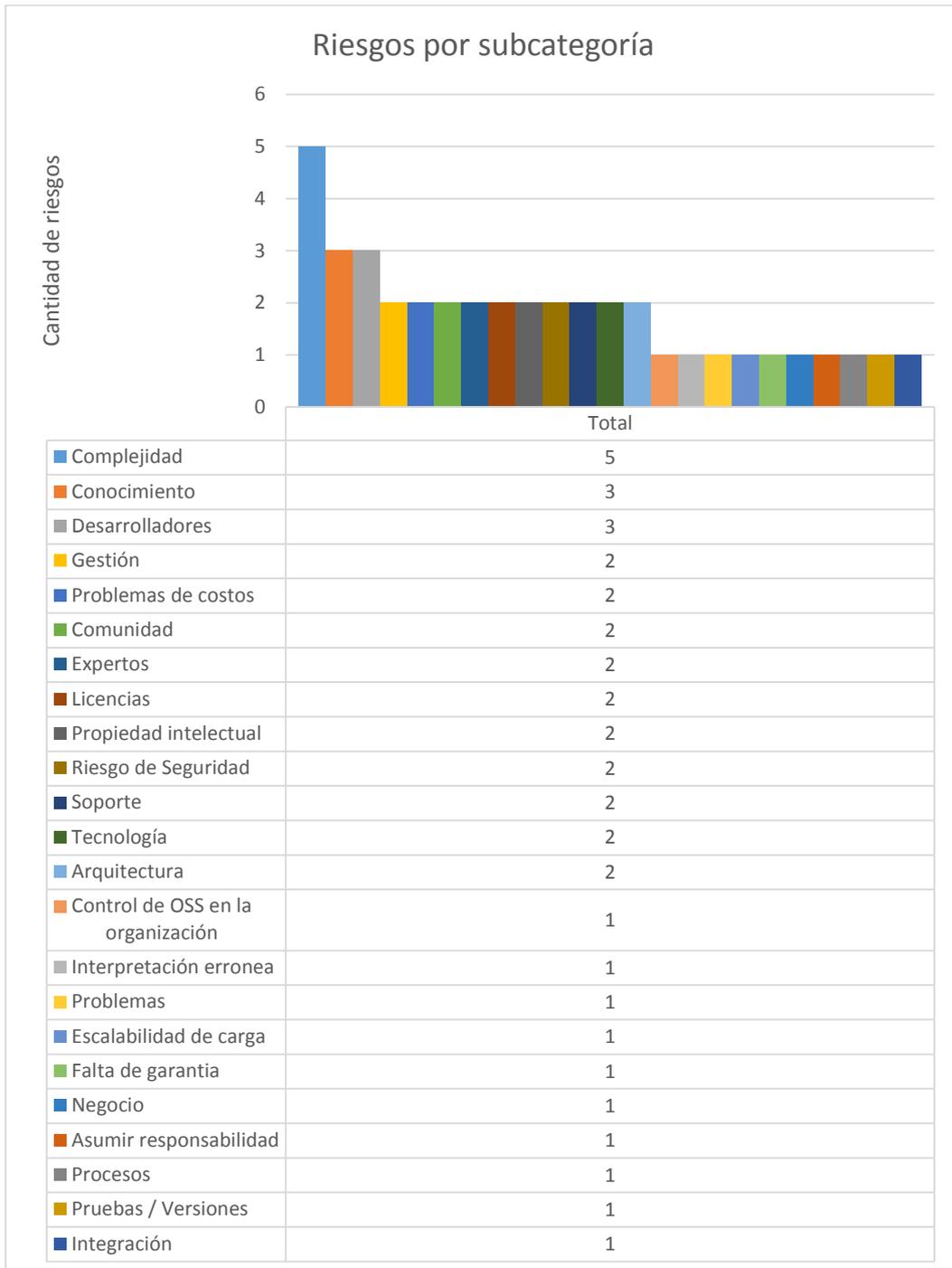


Figura 5.24 Cantidad de riesgos encontrados por subcategoría (Autoría propia, 2017).

En resumen, de los resultados obtenidos de la búsqueda de los principales riesgos identificados en la adopción de software libre, el 37% de los riesgos han sido clasificados en la categoría “Software”, un 22% en la categoría “Factor Humano”, el 12% están en la

categoría “*Riesgos Legales*”, otro 7% en la categoría “*Adopción*” y “*Costes*”, un 5% de riesgos asignados en la categoría “*Seguridad*”; mientras que el 3% pertenece a las categorías de “*Control*” y “*Documentación*”; finalmente en el 2% están las categoría, “*Escalabilidad*” y “*Modelo*”, como se presenta en la Figura nro. 5.25.

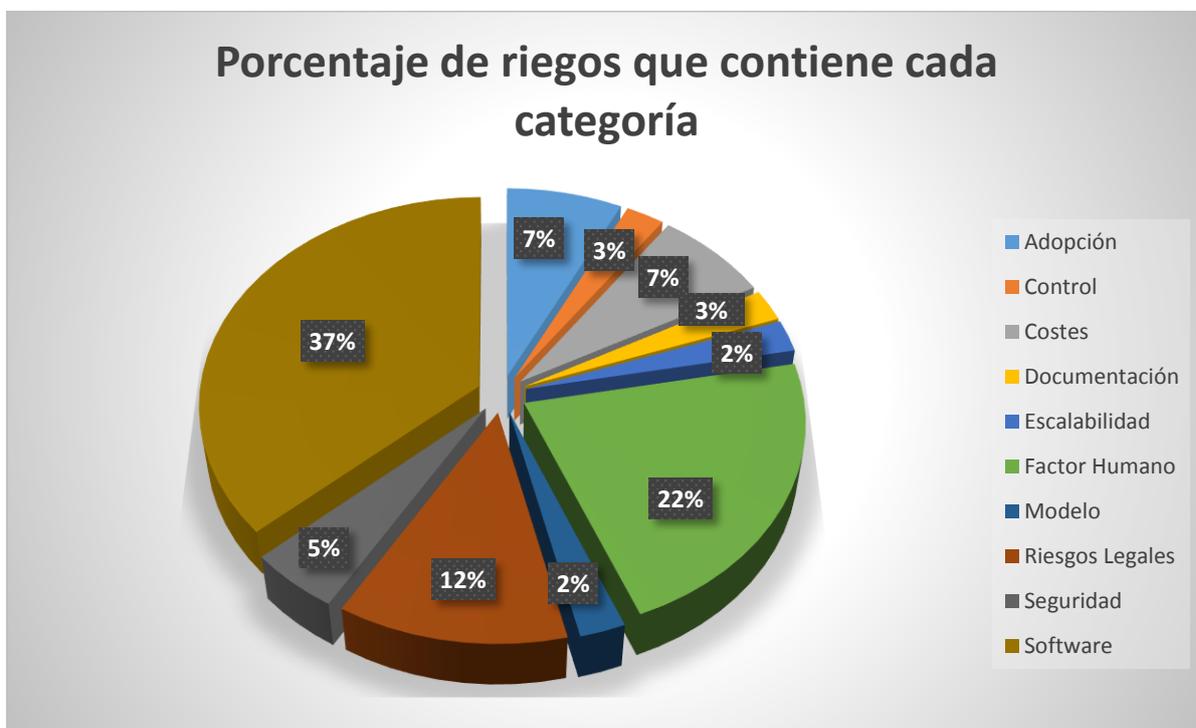


Figura 5.25 Porcentaje de riesgos que contiene cada categoría (Autoría propia, 2017).

Como se indica en secciones anteriores, producto de la lectura de los artículos seleccionados, se han identificado 41 riesgos en la adopción de software libre, que han sido mencionados por los autores de dicha literatura, los mismos que se presentan en la Tabla nro. 5.3.

Código de Riesgo	Categoría	Subcategoría	Riesgo	Frecuencia de riesgos por artículo	Porcentaje de incidencia
R1	Adopción (Adquisición de conocimiento de FOSS)	Conocimiento (Facultad para comprender aspectos relevantes de FOSS)	Adquisición de proyectos OSS por terceros	1	1%

R2			Adopción y conocimiento lento	1	1%
R3			Se necesita que el conocimiento tecnológico de OSS sea alto	1	1%
R4	Control (Función administrativa por medio de la cual se evalúa el rendimiento)	Control de FOSS en la organización (Verificar el desempeño de FOSS en la organización)	Falta de control de calidad	1	1%
R5		Interpretación errónea (Mal entendimiento de los costes de FOSS)	Pensar en el código abierto como gratuito	1	1%
R6	Costes (Gasto en que se incurre en la producción de un bien o la prestación de un servicio)	Problemas de costes (Obstáculos con valores reales o inversión de costes en FOSS)	Problemas con valores reales generados por los costes de OSS (Costes que se incurre luego de adoptar FOSS: costos ocultos, consultoría, mantenimiento).	5	7%
R7			Costos significativos en términos de inversión de recursos dentro de la empresa	1	1%
R8	Documentación (Conjunto de documentos relacionados al FOSS)	Problemas de documentación (Obstáculos referente a la documentación generada por el proyecto)	Documentación inexistente	1	1%

R9	Escalabilidad (Capacidad de adaptarse sin perder la calidad o manejar el crecimiento continuo)	Escalabilidad de carga (Hace fácil el ampliar y reducir los recursos para acomodar cargas más pesadas o más ligeras según se requiera.)	Sobrecarga de datos	1	1%
R10	Factor Humano (Comportamiento del personal desde el punto de vista de recursos humanos)	Comunidad (Grupo de individuos con un objetivo en común)	Falta de comprensión en el comportamiento y dinámica de las comunidades de OSS	1	1%
R11			Falta de seguimiento adecuados en la cantidad de mensajes que ingresan	1	1%
R12		Desarrolladores (Personas que integran soluciones de software dentro o fuera de la organización)	Cambio constante de desarrolladores es una amenaza para el proyecto	1	1%
R13			Esfuerzo sustancial para la integración	1	1%

R14			Subestimación de los esfuerzos de integración	1	1%
R15		Expertos (Persona reconocida como una fuente confiable de un tema)	Escasez de habilidades para administrar plataformas típicas OSS	1	1%
R16			Demora en la estandarización del OSS	1	1%
R17		Gestión (Conjunto de operaciones que se realizan para dirigir y administrar un negocio o una empresa)	Gestión de riesgos inadecuada	1	1%
R18			Mala publicidad por parte de la organización	1	1%
R19		Falta de garantía (Ausencia de respaldo para la implementación de FOSS)	Problemas debido a que no existen responsables definidos para el soporte de OSS	4	5%
R20	Modelo (Representación de una categoría o tipo de cosas definidas por ciertas características.)	Negocio (Forma de funcionar internamente en una empresa)	Modelo de negocio insostenible y poco fiable	1	1%
R21	Riesgos Legales (Incumplimientos con las leyes, reglas y prácticas de la utilización de FOSS)	Asumir responsabilidad (Implica aceptar los posibles riesgos legales de la adopción de FOSS)	Riesgo de asumir la responsabilidad legal (Al no tener responsables puede convertirse en un obstáculo en la parte	2	3%

			legal la adopción de FOSS)		
R22		Licencias (Autorización concedida para realizar una determinada actividad)	Infracciones por la falta de conocimiento de las licencias del FOSS (Preocupación de hacer mal uso del software FOSS por desconocimiento de estas)	5	7%
R23			Incumplimiento de uso de las licencias	3	4%
R24		Propiedad intelectual (Sistema jurídico relacionado a los derechos intangibles. Comprende todas las creaciones del intelecto humano.)	Problemas por derechos intelectuales (Obstáculos con los permisos y términos de distribución gratuita)	11	15%
R25			Falta de derechos de propiedad intelectual	1	1%
R26	Seguridad (Se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta)	Riesgo de Seguridad (Posibilidad de que algo suceda en cuestiones de seguridad de FOSS)	Falsa sensación de seguridad (Porque muchas personas revisan el código)	1	1%

R27			Vulnerabilidades de seguridad (Por uso de bibliotecas de código abierto, debilidad arquitectónica, error humano, modificación y contaminación de código)	6	8%
R28	Software (Equipo lógico que permite realizar tareas específicas)	Complejidad (Problemas de instalación, elección, interdependencia, interoperabilidad e incertidumbre por la difícil comprensión del software)	Complicaciones en la instalación	1	1%
R29			Dificultad para elegir entre opciones de software libre	2	3%
R30			Interdependencia entre componentes	1	1%
R31			Desafíos de Interoperabilidad (Por intercambios de documentos e información)	1	1%

R32			Incertidumbre. (relacionado a la reutilización de software en proyectos)	1	1%
R33		Procesos (Secuencia de pasos dispuesta con algún tipo de lógica que se enfoca en lograr algún resultado específico)	Riesgo operacional (Riesgo de pérdida resultante de procesos internos inadecuados o fallidos, personas y sistemas o de eventos externos)	1	1%
R34		Pruebas / Versiones (Diversos cambios que se realizan sobre elementos de algún producto)	Frecuencia en emisión de versiones (Problemas de actualizaciones continuas, que da la percepción de falta de estabilidad)	2	3%
R35		Soporte (Ayuda que se pueda administrar a la organización con respecto al FOSS)	Incertidumbre de apoyo	1	1%
R36			Soporte limitado.	1	1%
R37		Tecnología (Conjunto de conocimientos técnicos que satisfacen necesidades)	Tecnología obsoleta	1	1%
R38			Insuficientes aplicaciones orientadas al sector público	1	1%

R39		Integración (Conjunto de componentes que se unen para formar un todo)	Riesgo por falla de integración y despliegue de componentes (La subestimación de los riesgos de integración de componentes software es uno de los principales problemas al adoptar OSS)	2	3%
R40		Arquitectura (Representación de un sistema en la que hay una correlación de funciones con componentes de hardware y software)	Limitaciones con problemas de Arquitectura del Software	1	1%
R41			Falta de asistencia para la ingeniería de requisitos	1	1%
Totales				73	100%

Tabla 5.3 Riesgos de adopción de OSS (Autoría propia, 2017).

La Figura nro. 5.26, hace referencia a la Tabla nro. 5.3, de riesgos de adopción de OSS; en donde, en la gráfica se presenta el nombre de los factores con más de 1 incidencia y los factores con 1 incidencia denominados como “Otros”.

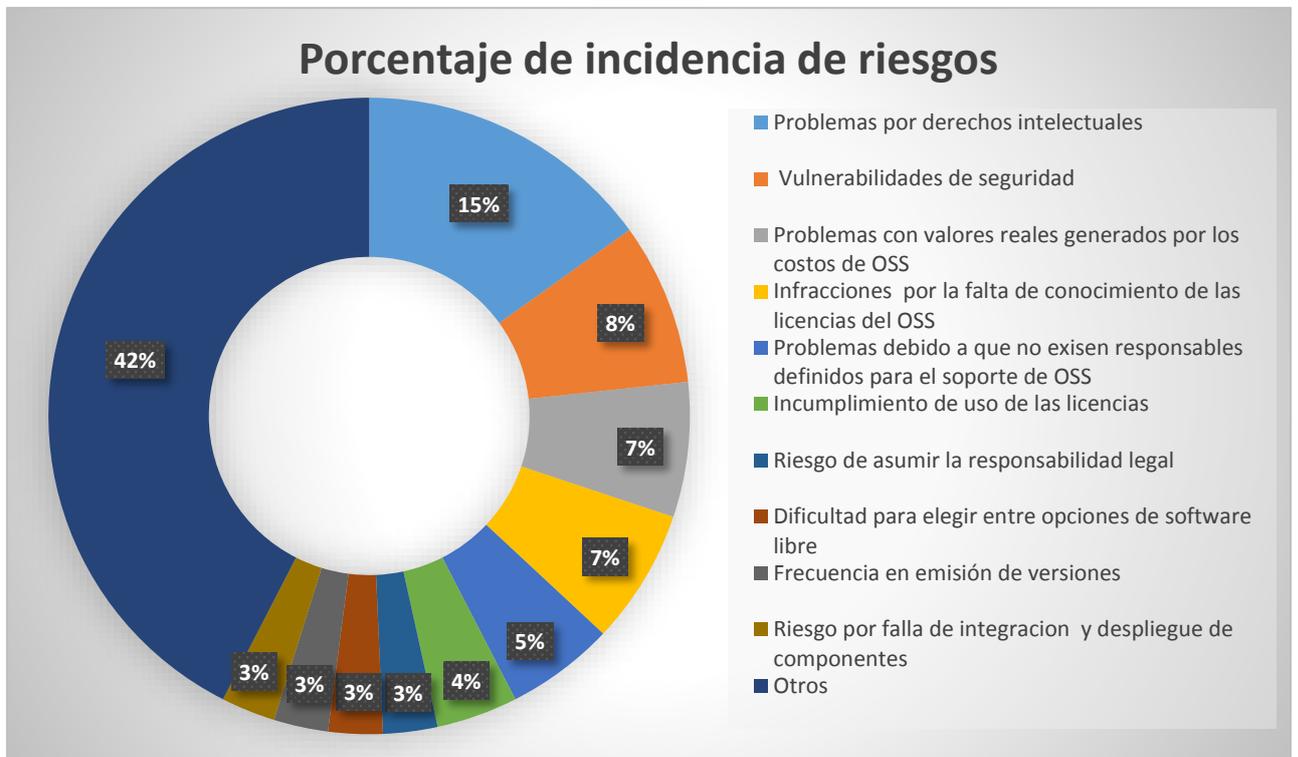


Figura 5.26 Representación gráfica de menciones de los riesgos de adopción de OSS (Autoría propia, 2017).

La información expuesta anteriormente, de acuerdo a la Tabla nro. 5.3 y a la Figura nro. 5.26, señala que el riesgo con más incidencia en la lectura de los artículos es *“Problemas por derechos intelectuales”*, con un 15% de incidencias; seguido por las *“Vulnerabilidades de seguridad”* con un 8% y los *“Problemas con valores reales generados por los costes de OSS”* con un y las *“Infracciones por la falta de conocimiento de las licencias del OSS”*, con un 7%, *“Problemas debido a que no existen responsables definidos para el soporte de OSS”* con un 5%, *“Incumplimiento de uso de las licencias”* con 4%, y con un 3% de incidencias se encuentran los siguientes: *“Riesgo de asumir la responsabilidad legal”*, *“Dificultad para elegir entre opciones de software libre”*, *“Frecuencia en emisión de versiones”*, *“Riesgo por falla de integración y despliegue de componentes”*; mientras que los demás riesgos se encuentran con 1% de incidencias se los denominó *“Otros”* y suman 42% del total; estos riesgos serán descritos a continuación:

Se recalca que la información obtenida de los riesgos en la implementación de OSS, en la mayoría de los autores de los artículos, hacen referencia sobre el tema o solo se menciona y no se presenta evidencia amplia de lo que se expone en el artículo, de esta manera, se obtiene lo siguiente:

- **Categoría *“Adopción”***: Subcategoría *“Conocimiento”* de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un riesgo en la adopción del software libre es la *“Adquisición de proyectos OSS por terceros”* y otro artículo indica que la *“Adopción y conocimiento lento”* son riesgos; sin embargo, en 1 artículo se señala que un riesgo, está basado en la falta *“Conocimiento tecnológico alto de OSS”*.
- **Categoría *“Control”***: Subcategoría *“Control de OSS en la organización”*, existe 1 artículo que menciona, que un riesgo en la adopción del software libre es la *“Falta de control de calidad”*.
- **Categoría *“Costes”***: Subcategoría *“Interpretación errónea”*, existe 1 artículo que menciona, que un riesgo en la adopción del software libre es *“Pensar en el código abierto como gratuito”*.

Dentro de la categoría *“Costes”* y subcategoría *“Problema de costes”*, de los 6 artículos encontrados que hacen referencia a esta subcategoría; 5 mencionan que

un riesgo son los “*Problemas con valores reales generados por los costes del OSS*”, que pueden ir desde el costo de propiedad intelectual, capacitación, cambio a la nueva plataforma, entre otros. También otro artículo menciona como un riesgo a los “*Costos significativos en términos de inversión de recursos dentro de la empresa*”.

En la Figura nro. 5.27, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “*Costes*” y subcategoría “*Problema de costes*”.

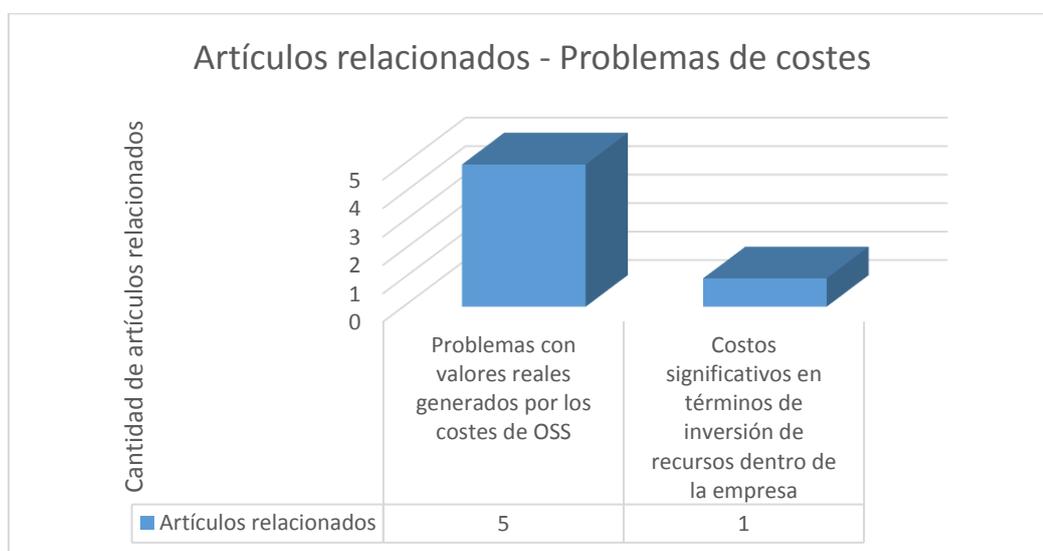


Figura 5.27 Artículos que hacen mención a la categoría “Costes” y subcategoría “Problema de costes” (Autoría propia, 2017).

- **Categoría “Documentación”:** Subcategoría “*Problemas de documentación*”, existe 1 artículo que menciona, que un riesgo en la adopción del software libre es la “*Documentación inexistente*”.
- **Categoría “Escalabilidad”:** Subcategoría “*Escalabilidad de carga*”, existe 1 artículo que menciona, que un riesgo en la adopción del software libre es la “*Sobrecarga de datos*”.
- **Categoría “Factor humano”:** Subcategoría “*Comunidad*” de los 2 artículos encontrados que hacen referencias a esta subcategoría: 1 artículo menciona que un riesgo es la “*Falta de comprensión en el comportamiento y dinámica de las*”.

comunidades del OSS” y otro la “Falta de seguimiento adecuados en la cantidad de mensajes que ingresan”.

Dentro de la categoría “*Factor humano*” y subcategoría “*Desarrolladores*” de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un riesgo es el “*Cambio constante de desarrolladores*”, debido a que este proceder es una amenaza constante para el proyecto; por otro lado 1 artículo menciona como riesgo el “*Esfuerzo sustancial para la integración*”; según Haider (2009), los productos OSS pueden necesitar esfuerzo adicional para la implementación ya que son mínimas las soluciones avanzadas de extremo a extremo. También añade que son pocos los casos en los que la integración del OSS es realizada por las organizaciones, ya que otras buscan subcontratar esa labor para evitar el esfuerzo de desarrollo. Si bien la adquisición del software libre, puede que, en algunos casos, sea gratuita, los costes podrían ir aumentando durante el transcurso de la implementación y uno de los factores que podría desencadenar el riesgo de costes, es la contratación de terceros para la integración del OSS. De igual manera dentro de los 3 artículos antes mencionados en la categoría “*Factor humano*” y subcategoría “*Desarrolladores*”, 1 artículo indica que un riesgo es la “*Subestimación de los esfuerzos para la integración*”.

Dentro de la categoría “*Factor humano*” y subcategoría “*Expertos*” 1 artículo encontrado menciona que un riesgo es la “*Escases de habilidades para administrar plataformas típicas de OSS*”, dicho riesgo encontrado por Haider (2009), explica que el producto del estudio realizado sobre la penetración de software de código abierto en Australia, está basado en las pocas habilidades y destrezas para el manejo de herramientas contrarias a Windows y demás plataformas de OSS como: Linux o PostgreSQL, además 1 artículo menciona como riesgo la “*Demora en la estandarización del OSS*”, Silic & Back (2017), señalan que *en* la implementación del proyecto de código abierto Linux en Múnich, la estandarización demoró mucho tiempo en alcanzar los mismos estándares dentro de toda la organización.

Dentro de la categoría “*Factor humano*” y subcategoría “*Gestión*”, de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un riesgo es la “*Gestión de riesgos inadecuada*”, otro artículo menciona que es la “*Mala publicidad por parte de la organización*”.

En la categoría “*Factor humano*” y subcategoría “*Falta de garantía*”, 4 artículos indican que un riesgo son los “*Problemas debido a que no existen responsables definidos para el soporte*” lo que significa que nadie va a asumir los inconvenientes que ocasione la adopción del OSS.

- **Categoría “Modelo”:** Subcategoría “*Negocio*” 1 artículo encontrado menciona que un riesgo es el “*Modelo de negocio insostenible y poco fiable*”.
- **Categoría “Riesgos Legales”:** Subcategoría “*Asumir responsabilidad*”, 2 artículos mencionan como riesgo a “*Asumir la responsabilidad legal*”, según Ebert (2007), basado en su experiencia, manifiesta que todo adoptante de OSS puede quedar expuesto legalmente, teniendo que asumir toda la responsabilidad del uso de OSS, sin capacidad de reclamo a los licenciantes, ya que muchos productos que se basan en software libre no tienen un responsable definido es decir carece de responsabilidad.

Dentro de la categoría “*Riesgos Legales*” y subcategoría “*Licencias*”, de los 8 artículos encontrados que hacen referencias a esta subcategoría: en 5 artículos se menciona que un riesgo es “*Las infracciones por falta de conocimiento de las licencias del OSS*”, según Ebert (2007), recalca que existe una variedad de licencias que no son del todo comprensibles en FOSS, porque pueden crear malentendidos en términos legales, por la ambigüedad de dichas licencias. De la misma manera dentro de los 8 artículos antes mencionados en la categoría “*Riesgos Legales*” y subcategoría “*Licencias*”, 3 indican que un riesgo es el “*Incumplimiento de uso de licencias*”, Ruffin & Ebert (2004), señalan que, si se desarrolla un producto basado en OSS para distribuirlo o venderlo; y si este no cumple con los terminos de licencias original del OSS, el productor se expone a demandas y problemas legales.

En la Figura nro. 5.28, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “*Riesgos Legales*” y subcategoría “*Licencias*”.

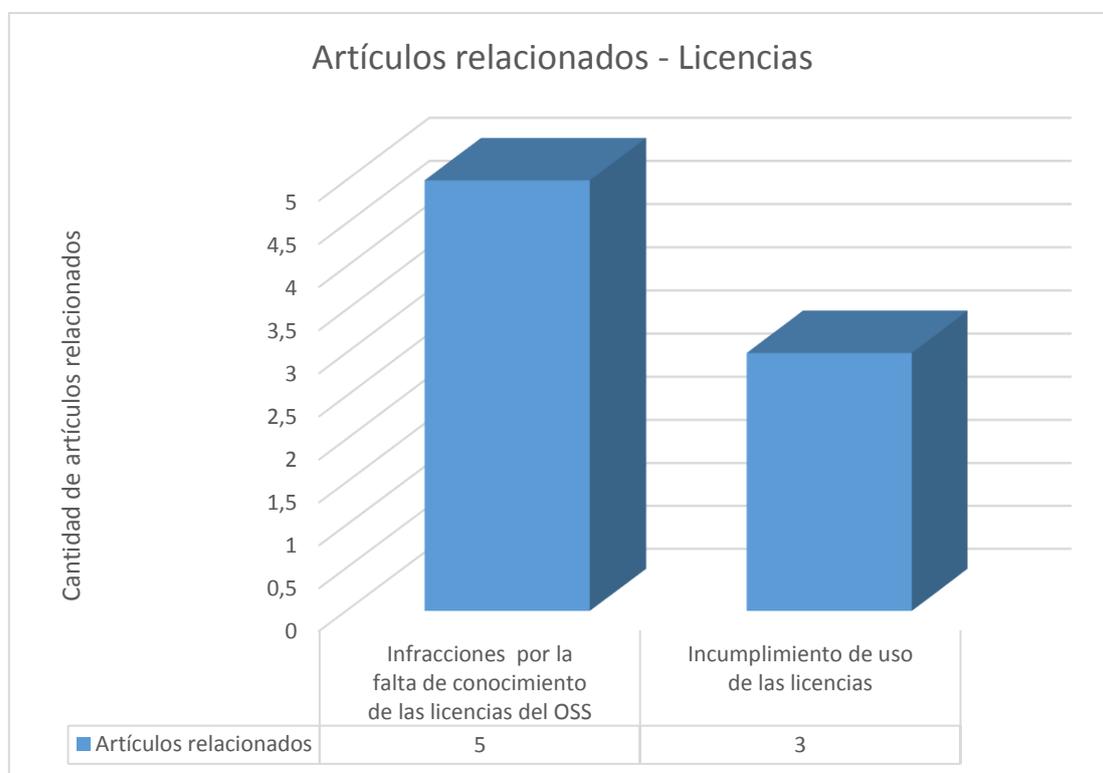


Figura 5.28 Artículos que hacen mención a la categoría “*Riesgos Legales*” y subcategoría “*Licencias*” (Autoría propia, 2017).

Dentro de la categoría “*Riesgos legales*” subcategoría “*Propiedad intelectual*”, de los 12 artículos encontrados que hacen referencias a esta subcategoría: 11 mencionan que un riesgo son los “*Problemas por derechos intelectuales*”, Bahn & Dan (2006), mencionan que uno de los escenarios de riesgos a los que están expuestas las organizaciones son los problemas de Propiedad Intelectual (PI), debido a la combinación del OSS con el software propietario. También, al uso de código que se piensa que es libre, pero que en realidad no lo es; provocando así, que la organización se vea inmersa en litigios legales por demandas inesperadas de terceros, suspensión de actividades rutinarias, entre otros; generando gastos que no están planeados. Además, los autores resaltan que los problemas de PI se dan, cuando se adopta un OSS que no garantiza en lo absoluto la protección contra demandas de PI. La idea antes expuesta, también es apoyada por Ruffin & Ebert (2004), mencionan en su artículo que está orientado a analizar los riesgos y

aspectos legales del OSS y aclaran, que el adoptante de OSS puede exponerse a problemas de derechos de propiedad intelectual (DPI), si no esta seguro de que el OSS esta libre de dichos derechos, debido a que el OSS contiene una recopilacion de muchas fuentes, lo que complica darse cuenta si en alguna parte se esta infringiendo algun DPI. Así mismo de los 10 artículos ya mencionados, 1 artículo indica que a más de la “*Pérdida de derechos intelectuales*” como riesgo, la “*Falta de derechos de propiedad intelectual*”, sería otro riesgo común.

En la Figura nro. 5.29, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “*Riesgos legales*” subcategoría “*Propiedad intelectual*”.

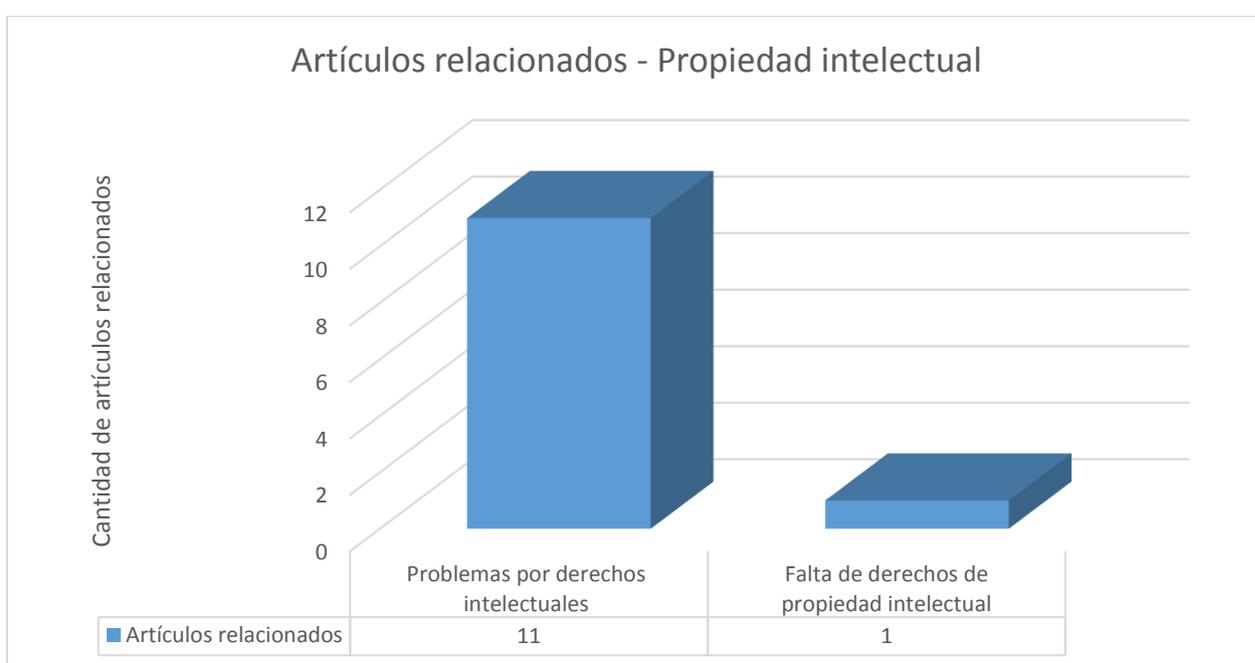


Figura 5.29 Artículos que hacen mención a categoría “Riesgos Legales” subcategoría “Propiedad intelectual” (Autoría propia, 2017).

- **Categoría “Seguridad”:** Subcategoría “*Riesgo de seguridad*”, 6 artículos mencionan , que un riesgo son las “*Vulnerabilidades de seguridad (por uso de bibliotecas de código abierto, debilidad arquitectónica, error humano, modificación y contaminación de código*”, Según Achuthan *et al.* (2014), argumenta en su artículo orientado al “Análisis de vulnerabilidades”, que las bibliotecas de código abierto usadas de terceros son más propensas a ataques maliciosos, teniendo pérdidas considerables en la información e inclusive en el

control total, otro artículo menciona el riesgo de tener “*Falsa sensación de seguridad*” en el software libre, debido a que muchos revisan el código.

En la Figura nro. 5.30, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “*Seguridad*” y subcategoría “*Riesgo de seguridad*”.

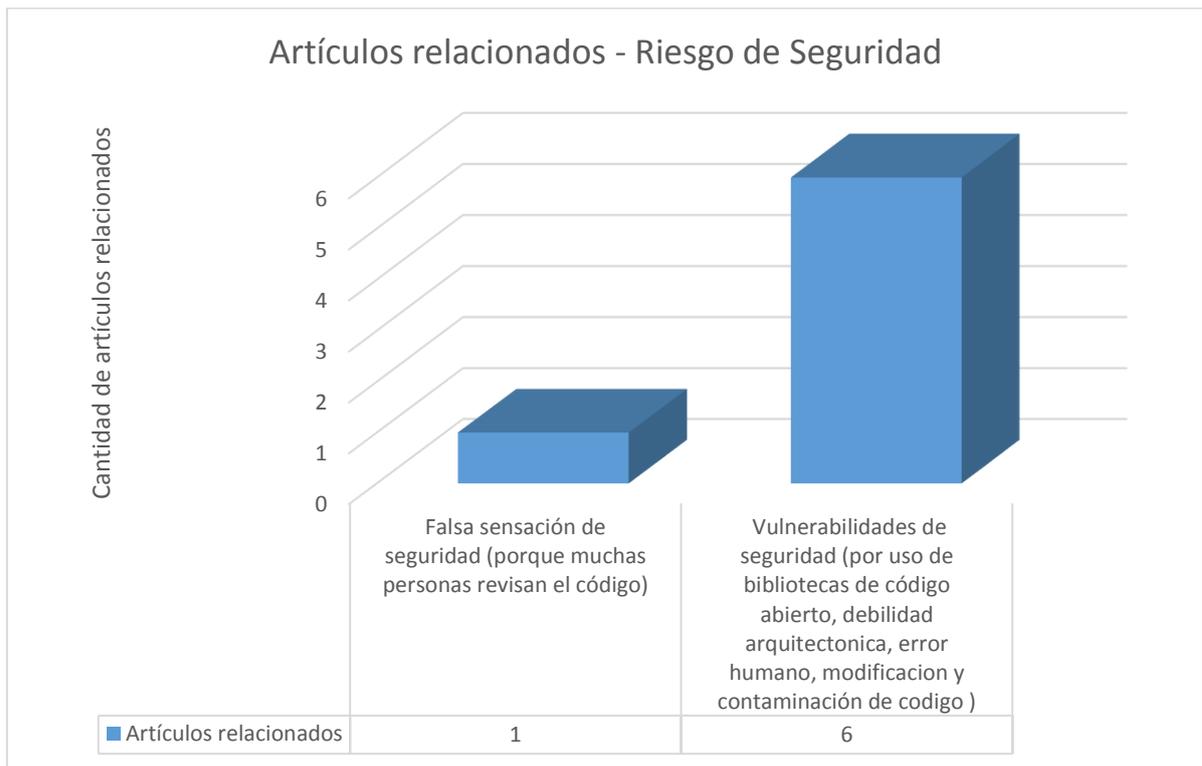


Figura 5.30 Artículos que hacen mención a categoría “*Seguridad*” y subcategoría “*Riesgo de seguridad*” (Autoría propia, 2017).

- **Categoría “*Software*”:** Subcategoría “*Complejidad*”, de los 6 artículos encontrados que hacen referencias a esta subcategoría: 1 artículo menciona que un riesgo son las “*Complicaciones en la instalación*”; otros 2 artículos indican que es la “*Dificultad para elegir entre opciones de software libre*”, y otros artículos con 1 sola incidencia, indican que otros riesgos son; la “*Interdependencia entre componentes*”, los “*Desafíos de Interoperabilidad (por intercambios de documentos e información)*”, y la “*Incertidumbre (relacionado a la reutilización de software en proyectos)*”.

En la Figura nro. 5.31, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “Software” y subcategoría “Complejidad”.

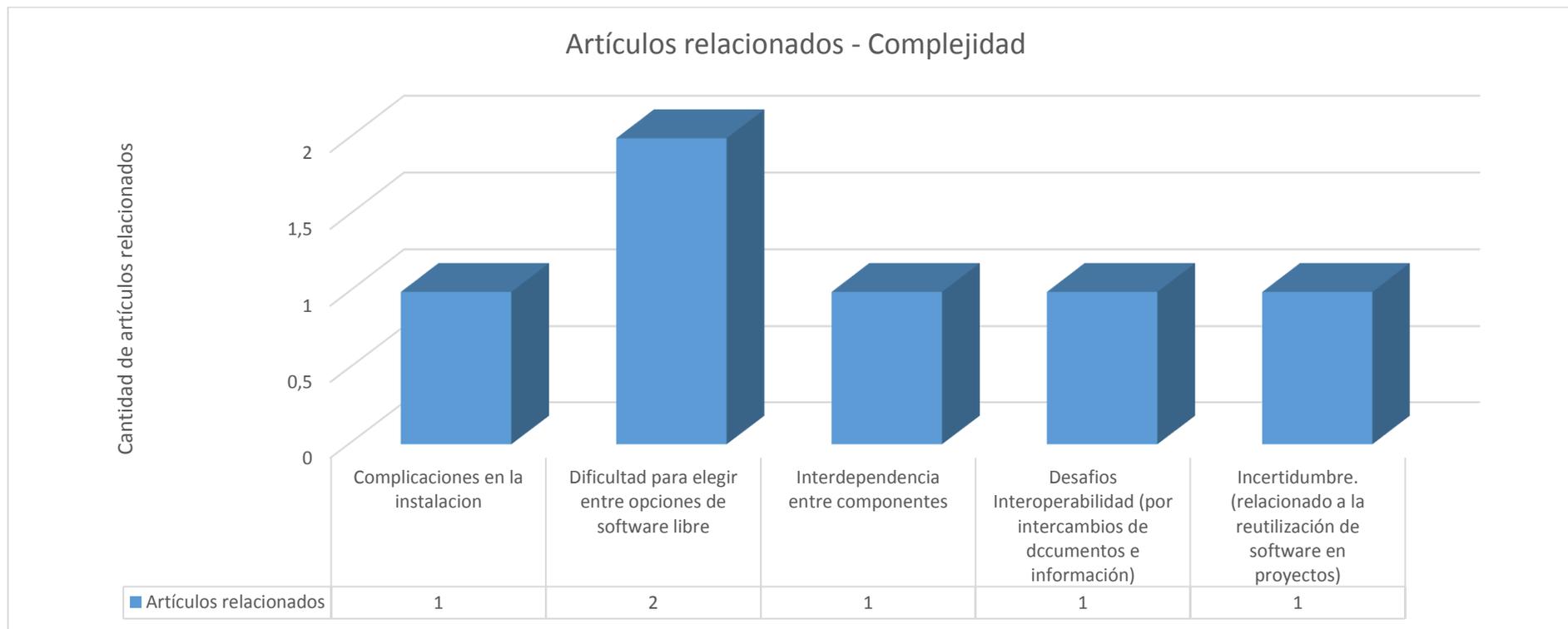


Figura 5.31 Artículos que hacen mención a categoría “Software” y subcategoría “Complejidad” (Autoría propia, 2017).

En la categoría “*Software*” y subcategoría “*Procesos*”, 1 artículo menciona, que el “*Riesgo operacional (riesgo de pérdida resultante de procesos internos inadecuados o fallidos, personas y sistemas o de eventos externos)*”, es un dominio de riesgo de adopción de software libre.

En la categoría “*Software*” y subcategoría “*Pruebas/Versiones*”, 2 artículos encontrados mencionan que un riesgo es la “*Frecuencia de emisión de versiones*”, según Haider (2009), recalca que casi todas las soluciones de OSS están generando nuevas versiones en el transcurso del tiempo, y esto demanda que el usuario o desarrollador esté preparado para usar las nuevas tecnologías. Para dicha capacitación el usuario final deberá disponer de tiempo.

En la categoría “*Software*” y subcategoría “*Soporte*”, de los 2 artículos encontrados que hacen referencia a la subcategoría: 1 artículo menciona que un riesgo es “*La incertidumbre del apoyo*”, según Haider (2009), indica que el adoptante de soluciones OSS podrá requerir la asistencia o soporte de personas que desarrollaron la solución; sin embargo, las respuestas a esas necesidades podrían ser prolongadas. Otro artículo señala que un riesgo es el “*Soporte limitado*”.

Dentro la categoría “*Software*” y subcategoría “*Tecnología*”, de los 2 artículos encontrados que hacen referencia a la subcategoría: 1 artículo menciona que un riesgo es “*La tecnología obsoleta*”. Otro señala a las “*Insuficientes aplicaciones orientadas al sector público*”, Haider (2009), señala que al no existir suficiente aplicativos para el sector público este se convierte en un riesgo para el usuario adoptante, debido a la falta de apoyo financiero y operativo que este conlleva.

Dentro la categoría “*Software*” y subcategoría “*Integración*”, 2 artículos encontrados indican que un riesgo en la adopción del software libre es el “*Riesgo por falla de integración y despliegue de componentes*”.

Dentro la categoría “*Software*” y subcategoría “*Arquitectura*”, de los 2 artículos encontrados que hacen referencias a esta subcategoría: 1 artículo menciona que un riesgo son las “*Limitaciones con problemas de arquitectura de software*” y otro señala la “*Falta de existencia para la ingeniería de requisitos*”.

5.2.3. ¿Cuáles son los beneficios de la utilización de software libre?

De la lectura de los 85 artículos, 57 artículos mencionan beneficios de la utilización de software libre, de estos artículos se han extraído 60 beneficios, los cuales se han agrupado en 13 categorías y 23 subcategorías. En la Figura nro. 5.32, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación planteada.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

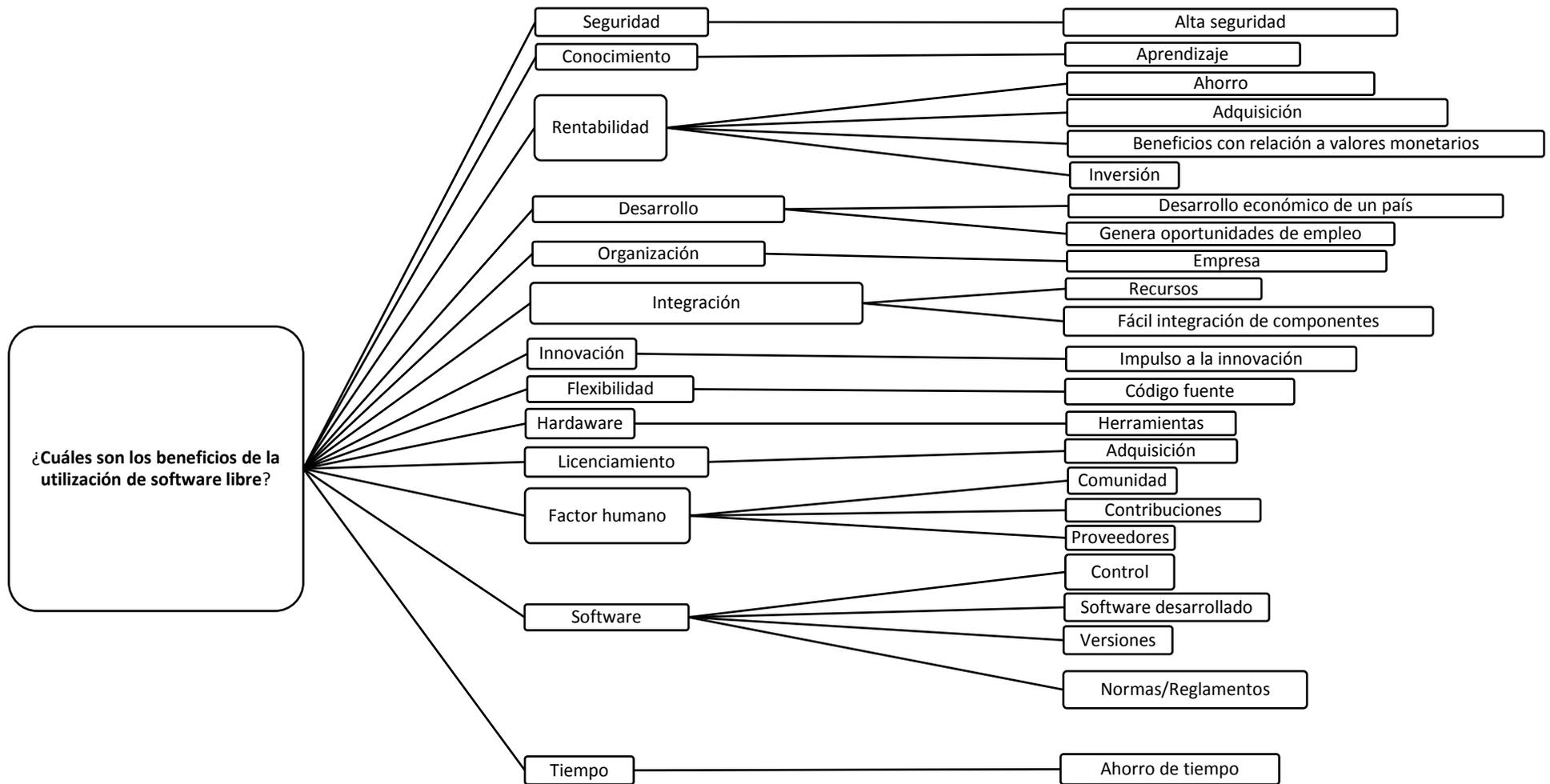


Figura 5.32 Categorías y subcategorías de los beneficios de adoptar software libre (Autoría propia, 2017).

En la Figura nro. 5.33, se presenta el número de subcategorías que contiene cada categoría.

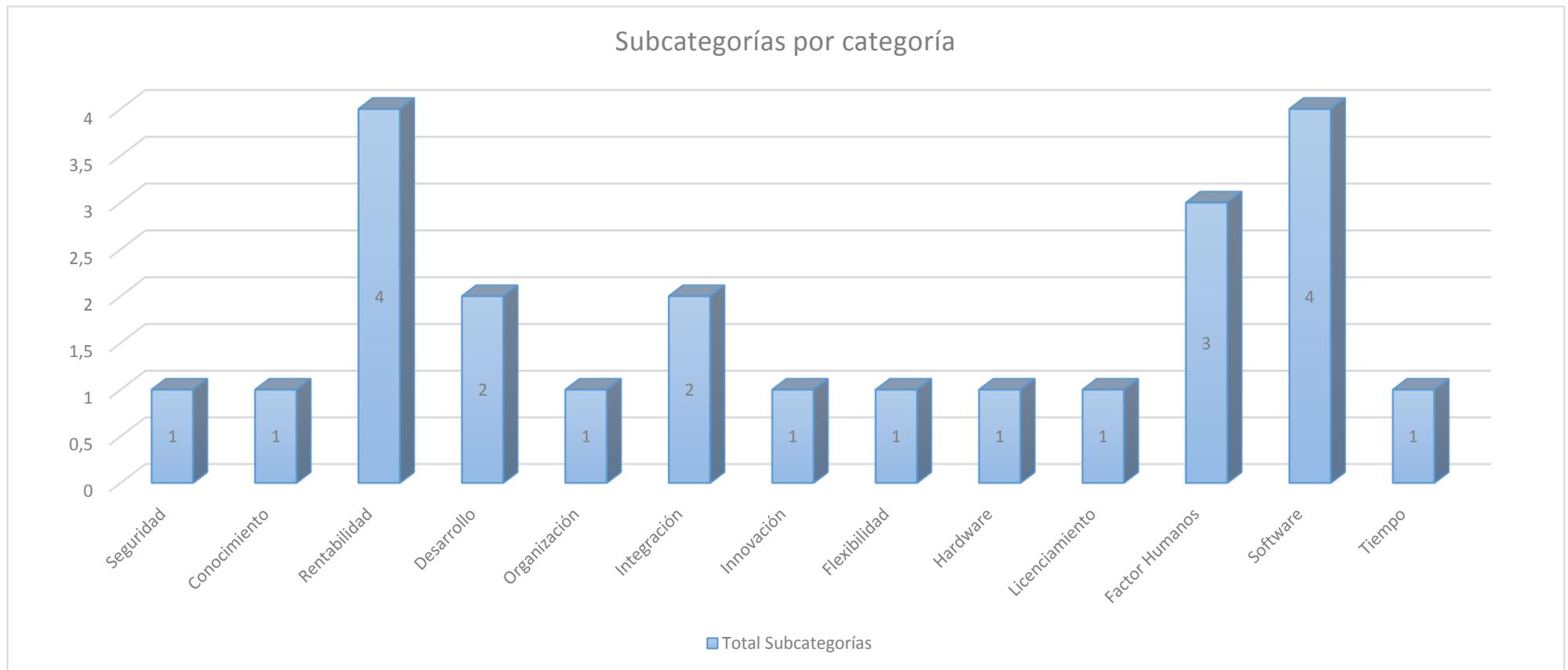


Figura 5.33 Número de subcategorías por categoría (Autoría propia, 2017).

En la Figura nro. 5.34, se presenta el número de beneficios encontrados que contiene cada subcategoría.

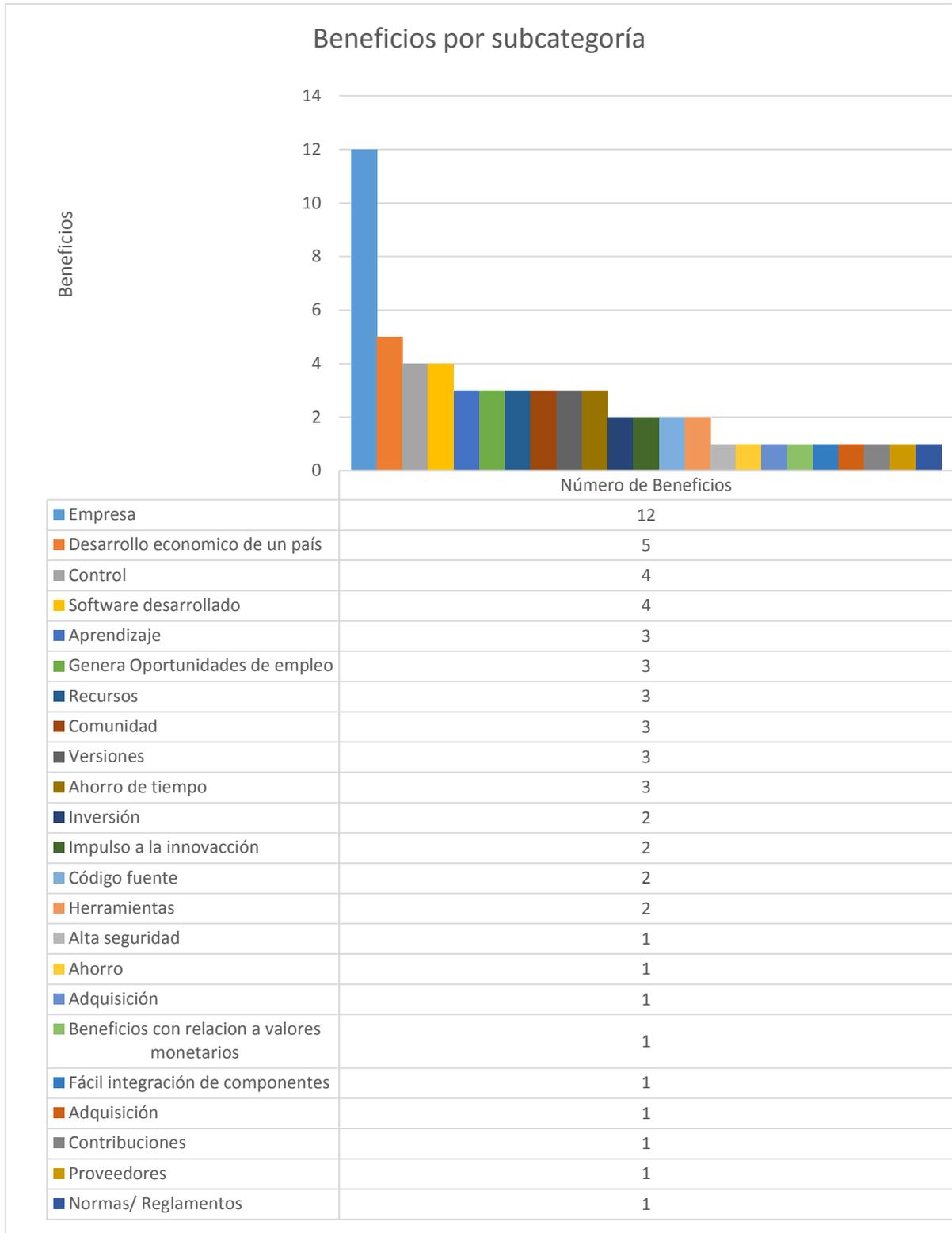


Figura 5.34 Cantidad de beneficios encontrados por subcategoría (Autoría propia, 2017).

En resumen, de los resultados obtenidos de la búsqueda de los beneficios de la utilización de software libre, el 20% de beneficios han sido clasificados en las categorías “Organización” y “Software”, un 13% en la categoría “Desarrollo”, con el 9%, se encuentran las categorías de “Rentabilidad”, con el 8% está la categoría “Factor Humano”, con un 7% la categoría “Integración”; mientras que el 5% pertenece a las categorías de “Conocimiento” y “Tiempo”, con un 3% están en las categorías “Innovación”, “Flexibilidad” y “Hardware” y finalmente con un 2% se encuentra en las categorías de “Seguridad” y “Licenciamiento”, como se presenta en la Figura nro. 5.35.

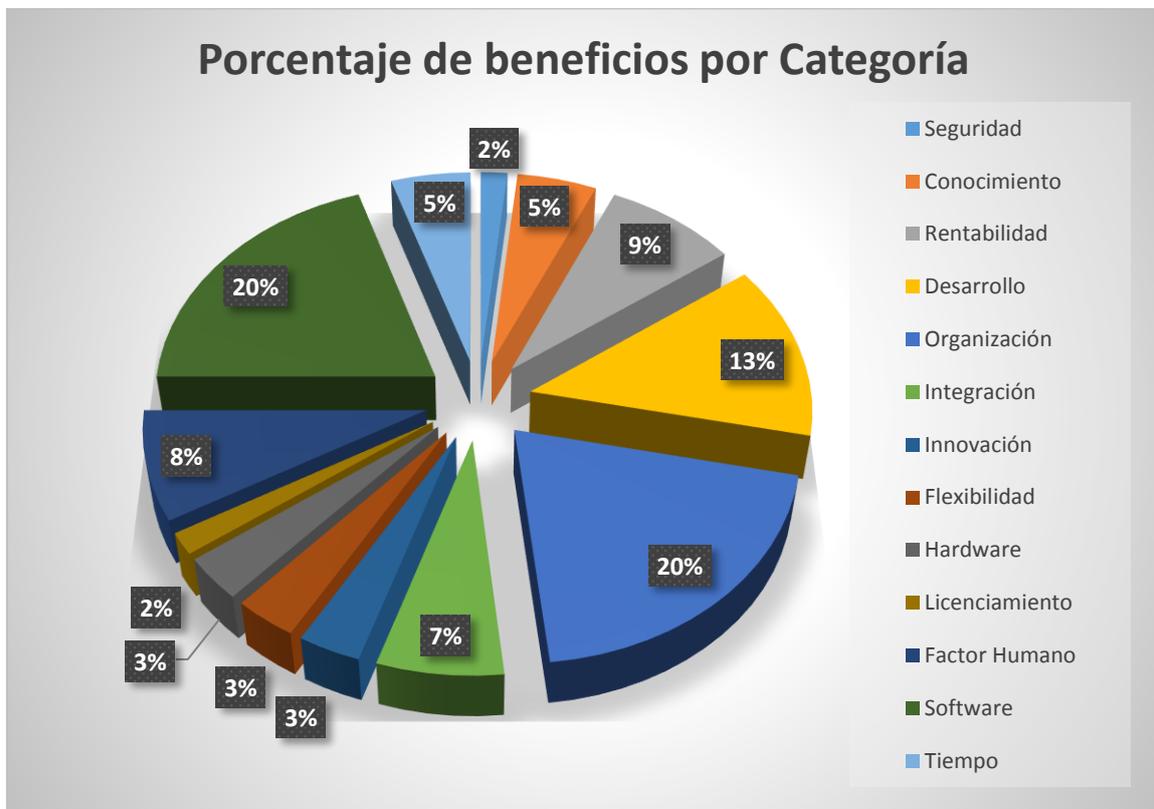


Figura 5.35 Porcentaje de beneficios que contiene cada categoría (Autoría propia, 2017).

Como se ha indicado anteriormente producto de la lectura de los artículos seleccionados, se ha encontrado 60 beneficios de adopción de OSS que han sido mencionados por los autores de dicha literatura, los mismos que se presentan en la Tabla nro. 5.4.

Número de beneficio	Categoría	Subcategoría	Beneficios	Frecuencia de beneficios por artículo	Porcentaje de incidencia
B1	Seguridad (Se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta)	Alta seguridad (Seguridad fuerte que ofrece FOSS)	Seguridad que ofrece el OSS (Indican que los desarrollos de FOSS maduros proporcionan mucha más seguridad que las opciones propietarias)	12	5,4%
B2	Conocimiento (Conjunto de información almacenada mediante la experiencia o el aprendizaje)	Aprendizaje (Adquisición del conocimiento de algo por medio del estudio)	Accesibilidad al conocimiento. (El código abierto mejora el acceso al conocimiento)	2	0,9%
B3			Motivación a la comunidad con nuevos desafíos	1	0,5%
B4			Desmitificación del proceso de desarrollo de software propietario.	1	0,5%
B5	Rentabilidad (Capacidad de producir o generar un beneficio adicional sobre la inversión o esfuerzo realizado)	Ahorro (Parte del ingreso que no se destina al gasto y que se reserva para necesidades futuras)	Ahorro económico (Ahorro que produce la adopción de FOSS por independencia de proveedores, licencias, mejoras de la comunidad)	17	7.7%
B6		Adquisición (Obtener o comprar bienes y servicios)	Bajo o sin costo de adquisición (Los precios de adquisición son nulos o muy bajos)	13	5,9%
B7		Beneficios con relación a valores monetarios (Ingresos monetarios que producen la adopción de FOSS)	Mejora las tasas de adopción de software libre	1	0,5%
B8		Inversión (Utilizar recursos con el objetivo de alcanzar algún beneficio económico o social)	No produce gastos de inversión (Indican que los paquetes de OSS son libres de costos a diferencias de los comerciales)	4	1,8%
B9			Produce Menos costes ocultos	1	0,5%
B10	Desarrollo	Desarrollo económico de un país (Capacidad de un país para mantener la prosperidad y bienestar económico)	Abre nuevas fronteras comerciales.	1	0,5%
B11			Ataca el dominio de monopolio del mercado	1	0,5%
B12			Enfoque global de desarrollo	1	0,5%
B13			Mejorar la gobernabilidad	1	0,5%
B14			Fuente de desarrollo del país (La adopción de FOSS genera recursos humanos con alta calidad)	6	2,7%

			y destrezas para el bien del país)		
B15		Genera oportunidades de empleo (Ayuda a la generación de oportunidades de negocio, nuevos empleos y promueve la cooperación y el intercambio)	Aumento de oportunidades de negocio en el futuro	1	0,5%
B16	Genera empleos (Indican que la adopción de FOSS, genera nuevas plazas de empleos)		4	1,8%	
B17	Promueve la cooperación y el intercambio		1	0,5%	
B18		Empresa (Unidad económica y social integrado por elementos humanos y técnicos)	Aprendizaje más estructurado / organizacional (Reúne a diversas personas con intereses similares para generar puntos de vista o ideas más heterogéneas)	2	0,9%
B19			Capacidad de elección (Proporcionar a los usuarios finales una gama más amplia de opciones de software que están continuamente bajo mejora)	2	0,9%
B20			Desarrollo de habilidades	1	0,5%
B21			Disminución de gastos de I+D (Investigación y Desarrollo)	1	0,5%
B22			Las Empresas buscan personal con experiencia en FLOSS	1	0,5%
B23			Fácil de analizar la actividad de los expertos en la comunidad	1	0,5%
B24	Organización (Grupo de personas que trabajan de forma coordinada para alcanzar sus metas)		Mejora la comunicación	1	0,5%
B25			Mejora fiabilidad (FOSS brinda gran fiabilidad de software al contar con un gran grupo de desarrolladores y al presentar ausencia de defectos que conducen a operaciones incorrectas, pérdida de información o fallos del sistema.)	4	1,8%
B26			Aumento de la interoperabilidad en términos de estructura organizacional	1	0,5%
B27			Modelo cooperativo de aprendizaje y desarrollo	1	0,5%
B28			Puede evitar el terrible desorden en la gestión de la información para las PYME	1	0,5%
B29		Respaldados por una empresa comercial o una universidad (Señalan que en la actualidad las principales organizaciones apoyan proyectos de FOSS)	2	0,9%	
B30			Fácil acceso a los recursos	1	0,5%

B31	Integración (Conjunto de componentes que se unen para formar un todo)	Recursos (Elementos que aportan algún tipo de beneficio)	Información sea altamente eficiente	1	0,5%	
B32			Tecnología bien documentada y disponible.	1	0,5%	
B33			Fácil integración de componentes (Unión de componentes sin problemas)	Facilidad de integración de componentes	4	1,8%
B34	Innovación (Modificar elementos ya existentes con el fin de mejorarlos o renovarlos)	Impulso a la innovación (Estimula a las organizaciones a innovar con la aplicación de nuevas tecnologías)	Favorece la innovación (Proporciona un mejor ambiente para estimular la innovación y el acceso al código fuente estimula la innovación)	3	1,4%	
B35			Innovación abierta (IO)	1	0,5%	
B36	Flexibilidad (Capacidad de adaptarse a distintas situaciones o necesidades)	Código fuente (Conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un programa)	Disponibilidad de código fuente (El código fuente está disponible para comunidad para trabajar en el)	22	10,0%	
B37			Personalización (Indican como beneficio el poder modificar a su conveniencia el código fuente)	24	10,9%	
B38	Hardware (Elementos físicos que constituyen un sistema informático)	Herramientas (Parte física de un sistema de informática)	Mayor selección de hardware	1	0,5%	
B39			Alarga la vida útil del hardware existente y reduce el desperdicio	1	0,5%	
B40	Licenciamiento (Autorización concedida para realizar una determinada actividad)	Adquisición (Obtener o comprar bienes y servicios)	Concesión de licencias	1	0,5%	
B41	Factor Humano (Comportamiento del personal desde el punto de vista de recursos humanos)	Comunidad (Grupo de individuos con un objetivo en común)	Colaboración directa con el usuario	1	0,5%	
B42			Comunicación directa con los desarrolladores	1	0,5%	
B43			Solución de errores (FOSS ofrece herramientas para detectar errores, también al contar con una gran comunidad que contribuye a la solución de errores)	12	5,5%	
B44			Contribuciones (Emplear ciertos recursos para conseguir un determinado fin)	Contribuciones al desarrollo de OSS (Soluciones desarrolladas por la comunidad se pueden usar por las organizaciones)	17	7,7%
B45			Proveedores (Entidad física o virtual que tiene el fin de brindar servicios)	Independencia de proveedores (La administración de los productos se las realiza sin depender de proveedores)	11	5,0%
B46	Software (Equipo lógico que permite realizar tareas específicas)	Control (Verificar el desempeño de FOSS en la organización)	Gama de opciones de software libre	1	0,5%	
B47			Alta calidad de software (FOSS ofrece productos de alta calidad y eficientes tales como licencias y esta actualizado)	4	1,8%	
B48			Bugs "errores" tienden a ser más visibles	1	0,5%	

B49			Garantizan ciclos de retroalimentación cortos	1	0,5%	
B50		Software desarrollado (Programa informático que ha sido planeado, diseñado, codificado y que ha sido lanzado al mercado con cierta madurez)	Mayor estabilidad (Estabilidad en el código fuente existente y presenta mayor estabilidad en comparación del software propietario)	3	1,4%	
B51			Mejora el rendimiento (Alto rendimiento en términos de capacidad y velocidad)	2	0,9%	
B52			Retroalimentación	1	0,5%	
B53			Reutilización de código (La adopción de FOSS facilita la utilización de código existente)	2	0,9%	
B54			Versiones (Diversos cambios que se realizan sobre elementos de algún producto)	Mayor control sobre las actualizaciones a nuevas versiones	1	0,5%
B55		Predice y mejora comportamientos futuros		1	0,5%	
B56		Se puede hacer lanzamientos de versiones frecuentes		1	0,5%	
B57		Normas/ Reglamentos (Principio que se adopta para dirigir la conducta o realización de una acción o actividad)	Adhesión a las normas	1	0,5%	
B58	Tiempo (Período de tiempo que se dispone para realizar una tarea)	Ahorro de tiempo (Evitar un gasto innecesario)	Reduce el tiempo en detectar errores (Es más fácil y rápido verificar y corregir errores)	5	2,3%	
B59				Reduce tiempo de desarrollo	1	0,5%
B60				Reduce tiempo en encontrar mejoras (Mas rápido de encontrar mejoras en configuración, instalación y parches; además al ser multiplataforma es se encuentran más rápido las mejoras)	6	2,7%
Totales				219	100%	

Tabla 5.4 Beneficios de adoptar OSS (Autoría propia, 2017).

La Figura nro. 5.36, hace referencia a la Tabla nro. 5.4, mencionada anteriormente; en donde, en la gráfica se presentan los nombres de los beneficios con más de 1 incidencia y los beneficios con 1 incidencia son considerados como “Otros”.

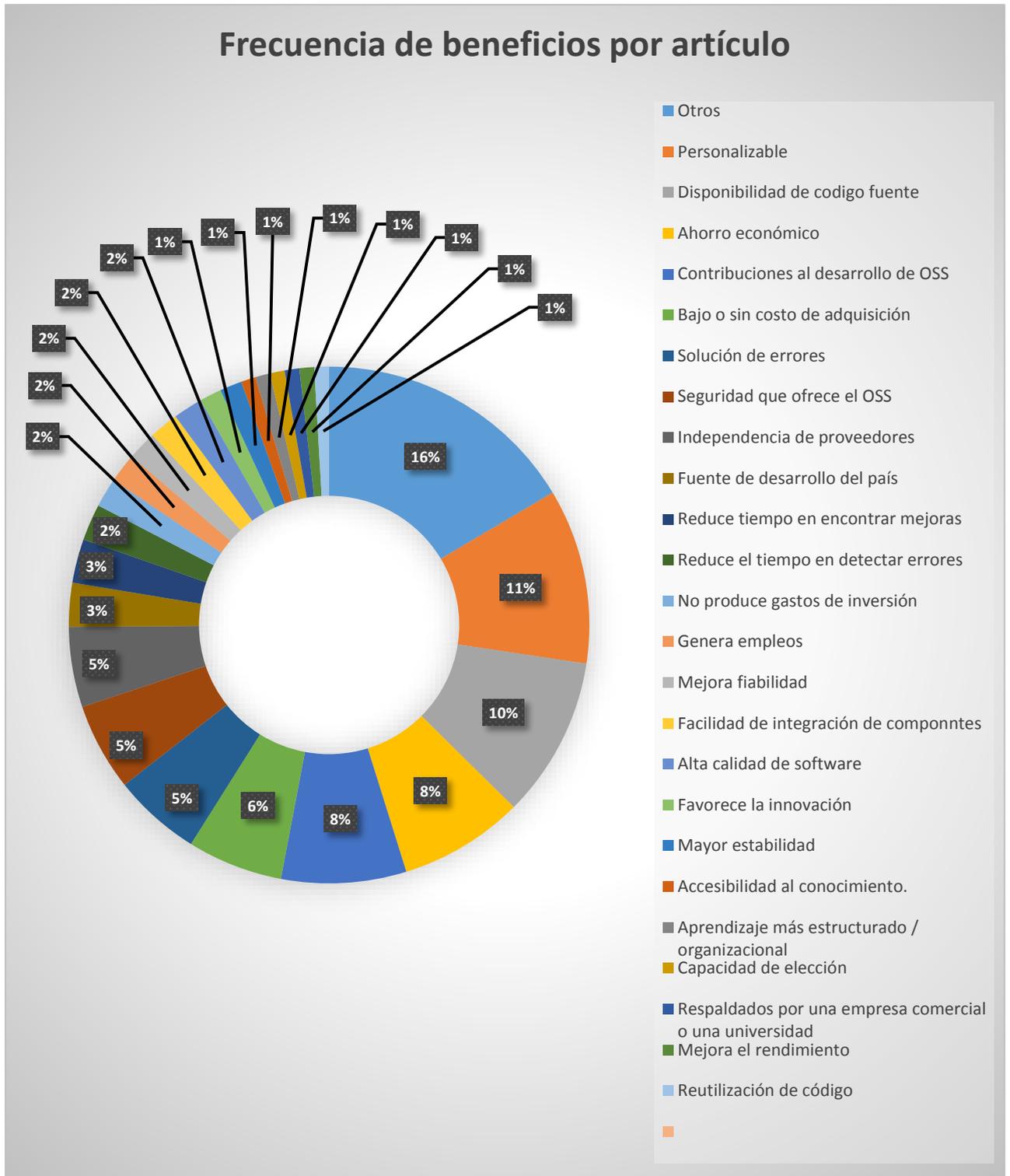


Figura 5.36 Representación gráfica de menciones de los beneficios de adopción de OSS (Autoría propia, 2017).

Partiendo de la información expuesta anteriormente en la Tabla nro. 5.4 y Figura nro. 5.36, se puede decir que el factor “*Personalizable*” es el que más se menciona con un 10.9% de incidencias; seguido por: “*disponibilidad de código fuente*” con un 10%, “*Ahorro económico*” con un 8.1%, “*Contribuciones al desarrollo de OSS*” con un 7.7%, “*Bajo o sin costo de adquisición*” con un 5.9%, “*Seguridad que ofrece el OSS*” y “*Solución de errores*” con un 5.5%, “*Independencia de proveedores*” con un 5%, “*Fuente de desarrollo del país* y, “*Reduce tiempo en encontrar mejoras*” con un 2.7%, “*Reduce el tiempo en detectar errores*” con un 2.3%; mientras que los demás beneficios se encuentran entre el 1.8% y el 0.5% de incidencias; estos beneficios serán descritos a continuación con más profundidad. Se recalca que la información obtenida de los beneficios de adopción de OSS, en la mayoría de los autores se hace una breve referencia sobre el tema, o solo se menciona y no se presenta evidencia amplia de lo que se expone en el artículo, de esta manera, se obtiene lo siguiente:

- **Categoría “*Seguridad*”:** Subcategoría “*Alta seguridad*” se encontraron 12 artículos que señalan que un beneficio de la implementación de OSS es la “*Seguridad que ofrece el OSS*”, entre los artículos que mencionan este beneficio podemos citar:
 - Ademolu Ajigini *et al.* (2014) señala, que ha crecido la confianza en software libre, debido a la examinación exhaustiva por parte de la comunidad.
 - Kovacs *et al.* (2004), indica que las violaciones de seguridad no representan un grave problema para los sistemas operativos Linux y que es mínima la probabilidad que los parches afecten a la estabilidad del sistema operativo.
 - Morgan & Finnegan (2007), señala que el código abierto entrega fuerte seguridad al tener disponible el código fuente, además disminuye el riesgo de contaminación por virus.

- **Categoría “*Conocimiento*”:** Subcategoría “*Aprendizaje*” se encontraron 3 artículos que hacen referencia a la subcategoría, de los cuales dos señalan que un beneficio de la implementación de OSS es la “*Accesibilidad al conocimiento*” que se entrega, como lo indica Khelifi *et al.* , señalando que la enseñanza partiendo del OSS ayuda en la accesibilidad del conocimiento, principalmente en los países

subdesarrollados con la adquisición de instrumentos de software educativos modernos que no sean tan costosos; este artículo también señala que un gran beneficio del uso de OSS es la “Desmitificación del proceso de desarrollo de software propietario”; mientras 1 artículo señala que el beneficio es la “Motivación a la comunidad con nuevos desafíos”.

En la Figura nro. 5.37, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Conocimiento” y subcategoría “Aprendizaje”.

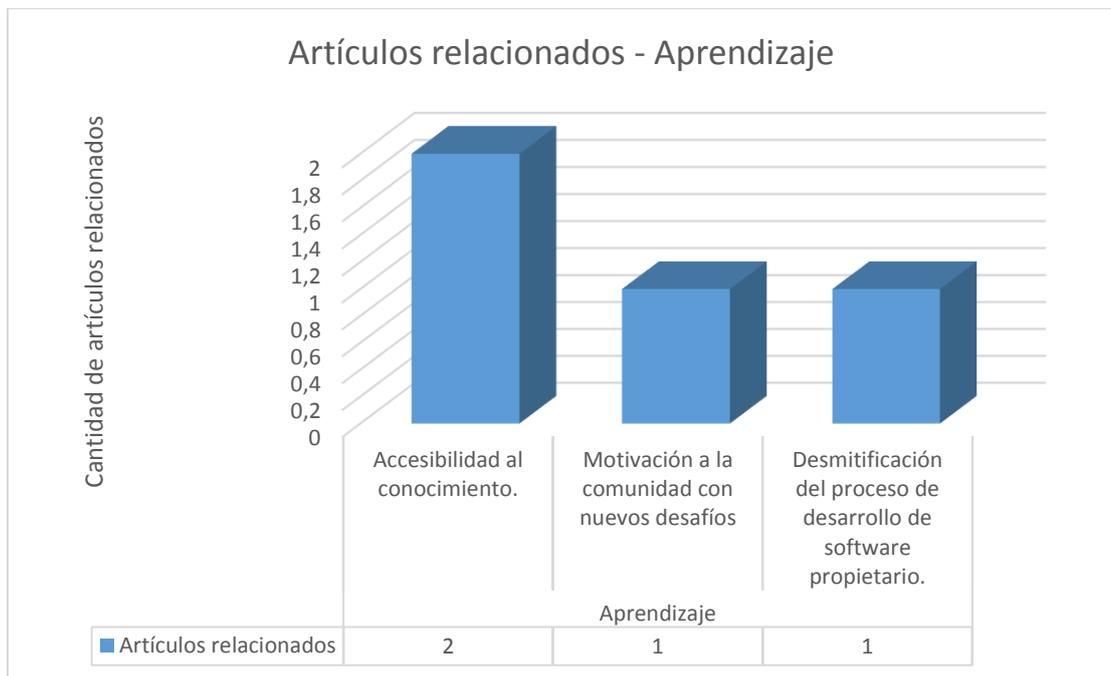


Figura 5.37 Artículos que hacen mención a la categoría “Conocimiento” y subcategoría “Aprendizaje” (Autoría propia, 2017).

- **Categoría “Rentabilidad”:** Subcategoría “Ahorro” se encontraron 17 artículos , los cuales señalan que un beneficio de la implementación de OSS es el “Ahorro económico” que produce la adopción de OSS, como lo indica Hamid *et al.* (2016), señalando que al no depender de las empresas comerciales el software de código abierto permite ahorrar costes; para Kamau & Namuye (2012), la implementación de OSS es bastante viable para reducir costes en países en desarrollo económico como en el caso de África, además se indica que China también se ha visto beneficiado por el ahorro de costes cuando se adopta OSS; también Fitzgerald & Kenny (2003), señala que las reducciones de costes al implementar OSS han sido

significativas, como en el caso del hospital Beaumont que ahorro 13 millones de euros en cinco años.

Dentro de la categoría “*Rentabilidad*”, también se encuentran las subcategoría “*Adquisición*” de la cual se encontraron 13 artículos, los cuales señalan que un beneficio de la implementación de OSS es el “*Bajo o sin costo de adquisición*” que produce la adopción de OSS, Li *et al.* (2005), señala que el precio de adquirir OSS es mínimo, además de la inexistencia de un gasto contractual debido a la naturaleza de este, como ejemplos de dominios que entregan este beneficio mencionan Linux y Apache, además Haider (2014), indica que partiendo del aspecto financiero el OSS se encuentra en el mercado con valores reducidos de adquisición e incluso sin generar gastos, gracias a esto se puede excluir los montos de licencias de propiedad, también Waringa & Maddocks (2005), señala que el mayor interés de adoptar OSS es su disponibilidad a un mínimo o ningún costo, llegando a conseguirlo incluso desde el internet libremente. También recalca que estos beneficios de costes se dan en aumento luego de la adquisición del sistema total, reduciendo los gastos en comparación con uno propietario.

Dentro de la categoría “*Rentabilidad*”, también está la subcategoría “*Beneficios con relación a valores monetarios*” de la cual se encontró 1 artículo, el cual señala que un beneficio de la implementación de OSS es la “*Mejora las tasas de adopción de software libre*”.

Dentro de la categoría “*Rentabilidad*”, también está la subcategoría “*Inversión*” en la cual se encontraron 5 artículos, de los cuales 4 artículos señalan que un beneficio de la implementación de OSS es que “*No produce gastos de inversión*” en la adopción de OSS, Nouman *et al.* (2013), indica que los paquetes de OSS son libres de costes a diferencia de las comerciales, motivo por el cual se utilizó OSS para realizar su proyecto, evitando gastos en su inversión; también 1 artículo indica que un beneficio es que “*Produce menos costes ocultos*”, como lo indica Gatta *et al* (2007).

En la Figura nro. 5.38, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Rentabilidad” y subcategoría “Inversión”.

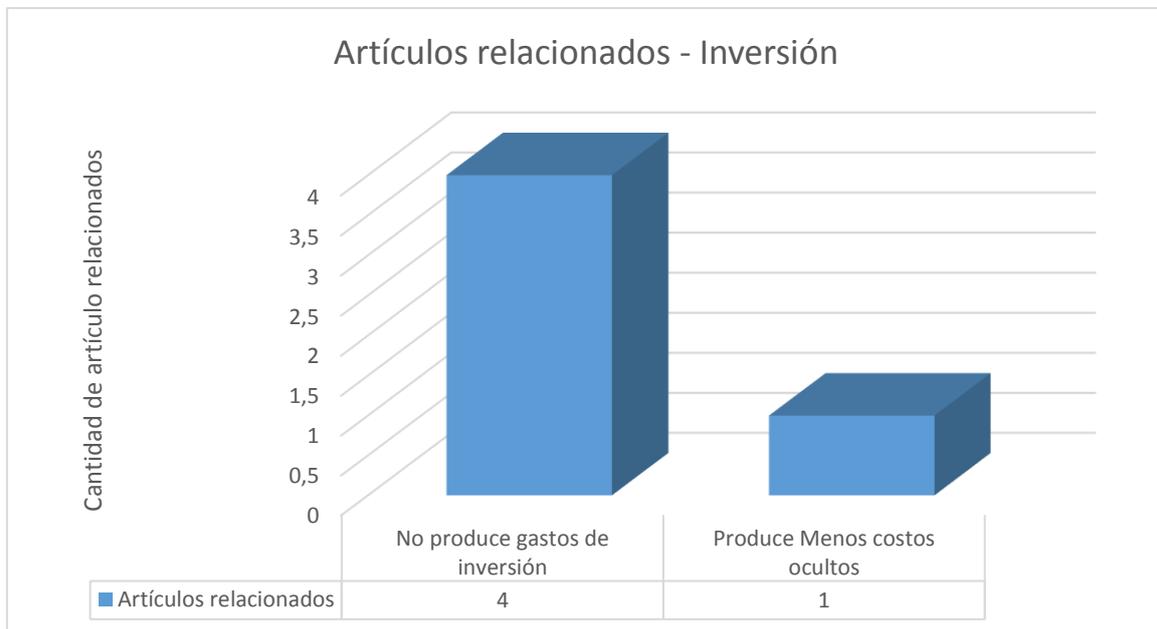


Figura 5.38 Artículos que hacen mención a la categoría “Rentabilidad” y subcategoría “Inversión” (Autoría propia, 2017).

- **Categoría “Desarrollo”:** Subcategoría “Desarrollo económico de un país” se encontraron 9 artículos que hacen referencia a la subcategoría; de los cuales 1 artículos señala que los beneficios de la implementación de OSS es que “Abre nuevas fronteras” y “Ataca el dominio de monopolio del mercado”, Rossi et al. (2012), señalan que la implementación de OSS en las organizaciones ayuda a la apertura de nuevas fronteras comerciales, también indica que uno de los beneficios del OSS es que lucha contra el monopolio que existe en el medio del software, lo cual incrementa su importancia en el sector estatal.

Dentro de esta categoría 1 artículo menciona que un beneficio de adoptar OSS, según Haider (2008), desde el panorama tecnológico es el “Enfoque global de desarrollo” que ofrece este; otro beneficio mencionado por Maldonado (2010), es que le permitirá a la administración pública “Mejorar la gobernabilidad”; otros 6 artículos mencionan que un beneficio de la adopción es que son la “Fuente de desarrollo del país”, Oreku (2010), señala que a nivel de pequeñas empresas y emprendedores en Tanzania es una gran ventaja la adopción de OSS para el

crecimiento económico del país, al igual que Kovacs *et al.* (2004), indican que la libertad de proveedores ayuda a las organizaciones estatales a encargar soporte a empresas locales, fomentando el crecimiento del país.

En la Figura nro. 5.39, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Desarrollo” y subcategoría “Desarrollo económico de un país”.

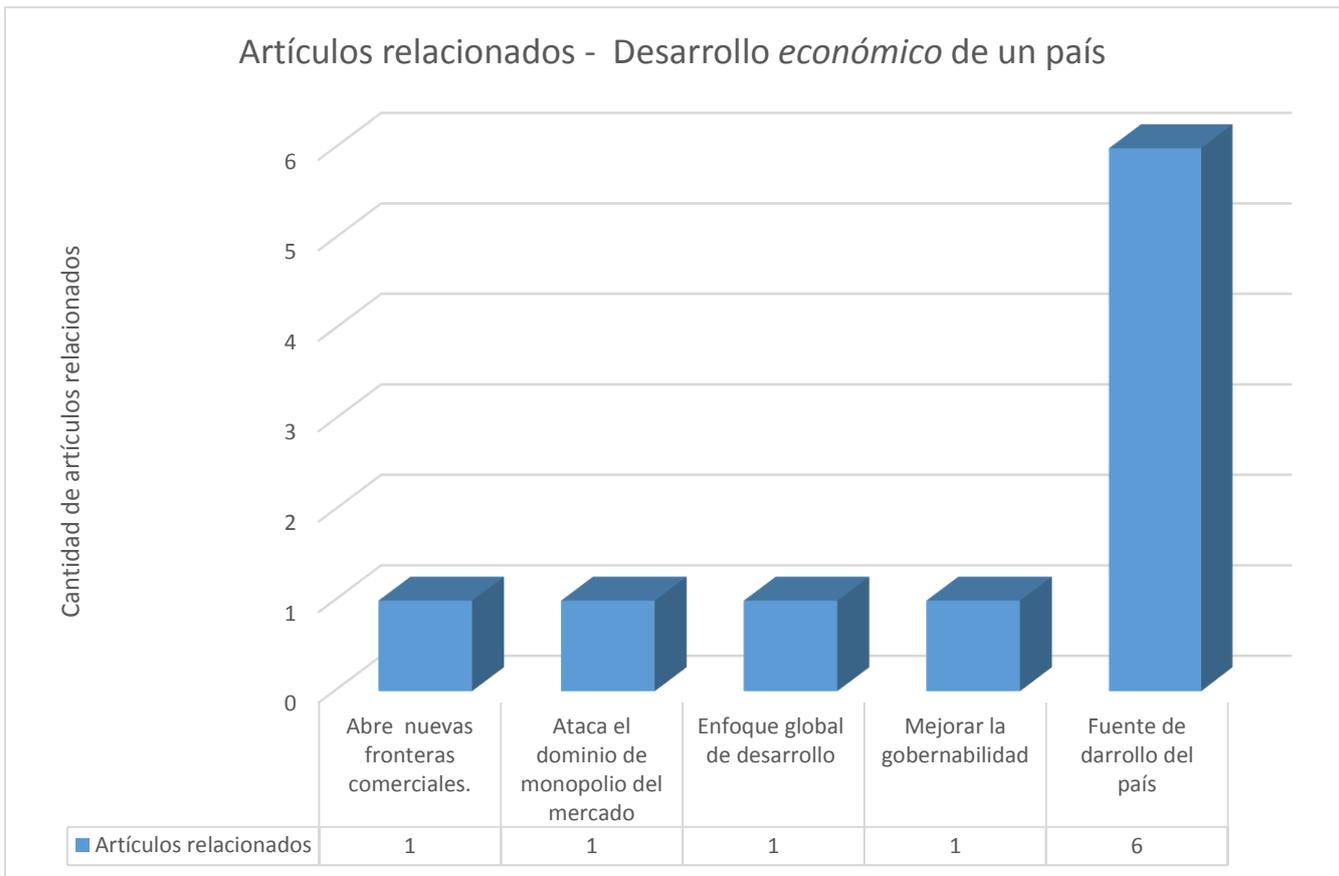


Figura 5.39 Artículos que hacen mención a la categoría “Desarrollo” y subcategoría “Desarrollo económico de un país” (Autoría propia, 2017).

Dentro de la categoría “Desarrollo”, también está la subcategoría “Genera oportunidad de empleo” de la cual se encontraron 6 artículos que hacen referencia a esta subcategoría, 1 artículo menciona que un beneficio de la implementación de OSS es el “Aumento de oportunidades de negocio en el futuro”, otros 4 artículos señalan que “Genera empleos”, como en el caso de Malasia que, con la implementación de OSS pudo ofrecer 6206 puestos de trabajo según lo indica Hamid (2016); mientras 1 artículo señala que “Promueve la cooperación y el

intercambio”, como indica Petrijevcenin Vuksanovic & Sudarevic (2012), que cuando un desarrollo es de interés mutuo entre las organizaciones se colaboran para lograr alta calidad y éxito.

En la Figura nro. 5.40, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Desarrollo” y subcategoría “Genera oportunidades de empleo”.

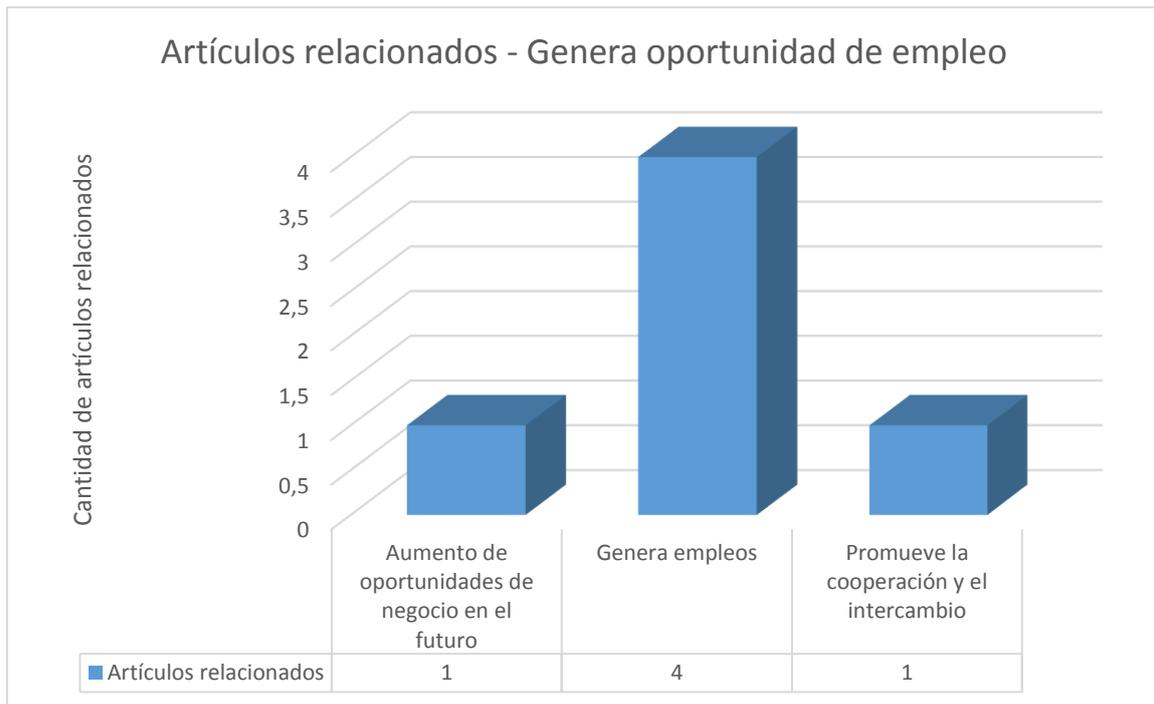


Figura 5.40 Artículos que hacen mención a la categoría “Desarrollo” y subcategoría “Genera oportunidad de empleo” (Autoría propia, 2017).

- **Categoría “Organización”:** Subcategoría “Empresa” se encontraron 13 artículos que hacen referencia a la subcategoría; de los cuales 2 artículos señalan que los beneficios de la implementación de OSS es que ayudan al “Aprendizaje más estructurado/organizacional”, así mismo uno de ellos menciona que los beneficios son: fomenta el “Desarrollo de habilidades”, “Las empresas buscan personal con experiencia en FLOOS” y ayuda al “Modelo cooperativo de aprendizaje y desarrollo”, otro artículo menciona como beneficio a la “disminución de gasto de I+D (investigación y Desarrollo), 2 artículos indican que el beneficio es la “Capacidad de elección”, en donde Waringa & Maddocks (2005), señalan que el OSS es una opción para enfrentar al código cerrado, al

presentar una gran cantidad de opciones que se encuentran en mejora continua, este artículo conjuntamente con otros 3 artículos mencionan como beneficio del OSS que *“Mejora la fiabilidad”*, Li *et al.* (2005), señalan que el OSS brinda gran fiabilidad de software al contar con un mayor grupo de desarrolladores que el software propietario, 1 artículo menciona que un beneficio es que es *“Fácil de analizar la actividad de los expertos en la comunidad”*, 1 artículo indica que *“Mejora la comunicación”*, otro artículo señala según Maldonado (2010), que el beneficio de adoptar OSS es el *“Aumento de la interoperabilidad en términos de estructura organizacional”*.

Otro artículo indica que un beneficio es que *“Puede evitar el terrible desorden en la gestión de la información para las PYME”*, como lo indica Zheng *et al.* (2008), además este artículo junto a otro mencionado anteriormente señalan que, como beneficio a los proyectos de OSS son los *“Respaldados por una empresa comercial o una universidad”*, Waringa & Maddocks (2005), señala que, aunque antes las empresas rechazaban la inversión en proyectos de OSS , en la actualidad las principales organizaciones están apoyando estos proyectos, entre las cuales tenemos a IBM, Hewlett-Packard y Oracle que apoyan a Linux.

En la Figura nro. 5.41, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Organización” y subcategoría “Empresa”.

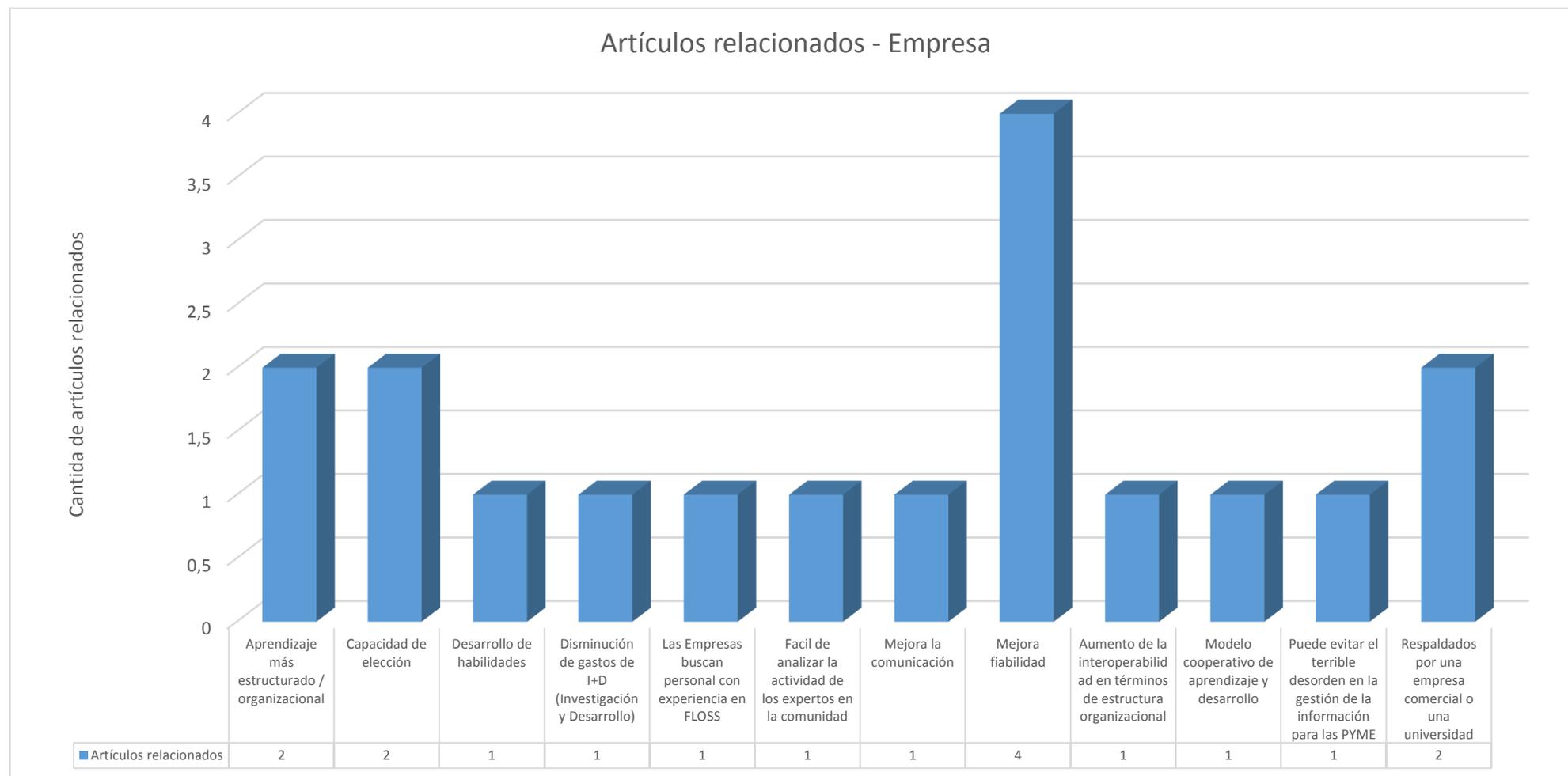


Figura 5.41 Artículos que hacen mención a la categoría “Organización” y subcategoría “Empresa” (Autoría propia, 2017).

- **Categoría “Facilita la integración de componentes”:** Subcategoría “Recursos” de la cual se encontraron 3 artículos que hacen referencia a la subcategoría; 1 artículo menciona como beneficio al “Fácil acceso a los recursos”, otro artículo señala que puede hacer que la “Información sea altamente eficiente”, como lo argumenta Zheng et al. (2008), al señalar que el OSS adopta el procedimiento de “planificación unificada e implementación paso a paso”. Además, se encontró que, en otro artículo, de Gatta et al. (2007), se indica como beneficio, que se ofrece “Tecnología bien documentada y disponible”.

Dentro de esta misma categoría se encuentra la subcategoría “Fácil integración de componentes”, donde 4 artículos mencionan que el beneficio de implementar OSS es la “Facilidad de integración de componentes” como parte de la adopción de OSS.

- **Categoría “Innovación”:** Subcategoría “Impulso a la innovación” de la cual se encontraron 3 artículos, que mencionan como beneficio que “Favorece la innovación” y uno de estos artículos también menciona que un beneficio es la “Innovación abierta (IO)” que ofrece, ya que según Morgan & Jensen (2014), actúa como catalizador de ideas nuevas en los procesos de desarrollo de software.

En la Figura nro. 5.42, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Innovación” y subcategoría “Impulso a la innovación”.

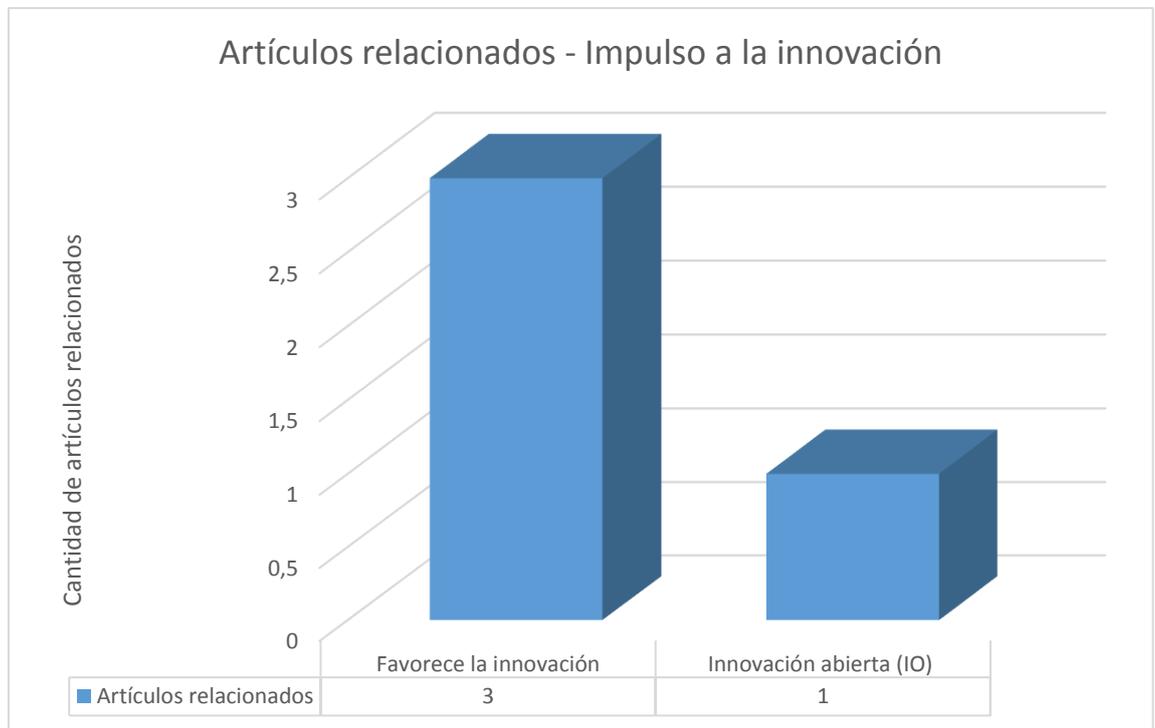


Figura 5.42 Artículos que hacen mención a la categoría “Innovación” y subcategoría “Impulso a la innovación” (Autoría propia, 2017).

- **Categoría “Flexibilidad”:** Subcategoría “Código fuente” de los 57 artículos encontrados que hacen referencia a la subcategoría: 22 mencionan, que un beneficio de la utilización de software libre es la “Disponibilidad de código fuente”; mientras que 24 artículos indican que un beneficio es la “Personalización (código fuente)”.

De acuerdo, a algunos autores como: Khelifi *et. al* (2009), Haider (2009), Morgan & Finnegan (2007) y Waring & Maddocks (2005), en sus informes resaltan la ventaja de la disponibilidad de acceso al código fuente para los adoptantes ya que facilita la personalización, ajustándose a las necesidades de cada usuario, a la reparación de errores, a la innovación de ideas de negocio entre otras; aumentando así, la fiabilidad de OSS. También otros autores como: Petrijevcanic Vuksanovic & Sudarevic (2012), señalan que una de las posibilidades de la utilización de software de código abierto en entidades públicas, radica en la solución de aquellos problemas que surgen tanto en el aspecto técnico, como en

lo político de dichas organizaciones; por tal motivo las soluciones de software de código abierto suministran la interoperabilidad, de manera directa o indirecta, ya que dichas aplicaciones al contar con la disponibilidad de código fuente facilitan que las mismas sean estudiadas, y adaptadas con diferentes sistemas.

La disponibilidad de código fuente visto como un beneficio también es apoyada por los autores, Maki-Asiala & Matinlassi (2006), que a partir de los resultados obtenidos a través de entrevistas a empresas finlandesas las cuales han hecho uso de componentes de código abierto, encontraron que la disponibilidad de código abierto permitía disminuir la dependencia del proveedor del componente, ya que alguno de los entrevistados tuvieron problemas de comunicación con proveedores de software propietario. Además se menciona que la disponibilidad de código fuente ayuda a reducir problemas de seguridad, De acuerdo Seiferth, (1999), en su artículo comenta que varios expertos en seguridad comprenden que la disponibilidad de código fuente facilita a los usuarios a que revisen fallas, lo reporten y propongan soluciones, ayudando a combatir este tipo de problemas; así como, el defecto del algoritmo de Skipjack (algoritmo de cifrado de datos).

En la Figura nro. 5.43, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Flexibilidad” y subcategoría “Código fuente”.

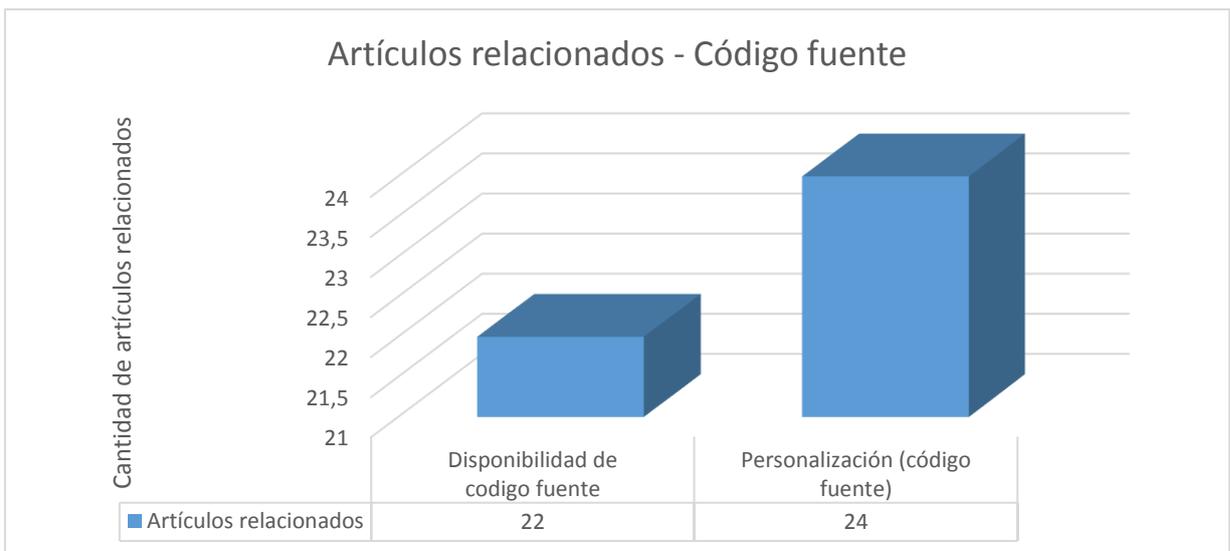


Figura 5.43 Artículos que hacen mención a la categoría “Flexibilidad” y subcategoría “Código fuente” (Autoría propia, 2017).

- **Categoría “Hardware”:** Subcategoría “*Herramientas*” de los 2 artículos encontrados que hacen referencia a la subcategoría: 1 artículo menciona, que un beneficio de la utilización de software libre es la “*Mayor selección de hardware*”; mientras que otro indica que el uso de las soluciones OSS “*Alarga la vida útil del hardware existente y reduce el desperdicio*”, según Oreku (2010), argumenta que muchas de las herramientas de FOSS comparadas con algunas herramientas propietarias requieran menos recursos de hardware, ya que las soluciones de FOSS evitan paquetes de software extensos e innecesarios.

- **Categoría “Licenciamiento”:** Subcategoría “*Adquisición*”, 1 artículo encontrado menciona, que un beneficio de la utilización de software libre es la “*Concesión de licencias*”, según Waring & Maddocks (2005), recalca que las múltiples licencias del OSS, facilitan a que las organizaciones públicas y privadas, puedan adquirir licencias más baratas, permitiendo el ahorro de costes por concesión de licencias, como ha ocurrido en diversos países, como es el caso de Rusia y Sudáfrica; y en continentes, como : Europa, entre otros

- **Categoría “Factores Humanos”:** Subcategoría “*Comunidad*” de los 13 artículos encontrados que hacen referencia a la subcategoría: 12 artículos mencionan, que un beneficio de la utilización de software libre es la “*Solución de errores*” y uno de ellos también indica la “*Colaboración directa con el usuario*”. Otro artículo señala como beneficio a la “*Comunicación directa con los desarrolladores*”, Según, los autores Maki-Asiala & Matinlassi (2006), resaltan que en el producto de una entrevista, encontraron que varias personas apreciaban la comunicación directa con los desarrolladores de componentes de OSS como fácil y rentable.

En la Figura nro. 5.44, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Factores humanos” y subcategoría “Comunidad”.

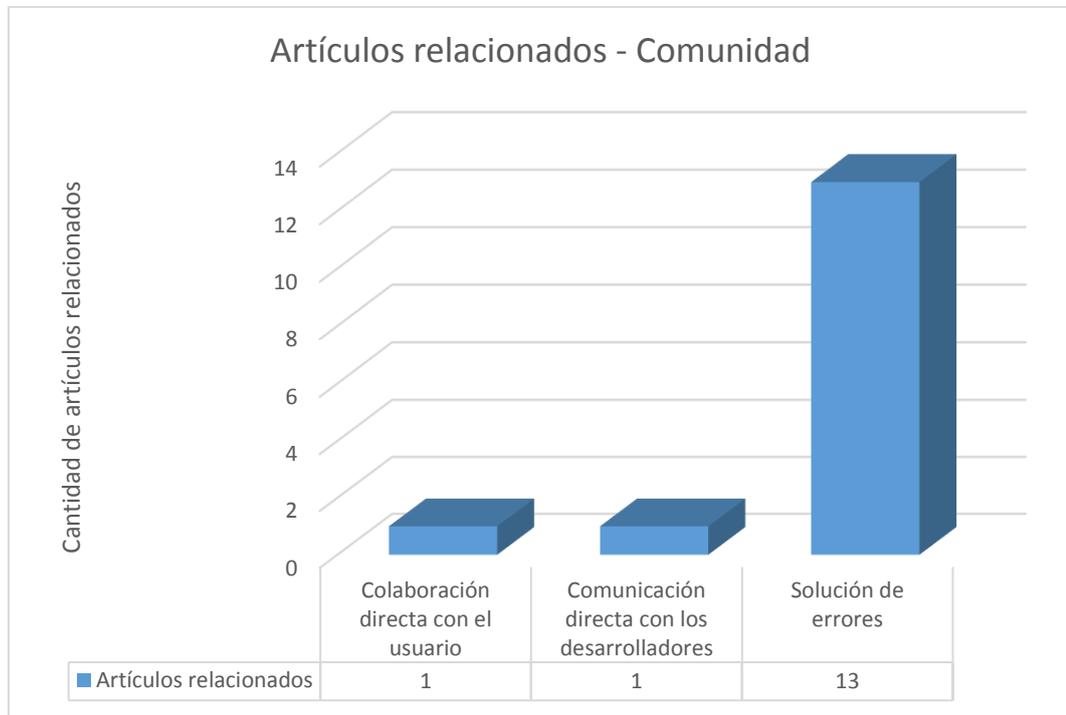


Figura 5.44 Artículos que hacen mención a la categoría “Factores humanos” y subcategoría “Comunidad” (Autoría propia, 2017).

En la categoría “Factores humanos” existe otra subcategoría, llamada “Contribuciones”, en los 17 artículos encontrados se menciona que un beneficio de utilizar software libre son las “Contribuciones al desarrollo de OSS”, de acuerdo con Oreku (2010), que indica que son varias las personas que apoyan o contribuyen con el desarrollo del software de código abierto, lo que ha permitido al adoptante no ser dependiente de una sola compañía. Además, los autores Morgan & Finnegan (2007), recalcan que la contribucion de OSS ofrece nuevas maneras de colaboración e intercambio de conocimientos, añaden que la contribución ayuda a que los gastos de una empresa sean compartidos con otras.

En la categoría “Factores humanos” y subcategoría “Proveedores”, 11 artículos encontrados mencionan que un beneficio de utilizar software libre es la “Independencia de proveedores”, de acuerdo con Khelifi *et al.* (2009), resaltan en su artículo, que los resultados obtenidos producto de la encuesta practicada por “Computer Economics” indican que para las personas relacionadas con TI la

ventaja mas importante que tiene el codigo abierto es la menor dependencia de proveedores.

- **Categoría “Software”:** Subcategoría “Control”, se han encontrado 7 artículos que hacen referencia a la subcategoría: 1 artículo menciona como beneficio a la “Gama de opciones de software libre”, Oreku (2010), indica que las ventajas que proporciona el OSS es la cantidad de opciones que presenta, 4 artículos mencionan como beneficio a la “Alta calidad de software”, que presenta el OSS, tal como lo indica sno-lfdg12 que el OSS es software de calidad y se encuentra a la vanguardia de la tecnología, otro artículo menciona que los “Bugs “errores” tienden a ser más visibles”, lo cual beneficia al obtener más control del software según Khelifi *et al.* (2009), además 1 artículo señala como beneficio según Koch (2004), que la adopción de OSS “Garantiza ciclos de retroalimentación cortos”.

En la Figura nro. 5.45, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “Software” y subcategoría “Control”.

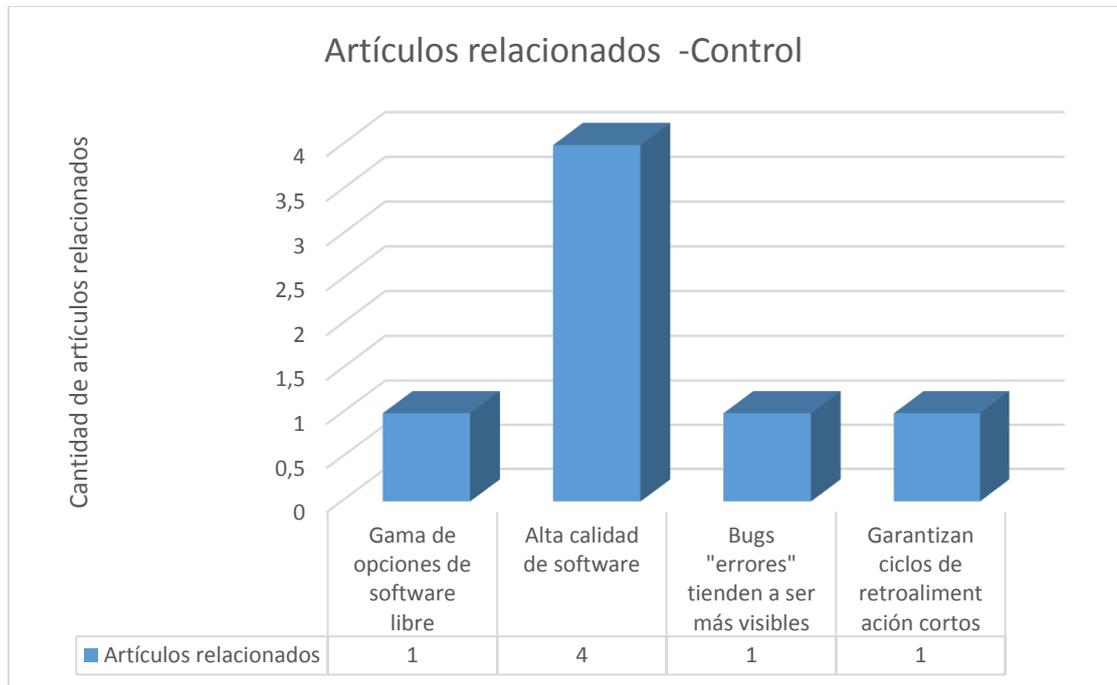


Figura 5.45 Artículos que hacen mención a la categoría “Software” y subcategoría “Control” (Autoría propia, 2017).

Dentro de esta misma categoría se encuentra la subcategoría “Software desarrollado”, donde de los 7 artículos encontrados que hacen referencia a la

subcategoría: 3 artículos señalan como beneficio la “*Mayor estabilidad*”, Oreku (2010), señala que el identificar y eliminar errores es muy rápido, debido a la gran comunidad que trabaja detrás del proyecto, lo hace muy estable, además Comino & Manenti (2005), indica que los proyectos OSS superan en estabilidad a los de software propietario“, 2 artículos mencionan que el beneficio de implementar OSS es que “*Mejora el rendimiento*”, como lo indica Tawileh & Rana (2006), en donde el crecimiento de adopción en OSS es debido a la mejora de su rendimiento, también Morgan & Finnegan (2007), señalan que un beneficio es el alto rendimiento que entrega el OSS en cuestiones de velocidad y capacidad. Otro artículo demuestra que el beneficio de adoptar OSS es la “*Retroalimentación*” que ofrece el OSS, como lo indica Munir *et al.* (2016), que al ser código abierto mejora la retroalimentación, este artículo junto a otro indica también que un beneficio de implementar OSS es la “*Reutilización del código*” de acuerdo a Ajila & Wu (2007), señalan que gracias a la reutilización de código se ahorra tiempo de comercialización, se ahorra en desarrollo y se mejoran los procesos de previsibilidad.

En la Figura nro. 5.46, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “*Software*” y subcategoría “*Software desarrollado*”.

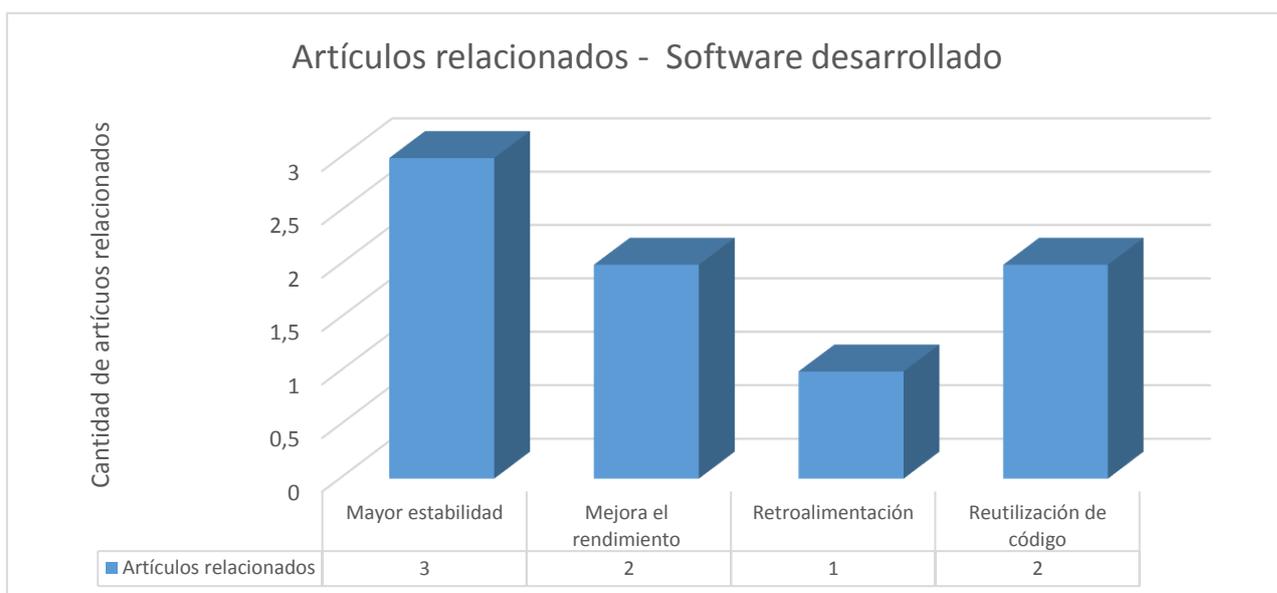


Figura 5.46 Cantidad de artículos que hacen mención a los beneficios en la categoría “*Software*” y subcategoría “*Software desarrollado*” (Autoría propia, 2017).

Dentro de la categoría “*Software*” también se encuentra la subcategoría “*Versiones*”, en donde de los 3 artículos encontrados que hacen referencia a la subcategoría: 1 artículo señala como un beneficio que existe “*Mayor control sobre las actualizaciones a nuevas versiones*”, otro artículo menciona que “*Predice y mejora comportamientos futuros*”, como lo indica Escribano *et al.* (2016), que gracias a esto permite tomar acciones correctivas en cuanto a problemas de versiones que se puedan dar, además 1 artículo señala que un beneficio es que “*Se puede hacer lanzamiento de versiones frecuentes*”.

Dentro de la categoría “*Software*” también se encuentra la subcategoría “*Normas/Reglamentos*”, en donde de 1 artículo encontrados, se señala como un beneficio a la “*Adhesión a las normas*”, Kovacs *et al.* (2004), señala que debido a la naturaleza modular del OSS y la estandarización del código abierto, hace común la adopción de código abierto.

- **Categoría “*Tiempo*”:** Subcategoría “*Ahorro de tiempo*”, de los 10 artículos encontrados que hacen referencia a la subcategoría: 5 artículos mencionan como un beneficio que “*Reduce el tiempo en detectar errores*”, tal como lo indica Xie (2008), que la velocidad al solucionar errores es mayor, por la ventaja de que cualquier persona puede revisar el código fuente, también Ebert (2007), señala que los informes de error son entregados de manera más rápida y segura, dos de los artículos anteriores junto a otros 4 artículos también mencionan como beneficio que “*Reduce tiempo en encontrar mejoras*”, como lo indican Ruffin & Ebert (2004), que el uso de OSS minimiza el tiempo de obtención de nuevas actualizaciones y correcciones, además 1 artículo indica que “*Reduce tiempo de desarrollo*”.

En la Figura nro. 5.47, se presenta el número de artículos que hacen mención a cada beneficio en la categoría “*Tiempo*” y subcategoría “*Ahorro de tiempo*”.

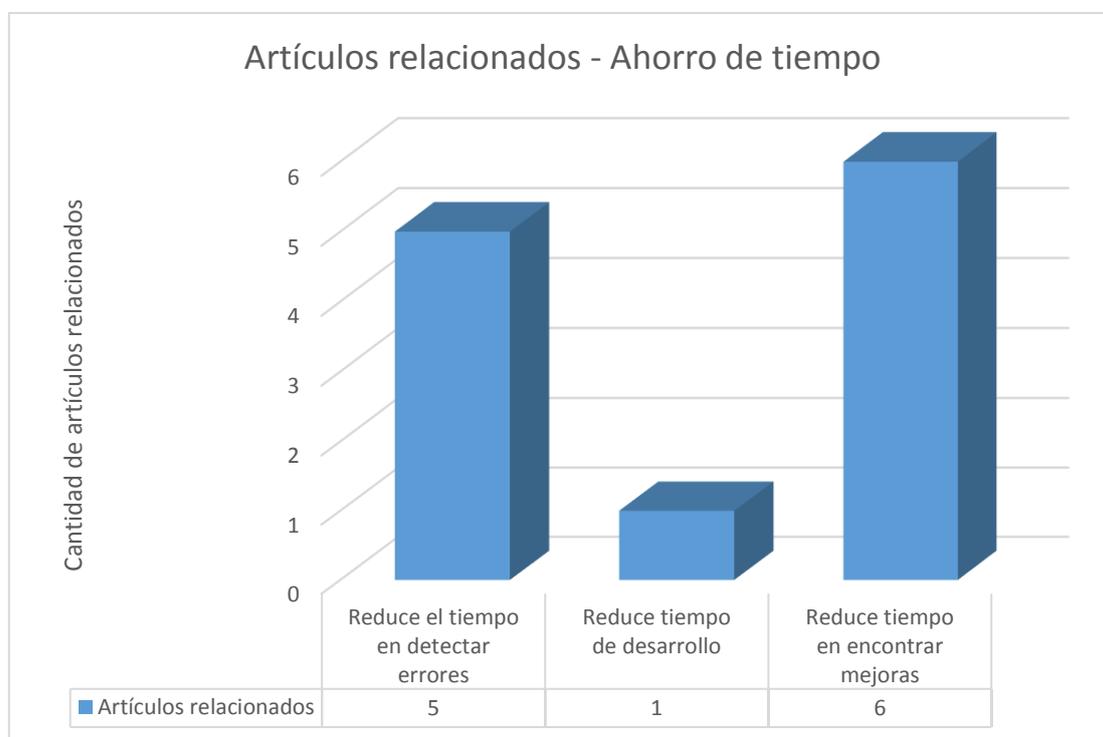


Figura 5.47 Cantidad de artículos que hacen mención a los beneficios en la categoría “*Tiempo*” y subcategoría “*Ahorro de tiempo*” (Autoría propia, 2017).

5.2.4. ¿En qué dominios de software ha resultado exitosa la adopción de software libre a nivel mundial?

De los 85 artículos seleccionados, se han encontrado 30 artículos de dominios de software exitosos, para la implementación. De estos 30 mencionados, se encontraron solo 16 dominios exitosos, en donde un 53% concuerda que el más exitoso en la adopción de software libre a nivel mundial es el “*software de sistema*”, el cual abarca los dominios de: “*sistemas operativos*”, “*sistemas de servidores*” y “*sistemas de diagnóstico*”; mientras que el 44% de las publicaciones que hablan de dominios de éxito en la adopción de software libre se basa en el “*software de aplicaciones*”, que está relacionado con: app móviles, app de tipo comercial, ofimática, aplicaciones para el área educativa y médica, para uso en las bases de datos; en la parte web, aplicaciones para los navegadores y finalmente para el uso en el área militar. Por otra parte, un 3% de artículos menciona a la categoría “*otros*”, “*entornos de desarrollo*” y “*otras áreas de investigación científica*”.

En la Figura nro. 5.48, se presenta de manera gráfica la distribución de las categorías de dominios exitosos.

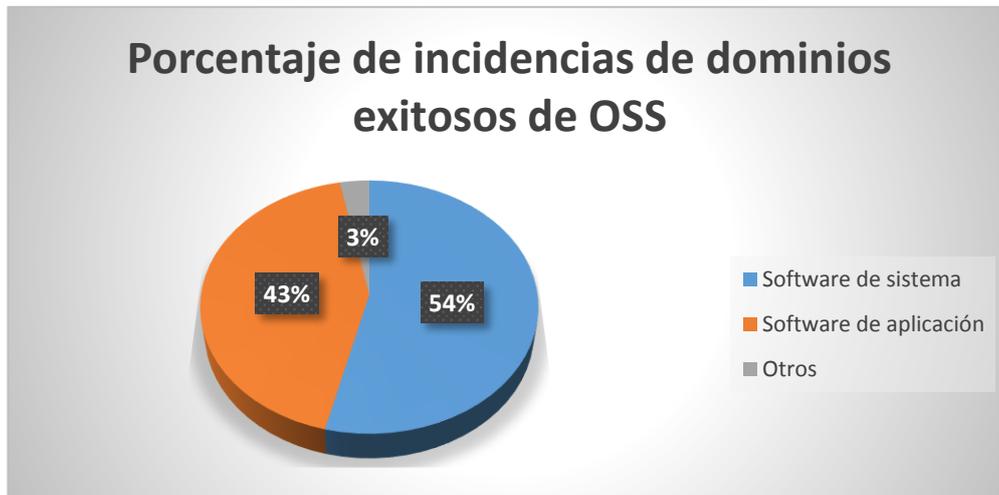


Figura 5.48 Categorías de dominios exitosos de adopción de OSS (Autoría propia, 2017).

Del total de artículo, se encontraron 16 dominios de implementación exitosa en la adopción de tecnologías de OSS, que se han agrupado en 3 categorías y 14 subcategorías. En la Figura nro. 5.49, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

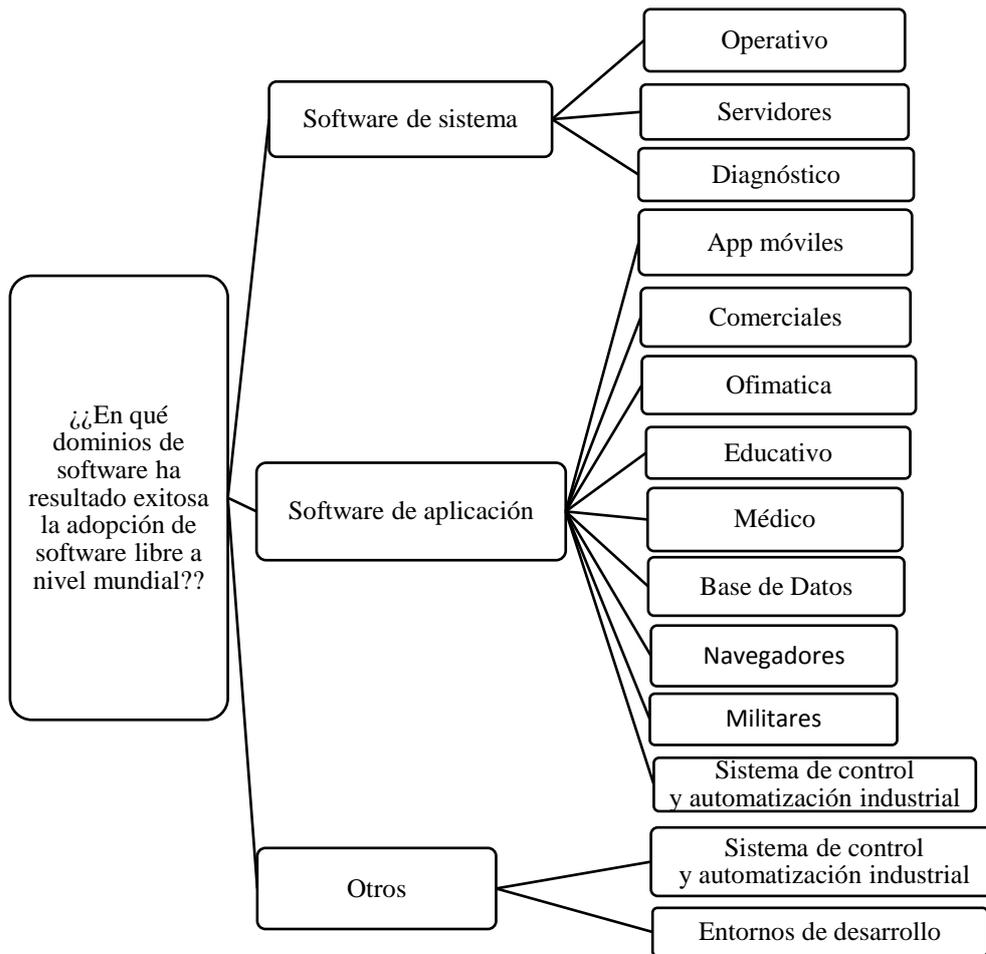


Figura 5.49 Categorías y subcategorías a los que pertenecen dominios exitosos en la adopción de OSS (Autoría propia, 2017).

En la Figura nro. 5.50, se presenta el número de subcategorías que contiene cada categoría.

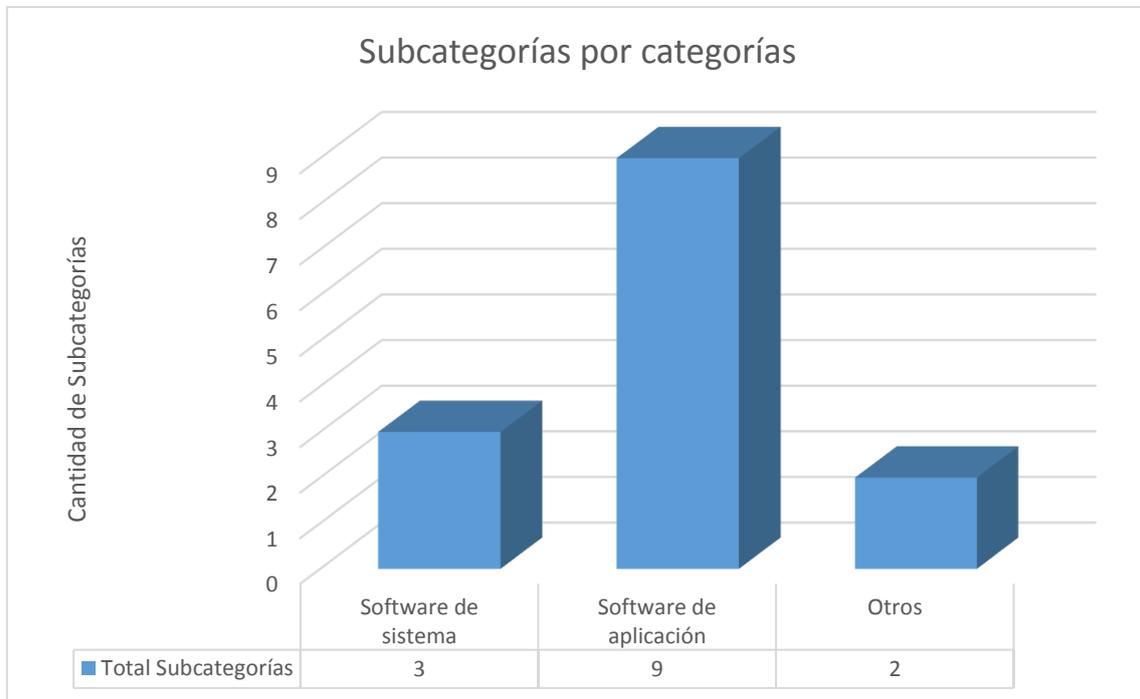


Figura 5.50 Número de subcategorías por categoría (Autoría propia, 2017).

En la Figura nro. 5.51, se presenta el número de dominios exitosos identificados, que contiene cada subcategoría.

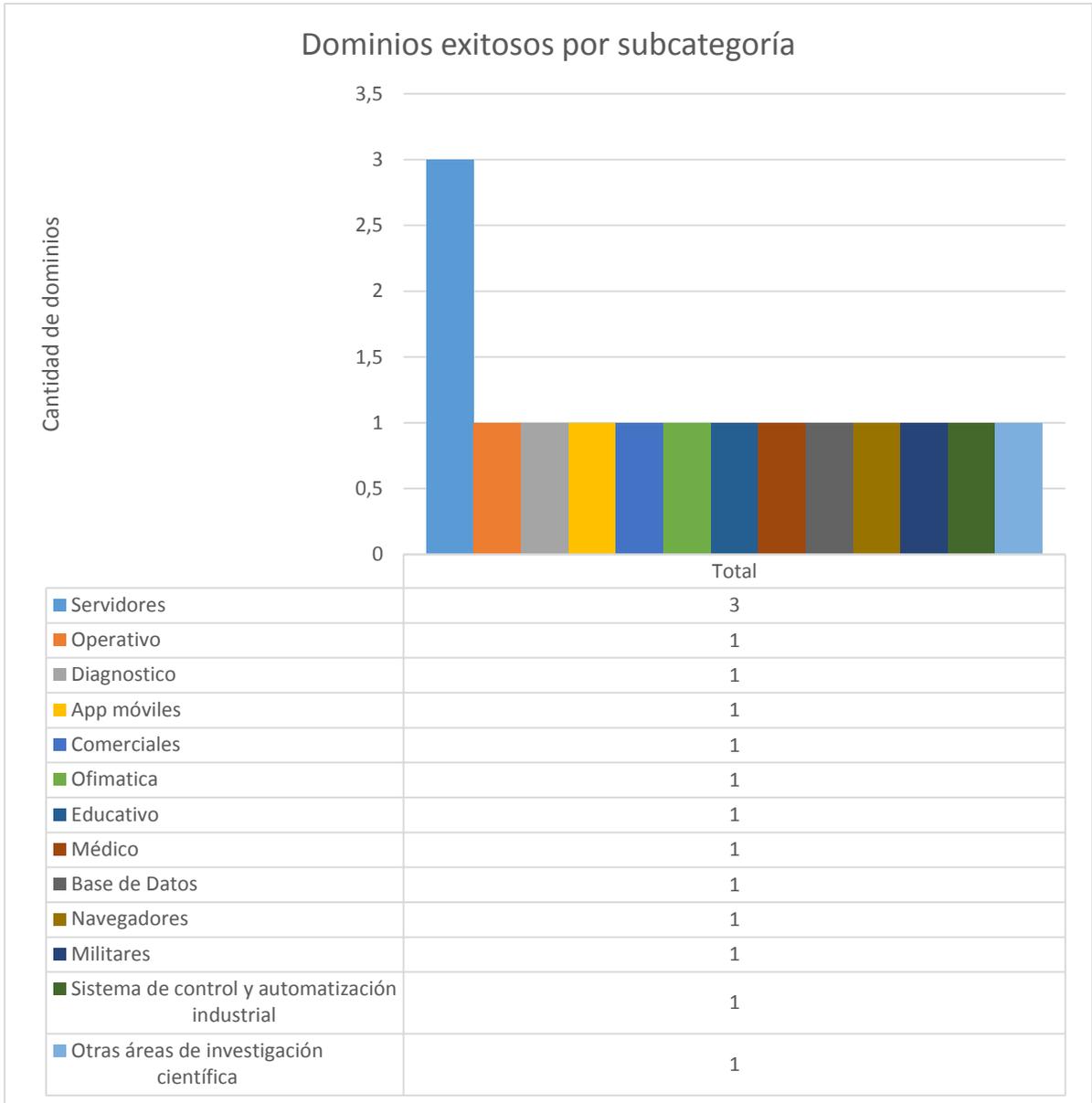


Figura 5.51 Cantidad de dominios de éxito encontrados por subcategoría (Autoría propia, 2017).

Como se ha indicado anteriormente producto de la lectura de los artículos seleccionados, se ha identificado 16 dominios de éxito en la adopción de software libre, que han sido mencionados por los autores de dicha literatura, los mismos se presentan en la Tabla nro. 5.5.

Número de dominio	Categoría	Subcategoría	Dominio exitoso	Frecuencia de dominios exitosos por artículo	Porcentaje de incidencia
D1	Software de sistema (Software que sirve para interactuar y controla el sistema operativo)	Sistemas Operativos (Conjunto de programa que administran los recursos de la computado y controlan su funcionamiento)	Sistemas Operativos (Sistemas operativos de escritorio y móviles)	13	20%
D2		Servidores (Es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.)	Web (Programa informático que procesa una aplicación del lado del servidor.)	12	18%
D3			Correo (Aplicación de red de computadoras para prestar servicio de correo electrónico)	7	11%
D4			Archivos e impresión (Servidor de archivos es en el que se guardan, almacenan y se accede a través de una red de archivos estáticos y Servidor de impresión es un tipo especial de servidor que contiene todas las impresoras de una red)	1	2%
D5			Diagnostico (Permiten acceder a una cantidad de información para equipos de PC y red para corregir defectos)	Seguridad (Software de seguridad diseñado para brindar seguridad informática)	2
D6		Software de aplicación (Programas diseñados para o por los usuarios para facilitar la realización)	App móviles (Aplicación informática diseñada para ser ejecutada dispositivos móviles)	App móviles	1

D7	de tareas específicas en la computadora)	Comerciales (Es el software (libre o propietario) que es comercializado.)	Aplicaciones comerciales	1	2%	
D8		Ofimática (Aplicación o paquete de aplicaciones que sirven para facilitar el trabajo en el ámbito de una oficina)	Aplicaciones ofimáticas (OpenOffice, Office Suite)	5	8%	
D9		Educativo (Software vinculado a la educación)	Aplicaciones educativas	9	14%	
D10		Médico (Software vinculado a los servicios médicos)	Aplicaciones médicas	4	6%	
D11		Base de Datos (Implica definir y organizar el contenido, las relaciones y la estructura de los datos necesarios para construir una base de datos.)	Base de datos	4	6%	
D12		Navegadores (Permite el acceso a la Web)	Navegadores	2	3%	
D13		Militar (Software vinculado al servicio militar)	Aplicaciones militares	1	2%	
D14		Sistema de control y automatización industrial (Utilizadas en el campo de la automatización y el control automático industrial)	Aplicaciones industriales	1	2%	
D15		Otros	Otras áreas de investigación científica (Software orientado a otras áreas no especificadas anteriormente)	Otras áreas de investigación científica	1	2%
D16			Entornos de desarrollo (Proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software)	Entornos de desarrollo	1	2%
Totales				65	100%	

Tabla 5.5 Dominios de éxito de adopción de OSS (Autoría propia, 2017).

La Figura nro. 5.52, hace referencia a la Tabla nro. 5.5, mencionada anteriormente; en donde, en la gráfica se presenta el nombre del dominio de éxito con más de 1 incidencia; mientras que los que presentan una incidencia son denominados “Otros”.

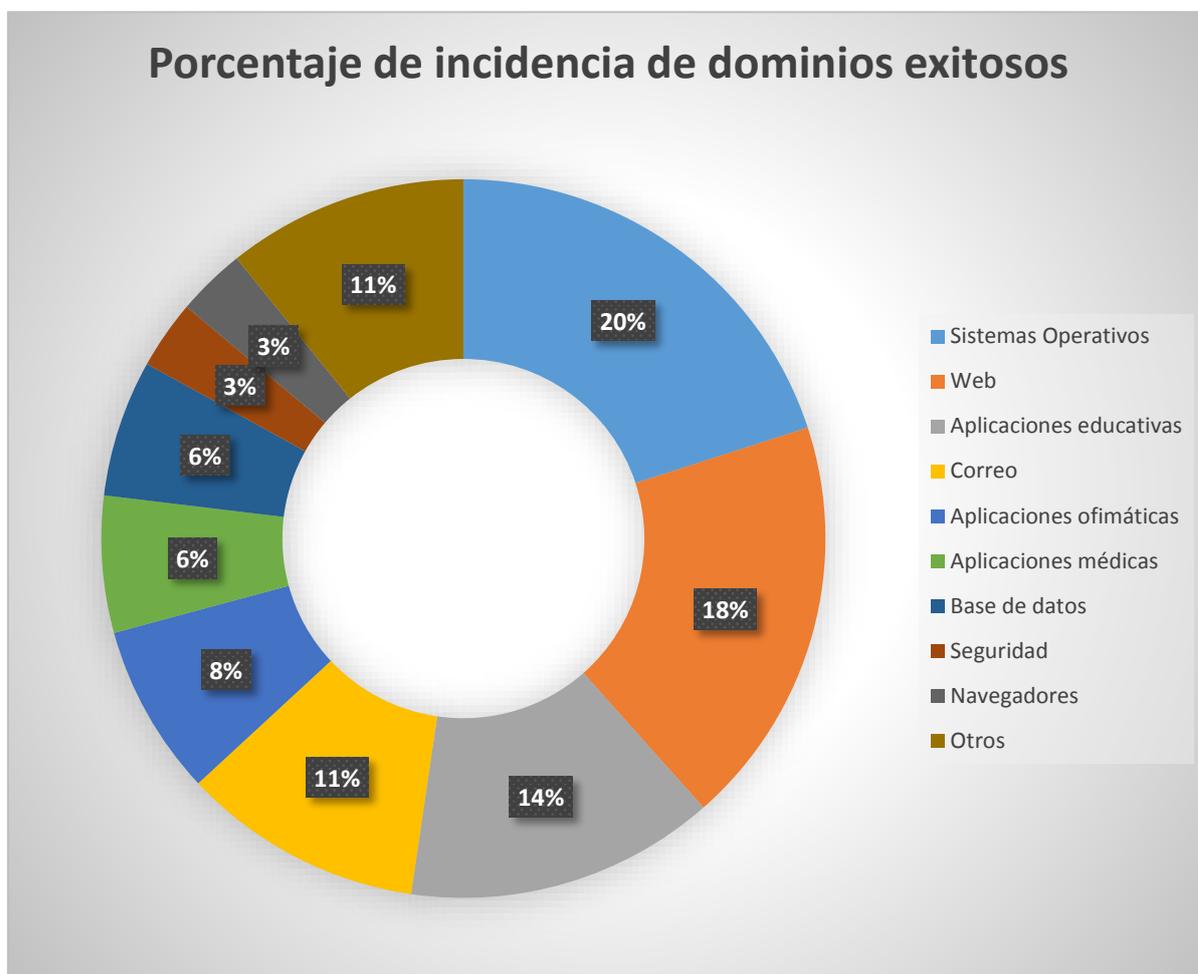


Figura 5.52 Representación gráfica de menciones de los dominios exitosos de adopción de OSS (Autoría propia, 2017).

A partir de la información expuesta anteriormente en la Tabla nro. 5.5, y Figura nro. 5.52, se puede decir que el dominio de éxito más mencionado es el de “*Sistemas Operativos*” con un 20%, luego se encuentra el dominio de la “*Web*” con un 18% de incidencias, seguido por: “*Aplicaciones educativas*” con un 14%. Además, se tiene el dominio de “*Correo*” con un 11%, “*Aplicaciones ofimáticas*” con un 8%, “*Aplicaciones médicas*” y “*Base de datos*” con el 6%, para un 3% de incidencias se encuentran los siguientes: “*Seguridad*” y “*Navegadores*”; mientras que los demás dominios se encuentran con un 1% de incidencias; estos dominios serán descritos a continuación con más profundidad.

Se recalca que la información obtenida de los dominios de éxito en la implementación de OSS, en la mayoría de los autores se hace una breve referencia sobre el tema, o solo se menciona y no se muestra evidencia amplia de lo que se expone en el artículo, de esta manera, se obtiene lo siguiente:

- **Categoría “Software de sistema”:** Subcategoría “Sistemas Operativos” los 13 artículos encontrados en esta subcategoría, mencionan que un dominio exitoso en la adopción del software libre son los “Sistemas operativos”, entre los artículos que mencionan este dominio exitoso podemos citar:
 - Oreku (2010), menciona que el código abierto es ampliamente exitoso en el desarrollo de Linux, debido a la flexibilidad que este ofrece en su uso.
 - Comino *et al.* (2005), indican que el sistema operativo Linux es un ejemplo de éxito, además de ser el más significativo dentro de los sistemas operativos, por lo cual, se ha demostrado un desarrollo importante en el mercado de servidores.
 - Sen (2007), señala que Linux es un sistema operativo, el cual ha alcanzado un gran éxito en comparación con el software propietario.

Dentro de la categoría “Software de sistema” y subcategoría “Servidores”, de los 13 artículos encontrados que hacen referencia a esta subcategoría, 12 mencionan que un dominio exitoso son los sistemas “Web”, Sen (2007), señala que el servidor web, Apache, ha alcanzado un gran éxito en comparación con su alternativa de software propietario. También 7 artículos mencionan como un dominio exitoso al servidor de “Correo”, como lo indica Sen (2007), que el servidor de correo, Sendmail, es muy popular como alternativa de su competencia en software propietario, además tenemos 1 artículo que menciona que un dominio exitoso son los servidores de “Archivos e Impresión”, ya que según Khelifi *et.al* (2009), indica que la suite Samba para servidores de archivos e impresión es un gran ejemplo de éxito en la implementación de tecnologías OSS.

En la Figura nro. 5.53, se presenta el número de artículos que hacen mención a cada riesgo en la categoría “*Software de sistema*” y subcategoría “*Servidores*”.

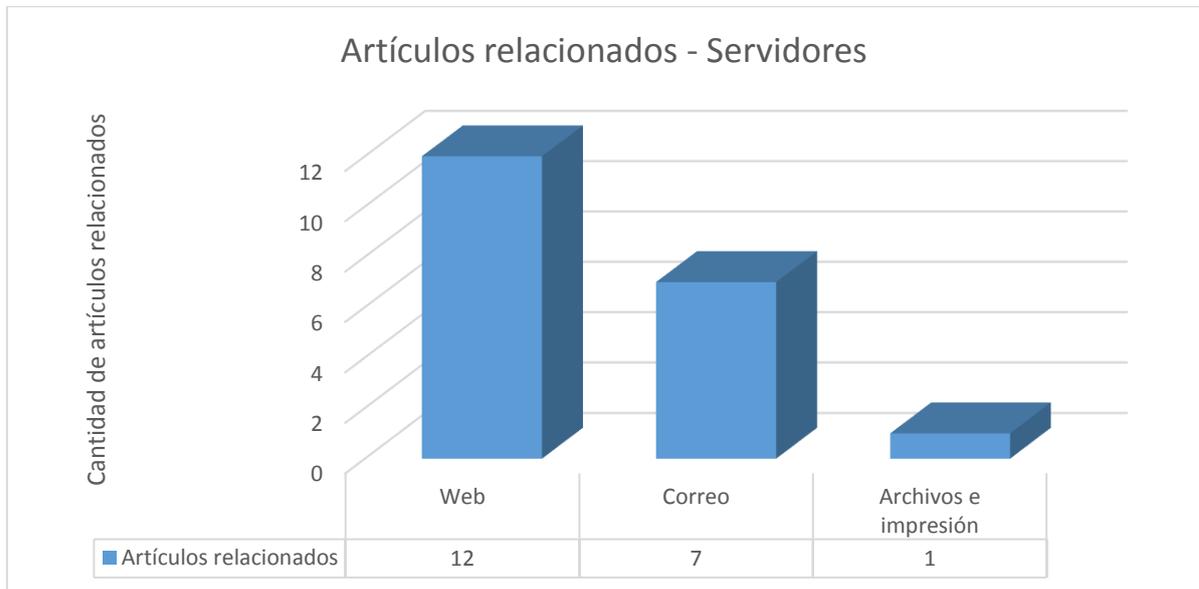


Figura 5.53 Artículos que hacen mención a la categoría “*Software de sistema*” y subcategoría “*Servidores*” (Autoría propia, 2017).

Dentro de la categoría “*Software de sistema*” y subcategoría “*Diagnóstico*”, los 2 artículos encontrados, mencionan que un dominio exitoso son los sistemas de “*Seguridad*”.

- **Categoría “*Software de aplicación*”:** Subcategoría “*App Móviles*” se ha encontrado 1 artículo que menciona que las “*App móviles*” son dominios exitosos en la adopción de tecnologías de OSS.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Comerciales*”, se ha identificado 1 artículo que señala que las “*Aplicaciones comerciales*”, son un dominio exitoso en la implementación de OSS.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Ofimática*”, se ha identificado 5 artículos que señalan que las “*Aplicaciones ofimáticas*”, son un dominio exitoso en la implementación de OSS. Ngamkajornwivat *et al.* (2008), indica que KOffice ha progresado con fuerte éxito en los últimos años, al igual

que Petrijevcenin (2012), señala que la OpenOffice (paquete ofimático) es uno de los dominios más exitosos de OSS.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Aplicaciones Educativo*”, se han identificado 9 artículos que señalan que las “*Aplicaciones educativas*”, son un dominio exitoso en la implementación de OSS. Gary *et al.*, (2009), indica que Unicon es una empresa de desarrollo de código abierto, desarrollado por Academus, que es un software de aplicación educativo que ha tenido mucho éxito en este ámbito, Sahraoui (2009), señala que el portal educativo U-portal y aplicaciones de ambientes educativos virtuales como Moodle y Sakai son exitosas en el campo de aplicaciones educativas.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Aplicaciones médicas*”, se han identificado 4 artículos que señalan que las “*Aplicaciones médicas*”, son un dominio exitoso en la implementación de OSS. Noll *et al.* (2011), muestra los resultados del estudio de OpenEMR, que produjo un software de registro electrónico médico(EMR), el cual se consideró que tuvo éxito en esta especialización, ya que el proyecto se mantuvo por un gran período de tiempo.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Base de datos*”, se han identificado 4 artículos que señalan que las “*Bases de datos*”, son un dominio exitoso en la implementación de OSS. Khelifi *et al.* (2009), indica que la Base de datos MYSQL fue exitosa, porque ha tenido un gran impacto en el mercado.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Navegadores*”, se han identificado 2 artículos que señalan que las “*Navegadores*”, son un dominio exitoso en la implementación de OSS, Raj & Kazemian (2006), señala que existe gran variedad de herramientas de OSS que son exitosas, se menciona como ejemplo, al navegador de Mozilla Firefox.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Aplicaciones militares*”, se ha identificado 1 artículo que señalan que las “*Aplicaciones militares*”, son un dominio exitoso en la implementación de OSS.

Dentro de la categoría “*Software de aplicación*” y subcategoría “*Sistema de control y automatización industrial*”, se ha identificado 1 artículo que señala que las “*Aplicaciones industriales*”, son un dominio exitoso en la implementación de OSS.

- **Categoría “*Otros*”:** Subcategoría “*Otros*”, se han identificado 2 artículos, de los cuales 1 señala que “*Otras áreas de investigación científica*” cuentan con dominios exitosos de implementación de OSS, también 1 artículo señala que un dominio exitoso son los “*Entornos de desarrollo*”.

5.2.5. ¿Cuáles son las barreras que han identificado las organizaciones al adoptar software libre?

De la lectura de los 85 artículos, 35 artículos mencionan a las barreras que impiden a las organizaciones adoptar software libre, de estos artículos se han extraído 60 barreras, las mismas que se han agrupado en 12 categorías y 27 subcategorías. En la Figura nro. 5.54, se puede observar un gráfico que contiene tres niveles:

- El primer nivel contiene la pregunta de investigación.
- El segundo nivel contiene la clasificación en categorías de lo investigado.
- El tercer nivel contiene la clasificación en subcategorías de lo investigado.

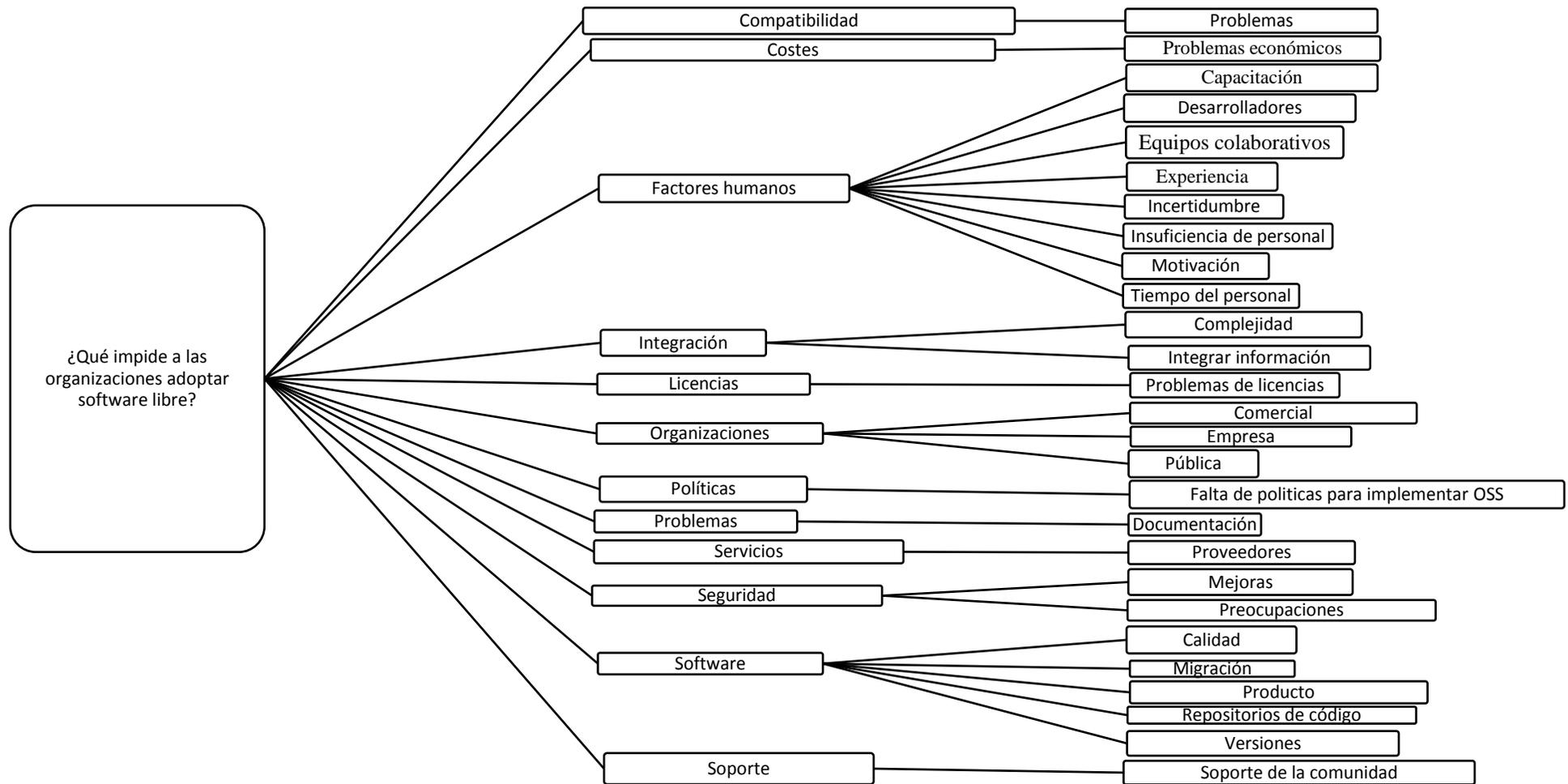


Figura 5.54 Categorías y subcategorías a los que pertenecen las barreras identificadas en la adopción de OSS (Autoría propia, 2017).

En la Figura nro. 5.55, se presenta el número de subcategorías que contiene cada categoría.

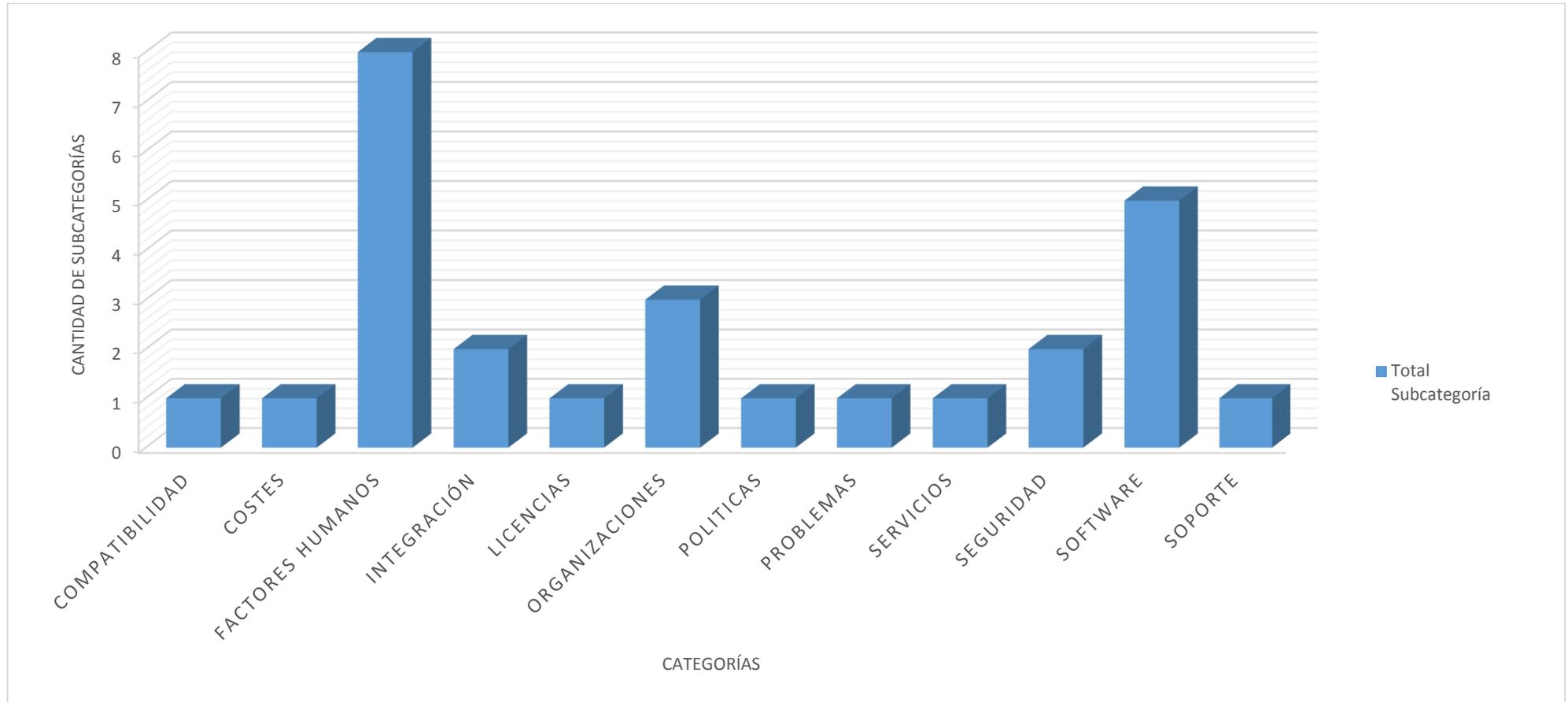


Figura 5.55 Número de subcategorías por categoría (Autoría propia, 2017).

En la Figura nro. 5.56, se presenta el número de barreras encontradas que contiene cada subcategoría.

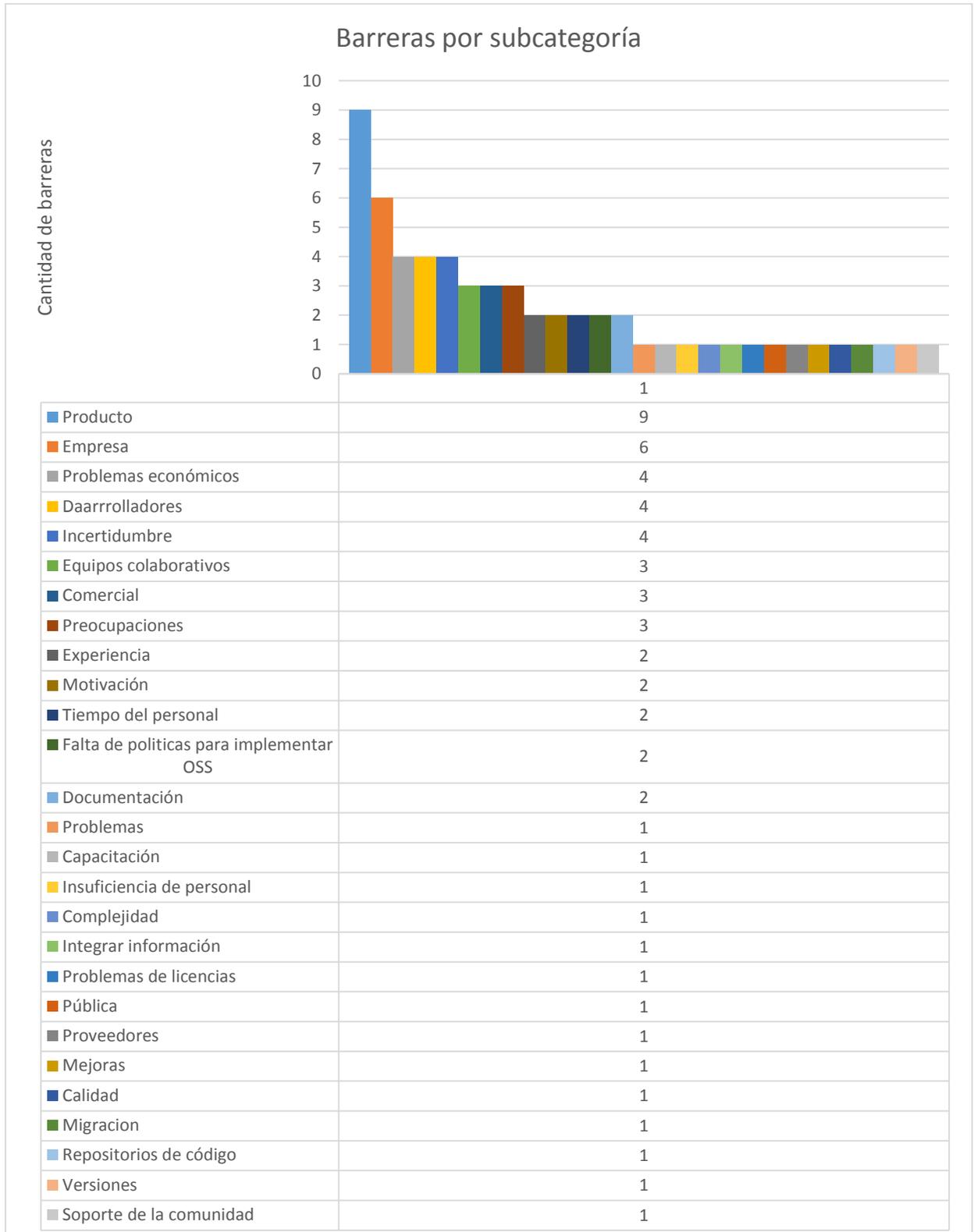


Figura 5.56 Cantidad de Barreras encontradas por subcategoría (Autoría propia, 2017).

En resumen, de los resultados obtenidos de la búsqueda de las barreras que impiden a las organizaciones adoptar software libre, el 32% de barreras han sido clasificadas en la categoría “*Factor Humano*”, un 22% en la categoría “*Software*”, con el 17%, se encuentra la categoría de “*Organizaciones*”, un 7% con las categorías de “*Seguridad*” y “*Costes*”; mientras que un 3%, pertenece a las categorías de “*Integración*”, “*Políticas*”, y “*Problemas*”; un 2% en las categorías de “*Soporte*” y “*Servicios*” y un 1% en las categorías de “*Compatibilidad*” y “*Licencias*”, como se presenta en la Figura nro. 5.57.

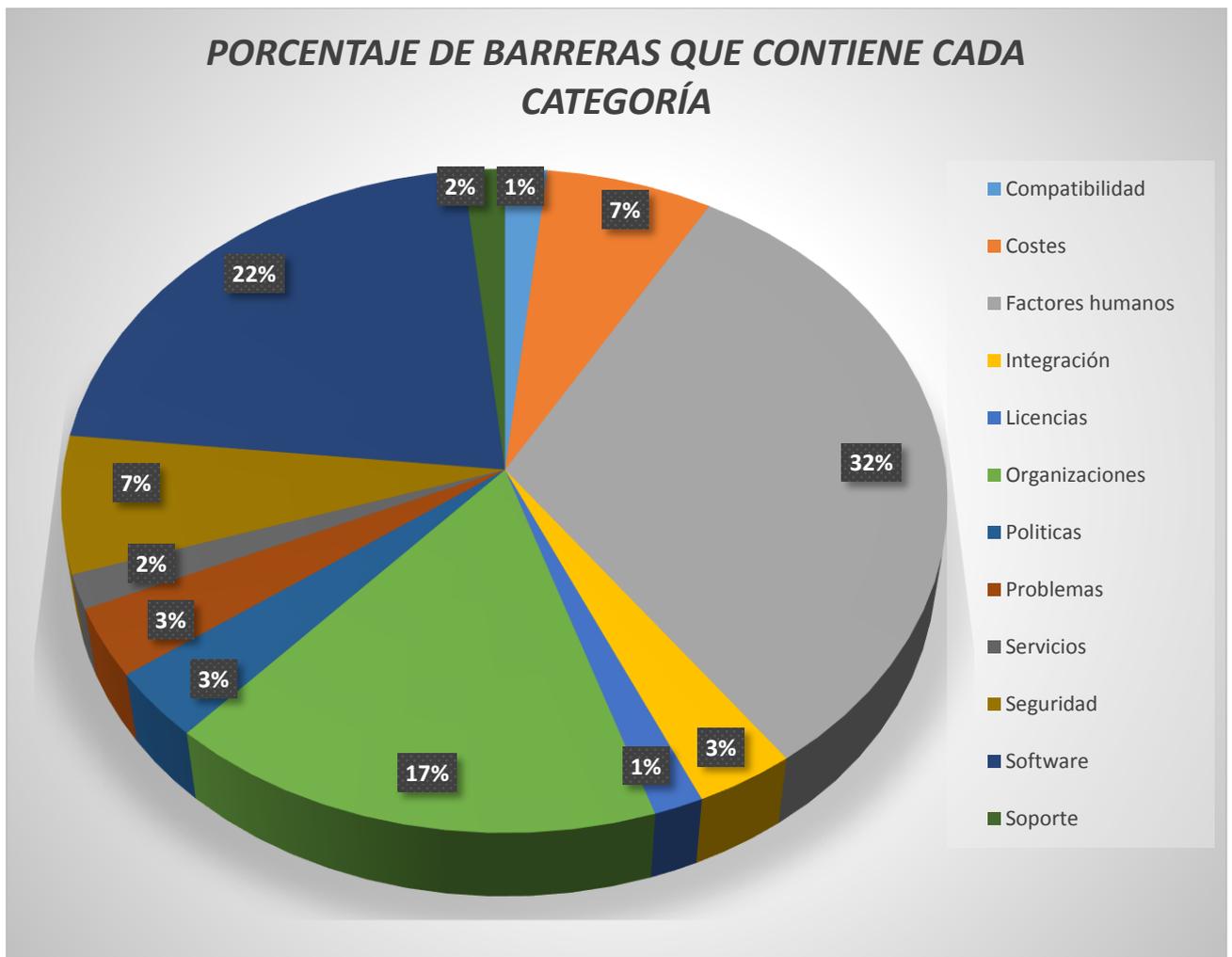


Figura 5.57 Porcentaje de barreras que contiene cada categoría (Autoría propia, 2017).

Como se ha indicado anteriormente, producto de la lectura de los artículos seleccionados, se ha encontrado 60 barreras que impiden a las organizaciones adoptar software libre, las cuales han sido mencionadas por los autores de dicha literatura, estas barreras se presentan en la Tabla nro. 5.6.

Número de beneficio	Categoría	Subcategoría	Beneficios	Frecuencia de factores por artículo	Porcentaje de incidencia	
B1	Compatibilidad (facilidad para que el software o hardware se pueda comprenderse directa o indirectamente)	Problemas (dificultad para la adaptación por parte del software o hardware.)	Problemas de compatibilidad de Software / Hardware (Como la tecnología actual prevalece sobre otras tecnologías adoptadas con anterioridad en una organización)	5	5%	
B2		Problemas económicos (Obstáculo que tiene una organización con los recursos económicos)	Aumento de costes en la organización	1	1%	
B3			Costes no recuperables	1	1%	
B4			Dificultad para estimar el coste total de la propiedad del software	1	1%	
B5			Inversiones grandes en la formación de OSS	2	2%	
B6	Factores humanos	Capacitación (Actividades para ampliar conocimiento, habilidades, del personal la organización relacionado a FOSS.	Formación al personal (El reclutamiento del personal y la participación como practicantes, desde el diseño del proyecto)	2	2%	
B7		Desarrolladores (programador, comunidad o compañía dedicada a uno o más aspectos del proceso de desarrollo de software)	Desarrollo de software es escrito por diferentes grupos	1	1%	
B8			Expectativas de trabajo poco claras, no se sabe el tiempo que se dispone en la comunidad	1	1%	
B9			Falta de entendimiento de la estructura o arquitectura del proyecto	1	1%	
B10			Falta de roles formales	1	1%	
B11			Equipos colaborativos	Falta de orientación a los nuevos miembros	1	1%
B12			(procesos intencionales de un grupo para alcanzar objetivos específicos)	Falta de personal que conduce a bajos rendimientos	1	1%
B13			Sobrecarga de comunicación con equipos colaborativos grandes	1	1%	
B14			Experiencia (conocimiento adquirido del FOSS a través de las vivencias obtenidas).	Cultura de desarrollo basada en voluntariado (desarrolladores no sujetos a contratos)	2	2%
B15			Falta de conocimiento (desconocimiento de: disponibilidad de software, implementación, desconocimiento de negocio para personalizarlo y uso).	4	4%	
B16			Incertidumbre (grado de desconocimiento de una acción futura en:	Incertidumbre en servicio y soporte (Al no conocer los equipos que dan servicio y soporte externos)	8	7%
B17			Incertidumbres legales	1	1%	
B18			Inmadurez tecnológica	1	1%	

B19		servicio y soporte, aspectos legales y otros.)	Poner a disposición el código fuente	1	1%
B20		Competencia (capacidad de combinar y utilizar conocimientos para obtener resultados esperados)	Competencia individual (Evaluación de competencia que se hace entre los desarrolladores a través de su experiencia.)	3	3%
B21		Motivación (impulso que conduce a una persona a elegir y realizar una acción en una determinada situación)	Falta de motivación por parte de la comunidad	1	1%
B22			Rechazo al OSS (Por considerarse crítico durante la implementación del producto OSS, además que estos productos tienen un futuro incierto y el ciclo de vida del proyecto no está claro en general. Personas sin deseo de abandonar las tecnologías que están acostumbradas a usar).	6	6%
B23		Tiempo del personal (Periodo de tiempo que se dispone para realizar una actividad)	Falta de tiempo (Está relacionado con el desconocimiento del tiempo que entregan tanto los desarrolladores, como los equipos de servicio de soporte, que se desconoce el tiempo que entregan para trabajar en el proyecto).	4	4%
B24			Reducción de coordinación en tiempo real	1	1%
B25	Integración (Conjunto de componentes que se unen para formar un todo)	Complejidad (Dificultad de establecer un modelo de gestión)	Complejidad de sistemas de gestión	1	1%
B26		Integrar información (Dificultad al unir la información)	Problemas con la transferencia de información de un sistema a otro	2	2%
B27	Licencias (Autorización concedida para realizar una determinada actividad)	Problemas de licencias (Obstáculos encontrados con el licenciamiento)	Problemas con las licencias al no estar claras o ser mal utilizadas.	5	5%
B28	Organizaciones (Grupo de personas y medios organizados con un fin determinado)	Comercial (vinculado con el comercio o con las personas que se dedican a comprar y/o vender bienes o servicios)	Existencia de acuerdos con organizaciones comerciales	1	1%
B29			Falta de interacción social	1	1%
B30		Insuficiente comercialización	1	1%	
B31		Ausencia de adopción de OSS	1	1%	
B32		Falta de apoyo de la dirección	1	1%	
B33		Falta de modelos de negocio	1	1%	
B34		Falta de marca de FLOOS	1	1%	
B35		Falta de voluntad de los gobiernos y las empresas para modernizar las aplicaciones heredadas	1	1%	
B36		La posición competitiva de la organización	1	1%	

B37		Pública (Es aquella que es propiedad del Estado, sea éste nacional, municipal o de cualquier otro estrato administrativo,)	Sector Público es impulsado por las directivas gubernamentales	1	1%	
B38	Políticas (Proceso de tomar decisiones que se aplican a todos los miembros de un grupo)	Falta de políticas para implementar OSS (Falta de leyes, normas, reglamento de derechos de autor o gobierno)	Falta de leyes, normas y reglamentos y derechos de autor (Por desconocimiento del personal de la empresa)	3	3%	
B39			Falta de políticas de Gobierno (Insuficiencia de políticas que apoyen la implementación de FOSS por parte del gobierno)	3	3%	
B40	Problemas (Circunstancia en la que se genera un obstáculo)	Documentación (Obstáculos relacionados a la documentación generada)	Documentación incompleta, no actualizada, obsoleta	1	1%	
B41			Insuficiente documentación	1	1%	
B42	Servicios (Conjunto de actividades que buscan satisfacer las necesidades de un cliente)	Proveedores (Obstáculos con los servicios brindados por los proveedores de FOSS a la organización)	Falta de proveedores de servicio	1	1%	
B43	Seguridad (Disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático)	Mejoras (Conjunto de medidas de cambio que se toman en una organización para mejorar su rendimiento)	Problemas al encontrar mejoras de seguridad (Problemas al encontrar parches y al ser código abierto puede ser modificado afectando la seguridad)	4	4%	
B44			Preocupaciones (Inquietud o temor con la seguridad del software)	Temor a puertas traseras (backdoors) en soluciones OSS	1	1%
B45			Complejidad en la evolución del entorno (Principal reto es como evolucionar su entorno sobre todo en cuestiones de seguridad y calidad del sistema)	2	2%	
B46			Clonación de código puede aumentar la productividad, pero ralentizar el software	1	1%	
B47	Software (Equipo lógico que permite realizar tareas específicas)	Calidad (Concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos)	Incertidumbre sobre calidad (Las preocupaciones sobre la calidad y la posterior confianza en la adopción de software desarrollado a través del proceso de código abierto)	2	2%	
B48			Migración (Transferencia de materiales digitales de un origen de datos a otro)	Migración de software (un proceso de cambio de un software a otro, o bien de un sistema operativo a otro).	4	4%
B49			Producto (Opción elegible, viable y repetible que	Calificación y selección de OSS	1	1%
B50				Crear en el OSS como un producto no lo suficiente experto para actividades comerciales	1	1%
B51	Complejidad de uso	1		1%		

B52		ayuda a realizar un trabajo)	Falta de apoyo a los productos	1	1%
B53			Falta de confianza o fiabilidad en el software final	1	1%
B54			Inmadurez en las aplicaciones finales	1	1%
B55			Inestabilidad del ciclo de liberación de producto	1	1%
B56			Problemas de instalación	1	1%
B57			Versatilidad del sistema operativo instalado	1	1%
B58		Repositorios de código (Sitio centralizado donde se almacena y mantiene información digital)	No se utilizan para colaboración	1	1%
B59		Versiones (Diversos cambios que se realizan sobre elementos de algún producto)	Proliferación de versiones (Confusión al contar con demasiadas versiones)	1	1%
B60	Soporte (Ayuda que se pueda administrar a la organización con respecto al FOSS)	Soporte de la comunidad (Ayuda que presta la comunidad a la organización)	Falta de soporte de la comunidad o desarrolladores (Existe poca ayuda por parte de la comunidad de FOSS)	5	5%
Totales				108	100%

Tabla 5.6 Barreras que impiden adoptar software libre (Autoría propia, 2017).

La Figura nro. 5.58, hace referencia a la Tabla nro. 5.6, mencionada anteriormente; en donde, en la gráfica se presentan los nombres de las barreras con más de 1 incidencia y las barreras con 1 incidencia se denominan “Otros”.

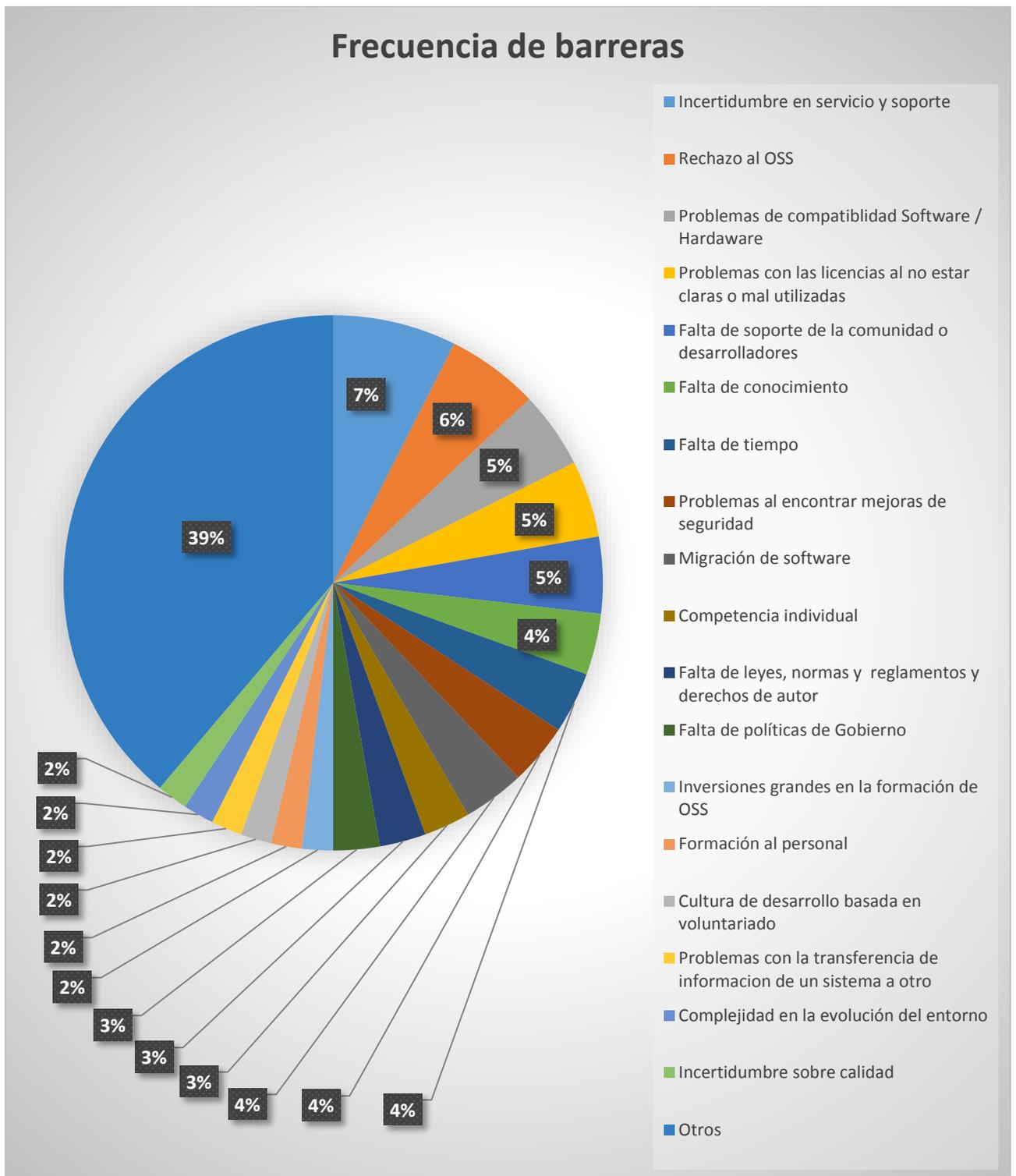


Figura 5.58 Representación gráfica de menciones de las barreras que impide adoptar software libre (Autoría propia, 2017).

Partiendo de la información expuesta anteriormente en la Tabla nro.5.6, y Figura nro. 5.58, se puede decir que la principal barrera es la *“Incertidumbre en servicio y soporte”*, con un 7% de incidencias; seguido por: *“Rechazo al OSS”* con un 6%; *“Problemas de compatibilidad de Software / Hardware”*, *“Problemas con las licencias al no estar claras o mal utilizadas”* y *“Falta de soporte de la comunidad o desarrolladores”* con un 5%; *“Falta de conocimiento”*, *“Falta de tiempo”*, *“Problemas al encontrar mejoras de seguridad”*, *“Migración de software”*, con un 4%; con un 3% *“Competencia individual”*, *“Falta de leyes, normas, reglamentos y derechos de autor”* y *“Falta de políticas de Gobierno”*; con un 2% se encuentran las barreras de *“Inversiones grandes en formación de OSS”*, *“Formación personal”*, *“Cultura de desarrollo basada en voluntariado”*, *“Problemas con la transferencia de información de un sistema a otro”*, *“Complejidad en la evolución del entorno”*, *“Incertidumbre sobre calidad”*; mientras que las demás barreras se encuentran con un 1% de incidencias, sumando 39% del total; estas barreras serán descritas a continuación con más profundidad:

- **Categoría “Compatibilidad”**: subcategoría *“Problemas”*, existen 5 artículos que mencionan que un impedimento para adoptar software libre son los *“Problemas de compatibilidad de software/hardware”*, esto hace referencia de como la tecnología actual, puede prevalecer; sobre otras tecnologías, que ya fueron adoptadas previamente por la organización; así como también, la tecnología moderna debe adaptarse a los sistemas informáticos que son antiguos; ya que siguen útiles para la entidad adoptante.
- **Categoría “Costes”**: subcategoría *“Problemas económicos”*, de los 4 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre es el *“Aumento de costes en la organización”* y la *“Dificultad para estimar el coste de la propiedad del software”*, según Rossi *et. al* (2012), tanto el aumento y la dificultad de costes imposibilitan estimar el recurso necesario para la adopción del FLOSS, añadido a esto, está la duda de que si habrá o no gastos ocultos.
Otro artículo indica que una barrera son los *“Costes no recuperables”*, según Nagy *et. al* (2010), resaltan que hay varias organizaciones que en la actualidad usan software propietario y al adoptar soluciones de código abierto muchas de

estas entidades deberán optar por dejar de utilizar software propietario, lo que tendría un efecto negativo al no poder recuperar lo invertido a lo que el autor lo llama como “Coste irrecuperable”, y esto es una de las razones por la que las organizaciones se muestran reacias a la adopción del OSS, debido a que los altos mandos de la empresa, siempre piden informes de los costes de las inversiones actuales que deberán justificarse; mientras que otros 2 artículos señalan que son las “Inversiones grandes en la formación de OSS”, son barreras en la adopción de OSS.

En la Figura nro. 5.59, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Costes” y subcategoría “Problemas económicos”.

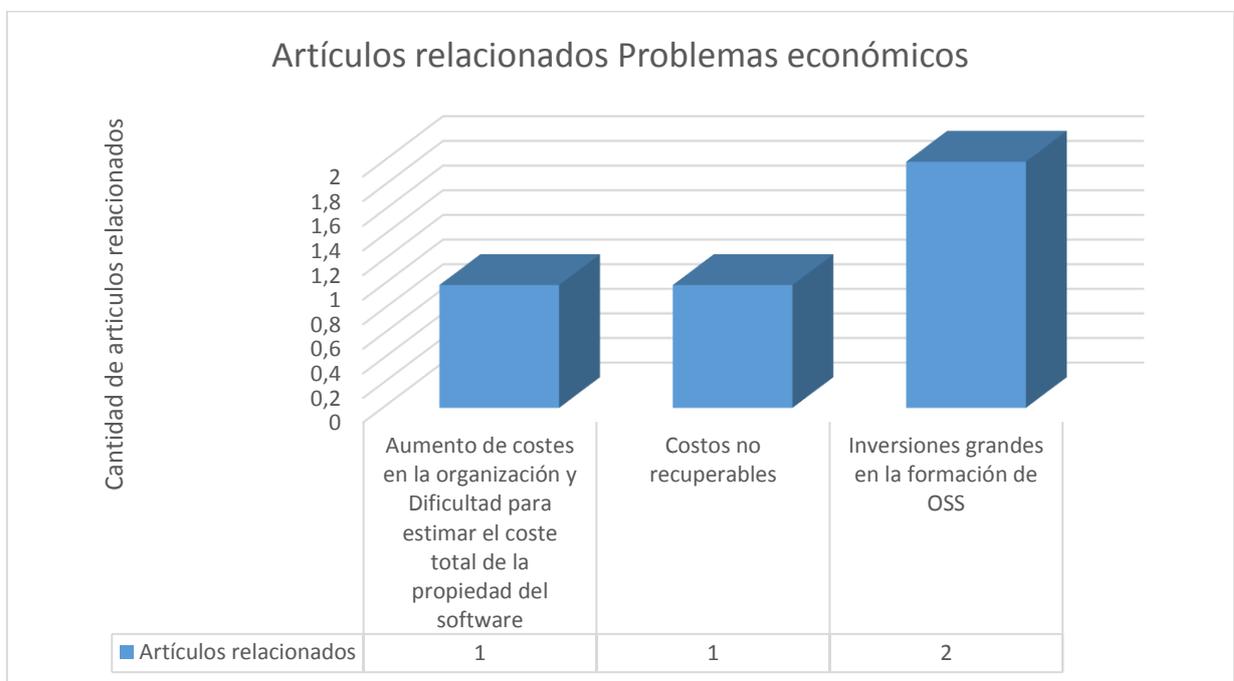


Figura 5.59 Artículos que hacen mención a categoría “Costes” y subcategoría “Problemas económicos” (Autoría propia, 2017).

- **Categoría “Factores humanos”:** subcategoría “Capacitación”, existen 2 artículos que mencionan que un impedimento para adoptar software libre es la “Formación de personal”.

En la categoría “Factores humanos” existe otra subcategoría, llamada “Desarrolladores”, de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre

es la “*Bifurcación (Desarrollo de software escrito por diferentes grupos)*” según Nagy *et. al* (2010), indican que el software que empezó con un código fuente original y al ser desarrollado por varios grupos independientes, existe la posibilidad de que cada uno de los grupos, al no trabajar conjuntamente enfocados en el mismo objetivo desarrollen una versión distinta de la misma solución de software; además, añaden que el problema de la “*Bifurcación*” ya ha tenido efectos en el pasado; desintegrando comunidades de código abierto y por lo tanto la bifurcación es un impedimento, ya que en ~~e~~ la mayoría de los casos el adoptante espera contar con una versión de software estable y dominante antes de proceder con la adopción de OSS.

Otro artículo señala que las “*Expectativas de trabajo poco claras, no se sabe el tiempo que se dispone en la comunidad*” y la “*Falta de roles formales*”, son barreras de adopción; sin embargo, en 1 artículo se señala que es la “*Falta de entendimiento de la estructura o arquitectura del proyecto*”, es otra barrera en la adopción de OSS.

En la categoría “Factores humanos” y subcategoría “*Equipos colaborativos*”, de los 2 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que, un impedimento para adoptar software libre es “*La falta de orientación a los nuevos miembros*” y la “*Falta de personal que conduce a bajos rendimientos*”; mientras que otro artículo señala que la “*Sobrecarga de comunicación con equipos colaborativos grandes*”, son barreras.

En la categoría “Factores humanos” y subcategoría “*Experiencia*”, de los 6 artículos encontrados que hacen referencia a esta subcategoría: 2 artículos mencionan que un impedimento para adoptar software libre es la “*Cultura de desarrollo de software basada en voluntariado*”, ya que los integrantes de una comunidad dedicada al desarrollo de software no poseen contratos que se deben cumplir; si no que, trabajan por voluntad propia y libre, con la libertad de abandonar el proyecto cuando deseen; mientras que otros 4 artículos señalan que es la “*Falta de conocimiento (experiencia) en OSS*”, es una barrera grande en la adopción de OSS.

En la Figura nro. 5.60, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Factores Humanos” y subcategoría “Experiencia”.

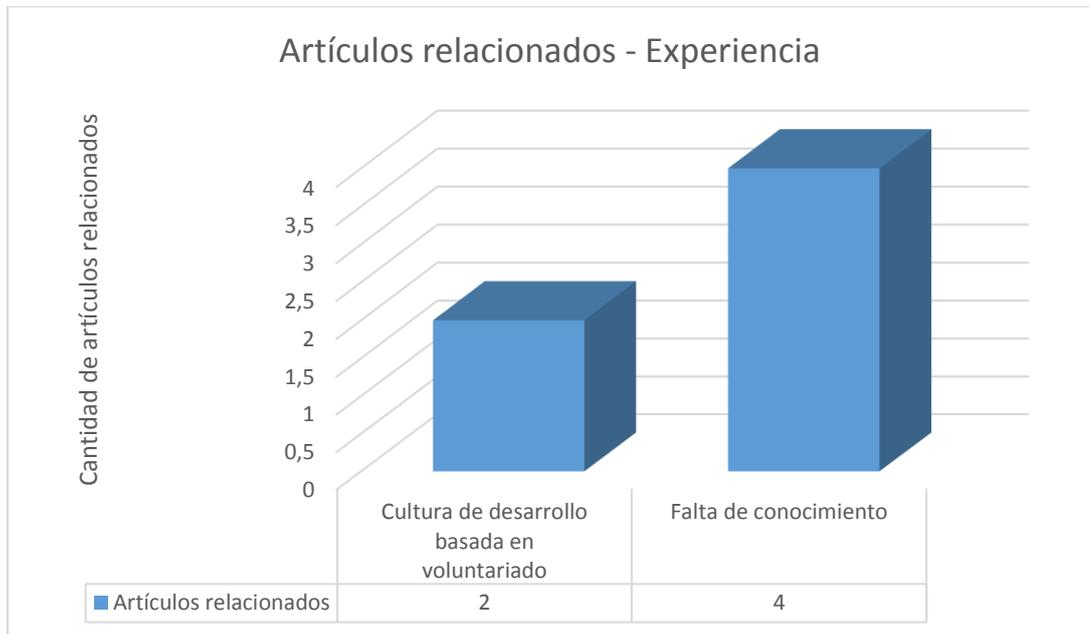


Figura 5.60 Artículos que hacen mención a la categoría “Factores Humanos” y subcategoría “Experiencia” (Autoría propia, 2017).

En la categoría “Factores humanos” y subcategoría “Incertidumbre”, de los 10 artículos encontrados que hacen referencia a esta subcategoría: 8 artículos mencionan que un impedimento para adoptar software libre es la “Incertidumbre en servicio y soporte”, entre estos artículos también se indica que otro impedimento es “Poner a disposición el código fuente”, según Morgan & Finnegan (2007), indican que la divulgación del código detiene el proceso de adopción del OSS, ya que algunos de los gerentes muestran cierta desconfianza sobre el tema.

Además, dentro de la misma categoría y subcategoría, 1 artículo indica que es la “Incertidumbre legal”, es una barrera de adopción; mientras que otro artículo, se refiere como barrera a la “Inmadurez tecnológica”, donde Nagy *et. al* (2010), comenta que las organizaciones son reacias a adoptar OSS, ya que piensan que la tecnología no ha madurado lo suficiente para el ámbito comercial o simplemente creen que la calidad de este tipo de software es inferior por no tener costo; sin embargo, los autores expresan que el temor de la “Inmadurez tecnológica” debería desaparecer, ya que existen organizaciones serias que están invirtiendo en este

tipo de tecnología como: IBM, Hewlett-Packard y Sun Microsystems. Además, existen varias soluciones populares que tienen millones de descargas.

En la Figura nro. 5.61, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Factores Humanos” y subcategoría “Incertidumbre”.

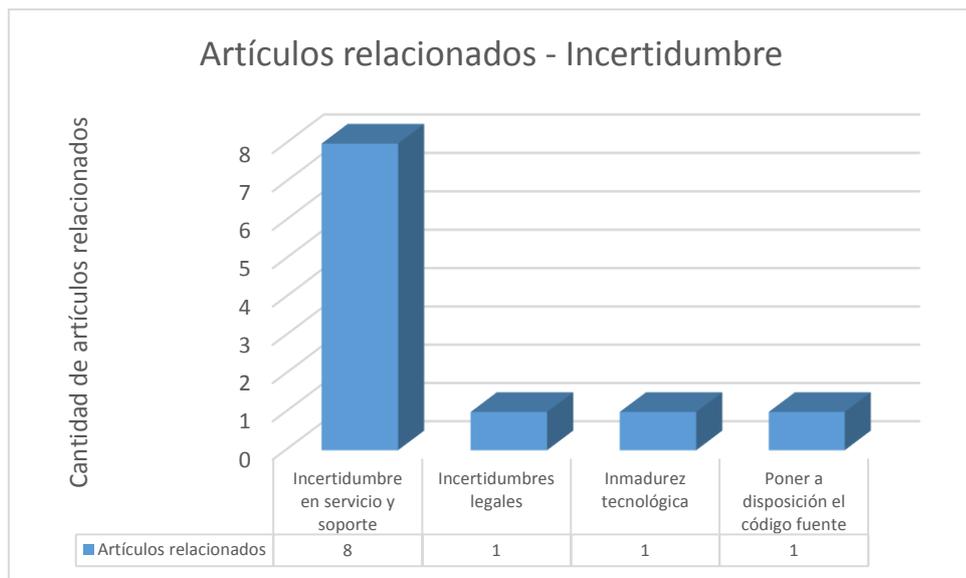


Figura 5.61 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Incertidumbre” (Autoría propia, 2017).

En la categoría “Factores humanos” y subcategoría “Insuficiencia de personal”, existen 3 artículos que mencionan que un impedimento para adoptar software libre es la “Competencia individual”.

En la categoría “Factores humanos” y subcategoría “Motivación”, de los 7 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre es la “Falta de motivación por parte de la comunidad”; mientras que, otros 6 artículos indican como barrera al “Rechazo al OSS”, según Russo *et. al* (2003), las personas se muestran reacias a abandonar las tecnologías que están acostumbradas a usar.

En la Figura nro. 5.62, se presenta el número de artículos que hacen mención a cada barrera en la categoría “*Factores Humanos*” y subcategoría “*Motivación*”.

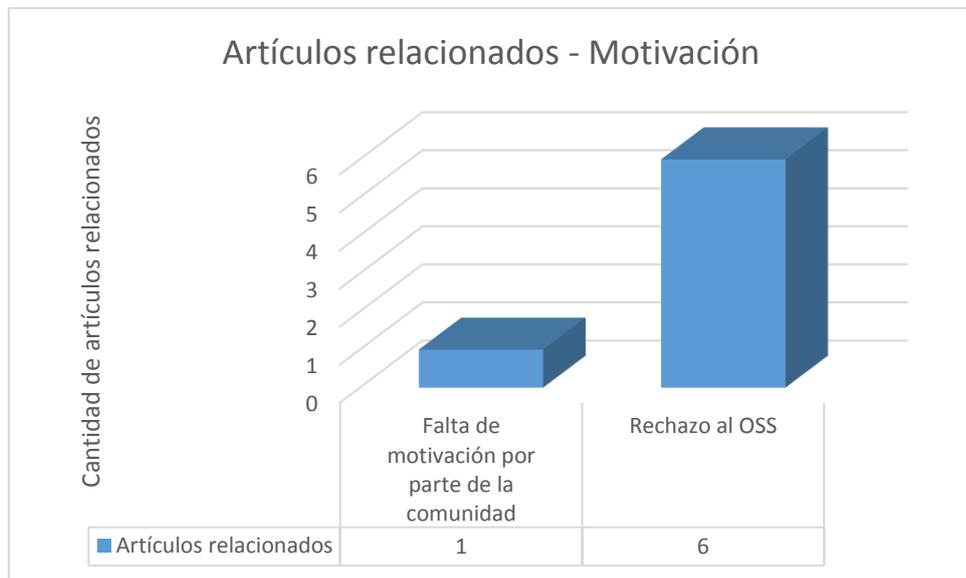


Figura 5.62 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Motivación” (Autoría propia, 2017).

En la categoría “Factores humanos” y subcategoría “*Tiempo del personal*”, de los 5 artículos encontrados que hacen referencia a esta subcategoría: 4 artículos mencionan que un impedimento para adoptar software libre es la “*Falta de Tiempo*”; mientras que 1 artículo indica que es la “*Reducción de coordinación en tiempo real*”, es otra barrera para impedir la adopción de OSS.

En la Figura nro. 5.63, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Factores Humanos” y subcategoría “Tiempo del personal”.

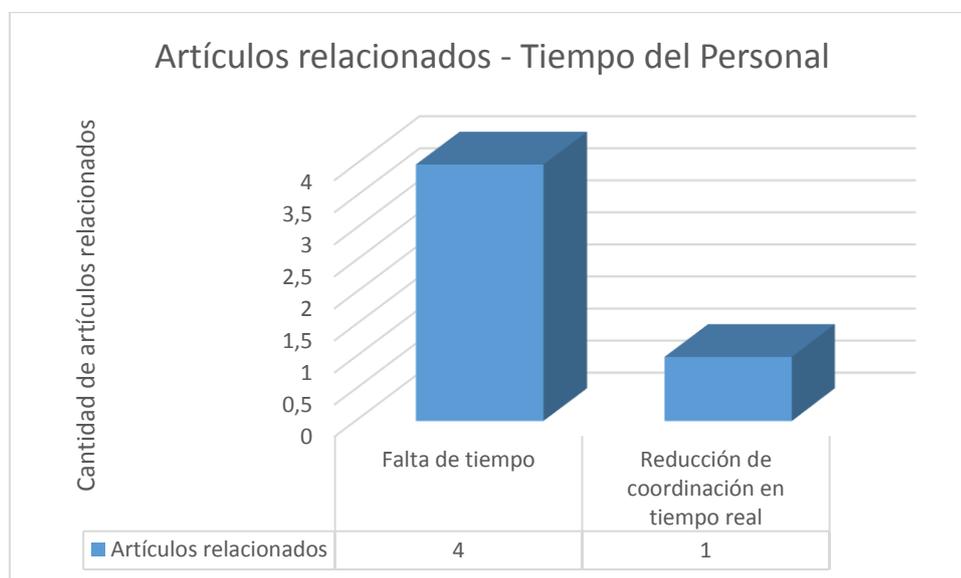


Figura 5.63 Cantidad de artículos que hacen mención a las barreras en la categoría “Factores Humanos” y subcategoría “Tiempo del personal” (Autoría propia, 2017).

- **Categoría “Integración”:** subcategoría “Complejidad”, existe 1 artículo que menciona que un impedimento para adoptar software libre es la “Complejidad de sistemas de gestión”.

En la categoría “Integración” y subcategoría “Integrar información”, existen 2 artículos que mencionan que un impedimento para adoptar software libre son los “Problemas con la transferencia de información de un sistema a otro”.

- **Categoría “Licencias”:** subcategoría “Problemas de licencia”, existen 5 artículos que mencionan que un impedimento para adoptar software libre son los “Problemas con las licencias al no estar claras o ser mal utilizadas”.
- **Categoría “Organizaciones”:** subcategoría “Comercial”, de los 3 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre es la “Existencia de acuerdos con organizaciones comerciales”; mientras que en otros 2 artículos se menciona que,

la “*Falta de interacción social*” y la “*Insuficiente comercialización*”, son barreras en la adopción.

En la categoría “*Organizaciones*” y subcategoría “*Empresa*”, de los 4 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre es la “*Ausencia de adopción de OSS*”; otro artículo indica como obstáculo a la “*Falta de apoyo de la dirección*”, en 1 artículo más se explica que la “*Falta de modelos de negocio*” y la “*Falta de marca de FLOSS*”, son barreras y finalmente en 1 artículo más, se hace referencia como impedimento a la “*Falta de voluntad de los gobiernos y las empresas para modernizar las aplicaciones heredadas (obsoletas)*” y a “*La posición competitiva de la organización*”.

En la categoría “*Organizaciones*” y subcategoría “*Pública*”, existe 1 artículo que menciona que unos impedimentos para adoptar software libre es el “*Sector Público que es impulsado por las directivas gubernamentales*”, según Rossi *et. al* (2012), menciona en su estudio, que la adopción de FLOSS es diferente en el sector público, comparado con el privado; ya que, en el sector público la mayoría de las veces la implementación de nueva tecnología se sustenta en las directivas gubernamentales; mientras que en el sector privado la adopción se lo hace por decisiones de los altos mandos que desean innovar.

- **Categoría “*Políticas*”**: subcategoría “*Falta de políticas para implementar OSS*”, de los 6 artículos encontrados que hacen referencia a esta subcategoría: en 3 artículos se menciona que un impedimento para adoptar software libre es la “*Falta de leyes, normas y reglamentos, licenciamiento y derechos de autor*”; mientras que 6 artículos indican que es la “*Falta de políticas de Gobierno*”, son obstáculos en la adopción de OSS.

En la Figura nro. 5.64, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Políticas” y subcategoría “Falta de políticas para implementar OSS”.

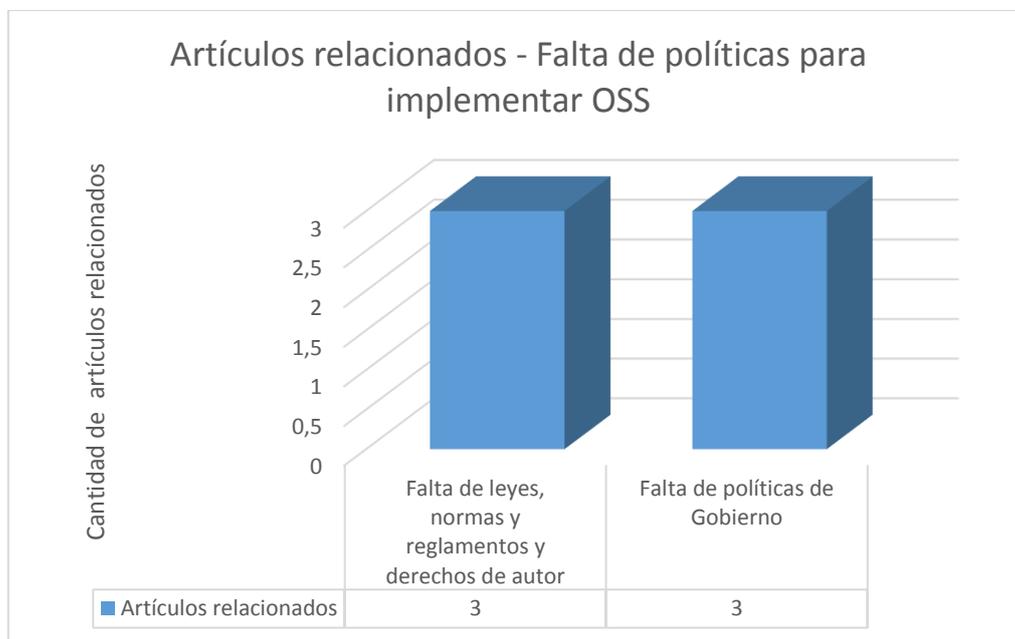


Figura 5.64 Cantidad de artículos que hacen mención a las barreras en la categoría “Políticas” y subcategoría “Falta de políticas para implementar OSS” (Autoría propia, 2017).

- **Categoría “Problemas”:** subcategoría “Documentación”, de los 2 artículos encontrados que hacen referencia a esta subcategoría, 1 artículo menciona que un impedimento para adoptar software libre es la “Documentación incompleta, no actualizada, obsoleta”, y otro la “Insuficiente documentación”.
- **Categoría “Servicios”:** subcategoría “Proveedores”, existe 1 artículo que menciona, que un impedimento para adoptar software libre es la “Falta de proveedores de servicios”, según Maldonado (2010), argumenta en su estudio denominado “El proceso de introducción de FLOSS en la administración pública: el caso de Venezuela”, que la inexistencia de proveedores de servicios afecta la adopción de FLOSS en Venezuela, debido a que no hay un sector que esté competente en temas de adopción software.

- **Categoría “Seguridad”**: subcategoría “Mejoras”, existen 4 artículos que mencionan que un obstáculo para adoptar software libre son los “Problemas al encontrar mejoras de seguridad”.

En la categoría “Seguridad” y subcategoría “Preocupaciones”, de los 4 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que un impedimento para adoptar software libre es el “*Temor a puertas traseras (backdoors) en soluciones OSS*”, otros 2 artículos señalan a la “*Complejidad en la evolución del entorno en cuestiones de seguridad*” y 1 la “*Clonación de código puede aumentar la productividad, pero ralentizar el software*”, como barreras de adopción de OSS.

En la Figura nro. 5.65, se presenta el número de artículos que hacen mención a cada barrera en la categoría “Seguridad” y subcategoría “Preocupaciones”.

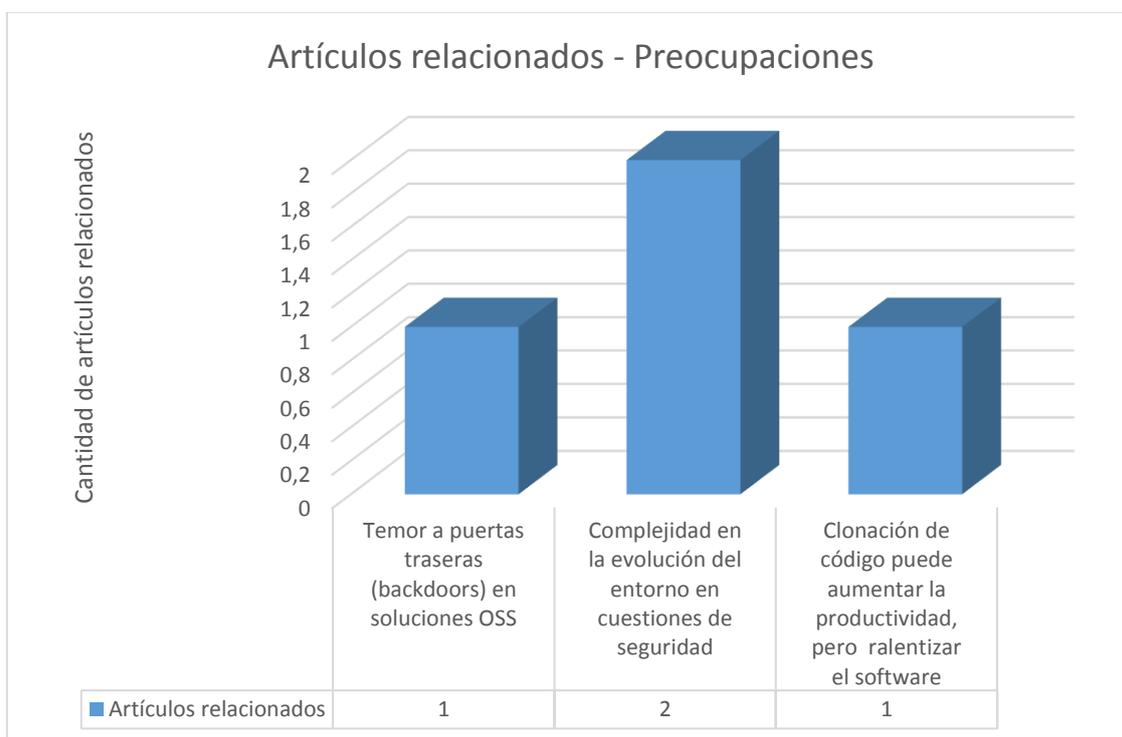


Figura 5.65 Cantidad de artículos que hacen mención a las barreras en la categoría “Seguridad” y subcategoría “Preocupaciones” (Autoría propia, 2017).

- **Categoría “Software”**: subcategoría “Calidad”, existe 2 artículos que mencionan que un impedimento para adoptar software libre es la “*Incertidumbre sobre la calidad*”.

En la categoría “Software” y subcategoría “Migración”, existe 4 artículos que mencionan que un impedimento para adoptar software libre es la “*Migración de software y de datos*”,

En la categoría “Software” y subcategoría “Producto”, de los 8 artículos encontrados que hacen referencia a esta subcategoría: 1 artículo menciona que una barrera para adoptar software libre es la “*Calificación y selección de OSS*”, 1 artículo señala como obstáculo para la adopción el “*Crear en el OSS como un producto no lo suficiente experto para actividades comerciales*” y la “*Falta de apoyo a los productos OSS*”; sin embargo otros 5 artículos indican como barrera de adopción a la “*Facilidad de uso*” y los “*Problemas de instalación*”; a la “*Falta de confianza o fiabilidad en el software final*”, a la “*Inmadurez en las aplicaciones finales*”, a la “*Inestabilidad del ciclo de liberación de producto*”, y a la “*Versatilidad del sistema operativo instalado*”.

En la categoría “Software” y subcategoría “Repositorios de código”, existe 1 artículo que menciona que un impedimento para adoptar software libre es “*La falta de utilización de repositorios para la colaboración*”.

En la categoría “Software” y subcategoría “Versiones”, existe 1 artículo que menciona que un impedimento para adoptar software libre es la “*Proliferación de versiones*”.

- **Categoría “Soporte”**: subcategoría “Soporte de la comunidad”, existe 5 artículos que mencionan que un impedimento para adoptar software libre es la “*Falta de soporte de la comunidad o desarrolladores*”, según Choi & Chengalur-Smith, (2009), argumentan que el soporte es algo clave para la adopción de OSS; sin embargo ha existido duda, relacionado al soporte que será brindado, otro argumento son los insignificantes esfuerzos realizados para tratar este asunto, lo

que trae como consecuencia que las organizaciones dejen de lado la implementación de OSS.

5.2.6. ¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?

En base a la lectura de los 85 artículos, ningún artículo ha mencionado si la documentación que genera la comunidad desarrolladora de software libre es suficientemente comprensible para la implementación eficiente; sin embargo, ciertos artículos que se refieren a documentación indican lo siguiente:

- Gerber *et al.* (2010), señalan que la disponibilidad de documentación de OSS es limitada, lo cual detiene la adopción y migración de OSS. El autor también recalca que no existe ningún tipo de información seria para la adopción.
- Gary *et al.* (2009), indican que la documentación disponible para la adopción de OSS es insuficiente e inexistente.
- Morgan & Jensen (2014), indican que un impedimento para los recién llegados a proyectos de OSS, es la insuficiente o no actualizada documentación que el mismo genera.
- Ademolu *et al.* (2014), mencionan que la documentación en los proyectos de OSS es deficiente, debido a que no es obligación legal, la producción de documentación en estos proyectos.

5.2.7. ¿Se ha incluido la opinión del usuario del uso del software libre?

En base a la lectura de los 85 artículos, ningún artículo ha mencionado si se ha tomado en cuenta la opinión del usuario al momento de implementar OSS en la organización; sin embargo, según Lyn (2009), señala que es importante investigar al usuario, comprendiendo sus necesidades y la relación técnica o funcionales que este tenga con el producto.

Capítulo 6. Discusión de resultados obtenidos

6.1. Introducción

Este capítulo presenta la discusión de los principales hallazgos encontrados en este estudio, también se describen las posibles amenazas a la validez de la revisión sistemática de literatura realizada.

6.2. Principales hallazgos

Producto de la revisión sistemática de literatura de la adopción de FOSS, de los 85 estudios identificados, se ha podido observar que casi en su totalidad carecen de soporte de evidencia empírica, o al existir dicha evidencia no es suficiente, ya que, por lo general dichos artículos no presentan información primaria de soporte, ej., artículos que mencionan la reducción de costos como ventaja del FOSS, no presentan cuadros de valores ni otro tipo de datos que prueben lo expuesto.

Entre los escasos estudios que presentan información un poco más detallada se pude citar el artículo de Fitzgerald & Kenny (2003), sobre “La experiencia del Hospital Beaumont”, la cual menciona ahorros de 13 millones de Euros en un lapso de 5 años producto de la adopción de OSS. La decisión de adoptar OSS en el hospital surgió del escaso presupuesto, y por el hecho que contaban con personal con cierta experiencia en OSS, el cual no tenía miedo al cambio. Se puede mencionar algunos aspectos en el estudio antes mencionado en relación el ahorro de costes:

- Si bien en el estudio se muestra una tabla donde se hace una comparación de costos iniciales y costos totales en un periodo de 5 años entre las soluciones de software de código abierto y el software propietario, solo se muestra valores generales de ahorros, no presenta evidencia de manera detallada; sin embargo entre los hallazgo está, que debido al presupuesto limitado, la institución buscaba alternativas de bajo costo no necesariamente soluciones FOSS, ya que el hospital se basaba en políticas de mercado mixto, para la adquisición de software, dichos aplicativos debían estar disponibles y ser beneficiosos para la institución. Es por ello por lo que el hospital había implementado varios servidores con sistemas operativos: Red Hat o SuSE Linux, HP Unix y Microsoft Windows NT. Esto deja

en evidencia que puede coexistir tanto el software libre como el software propietario dentro de una misma institución.

- Entre las aplicaciones que se menciona que hubo un ahorro están:
 - Sistema operativo Red Hat Linux.
 - Aplicaciones de escritorio como: Star Office, SuSE mail.
 - Sistema de Gestión de Contenidos (CMS) basado en Zope.
 - Servidor de aplicaciones de código abierto WildFly o también conocido JBoss.
 - E-Mail: SuSE Mail.
 - Herramienta E-Learning: Claroline.

Aunque las soluciones OSS adquiridas por el hospital eran gratuitas, tenían costos iniciales, ya que la institución debió pagar consultorías de instalación de diversos productos, entre los que se cuentan JBoss y el Sistema de Gestión de Contenidos, que fueron implementados por Open App, una organización con experiencia en código abierto, que se especializa en negociaciones de soluciones y consultoría.

Además de lo antes expuesto, otros autores como: Hamid et al. (2016), señalan que existe ahorro económico al no depender de proveedores en el sector público; Haider (2009), menciona como principal beneficio económico el coste de adquisición que es menor al de software propietario. Por el contrario, autores como: Rossi et al. (2012), Kovacs et al. (2004), señalan que la adopción de FOSS trae consigo aumento de costos de organización, entre los que se incluyen el aumento de los costes de gestión y los nuevos costos de cumplimiento de software, que dificultan una estimación precisa del recurso necesario para la adopción, produce temor a gastos imprevistos y costos ocultos y que genera gastos en formación del personal para las nuevas herramientas.

Partiendo de lo expuesto por los autores, en relación a costes en adopción de FOSS, es necesario investigar más acerca del tema, porque no se presenta la suficiente documentación para concluir si produce beneficios económicos o gastos, teniendo en cuenta los costos ocultos que puede generar la adopción de OSS, también se evidencia que, aunque los productos FOSS sean gratuitos lo más probable es que se tenga que incurrir en costos relacionados a consultorías para su instalación, servicios de mantenimiento y capacitación en el uso y manejo de los distintos productos, más aún si no se cuenta con personal con experiencia en FOSS.

Además, existen otros hallazgos que conviene mencionar:

- Silic & Back (2017), en su artículo “Open Source Software Adoption: Lessons from Linux in Múnich”, señalan que un riesgo de los sistemas operativos Linux son los “Problemas con valores reales generados por los costes de OSS”, refiriéndose a los gastos que ocasionan los consultores externos, ya que son mucho más altos de lo esperado; también señalan como riesgo a las “Vulnerabilidades de seguridad”, ya que el código fuente está disponible para cualquier persona, incluso para actores maliciosos que puedan cambiar estas fuentes e inyectar programas potencialmente peligrosos, por lo cual deberían ser revisados y evaluados previa su adopción. Aun así, la adopción de Linux en el proyecto resultó exitosa, lo cual concuerda con lo descrito en, Oreku (2010), Khelifi *et al.* (2009), Kovacs *et al.* (2004), Petrijevcanin Vuksanovic & Sudarevic (2012), Ven Verelstt *et al.* (2008), Sen (2007), Comino & Manenti (2005), quienes indican que la adopción de Linux ha sido beneficiosa para las organizaciones, permitiendo ahorrar costes, aumentar la seguridad y la flexibilidad del software. Por lo tanto, está claro que Linux es un sistema operativo que trae muchas ventajas en su adopción, presentando riesgos como cualquier otro producto.
- Gary *et al.* (2009), señalan que uPortal, al igual que muchos proyectos de código abierto, no proporciona una solución de extremo a extremo. La documentación es escasa, y los procesos de ejecución no son competitivos frente a ofertas de portales comerciales; mientras Sahraoui (2009), lo señala como uno de los proyectos mas destacados de FOSS en las universidades. Partiendo de esto se podría decir que este softwarees una solucion atractiva en el ámbito educativo, teniendo en cuenta que es necesario conocer la herramienta y tener un entrenamiento previo adecuado.
- La seguridad es uno de los aspectos mas contradictorios en relacion a la adopcion de FOSS. Por un lado, Lawton (2002), señala que al eliminar componentes innecesarios se reduce el impacto en problemas de seguridad

y que es poco probable que los parches afecten a los sistemas operativos estables como es el caso de Linux; Kamau & Namuye (2012), señalan que las ofertas de FOSS entregan beneficios de seguridad altos, por lo cual es apoyado cada vez más por organizaciones no gubernamentales internacionales; Haider (2009), menciona que los beneficios que ofrece el software de código abierto respecto a seguridad, son mejores que las de ofertas propietarias; también Morgan & Finnegan (2007), indican que los productos FOSS entregan alta seguridad, debido a la disponibilidad del código fuente, la reducción de virus y el conocimiento adicional de seguridad en la fase de diseño. Por otro lado, algunos autores mencionan que la adopción de FOSS presentan diversos riesgos de seguridad. Ardagna *et al.* (2009), mencionan que existen problemas de seguridad y licencias; Al-Ajlan (2009), señala que el principal reto en FOSS es encontrar mejoras de seguridad y calidad; Koch (2004), argumenta que es complicado encontrar mejoras de seguridad, debido que al liberar el código, puede ser modificado y no se indican que se realizaron cambios que pueden afectar a la seguridad. Partiendo de la información obtenida, en términos de seguridad la adopción de OSS es muy delicada, ya que mientras para muchos autores es una fortaleza, para otros representa una barrera de adopción, por lo cual el adoptante debería analizar los problemas de actualizaciones y la confianza que entregue la comunidad que apoya al proyecto.

- La disponibilidad y personalización del código fuente, es otro de los aspectos en los que se ha identificado contradicción. Mientras los autores Maki-Asiala & Matinlassi (2006), Xie (2008), Ademolu *et al.* (2014), Maldonado (2010), señalan que al tener acceso al código fuente se reduce claramente la dependencia de proveedores, permite que las soluciones OSS sean compatibles al poder alterarlas de acuerdo a sus necesidades y fomenta la innovación al producir nuevas ideas, otros autores como, Morgan & Finnegan (2007), mencionan que algunos se sienten incómodos al liberar el código fuente, por lo cual lo ven como una barrera al adoptar FOSS. Fendt *et al.* (2016), señalan que a pesar de la correcta verificación de código es imposible evitar la integración de un software externo no aprobado por los desarrolladores, porque el código fuente se lo puede copiar y pegar con

facilidad como si fuera propio. Silic & Back (2017), señalan que al encontrarse disponible el código fuente para cualquiera, puede ser editado por actores maliciosos e inyectar programas potencialmente peligrosos. Por lo tanto al adoptar FOSS es importante tener en cuenta que el código fuente debe ser entregado a la comunidad y al utilizar nuevas actualizaciones de código se debería revisar y evaluar la procedencia de ser posible, para evitar problemas con la seguridad y calidad del mismo.

- El factor de éxito más mencionado por los autores es la cooperación, como lo indican, Noll (2011), Deen Sethanandha (2010), Krishnamurthy *et al.* (2016), Koch (2004), quienes consideran la cantidad de actividad asociada por parte de la comunidad como factor de éxito en proyectos de adopción de FOSS. Por su parte, Silic & Back (2017), señalan como un factor de fracaso la pérdida de interés, por las partes interesadas en el proyecto. Otros autores Fendt *et al.* (2016), Xie (2008) y Sack *et al.* (2006), señalan que la falta de cooperación, apoyo y falta de responsables para soporte a proyectos FOSS introduce un riesgo a esta clase de proyectos. Maki-Asiala & Matinlassi (2006), señalan que es una barrera al momento de adoptar OSS que la cultura de desarrollo este basada en el voluntarismo y no se este sustentada por contratos. Esto introduce un riesgo por la incertidumbre en la calidad de servicio y soporte, tal como lo indican, Choi & Chengalur-Smith (2009), Lawton (2002), Crowston (2011) y Miscione & Johnston (2010). Partiendo de la información obtenida, el apoyo que la comunidad brinde al proyecto es muy importante para que un proyecto de adopción de FOSS sea exitoso; al no existir una comunidad activa que respalde al proyecto se introduce un elevado riesgo de fracaso.

6.3. Amenazas a la validez

Según Kitchenham (2007), y Genero (2014), expresan que, una revisión sistemática puede estar propensa a algunas limitaciones que se deben tener en consideración como: el sesgo en las publicaciones, sesgo en la selección de estudios, discrepancias en la extracción de datos y clasificación, entre otros. Las principales limitaciones en este estudio incluyen:

6.3.1. Validez de constructo

- Es probable que se hayan omitido literatura relevante al tema de estudio, debido a que no se tomaron en cuenta bibliotecas, como es el caso de: Science Direct, Emerald y Scopus, por ello para reducir el impacto de esta amenaza se utilizó la técnica de backward Snowballing.
- Existe la posibilidad que las preguntas de investigación, no cubran toda la información existente al área de estudio, por lo tanto para minimizar este riesgo se aplicó la técnica PICO, con la asesoría de personas con experiencia en el tema.
- No se puede garantizar cubrir toda la información del tema de estudio, porque los términos relevantes tienen más sinónimos, para reducir esta amenaza se emplearon términos de otras revisiones sistemáticas relacionadas a la investigación y también se empleó algunos sinónimos del diccionario

6.3.2. Validez interna

- Algunas decisiones subjetivas pueden haberse presentado durante la selección de los artículos, el análisis de calidad y la extracción de datos como una amenaza a la validez interna, haciendo difícil el uso objetivo de los criterios de inclusión / exclusión o la extracción imparcial de datos. Con el fin de minimizar la selección, el análisis de calidad y los errores de extracción, los procedimientos contemplados en esta SLR se realizaron bajo el enfoque de revisión por pares, y todos los conflictos y pormenores detectados fueron discutidos y resueltos por todos los autores. Incorrecta definición y categorización de variables, este inconveniente se minimizó con la lectura en contexto de todo el documento, para relacionar las definiciones de las variables a lo que habla el artículo.

6.3.3. Validez externa

- Existe la posibilidad que ciertos artículos, no hayan sido tomados en cuenta en la selección de estudios, debido a que los enlaces a los repositorios digitales estaban caídos, dificultando la obtención de dichos artículos, con el afán de reducir esta amenaza, la búsqueda de estos documentos se realizó con la ayuda del motor de búsqueda de Google.

6.3.4. Validez de conclusión

- Falta de conocimiento en técnicas de cruce de información, para su aplicación de acuerdo a los factores identificados en la investigación, esta amenaza se mitigo con la consultoria a expertos en el tema.
- La falta de evidencia detallada en la mayoría de los articulos, dificulta la elaboración de una conclusión fiable, esta amenaza se redujo con la aplicación de la técnica de backward Snowballing, con la cual, se identificaron publicaciones que pudieron ser omitidas en el análisis anterior.

Capítulo 7. Conclusiones y recomendaciones

7.1. Conclusiones

- El objetivo de este trabajo de titulación fue elaborar una revisión sistemática de literatura para identificar el estado del arte sobre los factores de éxito, fracaso, barreras, beneficios, riesgos, dominios exitosos, documentación generada por la comunidad y opinión del usuario al usar FOSS. Para ello se tomó como referente la guía Kitchenham y Charters(Kitchenham, 2007), la cual menciona los pasos para realizar SLRs en el campo de la ingeniería de software.
- Basados en la guía de Kitchenham (Kitchenham, 2007) se elaboró un protocolo que contiene preguntas de investigación, estrategias de búsqueda, criterios de inclusión y exclusión, criterios de calidad, y la estrategia de extracción de datos. Dicho protocolo, fue revisado y validado antes de ejecutar la revisión de literatura y tomado en cuenta durante toda la investigación realizada. El protocolo elaborado puede ser un referente para futuras SRLs que se relacionen al campo de la ingeniería del software.
- Para la identificación de artículos relevantes se hizo uso de 3 bibliotecas digitales (IEEE Xplore, ACM Digital Library, Springer Link) definidas en el protocolo de investigación, A partir de ello se pudo identificar un total 690 publicaciones, seleccionando en primera instancia 165 artículos como potenciales relevantes para el estudio, producto de la lectura del título y resumen. Luego de la preselección con la lectura completa de las publicaciones se obtuvieron 76 artículos, a los cuales se agregó 17 artículos producto de una segunda identificación de estudios mediante la técnica bola nieve, dando como resultado 93 artículos con los que se procedió a evaluar la calidad obteniendo finalmente 85 artículos, que se usaron para extraer la información en base a las preguntas de investigación rellenando el formulario de extracción de datos descrito en el protocolo.

- En la estrategia de búsqueda también se tomó en cuenta la “técnica de bola de nieve hacia atrás” (*backward SnowBalling*), revisando la lista de referencias de los artículos seleccionados las cuales podrían responder a las preguntas de investigación. Se pudo comprobar que con la técnica de bola de nieve es posible obtener literatura relevante la cual no es considerada en las búsquedas automáticas.
- De 85 artículos que aportan a la investigación planteada los cuales están comprendidos entre los años 1998 y 2016, para el análisis de los resultados se realizó una categorización y subcategorización partiendo de la relación de las respuestas obtenidas de las preguntas planteadas en este estudio, de lo cual se pudo obtener 68 factores de éxito de los cuales el más mencionado es el factor de cooperación de la comunidad. Dentro de los factores de fracaso se obtuvo 26 resultados, siendo el más citado la poca competitividad del FOSS frente a ofertas comerciales. Dentro de los riesgos se obtuvo 41 resultados siendo el más nombrado los problemas por derechos intelectuales. Por otra parte, se han encontrado 60 beneficios de los cuales el más citado es la personalización que permite el FOSS. También se ha encontrado 16 dominios exitosos en la implementación del FOSS siendo el más exitoso según la literatura consultada los sistemas operativos. Dentro de las barreras que impiden la adopción de FOSS se encontraron 60 resultados, encontrando como el más mencionado la incertidumbre en servicio y soporte que puede traer la adopción de FOSS; mientras para la documentación generada por la comunidad y la opinión del usuario no se encontraron resultados.
- Luego de haber realizado la revisión sistemática de literatura se puede concluir que faltan más artículos que den evidencia de primera mano, ya que si bien algunos beneficios son mencionados como el ahorro de costos no hay evidencia amplia que lo sustente. Por lo cual se considera que es temprano para concluir que entre los beneficios de la adopción de software libre este el ahorro de costos. También se pudo observar que existen

diversas contradicciones entre los autores, entre los ejemplos más destacados están:

- La disponibilidad de código fuente algunos autores lo mencionan como beneficio otros lo mencionan como una barrera.
 - La cooperación de la comunidad con FOSS es tomada por los distintos autores como: factor de éxito, factor de fracaso, barreras y riesgos.
 - La seguridad ciertos autores de manera distinta lo mencionan como: beneficio, riesgo y barrera.
- Existe varios SLR que hablan de software libre o software de código abierto; sin embargo, están enfocados al desarrollo y comunidades de desarrollo de software libre y no a la adopción de software libre en el ámbito organizacional, por lo cual se puede concluir que dichos SLR se enfocan a aspectos distintos a los considerados en nuestra investigación.

7.2. Recomendaciones

- Se recomienda utilizar la guía de Kitchenham y Charters (2007), ya que esta proporciona las directrices de cómo realizar SLRs claras y estructuradas, en el campo de la ingeniería de software, permitiendo resumir la evidencia empírica encontrada de un tema en concreto.
- Debido a que el diseño del protocolo de investigación requiere esfuerzo y cierta habilidad al ser un trabajo laborioso; pero a su vez importante para emprender una SLR. Se recomienda diseñar el protocolo de investigación basándose en otros protocolos de trabajos relacionados, luego de ello este debe ser revisado y validado por personas expertas antes de emprender la SLR.
- Es importante realizar una investigación del funcionamiento de las bases de datos electrónicas, además se debe tomar en cuenta que las palabras claves empleadas son las correctas, antes de emprender las búsquedas, para asegurar que la documentación extraída sea relevante para el estudio.

- Al momento de realizar la extracción de datos de los artículos, es importante añadir una pequeña descripción a lo extraído, con la finalidad de disminuir problemas de interpretación y tiempo empleado en tabulación de datos.
- La clasificación de la información recopilada se lo ha realizado de acuerdo con términos relacionados; sin embargo, pudiese existir una mejor manera de clasificar dicha información para que sea más comprensible. Por lo cual se recomienda revisar la técnica de análisis de contenido (content analys).
- Se recomienda ampliar el estudio, utilizando otras bibliotecas que no han sido tomadas en cuenta en este trabajo como las de: SCOPUS, LESEVIER, EMERALD. Con la finalidad de encontrar publicaciones que muestren evidencia amplia relacionada a la adopción de FOSS por organizaciones.
- Antes de emprender una nueva SLR se recomienda revisar si existen trabajos relacionados al tema de investigación, con el fin de determinar qué es lo que se ha hecho y que no, en caso de existir un trabajo semejante al que se pretende hacer una opción es realizar un estudio secundario con un alcance más amplio conocido como mapeo sistemático de literatura.

Referencias

- A. Ardagna, C., Banzi, M., Damiani, E., Frati, F., & El Ioini, N. (2009). An assurance model for OSS adoption in next-generation telco environments. *Digital Ecosystems and Technologies, 2009. DEST '09. 3rd IEEE International Conference on* (págs. 619-624). Istanbul, France: IEEE Xplore.
- Achuthan, K., SudhaRavi, S., Kumar, R., & Raman, R. (2014). Security vulnerabilities in open source projects: An India perspective. *2014 2nd International Conference on Information and Communication Technology (ICoICT)* (págs. 18-23). Bandung, Indonesia: IEEE Conference Publications.
- Ademolu Ajigini, O., A. van der Poll, J., & Kroeze, J. H. (2014). Towards a model on security challenges during closed source software to OSS migrations. *The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)* (págs. 274-283). London, UK: IEEE Conference Publications.
- Adewumi, A., Misra, S., Omoregbe, N., Crawford, B., & Soto, R. (2016). A systematic literature review of open source software quality assessment models . *Springer*.
- Ajila, S. A., & Wu, D. (2007). Empirical study of the effects of open source adoption on software development economics. *Journal of Systems and Software*, 1517-1529.
- Al-Ajlan, A. (2009). The Evolution of Open Source Software Using Eclipse Metrics. *2009 International Conference on New Trends in Information and Service Science* (págs. 211-218). Beijing, China: IEEE Conference Publications.
- Alasbali , N., & Benatallah, B. (2015). Open source as an innovative approach in computer science education A systematic review of advantages and challenges. *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on* (págs. 278-283). Amritsar: IEEE.
- Alasbali, N., & Benatallah, B. (2015). Open Source as an Innovative Approach in Computer Science Education - A systematic review of advantages and challenges. *Proceedings of the 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*.
- Årdal, C., Alstadsæter, A., & Røttingen, J.-A. (2011). Common characteristics of open source software development and applicability for drug discovery: a systematic review . *Springer*.
- Barcomb, A. (2015). A multiple case study of small free software businesses as social entrepreneurs. *OpenSym '15 Proceedings of the 11th International Symposium on Open Collaboration* (págs. 1-8). San Francisco, California: ACM Digital Library.
- Biblioteca de Informática y Documentación "Enric V. (12 de Febrero de 2014). ACM Digital Library. Valencia, Valencia, España.
- Breivold, H. P., Chauhan, M. A., & Babar, M. A. (2010). A Systematic Review of Studies of Open Source Software Evolution . *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*.

- Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *The Journal of Systems and Software* 80 (2007) 571–583 (pág. 13). Elsevier.
- Choi, N., & Chengalur-Smith, I. (2009). An Exploratory Study on the Two New Trends in Open Source Software: End-Users and Service. *2009 42nd Hawaii International Conference on System Sciences* (págs. 1-10). Big Island, HI, USA: IEEE Conference Publications.
- Coelho, J., & Valente, M. (2017). Why modern open source projects fail. *ESEC/FSE 2017 Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (págs. 186-196). Paderborn, Germany: ACM.
- Comino, S., & Manenti, F. (2005). Government Policies Supporting Open Source Software for the Mass Market. *Springer Link*, 217-240.
- Correa, R. (2008). *Decreto presidencial 1014*. Quito: Gobierno de la República del Ecuador.
- Crowston, K. (2011). Lessons from Volunteering and Free/Libre Open Source Software Development for the Future of Work. En M. Chiasson, & O. Henfridsson, *Researching the Future in Information Systems* (págs. 215-229). Turku, Finland: IFIP International Federation for Information Processing.
- Deen Sethanandha, B., Massey, B., & Jones, W. (2010). Managing open source contributions for software project sustainability. *PICMET 2010 TECHNOLOGY MANAGEMENT FOR GLOBAL ECONOMIC GROWTH* (págs. 1-9). Phuket, Thailand: IEEE Conference Publications.
- Dybå, T., Dingsøyr, T., & Hanssen, G. (2007). Applying Systematic Reviews to Diverse Study Types: An Experience Report. *First International Symposium on Empirical Software Engineering and Measurement* (pág. 10). Madrid: IEEE.
- Ebert, C. (2007). Open Source Drives Innovation. *IEEE Journals & Magazines*, 105-109.
- Escribano Barreno, J., García Muñoz, J., & García Valls, M. (2016). Integrated Metrics Handling in Open Source Software Quality Management Platforms. *Information Technology: New Generations* (págs. 509-518). Cham: Springer.
- Facultad de Filosofía y Letras: UBA. (2005). Introducción a SpringerLink. Púan, Buenos Aires, Argentina.
- Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. R. (2007). *Perspectives on Free and Open Source Software*. London: MIT Press.
- Fendt, O., Jaeger, M., & Jimenez, R. (2016). Industrial Experience with Open Source Software Process Management. *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)* (págs. 180-185). Atlanta, GA, USA: IEEE Conference Publications.
- Fitzgerald, B., & Kenny, T. (27 de Noviembre de 2003). Open Source Software can Improve the Health of the Bank Balance - The Beaumont Hospital Experience. *ResearchGate*, 1-20. Obtenido de <https://pdfs.semanticscholar.org/0932/1069fa68113bd74f935278d8dedc3716ab62.pdf>

- Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, 587-598.
- Free Software Foundation. (7 de Febrero de 2013). *static.fsf*. Obtenido de <http://static.fsf.org/common/what-is-fs-es.pdf>
- Gary, K., Koehnemann, H., Blakley, J., Goar, C., Mann, H., & Kagan, A. (2009). A Case Study: Open Source Community and the Commercial Enterprise. *Sixth International Conference on Information Technology: New Generations* (págs. 940-945). Las Vegas: IEEE Conference Publications.
- Gatta, R., Abeni, F., Buglione, M., Peveri, A., Barbera, F., Tonoli, S., . . . Magrini, S. M. (2007). Open-source, low-cost, high-reliability solutions for digital imaging systems: Example of a "dicom router". *La radiologia medica*, 12252-1259.
- Genero Bocco, M., Cruz-Lemus, J., & Mario, P. (2014). *Métodos de investigación en ingeniería del software*. Madrid: RA-MA.
- Gerber, A., Molefe, O., & Merwe, A. v. (2010). Documenting open source migration processes for re-use. *SAICSIT '10: Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists* (págs. 75-85). Bela Bela, South Africa: ACM.
- Haider, A. (2009). Open Source Software Utilization in Australian State Governments . *20th Australasian Conference on Information Systems* (págs. 743-753). Melbourne : Association for Information Systems AIS Electronic Library .
- Hamid, A., Abdullah, N. L., & Idrus, R. (2016). Framework for successful Open Source Software implementation in the Malaysian Public Sector. *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)* (págs. 1-6). George Town, Malaysia: IEEE Conference Publications.
- Hauge, Ø., Ayala, C., & Conradi, R. (2010). Adoption of open source software in software-intensive organizations – A systematic literature review. *Elseiver*, 52(11), 1133-1154.
- Kamau, J., & Namuye, S. (2012). A Review of Users Adoption of Open Source Software in Africa. *Computer and Information Science*, 45-49.
- Khelifi, A., Talib, M., Farouk, M., & Hamam, H. (2009). Developing an Initial Open-Source Platform for the Higher Education Sector—A Case Study: Alhosn University. *IEEE Transactions on Learning Technologies*, 239-248.
- Kitchenham, B. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Keele University.
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering* . Durham, UK: Technical report, EBSE Technical Report EBSE-2007-01.
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology* 52 (2010) 792–805 (pág. 14). Newton: Elsevier.

- Koch, S. (2004). Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion. *Extreme Programming and Agile Processes in Software Engineering* (págs. 85-93). Berlin, Heidelberg: Springer.
- Kovacs, G. L., Drozdik, S., Zuliani, P., & Succi, G. (2004). Open source software and open data standards in public administration. *Second IEEE International Conference on Computational Cybernetics, 2004. ICC 2004.* (págs. 421-428). Vienna, Austria: IEEE Conference Publications.
- Krishnamurthy, R., Jacob, V., Radhakrishnan, S., & Kutsal, D. (2016). Peripheral Developer Participation in Open Source Projects: An Empirical Analysis. *ACM Transactions on Management Information Systems (TMIS)* , 14:1 - 14:31.
- Lawton, G. (2002). Open source security: opportunity or oxymoron? *Computer Aided System Engineering (APCASE)*, 18-21.
- Lee, T. K., Wei, K. T., & Ghani, A. A. (2016). Systematic Literature Review on Effort Estimation for Open Sources (OSS) Web Application Development. *IEEE Xplore*.
- Lennerholt, C., Lings, B., & Lundell, B. (2008). Architectural issues in Opening up the advantages of Open Source. *2008 32nd Annual IEEE International Computer Software and Applications Conference* (págs. 1226-1227). Turku, Finland: IEEE Conference Publications.
- Li, Y., Chuan-Hoo, T., Hock-Hai, T., & Alex, S. (2005). A Human Capital Perspective of Organizational Intention to Adopt Open Source Software. *International Conference on Information Systems* (págs. 136-149). Singapore: ICIS 2005 Proceedings.
- Lyn Paul, C. (2009). A Survey of Usability Practices in Free/Libre/Open Source Software. *Open Source Ecosystems: Diverse Communities Interacting* (págs. 264-273). Berlin, Heidelberg: Springer.
- MacDonell, S., Shepperd, M., Kitchenham, B., & Mendes, E. (2010). How Reliable Are Systematic Reviews. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* (pág. 12). Piscataway: IEEE.
- Magazines, I. J. (2006). Liability and Control Risks with Open Source Software. *2006 International Conference on Information Technology: Research and Education* (págs. 242-245). Tel-Aviv, Israel: IEEE Conference Publications.
- Maki-Asiala, P., & Matinlassi, M. (2006). Quality Assurance of Open Source Components: Integrator Point of View. *30th Annual International Computer Software and Applications Conference (COMPSAC'06)* (págs. 189-194). Chicago, IL, USA: IEEE Conference Publications.
- Maldonado, E. (2010). The Process of Introducing FLOSS in the Public Administration: The Case of Venezuela. *Journal of the Association for Information Systems: Vol. 11 : Iss. 11*, 756-783.
- Miscione, G., & Johnston, K. (2010). Free and Open Source Software in developing contexts: From open in principle to open in the consequences. *Journal of Information, Communication and Ethics in Society*, 42-56.

- Morgan, B., & Jensen, C. (2014). Lessons Learned from Teaching Open Source Software Development. *Open Source Software: Mobile Open Source Technologies* (págs. 133-142). Berlin, Heidelberg: Springer.
- Morgan, L., & Finnegan, P. (2007). How Perceptions of Open Source Software Influence Adoption: An Exploratory Study. *ECIS 2007 PROCEEDINGS* (págs. 973-984). Geneva, Switzerland: AIS Electronic Library (AISeL).
- Munir, H., Wnuk, K., & Runeson, P. (2016). Open innovation in software engineering: a systematic mapping study. *Springer US*, 684-723.
- Nagy, D., Yassin, A. M., & Bhattacharjee, A. (2010). Organizational adoption of open source software: barriers and remedies. *Communications of the ACM Volume 53 Issue 3*, 148-151.
- Ngamkajornwiwat, K., Zhang, D., Koru, A. G., Zhou, L., & Nolker, R. (2008). An Exploratory Study on the Evolution of OSS Developer Communities. *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* (págs. 305-305). Waikoloa: IEEE Conference Publications.
- Noll, J., Beecham, S., & Seichter, D. (2011). A Qualitative Study of Open Source Software Development: The Open EMR Project. *International Symposium on Empirical Software Engineering and Measurement* (págs. 30-39). Banff: IEEE Conference Publications.
- Nouman, A., Anagnostou, A., & Taylor, S. J. (2013). Developing a Distributed Agent-Based and DES Simulation Using poRTIco and Repast. *ACM*, 97-104.
- Oreku, G. (2010). Open source software application and their impacts on SMEs: An action for building an OSS community environment in Tanzania. *2010 IST-Africa* (págs. 1-8). Durban: IEEE Conference Publications.
- Petrijevcanin Vuksanovic, I., & Sudarevic, B. (2012). Toward a framework for implementing open source software in public administrations: Legal, economic and societal aspects. *Information Technology Interfaces (ITI), Proceedings of the ITI 2012 34th International Conference on* (págs. 293-298). Cavtat, Croatia: IEEE Conference Publications.
- Rodríguez, G. (2008). El software libre y sus implicaciones jurídicas. *SCIELO*, 164-199.
- Rossi, B., Russo, B., & Succi, G. (2012). Adoption of free/libre open source software in public organizations: factors of impact. *Information Technology & People*, 156-187.
- Ruffin, M., & Ebert, C. (2004). Using open source software in product development: a primer. *IEEE Journals & Magazines*, 82-86.
- Russo, B., Zuliani, P., & Succi, G. (2003). Toward an Empirical Assessment of the Benefits of Open Source Software. *Taking Stock of the Bazaar: Proceedings of the 3rd Workshop on Open Source Software Engineering*, 117-120.
- Sack, W., Détienne, F., Ducheneaut, N., Burkhardt, J.-M., Mahendran, D., & Barcellini, F. (2006). A Methodological Framework for Socio-Cognitive Analyses of Collaborative Design of Open Source Software. *Kluwer Academic Publishers*, 229-250.
- Sahraoui, S. (2009). ICT governance in higher education: Case study of the rise and fall of open source in a Gulf university. *2009 International Conference on Information and*

- Communication Technologies and Development (ICTD)* (págs. 348 - 356). Doha, Qatar: IEEE Conference Publications.
- Samoladas, I., Stamelos, I., Angelis, L., & Oikonomou, A. (2004). Open source software development should strive for even greater code maintainability. *Communications of the ACM - Voting systems*, 83-87.
- Sen, R. (2007). A Strategic Analysis of Competition Between Open Source and Proprietary Software. *Journal of Management Information Systems*, 233-257.
- Silic, M., & Back, A. (2017). Open Source Software Adoption: Lessons from Linux in Munich. *IT Professional*, 42-47.
- Stallman, R. (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños.
- Steinmacher, I., Chaves, A. P., Conte, T., & Gorosa, M. A. (2014). Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects . *IEEE Xplore*.
- Steinmacher, I., Graciotto Silva, M. A., Gerosa, M. A., & Redmiles, D. F. (2015). A systematic literature review on the barriers faced by newcomers to open source . *Elsevier*, 59, 67-85.
- Sulaman, S. M., Orucevic-Alagic, A., Borg, M., Wnuk, K., Höst, M., & Vara, J. L. (2014). Development of Safety-Critical Software Systems Using Open Source Software -- A Systematic Map. *IEEE Xplore*.
- Syed, M. M., Kilamo, T., & Imed Hammouda, a. T. (2012). Open Source Prediction Methods: A Systematic Literature Review. *Springer*, 378, 280-285.
- Tawileh, A., & Rana, O. (2006). Free and Open Source Software Quality Assurance. *2006 2nd International Conference on Information & Communication Technologies* (págs. 2866-2871). Damascus, Syria: IEEE Conference Publications.
- Torres, J., & Petrizzo, M. (2015). *Política pública sobre el software libre*. Quito: Flok-Socyeti.
- Tsay, J. T., Dabbish, L., & Herbsleb, J. (2012). Social media and success in open source projects. *CSCW '12 Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion* (págs. 223-226). Seattle, Washington, USA: ACM.
- Universidad Autónoma de Madrid. (19 de Junio de 2017). *UAM_Biblioteca*. Obtenido de <http://biblioguias.uam.es/tutoriales/ieeexplore>
- Ven, K., Verelstt, J., & Mannaer, H. (2008). Should You Adopt Open Source Software? *IEEE Software*, 54-59.
- Waringa, T., & Maddocks, P. (2005). Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future. *ELSEVIER*, 411-428.
- WIPO. (3-7 de Marzo de 2008). *WIPO*. Obtenido de http://www.wipo.int/edocs/pubdocs/es/patents/434/wipo_pub_l434_11.pdf
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *EASE '14 Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (págs. 321-330). London: ACM.

- Wohlin, C., Runeson, P., Host, M., Ohlsson, M., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. New York: Springer.
- Wu, H., Shi, L., Chen, C., Wang, Q., & Boehm, B. (2017). Maintenance Effort Estimation for Open Source Software: A Systematic Literature Review. *IEEE Xplore*.
- Xie, R. (2008). The Economic Interests and Legal Issues of OSS. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing* (págs. 1-4). Dalian, China: IEEE Conference Publications.
- Zhang, H., & Ali Babar, M. (2013). Systematic reviews in software engineering: An empirical investigation. *Information and Software Technology 55 (2013) 1341–1354* (pág. 14). Elsevier.
- Zheng, L., Pan, T., Ren, G., & Fang, C. (2008). Development and Implementation of ERP/CRM System Based on Open Source Software to Small and Medium-Sized Enterprise in China. *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)* (págs. 725-729). Hunan, China: IEEE Conference Publications.

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

CERTIFICA:

Que, el Consejo de Facultad en sesión del 29 de mayo de 2017, conoció la petición de los estudiantes **OSCAR DAVID ARMIJOS MERA** con código **68271** y **LUIS FERNANDO DELGADO GUERRERO** con código **69019**, que presentan el diseño de su trabajo de titulación denominado: **“REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA IDENTIFICACIÓN DE FACTORES DE ÉXITO, FRACASO Y RIESGOS EN LA ADOPCIÓN DE SOFTWARE LIBRE”**, presentado previa a la obtención del título de Ingenieros de Sistemas y Telemática.- El Consejo de Facultad acogió el informe de la Junta Académica de Ingeniería de Sistemas y Telemática y resolvió aprobar el diseño. Designa como **Director al ingeniero Juan Pablo Carvalho Vega, Ph.D.** y como miembros del Tribunal Examinador a los ingenieros Francisco Salgado Arteaga, Ph.D. y Esteban Crespo Martínez.- En esta misma sesión el Consejo de Facultad fija como plazo para la entrega del trabajo de titulación, seis meses contados desde la fecha de su aprobación, esto es hasta el **29 de noviembre de 2017**, debiendo el Director presentar a la Junta Académica, dos informes bimensuales del desarrollo del trabajo de titulación.

Cuenca, mayo 30 de 2017

Dra. Jenny Ríos Coello
Secretaria de la Facultad de
Ciencias de la Administración



Decano de la Facultad de Ciencias de la Administración, Cuenca, 22 de noviembre de 2017.- Con autorización amplia y suficiente concedida por el Consejo de Facultad en sesión del 25 de febrero de 2016, conoció la petición de los estudiantes **OSCAR DAVID ARMIJOS MERA** con código 68271 y **LUIS FERNANDO DELGADO GUERRERO** con código 69019, quienes solicitan prórroga para la presentación del trabajo de titulación denominado: "**REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA IDENTIFICACIÓN DE FACTORES DE ÉXITO, FRACASO Y RIESGOS EN LA ADOPCIÓN DE SOFTWARE LIBRE**", previo a la obtención del título de Ingeniero de Sistemas y Telemática, cuyo plazo de presentación vence el 29 de noviembre de 2017, en apego al Reglamento de Régimen Académico y la normativa Institucional, *resuelve aprobar la solicitud y conceder una prórroga de seis meses, esto es hasta el 29 de mayo de 2018.*



Ing. Oswaldo Merchán Manzano
**Decano de la Facultad de
Ciencias de la Administración**

CONVOCATORIA

Por disposición de la Junta Académica de **Ingeniería de Sistemas y Telemática**, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: **"REVISIÓN SISTEMÁTICA DE LITERATURA PARA EL PROYECTO DE ADOPCIÓN DE OSS "ADOPTADOS"**, presentado por los estudiantes **Oscar Armijos Mera y Luis Delgado Guerrero**, previa a la obtención del grado de **Ingenieros en Sistemas y Telemática**, para el día **MARTES 16 DE MAYO DE 2017 A LAS 12h40**. La sustentación se realizará en el **laboratorio del IERSE**.

Cuenca, 15 de mayo de 2017



Dra. Jenny Ríos Coello
Secretaria de la Facultad

Ing. Juan Pablo Carvallo Vega

Ing. Esteban Crespo Martínez

Dr. Francisco Salgado Arteaga



mjmr/



Oficio Nro. 071-2017-DIST-UDA

Cuenca, 15 de mayo de 2017

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 15 de mayo del 2017, recibió el proyecto de tesis titulado "Revisión sistemática de literatura para el proyecto de adopción de OSS "adoptados"", presentado por Oscar Armijos Mera y Luis Delgado Guerrero estudiantes de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por Juan Pablo Carvallo Ph.D, previo a la obtención del título de Ingenieros de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior a Juan Pablo Carvallo Ph.D. y como miembros del Tribunal a Francisco Salgado Ph.D. e Ing. Esteban Crespo.

Atentamente,

Ing. Marcos Orellana Cordero
Coordinador Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: **Oscar Armijos Mera y Luis Delgado Guerrero**
- 1.2 Director sugerido: Ing. Juan Pablo Carvallo Vega
- 1.3 Codirector (opcional):
- 1.4 Tribunal: Ing. Esteban Crespo Martínez/ Dr. Francisco Salgado Arteaga
- 1.5 Título propuesto: **"REVISIÓN SISTEMÁTICA DE LITERATURA PARA EL PROYECTO DE ADOPCIÓN DE OSS "ADOPTADOS"**
- 1.6 Resolución:

1.6.1 Aceptado sin modificaciones _____

1.6.2 Aceptado con las siguientes modificaciones:

- 1. CAMBIO DE TITULO A: "REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA IDENTIFICACION DE FACTORES DE RITO Y PROCESO Y RIESGOS EN LA ADOCIÓN DE OSS"
- 2. AGREGAR LA PREGUNTA DE INVESTIGACION: "¿CUALES SON LOS BENEFICIOS DE USO DE OSS?"

1.6.3 Responsable de dar seguimiento a las modificaciones: Ing. Juan Pablo Carvallo Vega

1.6.4 No aceptado
• Justificación:

.....
Ing. Juan Pablo Carvallo Vega

Tribunal

.....
Ing. Esteban Crespo Martínez

.....
Dr. Francisco Salgado Arteaga

.....
Sr. Oscar Armijos Mera

.....
Sr. Luis Delgado Guerrero

.....
Dra. Jenny Ríos Coello
Secretario de Facultad

Fecha de sustentación: día MARTES 16 DE MAYO DE 2017 A LAS 12h40



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Oscar Armijos Mera y Luis Delgado Guerrero

1.2 Director sugerido: Ing. Juan Pablo Carvallo Vega

1.3 Codirector (opcional):

1.4 Título propuesto: "REVISIÓN SISTEMÁTICA DE LITERATURA PARA EL PROYECTO DE ADOPCIÓN DE OSS "ADOPTADOS"

1.5 Revisores (tribunal): Ing. Esteban Crespo Martínez/ Dr. Francisco Salgado Arteaga

1.6 Recomendaciones generales de la revisión:

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	/			
Título Propuesto				
2. ¿Es informativo?				
3. ¿Es conciso?	/			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	/			
5. ¿Describe la teoría en la que se enmarca el trabajo	/			
6. ¿Describe los trabajos relacionados más relevantes?	/			
7. ¿Utiliza citas bibliográficas?	/			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	/			
9. ¿Tiene relevancia profesional y social?	/			
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?	/			N/A
11. ¿Es factible de verificación?				N/A
Objetivo general				
12. ¿Concuerda con el problema formulado?	/			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	/			
Objetivos específicos				



14. ¿Concuerdan con el objetivo general?	✓			
15. ¿Son comprobables cualitativa o cuantitativamente?	✓			
Metodología				
16. ¿Se encuentran disponibles los datos y materiales mencionados?	✓			
17. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
18. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
19. ¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
Resultados esperados				
20. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
21. ¿Concuerdan con los objetivos específicos?	✓			
22. ¿Se detalla la forma de presentación de los resultados?	✓			
23. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓			
Supuestos y riesgos				
24. ¿Se mencionan los supuestos y riesgos más relevantes?	✓			
25. ¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
Presupuesto				
26. ¿El presupuesto es razonable?	✓			
27. ¿Se consideran los rubros más relevantes?	✓			
Cronograma				
28. ¿Los plazos para las actividades son realistas?	✓			
Referencias				
29. ¿Se siguen las recomendaciones de normas internacionales para citar?	✓			
Expresión escrita				
30. ¿La redacción es clara y fácilmente comprensible?	✓			
31. ¿El texto se encuentra libre de faltas ortográficas?	✓			

(*) Breve justificación, explicación o recomendación.



- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....
Ing. Juan Pablo Carvalho Vega

.....
Ing. Esteban Crespo Martínez

.....
Dr. Francisco Salgado Arteaga



Cuenca, 17 de mayo del 2017

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De mi consideración,

Yo **Juan Pablo Carvalho Vega** informo que he revisado los cambios realizados al protocolo del trabajo de titulación previo a la obtención del título de Ingeniero en Sistemas y Telemática, denominado "**REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA IDENTIFICACIÓN DE FACTORES DE ÉXITO, FRACASO Y RIESGOS DEL SOFTWARE LIBRE**", elaborado por los estudiantes **Oscar David Armijos Mera**, con código estudiantil 68271 y **Luis Fernando Delgado Guerrero**, con código estudiantil 69019. Trabajo que según mi criterio cumple con las modificaciones sugeridas por el Tribunal y puede continuar su desarrollo planificado.

Sin otro particular, suscribo

Atentamente

PhD. Juan Pablo Carvalho



Escuela
Sistemas y
Telemática

**Oficio Estudiante: Solicitud aprobación de
Protocolo de Trabajo de Titulación**

IST-RE-EST-02
Versión 01
04/04/2017
Página 1 de 1

Lugar de Almacenamiento
F: Archivo Secretaría de la Facultad

Retención
5 años

Disposición Final
Almacenar en archivo pasivo de la Facultad

Cuenca, 12 de mayo de 2017

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De nuestra consideración,

Estimado Señor Decano, nosotros **Oscar David Armijos Mera** con C.I. **0706811247**, código estudiantil **68271** y **Luis Fernando Delgado Guerrero** con C.I. **0105166797**, código estudiantil **69019**; estudiantes de la Carrera de Sistemas y Telemática, solicitamos muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema **"REVISIÓN SISTEMÁTICA DE LITERATURA PARA EL PROYECTO DE ADOPCIÓN DE OSS 'ADOPTADOS' "** previo a la obtención del título de Ingeniero en Sistemas y Telemática para lo cual adjuntamos la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipamos nuestro agradecimiento.

Atentamente:

Oscar Armijos Mera

Luis Delgado Guerrero

Estudiantes de la Carrera de Sistemas y Telemática



Lugar de Almacenamiento
F: Archivo Secretaría de la Facultad

Retención
5 años

Disposición Final
Almacenar en archivo pasivo de la Facultad

Cuenca, 15 de mayo de 2017

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De mi consideración,

Yo, **Juan Pablo Carvalho Vega** informo que he revisado el protocolo de trabajo de titulación previo a la obtención del título de Ingeniero en Sistemas y Telemática, denominado **"REVISIÓN SISTEMÁTICA DE LITERATURA PARA EL PROYECTO DE ADOPCIÓN DE OSS 'ADOPTADOS' "**, realizado por los estudiantes **Oscar David Armijos Mera**, con código estudiantil 68271 y **Luis Fernando Delgado Guerrero**, con código estudiantil 69019, protocolo que a mi criterio, cumple con los lineamientos y requerimientos establecidos por la carrera.

Por lo expuesto, me permito sugerir que sea considerado para la revisión y sustentación del mismo,

Sin otro particular, suscribo.

Atentamente

Ing. Juan Pablo Carvalho Vega, PhD



DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, el Señor **Luis Fernando Delgado Guerrero** registrado con código **69019** estudiante de la Escuela de Ingeniería de Sistemas y Telemática tiene aprobado más del 80% de su Pensum de estudios.

Que, el Señor **Luis Fernando Delgado Guerrero** le falta aprobar las siguientes asignaturas para finalizar sus estudios:

Metodología de la Investigación

Ingeniería de Software II

Desarrollo y Gestión de Proyectos Informáticos

Proyectos Telemáticos

Sistema de Información Gerencial

Producción II

Cuenca, 23 de Marzo de 2017



Derecho 115835

vcf.-



**DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD DE
CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY**

CERTIFICA:

Que, el Señor **Oscar David Armijos Mera** registrado con código **68271** estudiante de la Escuela de Ingeniería de Sistemas y Telemática tiene aprobado más del 80% de su Pensum de estudios.

Que, el Señor **Oscar David Armijos Mera** le falta aprobar las siguientes asignaturas para finalizar sus estudios:

Metodología de la Investigación

Ingeniería de Software II

Desarrollo y Gestión de Proyectos Informáticos

Proyectos Telemáticos

Sistema de Información Gerencial

Producción II

Cuenca, 23 de Marzo de 2017



Derecho 155099
vcf.-



1. DATOS GENERALES

1.1. Nombre de los estudiantes: Armijos Mera Oscar David / Delgado Guerrero Luis
Fernando

1.1.1. Códigos: 68271 / 69019

1.1.2. Contacto:

Teléfono convencional: 4036189

Celular:

Armijos Oscar: 0998251217

Delgado Luis: 0981305801

Correo electrónico:

Armijos Oscar: oskr14@live.com

Delgado Luis: lfdg_92@hotmail.com

1.2. Director sugerido: Carvallo Vega, Juan Pablo Ing. PhD

1.2.1. Contacto

Teléfono convencional:

Celular: 0990447559

Correo electrónico: jpcarvallo@gmail.com

1.3. Co-director sugerido: (opcional)

1.4. Asesor metodológico: (opcional).

1.5. Tribunal designado:

Presidente:

Vocal:

Sugerencias:

1.6. Aprobación:

1.7. Línea de Investigación de la carrera:

1.7.1. Código UNESCO:

1203 Informática de computadores

1203.17 Informática

1.7.2. Tipo de trabajo: Revisión Sistemática de Literatura.

1.8. Área de estudio: Innovación tecnológica.

1.9. Título propuesto: "REVISIÓN SISTEMÁTICA DE LITERATURA PARA LA IDENTIFICACIÓN DE FACTORES DE ÉXITO, FRACASO Y RIESGOS DEL SOFTWARE LIBRE".

1.10. Subtítulo:

1.11. Estado del proyecto: El proyecto es considerado nuevo, dado que consiste en la revisión sistemática de documentación existente, la cual permita a futuro generar una guía de implementación de software libre la cual no ha sido desarrollada. Este trabajo forma parte del proyecto "Adoptados" que se lleva a cabo en la Universidad del Azuay.

2. CONTENIDO

2.1. Motivación de la investigación.

El conocer los retos que afrontan las organizaciones al adoptar software libre y su influencia en la transición hacia una economía social del conocimiento común y abierto.

Comprender las ventajas y desventajas de la implementación de software libre en las organizaciones, así como también los pasos que deben ser aplicados para alcanzar el éxito y evitar el fracaso.

2.2. Problemática

Según la Free Software Foundation (FSF), el software libre es todo programa que otorga su libre utilización y que permita estudiarlo y compartir los conocimientos adquiridos con la comunidad en general.

Con la predisposición de lograr soberanía y autonomía tecnológica, además de minimizar los costos en recursos públicos al Gobierno del Ecuador en el Decreto Ejecutivo No. 1014, firmado el 10 de abril de 2008, en su artículo 1, constituye como política pública la implementación de Software Libre en las Entidades de Administración Pública Central en sus sistemas y equipamientos informáticos (Correa, 2008).

Al ser la implementación de software libre un aspecto relevante en la realidad ecuatoriana, y al no tener identificados ni analizados detalladamente los riesgos y las mejores prácticas asociadas a la adopción de Software Libre (OSS), razón por la cual se ve limitada claramente la óptima explotación de los beneficios de adoptar OSS en empresas públicas y privadas en el país.

Para atender esta necesidad se plantea el proyecto "Adoptados", que mantiene la Universidad del Azuay sobre la adopción de Software Libre, dentro del cual se pretende realizar una revisión sistemática de literatura (SLR) sobre las experiencias en la implementación de OSS en las diferentes organizaciones a nivel nacional e internacional; para agregarle un gran valor científico y colaborar con el esfuerzo que conlleva el estudio de la adopción de software libre en el país, permitiendo de esta manera levantar el estado del arte para que se pueda aplicar en el proyecto antes mencionado.

2.3. Preguntas de investigación

- ¿Cuáles han sido los factores de éxito y fracaso en la adopción de tecnologías de software libre en las organizaciones?
- ¿Cuáles son los beneficios de la utilización de software libre?
- ¿Cuáles son los principales riesgos que se han identificado en la adopción de software libre a nivel internacional?
- ¿En qué dominios de software ha resultado exitosa la adopción de software libre?
- ¿Qué impide a las organizaciones adoptar software libre?
- ¿La documentación generada por la comunidad desarrolladora de un software libre es lo suficientemente comprensible como para lograr una implementación eficiente?

05888979

2.4. Resumen

El gran impacto que causa el software libre en la sociedad es considerado una corriente exitosa, por ser un modelo innovador y por su libre utilización (Vergara Villadiego & Francisco, 2014).

El Open Source Software (OSS) ha cambiado la manera de distribuir y comercializar el software y debido a su importancia estratégica en los últimos años las distintas organizaciones públicas están optando por impulsar su implementación y fortalecimiento.

El presente trabajo de titulación de pregrado tiene como finalidad realizar una revisión sistemática de literatura de la adopción de OSS. Haciendo uso de la guía de Kitchenham & Charters (2007), la cual es una técnica de investigación empírica que requiere seguir todo un protocolo para la recopilación de literatura, se pretende identificar, evaluar e interpretar las investigaciones existentes relacionadas a la adopción del software libre, los factores de éxito, fracaso y los riesgos en las organizaciones, que conlleva la implementación de esta tecnología.

Al aplicar esta revisión exhaustiva se sintetiza la información existente, otorgándole un alto valor científico.

Luego de haber desarrollado la SLR, permitirá desarrollar el estado del arte sobre el uso del OSS para el proyecto "Adoptados" emprendido por la Universidad del Azuay.

2.5. Indagación exploratoria y marco teórico

La revisión sistemática es una técnica para evaluar e interpretar la literatura científica en base a preguntas previamente establecidas que están presentes en todo el proceso de la investigación, junto a un protocolo previamente establecido. Esto permite que otros investigadores tomen el trabajo como sustento para próximos proyectos de investigación (Matalonga & Martín, 2013).

Es considerado software libre todos aquellos programas que son compartidos a los usuarios con total libertad de uso, esto quiere decir que al usuario final se le conceden los permisos para experimentar con el programa para adecuarlo a sus necesidades, utilizarlo de acuerdo a su conveniencia compartirlo con otras personas las veces que amerite sin prohibición alguna, es decir con total libertad de distribución y modificación (Da Rosa & Heinz, 2007).

Según estudios realizados, existe preferencia y facilidad de uso en herramientas de software libre, como se da en los navegadores que muchas personas prefieren: Firefox, Chrome u. Opera que son basados en software libre a diferencia de Internet Explorer que es propietario. (Echeverría, 2014).

Por el gran impacto que el software libre ha causado en varios sectores de la sociedad, es considerado una corriente exitosa; ya sea por tratarse de un modelo innovador, o por su libre utilización, considerando ejemplos como: Android, Linux, Apache entre otros. (Vergara Villadiego & Francisco, 2014).

Hay una gran variedad de aplicaciones de software libre que se ofrecen en el mercado. Entre las más famosas que se pueden citar, según Rodríguez (2008) están:

- El sistema operativo Linux
- El servidor de web Apache
- El manejador de bases de datos objeto-relacional PostgreSQL
- El navegador Mozilla
- La suite de aplicaciones de escritorio OpenOffice
- El servidor de correo Sendmail.

Algo que impide adoptar e implementar software libre es que los gerentes o profesionales no cuentan con la suficiente información acerca de este tema y por consiguiente se tiende a crear ciertos conceptos erróneos con respecto al uso del modelo de negocio, a la calidad, filosofía. Esta desinformación provoca que al momento de elegir tecnología libre tienda a existir cierto rechazo (Tugnarelli, Pacífico, & Pérez, 2016). Otro factor que impide la adopción del software libre es la falta de soporte de fabricantes de software y hardware, dejando así a las organizaciones en manos de la comunidad, incrementando el riesgo de disponibilidad de información procesada o almacenada en este tipo de tecnologías OSS. (Echeverría, 2014).

Según Torres y Petrizzo(2015) la Free Software Foundation (FSF) determina las cuatro características para que un software sea libre:

1. Utilización libre del software con cualquier fin.
2. Acceder libremente al código del software para analizarlo y adaptarlo a las necesidades.
3. Libertad de redistribuir copias de software.
4. Mejorar el software y compartirlo al público en general.

También existe una gran cantidad de personas que se oponen a la utilización de software libre argumentando de que se pierde la propiedad intelectual, cuando el mismo se distribuye a terceros. Aun así, en cuanto a la utilización de software, es que la gran mayoría de organizaciones buscan alternativas que sean buenas y rápidas que no sean costosas, por lo cual usualmente usan software sin el debido licenciamiento. (Peralta, Saza, & Heredia, 2015).

Es deber de las universidades impartir la utilización de software libre en sus estudiantes, proveyéndoles bases fundamentales para su vida diaria y la productividad en el trabajo, con esto se colabora a salir de las barreras de depender tecnológicamente de software propietario y reducir los comportamientos inadecuados de adquirir los software sin los respectivos permisos legales. (Cárdenas, y otros, 2016).

Según Echeverría (2014) estas preferencias se dan debido a los beneficios que ofrece el software libre a diferencia del software propietario, entre los que se pueden rescatar:

- Ofrecen actualizaciones constantes y una gran cantidad de usuarios que están día a día resolviendo problemas con estos.
- Son ligeros, flexibles y al tener una gran comunidad que trabaja en actualizaciones muchas de las veces ofrecen características más innovadoras a diferencia de otro tipo de software.
- Ofrecen mucha más variedad de complementos y afines.
- Los podemos usar en gran variedad de plataformas.
- Cuentan con soporte adicional.

Lo que impulsa principalmente el desarrollo de software libre en los programadores es principalmente para tener nuevos conocimientos y destrezas, mientras que muy pocos lo hacen para generar ganancias económicas (Robert, 2006).

Muchas instituciones ya sean públicas o privadas, han optado por la utilización de software libre. Se asevera que España es uno de los países europeos que más adopta este tipo de software, además se dice que las organizaciones españolas utilizan algún tipo de aplicación de código abierto, sistemas operativos y servidores web. En América Latina los países han creado leyes que impulsa la utilización de software libre en sus territorios. (Montes de Oca, 2014)

2.6. Hipótesis: (opcional).

2.7. Objetivo general

- Elaborar una revisión sistemática de literatura para identificar el estado del arte sobre los factores de éxito, fracaso y riesgos en el uso del OSS.

2.8. Objetivos específicos

1. Elaborar un protocolo de investigación.
2. Seleccionar los artículos de las fuentes requeridas.
3. Realizar la revisión de la literatura.
4. Realizar la tabulación de análisis de datos de los artículos.
5. Analizar los resultados obtenidos.

2.9. Metodología

Para la consecución de los objetivos del proyecto, se ha establecido una metodología basada en los siguientes principios:

- Profundizar en las bases teóricas del proyecto con la utilización de bibliografía.
- Identificar el punto de partida del desarrollo del software libre
- Aplicación de la guía para elaboración de Revisión Sistemática de Literatura (SLR) orientada a la aplicación de OSS.

2.10. Alcance y resultados esperados:

Culminado el proyecto de investigación se pretende obtener:

- Responder las preguntas de investigación
- Estado del arte sobre el uso de OSS.
- Incrementar el conocimiento a nivel científico sobre la adopción de OSS.

2.11. Supuestos y riesgos

RIESGOS	Probabilidad	Solución
1. Incorrecta aplicación de la guía de directrices establecida.	Baja	Aplicar paso a paso la guía y solicitar ayuda a expertos en el tema.
2. Demora en la entrega del proyecto final.	Media	Desarrollar un cronograma realista

3. Poca interacción con los interesados claves.	Media	Establecer fechas de reunión fijas.
4. Falta de literatura referente al tema.	Baja	Buscar en diferentes fuentes e idiomas.
5. Cambio en el alcance del proyecto.	Baja	Establecer claramente el alcance del proyecto.
6. Problemas para solventar económicamente el proyecto.	Baja	Pedir ayuda económica a interesados claves o buscar quien ayude a solventar la realización del proyecto.

2.12. Presupuesto

No requieren presupuesto.

2.13. Financiamiento

No requiere financiamiento.

2.14. Esquema tentativo

Capítulo 1: Introducción

Capítulo 2: Revisión sistemática de literatura

Capítulo 3: Proceso de revisión

Capítulo 4: Realización de la revisión

Capítulo 5: Estudios de Mapeo Sistemático

Capítulo 6: Observaciones finales

Conclusiones y recomendaciones

Bibliografía

5886870

2.15. Cronograma

CRONOGRAMA SLR Mayo 2017 - Octubre 2017

ACTIVIDAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	FECHAS		
	MAYO					JUNIO					JULIO					AGOSTO					SEPTIEMBRE					OCTUBRE			
Elaborar un protocolo de investigación.																											15 al 28 de Mayo		
Seleccionar los artículos de las fuentes requeridas.																											29 de mayo al 2 de julio		
Realizar la revisión de la literatura.																											del 3 de julio al 27 de agosto		
Realizar la tabulación de análisis de datos de los artículos.																											del 28 de agosto al 1 de octubre		
Analizar los resultados obtenidos.																											del 2 al 22 de octubre		
TOTAL																													23 Semanas



2.16. Referencias bibliográficas

Cárdenas, M., Zamora Mayorga, J., Aguirre Pérez, R., Chiriboga Casanova, W. A., Ortega Acosta, J., Cañizares Galarza, F. P., & Reyes Pérez, J. J. (2016). SOFTWARE LIBRE. UNA HERRAMIENTA PARA LA FORMACIÓN ACADÉMICA EN EL ECUADOR. *Didas@caliá: Didáctica y Educación*, 223-224.

Correa, R. (2008). *Decreto presidencial 1014*. Quito: Gobierno de la República del Ecuador.

Da Rosa, F., & Heinz, F. (2007). *Guía práctica sobre software libre*. Montevideo: Mastergraf SRL.

Dartsch, G. (2012). *El movimiento del software libre como cuestionamiento*. La Plata : Sociedad Argentina de Informática - Universidad Nacional de La Plata.

Echeverría, M. A. (2014). ACCESO ABIERTO Y SOFTWARE LIBRE. *e-Ciencias de la Información*, 12.

Foundation, F. S. (7 de Febrero de 2013). *static.fsf*. Obtenido de <http://static.fsf.org/common/what-is-fs-es.pdf>

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Durham, UK: Technical report, EBSE Technical Report EBSE-2007-01.

Matalonga, S., & Martín, S. (2013). *Un programa de investigación en deuda técnica del software*. Montevideo: Universidad ORT Uruguay.

Montes de Oca, J. L. (2014). *Dimensión social del proceso migratorio hacia el software libre*. Norderstedt Germany: GRIN Verlag, Open Publishing GmbH.

Peralta, B., Saza, L. D., & Heredia, V. (2015). El software libre ¿Fin de la propiedad individual? *Inventum*, 9-13.

Robert, V. (2006). *La informática en Argentina. Desafíos a la especialización y a la competitividad*. Buenos Aires: Prometeo-UNGS.

Rodríguez, G. (2008). El software libre y sus implicaciones jurídicas. *SCIELO*, 66-67.

Torres, J., & Petrizzo, M. (2015). *Política pública sobre el software libre*. Quito: Flok-Socyeti.

Tugnarelli, M., Pacifico, C., & Pérez, M. (2016). Especificación de indicadores en el proceso de software con tecnologías libres. *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*, (págs. 641-645). Concepción.

Vergara Villadiego, J., & Francisco, T. (2014). El software libre en las empresas de Montería. *Ingeniería solidaria*, 106.

2.17. Anexos

2.18. Firma de responsabilidad (estudiantes)



Oscar Armijos-Mera

2.19. Firma de responsabilidad (estudiantes)



Luis Delgado Guerrero

2.20. Firma de responsabilidad (director sugerido)



Ing. PhD. Juan Pablo Carvalho

2.21. Firma de responsabilidad (metodólogo)



Prof. Dr. Francisco Salgado

2.22. Fecha de entrega: 17/05/2017

