

# **FACTORES PARA LA ADOPCIÓN DE SOLUCIONES BASADAS EN SOFTWARE LIBRE**

**VÍCTOR HUGO REA SÁNCHEZ**



**Universidad de Sevilla**

**Tesis dirigida por:**

**Dr. Pablo Neira Ayuso y Dr. David Benavides Cuevas**

First published in June 2022 by  
The Department of Computer Languages and Systems  
ETSI Informática  
Avda. de la Reina Mercedes s/n  
Sevilla, 41012. SPAIN

Copyright © 2022 Víctor Hugo Rea Sánchez  
vreas@unemi.edu.ec

**Categorías (ACM 2012):**

**Categories and subject descriptors:**

[500] Social and professional topics-Software selection and adaptation  
[500] Social and professional topics-Quality assurance  
[500] Social and professional topics-History of software  
[500] Social and professional topics-Licensing

**General Terms:** Guide, Software Adoption, Assessment, Licensing.

**Additional Key Words and Phrases:** Free software adoption, open source adoption, libre software adoption, software assessment.

**Financiación:** This work has been partially supported by University of Milagro (UNEMI) with its scholarship program. It has also been partially funded by the Project (RTI2018-101204-B-C22, OPHELIA), funded by: FEDER/Ministry of Science and Innovation - State Research Agency; and the Junta de Andalucía COPERNICA (P20\_01224) and METAMORFOSIS (FEDER\_US-1381375) projects.

Don Pablo Neira Ayuso, profesor del área de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla y Don David Benavides Cuevas, catedrático del Área de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.

### **HACEN CONSTAR**

que Don Víctor Hugo Rea Sánchez, Ingeniero en Sistemas Computacionales por la Universidad de Especialidades Espíritu Santo, Ecuador, ha realizado bajo nuestra supervisión el trabajo de investigación titulado

*Factores para la adopción de soluciones basadas  
en software libre*

Una vez revisado, autorizamos el comienzo de los trámites para su presentación como tesis doctoral al tribunal que ha de juzgarlo.

Fdo. Dr. Pablo Neira Ayuso y Dr. David Benavides Cuevas  
Área de Lenguajes y Sistemas Informáticos  
Universidad de Sevilla,  
Sevilla, junio de 2022

Yo, Víctor Hugo Rea Sánchez, con número de DNI 0917440521,

**DECLARO**

Ser el autor del trabajo que se presenta en la memoria de esta tesis doctoral que tiene por título:

*Factores para la adopción de soluciones basadas  
en software libre*

Lo cual firmo en Sevilla, junio de 2022.

Fdo. Víctor Hugo Rea Sánchez

A mi familia  
En especial a mi sobrino Diego

# Índice general

<b>Índice de figuras</b>	<b>8</b>
<b>Índice de tablas</b>	<b>10</b>
<b>Agradecimientos</b>	<b>11</b>
<b>Resumen</b>	<b>12</b>
<b>Abstract</b>	<b>14</b>
<b>I Preliminares</b>	<b>16</b>
<b>1 Introducción</b>	<b>17</b>
1.1. Contexto de la investigación . . . . .	18
1.1.1. Software libre, <i>open source</i> y FLOSS . . . . .	18
1.1.2. Situación actual de la adopción de FLOSS . . . . .	18
1.1.3. Dimensiones, factores y subfactores que determinan la adopción de FLOSS . . . . .	20
1.2. Contribuciones . . . . .	21
1.2.1. Resumen de las contribuciones . . . . .	21
1.3. Estructura de esta disertación . . . . .	22
<b>II Antecedentes</b>	<b>23</b>
<b>2 FLOSS en la actualidad</b>	<b>24</b>
2.1. ¿Qué es FLOSS? . . . . .	25
2.1.1. Software libre . . . . .	25
2.1.2. Software <i>open source</i> . . . . .	27
2.1.3. Diferencias entre software libre y <i>open source</i> . . . . .	28
2.2. Licencias FLOSS . . . . .	29
2.2.1. ¿Qué es <i>copyleft</i> ? . . . . .	29
2.2.2. Licencias <i>copyleft</i> . . . . .	30

<i>ÍNDICE GENERAL</i>	6
2.2.3. Licencias <i>copyleft débil</i> . . . . .	31
2.2.4. Licencias no <i>copyleft</i> . . . . .	31
2.3. Resumen . . . . .	32
<b>3 Motivación</b>	<b>33</b>
3.1. Introducción . . . . .	34
3.2. FLOSS en el ámbito científico . . . . .	35
3.2.1. Análisis de la literatura . . . . .	37
3.3. Identificación de factores de adopción FLOSS . . . . .	42
3.4. Metodologías existentes para la adopción FLOSS . . . . .	44
3.5. Resumen . . . . .	46
<b>III Contribuciones</b>	<b>47</b>
<b>4 Factores de adopción</b>	<b>48</b>
4.1. Introducción . . . . .	49
4.2. Definición del proceso sistemático . . . . .	50
4.2.1. Fase 1: planificación de la revisión sistemática . . . . .	50
4.2.2. Fase 2: identificación de estudios primarios . . . . .	51
4.2.3. Fase 3: extracción, síntesis y clasificación de datos . . . . .	55
4.2.4. Amenazas a la validez . . . . .	57
4.3. Factores de adopción FLOSS . . . . .	58
4.3.1. Factores tecnológicos . . . . .	58
4.3.2. Factores organizacionales . . . . .	60
4.3.3. Factores económicos . . . . .	64
4.4. Alcance de la investigación en la adopción FLOSS . . . . .	66
4.4.1. ¿Qué tipo de trabajos de investigación abarcan los factores de adopción FLOSS? . . . . .	67
4.4.2. ¿Cuántas publicaciones relacionadas con FLOSS se publicaron en los años que cubren esta revisión? . . . . .	68
4.5. Resumen . . . . .	69
<b>5 Hacia la elaboración de una guía para la adopción FLOSS</b>	<b>71</b>
5.1. Introducción . . . . .	72
5.2. Proceso de elaboración de la guía para adopción FLOSS . . . . .	72
5.2.1. Dimensiones, factores y subfactores . . . . .	73
5.2.2. Encuesta para valorar la importancia de los subfactores . . . . .	80
5.2.3. Indicadores de importancia de factores y subfactores . . . . .	83
5.2.4. Amenazas a la validez . . . . .	91
5.3. Resumen . . . . .	93
<b>6 Elaboración y validación de GUIOS para la adopción FLOSS</b>	<b>94</b>
6.1. Guía para adopción FLOSS . . . . .	95

<i>ÍNDICE GENERAL</i>	7
6.1.1. GUIOS PRO: un prototipo para la automatización de GUIOS	97
6.2. Una validación preliminar de GUIOS . . . . .	98
6.2.1. Diseño del estudio: contexto y unidad de análisis . . . . .	99
6.2.2. Recogida de datos: planificación, estrategias y análisis . . .	101
6.2.3. Resultados preliminares . . . . .	101
6.3. Resumen . . . . .	107
<b>IV Observaciones finales</b>	<b>109</b>
<b>7 Conclusiones y trabajos futuros</b>	<b>110</b>
7.1. Conclusiones . . . . .	111
7.2. Debate, limitaciones y extensiones . . . . .	111
<b>V Apéndice</b>	<b>116</b>
<b>A Encuesta FLOSS</b>	<b>117</b>
<b>Bibliografía</b>	<b>124</b>



# Índice de figuras

1.1. Uso del software libre en las entidades de la función ejecutiva del Ecuador.	19
2.1. Categorías de licencia de software.	30
3.1. Evolución de artículos sobre FLOSS en revistas indexadas por Scopus.	37
3.2. Distribución de la cantidad de artículos de revista sobre FLOSS indexados en Scopus desde 2004.	37
4.1. Definición proceso sistemático [91, 102].	50
4.2. Primera fase del estudio [91, 102]	51
4.3. Segunda fase del estudio [91, 102]	52
4.4. Proceso de búsqueda de documentos [91, 102]	54
4.5. Tercera fase del estudio [91, 102]	55
4.6. Relación entre el grupo de factores y los tipos de investigación.	67
4.7. Evolución temporal de las diferentes tendencias por dimensión y año de publicación.	68
4.8. Orden de factores	69
5.1. Proceso para la elaboración de GUIOS.	73
5.2. Distribución de los expertos por sector.	80
5.3. Distribución de los expertos por nivel de experiencia.	81
5.4. Distribución de los expertos por campo.	81
5.5. Distribución de las evaluaciones de los expertos por factor.	82
5.6. Distribución de las evaluaciones de los expertos por dimensión.	83
5.7. Evaluación de expertos a los subfactores de la dimensión tecnológica.	84
5.8. Evaluación de expertos a los subfactores de la dimensión organizacional.	85
5.9. Evaluación de expertos a los subfactores de la dimensión económica.	85
5.10. Función para determinar la importancia sugerida/relativa de un factor en función de la importancia de la literatura/sugerida y la importancia del experto/decisor.	89
6.1. Pasos a seguir de GUIOS.	96
6.2. Interfaz de GUIOS PRO correspondiente a los pasos 1 y 2 de GUIOS.	98
6.3. Interacción de la especialista en la interfaz de los pasos 1 y 2 de GUIOS	102
6.4. Interacción de la especialista en la interfaz de los pasos 3 y 4 de GUIOS	103

6.5. Interacción de la especialista en la interfaz de los pasos 5 y 6 de GUIOS	104
6.6. Casos alternativos de la decisión sugerida por GUIOS PRO. . . . .	106
A.1. Presentación de la encuesta a expertos. . . . .	118
A.2. Inicio de la evaluación de las características de los factores, parte I. . . .	119
A.3. Lista de características de los factores, parte II. . . . .	120
A.4. Lista de las características de los factores, parte III. . . . .	121
A.5. Lista de las características de los factores, parte IV. . . . .	122

# Índice de tablas

3.1. Distribución de artículos sobre FLOSS de acuerdo al principal tema que se aborda y el nivel de relevancia. . . . .	36
3.2. Consulta aplicada en la base de datos Scopus para la obtención de artículos científicos FLOSS. . . . .	38
3.3. Comparación de artículos relacionados . . . . .	43
3.4. Características de artículos relacionados con GUIOS . . . . .	46
4.1. Resultados de la búsqueda realizada en las bases de datos. . . . .	53
4.2. Clasificación de artículos basados en factores tecnológicos. . . . .	61
4.3. Clasificación de los documentos basada en factores organizacionales. . .	65
4.4. Clasificación de los artículos basada en factores económicos. . . . .	66
4.5. Resumen de factores identificados por artículo. . . . .	70
5.1. Factores y subfactores del marco de trabajo propuesto para la adopción FLOSS . . . . .	75
5.2. Indicadores definidos para caracterizar la importancia de los factores y subfactores en el proceso de adopción FLOSS. . . . .	86
5.3. Cálculo de la importancia sugerida para caracterizar a los factores y subfactores en el proceso de adopción FLOSS. . . . .	88
5.4. Clasificación de factores internos y externos bajo un enfoque FODA. . .	90

# Agradecimientos

Existen muchos motivos por el cuál agradecer una vez que culminas un proyecto, a todas esas personas que han sido una pieza fundamental en este arduo camino y que han contribuido con su experiencia y soporte permanentemente.

Quiero agradecer de una manera especial a mis abuelos, papá y sobrino, por estar siempre presente en mis pensamientos y por alentarme desde el cielo a cumplir este objetivo.

Gracias a mi madre, por alentarme día a día, su ejemplo me ayudó a no desmayar en el camino; a mi esposa e hijos por apoyarme en todo momento y la comprensión por el tiempo en el que no estuve presente, ¡gracias de corazón!; a mi Director Dr. Pablo Neira Ayuso, su guía fue fundamental para lograr este trabajo de investigación; también a mi Director Dr. David Benavides Cuevas, su experiencia fue clave para guiarme en este proceso; Al Dr. José Galindo Duarte, su contribución a este trabajo fue de gran valor para alcanzar el objetivo final. También un agradecimiento a mi colega Felipe Escalera por su oportuna revisión de la redacción a esta tesis, valioso aporte.

Gracias a la Universidad Estatal de Milagro y a sus autoridades, por darme el tiempo necesario para realizar mis estancias en la ciudad de Sevilla, y por toda la predisposición y confianza brindada en este proceso. A la Universidad de Sevilla por permitir que yo sea parte de su comunidad estudiantil, ¡un honor para mi!. Al Dr. Miguel Toro Bonilla, por su tiempo y sabios consejos que me permitieron avanzar con mi trabajo doctoral. A mi colega y amiga Dra. Jesennia Cárdenas Cobo, por abrir el camino para ser parte de este programa doctoral.

Gracias a mis amigos y colegas de profesión que aportaron con su experiencia e hicieron que este trabajo tenga un buen camino.

# Resumen

Hoy en día, el software libre o de código abierto (FLOSS) se está convirtiendo en una opción estratégica para cualquier organización del sector público y privado. Algunos países han optado por implantar políticas para el uso de FLOSS debido a la flexibilidad que este tipo de tecnología ofrece. El alcance de FLOSS tiene muchos beneficios y es por eso que la visión de muchos gobiernos y organizaciones se han enfocado en su adopción.

La falta de directrices bien definidas para los responsables de un área de tecnología de la información (TI) puede poner en peligro el proceso de adopción FLOSS. Los procedimientos de adopción FLOSS se desarrollan de manera personalizada en cada organización, lo que lleva a posibles situaciones de reinención de la rueda.

Es necesario definir bien los pasos a seguir en un proceso de adopción FLOSS, para evitar una selección inadecuada del software en cuestión. En primer lugar, es crucial identificar los factores que influyen y determinan la adopción, para que sean una base determinante para el diseño de una metodología que los oriente en el proceso. Para alcanzar este punto, analizamos la literatura existente mediante metodologías de revisión sistemática, para hacer visibles los factores técnicos, organizacionales y económicos que deben evaluarse en el proceso de adopción.

Los resultados obtenidos con la revisión de la literatura, demostraron la incidencia que tienen estos factores al momento de adoptar FLOSS. En particular, se identificaron 22 factores que fueron clasificados para una mejor interpretación de los resultados. Un grupo de factores llamados “organizacionales” se consideró el más relevante, ya que el 93 % de los trabajos seleccionados se refieren a ellos. Otro grupo de factores llamados “tecnológicos” consiguieron el 91 % y los factores “económicos” alcanzaron el 61 % de ser mencionados en los trabajos de investigación seleccionados.

Con los factores identificados, un nivel de granularidad más detallado por factor fue necesario descubrir para obtener características relevantes en cada artículo de investigación. A estas características las denominamos subfactores, un total de 61 subfactores fueron identificados. Asimismo, se definieron indicadores de medición para evaluar estos factores y subfactores a través de una herramienta prototipo. En este contexto, se propone una guía que permita orientar a los responsables de un área de tecnología de la información a evaluar un FLOSS de una manera ágil y sencilla.

Esta herramienta prototipo ofrece tres recomendaciones en general que dependen de la valoración realizada en el proceso de adopción FLOSS. El desarrollo de un pro-

prototipo para agilizar el proceso de evaluación FLOSS, es un complemento importante de esta propuesta, debido a la flexibilidad que esta herramienta ofrece a los responsables de la evaluación del software. Con el objetivo de validar preliminarmente la elaboración de GUIOS, hemos diseñado un piloto de caso de estudio para aplicarlo en la Universidad Estatal de Milagro. Los resultados obtenidos servirán para proponer mejoras a futuro en la guía.

Este trabajo de investigación abre una línea de investigación en el futuro, al proponer una guía para el proceso de adopción FLOSS. También es importante considerar la exploración de otros métodos y técnicas de evaluación para ofrecer un proceso de adopción aún más flexible y preciso. En la herramienta prototipo se deberán de incorporar nuevas funcionalidades que permitan mayor flexibilidad en el uso del software y un nivel de granularidad óptimo.

# Abstract

Nowadays, free libre open source software (FLOSS) is becoming a strategic option for any organization in the public and private sector. Some countries have chosen to implement policies for the use of FLOSS, due to the flexibility that this type of technology offers. The scope of FLOSS has many benefits and that is why many governments and organizations have focused on its adoption.

The lack of well-defined guidelines for those responsible for an information technology (IT) area can jeopardize the FLOSS adoption process. FLOSS adoption procedures are developed ad-hoc in every organization, hence, leading to potential wheel reinvention situations.

It is necessary to define well the steps to follow in a FLOSS adoption process, in order to avoid an inadequate selection of the software in question. First of all, it is crucial to identify the factors that influence and determine the adoption, to be a determining basis for the design of a methodology to guide them in the process. To reach this point, we analyzed the existing literature using systematic review methodologies to make visible the technical, organizational and economic factors to be evaluated in the adoption process.

The results obtained from the literature review demonstrated the incidence of these factors at the time of adopting FLOSS. In particular, 22 factors were identified and classified for a better interpretation of the results. A group of factors called “organizational” was considered the most relevant, since 93 % of the selected papers refer to them. Another group of factors called “technological” achieved 91 % and “economic” factors reached 61 % of being mentioned in the selected research papers.

With the factors identified, a more detailed level of granularity per factor was necessary to uncover relevant characteristics in each research article. We called these characteristics sub-factors; a total of 61 sub-factors were identified. Furthermore, measurement indicators were defined to assess these factors and sub-factors through a prototype tool. In this context, a guide is proposed to guide those responsible for an information technology area to evaluate a FLOSS in an agile and simple way.

This prototype offers three general recommendations that depend on the assessment made in the FLOSS adoption process. The development of a prototype to accelerate the FLOSS evaluation process is an important complement to this proposal, due to the flexibility that this tool offers to those responsible for software evaluation. In order to preliminarily validate the development of GUIOS, we have designed a pilot case study to be applied at the State University of Milagro. The results obtained

will be used to propose future improvements in the guide.

This research work opens a line of future research by proposing a guide for the FLOSS adoption process. It is also important to consider the exploration of other evaluation methods and techniques to offer an even more flexible and accurate adoption process. New functionalities should be incorporated in the prototype tool to allow greater flexibility in the use of the software and an optimal level of granularity.



**Parte I**

**Preliminares**

# Capítulo 1

## Introducción

Nuestro trabajo de investigación surge del aumento en la demanda de adoptar software libre en organizaciones públicas y privadas. Adoptar software no es una tarea sencilla de realizar, y es conveniente evaluar si el software a adoptar satisface los requisitos de la organización. Por este motivo, proponemos una guía que asista a los responsables del proceso de adopción a evaluar el FLOSS de una manera ágil y sencilla. Además, ofrecemos una herramienta prototipo que ayuda en dicho proceso.

En este capítulo damos una visión general de los conceptos principales que utilizamos en este documento y las contribuciones realizadas en este trabajo de investigación.

## 1.1. Contexto de la investigación

En esta sección se presentan brevemente los conceptos principales que utilizamos a lo largo de esta disertación. Primero presentamos el significado de software libre en la Sección 1.1.1. En la Sección 1.1.2 nos centramos en la adopción de software libre; finalmente, describimos los términos dimensión, factores y subfactores en la Sección 1.1.3.

### 1.1.1. Software libre, *open source* y FLOSS

A inicios de los años 80, se comienza a desarrollar el proyecto GNU (GNU no es Unix) por Richard Stallman <sup>1</sup>. El proyecto GNU tiene como objetivo la creación de un sistema operativo y herramientas libres. La decisión de desarrollar este proyecto fue motivada por restricciones impuestas por organizaciones de software privativo que no permitían trabajar de manera colaborativa con otros desarrolladores. Desde ese entonces, comienza a surgir el término software libre como una manera de escapar de las restricciones del software privativo [137].

Software libre se refiere a la libertad que poseen los usuarios para ejecutar programas en cualquier entorno, para modificarlo según necesidades de las instituciones, así como redistribuirlo [137].

El término libre ha creado confusión en algunos casos, pensando que el programa a adoptar de manera libre es gratis. Tampoco existe impedimento de vender copias de software libre, por lo tanto se ha convertido una manera de recaudar fondos para la comunidad que se dedica a mantener las actividades del software libre [137].

El término *open source* es, en práctica, equivalente. No obstante, incide en cómo se construye el software: ofreciendo acceso al código fuente y con un desarrollo de software sometido al escrutinio público. Los términos de distribución del *open source* deben cumplir principalmente criterios como la redistribución gratuita del software, el acceso al código fuente, la posibilidad de realizar modificaciones de software y la distribución del software modificado, entre otros criterios<sup>2</sup>.

FLOSS (por sus siglas en inglés *Free Libre Open Source Software*) es un acrónimo para unificar conceptos de software libre y *open source* [51]. FLOSS ha experimentado un crecimiento en su uso en los últimos 15 años y ha recibido una buena aceptación en administración pública, en entornos educativos y la industria [101].

### 1.1.2. Situación actual de la adopción de FLOSS

Las Tecnologías de la Información y Comunicación (TIC) están experimentando un aumento en su demanda a nivel mundial. En particular, afecta a países en vías de desarrollo, que tienen la desventaja del acceso limitado a estos sistemas a causa de los altos costes del software y hardware que imponen los grandes monopolios fabricantes de soluciones TIC [32].

---

<sup>1</sup><https://www.gnu.org/>

<sup>2</sup><https://opensource.org/osd>

En Latinoamérica, desde hace algunos años se está impulsando el uso de FLOSS frente al software privativo en busca de independencia de proveedores tecnológicos, seguridad y desarrollo local. Uno de los principales motivos para impulsar este cambio es la estabilidad y flexibilidad que garantiza el FLOSS, además que en actualidad suele ofrecerse en varios idiomas.

En el Ecuador, el entonces presidente Rafael Correa Delgado emitió un Decreto Presidencial No. 1014 con fecha 10 de Abril del 2008, en el cual establece una estrategia para la implantación de software libre para la administración pública central del país [32].

Como resultado de la implantación de esta estrategia y de la continuidad de la misma en el gobierno actual, a través del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, se muestra en la Figura 1.1 que se ha logrado el 64 % de implantación de software libre para los servicios estatales (sistemas operativos, bases de datos, entre otros) que ofrecen en sus plataformas digitales a la ciudadanía en general. Adicionalmente, el 43 % de software libre de escritorio ha sido instalado en las computadores para el desarrollo de las actividades propias de los servidores públicos (correo electrónico, aplicaciones de ofimática, gestión de proyectos, diseño gráfico, entre otros) [32].

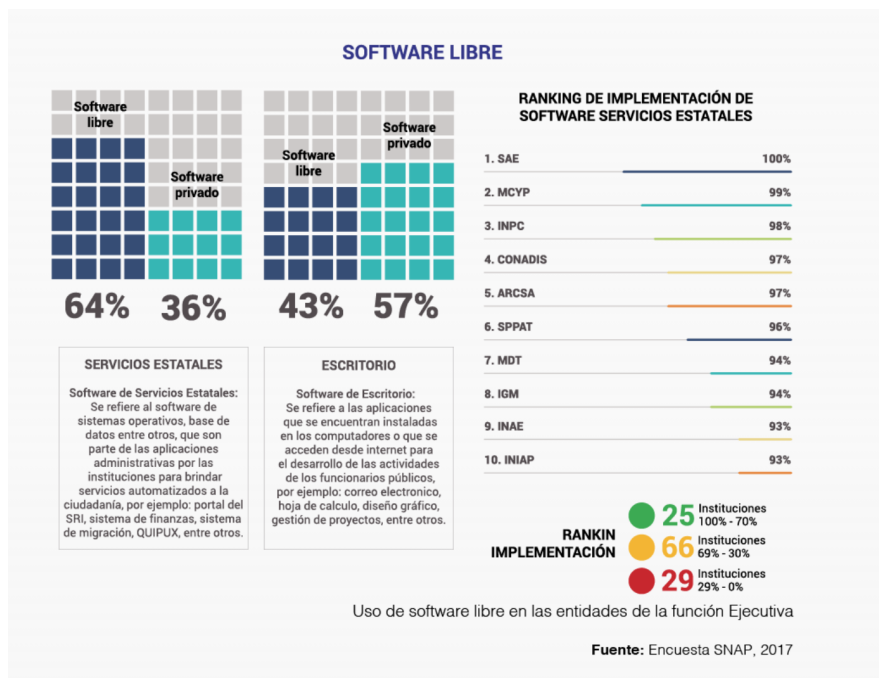


Figura 1.1: Uso del software libre en las entidades de la función ejecutiva del Ecuador.

Otros países han establecido estrategias de adopción FLOSS en sus administraciones públicas con resultados favorables. Por ejemplo, en Bélgica, Holanda ó Suecia han decidido utilizar algunas herramientas con estándares abiertos para facilitar las

operaciones internas entre instituciones públicas. La adopción FLOSS en las organizaciones públicas y privadas es un proceso que requiere la implantación de políticas bien estructuradas para una correcta selección del software a adoptar, optimizando de esta manera el gasto, fortaleciendo el desarrollo local y facilitando la inclusión digital<sup>3</sup>.

La selección del software a utilizar en una organización es, una actividad propensa a errores si no contamos con un proceso bien definido que oriente a los administradores de Tecnología de la Información (TI), a evaluar correctamente el FLOSS [101]. Este proceso es un conjunto de pasos sistemáticos conformados por factores y subfactores, los mismos que deberán ser evaluados para tomar una decisión efectiva.

Los aspectos a considerar en un proceso de adopción pueden tener distintos matices, tales como tecnológicos, organizacionales y económicos entre otros. Estos aspectos a evaluar dependerán de la necesidad propia de la organización en cuestión. Existen trabajos de investigación que proponen metodologías para que los administradores de TI seleccionen entre software de código abierto y software privativo [17]. También existen trabajos de investigación donde proporcionan recomendaciones sobre el uso del software libre en administraciones del sector público en Europa, basadas en un marco de política actual. El objetivo de estas recomendaciones de políticas, es proporcionar orientaciones y acciones que puedan ayudar a los gobiernos, las administraciones públicas y las organizaciones europeas a aprovechar plenamente los beneficios del software libre [16].

### 1.1.3. Dimensiones, factores y subfactores que determinan la adopción de FLOSS

Una dimensión es un conjunto de factores que repercuten en el proceso de adopción de software. Un factor es una característica que cubre un aspecto relevante del proceso de adopción. En este trabajo de investigación, los factores se convierten en una base importante para la construcción de una guía para la adopción de FLOSS. Los factores son agrupados según su relación en cada una de las dimensiones definidas. En algunos trabajos de investigación el término *factor* ha sido utilizado también como sinónimo de los términos: criterios, aspectos, características, entre otros [39, 64, 118, 134].

Siguiendo a Petersen en [102], donde define dimensiones para determinar el alcance de un investigación sistemática, nosotros utilizamos el término dimensión para:

- Identificar el tipo de investigación de los estudios primarios revisados en este trabajo de investigación.
- Agrupar factores comunes en el proceso de adopción de FLOSS.

---

<sup>3</sup><https://www.gobiernoelectronico.gob.ec/>

## 1.2. Contribuciones

Esta sección detalla la secuencia en la que se han obtenido los resultados de la tesis. En la Sección 1.2.1 se presenta un resumen de las contribuciones principales.

### 1.2.1. Resumen de las contribuciones

En esta sección resumimos las principales contribuciones de nuestra tesis y el trabajo de investigación.

Por un lado, nuestra tesis se centra en la construcción de una guía que ayude a evaluar factores para la adopción de FLOSS en una organización. Hemos analizado propuestas actuales que se relacionan con la adopción de FLOSS, y concluimos que no existe una guía completa que permita una evaluación adecuada en el proceso de adopción de FLOSS. La ausencia de un proceso formal bien definido de adopción de FLOSS introduce una serie de amenazas para la organización, tales como que los ficheros de datos no se puedan importar con el software a adoptar, la falta de documentación de usuarios, costo ocultos o resistencia al cambio del uso del software [6, 10, 40, 49].

Las contribuciones de esta tesis se resumen a continuación:

- Identificación de dimensiones, factores y subfactores: con base a la identificación de las dimensiones, factores y subfactores que determinan la adopción de FLOSS a través de la literatura existente. El objetivo de esta contribución, es obtener criterios de medición que serán incluidos en la guía para el proceso de evaluación del FLOSS a adoptar.
- Definición de indicadores para evaluar la importancia de factores y subfactores: los indicadores tienen como objetivo caracterizar la importancia de los factores en el contexto de la adopción de FLOSS. Se definieron cinco indicadores de importancia, donde se propone una metodología basada en técnicas de evaluación cuantitativa y cualitativa.
- Elaboración y aplicación de una encuesta a expertos sobre la adopción de FLOSS con preguntas de respuestas cerradas. Esta encuesta tuvo por objetivo conocer la importancia que los expertos le otorgan a los subfactores identificados en este trabajo de investigación. Adicionalmente, a través de esta encuesta los expertos podían retroalimentar acerca de nuevos subfactores.
- Diseño de GUIOS: es una guía para evaluar factores que repercuten en la adopción de FLOSS en una organización. Esta guía tiene por objeto asistir a los responsables de administrar los recursos tecnológicos de las organizaciones en la idoneidad de la adopción del software.
- Construcción de GUIOS PRO: es una aplicación prototipo multi-plataforma, caracterizada a través de un toolkit de Python llamado Flexx versión 0.80. Esta

tecnología es un FLOSS distribuido bajo la licencia AGPL3. El objetivo de GUIOS PRO es automatizar los pasos de GUIOS.

- Diseño de un piloto de un caso de estudio: es una validación preliminar de la guía propuesta aplicado en un área TIC de una universidad pública en Ecuador. El objetivo es analizar los resultados para la mejora a futuro de GUIOS.

### 1.3. Estructura de esta disertación

Esta disertación está organizada de la siguiente manera:

**Parte I: Preliminares.** En la primera parte de esta tesis mostramos una visión general de los conceptos más relevantes en el ámbito de software libre y *open source*.

**Parte II: Antecedentes.** En la segunda parte de la tesis exploramos información básica necesaria para comprender los objetivos de esta tesis. Aclaremos el término FLOSS (Free Libre Software Libre), *open source* (código abierto), así como exploramos las distintas licencias FLOSS disponibles. Finalmente, se hace un repaso sobre como FLOSS ha sido objeto de estudio en el ámbito científico para motivar este trabajo de investigación.

**Parte III: Contribuciones.** Esta parte es el núcleo de nuestra tesis y está organizada de la siguiente manera. En el Capítulo 4, presentamos las dimensiones y factores de adopción identificados a través de una revisión sistemática de la literatura. En el Capítulo 5, presentamos el proceso para la construcción de la guía para la adopción de FLOSS en organizaciones públicas y privadas. En el Capítulo 6, se describen los pasos a seguir para el uso de la guía propuesta y la validación de la misma a través de un piloto de caso de estudio.

**Parte IV: Observaciones finales.** En el Capítulo 7, se presentan las conclusiones y trabajos futuros por realizar basado en los resultados obtenidos.

**Parte V: Apéndice.** Se presenta la encuesta aplicada a expertos FLOSS en el Apéndice A.

**Parte II**

**Antecedentes**



## Capítulo 2

# FLOSS en la actualidad

En este capítulo, nos centramos en los fundamentos básicos relacionados con FLOSS, describiendo los aspectos más comunes del entorno de FLOSS y las tendencias actuales en la investigación científica. En primer lugar, en la Sección 2.1 se revisará la definición de FLOSS, destacándose las principales diferencias entre los movimientos software libre y *open source*. En la Sección 2.2 se describen las licencias FLOSS más relevantes. Finalmente, en la Sección 2.3 resumimos el capítulo.

## 2.1. ¿Qué es FLOSS?

El término *Free/Libre/Open Source Software* (FLOSS) se usa para resumir los diferentes términos y filosofías existentes sobre el software, cuyo código fuente es accesible por los desarrolladores de software [139]. FLOSS, es un hiperónimo para designar software que ha sido concebido bajo los movimientos: *Free Software* (software libre) promovido por la *Free Software Foundation*<sup>1</sup>, y *Open Source Software* establecido por la *Open Source Initiative*<sup>2</sup>.

El hecho de que un software determinado sea catalogado como *libre*, significa que los usuarios pueden ejecutar el software, cambiarlo según sus necesidades y distribuirlo, casi sin ninguna limitación [138]. No obstante, el término *free*, trajo consigo algunas confusiones, especialmente porque en idioma inglés es empleado frecuentemente para designar la gratuidad de algún producto o servicio. En este contexto, se podría interpretar que el proceso de adquisición del software es gratuito (algo que no es necesariamente cierto). Sin embargo, en el contexto de la *Free Software Foundation*, el término tenía por objetivo denotar el carácter libre del software en sí. Por tales razones, a menudo el término *Libre* es empleado en lugar de *Free*, para evitar tal confusión.

En este sentido, no es difícil encontrar notas aclaratorias en diversos escenarios que buscan dejar claro el sentido semántico del término. Por ejemplo, aparecen notas como *free as in “free speech”*, y no como *as “free beer”* [139]. Esta libertad se enfoca en diversos aspectos del software, tales como la utilización, distribución, modificación y copia del software.

Si bien el movimiento de software libre fue promovido en los años 80 del siglo pasado, el *open source* es más reciente (finales de los 90, inicios de los 00s). Su filosofía se centra más en el desarrollo del software que en los aspectos ideológicos. No obstante, según el propio Richard Stallman (padre del movimiento de software libre), en general, ambos términos describen a la misma comunidad que defiende objetivos comunes, pero desde perspectivas diferentes [57].

De manera que FLOSS se usa en esta tesis como un término para dar a entender que los usuarios son libres de ejecutar, alterar y distribuir el software e incluso vender con algunas limitaciones. En otras palabras, cualquier software catalogado con cualquiera de las siglas OSS, FOSS o FL/OS será considerado como FLOSS en esta investigación.

A continuación se profundiza más en estos dos movimientos con el objetivo de dejar claros sus principales fundamentos.

### 2.1.1. Software libre

El movimiento software libre es en términos generales una filosofía. Un enfoque para concebir el proceso de elaboración y distribución del software. La primera

---

<sup>1</sup><https://www.fsf.org/>

<sup>2</sup><https://opensource.org/>

alusión a esta filosofía, fue hecha por Richard Stallman en 1983, quien estableció cómo deberían relacionarse los desarrolladores con los usuarios de un software. Su principal motivación surge como respuesta a esa autoridad que el software privativo imponía (y aún lo sigue haciendo) a sus usuarios. La cual se traduce en que el software pertenece exclusivamente al desarrollador, y el usuario adquiere una licencia limitada para su uso. Basado en un principio social, Stallman buscaba romper con esa relación de dominio público, permitiendo que los usuarios pudiesen aprovechar al software de acuerdo a sus necesidades individuales, esto es, con plena libertad.

En el sitio web del movimiento del software libre se describen las siguientes libertades<sup>3</sup>:

*Libertad 0.* La libertad de ejecutar el programa, sin importar el propósito a realizar.

*Libertad 1.* La libertad de revisar la funcionalidad del programa, y modificarlo de acuerdo a las necesidades del usuario. El acceso al código fuente es requerido para esta realizar esta actividad.

*Libertad 2.* La libertad de redistribuir copias del software para compartirlas con los colegas de la comunidad.

*Libertad 3.* La libertad de distribuir copias de sus versiones modificadas a terceros. En caso de realizarlo, sería de gran beneficio que la comunidad aproveche las modificaciones realizadas. Es necesario el acceso al código fuente para realizar esta actividad. Si se cumple esta libertad, se estaría proporcionando a la comunidad oportunidades más flexibles de cambio. Se requiere acceso al código fuente para cumplir esta libertad.

Al interpretar cada una de estas libertades, podemos inferir que se tratan de derechos que comúnmente las personas obtendrían al adquirir un bien intangible como es el caso del software. Algo que en el software de tipo privativo no sería posible, debido a las restricciones de propiedad intelectual en licencias de software que manejan.

La *Libertad 0*, habilitaría al usuario para que pueda utilizar el software según sus necesidades. Esto es, utilizar el software sin necesidad de realizar algún esfuerzo adicional, por ejemplo, realizar alguna modificación previa a su uso. La intención de esta libertad es superar los obstáculos impuestos a los usuarios en algunas licencias por parte de la industria de software privativo. Como por ejemplo WordPress, un FLOSS que se puede usar para crear páginas webs, blogs y aplicaciones<sup>4</sup>.

Por su parte, la *Libertad 1*, es obviamente un derecho reservado para usuarios más avanzados. Esto es, con conocimientos técnicos suficientes como para comprender las principales funcionalidades del software. Esto implica que los usuarios accedan al código fuente con el propósito no solo de revisar cómo funciona, sino también de modificarlo conforme a su necesidad.

<sup>3</sup><https://www.gnu.org/philosophy/free-sw.html.en>

<sup>4</sup><https://wordpress.com/es/>

La *Libertad 2*, busca que el usuario de un programa cuente con el privilegio de compartir una copia del programa libremente. Se trata posiblemente de una de las libertades más controvertidas, pues podría verse como un prejuicio para el desarrollador (autor) del software. La razón es simple, éste podría ver como su creación se distribuye por el mundo sin ser retribuido por ello.

Con la *Libertad 3*, el movimiento del software libre busca alcanzar un impacto social favorable. Esto es, permitir que el software desarrollado por terceros, pueda ser compartido entre los usuarios que utilizan el software en sí. Esta característica actúa como una base importante para fomentar la existencia de comunidades FLOSS.

Una importante implicación de las libertades 1 y 3 es que el código del software se encuentre disponible. De otra forma, sería imposible gozar de los privilegios de ambas libertades. Son estas libertades también las que conectan a este movimiento con el *open source*, como se verá a continuación.

### 2.1.2. Software *open source*

El software *open source* puede verse como una extensión o reformulación del movimiento de software libre. Sus planteamientos estarían dirigidos a favorecer el beneficio económico del desarrollador [60]. De acuerdo al sitio oficial del movimiento de software *open source*, sus principales características son las siguientes<sup>5</sup>:

- *Libre redistribución del código*: la licencia del software no debe impedir la distribución el código fuente del software, ni tampoco la necesidad de pagar un valor por acceder al mismo.
- *Código fuente*: el software viene con el código fuente incorporado, esto es, integrado al software. En caso de no distribuirse conjuntamente con el código fuente, debe proveer un sistema flexible que propicie el acceso al código fuente.
- *Trabajos derivados*: la licencia incluida en el software permite que cualquier persona pueda modificar, incluso hasta trabajos de terceros. Asimismo, la licencia se distribuiría con estas modificaciones bajo las mismas condiciones del software original.
- *Integridad del código fuente del autor*: la versión modificada de un programa puede ser restringida por una licencia solo en el caso específico que así lo determine su fabricante. Esta condición debe de estar claramente definida en los términos de la licencia. Asimismo, que los programas lleven un nombre o número de versiones diferentes, es responsabilidad exclusiva del fabricante de la licencia.
- *No discriminación de personas o grupos*: no limitar a personas o grupos de personas a que contribuyan en la optimización del proceso del desarrollo o

---

<sup>5</sup><https://opensource.org/docs/definition.php>

modificación del software. En ese sentido, las licencias no restringen a que las personas puedan participar en el desarrollo de software. Sin embargo, pueden existir limitaciones legales según el país donde se fabrica el software y eso depende de las normativas que regulan el software en cada país. Para estos casos concretos, la licencia debería de ser explícita para evitar confusiones.

- *No limitación al uso del software*: la licencia debe tener las claramente definidas las condiciones al momento de describir el uso de los programas.
- *Aplicación universal de la licencia*: los programas *open source* deben tener derechos que se apliquen a nivel universal. Por lo tanto, no es posible que se agreguen requisitos adicionales a los ya descritos en la licencia original.
- *La licencia no debe restringir otro software*: es evidente que existen diferencias entre software libre y el *open source*, uno de los principales debates es no utilizar *copyleft* en sus licencias. El *open source* sostiene la teoría en que esta licencia no debe establecer limitaciones sobre otro programa que pueda ser distribuido en conjunto con el programa al que se le aplica la licencia.
- *La licencia debe ser tecnológicamente neutral*: la licencia del software debe tener plena libertad sin estar atadas a tecnologías o interfaces gráficas específicas.

### 2.1.3. Diferencias entre software libre y *open source*

A priori, no resulta fácil encontrar diferencias entre las libertades de software libre y las características fundamentales del *open source*. Por lo que consideramos que ambos movimientos tienen objetivos similares pero con un enfoque diferente. En nuestra opinión, entendemos que el movimiento de software libre es una filosofía que defiende la libertad de los usuarios con respecto al uso del software.

Las libertades del movimiento de software libre son importantes no solo para el usuario individual sino para una comunidad global, debido a que promueven el compartir y cooperar de manera conjunta. Mientras el uso de la tecnología se vuelve una actividad más frecuente en las organizaciones, la importancia de las libertades del movimiento de software libre crece proporcionalmente.

En cambio, el movimiento *open source*, se enfoca en cómo se construye el software, es decir, la forma de construir la tecnología y acceso al código, por lo que los beneficiados son principalmente los desarrolladores, sin establecer ni defender principios. Esta es una de las razones principales por la que movimiento de software libre está en desacuerdo con el movimiento *open source* y evitan usar ese término<sup>6</sup>.

Un ejemplo que ilustra las diferencias fundamentales entre ambos movimientos es la paradoja de la gestión de restricciones digitales (Digital Right Management, DRM, por sus siglas en inglés) [139]. DRM impone restricciones de uso y modificación de software y hardware privativo que no son compatibles con las libertades que

---

<sup>6</sup><https://www.gnu.org/philosophy/open-source-misses-the-point.html>

garantiza el Free Software. Sin embargo, sería posible implementar un DRM *open source*, por que la idea principal del movimiento *open source* con este proyecto, es promover una arquitectura *open source* para la DRM que contemple todos los dispositivos, sin importar el fabricante, es decir, asignarle los derechos a los usuarios en lugar que a los dispositivos. La empresa Sun Microsystems está tomando la iniciativa para desarrollar un estándar DRM independiente del fabricante, llamado DRM Opera y de esta manera proponer este proyecto a más empresas para uso universal<sup>7</sup>.

## 2.2. Licencias FLOSS

En términos generales, una licencia de software es un contrato legal entre el autor y el usuario de un software donde se establecen términos y condiciones que el usuario tiene que cumplir. El autor elige el tipo de licencia con el cual se distribuirá el software, y así, los usuarios tengan conocimiento de lo que pueden hacer o no con el software a explotar.

En general, hay dos grandes familias de licencias FLOSS: las licencias *copyleft* y las licencias permisivas. Las licencias *copyleft* obligan en mayor o menor medida a que el usuario devuelva el resultado de sus mejoras a la comunidad, mientras que las licencias permisivas no establecen obligaciones de este tipo. Existen licencias permisivas que no establecen restricciones de ningún tipo al usuario (similares al concepto de dominio público) tales como Creative Common Zero (CC0) o la licencia Unlicensed que permiten al usuario hacer lo que quieran con el software, incluyendo la posibilidad de implementar software privativo derivado.

A continuación realizamos una revisión de las licencias más relevantes del movimiento del software libre y el *open source*.

### 2.2.1. ¿Qué es *copyleft*?

*Copyleft* es, en palabras de la propia *Free Software Foundation*:

*“un método general para liberar un programa u otro tipo de trabajo (en el sentido de libertad, no de gratuidad), que requiere que todas las versiones modificadas y extendidas sean también libres”*<sup>8</sup>

En este caso, por método se entiende práctica legal para transmitir las condiciones de libertad de un software a sus posteriores extensiones y modificaciones. Es además, para su creador Richard Stallman, un principio que busca “difundir la libertad y la cooperación”<sup>9</sup>.

Para entender mejor en qué consiste el término *copyleft*, tengamos en cuenta que todo software que desea ser libre pero sin obligaciones deberá pertenecer a un do-

<sup>7</sup><https://opensource.com/>

<sup>8</sup><https://www.gnu.org/licenses/copyleft.es.html>

<sup>9</sup><https://www.gnu.org/philosophy/pragmatic.html>



Figura 2.1: Categorías de licencia de software.

10

minio público, es decir, puede ser modificado y extendido por terceros. Un software desarrollado bajo esta categoría perderá los derechos del autor original.

Este vacío legal lo sufrió el propio Stallman cuando le fue negado el acceso a las modificaciones que la compañía *Symbolics* le había hecho a su intérprete LISP y que éste le había proporcionado bajo dominio público. Como respuesta a esta práctica, Stallman decidió crear su propia licencia con reglas más justas. Nacen así en la década de los 80s del siglo pasado, la *Emacs General Public License*, y posteriormente la popular *GNU General Public License (GPL)*.

De esta forma, el autor o titular de los derechos de un software podía garantizar que la totalidad de derechos se transfiriera perpetuamente a los usuarios de un software, esto es, sin importar las revisiones posteriores que alguien hiciera al programa original. No obstante, es importante mencionar que esa GPL inicial no otorgaba derechos al público en general, sino solo a aquellos que ya habían recibido el programa.

El proyecto GNU tiene por objetivo ofrecer un sistema operativo y utilidades de usuario bajo licencias de tipo *copyleft*. Las licencias *copyleft* hacen uso de las leyes de copyright para asegurar que cualquier mejora en el software se devuelve a la comunidad de usuarios y desarrolladores del proyecto GNU. De manera despectiva, se suele decir que las licencias *copyleft* son “virales”, pues el software desarrollado a partir de otro software bajo licencia *copyleft* se “contagia” de estas obligaciones.

En concreto identificamos que las licencias del proyecto GNU tienen la obligación de extender la libertad del software original a las copias o versiones posteriores al mismo. Un requerimiento que es el núcleo de las licencias de tipo *copyleft*.

### 2.2.2. Licencias *copyleft*

Además de ser un método (práctica legal) y principio (idealismo), *copyleft* es un artefacto para obligar a compartir software haciendo uso de las leyes copyright. La Figura 2.1 muestra como FLOSS tiene dos categorías de *copyleft*: *copyleft* y *copyleft* débil.

Primero, que toda licencia *copyleft* contiene un copyright (del titular del software), el cuál le permite definir posteriormente las condiciones en que será distribuido.

En otras palabras, un software no “nace” siendo libre, primero es necesario establecer su autoría, y luego liberarlo<sup>11</sup>. Por otro lado, aunque la gráfica de la Figura 2.1 sugiera diferentes grados de libertad, desde lo más libre (licencias de dominio público) hasta lo más restrictivo (licencias de software privativo), es importante tener en cuenta solo las licencias *copyleft* garantizan traspasar esa libertad de uso a los derivados del software original. Dado que el principio del movimiento de software libre es mantener esa libertad, las licencias que mejor representan a FLOSS son de tipo *copyleft*, *copyleft* débiles y no *copyleft*. Las dos primeras son propias del movimiento de software libre, mientras que la última se puede encontrar en varios proyectos *open source*.

Las licencias *copyleft*, son muy estrictas en cuanto a mantener en las modificaciones y derivados del software original, las intenciones de éste último. Las implementaciones más notables de esta licencia son las GPL2, GPL3, y AGPL. Mientras que las dos primeras licencias son habituales en el ámbito del software tradicional, la última se emplea generalmente en el ámbito de las redes. En esencia, AGPL garantiza que el código fuente estará disponible para los usuarios del software a través de la red.

Hay que tener en cuenta que, no es recomendable que FLOSS sea de dominio público debido a que posibilita que cualquier persona sin ánimo de trabajar cooperativamente, convierta al software en privativo.

### 2.2.3. Licencias *copyleft* débil

Las licencias *copyleft* débiles difieren de la anterior en la relajación de las condiciones a los derivados del software original. Un escenario típico donde este tipo de licencia aparece con frecuencia es en el que un software determinado, sea este de cualquier tipo de licencia, usa una biblioteca de software *libre*. Dado que las licencias *copyleft* son muy restrictivas, sería imposible que bajo este tipo de licencia las bibliotecas de software pudieran usarse por otro software con licencia privativa. Por tanto las condiciones de la licencia *copyleft* solo aplica a las bibliotecas, no al software derivado.

Ejemplos de implementaciones de esta última licencia son la LGPL 2.1/3, Mozilla Public License (MPL) y Eclipse (EPL).

### 2.2.4. Licencias no *copyleft*

Las licencias no *copyleft* no obligan al usuario a devolver a la comunidad cualquier contribución o mejora en el software. No obstante, suelen establecer condiciones tales como publicitar al autor del software original en la web y la documentación del software derivado.

Como consecuencia, cualquier software no *copyleft* puede convertirse en software privativo. Las licencias no *copyleft* más populares son: 2BSD, 3BSD, Apache, MIT, entre otras.

---

<sup>11</sup><https://www.gnu.org/philosophy/pragmatic.html>



Las licencias de tipo dominio público son aquellas en las que el autor manifiesta que no impone restricciones de ningún tipo a los usuarios del software. La licencia *Unlicense*<sup>12</sup> y la CC0<sup>13</sup> son ejemplos de este tipo de licencia.

### 2.3. Resumen

Al término del presente capítulo, se puede concluir de manera parcial que:

- El término FLOSS es un hiperónimo para referirse a dos movimientos que si bien persiguen objetivos comunes sobre derechos de un usuario (final y/o desarrollador) sobre un software en específico, poseen diferencias de enfoques.
- Mientras el movimiento software libre constituye una filosofía enfocada en principios éticos, el movimiento de software *open source* constituye una forma de desarrollar software, una metodología cuyo pilar es el acceso al código fuente, así como su modificación y redistribución.
- FLOSS no es sinónimo necesariamente de software gratis. Tanto un software libre como *open source* puede ser comercializado.
- Las licencias *copyleft* son parte esencial en la distribución de FLOSS, y se agrupan en dos categorías: *copyleft* y *copyleft débil*.

---

<sup>12</sup><https://unlicense.org/>

<sup>13</sup><https://creativecommons.org/publicdomain/zero/1.0/deed.es>

## Capítulo 3

# Motivación

La adopción de FLOSS ha experimentado un incremento notable en la última década. La adopción se está realizando de manera rápida, sin seguir procedimientos formales. Existen metodologías que abarcan de manera parcial el problema de la adopción de FLOSS, sin embargo, no ofrecen un itinerario bien definido en dicho proceso. La identificación de los factores que determinan el proceso de adopción de FLOSS es fundamental para garantizar que la adopción de FLOSS se realice con éxito<sup>1</sup>. Es por ello que vemos conveniente elaborar una guía que asista a los responsables de TIC en el proceso de adopción de FLOSS.

En este capítulo ofrecemos un resumen de los trabajos relacionados con la adopción de FLOSS. Concretamente, en la Sección 3.3, hemos resumido la literatura existente que identifica los factores que determina la adopción de FLOSS. En la Sección 3.4, se detallan las metodologías existentes que facilitan el proceso de adopción de FLOSS. Finalmente, en la Sección 3.5 ofrecemos un resumen de las aportaciones realizadas en este capítulo.

---

<sup>1</sup>[https://www.iima.ac.in/web/cdt/assets/Adopting%20Open%20Source%20Software%20Report%202021\\_final.pdf](https://www.iima.ac.in/web/cdt/assets/Adopting%20Open%20Source%20Software%20Report%202021_final.pdf)

### 3.1. Introducción

En 2015, de acuerdo con una encuesta elaborada por North Bridge-Blackduck, el 66 % de los responsables de TIC consideraba que FLOSS era una opción por delante del software privativo <sup>2</sup>.

En 2022, el 92 % de los responsables de TIC considera que FLOSS es fundamental para afrontar sus retos relacionados con la COVID, según un informe realizado por Red Hat en el que participaron 1250 responsables TIC<sup>3</sup>. Sin embargo, la adopción de FLOSS se está realizando rápidamente sin seguir una metodología formal, lo que supone un riesgo para la organización [51, 63, 101, 157]. Existen metodologías que parcialmente abarcan el proceso de adopción FLOSS desde diferentes perspectivas [4, 94, 104]. También existen otros temas relacionados con la adopción de FLOSS, como por ejemplo, propuestas sobre políticas para la selección de FLOSS [16, 38, 65], marcos de trabajo para adopción de software [29, 149, 150] pero no ofrecen una guía con itinerarios bien definidos para la adopción con éxito de FLOSS.

Un desafío para cubrir esta debilidad, es automatizar el proceso de adopción de FLOSS a través de una guía que oriente a los responsables de TIC a evaluar el software a adquirir de una manera ágil.

En este capítulo se detallan los trabajos relacionados con:

- FLOSS en el ámbito científico: conocer la presencia de FLOSS en el ámbito científico permite identificar perspectivas diferentes en la que FLOSS ha ganado mayor interés para su adopción por parte de la comunidad. Diseñar instrumentos de evaluación de FLOSS sería una contribución importante para que los responsables de TIC cuenten con una herramienta para un proceso de evaluación estructurado de adopción FLOSS .
- Identificación de factores de adopción FLOSS: identificar factores en un proceso de adopción FLOSS permite definir una estructura para que el responsable de la evaluación del software pueda valorar con mejor precisión. Estos factores pueden ser evaluados desde varias perspectivas según las necesidades de la organización.
- Metodologías que faciliten la adopción FLOSS: automatizar un proceso de adopción FLOSS, requiere que se implanten procedimientos bien definidos para una mejor evaluación de dicho proceso. A través de una metodología establecida por expertos en esta área específica, se puede conseguir que una persona responsable de la selección de un FLOSS disponga de una guía que ayude a agilizar este proceso de adopción.

---

<sup>2</sup>[https://www.slideshare.net/blackducksoftware/2015-future-of-open-source-survey-results/11-CASE\\_STUDYCUSTOMERCASE\\_STUDY\\_FORweathercom\\_by](https://www.slideshare.net/blackducksoftware/2015-future-of-open-source-survey-results/11-CASE_STUDYCUSTOMERCASE_STUDY_FORweathercom_by)

<sup>3</sup><https://www.redhat.com/en/enterprise-open-source-report/2022>

### 3.2. FLOSS en el ámbito científico

FLOSS ha sido objeto de investigación científica desde diferentes áreas de estudio desde su surgimiento. Es un fenómeno que, por sus características especiales, como el desarrollo de grandes proyectos en línea por una comunidad de desarrolladores anónimos, ha resultado atractivo para numerosos investigadores.

Para brindar una panorámica general de las investigaciones más recientes sobre FLOSS, nos hemos centrado sólo en aquellos trabajos publicados desde el 2014 en revistas científicas. Nuestro objetivo es resaltar las mayores contribuciones sobre el tema en los últimos 8 años, el cual consideramos un periodo aceptable para resumir las principales tendencias. Es importante mencionar que se han dejado fuera publicaciones donde FLOSS es un aspecto secundario, como por ejemplo aquellas en donde se emplea un FLOSS para automatizar o mejorar algún proceso.

Como se aprecia en la Figura 3.1, desde 2004, el interés sobre FLOSS se ha mantenido y la cantidad acumulada de trabajos sigue creciendo en el tiempo. Un claro indicativo de que se trata de un tema científico de interés.

La Tabla 3.1 contiene información más detallada de los artículos obtenidos en la búsqueda realizada. Como se observa, hemos agrupado los estudios de acuerdo a dos criterios generales: el principal tema abordado en la investigación y la relevancia del trabajo. Este último factor fue calculado para cada artículo empleando la teoría de eficiencia de Pareto [100], específicamente teniendo en cuenta dos criterios: el número de citas del artículo y el año de publicación. En este sentido, un artículo es de relevancia 1 si pertenece al frente de Pareto óptimo (conjunto de soluciones no dominadas), es de relevancia 2 si, quitando a los de relevancia 1, pertenece al frente de Pareto óptimo, y así sucesivamente.

A partir de la Tabla 3.1 es posible ver que los trabajos más importantes hoy en día sobre FLOSS están orientados a diversos temas. Resaltan los casos de comunidades FLOSS, desarrollo de FLOSS, y enseñanza y aprendizaje de FLOSS. En particular, el tema evaluaciones, selección y adopción aunque posee varios artículos, estos no aparecen dentro del grupo con relevancia 1 (aquellos que logran un balance óptimo entre citas y actualidad). No obstante es un claro indicativo de que, aunque el tema resulta de interés por parte de la comunidad científica internacional, existe aún oportunidades de investigación para desarrollarlo convenientemente. La Figura 3.2 confirma esta percepción al ubicar a este tema en una posición intermedia en los gráficos de barras.

Tema	Descripción	Relevancia			
		1	2	3	≥4
Comunidades FLOSS	Estudios sobre las dinámicas, interacciones, etc. entre las personas vinculadas al desarrollo y mantenimiento de FLOSS.	[167], [108]	[25], [146]	[27]	[136], [44], [75], [71], [22]

Continúa en la siguiente página

Tabla 3.1 – continuación de la página anterior

Tema	Descripción	Relevancia			
		1	2	3	≥4
Desarrollo de FLOSS	Estudios sobre aspectos técnicos del desarrollo de FLOSS.	[28], [15]		[144], [158], [131]	[37], [20], [89], [126], [105], [24]
Enseñanza y aprendizaje de FLOSS	Estudios que abordan a FLOSS desde una perspectiva educativa.	[18]			[77], [92], [57], [55], [151]
Computación en la nube	Estudios sobre la gestión de FLOSS en la nube.	[58]			
Seguridad y privacidad	Estudios que investigan sobre aspectos de seguridad y privacidad en FLOSS.	[3]			
Soporte técnico y gestión de errores	Estudios sobre la gestión del soporte técnico y gestión de errores de FLOSS.	[26], [78]		[159]	
Aplicaciones de FLOSS	Estudios que aplican FLOSS en la solución de problemas concretos.			[53], [19]	[84], [166], [135], [117], [110], [45], [132]
Evaluación, selección y adopción	Estudios que abordan los procesos de evaluación, selección y adopción de un FLOSS.		[118], [1]	[148], [7]	[82], [124]
Temas generales y cultura FLOSS	Estudios sobre la filosofía, historia, cultura y buenas prácticas en FLOSS.		[139]		[65], [141], [76], [128], [97]
Evolución de FLOSS	Estudios que abordan el proceso de evolución en el desarrollo de un FLOSS.		[36]		[12]
Metodologías de desarrollo de FLOSS	Estudios que proponen, analizan o evalúan metodologías de desarrollo de software.		[50]		
Financiamiento	Estudios que analizan aspectos económicos y modelos de financiamiento de FLOSS.			[62]	
Interacción de usuarios	Estudios que investigan aspectos de interacción con el usuario final.				[87]
Licencias	Estudios que abordan el aspecto legal de FLOSS, sus licencias y formas de uso.				[46]

Tabla 3.1: Distribución de artículos sobre FLOSS de acuerdo al principal tema que se aborda y el nivel de relevancia.

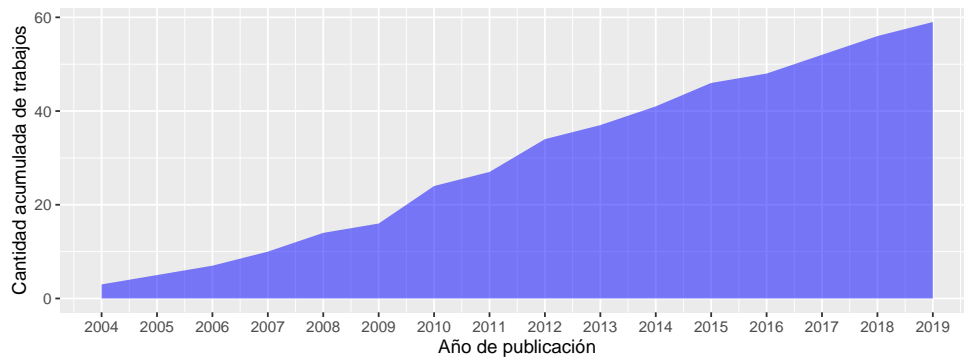


Figura 3.1: Evolución de artículos sobre FLOSS en revistas indexadas por Scopus.

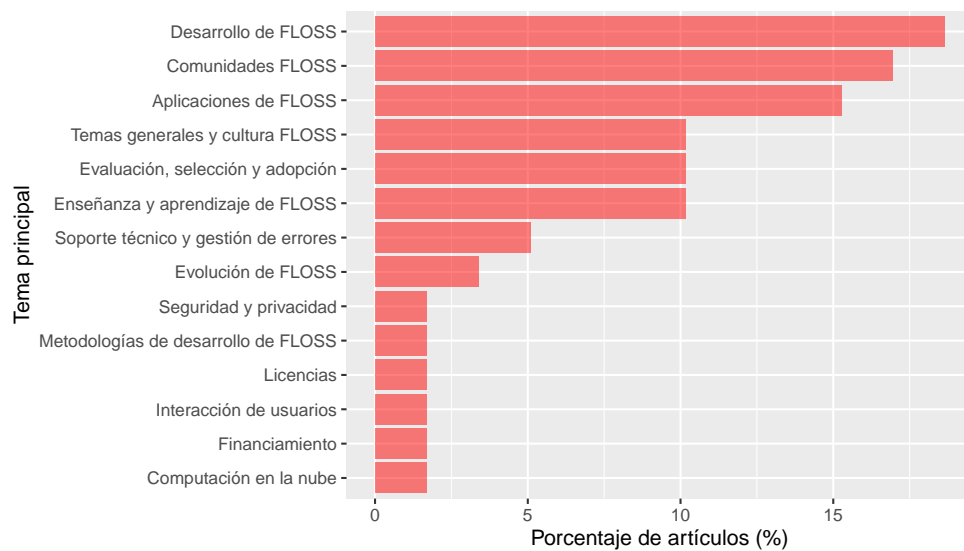


Figura 3.2: Distribución de la cantidad de artículos de revista sobre FLOSS indexados en Scopus desde 2004.

### 3.2.1. Análisis de la literatura

En esta sección presentamos un análisis de la literatura basada en los trabajos publicados en revistas, que abordan de alguna forma a FLOSS, y que se encuentran indexados en la base de datos Scopus<sup>4</sup>. Concretamente, se empleó la consulta que se muestra en la Tabla 3.2:

En el 2014, Kamp [62] expone la necesidad de dotar a los desarrolladores de FLOSS de algún tipo de financiación. La idea es garantizar un nivel de calidad adecuado de los proyectos FLOSS. Según este autor, la única manera de mejorar la calidad de FLOSS es logrando que los desarrolladores pasen tiempo en él. Necesitan

<sup>4</sup><https://www.scopus.com>

---

Cadena de búsqueda
TITLE (foss OR floss OR “Free/libre open source software” OR “free software” OR “open source”) AND ( LIMIT-TO ( SRCTYPE , “j” ) ) AND ( LIMIT-TO ( DOCTYPE , “ar” ) OR LIMIT-TO ( DOCTYPE , “re” ) ) AND ( LIMIT-TO ( SUBJAREA , “COMP” ) ) AND ( LIMIT-TO ( LANGUAGE , “English” ) )

---

Tabla 3.2: Consulta aplicada en la base de datos Scopus para la obtención de artículos científicos FLOSS.

tiempo para revisar los envíos de parches con cuidado, escribir y ejecutar casos de prueba, responder y corregir informes de errores, y programar. Sin embargo, dada la naturaleza humana de los desarrolladores, es lógico que éstos perciban alguna ayuda económica para al menos solventar sus necesidades básicas. En tal sentido, el autor expone algunas formas específicas de financiar proyectos FLOSS.

Por otro lado, Wei *et. al* [159] investigaron como la comunidad FLOSS es capaz de mantener grupos específicos de trabajo, dentro de los equipos de desarrollo. Desde una perspectiva sociolingüística, los autores caracterizaron el comportamiento de ese mantenimiento grupal a través de tácticas de comunicación interpersonal. Específicamente, estudiando la presencia social y las tácticas de cortesía, que interviene en el mantenimiento de las relaciones entre los miembros del grupo. Para llevar a cabo el estudio, los autores se basaron en los mensajes de correo electrónico que intercambiaron los desarrolladores de dos proyectos FLOSS con diferentes estados de terminación. En tal sentido, se pudieron encontrar diferencias que reflejan cambios en las prácticas de trabajo del proyecto.

Las leyes que describen y gobiernan la evolución de un proyecto FLOSS fue el tema principal investigado por Gonzalez-Barahona *et. al* [46]. El artículo examina en detalle la evolución del proyecto FLOSS *glibc*<sup>5</sup>, con el propósito principal de comprender cómo evolucionó y cómo se ajustó a las leyes de la evolución del software de Lehman [70]. Como resultado, los autores desarrollaron una metodología para estudiar la evolución de proyectos de larga duración (ej. con más de 15 años), esto es, basados en la información del repositorio de gestión de código fuente. Adicionalmente, se detallan aspectos de la historia de *glibc*, incluyendo algunas métricas de tamaño y actividad. Curiosamente, algunas de las leyes de la evolución del software resultaron no ser válidas en determinados casos, como por ejemplo la ley VIII que se refiere a la calidad decreciente del software, donde al menos se necesita que el software se adapte y evolucione rigurosamente para tener en cuenta los cambios en el entorno operativo.

Similar al trabajo anterior, Macho *et. al* [82] se enfocaron en la evaluación de FLOSS a través de su evolución como software. Como caso de estudio, los autores eligieron a *Moodle*<sup>6</sup>, un sistema de gestión de aprendizaje libre y *open source* muy popular en la comunidad educativa internacional. En contraste con otros modelos de

<sup>5</sup><http://www.gnu.org/software/libc/libc.html>

<sup>6</sup><https://moodle.org/>

evaluación que propuestos anteriormente a este trabajo, el que se presenta está basado en la recuperación de información histórica disponible públicamente de Internet. Dicha información sirvió para estudiar en detalle la evolución del software. Como resultado, se brinda una metodología que contribuye, entre otras cosas, a la toma de decisiones en organizaciones que invierten en FLOSS.

Un tema diferente, fue abordado por Di Ruscio *et. al* [37], quienes se interesaron por detectar fallos en aplicaciones FLOSS. En particular, los autores se centraron en detectar aquellos fallos que no pueden ser identificados durante el proceso de instalación de paquetes en sistemas operativos Linux. Como solución a este problema, se propone un enfoque basado en modelos (model-driven) y herramientas de soporte para detectar las fallas en la configuración del sistema antes de realizar algún proceso de instalación o actualización real.

En [19] Brovelli *et. al*, proponen una arquitectura para la creación de un sistema de información geográfico de participación pública (PPGIS). En dicha arquitectura, basada completamente en FLOSS, los datos provienen directamente de los usuarios que emplearían el sistema.

Una investigación interesante fue llevada a cabo por Teixeira *et. al* [146], la cual tuvo por objetivo explorar cómo ocurre la colaboración entre empresas de software rivales en el ámbito del software *open source*. Para lograrlo, aplicaron un enfoque de métodos múltiples (multiple-method) que combina el análisis cualitativo de datos de archivo con repositorios de minería de software, y el análisis de redes sociales. Entre los resultados obtenidos de la investigación, destaca el hecho de que la competencia por el mismo modelo de ingresos (ej. modelos de negocios en conflicto) no necesariamente afecta la colaboración dentro del ecosistema FLOSS.

El trabajo de Zhou *et. al* [167] estuvo dedicado a medir, comprender y predecir cómo la participación y el entorno de los colaboradores novatos en un sistema de seguimiento de problemas influyen en sus probabilidades de convertirse en un colaborador a largo plazo (LTC). Basados en los datos provistos por los proyectos FLOSS Mozilla<sup>7</sup> y Gnome<sup>8</sup>, así como las evidencias de la literatura, entrevistas y documentos en línea, se diseñaron medidas relacionadas con el nivel de participación y condiciones del entorno de trabajo. Los autores desarrollaron un modelo de regresión logística para explicar y predecir las probabilidades de dicha transición a LTC.

Al analizar las ilustraciones empíricas de dos proyectos FLOSS, Bolici *et. al* [15] desarrollaron un modelo para describir la *coordinación estigmérgica* que está presente en este tipo de proyecto basado en equipos de trabajo. Este modelo tiene en cuenta mecanismo de coordinación implícitos (ej. el software en sí) como explícitos (ej. características del entorno donde será compartido el software). Como resultado, los autores encontraron que ambos sistemas de coordinación juegan un rol (por lo general subestimado) en la gestión de dependencias a medida que surgen.

---

<sup>7</sup><https://www.mozilla.org/>

<sup>8</sup><https://www.gnome.org/>



El estudio de los roles y la cortesía en el desarrollo de FLOSS, fue el objeto de estudio de Wei *et. al* [158]. En esta ocasión, los autores analizaron cómo se relacionan los roles núcleo-periferia con el comportamiento social-relacional en términos de la cortesía. Los datos de dos proyectos FLOSS sugieren que tanto los miembros centrales (núcleo) como los miembros periféricos utilizan estrategias de cortesía positivas, en lugar de las negativas. Además, ambos miembros emplean diferentes estrategias para proteger apariencias positivas durante el desarrollo de cortesía positiva. Algo que los autores han denominado respeto e intimidad, respectivamente.

En la misma línea, Crowston *et. al* [27] estudiaron la relación entre las comunicaciones de los miembros centrales y periféricos, y el éxito de un proyecto FLOSS. El estudio se basó en los datos de 74 proyectos de la Incubadora de la Fundación de Software Apache. Los autores encontraron que el volumen de comunicación (núcleo-periferia) está relacionado con el éxito del proyecto.

En [50], Harzl se cuestiona si un proyecto FLOSS se puede ver beneficiado por la inclusión de Kanban [5], una metodología ágil de desarrollo software. En este sentido, se adoptó un enfoque de investigación-acción (*action research*) para estudiar un proyecto FLOSS híbrido desarrollado por estudiantes. Como resultado, el autor reporta que los participantes se sintieron satisfechos y valoran de positiva la integración entre ambas metodologías (FLOSS y Kanban).

La gestión de lanzamientos en proyectos de desarrollo de software a gran escala requiere una comunicación y coordinación significativas. Se trata de un proceso especialmente complejo para los ecosistemas FLOSS, en los que se coordinan cientos de desarrolladores con sus respectivos proyectos. Para comprender mejor este proceso y sus desafíos, Poo-Caamaño *et. al* [108] analizaron comunicaciones provenientes del ecosistema GNOME, a lo largo de más de dos años y medio, incluyendo las interacciones entre los desarrolladores. Los autores encontraron que tanto el cronograma de lanzamiento, como la influencia (en lugar del control directo) y la diversidad, son los factores principales que impactan positivamente el proceso de lanzamiento en el ecosistema de GNOME.

Un análisis de los productos FLOSS utilizados en la arquitectura web cliente-servidor, fue desarrollado por Singh *et. al* [131]. Los autores proporcionan algunas informaciones interesantes sobre algunos de los FLOSS más populares en este contexto: FireFox (navegador web), Apache (servidor web) y MySQL (RDBMS).

Las fundaciones funcionan como una infraestructura de soporte institucional vital para muchos de los proyectos *open source* más exitosos. Sin embargo, las diferentes funciones desempeñadas por estas entidades de soporte han sido poco abordadas en la literatura. En tal sentido, Lindman *et. al* [78] emplean una base de datos de *Open Hub*<sup>9</sup>. Anteriormente conocido como *Ohloh*, estas entidades apoyan a los proyectos que interactúan con otros proyectos. Este estudio se llevó a cabo utilizando teoría basada en la saturación teórica debido al gran volumen de datos disponibles. Los hallazgos derivaron en una taxonomía para las entidades de apoyo, una categorización

---

<sup>9</sup><https://www.openhub.net/>

de los mecanismos de apoyo, y un conjunto de dinámicas de interacciones entre las diferentes entidades de apoyo del software libre.

Alrabae *et. al* [3] proponen un sistema resistente y eficiente que identifica funciones y paquetes FLOSS en archivos binarios *malware*. El sistema propuesto, al cual denominaron FOSSIL, consta de tres componentes. El primero, extrae las características sintácticas de las funciones considerando las frecuencias de códigos de operación y aplicando una prueba estadística a un modelo oculto de Markov. El segundo componente, aplica un grafo de vecindad dispersa (hash) a las trayectorias aleatorias que se derivan de los grafos de control-flujo. Todo esto con el objetivo de extraer la semántica de las funciones. Por último, el tercer componente aplica la puntuación  $z$  ( $z$ -score) a las instrucciones normalizadas para extraer el comportamiento de las instrucciones en cada función. Estos tres componentes se integran utilizando un modelo de red bayesiana, que sintetiza los resultados para determinar la función FLOSS. FOSSIL fue probado en tres conjuntos de datos, incluyendo proyectos FLOSS reales.

Un estudio relacionado con el ámbito educativo, fue desarrollado por Braught *et. al* [18]. Aquí, los investigadores describen cinco modelos correspondientes a cinco tipos de instituciones diferentes, con el objetivo de propiciar el trabajo en proyectos sostenibles FLOSS en experiencias o cursos finales de universidad. De acuerdo a los autores, el objetivo de la exposición de estos modelos buscaría proveer a los instructores interesados en incluir el desarrollo de proyectos sostenibles de FLOSS en sus clases, con perspectivas y recursos adicionales para ayudar a adaptar este enfoque a las necesidades y objetivos específicos de su institución y estudiantes. Se trata sin dudas de un buen aporte, pues FLOSS es comúnmente abordado en el pre-grado con un enfoque práctico muy bajo.

Más recientemente, Adewumi *et. al* [1] proponen un marco de trabajo, denominado FOSSES, para la evaluación y selección de FLOSS. FOSSES fue validado aplicándolo a un estudio de caso, incluyendo además la opinión de expertos. En concreto, participaron 10 expertos, cuyas opiniones sugieren que el 80% de ellos está dispuesto a adoptar el enfoque propuesto para evaluar la selección o no de FLOSS.

La computación en la nube es un paradigma reciente que también ha sido empleado en el contexto de FLOSS. Sin embargo, debido a la falta de una descripción explícita y formal de la perspectiva de los recursos en las aplicaciones FLOSS existentes, no se puede verificar la corrección de la gestión de recursos en la nube. En tal sentido, Jlassi *et. al* [58] proponen una definición formal para los recursos de FLOSS, para garantizar una asignación correcta de estos en la nube. De esta forma, los autores desarrollaron un modelo de asignación de recursos en la nube, denominado *CRAM4FOSS*, utilizando el método *Event-B*<sup>10</sup>. Dicho modelo está concebido para consolidar formalmente la consistencia de la asignación de recursos en la nube de aplicaciones FLOSS durante la fase de diseño, y para analizar y verificar su corrección de acuerdo con los requisitos del usuario y las potencialidades de los recursos.

A partir de la revisión anterior, se puede ver que:

---

<sup>10</sup><http://www.event-b.org/>

1. Existe una tendencia a investigar sobre FLOSS desde múltiples perspectivas (sociológicas, económica, organizacional, educativa, etc.), lo cual se agrupa en la categoría que hemos denominado comunidades FLOSS.
2. La mayoría de las investigaciones se concentran en el estudio de desarrollo de FLOSS, comunidades FLOSS y aplicaciones (Figura 3.2).
3. Con excepción del trabajo Adewumi *et. al* [1], que es similar a nuestro trabajo de investigación pero que no ofrece una guía para evaluar un FLOSS, en los últimos años muy poco se ha propuesto en torno a diseñar instrumentos efectivos para evaluar la selección o adopción de FLOSS en el ámbito organizacional.

Mientras que el primer aspecto es un claro indicativo de la actualidad con que goza la investigación científica sobre FLOSS, el último aspecto ha sido considerado como motivación principal en las contribuciones de la presente investigación.

### 3.3. Identificación de factores de adopción FLOSS

Con el objetivo de identificar los factores que determinan la adopción FLOSS, hemos realizado una revisión sistemática de la literatura que se presenta en el Capítulo 4, en la que hemos comparado los artículos basados en el contexto, el año y el tipo de revisión. También, el periodo de los trabajos estudiados, el número de trabajos estudiados y el número de factores de adopción que se reportaron. La Tabla 3.3 muestra el resumen de esta comparación.

Marsan *et al.* [85] presentan una revisión de la literatura (LR) que no sigue un método sistemático para investigar los trabajos existentes sobre la adopción FLOSS en las organizaciones sanitarias. El propósito de este estudio es desarrollar un modelo para investigar a profundidad el proceso de toma de decisiones en este tipo de organizaciones. Marsan *et al.* [85] utilizan un modelo conceptual basado en la literatura de sistemas de información sobre el tema de la adopción organizacional de software. Este trabajo se centra específicamente en el ámbito de la atención de la salud, mientras que nuestra revisión es más general. Identifican ocho factores.

Badampudi *et al.* [10] presentan una revisión sistemática de la literatura sobre la toma de decisiones en materia de componentes de software: en la empresa, FLOSS, comercial (COTS), o subcontratado. El objetivo de este estudio es identificar los factores que influyen en la decisión de adopción para seleccionar entre diferentes componentes y soluciones de software. Estos factores se comparan entre los componentes COTS, FLOSS y las soluciones caseras. La decisión de adopción depende de la evaluación de estos factores. Hay que tener en cuenta que este estudio aborda aspectos relacionados con el software privativo y FLOSS, lo que difiere de nuestro estudio de investigación que se centra únicamente en FLOSS. En su estudio identificaron once factores.

Trabajo de revisión	Tipo	Año	Contexto	Periodo	# artículos estudiados	# factores	factores comunes
Marsan <i>et al.</i> [85]	LR	2013	Desarrolla un modelo de investigación para estudiar los antecedentes de las decisiones de adopción FLOSS en las organizaciones de salud médica.	1992-2003	78	8	compatibilidad, soporte, apoyo de la alta dirección, coste de licencia
Badampudi <i>et al.</i> [10]	SLR	2016	Identifica los factores que influyen en la decisión de adopción para seleccionar entre diferentes componentes y soluciones de software.	2004-2015	24	11	compatibilidad, personalización, mantenimiento, soporte, coste total de propiedad, coste de licencia
Ven <i>et al.</i> [154]	LR	2012	Identifica los factores que influyen en la adopción de productos FLOSS que dan soporte a la infraestructura de servidores de una organización.			7	coste total de propiedad, compatibilidad, fiabilidad, personalización, prueba, soporte
Rea <i>et al.</i> [115]	SLR	2020	Identifica factores que influyen en el proceso de toma de decisiones para adoptar FLOSS en organizaciones.	2008-2019	54	22	compatibilidad, fiabilidad, personalización, mantenimiento, prueba, soporte, apoyo de la alta dirección, coste total de propiedad

Tabla 3.3: Comparación de artículos relacionados

Ven *et al.* [154] presentan una revisión de la literatura sin un método sistemático. El objetivo de este trabajo de investigación es identificar los factores que influyen en la decisión de adopción en las organizaciones belgas. Ven *et al.* [154] utilizan un marco TOE a través de un estudio de caso para compararlo con los factores ya estudiados anteriormente en la literatura. Este trabajo se centra específicamente en la infraestructura de servidores, mientras que nuestro estudio es más amplio al no considerar únicamente un tipo de software. Se identifican siete factores.

En nuestro estudio sistemático [115], identificamos fuentes primarias en la literatura que describen los factores que influyen en el proceso de toma de decisiones para adoptar FLOSS en las organizaciones. En este contexto, aplicamos un método sistemático inspirado en las directrices descritas por Petersen *et al.* [102] y Kitchenham [66]. Además, queremos destacar que este estudio de investigación se diferencia de los trabajos de investigación existentes ya que definimos un grupo de factores que cubren todos los aspectos a la hora de evaluar FLOSS en las categorías tecnológica, organizacional y económica. Asimismo, este estudio de investigación puede aplicarse a cualquier tipo de institución pública o privada. Identificamos 22 factores diferentes, mientras que los estudios anteriores identificaron un número menor.

### 3.4. Metodologías existentes para la adopción FLOSS

Hemos identificado artículos científicos que proponen metodologías para abordar parcialmente el problema de la adopción FLOSS. Estos estudios se han desarrollado en los últimos 10 años, por lo que consideramos un dominio de investigación activo.

Saini *et al.* [124] propone un modelo que define una métrica denominada Índice de Adopción FLOSS (IAF) que determina las posibilidades de adopción FLOSS satisfactoria en una organización. La metodología tiene dos niveles de criterios: el primer nivel se refiere a las dimensiones identificadas y el segundo nivel se refiere a los factores. El primer nivel tiene 4 criterios: el criterio 1 evalúa la existencia de políticas sobre el uso de FLOSS, el criterio 2 evalúa la disponibilidad y estado del personal capacitado en FLOSS, el criterio 3 evalúa el grado de utilización de FLOSS con relación a su infraestructura informática y el criterio 4 evalúa el grado de utilización de FLOSS a nivel de aplicación, es decir, compatibilidad con otras aplicaciones.

Rossi *et al.* [118] realiza un estudio más completo que analiza la importancia de factores de adopción FLOSS en el sector público. Este marco de trabajo contempla cuatro dimensiones: tecnológica, organizacional, interna - externa y perspectiva individual. La dimensión interna - externa la definen como competencias y habilidades del personal de soporte técnico dentro de la organización, disponibilidad de soporte externo. La dimensión de perspectiva individual se refiere al análisis del coste por parte de un administrador de TI. Este marco de trabajo mide los factores en 2 fases distintas, al inicio y al final del proceso de adopción. Asimismo, para evaluar la importancia de factores considera la opinión de expertos y principalmente se basa en factores internos. La metodología está compuesta por 4 dimensiones y 21 factores.

Aversano *et al.* [7] propone el uso del marco de trabajo EFFORT en dos versiones, la primera en su versión original y la segunda en una versión personalizada donde se aumentan características para evaluar los criterios. EFFORT permite evaluar la calidad y la funcionalidad de FLOSS específicamente para Sistemas de Planificación de Recursos Empresariales (ERP). En este trabajo se evalúan cinco sistemas FLOSS ERP mediante formulación matemática. A cada métrica definida se le asignó una puntuación en el rango [1-5], donde 1 = inadecuado, 2 = deficiente, 3 = suficiente, 4 = bueno y 5 = excelente.

Macho *et al.* [82] propone evaluar la adopción FLOSS mediante el análisis de la información histórica disponible en repositorios públicos de código fuente. Los autores diseñaron una metodología que evalúa el código fuente basándose en el marco de trabajo OpenBRR, pues consideran que determina la calidad del software. OpenBRR utiliza cuatro fases: la primera fase *filtro de evaluación rápida*, la segunda fase *evaluación del uso de los objetivos*, la tercera fase *recolección y procesamiento de datos* y la última fase *traducción de datos*. Para evaluar las métricas utiliza una puntuación en el rango [1-5], donde 1 = inaceptable, 2 = deficiente, 3 = aceptable, 4 = muy bueno y 5 = excelente. Adicionalmente para complementar esta evaluación, se considera la importancia de la literatura para evaluar la importancia de los factores en cuestión. Este estudio se centra en el software Moodle, uno de los sistemas de aprendizaje en línea FLOSS más utilizados en la actualidad.

Adewumi *et al.* [1] propone FOSSES, un marco de trabajo para la evaluación FLOSS. FOSSES define una metodología para facilitar el proceso de adopción mediante la evaluación de factores tecnológicos y económicos. Para validar FOSSES, se utilizaron tres técnicas de evaluación con diferentes enfoques a cinco FLOSS: POX, Ryu, Trema, FloodLight y OpenDaylight. Las técnicas utilizadas fueron: AHP (proceso de jerarquía analítica); ER (razonamiento probatorio) y WSM (modelo de puntuación ponderada). Como complemento a este proceso de adopción, se considera como dato de entrada también a la importancia de la opinión de expertos para evaluar la importancia de los factores. Seis expertos realizaron la evaluación asignándole valores numéricos en el rango de [1-5], donde 1 = deficiente, 2 = regular, 3 = medio, 4 = bueno y 5 = excelente. Los resultados obtenidos permitieron mejorar los criterios de evaluación de FOSSES.

En nuestra propuesta GUIOS, extendemos el marco de las contribuciones existentes con una guía que asista al decisor sobre la idoneidad en la adopción de un FLOSS bajo un enfoque FODA, donde se evalúan factores internos y externos de la organización. De esta forma, se busca que la evaluación obtenida se exprese en un lenguaje más cercano a la toma de decisión en las organizaciones. Para la construcción de GUIOS nos basamos en la identificación de dimensiones, factores y subfactores a través de un proceso sistemático de la literatura y la opinión de expertos para evaluar la importancia de los factores, con el objetivo de que el proceso de adopción FLOSS sea ágil y sencillo. Adicionalmente, para una mejor evaluación de las características del FLOSS a adoptar, se desarrolló una herramienta prototipo basada en software libre. En la Tabla 3.4 se muestra un resumen de las características de estos trabajos relacionados basado en ocho criterios.

Crterios	Saini <i>et al.</i> [124]	Rossi <i>et al.</i> [118]	Aversano <i>et al.</i> [7]	Macho <i>et al.</i> [82]	Adewumi <i>et al.</i> [1]	GUIOS
Propone un primer nivel de criterios: Dimensiones	X	X	X	X	X	X
Propone un segundo nivel de criterios: Factores	X	X	X	X	X	X
Propone un tercer nivel de criterios: Subfactores	X	X	X	X	X	X
Ofrece una herramienta prototipo para evaluar FLOSS				X		X
Se hace uso de la guía para evaluar adopción de FLOSS en una entidad		X				X
Considera la importancia de la literatura para evaluar la importancia de los factores				X		X
Considera la importancia de la opinión de expertos para evaluar la importancia de los factores		X			X	X
Evalúa factores internos de la entidad adoptante		X				X

Tabla 3.4: Características de artículos relacionados con GUIOS

### 3.5. Resumen

Nuestro objetivo en este capítulo fue describir la motivación del por qué nosotros decidimos desarrollar esta tesis. Al término del presente capítulo, se puede concluir que:

- Hemos analizado trabajos de investigación que tienen relación con factores de adopción y pudimos identificar que ninguna de las propuestas anteriores encontradas en la literatura logra abordar todos los problemas a la vez.
- Hemos analizado trabajos de investigación que abordan sobre metodologías existentes sobre adopción FLOSS pero ninguna de las propuestas anteriores encontradas en la literatura logran cubrir todos los problemas identificados.
- En los siguientes capítulos, encontrarán propuestas específicas a los problemas identificados en este trabajo de tesis. Concretamente, la solución fue resolver las preguntas de investigación planteadas y que tienen relación con identificar factores de adopción, así como, diseñar una guía que asista a los responsables a evaluar el FLOSS en una organización.

**Parte III**

**Contribuciones**



## Capítulo 4

# Factores de adopción

En este capítulo presentamos una revisión sistemática de la literatura, donde identificamos y clasificamos factores a través de distintas perspectivas como factores tecnológicos, organizacionales y económicos. Nuestro objetivo principal en este capítulo es presentar la clasificación de factores que será la base para construir una guía para evaluar FLOSS en instituciones públicas y privadas<sup>1</sup>.

La estructura del capítulo es la siguiente: en la Sección 4.1, presentamos brevemente una introducción acerca del proceso de adopción FLOSS. En la Sección 4.2, mostramos el proceso sistemático definido para la selección de estudios primarios en la adopción FLOSS. En la Sección 4.3, clasificamos los factores por cada dimensión. Finalmente, en la Sección 4.5 resumimos el capítulo.

---

<sup>1</sup>Parte de este capítulo se ha publicado en la revista IEEE Access. Tenemos el acuerdo de todos los autores para incluir el texto correspondiente como parte de este manuscrito.

## 4.1. Introducción

El software libre o de código abierto (FLOSS) se está volviendo omnipresente. La capacidad de lograr un mayor grado de independencia de los proveedores, la interoperabilidad y las posibles reducciones de costos están motivando a las organizaciones a considerar la adopción FLOSS [16, 41, 52]. En una encuesta anual<sup>2</sup> se destaca que el 90 por ciento de los encuestados ya han adoptado FLOSS para sus empresas de alguna manera. Según esta encuesta, las adopciones se están realizando rápidamente sin ningún proceso formal y con una gestión eficaz. Los problemas en la adopción FLOSS pueden ocasionar un mal precedente en la organización, que podría aumentar la resistencia de los miembros de la organización para migrar a más FLOSS, o incluso llevar a un retroceso al software propietario en algunos casos.

La identificación de los factores que influyen en la adopción FLOSS es un paso inicial fundamental para definir directrices. Ello permite a los administradores y expertos en tecnología de la información evaluar los riesgos potenciales y asignar los recursos necesarios para mitigar los posibles inconvenientes. Es necesario evaluar convenientemente factores como la compatibilidad con formatos de datos existentes en la organización, la facilidad de uso de FLOSS, la disponibilidad de documentación, el apoyo externo, el mantenimiento y la capacitación, entre muchos otros. Este problema no es nuevo y ya ha sido cubierto, desde diferentes perspectivas, en una cantidad significativa de contribuciones de revisión de literatura en las últimas dos décadas [10, 51, 85] y, de acuerdo con los resultados que proporcionamos en este trabajo de tesis, lo consideramos un dominio activo de investigación (ver Sección 4.3).

Definimos un conjunto de dimensiones y factores que generalmente cubren aspectos relevantes para una evaluación de un FLOSS. Esta memoria de tesis pretende ser una base importante para la construcción de una guía para la adopción FLOSS en instituciones públicas y privadas.

En este capítulo se revisa la literatura de fuentes primarias a través de procedimientos de revisión sistemática para identificar y clasificar los factores que la literatura destaca como relevantes en el proceso de adopción FLOSS. Seguimos un proceso de revisión sistemática inspirado en las directrices de Petersen *et al.* [102] y Kitchenham [66] para extraer la literatura relevante sobre la adopción FLOSS. Filtramos más de 2.000 documentos de los últimos catorce años para obtener finalmente 54 estudios primarios de los que recopilamos datos y análisis. Como resultado, descubrimos veintidós factores diferentes que influyen en la toma de decisiones sobre la adopción FLOSS. Nueve factores tecnológicos, otros nueve organizacionales y cuatro económicos. Además, también definimos y describimos estos factores y los clasificamos por su relevancia de acuerdo como aparecen en la revisión de la literatura. Proporcionamos una serie de escenarios de adopción realistas para ser utilizados como ejemplos y que se describen en la literatura seleccionada. Además, esta tesis proporciona consejos para los investigadores que se interesan por este problema en

---

<sup>2</sup><http://nbvp.northbridge.com/2016-future-open-source-survey-results>

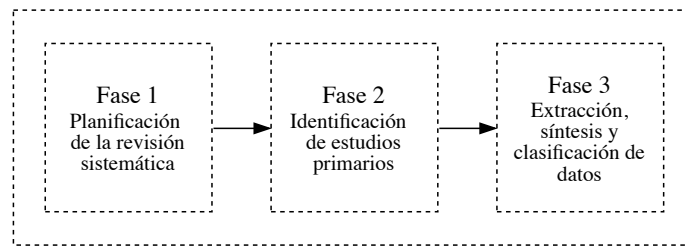


Figura 4.1: Definición proceso sistemático [91, 102].

caso de que quieran dirigir sus contribuciones a conferencias y publicaciones que ya han cubierto este tema.

## 4.2. Definición del proceso sistemático

El proceso sistemático que hemos seguido para este estudio se basa en las directrices propuestas por Petersen *et al.* [102] y Kitchenham [66]. La Figura 4.1 resume las diferentes fases que hemos seguido:

1. Planificar la revisión, cuya principal salida es el protocolo de búsqueda que afecta a todas las demás fases.
2. Identificación de estudios primarios, donde se seleccionan los estudios según el protocolo de búsqueda definido.
3. Extracción y clasificación de datos, donde se extraen los datos de los estudios y se clasifican los trabajos.

A continuación, describimos con más detalle cada una de las fases.

### 4.2.1. Fase 1: planificación de la revisión sistemática

Esta fase está conformada por dos procesos como se muestra en la Figura 4.2:

1. **Definición del protocolo:** donde decidimos cómo hacer la revisión de la literatura.
2. **Definición de las preguntas de investigación:** donde definimos las preguntas que queremos responder y enfocamos nuestro alcance de investigación. Definimos las siguientes preguntas de investigación:
  - **RQ1.** ¿Qué factores influyen en la adopción FLOSS?
  - **RQ2.** ¿Cuál es el alcance de la investigación sobre los factores de adopción FLOSS?

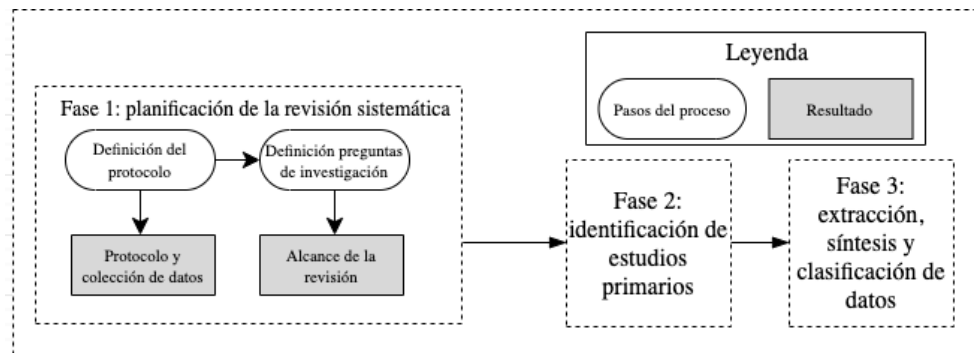


Figura 4.2: Primera fase del estudio [91, 102]

- **RQ2.1** ¿Qué tipo de trabajos de investigación están cubriendo la adopción FLOSS?
- **RQ2.2** ¿Cuántas publicaciones relacionadas con FLOSS se publicaron en los años cubiertos por esta revisión?

En RQ1, queremos identificar los factores que influyen en la adopción FLOSS en las organizaciones de acuerdo con lo que informa la literatura. En la RQ2, con esta pregunta general, pretendemos recopilar conocimientos sobre el alcance de la investigación en la adopción FLOSS y más específicamente en las subpreguntas:

- RQ2.1, con esta pregunta, buscamos el tipo de estudios relacionados con la adopción FLOSS. Asimismo, pretendemos detectar las brechas de investigación que permitan a los investigadores identificar los inconvenientes de la investigación dedicada a FLOSS.
- RQ2.2, con esta pregunta se intenta determinar cómo han evolucionado los trabajos de investigación según los tipos de publicación. Queremos identificar el estado actual y las tendencias de un tema para encontrar oportunidades de mayor colaboración en el área de adopción FLOSS.

#### 4.2.2. Fase 2: identificación de estudios primarios

Esta fase implicó la ejecución de cuatro procesos como se muestra en la Figura 4.3:

1. Realizando la búsqueda, donde realizamos una búsqueda en varias bases de datos que son relevantes para nuestro tema de investigación.
2. Estudios filtrados, que consiste en aplicar los criterios de inclusión y exclusión para filtrar lo que no consideramos relevante, y recoger lo que sí consideramos útil para nuestros objetivos.

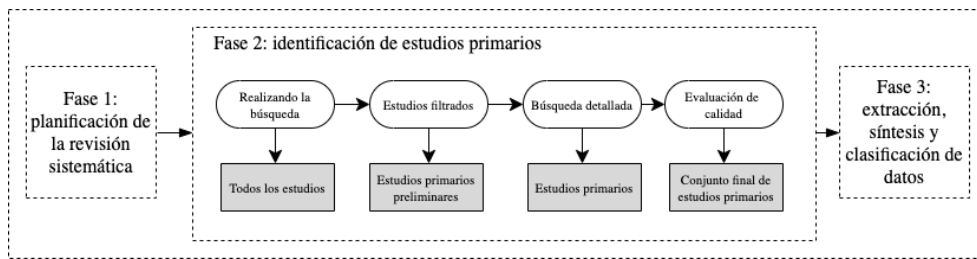


Figura 4.3: Segunda fase del estudio [91, 102]

3. Búsqueda detallada, donde se añadieron manualmente nuevos estudios para ampliar la muestra de estudios primarios.
4. Evaluación de calidad, donde realizamos una evaluación de calidad del conjunto final de artículos seleccionados para asegurarnos de que son relevantes para esta investigación.

A continuación, explicamos cómo identificamos los estudios incluidos en el proceso sistemático.

- **Realizando la búsqueda.** Para este primer proceso diseñamos consultas de búsqueda específicas. Específicamente, nos basamos en palabras clave relevantes y algunos de los sinónimos más comunes. La búsqueda se realizó en cuatro bases de datos con una gran cobertura en el campo de la Ingeniería informática [103]. Estas bases de datos son: ACM Digital Library (Expanded ACM Guide to Computing Literature), IEEE Xplore Digital Library, Scopus y Web of Science.

Los resultados obtenidos en cada base de datos se presentan en la Tabla 4.1, donde se puede ver que se obtuvieron 4.429 trabajos. También se observa que las consultas de búsqueda se formularon con el fin de lograr todos los resultados pertinentes relacionados con la adopción (incluidos sus diferentes términos) y FLOSS (incluidos sus sinónimos más comunes).

- **Estudios filtrados.** Para filtrar los diferentes artículos que obtuvimos anteriormente, aplicamos un conjunto de criterios de inclusión y exclusión que se describen a continuación.

*Criterios de inclusión:*

- Documentos publicados entre enero de 2008 y diciembre de 2019. La razón de esta selección es nuestra intención de actualizar la revisión llevada a cabo en [51], que cubría los trabajos hasta 2008.
- Revisiones por pares de artículos en revistas. Con este filtro, aseguramos que nuestros resultados provienen de fuentes de alta calidad.
- Documentos publicados y escritos en inglés.

Database	Query	Search results
IEEE	(adopt*) AND (“free software“ OR “libre software“ OR “os software“ OR “open source software“ OR floss OR foss OR oss)	516
ACM	((+adopt*) AND (“free software“ OR “libre software“ OR “free libre“ OR “os software“ OR “open source software“ OR floss OR foss OR oss))	519
Scopus	TITLE-ABS-KEY ((adopt*) AND (“free software“ OR “libre software“ OR “os software“ OR “open source software“ OR floss OR foss OR oss))	2.248
WOS	TS= ((adopt*) AND (“free software“ OR “libre software“ OR “os software“ OR “open source software“ OR floss OR foss OR oss))	1.146
<b>Total</b>		<b>4.429</b>

Tabla 4.1: Resultados de la búsqueda realizada en las bases de datos.

- Los documentos están relacionados con la adopción FLOSS. Durante el proceso de estudios filtrados, seleccionamos los trabajos candidatos cuyo título y resumen se refieren a la adopción FLOSS. También llevamos a cabo una serie de reuniones para validar que los documentos seleccionados son pertinentes al tema.

#### *Criterios de exclusión*

- Los documentos publicados en conferencias o como capítulos de un libro.
- Los documentos que, si bien se relacionan con la adopción FLOSS, no describen explícitamente los factores de adopción.
- Documentos duplicados

El número de estudios incluidos y excluidos en cada etapa se muestra en la Figura 4.4. Allí se puede ver que 2.742 documentos resultaron después de eliminar los duplicados. Además, 2.259 estudios fueron excluidos después de aplicar los criterios de inclusión y exclusión. Los 483 restantes se obtuvieron a través de una lectura rápida de títulos y resúmenes y de una lectura de texto completo, lo que permitió identificar 51 trabajos relevantes. En las diferentes etapas descritas anteriormente, los autores se dividieron en grupos para comprobar la selección y, de ser necesario, debatir y llegar a un consenso.

- **Búsqueda detallada.** Para evitar que se pierdan documentos relevantes, procedimos a realizar una búsqueda utilizando la técnica de muestreo de bola de nieve, como se sugiere en [96, 103].

Por supuesto, limitamos nuestra búsqueda al mismo tipo de documento (artículos de revistas) y período de tiempo (2008-2019). Utilizamos ambas: estrategias de lectura de bola de nieve hacia adelante y hacia atrás. En este sentido,

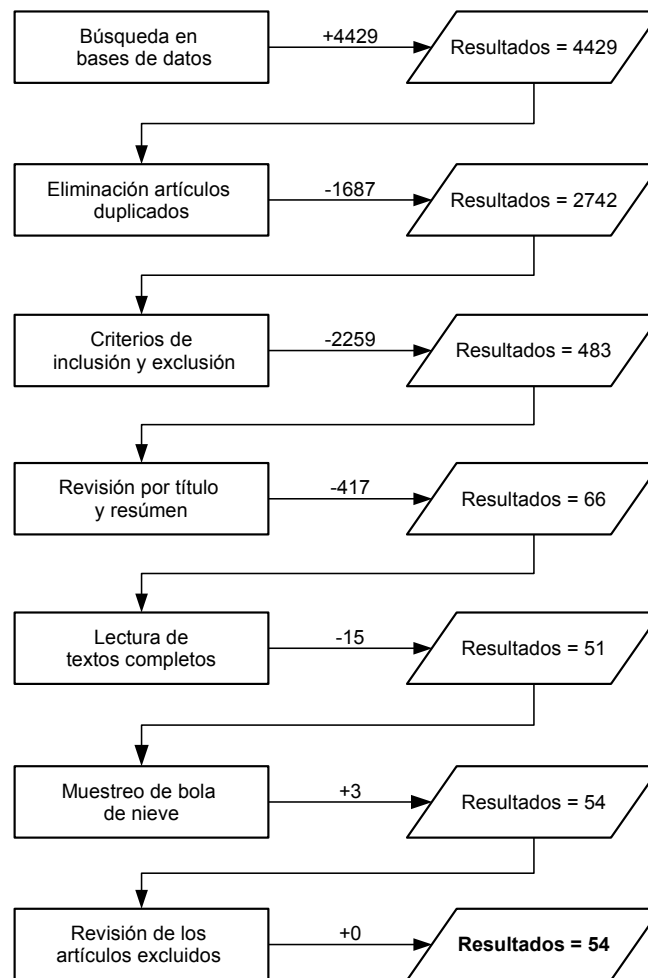


Figura 4.4: Proceso de búsqueda de documentos [91, 102]

encontramos 3 documentos adicionales que añadimos a los 51 documentos obtenidos en el paso anterior. Así, se consideraron 54 documentos para llevar a cabo la revisión. Este proceso está representado por la Figura 4.4.

- Evaluación de la calidad.** Además de los requisitos de calidad considerados en los pasos anteriores (por ejemplo, artículos de bases de datos respetables y publicados como artículos de revistas), evaluamos la calidad de las 54 fuentes primarias encontradas considerando sólo los artículos indexados en el ranking de la revista Scimago <sup>3</sup>. Luego, para cada artículo seleccionado, dimos una puntuación de 1 a 5 (1 siendo el menos relevante y 5 el más relevante) de acuerdo a nuestro criterio y experiencia seleccionando sólo aquellos que ob-

<sup>3</sup><http://scimagojr.com/>

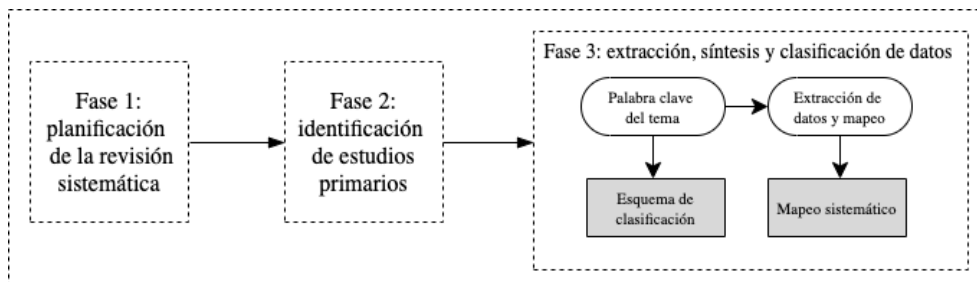


Figura 4.5: Tercera fase del estudio [91, 102]

tuvieron una puntuación de 4 a 5. Evaluamos que los 54 estudios primarios seleccionados pasaron esta evaluación de calidad.

### 4.2.3. Fase 3: extracción, síntesis y clasificación de datos

Esta fase implica la ejecución de dos procesos como se muestra en la Figura 4.5. En esta fase, extraemos los datos necesarios para determinar las conclusiones del conjunto de documentos seleccionados.

- **Palabra clave del tema:** Clasificamos los documentos usando dos dimensiones:
  - Los factores.
  - El tipo de investigación.

Estas dimensiones se basan en el proceso descrito por Petersen *et al.* [102] que se basa en un proceso de palabras claves para definir el alcance de una investigación sistemática. Este proceso de palabras clave consta de dos partes. Primero, leemos el resumen para revisar e identificar las palabras claves que pueden contribuir a nuestro estudio. Si el resumen no tiene información relevante, es posible que tengamos que leer otras partes del artículo que creemos que son necesarias para la comprensión de la información que se va a recopilar. Con esta información, definimos un conjunto de categorías para llevar a cabo un proceso sistemático que identifique las principales contribuciones de nuestro estudio de investigación. Como resultado de este proceso de palabras claves, definimos las dimensiones de los factores de adopción FLOSS:

- **Dimensión tecnológica:** estos son los factores que se relacionan con las características técnicas de los productos que se consideran para su adopción
- **Dimensión organizacional:** se refieren a los factores relacionados con los aspectos organizacionales y los recursos humanos de la institución que adoptan un producto.



- **Dimensión económica:** se refiere a los factores relacionados con el aspecto económico que se utilizan para adquirir, vender o alquilar productos y servicios que permitan satisfacer las necesidades de las organizaciones.

Para definir la dimensión tipo de investigación, usamos la propuesta de Wieringa *et al.* [160] y Petersen *et al.* [102]. Más específicamente:

- **Documentos de experiencia:** describen la experiencia que se realizó en la práctica (qué y cómo). Se trata de la propia experiencia del autor sobre un tema determinado en situaciones reales. Documentos que informan sobre escenarios reales de adopción FLOSS.
  - **Investigación de validación:** este tipo de investigación evalúa técnicas con usuarios finales que no han desplegado FLOSS en ambientes de producción.
  - **Investigación de evaluación:** este trabajo cubre una evaluación de un despliegue de FLOSS existente. El resultado del estudio ayudará a definir conclusiones positivas o negativas.
  - **Propuesta de solución:** este tipo de trabajo de investigación propone una solución al problema de la adopción FLOSS siguiendo una metodología nueva o ya existente. Esto puede aplicarse a estudios de casos sin necesidad de ser validado o evaluado.
  - **Documentos filosóficos:** este tipo de trabajo nos ayuda a definir el área de nuestro estudio a través de una clasificación de dimensiones en el proceso de adopción FLOSS.
  - **Documentos de opinión:** describen opiniones personales sobre las técnicas utilizadas en un proceso de adopción FLOSS. No se basan en metodologías de investigación.
- **Extracción de datos y mapeo:** se compone de los siguientes pasos:
1. Creamos una base de datos con todos los artículos a clasificar. Decidimos utilizar una base de datos bibtex a través de un gestor de referencias bibliográficas JabRef<sup>4</sup>. La base de datos contiene los siguientes campos para cada artículo:
    - Título del artículo.
    - Dimensión de los factores.
    - Tipo de investigación.
    - Los factores por cada dimensión.

---

<sup>4</sup><http://www.jabref.org/>

2. A través de reuniones coordinadas de los investigadores de este estudio, analizamos las categorías a utilizar y creamos una taxonomía de factores, agrupándolos según un tema.
3. Realizamos una síntesis de cada factor donde describimos un concepto general y ejemplos que encontramos en la literatura.
4. Dos de los autores de este estudio identificaron los trabajos por cada dimensión individualmente y los introdujeron en una matriz de hoja de cálculo.
5. Un tercer autor revisó el trabajo realizado en el paso anterior y, en una reunión en la que participaron todos los autores de este estudio, llegamos a un consenso sobre los criterios para definir la taxonomía propuesta.

#### 4.2.4. Amenazas a la validez

Aunque en este documento hemos seguido un mecanismo sistemático, hay algunos supuestos que hemos hecho que pueden afectar a su validez [162]:

**Validación externa:** incluimos documentos que fueron citados en revistas de diferentes bases de datos, como se muestra en la Tabla 4.1, podríamos haber pasado por alto algunas contribuciones relevantes no publicadas en esos lugares. Asimismo, la decisión de no incluir documentos publicados en conferencias podría haber dejado fuera conocimientos relevantes sobre la adopción FLOSS. Minimizamos el impacto de esta amenaza realizando la técnica de lectura de bola de nieve. Consideramos que la amenaza más importante para la validez externa son:

- Validez de la población. A través de las bases de datos seleccionadas para este trabajo de investigación, obtuvimos 54 documentos que están relacionados con la adopción FLOSS. Estos documentos se refieren a los factores que hemos definido en este trabajo de investigación. Por lo tanto, al buscar en estas bases de datos, no identificamos que contengan diferentes factores relacionados con la adopción FLOSS. Asumimos que los factores que no seleccionamos en nuestra investigación son irrelevantes para la adopción FLOSS según la literatura. Creamos manualmente esta clasificación de factores leyendo nuestro conjunto de documentos seleccionados. Para la generación de los gráficos, los generamos semiautomáticamente extrayendo datos de nuestra base de datos local, también seleccionamos las claves primarias para cada dimensión.
- Validez ecológica. Se centra en los posibles errores en los materiales y herramientas utilizados en el experimento. El proceso manual de clasificación de los factores podría generar desconfianza entre los investigadores de la comunidad. Para evitar esta percepción, todos los autores de este estudio realizaron individualmente la clasificación de los factores y luego, agregamos los resultados para construir una clasificación general.

**Validación interna:** es una medida que asegura que el diseño del experimento del investigador sigue el principio de causa y efecto. En este estudio de revisión de la literatura, realizamos un proceso sistemáticamente estructurado. Sin embargo, la agrupación de factores y un proceso de clasificación manual de los mismos puede dar lugar a errores. Una vez más, consideramos pertinente que el proceso de clasificación manual fuera revisado individualmente por cada uno de los autores de este estudio para minimizar el impacto.

### 4.3. Factores de adopción FLOSS

En esta sección, abordamos la primera pregunta de investigación (RQ1) que definimos en la Sección 4.2.1:

*RQ1: ¿Qué factores influyen en la adopción FLOSS?* Los factores que influyen en la adopción FLOSS se clasifican en grupos de factores tecnológicos, organizacionales y económicos. Clasificamos los artículos en grupos sobre la base de estos factores en las secciones 4.3.1, 4.3.2 y 4.3.3.

#### 4.3.1. Factores tecnológicos

Estos son los factores que se relacionan con las características técnicas de los productos que se consideran para su adopción. Encontramos que este es un factor típico que se aborda comúnmente en la literatura.

La Tabla 4.2 muestra un resumen de los trabajos dentro de esta categoría. Concretamente, observamos 49 trabajos relacionados que representan alrededor del 91 % del total de los trabajos examinados, de los cuales hay una gran cantidad que hacen referencia al factor de compatibilidad, fiabilidad, usabilidad, personalización y documentación. Nos dimos cuenta de que en la mayoría de los documentos no se mencionaban los factores relacionados con la prueba del software, la reutilización y la portabilidad. Es importante resaltar que el número total que muestra la Tabla 4.2 como resultado final, no es la suma de la columna subtotal de la tabla, sino el número total de artículos en el que los factores de la dimensión tecnológica son mencionados. A continuación, describimos en detalle los diferentes factores tecnológicos encontrados en nuestro estudio.

- **Compatibilidad<sup>5</sup>:** este factor se refiere a la compatibilidad del software a ser adoptado con respecto a los formatos de datos. Encontramos 34 documentos que se refieren a este factor. Este es el factor más referenciado en el grupo tecnológico. Por ejemplo, al adoptar *OpenOffice*, podríamos encontrar problemas de compatibilidad con los formatos propietarios existentes en la organización. Rossi *et al.* [119], informaron algunos problemas de compatibilidad al importar archivos en formatos propietarios.

---

<sup>5</sup>Sólo consideramos la compatibilidad del formato de los datos en este factor. Clasificamos la compatibilidad del sistema operativo (y cualquier otra dependencia del software) dentro del factor de portabilidad que se refiere a la compatibilidad de la plataforma.

- **Fiabilidad:** este factor se refiere a la forma en que se utiliza un software al ser adoptado en algunas condiciones durante un período de tiempo determinado. La baja fiabilidad crea incomodidad en los usuarios y complica las operaciones en la organización, en particular, en las primeras etapas de la adopción FLOSS. Por ejemplo, Linux es un sistema operativo FLOSS muy popular y exitoso que se ha adoptado ampliamente porque podría considerarse un sistema operativo fiable y bien probado que los usuarios han utilizado con normalidad.
- **Usabilidad:** este factor se refiere a cuán intuitiva y satisfactoria es la interfaz de usuario del software. Por ejemplo, cuando se despliega un sistema que registra los análisis de sangre de los pacientes en un centro de salud, es importante que el sistema que se adopte sea fácil de usar para garantizar el éxito de su adopción. Una crítica común a las soluciones de software libre es que no consideran la usabilidad como un primer elemento al diseñar un software y, aunque puede ser bien diseñada e implantada desde el punto de vista tecnológico, la experiencia del usuario se deja de lado. Sin embargo, podemos encontrar ejemplos como el software de escritorio Gnome que afirma que la usabilidad juega un papel crítico [51].
- **Personalización:** este factor se refiere al grado de personalización que permite el software mediante cambios en su configuración por defecto. Por lo tanto, se hace más fácil cumplir con los nuevos requisitos de personalización. Por ejemplo, la plataforma web FLOSS WordPress<sup>6</sup> ofrece una amplia gama de opciones de configuración que permiten desde cambiar los colores hasta la base de datos utilizada. Esto puede facilitar la adopción FLOSS.
- **Documentación:** este factor se refiere a la cantidad, disponibilidad y calidad de la documentación del FLOSS, como la documentación de usuario, administrador y desarrollador. La escasa e incompleta documentación dificulta la adopción de nuevo software. Por ejemplo, la suite ofimática OpenOffice <sup>7</sup> ofrece documentación básica y avanzada en línea para los usuarios. La falta de documentación aumenta el grado de dependencia de los proveedores de TI externos, lo que puede suponer costes adicionales.
- **Mantenimiento:** este factor se refiere a la cantidad de recursos que hay que dedicar para mantener el FLOSS. Esto incluye la fácil instalación de actualizaciones de software para arreglar errores, para soportar nuevas características y, por lo tanto, para mejorar la usabilidad. Un FLOSS bien mantenido por un proveedor de TI externo es conveniente. Por ejemplo, un sistema de tickets para reportar incidentes para resolver problemas y para ayudar a los usuarios es también una opción deseable. El mantenimiento también se refiere a las actualizaciones de seguridad que pueden comprometer la información sensible del usuario.

---

<sup>6</sup><https://www.wordpress.com/>

<sup>7</sup><https://www.openoffice.org/support/books.html>

- **Prueba:** este factor se refiere a lo fácil que es probar el software a adoptar. La facilidad de instalación y despliegue del software permite a los técnicos evaluar los factores de adopción detallados en este documento. Más específicamente, este factor es clave para evaluar factores como la usabilidad, la personalización, la fiabilidad, el mantenimiento o la compatibilidad entre otros. Por ejemplo, cuando se realiza una adopción FLOSS para un software de oficina, es conveniente tener una versión descargable que sea fácil de instalar y probar (por ejemplo, *OpenOffice*). Esto permite la evaluación preliminar del software para tener una opinión de primera mano sobre el mismo.
- **Reusabilidad:** este factor se refiere a la capacidad del FLOSS para ser reutilizado por diferentes organizaciones o unidades. Uno de los principios del FLOSS es que el código puede ser copiado y modificado. Encontramos ocho documentos que se refieren a este factor. Conjeturamos que esto se debe a que es una propiedad intrínseca de FLOSS. Por ejemplo, si un entorno estadístico programable quiere ser adoptado en las escuelas públicas, utilizando un FLOSS permitirá reutilizar cualquier software que se desarrolle en ese entorno sin pagar ninguna cuota de licencia.
- **Portabilidad:** este factor se refiere a la posibilidad de desplegar un software en más de una plataforma, como por ejemplo en diferentes sistemas operativos. Encontramos seis documentos que se refieren a este factor y por lo tanto se convierte en el factor menos citado de los documentos revisados dentro de la dimensión tecnológica. Un FLOSS portable podría ser una opción deseable para promover la independencia de la plataforma fundamental. Por ejemplo, OpenOffice puede ser fácilmente instalado en diferentes sistemas operativos siendo un ejemplo bien conocido de un FLOSS portable.

#### 4.3.2. Factores organizacionales

Estos son los factores que se relacionan con los aspectos organizacionales y de recursos humanos de la institución que adopta un producto. Encontramos en nuestro estudio que este es un factor típico que aparece en la literatura. En este grupo, resumido en la Tabla 4.3 y descrito a continuación, observamos 50 trabajos relacionados que representan alrededor del 93 % del total de los trabajos revisados. El soporte es el factor más referenciado. Esto podría indicar que las empresas necesitan tener garantías para resolver problemas para no perder tiempo en sus actividades. También encontramos factores como la formación, el apoyo de la alta dirección, el bloqueo de los proveedores o la actitud hacia el cambio entre otros. También observamos que la mayoría de las investigaciones no hacen referencia a ciertos factores como: casos de estudio de adopción FLOSS, adopción de tiempo, centralidad de la tecnología de la información y reingeniería de procesos de negocios. Asimismo, el número total que muestra la Tabla 4.3 como resultado final, no es la suma de la columna subtotal de la

<b>Factores</b>	<b>Descripción</b>	<b>Referencia</b>	<b>Subtotal</b>
Compatibilidad	La compatibilidad del software que se adopte con respecto a los formatos de los datos	[6] [10] [14] [16] [17] [30] [33] [41] [52] [59] [85] [95] [119] [130] [142] [143] [73] [40] [107] [64] [29] [134] [8] [154] [111] [61] [83] [88] [90] [147] [56] [49] [35] [145]	34
Fiabilidad	El número de errores de programación que un usuario encuentra al usar un software de computadora	[6] [14] [17] [30] [33] [41] [52] [104] [123] [142] [155] [29] [64] [153] [134] [154] [61] [83] [90] [147] [125] [129] [106]	23
Usabilidad	Lo intuitiva y satisfactoria que es la interfaz de usuario del software	[6] [14] [17] [33] [52] [59] [123] [29] [107] [153] [111] [61] [125] [122] [21] [106] [145]	17
Personalización	El grado de personalización que permite el software mediante cambios en su configuración por defecto	[6] [10] [14] [17] [30] [33] [52] [155] [154] [111] [48] [61] [83] [122] [129] [68] [106]	17
Documentación	La cantidad y calidad de la documentación disponible del software a adoptar	[6] [33] [52] [59] [104] [142] [112] [111] [61] [125] [49] [68]	12
Mantenimiento	La cantidad de recursos que hay que dedicar para mantener el FLOSS	[6] [10] [17] [30] [33] [104] [142] [134] [48] [61] [125] [129]	12
Prueba	La facilidad para probar el software a adoptar	[17] [52] [81] [39] [153] [154] [61] [90] [125]	9
Reusabilidad	Cantidad de código que puede ser reutilizado a través de bibliotecas y paquetes	[16][33] [52] [98] [134] [61] [125] [21]	8
Portabilidad	La posibilidad de desplegar un software en más de una plataforma, como diferentes sistemas operativos	[6] [33] [134] [83] [122] [68]	6
<b>Total</b>			<b>49</b>

Tabla 4.2: Clasificación de artículos basados en factores tecnológicos.

tabla, sino el número de artículos en el que los factores de la dimensión organizacional son mencionados. A continuación, describimos en detalle los diferentes factores organizacionales encontrados en nuestro estudio.

- **Soporte:** este factor se refiere a la disponibilidad de soporte técnico interno y externo. Ese soporte proporciona experiencia, conocimientos y aptitudes en el FLOSS que se adopte para ayudar y contribuir a resolver los problemas dentro de una organización. La falta de soporte técnico de los expertos en tecnología de la información puede poner en peligro la adopción FLOSS en una organización según Marsan *et al.* [85]. Encontramos 45 documentos que se refieren a este factor, siendo el factor más referenciado dentro de los documentos que seleccionamos en nuestro estudio y en ese sentido siendo el más importante según la literatura que revisamos.
- **Formación:** este factor se refiere a las acciones de capacitación para mejorar las habilidades y los conocimientos de los usuarios dentro de la organización. La capacitación ayuda a los usuarios a resolver problemas técnicos por sí mismos. También ayuda a dar soporte a otros usuarios que forman parte de la organización. Según Rossi *et al.* [119], las entrevistas con los usuarios y los gerentes confirmaron que la capacitación es esencial para aumentar la aceptación de la tecnología. La falta de capacitación suficiente suele estar relacionada con un presupuesto insuficiente. También es importante identificar a los usuarios capacitados que pueden entrenar a los menos capacitados
- **Apoyo de la alta dirección:** este factor se refiere al grado de respaldo de la dirección a la decisión de adoptar FLOSS. Según Rossi *et al.* [119], la literatura revisada describe dos escenarios de adopción similares, uno en el que la dirección apoya la decisión de adopción y otro en el que no. La falta de respaldo de la dirección provoca incertidumbre entre los usuarios. Sin el respaldo adecuado de la gestión, los usuarios encuentran difícil expresar su resistencia al cambio según Ramdani *et al.* [114].
- **Bloqueo del vendedor:** esto se refiere a las dependencias de un software específico de una manera que hace que sea costoso cambiar a una alternativa. Los proveedores de software deciden cuando los productos de software entran en el fin del ciclo de vida útil. A partir de ese momento, el software deja de ser mantenido, por lo que no hay más actualizaciones disponibles. Esto es especialmente problemático en términos de seguridad, ya que las vulnerabilidades se acumularán esperando a ser explotadas por los atacantes a lo largo del tiempo, exponiendo el software a violaciones de seguridad. La facilidad de actualización y las fechas conocidas en el fin del ciclo de vida útil del software son fundamentales para estimar los costos de actualización. La diversidad de proveedores es conveniente para tener una elección de proveedores más amplia que permita a las organizaciones ser más independientes del proveedor de tecnología de la información.

- **Actitud hacia el cambio:** este factor se refiere a la forma en que los empleados se comportan cuando se enfrentan a los cambios tecnológicos en la organización. Es conveniente evaluar la predisposición de los miembros de la organización cuando se va a adoptar un nuevo software. En seis artículos (véase la Tabla 4.3) se sugiere que las organizaciones deben considerar incentivos para reducir la resistencia al cambio cuando planifiquen la adopción FLOSS. Por ejemplo, cuando se actualiza a una nueva plataforma, como la basada en el sistema operativo Linux, se puede ofrecer nuevo equipo a los usuarios si aceptan migrar sus antiguos escritorios a Linux.
- **Casos exitosos de adopción FLOSS:** este factor se refiere a las historias exitosas existentes de la adopción FLOSS por organizaciones similares. Muchos proveedores de tecnología de la información muestran su cartera de clientes como una estrategia de marketing para atraer a más clientes nuevos para que adopten su software. También documentan historias de éxito en relación con el despliegue satisfactorio de productos y la mejora de las operaciones. Esto incluye citas de clientes que describen el resultado de la adopción FLOSS<sup>8</sup>. Las historias de éxito ayudan a las organizaciones a considerar la adopción FLOSS en el proceso de toma de decisiones. Por ejemplo, Blind Audio Tactile Mapping System (BATS<sup>9</sup>) proporcionan acceso a los mapas para los ciegos y las personas con discapacidad visual. Esta historia de éxito del lenguaje de programación Python, puede fomentar que otros usuarios potenciales de un lenguaje de programación adopten este software de lenguaje de programación FLOSS.
- **Tiempo para adoptar:** este factor se refiere al tiempo que una organización requiere para desplegar FLOSS. Este factor depende del nivel de capacitación y las aptitudes de los miembros de la organización, la disponibilidad de un proveedor de tecnología de la información externo que pueda ayudar en el proceso de adopción, y la resistencia del usuario al cambio. Los plazos más largos de adopción pueden dar lugar a costos adicionales en el proceso de adopción. La compatibilidad entre los programas informáticos más antiguos y los nuevos es también una propiedad deseada.
- **Centralidad de TI:** este factor se refiere al grado de dependencia de la organización respecto de su propia infraestructura de tecnología de la información. Si la organización depende en gran medida de los programas informáticos que tiene previsto sustituir, la eficacia de la organización puede verse afectada. Por ejemplo, los problemas en el despliegue de un software de VoIP/PBX basada en FLOSS pueden tener un impacto negativo en las operaciones comerciales en caso de que las telecomunicaciones sean cruciales para las operaciones diarias de la organización. Por lo tanto, el cambio al nuevo software conlleva un alto riesgo.

---

<sup>8</sup><https://www.python.org/about/success/>

<sup>9</sup><https://www.python.org/success-stories/bats/>



- **Reingeniería de procesos empresariales:** este factor puede aparecer cuando una organización está cambiando sus procesos comerciales internos debido a alguna circunstancia particular (por ejemplo, la mejora de la calidad, la reestructuración de la organización). Encontramos sólo un documento que se refiere a este factor, siendo el menos citado dentro de los artículos que seleccionamos en este grupo de factores. Por ejemplo, si se realiza una mejora de la calidad de los procesos empresariales dentro de una organización, la dirección de la tecnología de la información puede aprovechar la oportunidad para integrar nuevas tecnologías, de modo que FLOSS pueda utilizarse para diferentes tareas (por ejemplo, generar documentos en estándares abiertos). Los usuarios se verían obligados a utilizar el nuevo software para manejar estos documentos, ganando así confianza en la nueva tecnología [119].

### 4.3.3. Factores económicos

Son los factores relacionados con el aspecto económico que se utilizan para adquirir, vender o alquilar productos y servicios que permitan satisfacer las necesidades de las organizaciones. En la Tabla 4.4 se resumen los artículos de este grupo. Observamos 32 artículos relacionados que representan alrededor del 60% del total de los trabajos revisados. Es importante resaltar que el número total que muestra la Tabla 4.4 como resultado final, no es la suma de la columna subtotal de la tabla, sino el número total de artículos en el que los factores de la dimensión económica son mencionados. A continuación, describimos en detalle los diferentes factores económicos encontrados en nuestro estudio.

- **Costo total de propiedad (CTP):** este factor, según la literatura, se refiere a los costos de licencia, operación y soporte entre otros. Es un factor general que puede interpretarse de manera diferente según la organización. Los expertos en tecnología de la información pueden tener en cuenta únicamente los costos de las licencias, pero también son pertinentes otros costos asociados, por ejemplo, los de despliegue y soporte. Hemos clasificado aquí los artículos cuando se referían explícitamente al CTP sin ningún otro detalle. Encontramos 19 artículos que se refieren a este factor. Por ejemplo, Ven *et al.* [154] muestra que en algunas organizaciones consideraron la posibilidad de migrar a un FLOSS debido a los bajos costos de hardware y software. Cinco organizaciones que anteriormente utilizaban Unix declararon que el uso de Linux dio como resultado una reducción significativa de los costos de hardware debido a que Linux es compatible con el hardware de Intel, que es relativamente más barato que el hardware de Unix.
- **Costos de licencia:** este factor se refiere al costo de adquisición de una licencia del software. En el trabajo de investigación que propone Marsan *et al.* [85], el Gerente de TI de un centro de salud ha observado que los costos de licencia del software privativo para un laboratorio clínico se

Factores	Descripción	Referencia	Subtotal
Soporte	La disponibilidad de soporte técnico interno y externo	[6] [10] [14] [16] [17] [30] [33] [41] [52] [74] [81] [85] [94] [95] [120] [123] [130] [73] [40] [155] [112] [29] [39] [107] [153] [134] [8] [109] [154] [111] [48] [61] [83] [90] [147] [56] [125] [49] [122] [129] [68] [21] [35] [106] [145]	45
Formación	Medidas para mejorar las aptitudes y los conocimientos de los usuarios de la organización	[16] [17] [33] [59] [95] [104] [119] [130] [73] [40] [112] [29] [39] [134] [8] [111] [88] [90] [147] [56] [125] [49] [129] [68] [106]	25
Bloqueo de vendedores	Dependencia de un software específico de una manera que hace que sea costoso cambiar a una alternativa	[17] [33] [52] [94] [40] [155] [134] [154] [111] [48] [88] [49] [106]	13
Apoyo de la alta dirección	Grado de apoyo de la dirección a la decisión de adoptar FLOSS	[17] [33] [52] [85] [94] [119] [143] [73] [40]	10
Actitud hacia el cambio	Comportamiento de los empleados en relación con los cambios tecnológicos que se adopten	[17] [59] [119] [123] [40] [154]	6
Estudios de caso de adopción FLOSS	Historias exitosas existentes de la adopción FLOSS por organizaciones similares	[16] [17]	2
Tiempo para adoptar	El tiempo que una organización requiere para desplegar FLOSS	[6] [14]	2
Centralidad de TI	El grado de dependencia de la organización de su propia infraestructura informática	[17] [119]	3
Reingeniería de procesos empresariales	Un cambio en el software motivado por una mejora en la calidad del servicio	[119]	1
<b>Total</b>			<b>50</b>

Tabla 4.3: Clasificación de los documentos basada en factores organizacionales.

Factores	Descripción	Referencia	Subtotal
Costo total de la propiedad (CTP)	Los costos de licencia, de operación y de soporte	[6] [10] [14] [30] [33] [52] [130] [73] [165] [64] [134] [154] [111] [147] [56] [122] [129] [68] [21]	19
Costos de licencia	El costo de adquirir una licencia del software	[10] [16] [17] [81] [85] [95] [104] [155] [29] [107] [153] [134] [8] [48] [83] [56]	16
Costos operacionales	El costo del mantenimiento, el desarrollo, el despliegue y la migración	[94] [107] [83] [90]	4
Costo de soporte	Soporte externo y acceso a actualizaciones	[40] [147]	2
<b>Total</b>			<b>32</b>

Tabla 4.4: Clasificación de los artículos basada en factores económicos.

volvieron más caros a lo largo del tiempo, mientras que los costos de FLOSS se mantuvieron mucho más económicos.

- **Costos Operacionales:** este factor indica el costo de la migración, el mantenimiento y el despliegue. Por ejemplo, Maglogiannis *et al.* [83] promueve el FLOSS como una solución para la informatización de las instituciones de salud pública. El modelo FLOSS permite compartir los costos de desarrollo y mantenimiento entre instituciones que tienen recursos de financiación limitados y objetivos similares.
- **Costos de soporte:** este factor se refiere al soporte externo y al acceso a las actualizaciones. Por lo tanto, es conveniente hacer una encuesta en el mercado para las empresas que pueden proporcionar apoyo a FLOSS. Sólo hemos encontrado dos estudios que se refieren a este factor. Por ejemplo, en caso de adoptar la suite OpenOffice, hay una lista pública de consultores que están disponibles para proporcionar soporte externo <sup>10</sup>.

#### 4.4. Alcance de la investigación en la adopción FLOSS

En esta sección, abordamos la segunda pregunta de investigación (RQ2) definida previamente en la Sección 4.2.1:

*RQ2: ¿Cuál es el alcance de la investigación en factores de adopción en soluciones FLOSS?*

Con esta pregunta, pretendemos dar pautas a los investigadores de FLOSS para que exploren nuevas líneas de investigación. Dividimos esta pregunta en tres preguntas más específicas:

<sup>10</sup><https://www.openoffice.org/bizdev/consultants.html/>

- RQ2.1: ¿Qué tipo de trabajos de investigación abarcan los factores de adopción FLOSS? (ver Sección 4.4.1)
- RQ2.2: ¿Cuántas publicaciones relacionadas con FLOSS se han publicado en los años que cubren este estudio? (véase la Sección 4.4.2)

#### 4.4.1. ¿Qué tipo de trabajos de investigación abarcan los factores de adopción FLOSS?

Esta pregunta de investigación tiene como objetivo encontrar las áreas que carecen de contribuciones de investigación relacionadas con la adopción FLOSS en las organizaciones. La Figura 4.6, presenta un mapa de calor en el que hemos clasificado el número de artículos que tomamos basados en las dos dimensiones que definimos en la Sección 4.2.3.

En primer lugar, en cuanto a los tipos de investigación, cabe destacar que el alto número de artículos de evaluación y el aumento de las contribuciones de validación reflejan la madurez de los estudios de adopción FLOSS. Sin embargo, esta área de investigación aún no ha llegado al punto de aportar informes de experiencia. En segundo lugar, en lo que respecta a los factores de adopción FLOSS, observamos que la mayoría de los estudios se han centrado en los factores organizacionales y tecnológicos, dejando los factores económicos sin cubrir. Suponemos que esta falta de resultados de investigación en los factores económicos se debe a la desconfianza de las empresas en proporcionar detalles económicos. Además, los expertos en FLOSS consideran que las organizaciones ya son conscientes de los costes ocultos al adoptar FLOSS y, por tanto, tienden a centrarse más en la investigación de los factores tecnológicos y organizacionales. Además, sólo encontramos dos propuestas de solución relacionadas con factores económicos y una con factores tecnológicos y organizacionales. También observamos que la investigación de validación, los artículos de opinión y los artículos filosóficos están ganando madurez en el área de la adopción FLOSS porque encontramos taxonomías, revisiones de la literatura y mapas sistemáticos.

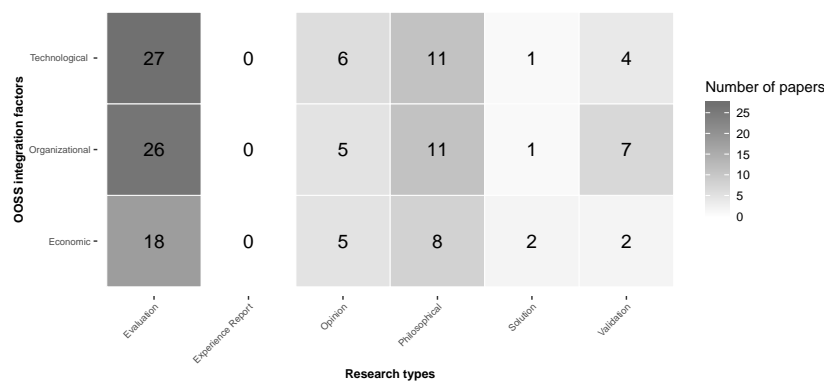


Figura 4.6: Relación entre el grupo de factores y los tipos de investigación.

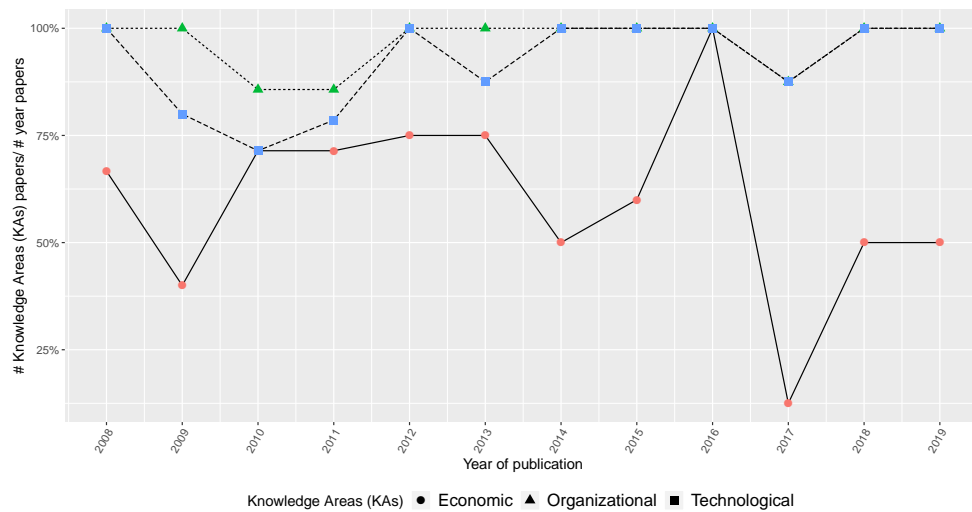


Figura 4.7: Evolución temporal de las diferentes tendencias por dimensión y año de publicación.

#### 4.4.2. ¿Cuántas publicaciones relacionadas con FLOSS se publicaron en los años que cubren esta revisión?

En esta pregunta, presentamos la evolución de diferentes tendencias en función del año de publicación (véase la Figura 4.7). El eje y muestra el porcentaje de trabajos por los factores, así, extraemos los intereses de los investigadores por año. Observamos que hay áreas que mantienen una atención estable a lo largo del tiempo como son los factores organizacionales y los tecnológicos. Sin embargo, el factor económico ha experimentado una mayor variación en los últimos años.

En este contexto, queremos destacar que el factor organizacional es el que tiene el mayor número de trabajos por factores con un total de 50 según la Tabla 4.3. Creemos que los investigadores están más interesados en identificar los factores organizacionales que pueden influir en las adopciones FLOSS, ya que estos factores están relacionados con los miembros de la organización que participan en el proceso de toma de decisiones de adopción FLOSS, como los directivos, los expertos en TI y los usuarios finales, entre muchos otros.

Asimismo, observamos que el factor tecnológico y los factores organizacionales son igualmente importantes, ya que encontramos un total de 49 artículos según la Tabla 4.2. Sin embargo, hay una excepción en 2013, donde el factor organizativo aparentemente comenzó a atraer más interés entre los investigadores. Por último, observamos que el interés por los factores económicos ha seguido siendo variable. En 2009, 2014 y 2017 observamos que el porcentaje de trabajos de investigación ha disminuido considerablemente. Suponemos que esto se debe a que las organizaciones

están aprendiendo sobre los costes ocultos en la adopción FLOSS, por lo que puede ser interesante para los investigadores explorar los campos económicos además de seguir centrándose en los factores tecnológicos y organizacionales.

### 4.5. Resumen

En la Tabla 4.5, describimos los factores que identificamos en cada artículo durante nuestro estudio sistemático. En particular, el grupo de factores organizacionales se consideran los más relevantes, ya que el 93 % de los documentos seleccionados se refieren a ellos. Asimismo, los factores tecnológicos también son muy relevantes ya que se describen en el 91 % de los artículos seleccionados. Por último, los factores económicos son los menos referidos en el 60 % de los estudios.

La Figura 4.8, muestra a nivel general el orden de los factores de manera descendente según la información que descubrimos en este estudio sistemático.

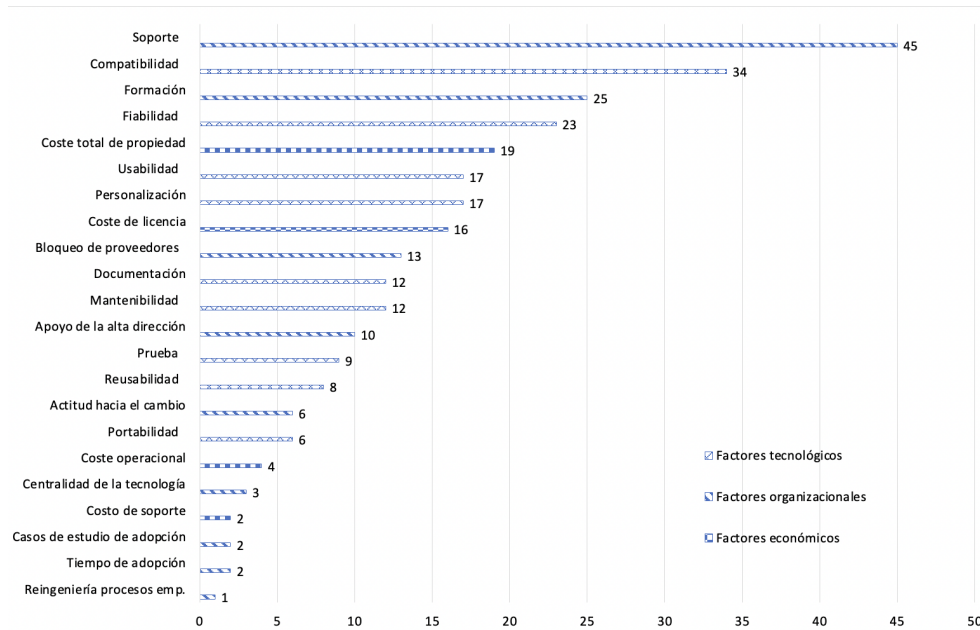


Figura 4.8: Orden de factores [115].

Artículos	Compatibilidad	Fiabilidad	Usabilidad	Personalización	Documentación	Mantenimiento	Reusabilidad	Prueba	Portabilidad	Soporte	Formación	Apoyo alta dirección	Bloqueo vendedor	Actitud	Casos exitosos FLOSS	Tiempo adoptar	Centralidad TI	Reingeniería proc. emp.	Total cost ownership	Costos licencia	Costos operacionales	Costos soporte
Aversano <i>et al.</i> [6]	X	X	X	X	X	X			X	X						X			X			
Ayala <i>et al.</i> [8]	X									X	X									X		
Badampudi <i>et al.</i> [10]	X			X	X					X									X	X		
Benlian <i>et al.</i> [14]	X	X	X	X						X						X			X			
Bouras <i>et al.</i> [16]	X					X				X	X	X			X					X		
Bouras <i>et al.</i> [17]	X	X	X	X		X		X		X	X	X	X	X	X		X			X		
Choi <i>et al.</i> [21]			X				X			X									X			
Daneshgar <i>et al.</i> [30]	X	X		X	X					X									X			
Del Bianco <i>et al.</i> [33]	X	X	X	X	X	X	X		X	X	X	X							X			
Dhir <i>et al.</i> [35]	X									X												
D Macredie <i>et al.</i> [29]	X	X	X							X	X									X		
Fitzgerald <i>et al.</i> [39]								X		X	X	X										
Gallego <i>et al.</i> [40]	X									X	X	X	X	X								X
Gangadharan <i>et al.</i> [41]	X	X								X												
Gurusamy <i>et al.</i> [48]				X	X					X		X								X		
Hanumappa <i>et al.</i> [49]	X				X					X	X	X										
Hauge <i>et al.</i> [52]	X	X	X	X	X		X	X		X		X	X						X			
Idrizi <i>et al.</i> [56]	X									X	X								X	X		
Joia <i>et al.</i> [59]	X		X		X							X		X								
Jusoh <i>et al.</i> [61]	X	X	X	X	X	X	X	X		X												
Kimppa <i>et al.</i> [64]	X	X																	X			
Koloniari <i>et al.</i> [68]				X	X				X	X	X								X			
Li <i>et al.</i> [73]	X									X	X	X							X			
Li <i>et al.</i> [74]										X												
Lundell <i>et al.</i> [81]								X		X										X	X	
Maglogiannis <i>et al.</i> [83]	X	X		X					X	X										X	X	
Marsan <i>et al.</i> [85]	X									X		X								X		
Meetoo-Appavoo <i>et al.</i> [88]	X										X		X									
Mijinyawa <i>et al.</i> [90]	X	X						X		X	X										X	
Munoz <i>et al.</i> [94]										X		X	X								X	
Nagy <i>et al.</i> [95]	X									X	X									X		
Paschali <i>et al.</i> [98]							X															
Petrinja <i>et al.</i> [104]		X			X	X					X									X		
Poba <i>et al.</i> [106]		X	X	X						X	X		X									
Ponelis <i>et al.</i> [107]	X		X							X										X	X	
Qu <i>et al.</i> [109]										X												
Rafiq <i>et al.</i> [112]					X					X	X											
Rafiq <i>et al.</i> [111]	X		X	X	X					X	X		X						X			
Rossi <i>et al.</i> [119]	X										X	X		X			X	X				
Roumani <i>et al.</i> [120]										X												
Safadi <i>et al.</i> [122]			X	X					X	X									X			
Saghafi <i>et al.</i> [123]		X	X							X			X									
Sarrab <i>et al.</i> [125]		X	X		X	X	X	X		X	X											
Silic <i>et al.</i> [129]		X		X	X					X	X								X			
Silic <i>et al.</i> [130]	X									X	X								X			
Spagnoletti <i>et al.</i> [134]	X	X				X	X		X	X	X		X						X	X		
Stol <i>et al.</i> [143]	X											X										
Stol <i>et al.</i> [142]	X	X			X	X																
Taha <i>et al.</i> [145]	X		X							X												
Tome <i>et al.</i> [147]	X	X								X	X								X			X
Ven <i>et al.</i> [155]		X		X						X		X								X		
Ven <i>et al.</i> [153]		X	X					X		X										X		
Ven <i>et al.</i> [154]	X	X		X				X		X		X	X						X			
Zaffar <i>et al.</i> [165]																			X			
Total	34	23	17	17	12	12	8	9	6	45	25	10	13	6	2	2	3	1	19	16	4	2

Tabla 4.5: Resumen de factores identificados por artículo.

## **Capítulo 5**

# **Hacia la elaboración de una guía para la adopción FLOSS**

En este capítulo presentamos el proceso para la construcción de GUIOS, una guía para la adopción FLOSS en organizaciones públicas y privadas. GUIOS se basa en factores identificados a través de un proceso de revisión sistemático de la literatura. Estos factores son ponderados por expertos FLOSS a través de una encuesta diseñada con preguntas de selección múltiple. Asimismo, para la encuesta se definieron escalas de relevancia para mejor evaluación por parte de los expertos.

Nuestro objetivo principal en este capítulo es definir los pasos que permitieron elaborar una guía de trabajo para evaluar la adopción FLOSS.

La estructura del capítulo es la siguiente: la Sección 5.1 ofrece una introducción donde se justifica la necesidad de sistematizar el proceso de adopción FLOSS. En la Sección 5.2, describe el proceso para la elaboración de la guía. Finalmente, en la Sección 5.3 resumimos el capítulo.



## 5.1. Introducción

La adopción FLOSS es un proceso que depende de la identificación de factores tecnológicos, organizacionales y económicos [115], tal y como hemos visto en el Capítulo 4. La identificación de factores relevantes nos permite avanzar en la construcción de una guía que sistematice el proceso de adopción FLOSS. Si bien los factores identificados ayudan a caracterizar el proceso de adopción, estos no proporcionan una ruta bien definida que permita valorar los requisitos mínimos para una adopción satisfactoria. Por tanto vemos necesario identificar factores más relevantes para la organización, y una opción en que dicha relevancia sea ponderada con la relevancia de la literatura existente. Un proceso que sistematice la adopción FLOSS debe de ser lo suficientemente flexible para personalizar la relevancia de los factores según la necesidad de su organización. Este problema ha sido abordado parcialmente en la literatura existente [1, 7, 118, 124, 148] que nos sirve como punto de partida para nuestra investigación tal y como hemos visto en el Capítulo 3.

Una de las contribuciones de nuestra propuesta consiste en la elaboración de una encuesta para expertos FLOSS. Dicha encuesta permite conocer la valoración de los factores desde el punto de vista del experto. Esta valoración sirve para ponderar la importancia de los factores obtenidos a partir de la literatura existente. Para el diseño de la encuesta se definió una escala de relevancia para evaluar los factores y sus características en el ámbito de la adopción FLOSS, de esta manera se pudieron obtener los resultados con mayor precisión. Estas escalas son utilizadas tanto para la encuesta dirigida a expertos FLOSS como para los responsables de evaluar el FLOSS.

A continuación se describen los pasos seguidos para la elaboración de la guía propuesta, denominada GUIOS. En primer lugar, se describe el proceso empleado para la elaboración de GUIOS, esto es, determinar las dimensiones, los factores y la caracterización que se utilizarán finalmente en el proceso de adopción. A continuación, la elaboración de una encuesta dirigida a expertos que ayude a valorar la caracterización de los factores y subfactores, así como determinar un mecanismo para que el responsable de la evaluación FLOSS pondere los factores.

## 5.2. Proceso de elaboración de la guía para adopción FLOSS

Hemos definido un proceso *ad-hoc* que describe los pasos para elaborar una guía que evalúe la adopción FLOSS en una organización. A continuación, se describen los pasos a seguir y que se resumen en la Figura 5.1:

1. Definición de dimensiones, factores y subfactores que determinan la adopción. Para ello, nos basamos en dos fuentes principales: a) revisión de artículos científicos extraídos de la revisión presentada en el Capítulo 4, b) revisión de artículos científicos [2, 9, 34, 42, 69, 79, 104, 113, 127, 133, 140, 149, 150, 152, 156] de congresos a través de la técnica de muestreo bola de nieve [96, 103].

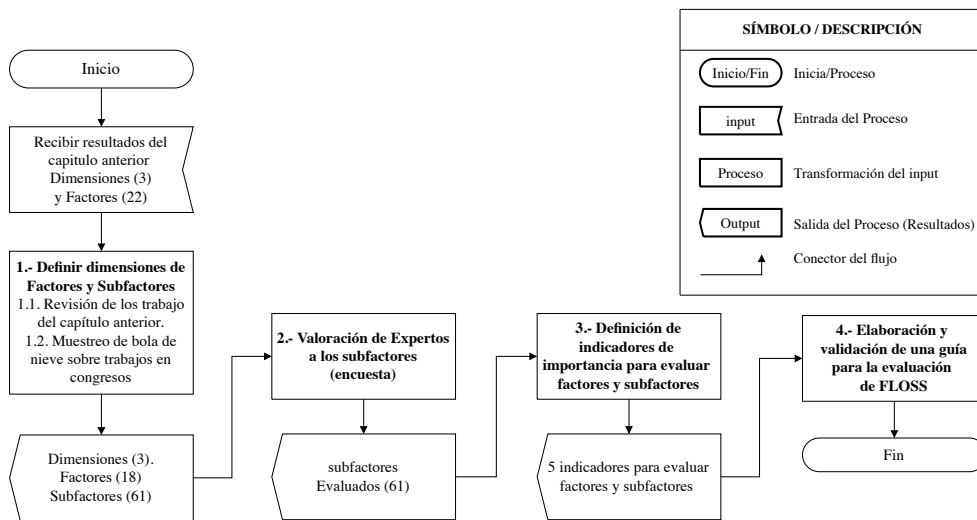


Figura 5.1: Proceso para la elaboración de GUIOS.

2. Elaboración de una encuesta para valorar la importancia de los subfactores por parte de los expertos. El resultado a obtener con este paso es: la importancia de los factores y subfactores basados en diferentes fuentes: literatura existente, expertos y el responsable de evaluar el FLOSS.
3. Definición de indicadores para evaluar la importancia de los factores y subfactores. Entendemos por escala de relevancia la importancia que el responsable de la adopción FLOSS asocia a los factores y subfactores.

A continuación, se detallan los tres primeros pasos del proceso para el desarrollo de GUIOS. El cuarto paso, dedicado a la elaboración y validación de GUIOS, será detallado en el Capítulo 6.

### 5.2.1. Dimensiones, factores y subfactores

En este paso identificamos las dimensiones, factores y subfactores para nuestra guía. Un subfactor es un aspecto concreto del FLOSS a adoptar que forma parte de los factores identificados. Por definición, un subfactor está contenido en un factor, y a su vez un factor pertenece a una cierta dimensión. El subfactor ofrece una definición más específica de un aspecto relevante del FLOSS que debe ser tomado en cuenta cuando se plantea la adopción. Si bien un importante número de dimensiones y factores fueron identificados en este trabajo de investigación, en este paso nos planteamos la tarea de revisar nuevamente estos resultados con el objetivo de verificar si existen o no nuevas dimensiones y factores producto de una revisión de trabajos publicados en congresos como extensión de la literatura ya revisada previamente (ver Capítulo 4). Los artículos de congreso fueron excluidos de la búsqueda en una primera instancia debido a la gran cantidad de artículos obtenidos como resultado de esa búsqueda y

por tal motivo, lo consideramos pertinente incluir el criterio de artículos publicados en congresos a través de la técnica de muestreo *bola de nieve* [96, 103] tanto hacia delante como hacia atrás. Como resultado de este proceso, se obtuvieron 15 trabajos [2, 9, 34, 42, 69, 79, 104, 113, 127, 133, 140, 149, 150, 152, 156], que fueron filtrados con algunos de los criterios de inclusión aplicados a los artículos obtenidos en el capítulo anterior (ver Sección 4.2.2), esto es, artículos publicados entre 2008 y 2019, artículos escritos en inglés, y por tener al menos un factor relacionado con los factores ya identificados en el capítulo anterior. Este resultado obtenido no aportó nuevas dimensiones o factores.

Se definieron un total de 3 dimensiones y 22 factores. No obstante, es importante mencionar que en este capítulo hemos excluido un total de 4 factores al no identificar al menos un subfactor que se refiera a dicho factor. De esta manera hemos incluido solo 18 de los 22 factores identificados anteriormente. Concretamente, hemos descartado los siguientes factores:

- *Reingeniería de proceso empresariales*: no encontramos información al detalle relacionada a este factor, y en consenso con un grupo de expertos consideramos excluirlo del grupo de factores.
- *Coste de licencia*: lo consideramos como una característica dentro del factor coste total de propiedad (CTP), y en consenso con un grupo de expertos consideramos excluirlo del grupo de factores.
- *Coste operacional*: lo consideramos como una característica dentro del factor CTP, y en consenso con un grupo de expertos consideramos excluirlo del grupo de factores.
- *Coste de soporte*: lo consideramos como una característica dentro del factor CTP, y en consenso con un grupo de expertos consideramos excluirlo del grupo de factores.

Como resultado final de una revisión en profundidad de los artículos que mencionan a estos factores, hemos identificado un total de 61 subfactores. Para identificar a los subfactores, hemos realizado una búsqueda en el texto de cada artículo sobre características relevantes por cada factor, con el objetivo de que los factores sean desglosados y permitan una evaluación FLOSS más precisa. Cada subfactor se redactó en una frase positiva, corta y sencilla con el objetivo de que el subfactor sea claro y fácil de entender para el evaluador, como recomiendan Kitchenham y Pfleeger [67].

En resumen, los 61 subfactores los hemos agrupado en 3 dimensiones y 18 factores, tal y como se lista a continuación en la Tabla 5.1:

Tabla 5.1: Factores y subfactores del marco de trabajo propuesto para la adopción FLOSS

Dimensión	Factor	Descripción	Subfactor/citas
Tecnológica	Compatibilidad	Compatibilidad de la solución a ser adoptada con respecto a los formatos de los datos.	<p>Una empresa proporciona una infraestructura de nube lista para usar para este software [41].</p> <p>El software puede exportar formatos propietarios. [6][8][29][147].</p> <p>El software interactúa y se integra con el software propietario existente [10, 42, 56, 149][40][49][73][61][83][90].</p> <p>El software está certificado para operar en su nicho de mercado [85].</p> <p>El software es compatible con los casos de uso y las funcionalidades más comunes [33].</p> <p>El software es compatible con múltiples componentes de hardware [29][42][83].</p> <p>El software usa formatos estándar [17, 29, 33].</p> <p>El software es compatible con varios sistemas operativos diferentes [29][41].</p>
	Personalización	El grado de personalización permitido por el software a través de cambios en su configuración predeterminada	<p>El acceso al código fuente es un incentivo para la organización [10, 16, 150][83][106][48][129].</p> <p>El software se puede ampliar fácilmente para satisfacer mis necesidades mejorando el código fuente [6, 14, 17, 69, 133, 150] [52][61][83][48][30][68][111].</p> <p>Las innovaciones se introducen en el software a un ritmo rápido [52][61][68].</p> <p>El software es fácil de personalizar sin necesidad de actualizar el código fuente [14, 149].</p> <p>El software soporta nuevas funciones a través de módulos [6, 17][48][68].</p> <p>Hay un repositorio público de extensiones para este software [6].</p>

**Tabla 5.1 (continuación de la página anterior)**

Dimensión	Factor	Descripción	Subfactor
	Fiabilidad	Este factor se refiere a la forma en que se utiliza una solución que debe adoptar en las condiciones durante un período de tiempo determinado. Se refiere al comportamiento que demuestra el sistema, si es confiable.	<p>El software tiene un buen historial en cuanto a errores de seguridad. El software es seguro) [34, 61, 69, 149][125].</p> <p>El software es fiable y estable [14, 17, 34, 59, 123][90].</p> <p>El software es más flexible que la solución propietaria [69][61][125][134].</p> <p>El software es más confiable que la solución propietaria [61, 69, 147, 149][125][134].</p> <p>El software proporciona una amplia variedad de funciones de control de acceso [41, 64][125][134].</p>
	Reutilización	Cantidad de código que se puede reutilizar a través de bibliotecas y paquetes	<p>La licencia permite extensiones propietarias [6].</p> <p>El software se ofrece como una biblioteca / marco de trabajo [98][125][134].</p>
	Usabilidad	Cuán intuitiva y satisfactoria es la interfaz de usuario del software	<p>El software proporciona una interfaz gráfica de usuario (GUI) [29][61][133][125].</p> <p>El software es más fácil de usar que la alternativa propietaria [2, 14, 17, 59, 61, 123][125].</p> <p>El software es fácil de aprender [17, 61].</p> <p>El usuario está descontento con el software propietario [14].</p>

**Tabla 5.1 (continuación de la página anterior)**

Dimensión	Factor	Descripción	Subfactor
	Documentación	La cantidad y calidad de la documentación disponible de la solución que se adoptará.	<p>La documentación de desarrollo cubre todas las características [9][125].</p> <p>El software está bien documentado [9], [142].</p> <p>La documentación está disponible en múltiples formatos [104][125].</p> <p>La documentación es fácil de entender [142][125].</p> <p>La documentación está actualizada [140][125].</p> <p>La documentación está escrita por escritores especializados [104].</p> <p>La documentación del software es de alta calidad [104][111].</p> <p>El software viene con documentación de desarrollo [6][61][68][112][125].</p> <p>El software viene con documentación del usuario [6][61][68].</p> <p>La documentación del usuario cubre todas las características [9]. Los formatos de datos están bien documentados [16].</p>
	Mantenimiento	Cantidad de recursos que deben dedicarse para mantener la solución FLOSS.	El software es mantenido activamente por los desarrolladores [17][48][30][134].
	Portabilidad	La posibilidad de implementar una solución en más de una plataforma, como diferentes sistemas operativos	<p>Una versión de aplicación móvil de este software está disponible [6][83].</p> <p>El software es una sistemas de administración de base de datos independiente [6][83].</p>
	Prueba	La facilidad para probar la solución para adoptar.	El software es fácil de instalar y de probar [52][61][90][125].

**Tabla 5.1 (continuación de la página anterior)**

Dimensión	Factor	Descripción	Subfactor
Organizacional	Soporte	La disponibilidad de soporte técnico interno y externo.	<p>El soporte de la comunidad para este software está disponible [6, 8, 9, 10, 29, 41, 49, 56, 61, 94, 95, 113, 120, 123, 150, 152, 156][73][83][48][68][111][112][125][79].</p> <p>Soporte de expertos y consultores externos para consultas específicas está disponible[74, 79], [85, 147][48][125].</p> <p>El soporte comercial de este software está disponible 24/7/365 [6, 10][48][68].</p> <p>Es fácil contratar personal informático en la comunidad que conozca este software [2].</p> <p>Hay desarrolladores en su organización que saben cómo desarrollar este software [16, 69, 72, 74, 85, 94][30].</p> <p>Soporte comercial para la personalización de software está disponible [74].</p>
	Tiempo de adopción	El tiempo que requiere una organización para implementar la solución FLOSS.	<p>Los requisitos de instalación y despliegue del software son fáciles de cumplir [6].</p> <p>El tiempo requerido para adoptar este software es bajo [69].</p>
	Formación	Acciones para mejorar las habilidades y el conocimiento de los usuarios en la organización.	<p>La adopción de este software permite a los usuarios mejorar las habilidades técnicas de TI [16, 49][73][111][125][134].</p> <p>El personal de la organización puede aprender fácilmente por sí mismo a utilizar este software [8][29][49][150][73][90][134].</p> <p>El personal de la organización está capacitado para resolver problemas tecnológicos [2][73][111][134].</p> <p>Los planes de entrenamiento de este software están disponibles [2, 52, 56, 104, 119, 133, 149, 150][112][125].</p>

**Tabla 5.1 (continuación de la página anterior)**

Dimensión	Factor	Descripción	Subfactor
	Casos de estudio de la adopción de FLOSS	Historias exitosas existentes de la adopción de FLOSS por organizaciones similares.	Hay informes públicos disponibles en Internet que describen el éxito de la adopción de este software (de organizaciones similares) [9, 42].
	Centralidad de TI	El grado de dependencia de la organización en su propia infraestructura de TI.	La adopción de este software mejora el entorno de trabajo de los usuarios [17]. Centralizar la infraestructura de TI ayuda a acelerar la adopción de este software [17].
	Soporte de alta dirección	Grado de respaldo de la gerencia a la decisión de adoptar la solución FLOSS.	La alta dirección apoya la adopción exitosa de este software [17][73]. El personal de TI de su organización respalda la adopción de este software [52, 143, 152].
	Bloqueo de proveedores	Dependencias en una solución de software específica de manera que sea costoso cambiar a una alternativa.	El software reduce las dependencias de los proveedores en tu entorno [16][49][48][150][69][79][152][127].
	Actitud hacia el cambio	Comportamiento de los empleados relacionado con los cambios tecnológicos que se adoptarán.	El personal de la organización muestra poca resistencia al cambio tecnológico [42].
Económica	TCO	Los costos de licencia, operación y soporte	Es poco probable que haya costos ocultos al adoptar este software [8, 42, 52, 150][48][129]. La adopción de este software es menos costosa que la alternativa propietaria [8] [29] [49][56][64] [147] [73][83] [90] [107] [48] [68] [111] [134] [10, 42, 133, 150].



### 5.2.2. Encuesta para valorar la importancia de los subfactores

Para ponderar la importancia de los subfactores identificados y si dicha importancia concuerda con la de la literatura, hemos elaborado una encuesta para expertos en FLOSS reformulando los subfactores como preguntas cerradas (ver Apéndice A). Las invitaciones para participar en la encuesta se enviaron a través de correo electrónico a profesionales TIC expertos en FLOSS de Ecuador y Europa.

La encuesta permite a los expertos describir su perfil profesional sobre la base de:

- Sector laboral: público, privado e independiente. Entendemos por independiente a consultores *freelance*.
- Experiencia en FLOSS: número de años, de 1 a 5 años, de 6 a 10 años y mayor a 10 años.
- Especialidad en el ámbito de las TIC: experiencia en educación/investigación, sistema operativo de escritorio, ofimática, desarrollo de software, tecnología web, seguridad informática, infraestructura de la nube y redes, finanzas y gestión.

En la Figura 5.2 podemos observar que la mayoría de los expertos que contestaron la encuesta pertenecen al sector público con un 77.2%, seguido del sector privado con un 15.8% y en menor porcentaje como expertos independientes con un 3.5%.

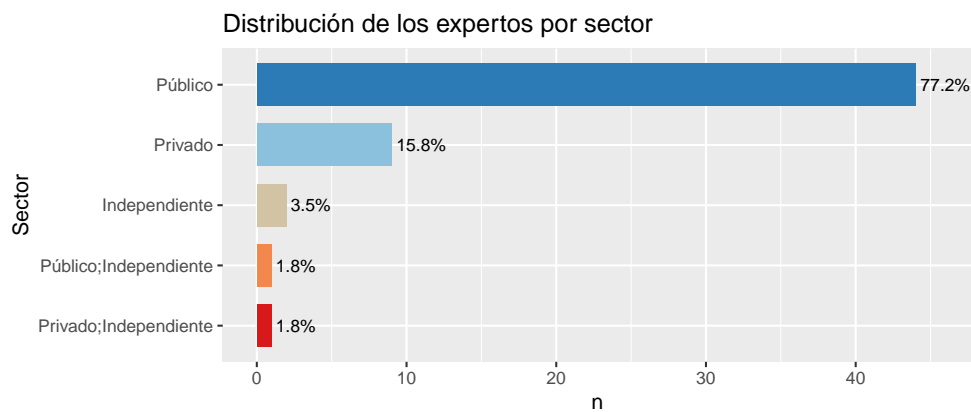


Figura 5.2: Distribución de los expertos por sector.

En la Figura 5.3 podemos observar que el 47.7% de los encuestados posee más de 10 años de experiencia en herramientas FLOSS. Asimismo, el 28.1% de los encuestados poseen experiencia en herramientas FLOSS entre 6 y 10 años. Con menor porcentaje 24.6% tenemos a un grupo de expertos en un lapso de 1 a 5 años.

Podemos observar que en la Figura 5.4 el campo laboral de los encuestados relacionado con el campo de la educación e investigación representan el 28.1% del

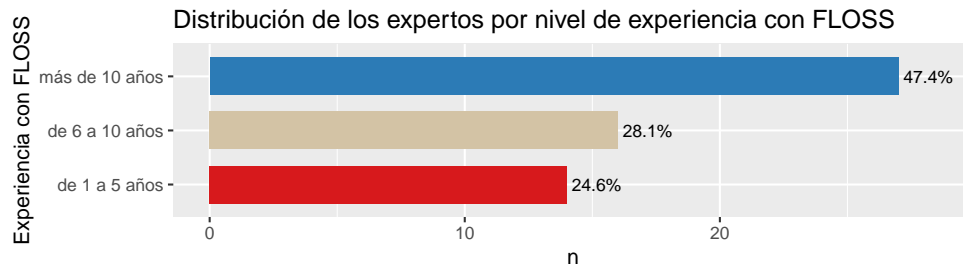


Figura 5.3: Distribución de los expertos por nivel de experiencia.

total de encuestados, seguido de un 21.1% de encuestados que tienen experiencia en sistemas operativos de escritorio. Entre los porcentajes de menor valor tenemos a encuestados con experiencia en el campo de infraestructura en la nube y redes con un 5.3% y el campo de las finanzas y gestión con el 1.8%.

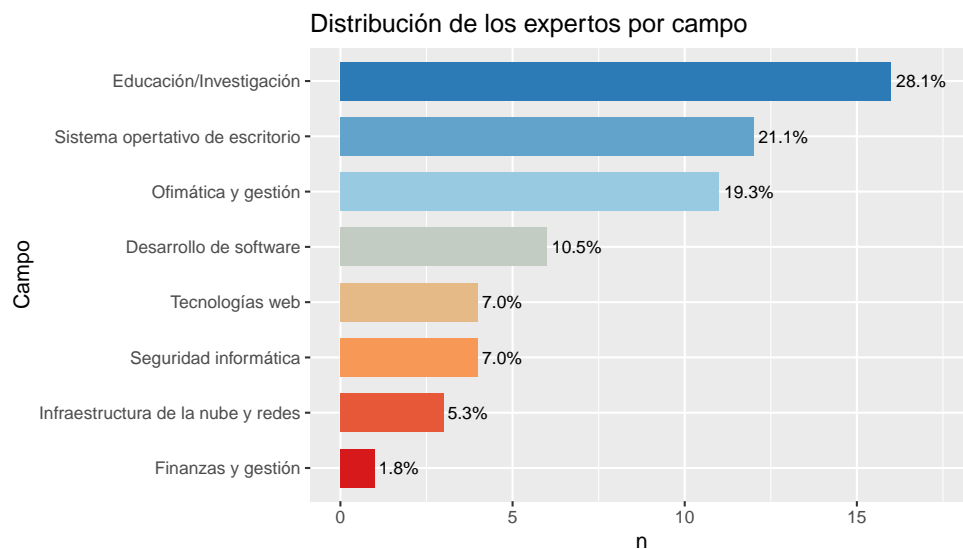


Figura 5.4: Distribución de los expertos por campo.

Para la evaluación de cada subfactor se empleó una escala de cuatro niveles, donde 1 significa que el subfactor es *Irrelevante*, 2 *Opcional*, 3 *Importante*, y 4 *Fundamental*. Muchas investigaciones se han realizado utilizando diferentes puntos en la escala de Likert de hasta 7 puntos [23]. Diferentes autores han demostrado que el número óptimo de categorías es una cuestión de determinación empírica dependiendo de la situación, y que la confiabilidad y validez de un instrumento no se ve afectada por el número de puntos de la escala utilizados para los ítems [47, 86]. Para mostrar los resultados obtenidos de los subfactores, se consideró la proporción de los votos de los expertos por nivel de importancia (fundamental, importante, opcional e irrelevante) en cada subfactor. Esta proporción se calcula dividiendo el total de votos

recibidos por el subfactor en una categoría específica, sobre el total de votos recibidos en dicho subfactor. Los subfactores fueron ordenados dentro de cada factor de forma descendente de acuerdo a la evaluación global obtenida.

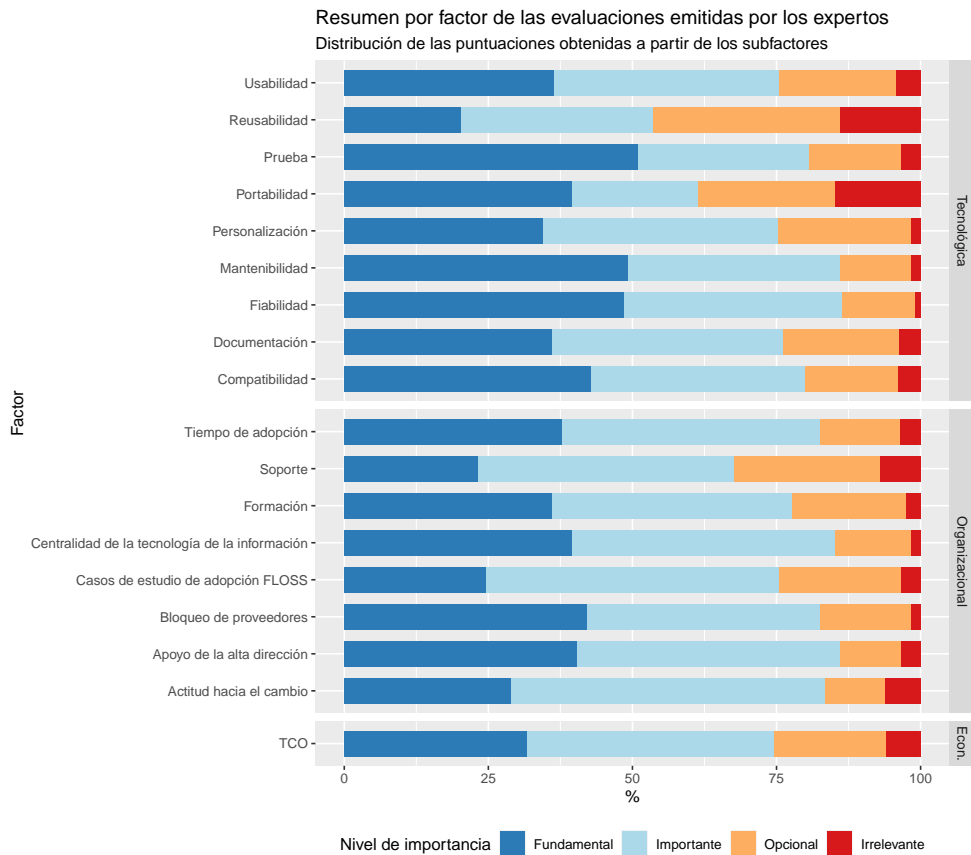


Figura 5.5: Distribución de las evaluaciones de los expertos por factor.

La Figura 5.5 muestra la distribución de las evaluaciones de los expertos por factor y podemos observar que según la valoración de los subfactores del factor *Prueba* es el más representativo en la dimensión *tecnológica* debido a que el nivel de importancia *Fundamental* es la más alta valoración realizada. En la dimensión *organizacional*, el factor *bloqueo de proveedores* a través de sus subfactores consiguió la valoración más alta en el nivel de importancia *Fundamental*. En cambio, en la dimensión *económica*, su único factor alcanzó su mejor valoración en el nivel de importancia *Importante*.

La Figura 5.6 muestra la distribución de las evaluaciones de los expertos por dimensión y observamos que la dimensión *tecnológica* a través de los subfactores evaluados tienen los factores más representativos del proceso. Seguido de la dimensión *organizacional* que a pesar de tener el mismo porcentaje de nivel de importancia *Fundamental*, esta tiene una mayor valoración en el nivel de importancia *Importante*.

Puede verse en la Figura 5.7 que el subfactor más importante dentro del factor *compatibilidad* es *el software es compatible con los casos de uso y funcionalidades más comunes*, mientras que en el menos importante es *el software está certificado para operar en su nicho de mercado*.

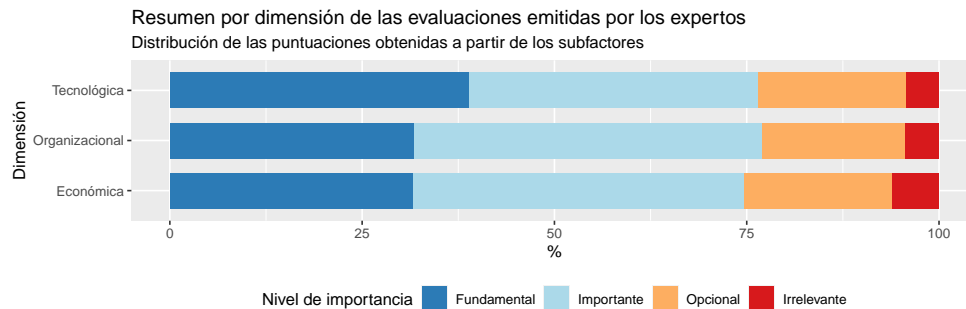


Figura 5.6: Distribución de las evaluaciones de los expertos por dimensión.

En la Figura 5.8 el subfactor más importante dentro del factor *tiempo de adopción* es *los requisitos de instalación y despliegue del software son fáciles de cumplir*, mientras que en el menos importante es *el soporte comercial para la personalización de software está disponible*.

En la Figura 5.9 el subfactor más importante del factor *coste total de propiedad* es *la adopción de este software es menos costosa que la alternativa patentada*, por lo tanto el otro subfactor *es poco probable que haya costos ocultos al adoptar este software* se convierte en el menos importante al existir solo dos subfactores en este factor.

### 5.2.3. Indicadores de importancia de factores y subfactores

En esta sección proponemos indicadores de importancia que permiten evaluar los factores y subfactores que repercuten en el proceso de adopción FLOSS. Para la elaboración de estos indicadores de importancia nos hemos basado en técnicas de evaluación cuantitativa y cualitativa [31]. Los indicadores de importancia permiten determinar cómo de relevante son los factores y subfactores desde tres puntos de vista:

1. Los expertos FLOSS seleccionados, que aportan una valoración externa a la organización que pretenden adoptar el FLOSS.
2. La literatura existente, basado en la cantidad de referencias encontradas a los factores y subfactores identificados.
3. La valoración del propio responsable de la organización que pretende adoptar el FLOSS, puesto que conoce la idiosincracia y el funcionamiento de la organización.

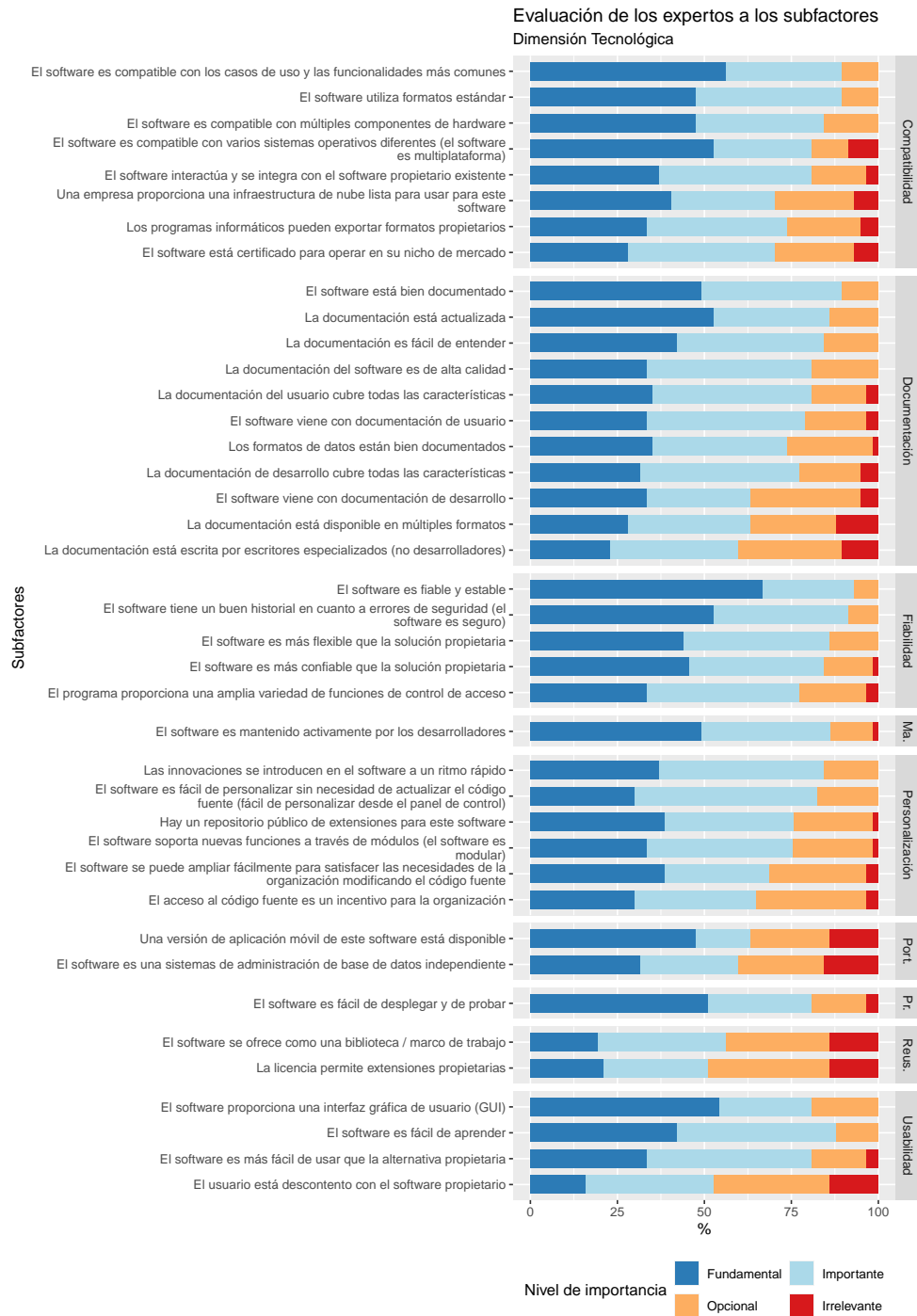


Figura 5.7: Evaluación de expertos a los subfactores de la dimensión tecnológica.

En diferentes investigaciones se han obtenido hallazgos fructíferos y significativos en el uso de número discretos en la escala de Likert dado que se obtenían resultados estadísticos significativos [80, 164], por lo tanto, hemos considerado utilizar números discretos para la valoración de los indicadores de importancia.

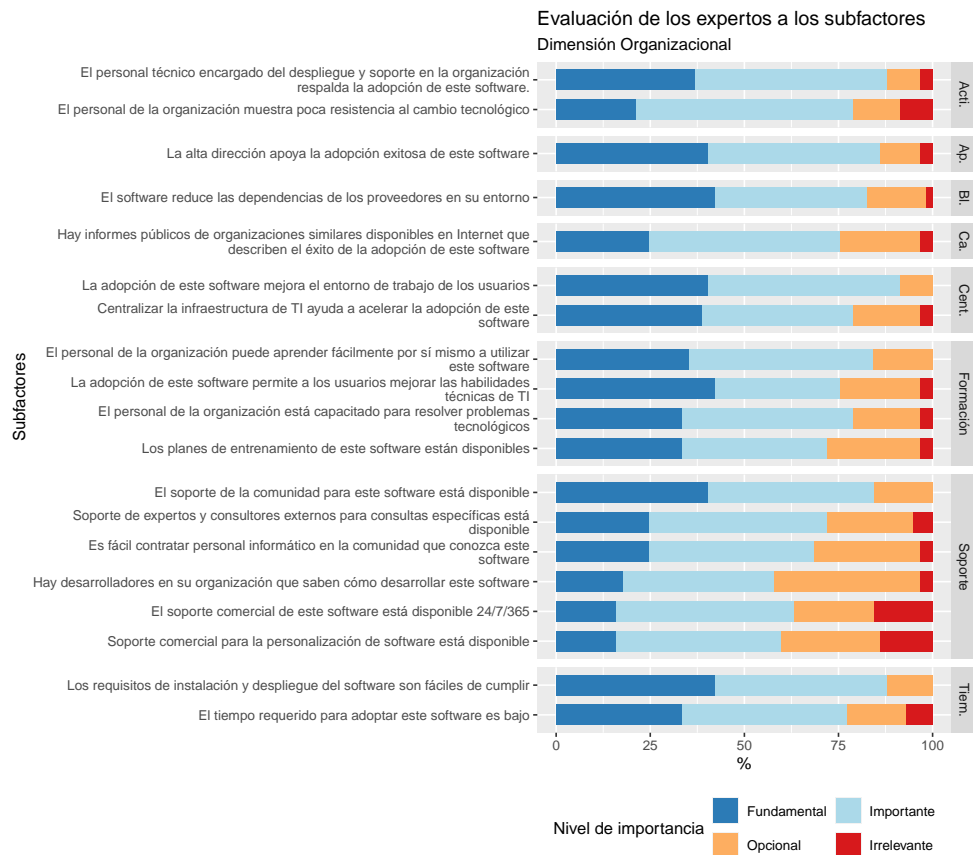


Figura 5.8: Evaluación de expertos a los subfactores de la dimensión organizacional.

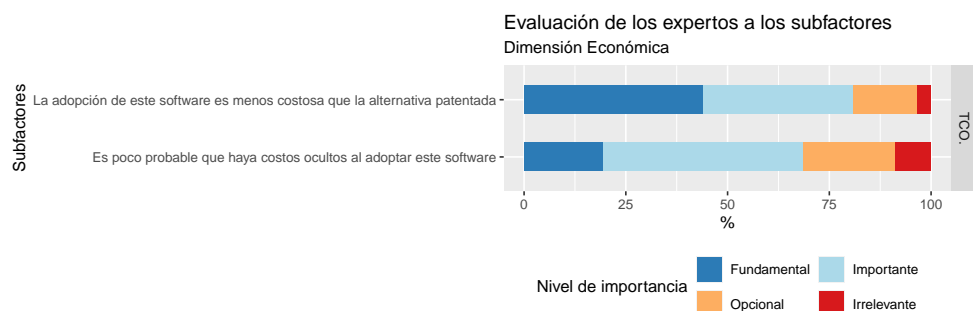


Figura 5.9: Evaluación de expertos a los subfactores de la dimensión económica.

Estos indicadores de importancia aparecen resumidos en la Tabla 5.2. A continuación describimos cada uno de los indicadores propuestos para valorar la importancia de los factores y subfactores en el proceso de adopción FLOSS:

1. **Importancia del experto (IE):** Se definen cuatro niveles de importancia: 1 (irrelevante), 2 (opcional), 3 (importante), 4 (fundamental). Esta evaluación numérica se obtiene por medio de la encuesta propuesta en la Sección 5.2.2

Para obtener la importancia de cada subfactor se calcula la media de las evaluaciones emitidas, con lo que cada factor obtiene una evaluación global en el rango de [1,0,4,0].

Para llevar esta magnitud continua a una escala discreta con un decimal, se redondea de manera usual siguiendo estas 2 reglas:

- a) Si el próximo dígito decimal a redondear es menor o igual a 4, el dígito entero queda igual. Ejemplo: 3,3 se convierte en 3,0.
- b) Si el próximo dígito decimal a redondear es mayor o igual a 5, el dígito entero se incrementa en 1. Ejemplo: 3,7 se convierte en 4,0.

2. **Importancia de la literatura (IL):** La importancia de la literatura se obtiene a partir del número de referencias encontradas a los factores y subfactores similar al descrito por Rincón *et al.*[116]. Los pasos para calcular este indicador son los siguientes:

Indicador	Valoración	Objetivo	Resultado
Importancia del experto (IE)	A partir de la evaluación de los expertos. 4 niveles: irrelevante (1), opcional (2), importante (3), fundamental (4)	Incluir la opinión de expertos.	Obtener la Importancia sugerida.
Importancia de la literatura (IL)	A partir de la cantidad de subfactores y citas. 4 niveles: irrelevante (1), opcional (2), importante (3), fundamental (4)	Incluir la evaluación de la literatura.	Obtener la Importancia sugerida.
Importancia sugerida (IS)	A partir de IE y IL. 4 niveles: irrelevante (1), opcional (2), importante (3), fundamental (4)	Unificar la opinión de expertos y la literatura.	Obtener la Importancia relativa.
Importancia del decisor (ID)	4 niveles: irrelevante (1), opcional (2), importante (3), fundamental (4)	Adaptar GUIOS a los requerimientos de la organización.	Obtener la Importancia relativa.
Importancia relativa (IR)	4 niveles como resultado de la combinación de ID e IS: irrelevante (1), opcional (2), importante (3), fundamental (4)	Caracterizar a los factores.	Seleccionar factores relevantes ( $IR > 1$ ).

Tabla 5.2: Indicadores definidos para caracterizar la importancia de los factores y subfactores en el proceso de adopción FLOSS.

- a) Por cada factor  $i$  se obtiene la cantidad de citas ( $c_i$ ) recibidas por sus respectivos subfactores ( $n$ ). Las citas son obtenidas de bases de datos importantes como IEEE, ACM, Scopus y Web of Science (WOS);
- b) Se divide la cantidad de citas entre el total de subfactores del factor en cuestión:

$$r_i = \frac{c_i}{n} \quad (5.1)$$

- c) Se le asigna un rango de cuartil en función de la posición que ocupa  $r_i$  en el conjunto de todas las  $r$  calculadas a los factores:

$$IL(r_i) = \begin{cases} 1 & \text{si } r_i \leq r_{q1}, \\ 2 & \text{si } r_{q1} < r_i \leq r_{q2}, \\ 3 & \text{si } r_{q2} < r_i \leq r_{q3}, \\ 4 & \text{si } r_{q3} < r_i \leq r_{q4} \end{cases} \quad (5.2)$$

De lo anterior es posible ver que los factores con  $IL = 4$  serán los más importantes, mientras que los menos importantes tendrán un valor  $IL = 1$ . Además, los valores  $r_x$  corresponden a los valores máximos dentro de cada cuartil. Al aplicar este cálculo al conjunto de factores identificados previamente, se obtuvieron los valores que se muestran en la columna  $IL$  de la Tabla 5.3.

Es importante notar que tanto la importancia del experto como la importancia de la literatura fueron redondeadas porque en primer lugar buscamos que estos valores sean interpretables fácilmente para el decisor. Nótese que hemos utilizado un mismo rango de [1.0,4.0] para valorar las importancias, por lo que un valor discreto en este rango resultaría más fácil de hacer corresponder a las etiquetas que se han definido (irrelevante, opcional, importante o fundamental). Por otro lado, estas importancias si bien son valores intermedios para poder determinar otros valores más globales, no son empleados en una operación usual (ej. media aritmética). Como se detallará más adelante, estas importancias son usadas en una operación de producto cartesiano (véase la Figura 5.10), en la que la precisión no es relevante para obtener un resultado específico. Insistimos, esta forma de hacer los cálculos ha sido considerada para favorecer la comprensión del decisor. Reconocemos que existen otras formas que podrían haber aprovechado la precisión de los cálculos intermedios (ej. técnicas de decisión multicriterio [31]).

3. **Importancia sugerida (IS):** la importancia sugerida unifica la importancia que otorgan los expertos y la identificada en la literatura. Hemos definido los siguientes niveles de IS: 1 (irrelevante), 2 (opcional), 3 (importante), 4 (fundamental). Siguiendo a Rincón *et al.* [116], donde propone una función para calcular indicadores de importancia y la cual se asigna a cada factor que se describe en la Figura 5.10. Por ejemplo, para el factor *personalización* si el



Dimensión	Factor	NS*	TC*	IE*	IL*	IS*
Tecnológica	Compatibilidad	8	33	3	3	Importante (3)
	Fiabilidad	5	19	3	3	Importante (3)
	Mantenibilidad	1	4	3	3	Importante (3)
	Personalización	6	30	3	4	Importante (3)
	Prueba	1	4	3	3	Importante (3)
	Documentación	11	20	3	1	Opcional (2)
	Portabilidad	2	4	3	2	Opcional (2)
	Reusabilidad	2	5	3	2	Opcional (2)
	Usabilidad	4	14	3	2	Opcional (2)
Organizacional	Apoyo de la alta dirección	1	5	3	4	Importante (3)
	Formación	4	27	3	4	Importante (3)
	Soporte	6	43	3	4	Importante (3)
	Actitud hacia el cambio	2	3	3	1	Opcional (2)
	Bloqueo de proveedores	1	3	3	2	Opcional (2)
	Casos de estudio de adopción FLOSS	1	1	3	1	Opcional (2)
	Centralidad de la tecnología de la información	2	2	3	1	Opcional (2)
	Tiempo de adopción	2	2	3	1	Opcional (2)
Económica	TCO	2	24	3	4	Importante (3)

Tabla 5.3: Cálculo de la importancia sugerida para caracterizar a los factores y subfactores en el proceso de adopción FLOSS.

\*NS: número de subfactores, TC: Total de citas, IE:Importancia del experto, IL:Importancia de la literatura, IS:Importancia sugerida

valor de la importancia del experto es 3 y el valor de la importancia de la literatura es 4, según la intersección de valores de la función que se muestra en la Figura 5.10 es 3. Hemos representado los valores obtenidos a partir de la encuesta elaborada en la Sección 5.2.2 en la Tabla 5.3.

Como se puede observar en la Tabla 5.3, en la dimensión *tecnológica*, 6 de los 9 factores tienen una *IS* de 3 (importante). El factor con mayor número de citas y número de subfactores es *compatibilidad* con 33 y 8 respectivamente, siendo uno de los factores con mayor presencia en la literatura y mejor ponderación a través de la participación de expertos. En la Dimensión *organizacional*, 3 de los 8 factores tienen una *IS* de 3, producto de una importancia de la literatura de 4 y una ponderación de expertos de 3, siendo el factor *Soporte* el más representativo de esta dimensión con 43 citas y 6 subfactores.

4. **Importancia del decisor (ID):** es la importancia que la persona responsable acerca de la adopción FLOSS otorga a cada factor. Para ello se emplea el siguiente nivel de importancia: 1 (irrelevante), 2 (opcional), 3 (importante), 4 (fundamental). La ID junto con la IS permite obtener la importancia relativa

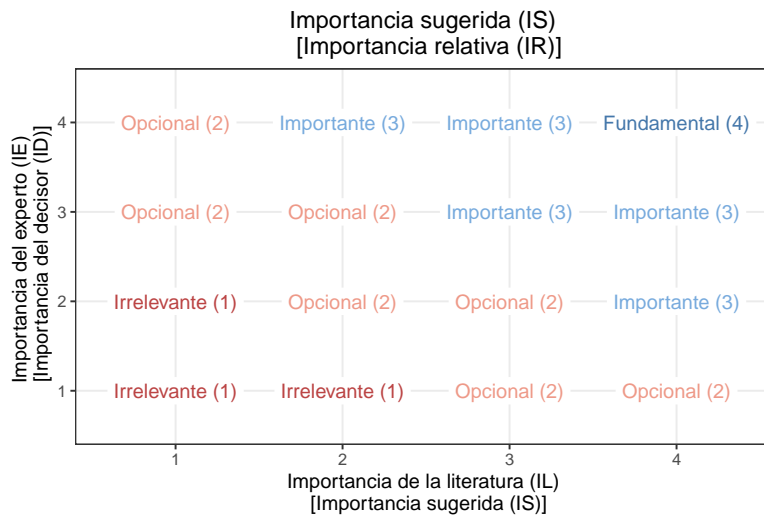


Figura 5.10: Función para determinar la importancia sugerida/relativa de un factor en función de la importancia de la literatura/sugerida y la importancia del experto/decisor.

del factor (IR). Esto permite al decisor valorar los subfactores de acuerdo con los requisitos de la organización. A continuación se describen cómo se obtiene la ID:

- a) **Ponderación del subfactor (PS):** la ponderación del subfactor evalúa la relevancia de los subfactores que tienen la importancia relativa mayor a 1, es decir, solo se evaluarán factores relevantes para la organización. En tal sentido, se ha establecido una escala para evaluar si el subfactor cumple con requisitos suficientes para su adopción, esto es, si la organización dispone de recursos propios para aplicar esta característica. Esta escala es: 1 (No cumple el requisito), 2 (No sé), 3 (Cumple parcialmente el requisito), 4 (Cumple el requisito).
- b) **Ponderación media del factor (PM):** la ponderación media del factor, promedia las ponderaciones de los subfactores por cada factor evaluado. Los valores que se obtienen como resultado están dentro del rango [1,0,4,0] y que servirá posteriormente como una de las entradas de información para clasificar a los factores bajo un esquema FODA.
- c) **Clasificación FODA:** para analizar la situación actual de una empresa, el FODA es una de la herramientas más conocidas que permite mejorar e innovar los procesos en una organización [54]. Para esto, es necesario clasificar a los factores como internos y externos previamente para que, con su respectiva PM, clasificarlos bajo el enfoque FODA (Fortaleza, Oportunidad, Debilidad y Amenaza).

Para la clasificación de los factores como internos y externos, hemos considerado lo siguiente:

- *Los factores internos* son los que hacen referencia a factores que vienen determinados por la propia organización que va a adoptar el FLOSS.
- *Los factores externos* son los que hacen referencia a aspectos tecnológicos que viene predeterminados por el software a adoptar.

A continuación en la Tabla 5.4 se muestran los 18 factores que fueron identificados en el capítulo anterior, agrupados en factores internos y factores externos según las consideraciones descritas previamente para su clasificación:

Factores	
Internos	Formación
	Tiempo de adopción
	Centralidad de TIC
	Apoyo de la alta dirección
	Soporte (*)
	Actitud hacia el cambio
	Coste total de propiedad
Externos	Compatibilidad
	Personalización
	Prueba
	Fiabilidad
	Soporte (*)
	Reusabilidad
	Usabilidad
	Mantenibilidad
	Portabilidad
	Documentación
	Casos de estudio de adopción FLOSS
Bloqueo de proveedores	

Tabla 5.4: Clasificación de factores internos y externos bajo un enfoque FODA.

En (\*) aparece el factor soporte que puede ser evaluado tanto desde un alcance interno como externo. Consideramos que el factor soporte se evalúa internamente cuando el soporte del software a adoptar se va a realizar con ingenieros empleados por la propia organización. En este caso, estamos evaluando el *Know-How* del personal experto responsable del soporte tecnológico de la organización. El soporte se evalúa externamente cuando se tiene en cuenta el apoyo en la adopción ofrecidos por terceras

partes, tales como empresas especializadas en dicho software, así como la comunidad entorno al FLOSS.

Siguiendo lo sugerido en [116], estas categorías serán aplicables de acuerdo a las siguientes reglas de asignación:

$$FODA(pm_i, impacto_i) = \begin{cases} Fortaleza & \text{si } pm_i \geq 3 \text{ e } impacto_i = \text{interno}, \\ Oportunidad & \text{si } pm_i \geq 3 \text{ e } impacto_i = \text{externo}, \\ Debilidad & \text{si } pm_i < 3 \text{ e } impacto_i = \text{interno}, \\ Amenaza & \text{si } pm_i < 3 \text{ e } impacto_i = \text{externo} \end{cases} \quad (5.3)$$

donde  $pm_i$  es la ponderación media del factor  $i$ , mientras que  $impacto_i$  es el impacto del factor en la organización, el cual puede ser o bien interno o externo.

5. **Importancia relativa (IR):** a partir de los factores evaluados por el decisor y la literatura, es posible calcular la importancia relativa de cada uno de ellos. Este indicador se ha definido de forma similar al propuesto por Rincón *et al.* [116], tal y como se ilustra en la matriz de la Figura 5.10, utilizando el mismo mecanismo de asignación de valores que la IS. Nótese que esta matriz es una función que asigna a cada combinación de valores de las ID y la IS, un único valor de IR al factor.

#### 5.2.4. Amenazas a la validez

Aunque en este trabajo hemos diseñado un instrumento tipo encuesta para recolectar y procesar información hay algunas suposiciones que hemos hecho que pueden afectar a su validez [162]:

**Validación externa:** Seleccionamos expertos en el área FLOSS según los artículos de los estudios primarios de esta investigación, pero podríamos haber pasado por alto otros expertos del área que han publicado en otros espacios científicos no identificados en nuestro conjuntos de artículos. Asimismo, el no contactarnos con expertos locales o internacionales que propician el movimiento FLOSS como la *Free Software Foundation* (FSF) o la Asociación de software libre del Ecuador (ASLE) podría haber dejado fuera valoraciones relevantes en la encuesta sobre la adopción FLOSS. Hemos minimizado el impacto de esta amenaza mediante la selección de expertos en tecnologías FLOSS que laboran en instituciones públicas y privada. Consideramos que las amenazas más importantes para la validez externa son:

- Validez de la población. Los expertos seleccionados para llenar la encuesta son profesionales del área de la tecnología de la información que tienen experiencia en el uso de herramientas FLOSS. Muchos de los expertos seleccionados para participar de esta encuesta fueron autores y co-autores de los estudios

primarios que obtuvimos en el proceso sistemático de la literatura y que están relacionados con el tema principal de esta investigación. Al ser enviada la encuesta, no obtuvimos una cantidad de respuestas deseadas por parte de este grupo de expertos. Asumimos que debido al tiempo no disponible de los expertos, no pudieron participar de esta actividad. Se enviaron los enlaces de las encuestas digitales a través de correo electrónico a expertos en tecnologías FLOSS en instituciones públicas y privadas en el Ecuador y España.

- Validez ecológica. Se centra en los posibles errores en los materiales y herramientas utilizados en el experimento. La identificación de los subfactores en los artículos que componen este estudio de investigación podría generar desconfianza entre los investigadores de la comunidad. Para evitar esta percepción, todos los autores de este estudio realizaron individualmente la identificación de los subfactores y luego, en consenso discutimos sobre los resultados obtenidos para construir una lista de subfactores general.

**Validación interna:** es una medida que asegura que el diseño del experimento del investigador sigue el principio de causa y efecto. Para el diseño de la encuesta, se definieron subfactores basados en características identificadas en los artículos de investigación que forman parte de este estudio. Sin embargo, la cantidad de subfactores identificados por factor pueden dar lugar a errores. En este contexto, consideramos pertinente que la identificación de los subfactores fueran revisados individualmente por cada uno de los expertos asociados a este trabajo de investigación para minimizar el impacto.

### 5.3. Resumen

En este capítulo hemos presentado un proceso que detalla la elaboración de GUIOS, una guía para evaluar el proceso de adopción FLOSS. En tal sentido, se adoptó un proceso *ad-hoc* que aprovechó en gran medida los resultados obtenidos del Capítulo 4. Esta información fue complementada a partir de dos fuentes adicionales: una revisión de la literatura en profundidad de trabajos de congreso que abordan la adopción FLOSS, y en segundo lugar, la validación de contenido ofrecida por un conjunto de expertos a través de una encuesta sobre subfactores de adopción FLOSS. Concretamente se realizó lo siguiente:

- Revisión de dimensiones y factores: con base a la revisión de la literatura, se confirmaron las dimensiones y factores identificados en el capítulo anterior. El objetivo de esta actividad fue la de identificar nuevas dimensiones y factores a los ya identificadas previamente, y que contribuyan con los criterios de medición para la guía de adopción FLOSS.
- Creación de subfactores: con la finalidad de obtener un nivel de granularidad que caractericen con mayor precisión a los factores identificados y sobre la base de la revisión de la literatura, se definieron un total de 61 subfactores, los cuales fueron organizados en 18 factores y tres dimensiones: tecnológica, organizacional y económica.
- Elaboración y aplicación de una encuesta a expertos sobre la adopción FLOSS con preguntas de respuestas cerradas. Esta encuesta tuvo por objetivo conocer la importancia que los expertos le otorgan a los subfactores identificados en este trabajo de investigación. Adicionalmente, los expertos tenían la posibilidad de sugerir nuevos subfactores. No se obtuvieron sugerencias por parte de los expertos.
- Definición de indicadores para evaluar la importancia de factores y subfactores: los indicadores tenían como objetivo caracterizar la importancia de los factores en el contexto de la adopción FLOSS. Se definieron cinco indicadores de importancia, donde se propuso una metodología basada en técnicas de evaluación cuantitativa y cualitativa.

## Capítulo 6

# Elaboración y validación de GUIOS para la adopción FLOSS

Dando continuidad a los resultados del capítulo precedente, en este se describen los pasos a seguir para el uso de GUIOS. A través de un enfoque FODA que integra las valoraciones de la literatura, expertos y decisores, GUIOS recomienda qué acción tomar con respecto a la adopción del FLOSS. Para hacer operativa esta guía, nos planteamos el desarrollo de un herramienta prototipo que hemos denominado GUIOS PRO. Mediante tres interfaces generales, GUIOS PRO permite al decisor introducir sus ponderaciones a nivel de factores y subfactores para visualizar la recomendación final. Con el objetivo de validar preliminarmente ambas propuestas, hemos desarrollado un piloto de caso de estudio en la Universidad Estatal de Milagro. Los resultados obtenidos fueron positivos y nos permitieron identificar puntos de mejora a futuro.

La estructura del capítulo es la siguiente: la Sección 6.1 describe los pasos a seguir para el uso de GUIOS. En la Sección 6.2 se muestra el desarrollo de un piloto de caso de estudio para validar preliminarmente GUIOS. En la Sección 6.3 concluimos con un resumen de los principales resultados

## 6.1. Guía para adopción FLOSS

GUIOS es una guía para evaluar factores que repercuten en la adopción de software libre en una organización. Esta guía tiene por objeto asistir a los responsables de administrar los recursos tecnológicos de las organizaciones en la idoneidad de la adopción del software. La guía está dirigida a los responsables de la toma de decisión para justificar la selección del software a adoptar.

GUIOS incluye un diagrama de flujo compuesto de una serie de entradas y de pasos que formaliza el proceso de toma de decisión en la adopción FLOSS.

Las entradas se refieren a valoraciones de las dimensiones, factores y subfactores que son relevantes en la adopción FLOSS. Dichas valoraciones están basadas en estudios de trabajos de literatura existente, encuestas a expertos y al decisor que se está planteando la adopción.

Los pasos de esta guía, son el resultado de procesar las valoraciones de entrada de las dimensiones, factores y subfactores desde diferentes perspectivas. La Figura 6.1 resume este proceso a través de un diagrama de flujo.

A continuación se detallan los pasos de GUIOS:

- **Paso 1. Cálculo de la importancia relativa de los factores:** en este paso, GUIOS toma como entradas la importancia del decisor y la importancia sugerida por cada factor. Con estas dos entradas, GUIOS emplea la función definida previamente para este propósito y le asigna a cada factor una *Importancia relativa*. Nótese que, como bien se ilustra en la Figura 6.1, la asignación de esta importancia a los factores de adopción, es la principal salida en este paso.
- **Paso 2. Selección de los factores relevantes:** teniendo en cuenta la importancia relativa de cada factor, GUIOS selecciona solo los factores que han sido considerados como relevantes. Concretamente, por relevantes se entienden aquellos factores con importancia relativa etiquetada como *Opcional*, *Importante* o *Fundamental*. Los factores etiquetados como *Irrelevante* son excluidos para reducir la cantidad de factores a considerar en la adopción FLOSS. Como ilustra la Figura 6.1, la salida de este paso son los factores seleccionados como relevantes que son, desde un punto de vista formal, un subconjunto de los factores que sirvieron de entrada a GUIOS en el paso 1. Adicionalmente, una vez clasificados todos los factores como internos y externos a excepción del factor *Soporte*, el decisor procede a evaluar el factor restante como interno o externo.



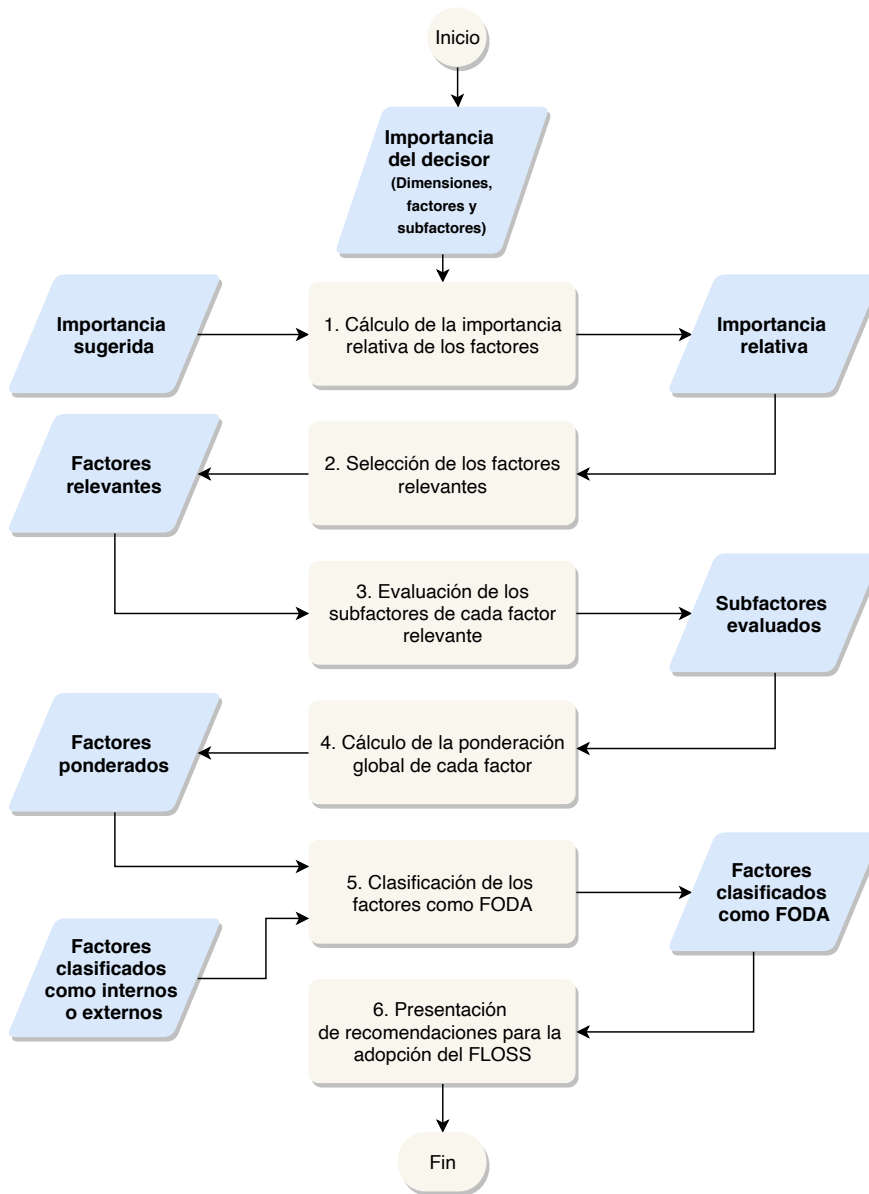


Figura 6.1: Pasos a seguir de GUIOS.

- Paso 3. Evaluación de los subfactores de los factores relevantes:** por cada factor seleccionado en el paso anterior, el decisor deberá evaluar si los subfactores cumplen con los requisitos necesarios, sean estos requisitos propios del software o recursos relacionados con la organización en sí. El resultado obtenido en este paso es fundamental, para que GUIOS sugiera si se puede adoptar o no el FLOSS. El nivel de importancia empleado fue definido en la sección anterior. Como se muestra en la Figura 6.1, la salida de este paso está constituida precisamente por los subfactores evaluados. Es importante notar que este

es uno de los dos pasos (junto con el Paso 1), en el que el decisor genera la entrada necesaria para evaluar la importancia de los subfactores. Puesto que la importancia sugerida viene ya predeterminada por la literatura y la encuesta a expertos.

- **Paso 4. Cálculo de la ponderación media (PM):** a partir de las evaluaciones emitidas a los subfactores, GUIOS calcula por cada factor una ponderación media. La principal salida de este paso son los factores ponderados globalmente (Figura 6.1).
- **Paso 5. Clasificación de los factores como FODA:** una vez obtenido el resultado del paso anterior (PM), y teniendo los factores clasificados como internos y externos previamente, GUIOS clasifica a los factores como Fortaleza, Oportunidad, Debilidad y Amenaza.
- **Paso 6. Presentación de recomendaciones para la adopción FLOSS:** como paso final, GUIOS le presenta al decisor un resumen de la evaluación realizada. Concretamente, a partir de la caracterización de los factores a través de los diferentes niveles de importancia relativa, la ponderación de los factores y sobre todo con la clasificación FODA de los factores, GUIOS mostrará recomendaciones que ayuden a la toma de decisiones en cuanto a la adopción (o no) del FLOSS. Nótese que la principal entrada de este paso son los factores ya clasificados como FODA.

### 6.1.1. GUIOS PRO: un prototipo para la automatización de GUIOS

Una pregunta natural que surge a partir de la guía descrita anteriormente, es cómo hacerla operativa de manera que pueda ser explotada por usuarios reales, esto es, por decisores que tienen la responsabilidad de evaluar la adopción FLOSS en una organización en particular. En tal sentido, hemos implantado una herramienta prototipo que, lejos de ser la solución definitiva a esta cuestión, servirá de medio para la validación de GUIOS. Intuitivamente, hemos llamado a esta herramienta *GUIOS PRO*. Se trata de una aplicación prototipo multi-plataforma desarrollada en *Flexx* versión 0.8.0<sup>1</sup>. *Flexx* es una toolkit de Python para la creación de interfaces gráficas de usuario (GUI) y que utiliza la tecnología web para su representación. Esta tecnología es un FLOSS distribuido bajo licencia AGPL3.

Como se aprecia en la Figura 6.2, GUIOS PRO agrupa en diferentes interfaces, los 6 pasos de GUIOS, esto es, de 2 en 2:

- *Paso 1 y 2. Obtención de factores relevantes,*
- *Paso 3 y 4. Obtención de factores ponderados, y*
- *Paso 5 y 6. Evaluación y recomendación sobre la adopción.*

---

<sup>1</sup>Copyright (c) 2015-2018, Flexx developers. All rights reserved. (<https://flexx.readthedocs.io/>)

Como entrada primaria, GUIOS PRO permite introducir información sobre factores, subfactores e importancias sugeridas (IS), y que, como se estableció en el capítulo anterior, fueron obtenidas mediante la revisión de la literatura y encuesta a expertos. La Figura 6.2 muestra un ejemplo de esta funcionalidad en GUIOS PRO. Con el ánimo de no hacer demasiada extensa la descripción del resto de funcionalidades, hemos decidido ir presentándolas a través de la validación que mostraremos en la siguiente sección.

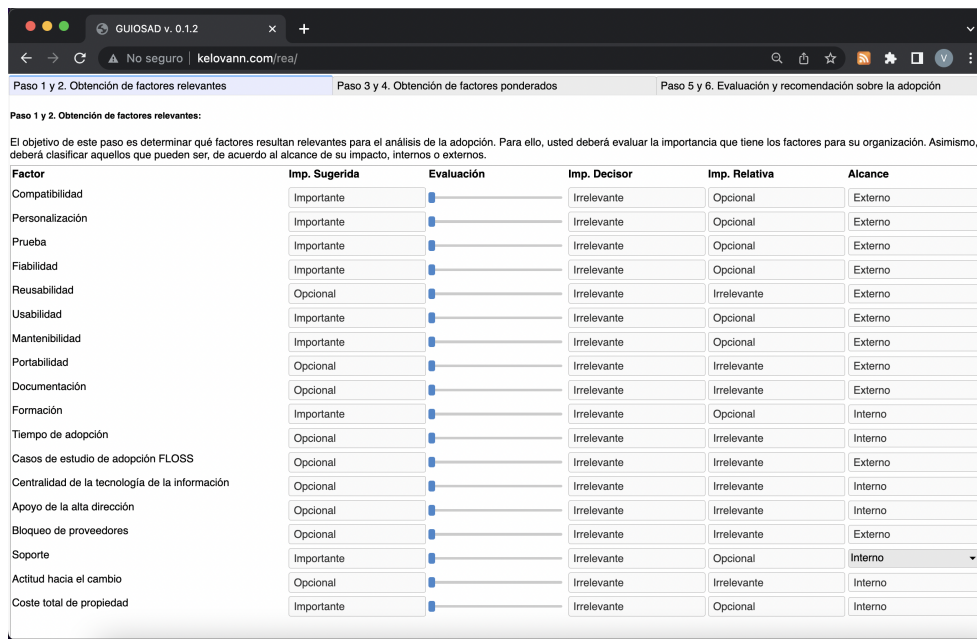


Figura 6.2: Interfaz de GUIOS PRO correspondiente a los pasos 1 y 2 de GUIOS.

## 6.2. Una validación preliminar de GUIOS

Con el objetivo de validar preliminarmente a GUIOS, y más específicamente los pasos que proponemos para evaluar la adopción FLOSS en organizaciones, hemos diseñado un piloto de un caso de estudio. Para desarrollarlo, nos hemos apoyado en nuestra herramienta prototipo GUIOS PRO y en los pasos descritos por Runeson *et al.* [121] y Wholin *et al.* [161]. De acuerdo con Runeson *et al.* y Wholin *et al.* el primer paso es el diseño del estudio, donde se planifican y definen los objetivos del piloto del caso de estudio. El segundo, es planificar la recogida de datos, donde se definen los procedimientos y protocolos de este proceso. En tercer lugar, se recopilan los datos, para luego analizarlos en el cuarto paso. Por último, en el quinto paso, la presentación de informes, se deben difundir los resultados y las conclusiones. Dado que nuestro caso es un primer piloto de la validación, a continuación describiremos no solo los detalles de cada paso, sino también las adaptaciones realizadas a este proceso.

### 6.2.1. Diseño del estudio: contexto y unidad de análisis

El piloto del caso de estudio tiene por objetivo general evaluar la utilidad de GUIOS y su implantación en la herramienta prototipo GUIOS PRO. Concretamente, nos hemos planteado las siguientes preguntas:

1. ¿Facilita GUIOS la adopción sistemática de FLOSS en una organización?
2. ¿Es GUIOS PRO una prototipo adecuado de una futura herramienta que automatice eficientemente a GUIOS?

Para responderlas, hemos elegido como contexto del piloto del caso de estudio a la Universidad Estatal de Milagro (UNEMI), la cual es una institución pública con 21 años de antigüedad y situada en Ecuador, América del Sur. Su misión es formar profesionales competentes con actitud proactiva y valores éticos, desarrolla investigación relevante y oferta servicios que demanda el sector externo, contribuyendo al desarrollo de la sociedad. UNEMI está conformada por 4 facultades académicas, 21 carreras modalidad presencial, 3 carreras en modalidad semipresencial y 10 carreras modalidad en línea. UNEMI cuenta con 242 docentes titulares y 36.604 estudiantes en total.

Entre sus departamentos de tecnología, está el de tecnología de la información y comunicaciones (TICs), que a su vez se subdivide en gestión de desarrollo de sistemas de información, gestión de infraestructura tecnológica y gestión de operaciones. Otro departamento de tecnología se denomina dirección de servicios informáticos, y este se divide en gestión de soporte tecnológico, gestión de infraestructura de seguridad y gestión de seguridad de la información.

El departamento de TICs ha desarrollado un FLOSS de gestión académica donde permite a sus estudiantes el registro de la matrícula, tener reportes sobre toda la información académica y demás procesos académicos. Asimismo, permite al docente registrar y disponer de información pertinente para su correcta gestión académica. El departamento de TICs, ha implantado exitosamente *Moodle*, un FLOSS como herramienta de entorno virtual de aprendizaje para un interacción flexible entre docentes y alumnos.

Se trata por tanto de un contexto elegido por la cercanía al contexto de la presente investigación, pero que, para los propósitos de la misma, cumple suficientemente con las características que buscábamos para el piloto.

Por otro lado, como *unidad de análisis*, nos hemos centrado en una especialista en TICs que actualmente desempeña el cargo de Directora de TICs. Este especialista, que actuará de decisor en nuestra validación, es una profesional con grado en Ingeniería en Sistema Computacionales y Magíster en Gestión y Diseño Web. Posee certificaciones en *COBIT 5 Foundation Certificate*, *ITIL Foundation Certificate in IT Service Management* y *Scrum Master*. Con 15 años de experiencia laboral, ha incurrido en temas como usabilidad y accesibilidad web, servicios web, técnicas de posicionamiento, analítica web, marketing digital, metodologías ágiles, estándares web, diseño y desarrollo para dispositivos móviles.

Es importante aclarar que, para el desarrollo de este piloto del caso de estudio, se contó con la autorización del comité de ética de la UNEMI, así como con el consentimiento informado de la especialista participante. En ambas solicitudes, se manifestó que los fines de la investigación son puramente académicos y que el anonimato de la persona participante estaría garantizado.

El FLOSS elegido para el piloto del caso de estudio es la suite ofimática *LibreOffice*<sup>2</sup>. Esta suite es un proyecto de software libre bajo licencia MLP (licencia pública de Mozilla) en su versión 2.0. La elección de este ha sido consensuada con la especialista que participará en el estudio. En tal sentido, hay que tener en cuenta que UNEMI no cuenta con la alternativa de optar por otras suites ofimáticas como Microsoft Office, debido a que estas son de pago y además, no se alinean a las políticas de la institución que promueven el uso de FLOSS.

Formalmente, el piloto del caso de estudio sería de tipo *exploratorio* [99], ya que nos interesa conocer el grado de utilidad de GUIOS en la adopción de un FLOSS en una organización real. Dado que GUIOS PRO fue implantada *ad-hoc* para esta investigación, existe incertidumbre acerca de su grado de aceptación por parte de usuarios reales.

Las actividades a desarrollar dentro del piloto del caso de estudio son las siguientes:

1. Capacitación sobre GUIOS y GUIOS PRO.
  - a) Duración: 1 hora y 30 minutos (1 hora de capacitación, y 30 minutos de interacción con GUIOS PRO y aclaración de dudas).
  - b) Lugar: Departamento de TICs de la UNEMI.
2. Evaluación de la adopción de LibreOffice a través de GUIOS PRO.
  - a) Duración: 1 hora (1 hora de interacción observada, pero sin intervención directa por parte del investigador).
  - b) Lugar: Departamento de TICs de la UNEMI.
3. Desarrollo de la entrevista final a la especialista.
  - a) Duración: 45 min (se realizan preguntas y se registran las respuestas en audio).
  - b) Lugar: Departamento de TICs de la UNEMI.

Todas estas actividades fueron planificadas temporalmente en ese mismo orden, y desarrolladas en un mismo día con recesos de 10 minutos entre cada actividad.

---

<sup>2</sup><https://es.libreoffice.org/>

### 6.2.2. Recogida de datos: planificación, estrategias y análisis

Para obtener suficiente información que facilite las respuestas a las preguntas de este piloto del caso de estudio, hemos considerado dos tipos de estrategias de recogida de datos. La primera, es la *observación* [11]. A través de este método, buscamos capturar fundamentalmente comportamientos del usuario que interactuará con la herramienta GUIOS PRO. Además, registraremos los tiempos de capacitación que recibirá la especialista, así como los de interacción con GUIOS PRO.

Por otro lado, nos apoyaremos en la técnica de *entrevista* para conocer de primera mano, las valoraciones emitidas por el usuario sobre GUIO y GUIOS PRO. En tal sentido, hemos diseñado una entrevista estructurada [163], compuesta por las siguientes preguntas:

1. ¿Cómo valoraría usted la utilidad de la guía GUIOS? ¿Por qué?
2. ¿Cómo valoraría usted las principales funcionalidades de GUIOS PRO? ¿Por qué?
3. ¿Qué funcionalidades considera usted que serían importantes incluir en próximas versiones de la guía y de la herramienta? ¿Por qué?

Los datos recolectados de la entrevista serán codificados y procesados a través del *análisis de contenido* [13]. Concretamente, basados en el objetivo y preguntas del piloto del caso de estudio, se planea realizar un proceso de codificación de las entrevistas para sintetizar el significado de cada respuesta.

### 6.2.3. Resultados preliminares

A continuación se describen los resultados obtenidos por cada actividad desarrollada en el piloto del caso de estudio.

- **Capacitación sobre GUIOS y GUIOS PRO.** Durante la capacitación, se realizó una presentación sobre ambos productos a la especialista, tomando como principal fuente, los aspectos descritos en la Sección 6.1. En tal sentido, se observó que la especialista captó adecuadamente la motivación y los fundamentos de la guía de adopción propuesta. Adicionalmente, se pudo constatar cómo realizó una correspondencia efectiva entre los pasos de GUIOS y las funcionalidades de la herramienta prototipo GUIOS PRO. Esta actividad duró 1 hora y 40 minutos, pues fue preciso desarrollar más de un ejemplo durante la interacción con GUIOS PRO para abordar los diferentes escenarios de decisión finales.
- **Evaluación de la adopción de LibreOffice a través de GUIOS PRO.** Esta actividad fue desarrollada por la especialista sin ayuda del investigador. Sin embargo, tanto el tiempo como los comportamientos de la interacción con GUIOS PRO fueron observados por este último. Es importante mencionar que, las capturas que se emplearán a continuación, para describir la interacción de

la especialista fueron recreadas por el investigador a partir de la observación realizada.

En particular, la especialista dedicó 15 minutos en realizar los pasos 1 y 2 de GUIOS. Como se especificó en la Sección 6.1, estos tienen por objetivos: definir qué factores resultan relevantes para la adopción, y clasificarlos de acuerdo al alcance de su impacto. Como se ilustra en la Figura 6.3, la configuración realizada a través de controles tipo *sliders* (columna *Evaluación*) reconoce como factores fundamentales a la *compatibilidad* y al *soporte*, mientras que factores opcionales se consideraron a *reusabilidad*, el *tiempo de adopción*, la existencia de *casos de estudio de adopción*, entre otros. El resto de los factores fueron considerados como importantes. Nótese que el único factor que hemos considerado que puede tener un alcance interno o externo es el *soporte*, y que la especialista optó por considerarlo interno. A priori, se puede decir que esta evaluación preliminar es consistente con un FLOSS como LibreOffice. En tal sentido, es esperable que una gran parte de los futuros usuarios dentro de la organización, requieran de un soporte adecuado para capacitarlos, y por otro lado, compatibilidad con documentos que habitualmente se trabajan en formatos de Microsoft Office.

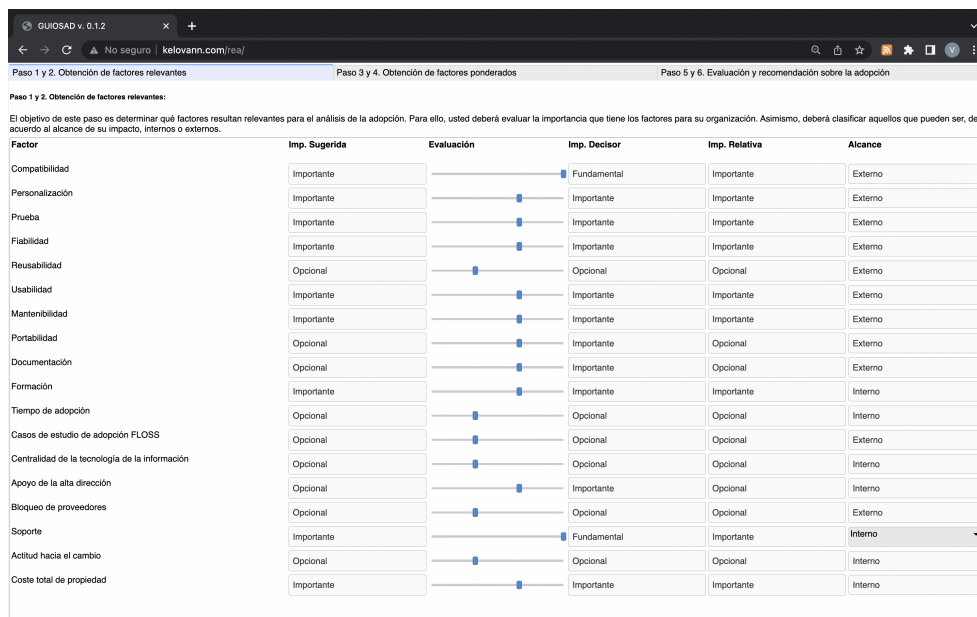


Figura 6.3: Interacción de la especialista en la interfaz de los pasos 1 y 2 de GUIOS

Posteriormente, la especialista centró su interacción en los pasos 3 y 4, con el objetivo de especificar en qué medida, los subfactores que componen a los factores elegidos previamente, son relevantes para la evaluación. Esta interacción duró aproximadamente 25 minutos. De acuerdo a la observación realizada, se pudo ver que la principal razón de esta demora es la complejidad de la entrada

de datos de GUIOS PRO. A diferencia de la interfaz anterior, en la que el usuario requiere de un solo *clic* para establecer la relevancia del factor, en esta, el usuario necesita mínimo tres. Específicamente, uno para seleccionar el factor en la lista superior, uno para evaluar al menos un subfactor, y finalmente uno para guardar la selección. A modo de ejemplo, hemos considerado solo una de las diferentes entradas de la especialista en esta interfaz de GUIOS PRO, esto es, la que se muestra en la Figura 6.4. Como se puede apreciar, el ejemplo se refiere al factor *compatibilidad*. Aquí, la especialista optó por evaluar a la mayoría de subfactores que *cumplen con el requisito* en la organización. En particular, declaró que desconoce si existen software en la nube asociadas a LibreOffice. En nuestra opinión, esta evaluación es consistente con las características de LibreOffice.

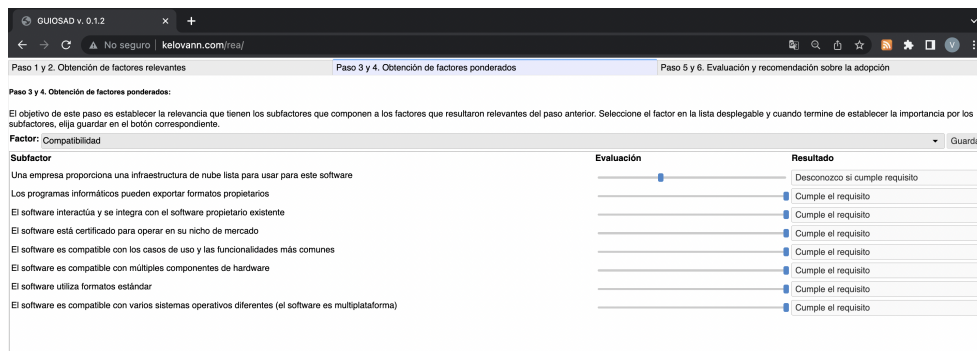


Figura 6.4: Interacción de la especialista en la interfaz de los pasos 3 y 4 de GUIOS

Finalmente, la especialista pasó a la interfaz correspondiente a los pasos 5 y 6. Esta tiene como objetivo mostrar el análisis FODA de cada factor, esto es, a partir de la ponderación media del factor y la clasificación del alcance. Dado que esta interfaz tiene un propósito de resumen, la especialista dedicó apenas 10 minutos en analizar la información individual de cada factor y la recomendación hecha por GUIOS PRO (que aparece cuando se da clic en el botón *Ver recomendación*, Figura 6.5). Un comportamiento que se observó durante esta fase fue la navegación hacia las interfaces anteriores. En la percepción personal del investigador, este comportamiento fue motivado por la intención de encontrar una explicación a las evaluaciones finales y recomendación de GUIOS PRO. Aunque este puede sugerir la inclusión de una funcionalidad futura que facilite esta acción, también es señal de que la especialista comprendió el flujo de información que implanta GUIOS PRO. En otras palabras, la especialista fue consciente del origen de las evaluaciones resumidas en esta última interfaz. La Figura 6.5 muestra cómo quedó esta información al concluir con la interacción. La columna FODA muestra que factores como la *compatibilidad*, *prueba*, *fiabilidad*, *reusabilidad*, *usabilidad*, *mantenibilidad*, *casos de estudio de adopción FLOSS* son vistos como oportunidades en la adopción, mientras



que *tiempo de adopción, apoyo de la alta dirección, actitud hacia el cambio y coste total de propiedad* son fortalezas. En cambio, la *personalización, portabilidad, documentación y el bloqueo de proveedores*, son considerados como amenazas, mientras que la *formación, centralidad de la TI y el soporte* como debilidades. Esta combinación de evaluaciones condicionan la recomendación tipo C, mostrada en la parte inferior de la interfaz. Es importante mencionar que GUIOS PRO considera otros dos tipos de recomendaciones, como se ilustra en la Figura 6.6. Estas dependerán de otras configuraciones de resultados que involucren en mayor medida la presencia de factores considerados como fortalezas u oportunidades.

Factor	Ponderación media del factor	Alcance	FODA
Compatibilidad	3.8	Externo	Oportunidad
Personalización	1.7	Externo	Amenaza
Prueba	3.0	Externo	Oportunidad
Fiabilidad	3.2	Externo	Oportunidad
Reusabilidad	3.0	Externo	Oportunidad
Usabilidad	3.5	Externo	Oportunidad
Mantenibilidad	4.0	Externo	Oportunidad
Portabilidad	2.0	Externo	Amenaza
Documentación	2.4	Externo	Amenaza
Formación	2.8	Interno	Debilidad
Tiempo de adopción	3.5	Interno	Fortaleza
Casos de estudio de adopción FLOSS	3.0	Externo	Oportunidad
Centralidad de la tecnología de la información	2.5	Interno	Debilidad
Apoyo de la alta dirección	4.0	Interno	Fortaleza
Bloqueo de proveedores	2.0	Externo	Amenaza
Soporte	2.3	Interno	Debilidad
Actitud hacia el cambio	3.0	Interno	Fortaleza
Coste total de propiedad	3.5	Interno	Fortaleza
Ver recomendación		<b>Recomendación C:</b> La organización debe de proporcionar los recursos necesarios que garanticen una adopción satisfactoria. Si se trata de factores internos deben de ser aspectos a mejorar dentro de la organización y si son factores externos, dedicar recursos de ingeniería para mejorar el software.	

Figura 6.5: Interacción de la especialista en la interfaz de los pasos 5 y 6 de GUIOS

Las recomendaciones de GUIOS PRO mostradas en las figuras 6.5 6.6 ofrecen información genérica para cual describimos a continuación lo siguiente:

- **Recomendación A:** Adoptar el FLOSS seleccionado. La organización cumple satisfactoriamente con la mayoría de los requisitos necesarios para adoptar el software seleccionado. Los requisitos de un software son las condiciones o componentes que un sistema debe de tener para realizar ciertas actividades específicas. Se debe de entender que la organización puede o no contar con infraestructura adecuada para cumplir con estos requisitos.

En caso de que existan características del software (detalladas en la encuesta) que no cumplan con el requisito mínimo requerido, sugerimos identificar las debilidades/amenazas asociados a estos subfactores para

intentar realizar una segunda evaluación minuciosa. Esta segunda evaluación no se la realizaría en primera instancia en GUIOS PRO sino de manera manual, es decir codificando las valoraciones en una hoja de cálculo para analizar los resultados obtenidos. El objetivo de esa actividad es identificar las características más críticas que afecten una posible adopción de FLOSS y proponer a la alta gerencia una solución al problema detectado.

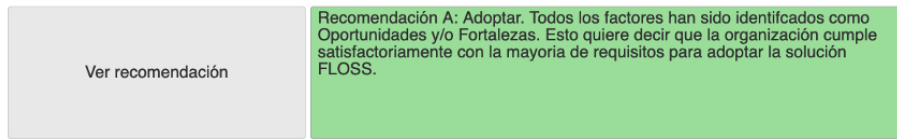
- **Recomendación B:** Es posible adoptar, con matices.

Se han detectado amenazas y debilidades en características cuya importancia relativa es opcional. Sugerimos realizar una segunda evaluación del software más específica solo con los subfactores que no cumplan con los requisitos necesarios para tener una funcionalidad del software idónea. Estos requisitos están relacionados a factores internos o externos como por ejemplo: acciones formativas orientadas al personal TIC, contratación de consultores externos, garantizar que la documentación esté disponible en múltiples formatos, que el software reduzca las dependencias de proveedores, evitar que existan costes ocultos del software, entre otros. El procedimiento de la segunda evaluación sugerida es la misma que hemos descrito en la recomendación A.

- **Recomendación C:** Se sugiere analizar la adopción y debería ser pospuesta. Las más amenazas y debilidades son mayores a las oportunidades y fortalezas. Una cantidad considerable de las características de los factores no se cumplen o se desconoce si cumplen con el mínimo requerido para poder ser adoptada por la organización. La organización debe proporcionar los recursos necesarios que garanticen una adopción satisfactoria. Si se trata de factores internos deben de ser aspectos a mejorar dentro de la organización como por ejemplo: planes de entrenamiento para empleados, personal técnico que brinde soporte 24/7/365, desarrolladores de software que sepan desarrollar adaptaciones al software que se está evaluando, usuarios finales muestran buena actitud hacia el cambio, entre otros.

Si son factores externos, dedicar recursos de ingeniería para mejorar el software como por ejemplo: que el software sea compatible con múltiples componentes de hardware, que el software sea compatible con varios sistemas operativos diferentes, que el software sea fácil de personalizar sin necesidad de modificar el código fuente, que el software sea fácil de desplegar y de probar, entre otros.

**Recomendación A:** Adoptar sin observaciones



**Recomendación B:** Es posible adoptar con sugerencias por revisar

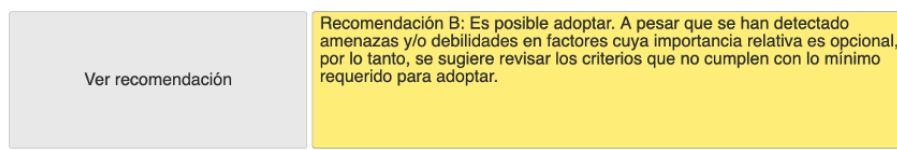


Figura 6.6: Casos alternativos de la decisión sugerida por GUIOS PRO.

- **Desarrollo de la entrevista final a la especialista.** Tal y como fue planificado, esta actividad fue desarrollada al concluir la interacción. En relación a la primera pregunta, orientada a evaluar la utilidad de GUIOS en la adopción FLOSS, la entrevistada indica que la guía le parece pertinente y de gran utilidad. Concretamente, destaca el carácter sistemático de la misma cuando afirma que:

*“Lo más importante de la guía es que organiza el proceso de adopción, el cual habría sido propenso a errores si no se contara con pasos bien definidos. La lógica que siguen los pasos tiene sentido para mí.”*

En la misma línea, la especialista comentó que durante su labor como responsable de TICs en la UNEMI, las adopciones se hacían basado principalmente siguiendo criterios muy limitados, como el costo económico del mismo y las opiniones existentes en portales de comparación de software en Internet.

En cuanto a las funcionalidades de GUIOS PRO, la especialista destacó su alto grado de correspondencia con los pasos propuestos en GUIOS. Específicamente, encontró muy fácil de usar la primera y tercera interfaz. Valoró positivamente que el proceso de evaluación sea transparente para el usuario, pues los factores, subfactores y sus respectivas ponderaciones siempre están presentes para el usuario. Esta percepción se manifiesta explícitamente cuando afirma que:

*“...incluso las ponderaciones que GUIOS pre-calcula para desarrollar la evaluación de adopción son mostradas por la herramienta. Algo que considero positivo para la comprensión del proceso de evaluación final. De hecho, cuan-*

*do obtuve los resultados finales, me resultó muy útil el poder regresar a las pantallas anteriores y comprender el porqué de estos resultados.”*

Sin embargo, mencionó que la segunda interfaz, que corresponde a la definición de las ponderaciones de los subfactores, es algo compleja. En concreto, indica que la entrada de datos pudo ser más fácil. Algo que confirma la percepción que tuvimos durante la observación de esa interacción.

Finalmente, en cuanto a las posibles funcionalidades a incluir en una versión más madura de GUIOS PRO, la especialista sugirió, además de una mejor entrada de datos para la interfaz de los pasos 3 y 4, las siguientes:

*“...posibilidad de trabajo colaborativo entre decisores. En mi experiencia, la decisión de adoptar un software no es responsabilidad de una persona en particular. Sería más adecuado que la herramienta permita que se llegue a un consenso entre varias personas involucradas en la decisión. Además, creo que se podría personalizar aún más la recomendación que GUIOS PRO brinda al final del proceso de evaluación, pues los mensajes que muestra actualmente son algo genéricos.”*

Los resultados anteriores, aunque preliminares y limitados (por el contexto, unidad de análisis y FLOSS elegidos), nos permiten concluir que, tanto GUIOS como su implantación prototipo GUIOS PRO, cumplen favorablemente con sus propósitos. En general, consideramos que: 1) efectivamente la guía resulta útil en la adopción sistemática de un FLOSS; y 2) que la herramienta desarrollada es funcional y constituye un prototipo adecuado para el desarrollo de una versión más madura. Por estas razones, se concluye que el objetivo del piloto del caso de estudio fue cumplido.

### 6.3. Resumen

En este capítulo se presentó una guía para asistir la toma de decisiones de las organizaciones en relación a la adopción FLOSS. En tal sentido, se adoptó un proceso ad-hoc descrito en el capítulo anterior y que aprovechó en gran medida los resultados obtenidos del Capítulo 4 y del Capítulo 5. Concretamente se realizó lo siguiente:

- Creación de GUIOS: se elaboró una guía con pasos a seguir para evaluar factores que repercuten en la adopción FLOSS. Esta guía tiene por objeto asistir a los responsables de administrar los recursos tecnológicos de las organizaciones en la idoneidad de la adopción del software.
- Creación de GUIOSPRO: se elaboró una aplicación prototipo multi-plataforma, caracterizada a través de un toolkit de Python llamado Flexx versión 0.80. Esta tecnología es un FLOSS distribuido bajo la licencia 2-BSD. El objetivo de GUIOSPRO fue automatizar los pasos de GUIOS para una correcta ejecución de la guía propuesta.

- Desarrollo de un piloto de caso de estudio con el objetivo de evaluar la utilidad de GUIOS en un escenario real de adopción FLOSS. Los resultados indican que la guía es útil en este proceso, y que la herramienta prototipo GUIOS PRO, es adecuada para el desarrollo de una herramienta más completa en el futuro.

## **Parte IV**

# **Observaciones finales**

## **Capítulo 7**

# **Conclusiones y trabajos futuros**

En esta tesis se presenta la necesidad de construir una guía para la evaluación FLOSS en organizaciones, para conseguirlo fue necesario utilizar una metodología que nos conduzcan a revisar sistemáticamente la literatura, así como, el diseño y aplicación de una encuesta que nos permita conocer la opinión de expertos FLOSS y valorar características de los factores identificados. El objetivo de esta tesis fue la construcción de GUIOS para automatizar el proceso de adopción FLOSS. Este capítulo presenta las conclusiones y el trabajo futuro basado en los resultados obtenidos.

## 7.1. Conclusiones

En esta tesis hemos demostrado que:

Una guía basada en procedimientos claramente definidos, con características que permitan analizar y evaluar FLOSS con un nivel de granularidad idóneo puede ofrecer un mecanismo ágil para un proceso de adopción FLOSS satisfactorio en general.

En la actualidad, el uso de software en las organizaciones es un factor primordial para automatizar los procesos y alcanzar los objetivos. La selección del software se trata de una actividad complicada si no se cuenta con un mecanismo bien definido, que oriente a los administradores de TI a la correcta selección del software a adoptar. La decisión de seleccionar software privativo y libre para adoptar en una organización, es un tema que ha estado en debate hace algunos años, debido a los beneficios que estos software puedan contribuir a las organizaciones.

El FLOSS ha tenido un crecimiento importante en los últimos 15 años y ha tenido buena aceptación en sectores públicos y privados. El FLOSS se ha convertido en una manera popular de adopción en el mercado, como una alternativa al software privativo por la flexibilidad de uso, ahorro de costes, etc., que brinda a las organizaciones, lo cual las motiva a la adopción FLOSS a gran escala.

Por tal motivo, existen propuestas metodológicas para que los involucrados en el proceso de selección de software puedan seguir pautas para una toma de decisión idónea. Sin embargo, estas propuestas no cubren todos los aspectos de evaluación en general, en el sentido que se centran en aspectos específicos tales como: el ámbito de atención sanitaria y su relación con aspectos organizacionales; infraestructura de servidores; diferentes maneras de desarrollo de software; calidad de software, etc., lo que puede dar lugar a que no se evalúen de manera más general las características relevantes del software por adoptar.

Nuestro objetivo en esta tesis ha sido proponer una guía llamada GUIOS para superar los problemas mencionados. GUIOS es una guía que cubre criterios de evaluación desde una perspectiva general y con un nivel de granularidad adecuado para un proceso de adopción satisfactorio. Estos aspectos generales los hemos denominado dimensiones y estas pueden ser: tecnológicas, organizacionales y económicas. A su vez el nivel de granularidad se conforma de factores y subfactores.

GUIOS puede ser ampliado para personalizar las dimensiones, factores y subfactores de evaluación definidas, según el contexto y la necesidad de la organización. Para soportar GUIOS, se construyó una herramienta prototipo denominada GUIOS PRO para facilitar el uso de esta guía propuesta.

## 7.2. Debate, limitaciones y extensiones

A continuación, discutimos algunas de las principales decisiones que hemos tomado en esta disertación destacando sus principales limitaciones y posibles extensio-



nes.

■ **¿Es adecuada una revisión sistemática de la literatura para identificar el estudio del arte en trabajos de investigación sobre la adopción FLOSS?**

Hemos utilizado una revisión sistemática de la literatura (RSL) con el objetivo de identificar y analizar evidencia de estudios primarios a través de un método riguroso. Sin embargo, el uso de un RSL podría presentar algún inconveniente. Por un lado, la selección adecuada de las bases de datos a utilizar puede ser una actividad propensa a errores si no se conocen bases de datos importantes para dicho propósito. Esto puede ser un factor negativo a la hora de buscar estudios primarios en revistas con un factor de impacto alto.

Por otro lado, entre los criterios de selección de búsqueda de los trabajos de investigación, consideremos excluir las conferencias, congresos, capítulos de libros y literatura gris, por lo que podríamos no contar con trabajos de investigación que contribuyan con dimensiones, factores y subfactores relevantes para esta tesis.

- *Conclusión:* si volviéramos al principio y tuviéramos que decidir si utilizar un RSL o no, volveríamos a decidir utilizarlo para nuestra disertación.
- *Extensión:* revisar nuestro proceso sistemático de la literatura y combinarlo con otra técnica de búsqueda para abordar trabajos de investigación relevantes que se pudieran quedar fuera de esta tesis.

■ **¿Las dimensiones, factores y subfactores identificados en esta tesis son suficientes para evaluar la adopción FLOSS?**

Hemos considerado dimensiones, factores y subfactores basados en la RSL existente, y hemos agrupado factores por cada dimensión según el criterio tanto de expertos de esta tesis como lo encontrado a través del RSL. Sin embargo, la identificación de las dimensiones, factores y subfactores podrían verse afectados debido a la perspectiva de este grupo de expertos y los criterios de inclusión y exclusión definidos en esta tesis.

Por ejemplo, en algunos trabajos de investigación encontramos dimensiones con otros nombres similares o que tenían una relación con la dimensión organizacional y con el objetivo de simplificar dimensiones, las agrupamos en una sola dimensión. Así mismo sucedió con la dimensión económica.

Una situación similar encontramos al identificar y clasificar los factores. Por un lado, tuvimos que agrupar algunos factores en un solo factor, debido a la cantidad de factores encontrados en la literatura y que tenían una relación muy cercana, por ejemplo en el factor compatibilidad agrupamos a: flexibilidad, complejidad, etc.

Por otro lado, el proceso de clasificación manual de los factores para asignarlos a las dimensiones podría dar lugar a errores. Una vez más, consideramos per-

tinente que el proceso de clasificación manual fuera revisado individualmente por cada uno de los autores de este estudio para minimizar el impacto.

Esto podría incidir en el tiempo de evaluación del software, debido a que le aumentaría el tiempo a la persona encargada de evaluar el software si se aumentan dimensiones, factores y subfactores, volviéndolo más complejo.

- *Conclusión:* Para esta primera versión de GUIOS, creemos que la guía cuenta con una cantidad adecuada de dimensiones, factores y subfactores. Una vez aplicada GUIOS PRO y según la retroalimentación de los decisores, se analizaría la posibilidad de aumentar o disminuir dichos elementos.
- *Extensión:* nueva revisión de la literatura así como extensión de otros trabajos empíricos como casos de estudio o experimentos para identificar nuevas dimensiones, factores y subfactores para incluirlas en una nueva versión de este trabajo de investigación.

#### ■ ¿Están todas las métricas de evaluación definidas en GUIOS?

Aunque hemos enumerado algunas métricas de evaluación en el Capítulo 5, basadas en técnicas de evaluación cualitativa y cuantitativa, existen otras maneras de calcular las métricas definidas en esta tesis.

Por ejemplo, técnicas multicriterio que tengan en cuenta la relevancia de los factores desde el punto de vista de los contextos que se han considerado. Aquí, se podría adoptar una técnica que, en lugar de agregar en una sola métrica la importancia a través de un cálculo aritmético, se apoye en un análisis alternativo como el de la eficiencia de Pareto [31]. También se podrían usar otras técnicas de evaluación cuantitativa, cualitativa y semicualitativa [43, 93].

- *Conclusión:* Extender GUIOS para que soporte otras métricas de evaluación es algo que se podría estudiar con más detalle, debido a los cálculos avanzados que esto implicaría. En este contexto, la herramienta prototipo GUIOS PRO también se vería afectada para incluir las nuevas métricas, lo que podría complicar la flexibilidad de la herramienta al momento de evaluar.
- *Extensión:* modificación de GUIOS con nuevas métricas de evaluación.

#### ■ ¿GUIOS PRO cuenta con todas las características suficientes para evaluar satisfactoriamente un FLOSS?

GUIOS PRO es una herramienta prototipo básica, diseñada sobre la base de los resultados obtenidos en esta tesis. Sin embargo, GUIOS PRO no cuenta con algunas funcionalidades que el propio evaluador las ejecute, como por ejemplo, opciones para agregar, modificar y eliminar factores y subfactores.

Esto puede ser un aspecto negativo a la hora de identificar nuevos factores y subfactores para el proceso de evaluación FLOSS, ya que le demandaría

tiempo adicional a la persona encargada en analizar y modificar GUIOS PRO, para agregar las nuevas características identificadas.

Por otro lado, GUIOS PRO se ejecuta desde un computador, tablet o dispositivo móvil de manera local, por lo tanto no está diseñado para que un usuario pueda iniciar sesión para autenticar su acceso al momento de ejecutarlo.

Esto podría ser otro punto negativo, debido a que no se contaría con información relevante histórica de las evaluaciones realizadas en un organización, esto es, no se identificaría quien realizó las anteriores evaluaciones, la fecha en que se lo realizó, ni los resultados obtenidos anteriormente.

También es importante comentar, que otra limitante que tiene la herramienta es que no ofrece una evaluación multiusuario, es decir, que varios decisores puedan evaluar el mismo FLOSS de manera colaborativa en una organización. La decisión de adoptar un software no es responsabilidad de un sola persona.

- *Conclusión:* GUIOS PRO es un prototipo con ciertas limitantes en su primera versión, pero consideramos que cuenta con características suficientes para obtener una primera evaluación satisfactoria.
- *Extensión:* agregar nueva funcionalidad a GUIOS PRO para que acepte la opción de mantenimiento de factores y subfactores. Adicionalmente, modificar GUIOS PRO para que los usuarios puedan autenticarse al iniciar sesión. Otra funcionalidad de la herramienta es que permita una evaluación multiusuario.

#### ■ **¿La aplicación de la encuesta contribuyó favorablemente para evaluar los factores identificados en este proceso?**

La encuesta aplicada al grupo de expertos contribuyó para conocer uno de los indicadores de medición, esto es, la importancia del experto, un total de 57 expertos realizaron la encuesta, la mayoría de los expertos pertenecen al sector público. En este sentido, obtener resultados de un grupo mínimo de expertos no podría ser suficiente para evaluar subfactores. Evidentemente, esto podría ser un aspecto negativo al no tener la opinión de más expertos que puedan valorar las características de los factores, agregar nuevas características o sugerir modificaciones a los factores ya identificados.

- *Conclusión:* Para esta primera versión de GUIOS pensamos que fue considerable el número de expertos que contestaron la encuesta para analizar los resultados obtenidos y seguir con nuestro proceso de construcción de los indicadores de importancia establecidos.
- *Extensión:* enviar nuevamente la solicitud de llenar la encuesta, al grupo de expertos que no pudieron aplicarla anteriormente. O también, buscar nuevo grupo de expertos en el área de FLOSS. El objetivo de esta actividad, es alcanzar una participación mayor de expertos que contribuyan con la evaluación de los subfactores.

■ **¿Las recomendaciones que sugiere GUIOS PRO le ofrece al decisor información suficiente para resolver los problemas detectados?**

GUIOS proporciona recomendaciones según la evaluación realizada por el decisor, sin embargo, la guía propuesta no le brinda al decisor información de como solucionar las amenazas y debilidades. Esto supone un aspecto negativo para el decisor que no cuenta con información detallada sobre algún procedimiento a seguir para la posible solución a realizar.

- *Conclusión:* Si bien es cierto las recomendaciones son algo genéricas pensamos que las recomendaciones que proporciona GUIOS PRO dan una idea del camino a seguir para intentar solucionar las debilidades o amenazas detectadas.
- *Extensión:* una guía que describa acciones concretas para solucionar las debilidades y amenazas de manera que el FLOSS se pueda adoptar a corto plazo.

Los resultados obtenidos en esta tesis nos permiten establecer retos a corto plazo con la intención de cubrir la mayoría de extensiones descritas en el presente capítulo. Proponer una versión más madura tanto de GUIOS como de GUIOS PRO sería conveniente desarrollar para un proceso de adopción FLOSS efectivo y óptimo en las organizaciones.

**Parte V**  
**Apéndice**

## **Apéndice A**

# **Encuesta FLOSS**

El objetivo de este capítulo es mostrar la encuesta diseñada y aplicada a los expertos para valorar la importancia de los subfactores. Los resultados obtenidos nos permiten procesar la información que nos sirven como entrada para determinar la importancia sugerida y de esta manera agilizar el proceso de adopción FLOSS. Además, los expertos tiene la opción de sugerir nuevos subfactores para mejorar de esta manera la guía propuesta.

## FLOSS Features Survey

Fields marked with \* are mandatory.

Dear colleague,

My name is Víctor Rea, I am Assistant Professor with the University of Milagro, Ecuador. I am currently pursuing the Ph.D. degree in computer engineering with the University of Seville, Spain.

We have published the work entitled: "Open source adoption factors: A systematic literature review" at IEEE ACCESS journal <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9089866> in which we analyze which factors can impact FLOSS (Free Libre Open Source Software) adoption in organizations. We are contacting you as an expert in this area of expertise.

We are currently preparing an evaluation of the selected features that determine their relevance in FLOSS adoption process. We kindly request for your feedback to evaluate these selected features through a survey. The objective of this survey is to evaluate which of these features can have impact on the FLOSS adoption and thus, be used for a tool to guide practitioners on both, public and private institutions.

We provide definitions for the categories to evaluate the features: Fundamental, Important, Optional and Irrelevant.

We estimate that this survey takes between 10 to 15 minutes to complete. We really appreciate your cooperation and we are looking forward to your inputs.

In case you have any comment, you can reach us at: [vreas@unemi.edu.ec](mailto:vreas@unemi.edu.ec)

Best regards,

Víctor

### Expert profile

In this section we want to know specific information about the respondent.

\*What type of institution do you work in?

- Public
- Private
- Independent
- Other

1

Figura A.1: Presentación de la encuesta a expertos.

En la Figura A.1 mostramos la página principal de la encuesta con una breve presentación del autor de este trabajo de investigación y el objetivo de la aplicación de la encuesta. En la siguiente sección de la encuesta *Perfil del experto*, nuestra intención es conocer información específica sobre el experto como por ejemplo: el tipo de institución en la cual trabaja. Es importante resaltar que la encuesta fue escrita en el idioma inglés debido a que los expertos eran de distintas nacionalidades.

- How many years of experience do you have in FLOSS technologies?
  - 1 to 5 years
  - 6 to 10 years
  - >10 years
  
- Pick one FLOS software for adoption at your choice to respond to this survey. A proprietary software for reference is also provided for you to consider migration from this proprietary software to the FLOSS alternative. Please, respond to this survey according to your expectations on your software selection. You do not need to be an expert on the software you have picked.
  - Office automation and management => LibreOffice vs. MS Office
  - Cloud & Network infrastructure => Openstack vs. Amazon AWS
  - Computer security => OpenVAS vs. Nessus
  - Education/Research => Moodle vs. Blackboard Learn
  - Software development => Eclipse vs. MS Visual Studio
  - Embedded Systems & IoT => OpenWRT vs. Xilinx Vitis
  - Web technologies => Wordpress vs. Weebly
  - Graphics & Design => Inkscape/Gimp vs. Photoshop
  - Telecommunications => Osmocom femtocell vs. ip.access femtocell
  - Industry & automation (SCADA, PLC) => IndigoSCADA vs. SIMATIC WinCC
  - Health technologies => GNU Health vs. WebPT
  - Finance & management => Odoo vs. SAP
  - Desktop Operating System => Ubuntu Linux vs. Windows

**Features assessment**

The following section show a list of features that were written in a positive, short and simple sentence to avoid ambiguity and misunderstandings. The questions in this survey come in no particular order.

---

**Categories definition**

- Fundamental: it is not dispensable, this feature must be complied with.
- Important: it is desirable that this feature is present, if it is not present there could be problems in the adoption of FLOSS.
- Optional: it is a feature that is not a problem if it is not present, it would not be a problem if this feature is not met.
- Irrelevant: It is not worth dedicating resources to comply with this feature, it can be obviated.

In your opinion, how relevant are the following features in your FLOSS adoption process? Please, use the FLOS software you have selected as reference to respond to this survey.

	Irrelevant	Optional	Important	Fundamental
• A company provides ready-to-use cloud infrastructure for this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Figura A.2: Inicio de la evaluación de las características de los factores, parte I.

En la Figura A.2 mostramos la continuación de la sección del *Perfil del experto*, dónde preguntamos acerca del número de años de experiencia que el experto tiene en tecnologías FLOSS. La última pregunta de la presente sección se refiere a que el experto seleccione un FLOSS como referencia para valorar las características del factor. En la siguiente sección llamada *Evaluación de las características* de la encuesta,



se muestra una lista de características que fueron escritas de manera afirmativa, corta y oraciones sencillas para evitar ambigüedades y malos entendidos. Estas características son las que el experto tiene como objetivo seleccionar una opción por única vez en cada característica. Además se ofrece una definición de las categorías: fundamental, importante, opcional e irrelevante.

* The software can export data in proprietary formats	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software interacts and integrates with existing proprietary software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is certified for operation in your niche market	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is reliable and stable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is compatible with the most common use cases and functionalities	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is compatible with multiple hardware components	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software uses standard formats	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software supports for several different operating systems (software is multiplatform)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is well documented	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Access to source code is an incentive for the organization	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software can be easily extended to meet my needs by enhancing the source code	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Innovations are introduced to the software at a rapid pace	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is easy to customize without updating the source code (easy to customize from the control panel)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software supports new functions through modules (the software is modular)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* There is a public repository of extensions for this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is easy to install and to try	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software has a good record in terms of security errors (the software is secure)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is more flexible than the proprietary solution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software is more reliable than the proprietary solution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software provides a wide variety of access control functions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The data formats are well documented	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura A.3: Lista de características de los factores, parte II.

En esta Figura A.3, se muestra la continuación de la lista de características del FLOSS a evaluar por parte de los expertos.

• The license allows for proprietary extensions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software is offered as a library/framework	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software provides a graphical user interface (GUI)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software is easier to use than the proprietary alternative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software is easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The user is displeased with the proprietary software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software is actively maintained by the developers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• A mobile application version of this software is available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software is DBMS independent	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• Adopting this software allows users to improve technical IT skills	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The organization staff can easily self-learn how to use this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The organization staff is trained to solve technological problems	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• Training plans on this software are available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The development documentation covers all features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The documentation is available in multiple formats	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The documentation is easy to understand	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The documentation is up to date	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The documentation is written by specialized writers (not developers)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software documentation is of high quality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software comes with development documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software comes with user documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The user documentation covers all features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
• The software installation and deployment requirements are easy to fulfill	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4

Figura A.4: Lista de las características de los factores, parte III.

En esta Figura A.4, se muestra la continuación de la lista de características del

FLOSS a evaluar por parte de los expertos.

* The time required to adopt this software is low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* There are public reports available on the internet describing successful adoption of this software (from similar organizations)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Adopting this software improves the working environment of users	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Centralizing your IT infrastructure helps to accelerate the adoption of this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Top management supports successful adoption of this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The IT staff in your organization endorses the adoption of this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The software reduces vendor lock-in dependencies in your environment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Community support for this software is available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* External expert and consultant support for specific queries is available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* 24/7/365 commercial support for this software is available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* There are developers in your organization that know how to develop this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Commercial support for software customizations is available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Easy to hire IT personnel in the community that is knowledgeable in this software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* The organization staff shows low resistance to technology change	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Hidden costs when adopting this software are unlikely	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* Adopting this software is less expensive than the proprietary alternative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Contact**

vreas@unemi.edu.ec

Figura A.5: Lista de las características de los factores, parte IV.

En esta Figura A.5, se muestra la última parte de la lista de características del FLOSS a evaluar por parte de los expertos. Al final de la encuesta se puede visualizar el correo electrónico del autor para que los expertos se puedan contactar y puedan

sugerir alguna modificación de las características planteadas en esta encuesta.

# Bibliografía

- [1] A. Adewumi, S. Misra, N. Omoregbe, and L.F. Sanz. Fosses: Framework for open-source software evaluation and selection. *Software - Practice and Experience*, 49(5):780–812, 2019. 36, 41, 42, 45, 46, 72
- [2] Y. Alduraywish, Y. Xu, and K. Salonitis. Evaluating state of information systems failure in developing countries using itposmo model. In *ICAC*, pages 1–5. IEEE, 2017. 72, 74, 76, 78
- [3] S. Alrabaee, P. Shirani, L. Wang, and M. Debbabi. Fossil: A resilient and efficient system for identifying foss functions in malware binaries. *ACM Transactions on Privacy and Security*, 21(2), 2018. 36, 41
- [4] E. Amoroso, C. Taylor, J. Watson, and J. Weiss. A process-oriented methodology for assessing and improving software trustworthiness. In *Proceedings of the 2nd ACM Conference on Computer and communications security*, pages 39–50. ACM, 1994. 34
- [5] D.J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010. 40
- [6] L. Aversano and M. Tortorella. Quality evaluation of floss projects: Application to erp systems. *IST*, 55(7):1260–1276, 2013. 21, 61, 65, 66, 70, 75, 76, 77, 78
- [7] L. Aversano and M. Tortorella. Quality evaluation of floss projects: Application to erp systems. *Information and Software Technology*, 55(7):1260–1276, 2013. 36, 45, 46, 72
- [8] C. Ayala, D. Cruzes, O. Hauge, and R. Conradi. Five facts on the adoption of open source software. *IEEE software*, 28(2):95–99, 2011. 61, 65, 66, 70, 75, 78, 79
- [9] C. Ayala, D. S Cruzes, X. Franch, and R. Conradi. Towards improving oss products selection-matching selectors and oss communities perspectives. In *OSS*, pages 244–258. Springer, 2011. 72, 74, 77, 78, 79

- [10] D. Badampudi, C. Wohlin, and K. Petersen. Software component decision-making: In-house, oss, cots or outsourcing—a systematic literature review. *JSS*, 121:105–124, 2016. 21, 42, 43, 49, 61, 65, 66, 70, 75, 78, 79
- [11] L. Baker. Observation: A complex research method. *Library Trends*, 55(1):171–189, 2006. 101
- [12] K. Beecher, A. Capiluppi, and C. Boldyreff. Identifying exogenous drivers and evolutionary stages in floss projects. *Journal of Systems and Software*, 82(5):739–750, 2009. 36
- [13] M. Bengtsson. How to plan and perform a qualitative study using content analysis. *NursingPlus Open*, 2:8–14, 2016. 101
- [14] A. Benlian and T. Hess. Comparing the relative importance of evaluation criteria in proprietary and open-source enterprise application software selection—a conjoint study of erp and office systems. *ISJ*, 21(6):503–525, 2011. 61, 65, 66, 70, 75, 76
- [15] F. Bolici, J. Howison, and K. Crowston. Stigmergic coordination in floss development teams: Integrating explicit and implicit mechanisms. *Cognitive Systems Research*, 38:14–22, 2016. 36, 39
- [16] C. Bouras, A. Filopoulos, V. Kokkinos, S. Michalopoulos, D. Papadopoulos, and G. Tseliou. Policy recommendations for public administrators on free and open source software usage. *Telematics and Informatics*, 31(2):237–252, 2014. 20, 34, 49, 61, 65, 66, 70, 75, 77, 78, 79
- [17] C. Bouras, V. Kokkinos, and G. Tseliou. Methodology for public administrators for selecting between open source and proprietary software. *TEIN*, 30(2):100–110, 2013. 20, 61, 65, 66, 70, 75, 76, 77, 79
- [18] G. Braught, J. Maccormick, J. Bowring, Q. Burke, B. Cutler, D. Goldschmidt, M. Krishnamoorthy, W. Turner, S. Huss-Lederman, B. Mackellar, and A. Tucker. A multi-institutional perspective on h/foss projects in the computing curriculum. *ACM Transactions on Computing Education*, 18(2), 2018. 36, 41
- [19] M. Brovelli, M. Minghini, and G. Zamboni. Public participation gis: a foss architecture enabling field-data collection. *International Journal of Digital Earth*, 8(5):345–363, 2015. 36, 39
- [20] A. Capiluppi and D. Izquierdo-Cortázar. Effort estimation of floss projects: A study of the linux kernel. *Empirical Software Engineering*, 18(1):60–88, 2013. 36

- [21] J. Choi, M. Hwang, H. Kim, and J. Ahn. What drives developing countries to select free open source software for national spatial data infrastructure? *Spatial Information Research*, 24(5):545–553, 2016. 61, 65, 66, 70
- [22] G. Conaldi. Flat for the few, steep for the many: Structural cohesion and rich-club effect as measures of hierarchy and control in floss communities. *International Journal of Open Source Software and Processes*, 2(2):14–28, 2010. 35
- [23] J. Croasmun and L. Ostrom. Using likert-type scales in the social sciences. *Journal of adult education*, 40(1):19–22, 2011. 81
- [24] K. Crowston. Future research on floss development. *First Monday*, 2(SPEC), 2005. 36
- [25] K. Crowston, Q. Li, K. Wei, U.Y. Eseryel, and J. Howison. Self-organization of teams for free/libre open source software development. *Information and Software Technology*, 49(6):564–575, 2007. 35
- [26] K. Crowston and B. Scozzi. Bug fixing practices within free/libre open source software development teams. *Journal of Database Management*, 19(2):1–30, 2008. 36
- [27] K. Crowston and I. Shamshurin. Core-periphery communication and the success of free/libre open source software projects. *Journal of Internet Services and Applications*, 8(1), 2017. 35, 40
- [28] K. Crowston, K. Wei, J. Howison, and A. Wiggins. Free/libre open-source software development: What we know and what we do not know. *ACM Computing Surveys*, 44(2), 2012. 36
- [29] R. D Macredie and K. Mijinyawa. A theory-grounded framework of open source software adoption in smes. *European Journal of Information Systems*, 20(2):237–250, 2011. 34, 61, 65, 66, 70, 75, 76, 78, 79
- [30] F. Daneshgar, G. Low, and L. Worasinchai. An investigation of ‘build vs. buy’decision for software acquisition by small to medium enterprises. *IST*, 55(10):1741–1750, 2013. 61, 65, 66, 70, 75, 77, 78
- [31] E. Davidson. *Evaluation methodology basics: The nuts and bolts of sound evaluation*. Sage, 2005. 83, 87, 113
- [32] Subsecretaria de Informática. Estrategia para la implantación de Software Libre en la Administración Pública. [https://cti.gobiernoelectronico.gob.ec/ayuda/manual/decreto\\_1014.pdf](https://cti.gobiernoelectronico.gob.ec/ayuda/manual/decreto_1014.pdf), 2009. [Online; accessed 14-May-2021]. 18, 19

- [33] V. Del Bianco, L. Lavazza, S. Morasca, and D. Taibi. A survey on open source software trustworthiness. *IEEEES*, 28(5):67–75, 2011. 61, 65, 66, 70, 75
- [34] V. Del Bianco, S. Lavazza, L. and Morasca, D. Taibi, and D. Tosi. An investigation of the users' perception of oss quality. In *OSS*, pages 15–28. Springer, 2010. 72, 74, 76
- [35] S. Dhir and S. Dhir. Adoption of open-source software versus proprietary software: An exploratory study. *Strategic Change*, 26(4):363–371, 2017. 61, 65, 70
- [36] R. Di Cosmo, D. Di Ruscio, P. Pelliccione, A. Pierantonio, and S. Zacchiroli. Supporting software evolution in component-based foss systems. *Science of Computer Programming*, 76(12):1144–1160, 2011. 36
- [37] D. Di Ruscio and P. Pelliccione. A model-driven approach to detect faults in foss systems. *Journal of Software: Evolution and Process*, 27(4):294–318, 2015. 36, 39
- [38] European Centre for the Development of Vocational Training. *Terminology of European education and training policy: a selection of 130 key terms*. Luxembourg: Publications Office of the European Union, 2 edition, 2014. 34
- [39] B. Fitzgerald. Open source software adoption: anatomy of success and failure. *International Journal of Open Source Software and Processes (IJOSSP)*, 1(1):1–23, 2009. 20, 61, 65, 70
- [40] M. Gallego, P. Luna, and S. Bueno. Designing a forecasting analysis to understand the diffusion of open source software in the year 2010. *Technological Forecasting and Social Change*, 75(5):672–686, 2008. 21, 61, 65, 66, 70, 75
- [41] G. R. Gangadharan. Open source solutions for cloud computing. *COMP*, 50(1):66–70, 2017. 49, 61, 65, 70, 75, 76, 78
- [42] G. R. Gangadharan and M. Butler. Free and open source software adoption in emerging markets: An empirical study in the education sector. In *IFIP International Conference on Open Source Systems*, pages 244–249. Springer Berlin Heidelberg, 2012. 72, 74, 75, 79
- [43] O. Gelo, D. Braakmann, and G. Benetka. Quantitative and qualitative research: Beyond the debate. *Integrative psychological and behavioral science*, 42(3):266–290, 2008. 113
- [44] G. Giaglis and D. Spinellis. Division of effort, productivity, quality, and relationships in floss virtual teams: Evidence from the freebsd project. *Journal of Universal Computer Science*, 18(19):2625–2645, 2012. 35
- [45] A. Glorioso. Floss methods in biotechnology. *First Monday*, 11(7), 2006. 36



- [46] J. Gonzalez-Barahona, G. Robles, I. Herraiz, and F. Ortega. Studying the laws of software evolution in a long-lived floss project. *Journal of Software: Evolution and Process*, 26(7):589–612, 2014. 36, 38
- [47] J. Guilford. *Psychometric methods*. McGraw-Hill series in psychology. McGraw-Hill, 1954. 81
- [48] K. Gurusamy and J. Campbell. Enablers of Open Source Software adoption: A case study of APS organisations. *Australasian Journal of Information Systems*, 17(2), 2012. 61, 65, 66, 70, 75, 77, 78, 79
- [49] A. Hanumappa, M. Dora, and V. Navik. Open source software solutions in indian libraries. *Library hi tech*, 2014. 21, 61, 65, 70, 75, 78, 79
- [50] A. Harzl. Can floss projects benefit from integrating kanban: a case study. *Journal of Internet Services and Applications*, 8(1), 2017. 36, 40
- [51] Ø. Hauge, C. Ayala, and R. Conradi. Adoption of open source software in software-intensive organizations—a systematic literature review. *Information and Software Technology*, 52(11):1133–1154, 2010. 18, 34, 49, 52, 59
- [52] Ø. Hauge, D. Cruzes, R. Conradi, K. Velle, and T. Skarpenes. Risks and risk mitigation in open source software adoption: Bridging the gap between literature and practice. *OSSNH*, pages 105–118, 2010. 49, 61, 65, 66, 70, 75, 77, 78, 79
- [53] J. Howison, M. Conklin, and K. Crowston. Flossmole: A collaborative repository for floss research data and analyses. *International Journal of Information Technology and Web Engineering (IJITWE)*, 1(3):17–26, 2006. 36
- [54] D. Huerta. *Analisis Foda o Dafo*. Bubok, 2020. 89
- [55] J.B. Huett, J.H. Sharp, and K.C. Huett. What’s all the floss? how freedom and openness are changing the face of our educational landscape. *International Journal of Open Source Software and Processes*, 2(1):1–14, 2010. 36
- [56] B. Idrizi, S. Zhaku, S. Izeiroski, I. Kabashi, and P. Nikolli. Defining methodology for selecting most appropriate gis software. *Survey review*, 46(338):383–389, 2014. 61, 65, 66, 70, 75, 78, 79
- [57] S. Jahn. Teaching open source competency. Bachelor thesis, Friedrich-Alexander University Erlangen-Nürnberg, 2014. 25, 36
- [58] S. Jlassi, A. Mammam, I. Abbassi, and M. Graiet. Towards correct cloud resource allocation in floss applications. *Future Generation Computer Systems*, 91:392–406, 2019. 36, 41

- [59] L. Joia and J. Dos Santos. From closed source to open source software: Analysis of the migration process to open office. *JHTMR*, 28(2):261–272, 2017. 61, 65, 70, 76
- [60] J. Jonathon, R. Prof Gomulkiewicz, and D. Prof Laster. Some economic & legal aspects of open source software. (Electrónico) <http://goo.gl/oFjBY>, 2005. 27
- [61] Y. Jusoh, K. Chamili, J. Yahaya, and N. Pa. The selection criteria of open source software adoption in malaysia. *International Journal of Advancements in Computing Technology*, 4(21):278–287, 2012. 61, 65, 70, 75, 76, 77, 78
- [62] P. Kamp. Quality software costs money-heartbleed was free: How to generate funding for foss. *Queue*, 12(6), 2014. 36, 37
- [63] S. Khode and S. Chandel. Adoption of open source software in india. *DESIDOC Journal of Library & Information Technology*, 35(1), 2015. 34
- [64] K. Kimppa, G. Miscione, and K. Johnston. Free and open source software in developing contexts. *Journal of Information, Communication and Ethics in Society*, 2010. 20, 61, 66, 70, 76, 79
- [65] K. Kimppa, L. Weilbach, and E. Byrne. A human environmentalist approach to diffusion in ict policies: A case study of the foss policy of the south african government. *Journal of Information, Communication and Ethics in Society*, 8(1):108–123, 2010. 34, 36
- [66] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004. 44, 49, 50
- [67] B. Kitchenham and S. Pfleeger. Principles of survey research: part 3: constructing a survey instrument. *ACM SIGSOFT Software Engineering Notes*, 27(2):20–24, 2002. 74
- [68] S. Koloniaris, G. Kousiouris, D. Anagnostopoulos, M. Nikolaidou, and K. Tserpes. Survey-based investigation, feature extraction and classification of greek municipalities maturity for open source adoption and migration prospects. *Journal of Systems and Software*, 158:110431, 2019. 61, 65, 66, 70, 75, 77, 78, 79
- [69] V. Kuechler, C. Jensen, and D. Bryant. Misconceptions and barriers to adoption of foss in the us energy industry. In *OSS*, pages 232–244. Springer, 2013. 72, 74, 75, 76, 78, 79
- [70] M. Lehman. On understanding laws, evolution, and conservation in the large-program life cycle. *J. Syst. Softw.*, 1:213–221, September 1984. 38

- [71] F. Lehmann. Floss developers as a social formation. *First Monday*, 9(11), 2004. 35
- [72] J. Li, F. Bjørnson, R. Conradi, and V. Kampenes. An empirical study of variations in cots-based software development processes in the norwegian it industry. *Empirical Software Engineering*, 11(3):433–461, 2006. 78
- [73] J. Li, R. Chen, J. Lee, and H. Rao. A case study of private–public collaboration for humanitarian free and open source disaster management software deployment. *Decision Support Systems*, 55(1):1–11, 2013. 61, 65, 66, 70, 75, 78, 79
- [74] Y. Li, Ch. Tan, and X. Yang. It is all about what we have: A discriminant analysis of organizations’ decision to adopt open source software. *DSS*, 56:56–62, 2013. 65, 70, 78
- [75] Y. Lin. The future of sociology of floss. *First Monday*, 10(SPEC. ISS. 2), 2005. 35
- [76] Y.-W. Lin. Hacker culture and the floss innovation. *International Journal of Open Source Software and Processes*, 4(3):26–37, 2012. 36
- [77] Y.-W. Lin and E. Zini. Free/libre open source software implementation in schools: Evidence from the field and implications for the future. *Computers and Education*, 50(3):1092–1102, 2008. 36
- [78] J. Lindman and I. Hammouda. Support mechanisms provided by floss foundations and other entities. *Journal of Internet Services and Applications*, 9(1), 2018. 36, 40
- [79] D. López, C. de Pablos, and R. Santos. Profiling f/oss adoption modes: An interpretive approach. In *OSS*, pages 354–360. Springer, 2010. 72, 74, 78, 79
- [80] F. Lord. On the statistical treatment of football numbers. *American Psychological Association*, 8(12):750–751, 1953. 85
- [81] B. Lundell, B. Lings, and A. Syberfeldt. Practitioner perceptions of open source software in the embedded systems area. *JSS*, 84(9):1540–1549, 2011. 61, 65, 66, 70
- [82] H. Macho, G. Robles, and J. González-Barahona. Evaluation of floss by analyzing its software evolution: An example using the moodle platform. *Journal of Information Technology Research*, 8(1):62–81, 2015. 36, 38, 45, 46
- [83] I. Maglogiannis. Towards the adoption of open source and open access electronic health record systems. *Journal of Healthcare Engineering*, 3(1):141–161, 2012. 61, 65, 66, 70, 75, 77, 78, 79

- [84] E. Maldonado. The process of introducing floss in the public administration: The case of venezuela. *Journal of the Association of Information Systems*, 11(11):756–783, 2010. 36
- [85] J. Marsan and G. Paré. Antecedents of open source software adoption in health care organizations: A qualitative survey of experts in canada. *IJMI*, 82(8):731–741, 2013. 42, 43, 49, 61, 62, 64, 65, 66, 70, 75, 78
- [86] M. Matell and J. Jacoby. Is there an optimal number of alternatives for likert scale items? study i: Reliability and validity. *Educational and psychological measurement*, 31(3):657–674, 1971. 81
- [87] O. McGrath. Data mining user activity in free and open source software (foss)/open learning management systems. *International Journal of Open Source Software and Processes*, 2(1):65–75, 2010. 36
- [88] A. Meetoo-Appavoo, A. Chutoo, P. Appavoo, and B. Durgahee. Open source ict framework for developing countries. *IEEE Technology and Society Magazine*, 32(3):39–47, 2013. 61, 65, 70
- [89] M. Michlmayr and B. Fitzgerald. Time-based release management in free and open source (foss) projects. *International Journal of Open Source Software and Processes*, 4(1):1–19, 2012. 36
- [90] M. Mijinyawa and L. Abdulwahab. An extended framework for evaluation of open source software adoption in small businesses. *Research Journal of Information Technology*, 6(4):248–269, 2014. 61, 65, 66, 70, 75, 76, 77, 78, 79
- [91] L. Montalvillo and O. Díaz. Requirement-driven evolution in software product lines: A systematic mapping study. *Journal of Systems and Software*, 122:110 – 143, 2016. 8, 50, 51, 52, 54, 55
- [92] R. Morelli and T. De Lanerolle. Foss 101: Engaging introductory students in the open source movement. *SIGCSE Bulletin Inroads*, 41(1):311–315, 2009. 36
- [93] B. Munarriz. Técnicas y métodos en investigación cualitativa. 1992. 113
- [94] G. Munoz, C. Seaman, and A. Koru. An empirical investigation into the adoption of open source software in hospitals. *DHIST*, page 175, 2010. 34, 65, 66, 70, 78
- [95] D. Nagy, A.M Yassin, and A. Bhattacharjee. Organizational adoption of open source software: barriers and remedies. *CACM*, 53(3):148–151, 2010. 61, 65, 66, 70, 78

- [96] C. Noy. Sampling knowledge: The hermeneutics of snowball sampling in qualitative research. *International Journal of social research methodology*, 11(4):327–344, 2008. 53, 72, 74
- [97] R. Padmanabha. Floss (free/libre open source software): A theme for studying cultural differences. *European Journal of Scientific Research*, 20(1):115–133, 2008. 36
- [98] M. Paschali, A. Ampatzoglou, S. Bibi, A. Chatzigeorgiou, and I. Stamelos. Reusability of open source software across domains: A case study. *JSS*, 134:211–227, 2017. 61, 70, 76
- [99] M. Patton. *Qualitative evaluation and research methods*. SAGE Publications, inc, 1990. 100
- [100] W. Pedrycz, P. Ekel, and R. Parreiras. Continuous Models of Multicriteria Decision-Making and their Analysis. In *Fuzzy Multicriteria Decision-Making*, pages 103–136. John Wiley & Sons, Ltd, Chichester, UK, 2010. 35
- [101] K. Perera and S. Weerawarna. Free and open source software technology adoption in the sri lankan banking industry. *Journal of Computer Science*, 1(1):20–34, 2013. 18, 20, 34
- [102] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *12th international conference on evaluation and assessment in software engineering*, volume 17-1, pages 1–10. sn, 2008. 8, 20, 44, 49, 50, 51, 52, 54, 55, 56
- [103] K. Petersen, S. Vakkalanka, and L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015. 52, 53, 72, 74
- [104] E. Petrinja, A. Sillitti, and G. Succi. Adoption of oss development practices by the software industry: A survey. *OSSGR*, pages 233–243, 2011. 34, 61, 65, 66, 70, 72, 74, 77, 78
- [105] E. Petrinja and G. Succi. Trustworthiness of the floss development process. *Computer Systems Science and Engineering*, 25(4):297–304, 2010. 36
- [106] P. Poba-Nzaou, S. Uwizeyemungu, and M. Saada. Critical barriers to business intelligence open source software adoption. *International Journal of Business Intelligence Research (IJBIR)*, 10(1):59–79, 2019. 61, 65, 70, 75
- [107] S. Ponelis and P. Adoma. Diffusion of open source integrated library systems in academic libraries in africa. *Library Management*, 2018. 61, 65, 66, 70, 79
- [108] G. Poo-Caamaño, E. Knauss, L. Singer, and D.M. German. Herding cats in a foss ecosystem: a tale of communication and coordination for release management. *Journal of Internet Services and Applications*, 8(1), 2017. 35, 40

- [109] W. Qu, Z. Yang, and Z. Wang. Multi-level framework of open source software adoption. *Journal of Business Research*, 64(9):997–1003, 2011. 65, 70
- [110] F. Queiroz, F. Gomes, L. De Freitas, and V. Gama. Development of a foss-based hardware-in-the-loop platform for control engineering education. *Journal of Control, Automation and Electrical Systems*, 24(3):244–252, 2013. 36
- [111] M. Rafiq. Lis community’s perceptions towards open source software adoption in libraries. *The International Information & Library Review*, 41(3):137–145, 2009. 61, 65, 66, 70, 75, 77, 78, 79
- [112] M. Rafiq and K. Ameen. Issues and lessons learned in open source software adoption in pakistani libraries. *The Electronic Library*, 2009. 61, 65, 70, 77, 78
- [113] L. Ramanathan and S. Iyer. A qualitative study on the adoption of open source software in information technology outsourcing organizations. In *OSS*, pages 103–113, 2015. 72, 74, 78
- [114] B. Ramdani and P. Kawalek. Smes & is innovations adoption: a review & assessment of previous research. *Academia. Revista Latinoamericana de Administración*, 2007. 62
- [115] V. Rea, P. Neira, J. Galindo, and D. Benavides. Open source adoption factors—a systematic literature review. *IEEE Access*, 8:94594–94609, 2020. 43, 44, 69, 72
- [116] L. Rincón, R. Mazo, and C. Salinesi. APPLIES: A framework for evaluating organization's motivation and preparation for adopting product lines. In *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE, 2018. 86, 87, 91
- [117] N. Rizk. Free and open source software (foss) as a vehicle for human development in egypt: Some evidence and insights. *International Journal of Technological Learning, Innovation and Development*, 5(3):221–236, 2012. 36
- [118] B. Rossi, B. Russo, and G. Succi. Adoption of free/libre open source software in public organizations: Factors of impact. *Information Technology and People*, 25(2):156–187, 2012. 20, 36, 44, 46, 72
- [119] B. Rossi, B. Russo, and G. Succi. Adoption of free/libre open source software in public organizations: factors of impact. *ITP*, 25(2):156–187, 2012. 58, 61, 62, 64, 65, 70, 78
- [120] Y. Roumani, J. Nwankpa, and Y. Roumani. Adopters’ trust in enterprise open source vendors: An empirical examination. *JSS*, 125:256–270, 2017. 65, 70, 78

- [121] P. Runeson, M. Host, A. Rainer, and B. Regnell. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012. 98
- [122] H. Safadi, D. Chan, M. Dawes, M. Roper, and S. Faraj. Open-source health information technology: A case study of electronic medical records. *Health Policy and Technology*, 4(1):14–28, 2015. 61, 65, 66, 70
- [123] F. Saghafi, E. Moghaddam, and A. Aslani. Examining effective factors in initial acceptance of high-tech localized technologies: Xamin, iranian localized operating system. *TFSC*, 122:275–288, 2017. 61, 65, 70, 76, 78
- [124] S. Saini, C. Krishnan, and L. Rajaram. Open source adoption index: Quantifying foss adoption by an organisation. *International Journal of Open Source Software and Processes*, 2(3):48–60, 2010. 36, 44, 46, 72
- [125] M. Sarrab and O. Rehman. Empirical study of open source software selection for adoption, based on software quality characteristics. *Advances in Engineering Software*, 69:1–11, 2014. 61, 65, 70, 76, 77, 78
- [126] C. Schweik and R. English. Tragedy of the foss commons? investigating the institutional designs of free/libre and open source software projects. *First Monday*, 12(2), 2007. 36
- [127] M. Shaikh and T. Cornford. Framing the conundrum of total cost of ownership of open source software. In *OSS*, pages 208–219. Springer, 2011. 72, 74, 79
- [128] H. Shimizu, J. Iio, and K. Hiyane. Realities of free/libre/open source software developers in japan and asia. *First Monday*, 9(11), 2004. 36
- [129] A. Silic, M. Back and D. Silic. Taxonomy of technological risks of open source software in the enterprise adoption context. *Information & Computer Security*, 2015. 61, 65, 66, 70, 75, 79
- [130] M. Silic and A. Back. Open source software adoption: Lessons from linux in munich. *ITPRO*, 19(1):42–47, 2017. 61, 65, 66, 70
- [131] P. Singh and N. Singh. Analysis of free and open source software (foss) product in web based client-server architecture. *International Journal of Open Source Software and Processes*, 9(3):36–47, 2018. 36, 40
- [132] M. Soares. Reducing transaction costs in information infrastructures using floss. *First Monday*, 9(11), 2004. 36
- [133] S. Sowe and M. McNaughton. Using multiple case studies to analyse open source software business sustainability in sub-saharan africa. In *OSS*, pages 160–177. Springer, 2012. 72, 74, 75, 76, 78, 79

- [134] P. Spagnoletti and T. Federici. Exploring the interplay between floss adoption and organizational innovation. *Communications of the association for information systems*, 29(1):15, 2011. 20, 61, 65, 66, 70, 76, 77, 78, 79
- [135] P. Spagnoletti and T. Federici. Exploring the interplay between floss adoption and organizational innovation. *Communications of the Association for Information Systems*, 29(1):279–298, 2011. 36
- [136] M. Squire. How the floss research community uses email archives. *International Journal of Open Source Software and Processes*, 4(1):37–59, 2012. 35
- [137] R. Stallman. *Free software, free society: Selected essays of Richard M. Stallman*. Lulu. com, 2002. 18
- [138] R. Stallman. *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños, 2004. 25
- [139] R. Stallman. Why open source misses the point of free software. <https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>, 2019. 25, 28, 36
- [140] I. Steinmacher, M. Graciotto, and M. Gerosa. Barriers faced by newcomers to open source projects: A systematic review. In *OSS*, pages 153–163, 2014. 72, 74, 77
- [141] S. Stevenson. Public libraries, public access computing, floss and ci: There are alternatives to private philanthropy. *First Monday*, 12(5), 2007. 36
- [142] K. Stol, A. Babar, and P. Avgeriou. The importance of architectural knowledge in integrating open source software. *OSSGR*, pages 142–158, 2011. 61, 70, 77
- [143] K. Stol and B. Fitzgerald. Inner source—adopting open source development practices in organizations: A tutorial. *IEEEES*, 32(4):60–67, 2015. 61, 65, 70, 79
- [144] R. Subramanyam and M. Xia. Free/libre open source software development in developing and developed countries: A conceptual framework with an exploratory study. *Decision Support Systems*, 46(1):173–186, 2008. 36
- [145] A. Taha. Identifying the affecting factors for adoption of open source software in it community\*“ahmed majid taha,“<sup>1</sup>laa ahmed abbood,“<sup>2</sup>theer akram abdul razzaq and\*\* as al-bahri“businesses informatics college, university of information technology and. *Journal of Engineering and Applied Sciences*, 13(14):5771–5780, 2018. 61, 65, 70



- [146] J. Teixeira, G. Robles, and J.M. González-Barahona. Lessons learned from applying social network analysis on an industrial free/libre/open source software ecosystem. *Journal of Internet Services and Applications*, 6(1), 2015. 35, 39
- [147] L. Tome, K. Johnston, A. Meadows, and M. Nyemba-Mudenda. Barriers to open source erp adoption in south africa. *The African Journal of Information Systems*, 6(2):1, 2014. 61, 65, 66, 70, 75, 76, 78, 79
- [148] R. Torkar, P. Minoves, and J. Garrigós. Adopting free/libre/open source software practices, techniques and methods for industrial use. *Journal of the Association of Information Systems*, 12(1):88–122, 2011. 36, 72
- [149] D. Tosi, L. Lavazza, S. Morasca, and M. Chiappa. Surveying the adoption of floss by public administration local organizations. In *OSS*, pages 114–123, 2015. 34, 72, 74, 75, 76, 78
- [150] J. Van Belle and M. Reed. Oss adoption in south africa: Applying the toe model to a case study. In *OSS*, pages 304–309. Springer, 2012. 34, 72, 74, 75, 78, 79
- [151] S. Van Rooij. Higher education and floss for e-learning: The role of organizational sub-cultures in enterprise-wide adoption. *International Journal of Open Source Software and Processes*, 2(1):15–31, 2010. 36
- [152] K. Ven and J. Verelst. A field study on the barriers in the assimilation of open source server software. In *OSS*, pages 281–293. Springer, 2010. 72, 74, 78, 79
- [153] K. Ven and J. Verelst. An empirical investigation into the assimilation of open source server software. *Communications of the association for information systems*, 28(1):9, 2011. 61, 65, 66, 70
- [154] K. Ven and J. Verelst. A qualitative study on the organizational adoption of open source server software. *Information systems management*, 29(3):170–187, 2012. 43, 44, 61, 64, 65, 66, 70
- [155] K. Ven, J. Verelst, and H. Mannaert. Should you adopt open source software? *IEEE software*, 25(3):54–59, 2008. 61, 65, 66, 70
- [156] I. Voras, B. Mihaljević, and M. Orlić. Criteria for evaluation of open source cloud computing solutions. In *ITI*, pages 137–142. IEEE, 2011. 72, 74, 78
- [157] S. Weber. *The success of open source*, volume 368. Cambridge Univ Press, 2004. 34
- [158] K. Wei, K. Crowston, U. Eseryel, and R. Heckman. Roles and politeness behavior in community-based free/libre open source software development. *Information and Management*, 54(5):573–582, 2017. 36, 40

- [159] K. Wei, K. Crowston, N.L. Li, and R. Heckman. Understanding group maintenance behavior in free/libre open-source software projects: The case of fire and gaim. *Information and Management*, 51(3):297–309, 2014. 36, 38
- [160] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006. 56
- [161] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012. 98
- [162] H. Wright, M. Kim, and D. Perry. Validity concerns in software engineering research. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 411–414. ACM, 2010. 57, 91
- [163] P. Wright, P. Lichtenfels, and E. Pursell. The structured interview: Additional studies and a meta-analysis. *Journal of Occupational Psychology*, 62(3):191–199, sep 1989. 101
- [164] H. Wu and S. Leung. Can likert scales be treated as interval scales?—a simulation study. *Journal of Social Service Research*, 43(4):527–532, 2017. 85
- [165] M. Zaffar, R. Kumar, and K. Zhao. Diffusion dynamics of open source software: An agent-based computational economics (ace) approach. *Decision support systems*, 51(3):597–608, 2011. 66, 70
- [166] S. Zhang, M. Xiao, and F. Liu. An enhanced microkernel for the design of location based services (lbs) using free open source software (foss). *International Journal of Simulation: Systems, Science and Technology*, 17(44):20.1–20.4, 2016. 36
- [167] M. Zhou and A. Mockus. Who will stay in the floss community? modeling participant’s initial behavior. *IEEE Transactions on Software Engineering*, 41(1):82–99, 2015. 35, 39