



FACULTAD 1

CENTRO DE SOFTWARE LIBRE

TRABAJO FINAL PRESENTADO EN OPCIÓN AL TÍTULO DE MÁSTER DE CALIDAD DE  
SOFTWARE

---

**ESTRATEGIA DE SELECCIÓN DE ALTERNATIVAS LIBRES PARA EL  
PROCESO DE MIGRACIÓN A CÓDIGO ABIERTO**

---

**AUTORA**

Ing. María Leisy González Carrera

**TUTORES**

Dr.C Yaimí Trujillo Casañola

MS.c Yoandy Pérez Villazón

La Habana, Julio de 2018

“Año 60 de la Revolución”

## **DECLARACIÓN JURADA DE AUTORÍA**

Declaro por este medio que yo **María Leisy González Carrera**, con carné de identidad **90112010975**, soy la autora principal del trabajo final de maestría “**Estrategia de selección de alternativas libres para el proceso de migración a código abierto**”, desarrollada como parte de la 4<sup>ta</sup> Edición de la **Maestría de Calidad de Software** y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los \_\_\_\_ días del mes de julio del año 2018.

---

Ing. María Leisy González Carrera

---

Dr.C Yaimí Trujillo Casañola

---

MS.c. Yoandy Pérez Villazón

**Dedicatoria**

*A mis padres por ser mis guías cada día y alentarme para lograr mis objetivos.*

*A mi abuela querida por sus sabios consejos y a mi hermana por apoyarme siempre incondicionalmente.*

*De manera general, dedico este resultado a mi gran familia por ser tan unidad y estar al tanto de mi.*

*En especial, a mi tía Noelia por ser un gran ejemplo de lucha y perseverancia para mi.*

**Agradecimientos**

*A toda mi familia por apoyarme siempre y sobre todo en mi superación profesional.*

*A Cesar por estar conmigo en los momentos buenos y malos en estos últimos 5 años y por brindarme sus criterios y experiencias en la presente investigación.*

*A todos mis compañeros de trabajo del centro CESOL, que de una forma u otra colaboraron a este gran día, y en especial a Dayli, Nurisel, Gladys, Marielis, Yadiel, Yasiel, Yosel y Gustavo por los momentos de alegría y de apoyo.*

*A mis amigas Adalgisa, Madeleinis, Anisleidys y Leanni por el apoyo incondicional.*

*A mis tutores por apoyarme durante el desarrollo de la maestría.*

*A mis profesores y compañeros de los cursos de la Maestría y al comité científico de la Facultad1.*

*A mi oponente y al presente tribunal por formar parte de este día.*

## **RESUMEN**

El libro “Buenas Prácticas para la Migración a Código Abierto” no define cómo realizar la selección de alternativas libres, solo plantea elementos a grandes rasgos que son necesarios para la selección. El impacto de una mala selección causa disminución de la productividad de los usuarios en las organizaciones, y rechazo y desmotivación del personal con el proceso de migración. La presente investigación propone una estrategia encaminada en apoyar a los especialistas en migración para seleccionar las alternativas libres más adecuadas en un proceso de migración a código abierto de las estaciones de trabajo, con el propósito de satisfacer las necesidades de los usuarios finales en las organizaciones. Ésta se basa en atributos propuestos por modelos de evaluación de software libre y de código abierto, y criterios establecidos por los especialistas en migración para la obtención de los atributos genéricos de la alternativa libre, en la Norma ISO/IEC 2502n (25022-25023) para la adquisición de las métricas específicas de la alternativa libre y en el área Análisis de Decisiones y Resolución de Integración de Modelos de Capacidad y Madurez nivel 3 para la definición de las actividades durante un proceso de evaluación formal. La estrategia se valida a través de la aplicación del criterio de expertos en su variante Delphi, lo que permitió evaluar de muy adecuado el nivel de influencia y su aplicabilidad. Finalmente, con la aplicación de la estrategia se corrobora que los usuarios del Ministerio de Comunicaciones están satisfechos con las alternativas libres seleccionadas, pero como se demuestra en el estudio de caso fue necesario la instalación de las dos alternativas LibreOffice Writer y WPS Office Free como paquete ofimático.

**Palabras clave:** atributos, estrategia, migración, selección

## **ABSTRACT**

The book "Good Practices for Open Source Migration" does not define how to make the selection of free alternatives, only raises elements that are necessary for the selection. The impact of poor selection causes decreased productivity of users in organizations, and rejection and demotivation of staff with the migration process. This research proposes a strategy aimed at supporting migration specialists to select the most appropriate free alternatives in an open source migration process of the workstations, with the purpose of satisfying the needs of the end users in the organizations. This is based on attributes proposed by free software and open source evaluation models, and criteria established by migration specialists to obtain the generic attributes of the free alternative, in ISO/IEC 2502n (25022-25023) for the acquisition of the specific metrics of the free alternative and in the area Decision Analysis and Resolution of Capacity Maturity Model Integration level 3 for the definition of the activities during a formal evaluation process. The strategy was validated through the application of the expert criterion in its Delphi variant, which made it possible to evaluate the level of influence and its applicability very well. Finally, with the application of the strategy it was corroborated that the users of the Ministry of Communications are satisfied with the selected free alternatives, but as it was demonstrated in the study of case, it was necessary to install the two alternatives LibreOffice Writer and WPS Office Free as a package office.

**Keywords:** attributes, migration, selection, strategy

## ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	I
CAPÍTULO 1. MARCO TEÓRICO DE LA INVESTIGACIÓN.....	9
1.1 Software Libre y Código Abierto (FLOSS).....	9
1.2 Migración a código abierto en Cuba.....	10
1.2.1 Guía Cubana y Libro de Buenas Prácticas para Migración a Código Abierto.....	11
1.3 Relación de las etapas AUP-UCI y las etapas de la metodología de Migración.....	14
1.4 Análisis de decisiones.....	16
1.5 Modelos FLOSS.....	19
1.5.1 Análisis de los modelos FLOSS.....	31
CAPÍTULO 2. ESTRATEGIA DE SELECCIÓN DE ALTERNATIVAS LIBRES.....	34
2.1 Diagnóstico del proceso actual.....	34
2.2 Estrategia de selección de alternativas libres.....	35
2.2.1 Etapa 1. Definición.....	40
2.2.2 Etapa 2. Identificación.....	54
2.2.3 Etapa 3. Evaluación.....	55
2.2.4 Etapa 4. Selección.....	61
CAPÍTULO 3. VALIDACIÓN DE LA ESTRATEGIA.....	63
3.1 Validación de la estrategia a través del criterio de expertos.....	63
3.1.1 Valoración de la estrategia de selección por los expertos seleccionados.....	65
3.2 Resultados del cuestionario de satisfacción de los usuarios en el MINCOM.....	68
3.3 Estudio de caso: Alternativa libre para la herramienta privativa Microsoft Office Word.....	70
3.3.1 Definición.....	71
3.3.2 Identificación.....	72
3.3.3 Evaluación.....	72
3.3.4 Selección.....	74
CONCLUSIONES GENERALES.....	76
RECOMENDACIONES.....	77
REFERENCIAS BIBLIOGRÁFICAS.....	78

ANEXOS.....	82
Anexo 1: Listado de modelos FLOSS.....	82
Anexo 2: Modelos FLOSS vs Criterios.....	83
Anexo 3: Modelos FLOSS vs Métricas.....	84
Anexo 4: Modelos por atributos (tabla 1).....	85
Anexo 5: Modelos por atributos (tabla 2).....	87
Anexo 6: Guión de la entrevista aplicada a los especialistas en migración del centro CESOL.....	88
Anexo 7: Principales elementos de las respuestas a la entrevista aplicada a los especialistas en migración del centro CESOL.....	88
Anexo 8: Licencias más populares en aplicaciones de código abierto.....	95
Anexo 9: Encuesta para determinar nivel de competencia de los expertos.....	95
Anexo 10: Tablas del método Delphi.....	97
Anexo 11: Encuesta criterios de los expertos.....	100
Anexo 12: Cuestionario aplicado para determinar nivel de satisfacción.....	101
Anexo 13: Tarjetas de identificación de LibreOffice Writer, Abiword y WPS Office Free.....	103
Anexo 14: Acta de aceptación de la Migración en el MINCOM.....	105

## **INTRODUCCIÓN**

Las empresas desean entregar a sus clientes mejores productos y servicios en menos tiempo y con menos costes. Al mismo tiempo, se requiere de productos y servicios cada vez más complejos en los entornos de alta tecnología. Las organizaciones deben ser capaces de gestionar y controlar el complejo proceso de desarrollo y mantenimiento en dichos entornos. Los problemas que estas presentan implican soluciones que conciernen a toda la empresa y requieren un enfoque integrado. La gestión eficaz de los activos de la organización es crítica para el éxito de su actividad. En esencia, estas organizaciones son desarrolladoras de productos y servicios que necesitan una manera de gestionar sus actividades de desarrollo como parte de la consecución de sus objetivos de negocio. En el mercado actual existen modelos de madurez, estándares, metodologías y guías que pueden ayudar a una organización a mejorar la forma de hacer su negocio (CMMI-DEV, 2010).

El enfoque del modelo de calidad es uno de los dos enfoques principales para comprender la calidad de los productos de software. Algunos modelos de calidad se centran alrededor de un conjunto de atributos / métricas utilizados para evaluar de forma distinta la calidad haciendo de la calidad un concepto cuantificable. Ejemplo es el modelo de McCall (1977), el modelo Boehm (1976, 1978) y el estándar de calidad del producto ISO/IEC 9126: 2001. Debido a que estos modelos de calidad ignoraron algunos atributos (como la comunidad) únicos para el software de código abierto, en 2003 surgieron nuevos modelos de calidad específicos para software de código abierto (Adewumi y otros, 2013).

La demanda y la necesidad de que las compañías encuentren una manera de evaluar las diferentes soluciones de software libre y código abierto (FLOSS, siglas en inglés) en relación con su funcionalidad y los estándares de calidad esperados, está creciendo constantemente como resultado de la amplia y creciente gama de soluciones de software libre disponibles. Además de evaluar el vencimiento de un producto FLOSS, la intención de un Modelo de Madurez de Código Abierto (OSMM, siglas en inglés) también considera una comparación del software con alternativas comerciales, así como su correspondencia con negocios específicos, pero especialmente con los requisitos de Tecnologías de Información (Gorhan y otros, 2012).

Instituciones internacionales han formulado modelos para la evaluación y selección de software de código abierto, la mayoría de ellas se centran en la madurez, durabilidad y la estrategia de organización del desarrollo, otras están focalizadas en los aspectos funcionales (Glott y otros, 2010) (Soto y otros, 2009).

Oficialmente, los primeros modelos de calidad y madurez de FLOSS surgieron entre los años 2003 y 2005 (Gorhan y otros, 2012). Los modelos para evaluar aplicaciones de código abierto se pueden dividir en dos generaciones, la diferencia de los descritos como de “segunda generación” es que incluyen herramientas de soporte para automatizar al menos parcialmente el proceso de evaluación (Glott y otros, 2010) (Soto y otros, 2009).

Algunos de estos modelos son: Modelo de madurez de código abierto (OSMM) de Capgemini (2003), Modelo de madurez de código abierto (OSMM) de Navica (2004), Metodología de calificación y selección de software de código abierto (QSOS) (2004), Calificación de preparación de negocios abiertos (OpenBRR) (2005), Calificación de calidad de negocios abiertos (OpenBQR) (2007), QualiPSo Modelo de madurez de código abierto (OMM) (2008), entre otros (Petrinja y otros, 2010).

Sin embargo dichos modelos no tienen en cuenta condiciones tecnológicas propias de Cuba, como son computadoras de baja<sup>1</sup> y media gama. Además, los atributos de calidad que proponen deben ser revisados, ajustados y/o cambiados para una mejor aplicación en las instituciones cubanas.

Cuba está consciente de que, una sociedad para ser más eficaz, eficiente y competitiva, debe aplicar la informatización en todas sus esferas y procesos, y convencida de que para los países subdesarrollados resulta imprescindible el logro de este propósito, ya que su objetivo fundamental es lograr la supervivencia de sus pueblos. En este sentido, Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las tecnologías de la información y las comunicaciones y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a la sociedad acercarse más hacia el objetivo de un desarrollo sostenible (González y otros, 2007).

La informatización de los procesos de la sociedad cubana se ha sustentado en el uso de software privativo, proveniente en mayor medida de empresas como Microsoft. El bloqueo exime a Cuba del pago obligatorio de productos de software norteamericanos empleados en el territorio nacional, a la vez que la ausencia de un mecanismo legal que muestre obligatoriedad hacia el uso de plataformas libres permite la libertad a las instituciones de usar cualquier variante de software, independiente del modo de licenciamiento. Ejemplo de ello es la expansión en la isla del sistema operativo Microsoft Windows, presente en escuelas, industrias, oficinas y restantes sectores (Pérez, 2015).

Por ello, a partir del 2004 el país decide emprender la migración a plataformas de código abierto. Para el

---

1 Computadoras con RAM entre 256 y 1024 MB.

cumplimiento de esta misión se designó al Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI) como ente para definir las directrices de migración nacional, desarrollar la distribución cubana de GNU/Linux y asesorar a organizaciones durante sus procesos de migración a código abierto (Pérez y otros, 2014).

Durante la ejecución del proceso de migración a código abierto, en las organizaciones, intervienen dos componentes esenciales: migración social y técnica. La migración social está asociada al grado de aceptación/compromiso que logren alcanzar los usuarios con el proceso ante cambio tecnológico. Si los usuarios no están convencidos de la importancia del proceso es muy probable que este fracase. La migración técnica (o tecnológica) es el cambio de la tecnología en la organización. Esta requiere como paso imprescindible para su cumplimiento el cambio de las aplicaciones privativas de la entidad por herramientas libres en el mayor porcentaje posible. Para ello, se debe tener en cuenta que la migración puede que se realice de manera parcial, debido al uso de algunas aplicaciones privativas que no posean una alternativa libre a corto plazo. Además, la distribución empleada en el cambio tecnológico es GNU/Linux Nova (Pérez, 2015).

GNU/Linux Nova es la distribución cubana, que utiliza el núcleo Linux e incluye determinados paquetes de aplicaciones informáticas para satisfacer las necesidades de la migración a plataformas de código abierto, que experimenta Cuba como parte del proceso de informatización de la sociedad. La distribución está constituida por Nova Escritorio, Nova Ligero y Nova Servidores. Nova Escritorio es el proyecto donde se desarrollan las aplicaciones que componen el escritorio del sistema operativo Nova. Trata los temas referentes a los entornos de escritorio para estaciones de trabajo con prestaciones relativamente modernas. Nova Ligero es el proyecto donde se desarrollan los componentes de un escritorio alternativo de la distribución Nova Escritorio. Trata los temas referentes a los entornos de escritorio para estaciones de trabajo con bajas prestaciones, así como métodos de optimización para sistemas operativos mínimos, minimalistas y ligeros. Nova Servidores es el proyecto donde se desarrollan e integran las aplicaciones necesarias para brindar servicios telemáticos confiables y administrarlos (Pierra, 2011).

Especialistas del centro CESOL desarrollaron la “Guía Cubana para la Migración a Código Abierto” en el año 2009, y en el año 2016 se publicó el libro “Buenas Prácticas para la Migración a Código Abierto”. De forma general en la guía se definen las etapas, procesos y actividades que se deben seguir para la migración en las organizaciones hacia código abierto. Pérez y otros autores (2013) definen las etapas: Preparación, donde se ejecutan todas las tareas de diagnóstico de los procesos, personas y tecnologías

de la entidad, se realizan tareas de análisis de la información recuperada y se emite el plan de migración institucional; Ejecución, comprende las actividades necesarias para la migración definitiva de usuarios y tecnologías de la organización, incluye la migración de los servicios telemáticos y las computadoras de escritorio y, finalmente, Consolidación es la etapa que comprende tareas destinadas a garantizar el soporte técnico a los usuarios e infraestructura.

En el libro se describen las buenas prácticas a seguir para migrar una organización a software de código abierto, es una versión mejorada de guía basada en la experiencia de los especialistas en migración. Durante la etapa Preparación una de las actividades es “realizar inventario de hardware y software” donde se realiza un diagnóstico de las estaciones de trabajo, con el fin obtener la información referente a las aplicaciones informáticas instaladas y utilizadas por de los usuarios. Luego del diagnóstico inicial realizado en etapa Preparación se obtiene un plan de migración con el propósito de ejecutar la migración a código abierto cuando la organización estime conveniente.

Sin embargo, no se encuentra definido cómo realizar la selección de alternativas libres, solo se plantea que se debe evaluar el software de escritorio teniendo en cuenta elementos puntuales (etapa Preparación), además que es fundamental que estas sean instaladas en el entorno privativo (etapa Ejecución), y otros elementos a grandes rasgos. No se tienen en cuenta las diferentes actividades que distinguen la selección de alternativas libres ni se definen atributos y métricas a tener en cuenta en la evaluación comparativa de las alternativas libres para seleccionar la aplicación adecuada. Es válido aclarar que el libro de buenas prácticas propone un grupo de aplicaciones que son comunes o más usadas por los usuarios en las instituciones.

Las entrevistas realizadas a los especialistas en migración del centro CESOL, para determinar cómo era ejecutada la evaluación del software de escritorio, permitieron identificar que éstos realizan la selección basados en la experiencia individual, haciéndolo de una manera diferente debido a que todos no poseen la misma experiencia.

A pesar de contar con el libro de Buenas Prácticas para Migración a Código Abierto y de haber definido a la migración social como centro, todavía existen deficiencias tales como:

- No está definido cómo debe realizarse la selección de alternativas libres en el proceso de migración.
- Las aplicaciones que se proponen como soluciones alternativas al software privativo en las

instituciones cubanas no presentan en todos los casos una adecuada correspondencia con las necesidades de la organización.

- Los usuarios han mostrado insatisfacción con las herramientas disponibles en la distribución GNU/Linux Nova Escritorio y Ligerio.
- Existe carencia de habilidades de los usuarios en el uso de tecnologías de software libre y código abierto.
- Las aplicaciones propuestas en el libro de Buenas Prácticas para Migración a Código Abierto se encuentran desactualizadas.
- Las aplicaciones alternativas al software privativo no requieren de todas las funcionalidades de este.

El impacto de una mala selección causa disminución de la productividad de los usuarios en las organizaciones, rechazo y desmotivación del personal al trabajo, desconfianza y resistencia al cambio en las personas involucradas durante el proceso, por consiguiente aumento considerable de las posibilidades de fracaso de la migración.

A partir de lo anterior se plantea como **problema científico**: ¿Cómo incrementar la satisfacción de los usuarios durante el proceso de migración a código abierto de las estaciones de trabajo?

Se deriva como **objetivo general**: desarrollar una estrategia de selección de alternativas libres en el proceso de migración a código abierto de las estaciones de trabajo, de forma que se incremente la satisfacción de los usuarios.

El **objeto de estudio** de la investigación es el proceso de migración a código abierto de las estaciones de trabajo, enmarcado en el **campo de acción** la selección de alternativas libres en el proceso de migración a código abierto de las estaciones de trabajo.

A partir del objetivo general propuesto y en aras de darle cumplimiento al mismo se derivan los siguientes **objetivos específicos**:

1. Construir el marco teórico de la investigación relacionado con la selección de alternativas libres en las estaciones de trabajo dentro del proceso de migración a código abierto.
2. Diagnosticar el estado actual de la selección de alternativas libres en las estaciones de trabajo durante el proceso de migración a código abierto.

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

3. Fundamentar la estrategia de selección de alternativas libres en las estaciones de trabajo para el proceso de migración a código abierto.
4. Validar la estrategia de selección de alternativas libres en las estaciones de trabajo para el proceso de migración a código abierto.

Se plantea como **hipótesis** que una estrategia de selección de alternativas libres en el proceso de migración a código abierto de las estaciones de trabajo, contribuye a incrementar la satisfacción de los usuarios con dicho proceso. Como se puede apreciar en la Tabla 1 se identifica como **variable independiente** la estrategia de selección de alternativas libres en las estaciones de trabajo y como variable **dependiente** la satisfacción de los usuarios.

**Tabla 1. Operacionalización de las variables.** *Elaboración propia*

Variable		Dimensión	Indicadores	Unidad de medida
<b>Independiente</b>	Estrategia de selección de alternativas libres en las estaciones de trabajo	Aplicabilidad en el proceso de migración a código abierto	Nivel de aplicabilidad	Muy adecuado Bastante adecuado Adecuado Poco adecuado Inadecuado
		Influencia en el proceso de migración a código abierto	Nivel de influencia	Muy adecuado Bastante adecuado Adecuado Poco adecuado Inadecuado
<b>Dependiente</b>	Satisfacción de los usuarios	Social	Grado de satisfacción de los usuarios	Muy satisfecho Bastante satisfecho Satisfecho Poco satisfecho Insatisfecho

Para el desarrollo de la investigación fue necesario hacer uso de un conjunto de procedimientos que permitieron obtener información no observable directamente. Fueron empleados los siguientes **métodos**

**científicos y técnicas:**

**Analítico-Sintético:** la aplicación de este método permitió la extracción de los elementos más importantes relacionados con el proceso de migración a código abierto de las estaciones de trabajo. Se analizaron teorías, documentos, bibliografías relacionadas con la selección de alternativas libres dentro del proceso de migración y a partir de esto se concretaron resultados que contribuyen a una mejor estructuración del modo de ejecución del proceso de migración a código abierto.

**Análisis Histórico-Lógico:** permitió constatar teóricamente cómo ha evolucionado el proceso de migración a código abierto en toda su trayectoria lo que permitió inferir elementos para proponer una mejor solución.

**Observación:** desde un punto de vista contemplativo, la aplicación de esta técnica permitió observar el curso de la ejecución de los procesos de migración que se desarrollaban.

**Entrevista:** aplicada a los especialistas en migración a código abierto del centro CESOL para evaluar y constatar cómo se realizaba el proceso de selección de alternativas libres durante la migración a código abierto.

**Encuesta:** utilizada en la validación de la estrategia durante la aplicación del método Delphi, aplicada en la selección de los expertos y recopilación de sus criterios; y un cuestionario aplicado a los usuarios del Ministerio de Comunicaciones (MINCOM) para determinar su grado de satisfacción con las alternativas libres durante el proceso de migración realizado.

Para mostrar el desarrollo de la investigación y los resultados, el trabajo se ha estructurado en tres capítulos, además de las Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos.

En el primer capítulo “**Marco teórico de la investigación**” se realiza un estudio del estado del arte referente a la selección de alternativas libres, modelos para el software libre y de código abierto. Se aborda el proceso de migración a código abierto en Cuba, la Guía cubana y el Libro de Buenas Prácticas para Migración a Código Abierto.

En el segundo capítulo “**Estrategia de selección de alternativas libres**” se fundamenta la estrategia de selección de alternativas libres a aplicar en los procesos de migración a código abierto y se proponen etapas, actividades y artefactos.

En el tercer capítulo “**Validación de la estrategia**” se realiza la validación de la estrategia de selección de

### *Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

alternativas libres para el proceso de migración a código abierto, a través de la aplicación del criterio de expertos en su variante Delphi para determinar la aplicabilidad y nivel de influencia de la estrategia. Se realiza un estudio de caso para seleccionar la alternativa más adecuada a la herramienta privativa Microsoft Office Word.

## **CAPÍTULO 1. MARCO TEÓRICO DE LA INVESTIGACIÓN**

La selección de alternativas libres es una de las actividades fundamentales en desarrollarse durante un proceso de migración a código abierto. El éxito del proceso de migración depende, en gran medida, de la realización de una correcta selección de alternativas libres. En el presente capítulo se abordan los conceptos y temas relacionados con software libre y código abierto. Se analizan aspectos relacionados con el proceso de migración a código abierto en Cuba y las actividades del área *Análisis de Decisiones y Resolución* de CMMI nivel 3. Además, se realiza una descripción y comparación de modelos de madurez orientados a FLOSS.

### **1.1 Software Libre y Código Abierto (FLOSS)**

A nivel mundial los sistemas basados en aplicaciones libres tienen una gran importancia en diferentes áreas, ya sea gobierno, educación, empresas privadas, entre otros, porque se cuenta con diferentes aplicaciones usadas diariamente como son los sistemas operativos, correos electrónicos, programas ofimáticos, programas educativos, telefonía (Martínez, 2017).

"Software libre" significa software que respeta la libertad y la comunidad de los usuarios. En términos generales, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Así, el "software libre" es una cuestión de libertad, no de precio. A veces se llama "software libre", tomando prestado la palabra francesa o española para "libre" como en libertad, para mostrar que no se quiere decir que el software es gratis (Free Software Foundation, 2017).

"Código abierto" se refiere a que las líneas de texto en las cuales están escritas las instrucciones que sigue la computadora para ejecutar dicho programa, están disponibles para que todos los usuarios puedan conocerlas y así, mejorar o adaptar dicho código según sus necesidades (Soria, 2017). El software de código abierto presenta diferentes tipos de licencia, tales como: GPL, LGPL, BSD y MIT (Akbari, 2014).

"Proceso de migración" se refiere al conjunto de subprocesos y tareas que son necesarias realizar para llevar a una institución desde un estado inicial donde existe software privativo hacia un estado final donde existan mayoritariamente aplicaciones libres, teniendo en cuenta para ello la migración social y tecnológica (Pérez, 2015).

"Migración a tecnologías de software libre y código abierto" puede definirse como un proceso ordenado en

el cual se sustituye, parcial o totalmente, el software existente en la organización por alternativas liberadas bajo licencias libres o de código abierto. Es recomendado incluir la adopción de estándares abiertos para la documentación (Pérez y otros, 2014).

El FLOSS es producido tanto por desarrolladores individuales en la comunidad, como por organismos y dependencias públicas. Las empresas pueden obtener ingresos tanto por la vía de la venta y distribución de software FLOS (de terceros o desarrollados internamente, a medida o enlatado), como por servicios a partir de software FLOS (de terceros o desarrollados internamente) (Morero y otros, 2016).

## **1.2 Migración a código abierto en Cuba**

Como fue abordado en la introducción de la presente investigación, a partir del año 2004 Cuba decidió adoptar gubernamentalmente la migración a software libre. Por ende, se ha realizado un conjunto de acciones en aras de garantizar lineamientos que sirvan a los Organismos Centrales del Estado (OACE) para guiar su migración. Posteriormente, surge la Metodología Cubana para la Migración a Código Abierto (Pérez, 2008). Dicha metodología viene contenida dentro de la Guía Cubana para la Migración a Código Abierto y esta a su vez ha sido mejorada en el libro de Buenas Prácticas para Migración a Código Abierto. Sin embargo como resultado de una investigación llevada por un conjunto de autores se procedió a una reestructuración metodológica del proceso de migración en Cuba. Se proponen tres etapas en la guía (Pérez y otros, 2013):

### ***Preparación***

Esta etapa incluye el diagnóstico a las personas, tecnología e institución. El correcto diagnóstico a las personas, las tecnologías que usan y la información de los procesos institucionales, permite tener una clara visión de la entidad, elemento clave para proyectar una acertada estrategia de migración. Por tanto es necesario obtener por cada elemento: Personas, Nivel de conocimiento asociado al software libre y la migración, Nivel de aceptación o rechazo del software de código abierto, Niveles de capacitación, preparación y disposición para asumir la nueva tecnología, Tecnología, Información de las aplicaciones informáticas empleadas y sus versiones, Información del hardware y los periféricos, Institución (misión, visión, objetivos estratégicos, políticas y procesos institucionales).

Posteriormente se debe realizar el análisis de la información. Luego de recopilar la información de la entidad, se procede a procesar la misma para elaborar un acertado plan de migración. Las actividades fundamentales a realizar son: selección de alternativas libres más adecuadas, certificar el hardware

detectado, elaborar una estrategia de sensibilización por niveles de usuario, identificando prioridades y elementos clave que permitan una concientización de los involucrados, elaborar una estrategia de capacitación que permita formar a los usuarios en las aplicaciones libres que se instalan y una estrategia de soporte técnico que garantice tras su implementación la asistencia a los usuarios e infraestructuras.

### **Ejecución**

El proceso de ejecución debe estar guiado estrictamente (a menos que haya cambios en el tiempo) por lo descrito en el plan de migración elaborado. Por cuestiones asociadas a un mayor grado de aceptación de los administradores de red y especialistas de soporte, y en aras de lograr una compatibilidad acertada entre los sistemas, se propone un determinado nivel de solapamiento durante la migración de los servicios telemáticos y la instalación de software de fuentes abiertas sobre el sistema operativo privativo.

### **Consolidación**

Esta constituye la última etapa propuesta e incluye el soporte a los sistemas, las respuestas a las inquietudes de los usuarios y el monitoreo constante del ambiente migrado.

Actualmente en Cuba, los especialistas del centro CESOL se rigen por el libro de Buenas Prácticas para Migración a Código Abierto para ejecutar la migración a código abierto en organizaciones, dicho documento dirige el proceso nacional de migración a código abierto.

#### **1.2.1 Guía Cubana y Libro de Buenas Prácticas para Migración a Código Abierto**

Siguiendo los objetivos declarados para el Grupo Técnico Nacional durante el 2009 es lanzada la Guía Cubana para la Migración a Código Abierto. La guía aborda contenidos relacionados con el software libre como son las definiciones de código abierto, GNU, Linux y distribución. Se definen las diversas categorías de software existentes, tipos de licencias y de estándares informáticos (Pérez, 2009).

De forma general la Guía Cubana para la Migración a Código Abierto se encuentra estructurada de la siguiente forma: Metodología de migración, Preparación, Migración parcial, Migración total y Consolidación. La guía cubana cuenta además, con un conjunto de anexos importantes que sirven de apoyo a los especialistas que guían la migración, anexándose a la misma la lista de herramientas alternativas que recomienda el Grupo Técnico Nacional para la migración, organizadas por categorías de software. Con una amplia lista de referencias bibliográficas culmina este importante documento que ha servido de guía en la migración de los OACE desde el 2009.

El libro "Buenas Prácticas para la Migración a Código Abierto" se basa en las experiencias adquiridas y resultados satisfactorios durante la migración en disímiles organizaciones, tanto en Cuba como en el extranjero, así como en la evolución de las TIC. En él se reflejan elementos importantes a tener en cuenta para llevar a cabo un proceso exitoso de migración a código abierto. Las buenas prácticas expresadas en este documento han sido probadas satisfactoriamente tanto en entidades nacionales como en empresas privadas e instituciones del sector público en el extranjero (Pérez y otros, 2014).

La actividad de evaluación del software de escritorio constituye una de las buenas prácticas del libro. Esta actividad se realiza durante el servicio de consultoría.

### ***Consultoría en migración a código abierto***

La consultoría constituye un eslabón fundamental dentro del proceso de migración a código abierto (Pérez y otros, 2014). Esta se ejecuta durante la etapa de preparación, a continuación se describe la actividad de evaluación del software de escritorio definida en este servicio:

#### **Etapa Preparación. Actividad: Evaluación del software de escritorio**

Es importante tener en cuenta que la herramienta al usarse en la organización no tiene porque ser necesariamente la que más cumple con las funcionalidades especificadas, debe ser en todos los casos, la que más se adecue a las condiciones del proceso que va a automatizar. Para seleccionar la aplicación de código abierto deben tenerse en cuenta varios criterios obtenidos de la información recopilada en las actividades anteriores y que respondan a las siguientes preguntas:

- ¿Cuál es el proceso que automatizará la aplicación?
- ¿Cuál es el nivel de preparación y conocimientos que tienen los usuarios sobre cómo aplicar ese proceso?
- ¿Cuál es el nivel de dominio que posee el usuario de la herramienta privativa que automatiza ese proceso?
- ¿Está ese proceso automatizado o no?
- ¿Qué funcionalidades debe tener la futura aplicación a usar?
- ¿Cuáles son las características técnicas que dispone el hardware sobre el que se va a ejecutar la nueva aplicación?

Las respuestas a las preguntas anteriores resultarán ser los elementos a tener en cuenta para seleccionar la herramienta libre adecuada. Se recomienda además:

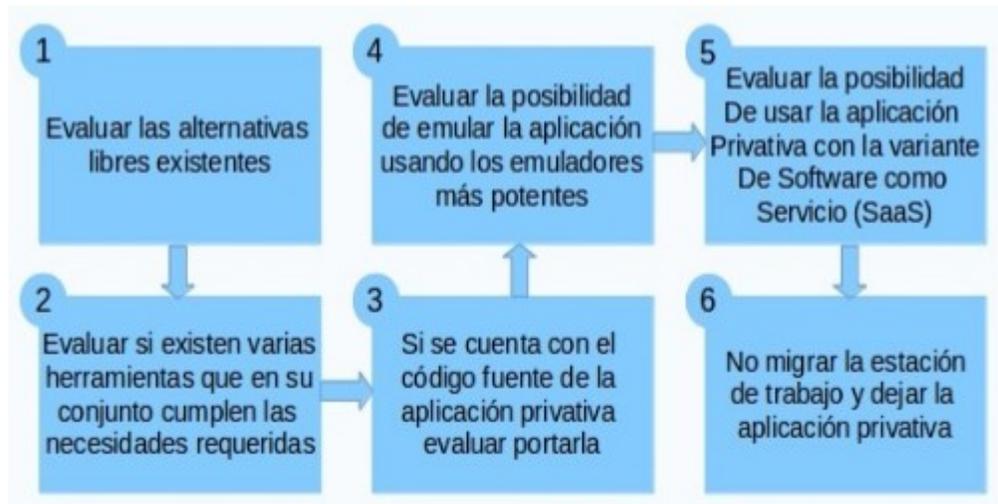
- No usar varias herramientas que correspondan a un mismo grupo funcional (por ejemplo

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

LibreOffice Writer y Abiword), pues esto dificultaría las actividades de soporte, capacitación y la interoperabilidad.

- Seleccionar preferiblemente soluciones que se ajusten e integren al entorno de escritorio que se seleccione.
- Seleccionar de todas las herramientas evaluadas, la más simple de comprender para el usuario.
- Siempre pensar antes de proponer la solución, en el usuario y el desarrollo del proceso, que es pensar en la futura productividad.

Lamentablemente, en casos muy específicos existe software privativo que no dispone de una alternativa libre con las funcionalidades necesarias, o no existe una alternativa para ser sustituido. Entonces se sugiere seguir el orden mostrado en la Figura 1 que ilustra las posibles soluciones que pueden brindarse.



**Figura 1. Evaluación del software de escritorio.** Fuente: Pérez y otros, 2014

Los pasos 1, 2 y 3 permitirán reemplazar la aplicación usada por una alternativa libre, en cambio, los restantes pasos tienen como objetivo mantener la aplicación privativa en aras de garantizar la productividad y a la vez disminuir la cantidad de software privativo instalado en cada ordenador donde se use la solución seleccionada.

Si luego de realizar múltiples pruebas y transitar los caminos desde el 1 hasta 4 no hay solución factible, entonces se debe valorar la variante 5. Los usuarios que empleen dicho software no tendrían que poseer el sistema operativo privativo en su ordenador, sino que se conectarían a un servidor y desde allí ejecutarían dicha herramienta, disminuyendo de esta forma la cantidad de software privativo empleado.

Si ninguna de las posibles soluciones puede implementarse en la futura migración, entonces deben mantenerse los ordenadores de los usuarios que las emplean con software privativo; no se pueden detener los procesos fundamentales de la institución por usar 100% software de fuentes abiertas. En caso de que para alguna aplicación sea necesario implementar alguno de los pasos 4, 5 ó 6 de la Figura 1, entonces la migración se realizará la migración de las aplicaciones de manera parcial. A continuación, se describe la relación de las etapas de la metodología AUP-UCI y las etapas de la metodología de Migración.

### 1.3 Relación de las etapas AUP-UCI y las etapas de la metodología de Migración

De acuerdo a las etapas definidas en el proceso de migración, estas se distribuyen de la siguiente manera (ver Tabla 2, Tabla 3 y Tabla 4), teniendo en cuenta la estructura de la actividad de desarrollo-producción propuesta por la metodología AUP-UCI y las actividades de los servicios de consultoría y migración a código abierto. Como se puede apreciar en la Tabla 2, Tabla 3 y Tabla 4 se describen las actividades de los servicios de Consultoría, Migración y Migración Integral (Pérez, 2015).

**Tabla 2. Servicio de Consultoría.** Fuente: Pérez, 2015. Elaboración propia

Fases AUP-UCI	Disciplinas AUP- UCI	Etapas de la metodología de Migración	Actividades del servicio de Consultoría
Inicio		Preparación	Diagnóstico a la institución, personas y la tecnología.
			Análisis de factibilidad de la migración.
			<b>Evaluación del software de escritorio.</b>
			Certificación de hardware.
			Elaboración de las estrategias de sensibilización, capacitación y soporte técnico.
			Elaboración del plan de migración institucional.
Ejecución	Modelado de negocio	Ejecución	
	Requisitos		
	Análisis y diseño		
	Implementación		

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

	Pruebas internas		
	Pruebas de liberación		
	Pruebas de aceptación		
Cierre		Consolidación	

**Tabla 3. Servicio de Migración.** Fuente: Pérez, 2015. Elaboración propia

Fases AUP-UCI	Disciplinas AUP- UCI	Etapas de la metodología de Migración	Actividades del servicio de Migración
Inicio		Preparación	
Ejecución	Modelado de negocio	Ejecución	Instalar aplicaciones libres sobre Microsoft Windows.
	Requisitos		Entrenamiento en aplicaciones libres.
	Análisis y diseño		Realizar la migración de los formatos de archivos
	Implementación		Desinstalar aplicaciones privativas
	Pruebas internas		Migrar el sistema operativo
	Pruebas de liberación		Capacitar los usuarios en GNU/Linux
	Pruebas de aceptación		Acompañamiento
Cierre		Consolidación	

**Tabla 4. Servicio de Migración Integral.** Elaboración propia

Fases AUP-UCI	Disciplinas AUP- UCI	Etapas de la metodología de Migración	Servicio de Migración Integral
Inicio		Preparación	Consultoría (ver Tabla 2)
Ejecución	Modelado de negocio	Ejecución	Migración (ver Tabla 3)
	Requisitos		
	Análisis y diseño		
	Implementación		
	Pruebas internas		
	Pruebas de liberación		

	Pruebas de aceptación		
Cierre		Consolidación	

De manera general, la actividad evaluación del software de escritorio se realiza en el servicio de Consultoría, y posteriormente se mide la satisfacción de los usuarios con las alternativas propuestas en la actividad acompañamiento del servicio de Migración. Para el proceso de selección de alternativas libres es imprescindible el análisis y la toma de decisiones por parte de los especialistas en migración.

#### **1.4 Análisis de decisiones**

La Universidad de las Ciencias Informáticas (UCI) decidió adoptar procesos para el desarrollo de aplicaciones informáticas a nivel de la organización, con el propósito de asegurar la calidad de sus productos y ganar en competitividad. Por ello, apostó la utilización del modelo de calidad CMMI para desarrollo (CMMI-DEV) nivel 2. Actualmente la UCI se encuentra certificada en el nivel 2 de Madurez y los grupos técnicos trabajan en aras de certificar el nivel 3 de dicho modelo.

El centro CESOL realiza servicios de migración a código abierto y desarrolla la distribución cubana GNU/Linux Nova, guiado por los procesos de calidad con CMMI nivel 2 y 3. Por lo que, es necesario que la presente investigación se ajuste a los lineamientos de la universidad. A continuación se describe el área de proceso *Análisis de Decisiones y Resolución (DAR)* del nivel 3, la cual tiene como propósito analizar las posibles decisiones utilizando un proceso de evaluación formal para evaluar las alternativas identificadas.

El área de proceso DAR tiene como propósito analizar las posibles decisiones utilizando un proceso de evaluación formal que evalúa las alternativas identificadas, frente a unos criterios establecidos. Además, implica establecer guías para determinar qué cuestiones deberían estar sujetas a un proceso de evaluación formal. Las guías sugieren la utilización de procesos de evaluación formal cuando las cuestiones están asociadas con riesgos de impacto medio o alto, o cuando las cuestiones afectan a la capacidad para conseguir los objetivos del proyecto. Durante el proceso se pueden utilizar tanto criterios numéricos como no numéricos. Los criterios numéricos utilizan pesos para reflejar la importancia relativa de los criterios y los criterios no numéricos utilizan una escala de rangos subjetiva. La eventual selección de una solución puede implicar actividades iterativas de identificación y de evaluación (CMMI-DEV, 2010).

A continuación se lista la meta (SG) y las prácticas específicas (SP) del área de proceso:

*SG 1 Evaluar las alternativas.*

SP 1.1 Establecer las guías para el análisis de decisiones.

*Subprácticas:*

- Establecer guías para determinar cuándo utilizar un proceso de evaluación formal.
- Incorporar la utilización de guías en el proceso definido según proceda.

SP 1.2 Establecer los criterios de evaluación.

*Subprácticas:*

- Definir los criterios para evaluar las soluciones alternativas.
- Definir el rango y la escala para clasificar los criterios de evaluación.
- Clasificar los criterios.
- Evaluar los criterios y su importancia relativa.
- Evolucionar los criterios de evaluación para mejorar su validez.
- Documentar el análisis razonado para la selección y el rechazo de los criterios de evaluación.

SP 1.3 Identificar las soluciones alternativas.

*Subprácticas:*

- Realizar una búsqueda bibliográfica
- Identificar otras alternativas a considerar además de las alternativas que pueden ser proporcionadas con la cuestión.
- Documentar las alternativas propuestas.

SP 1.4 Seleccionar los métodos de evaluación.

*Subprácticas:*

- Seleccionar métodos en base al propósito de analizar una decisión y a la disponibilidad de la información utilizada para dar soporte al método.

## *Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

- Seleccionar los métodos de evaluación en base a su capacidad para centrarse en los temas en cuestión, sin estar demasiado influenciados por cuestiones secundarias.
- Determinar las medidas necesarias para dar soporte al método de evaluación.

### SP 1.5 Evaluar las soluciones alternativas.

#### *Subprácticas:*

- Evaluar soluciones alternativas propuestas utilizando los criterios de evaluación establecidos y los métodos seleccionados.
- Evaluar supuestos relacionados con los criterios de evaluación y la evidencia que sustenta las suposiciones.
- Evaluar si la incertidumbre en los valores de las soluciones alternativas afecta a la evaluación, y tratar estas incertidumbres según proceda.
- Realizar simulaciones, modelados, prototipos y pilotos, según sea necesario, para ejercitar los criterios y los métodos de evaluación y las soluciones alternativas.
- Considerar nuevas soluciones alternativas, criterios o métodos si los resultados de las pruebas no fueron satisfactorios y repetir las evaluaciones hasta que los resultados sean satisfactorios.
- Documentar los resultados de la evaluación.

### SP 1.6 Seleccionar las soluciones.

#### *Subprácticas:*

- Evaluar los riesgos asociados con la implementación de la solución recomendada.
- Documentar y comunicar a las partes interesadas relevantes los resultados y el análisis razonado para la solución recomendada.

En la presente investigación se utiliza esta área para la selección de alternativas libres, debido a que no puede obviar ninguna de las actividades que propone, con el fin de seleccionar cuál es alternativa libre más adecuada y si ésta realmente cumple con las necesidades de los usuarios. Para la toma de decisiones se emplea tanto atributos numéricos como no números. Los especialistas en migración deben

realizar la actividad de evaluación de las alternativas libres de forma iterativa, por la constante evolución de las tecnologías y puede que en el mercado exista una alternativa más competitiva. Existen diversos modelos, metodologías y métodos que plantean un conjunto de buenas prácticas para seleccionar un software de código abierto, los mismos son abordados en el siguiente epígrafe.

## **1.5 Modelos FLOSS**

En la actualidad existe un amplio número de modelos orientados a los FLOSS. Durante la revisión bibliográfica se identificaron 25 modelos FLOSS, de los cuales dos no pudieron ser analizados, pues no se encuentra disponible la información. A continuación se describen brevemente los 23 modelos para su posterior comparación en cuanto a forma de evaluación y selección de las aplicaciones, las etapas y las actividades.

### **1. Modelo de Madurez de Código Abierto Capgemini**

El Modelo de Madurez de Código Abierto Capgemini (C-OSMM) fue desarrollado por la Compañía Capgemini en el 2003, con el fin poder entender el nivel de madurez del grupo que está desarrollando el software. C-OSMM (versión 2) consta de 13 preguntas, que cubren 4 áreas de interés (producto, ambiente, aceptación y uso). Cada pregunta es un indicador de madurez, con un total 16 indicadores de madurez, de los cuales 13 se utilizan por punto de vista (integrador o consumidor) (Duijnhouwer, 2003).

Cada indicador consiste en una pregunta y tres criterios predefinidos. Estos criterios están vinculados a un puntaje del indicador correspondiente (1, 3 o 5). Se responde la pregunta y ver qué criterios coinciden, para aquellos que son más que 1 y menos que 3 se puede obtener 2 (un poco de los criterios por 1 punto y un poco del criterio de 3 puntos, pero no satisface ninguno) o 4 (lo mismo, pero ahora por 3 puntos y 5 puntos). Al responder todas las preguntas, se alcanza un puntaje total y un puntaje por área de interés. El puntaje es un promedio no ponderado de las respuestas en esa área. Como regla general, el OSMM v2 considera que cualquier puntaje menor a 3 es inmaduro y un puntaje superior indica que la madurez del software es lo suficientemente alta para ser utilizado sin riesgos indebidos. Cuando el puntaje es exactamente 3, necesita observar cualquier signo de peligro, es decir, las respuestas individuales que están por debajo de 3.

### **2. Marco de Evaluación para Software de Código Abierto**

El Marco de Evaluación para Software de Código Abierto (EF-OSS) fue desarrollado por Koponen y Hotti en el 2004. Este ayuda a comparar productos y proyectos de software de código abierto. El marco divide

el software de código abierto en el producto de información, el proyecto y los proyectos de servicio. El producto de información describe las características del software. El proyecto describe las características de un proyecto y los productos de servicio describen los servicios comerciales proporcionados y sus proveedores. Separan las licencias, la audiencia y el modelo. Las licencias están divididas en cuatro grupos. Las categorías de la audiencia son desarrolladores, usuarios, administradores, clientes, investigadores. Los modelos propuestos son Bazaar (requiere de una estructura jerárquica de la comunidad de desarrollo) y Cathedral (desarrolla actividades dirigidas a los líderes). El marco evalúa los productos y proyectos de software de código abierto analizando sus sitios web, listas de correos, versiones entre otros (Koponen, 2004).

### **3. Modelo para la Evaluación Comparativa de Productos de Código Abierto**

El Modelo para la Evaluación Comparativa de Productos de Código Abierto (MCA-OSP) fue desarrollado por Polančič y Horvat en el 2004, este presenta una base cuantitativa para seleccionar entre múltiples productos alternativos del sistema operativo en relación con múltiples criterios relacionados con la calidad del software. Implica la evaluación objetiva y subjetiva de la calidad, que corresponde a dos fases principales: la fase de evaluación de la calidad y la fase de evaluación de la idoneidad. Además, se propone una fase de preevaluación para identificar posibles productos de software (Polančič, 2004).

La fase de evaluación de la calidad se refiere a la evaluación objetiva de la calidad. Se basa en una suposición previa de que la calidad del software del sistema operativo está altamente correlacionada con los atributos del proceso de desarrollo del software del sistema operativo. Esta fase presenta un enfoque simple y eficiente en el tiempo (basado en atributos específicos del proyecto cuantificables) para filtrar posibles alternativas de baja calidad. En función de los atributos seleccionados, se definen 10 métricas de calidad, según lo propuesto la norma ISO/IEC 9126-2: 2003. Cada una de las métricas tiene un peso de definido, que presenta el impacto de la métrica en la evaluación de la calidad. El peso de las métricas oscila de 1 (bajo) a 3 (alto). Debido a que la definición de pesos inevitablemente requiere cierta subjetividad, se recomienda que varios expertos en código abierto realicen la definición de pesos. Además, al obtener valores para todas las alternativas, los valores métricos deben normalizarse de dos maneras (dos ecuaciones): una se usa cuando se prefieren valores más altos y la otra para cuando los valores más bajos son favorecidos. Finalmente, se obtiene la evaluación cuantitativa de la calidad de la alternativa con una ecuación final.

En la fase de evaluación de la idoneidad se determina la idoneidad de un producto. Esta fase depende de los requisitos definidos por el usuario en lugar de una breve lista de criterios generados por el evaluador.

Además, aunque los usuarios a menudo expresan información cualitativa, ambos tipos de datos pueden manipularse en esta fase: cuantitativa y cualitativa. Un modelo adecuado, que puede considerar simultáneamente los criterios cualitativos y cuantitativos, es el Proceso Analítico de Jerarquía (AHP).

#### **4. Modelo de Madurez de Código Abierto Navica**

El Modelo de Madurez de Código Abierto Navica (N-OSMM) fue desarrollado por Golden en el 2004, cuyo objetivo es evaluar la madurez de un producto en tres fases: evaluar la madurez de cada elemento de producto y asignar un puntaje de madurez, definir un factor de ponderación para cada elemento en función de los requisitos de la organización y calcular el puntaje de madurez general del producto (Golden, 2004).

En la primera fase se identifican los elementos clave del producto y se evalúa el nivel de madurez de cada elemento. Los elementos clave del producto son: software del producto, soporte, documentación, formación, integraciones del producto y servicios profesionales. Cada elemento se evalúa y se le asigna un puntaje de madurez a través de un proceso de cuatro pasos: definir los requisitos de la organización, localizar los recursos, evaluar la madurez del elemento y asignar la calificación de madurez del elemento en una escala de 1 a 10.

En la segunda fase se asigna una ponderación al puntaje de madurez de cada elemento. La ponderación permite que cada elemento refleje su importancia para la madurez general del producto (software del producto [4], soporte [2], documentación [1], formación [1], integraciones del producto [1] y servicios profesionales [1]). En la tercera fase el puntaje ponderado de cada elemento se suma para proporcionar un puntaje general de madurez para el producto. Las organizaciones pueden optar por ajustar los factores de ponderación predeterminados en función de sus necesidades específicas.

Después de evaluar cada elemento y asignarle un factor de ponderación, se calcula el puntaje total de madurez del producto. Los puntajes de los elementos se suman para dar una calificación general de madurez del producto en una escala de 1 a 100.

#### **5. Modelo de Madurez de Código Abierto de Woods y Guliani**

El Modelo de Madurez de Código Abierto (OSMM) es desarrollado por Woods y Guliani en el 2005, este intenta cuantificar la madurez de un producto de código abierto. El modelo evalúa la madurez en tres áreas principales: producto, uso e integración. Los criterios del producto son específicos del producto en sí. Los criterios de uso son específicos sobre el uso diario del producto, desde el esfuerzo inicial de instalación y configuración hasta el trabajo requerido para el mantenimiento diario y los mecanismos de soporte disponibles para ayudar a adaptar el producto a las necesidades de la empresa y corregir los defectos encontrados. Los criterios de integración son específicos sobre lo que se necesita para que el

producto funcione en el entorno de la empresa. Para cada criterio se asigna una puntuación de 1 (inmaduro), 2 (razonable) o 3 (muy maduro) (Woods, 2005).

## **6. Clasificación de Preparación para Negocios Abiertos**

La Clasificación de Preparación para Negocios Abiertos (OpenBRR) es un modelo desarrollado por *Spike Source* en el 2005, el Centro de Investigación de Código Abierto en *Carnegie Mellon West* en asociación con *Intel Corporation*. Está encaminada en apoyar a los administradores de TI a evaluar qué software de fuente abierta sería el más adecuado para sus necesidades. Los usuarios de código abierto también pueden compartir sus calificaciones de evaluación con posibles adoptantes, continuando el círculo virtuoso y la "arquitectura de participación" de código abierto (OpenBRR, 2005).

Este modelo estandariza diferentes tipos de datos de evaluación, agrupándolos en categorías. El modelo propone cuatro fases de evaluación de software:

- Filtro de evaluación rápida: el objetivo de este primer paso es reducir la lista de selección de software libre eliminando aquellos que no cumplen con los criterios básicos. Los usuarios pueden usar la lista propuesta de criterios de filtrado y pueden agregar la suya propia para cualquier situación en particular.
- Evaluación de uso del objetivo: clasifica las categorías como primer paso, según su interés (1 "la más alta", 12 "la más baja") y selecciona las primeras 7 categorías para asignar un porcentaje a cada una, respetando el 100%.
- Recopilación y procesamiento de datos: recopila los datos medidos utilizados en la categoría correspondiente y calcula la ponderación aplicada para las mediciones.
- Calcula la puntuación BRR: utilizando las clasificaciones de categoría y los factores funcionales de ponderación.

## **7. Calificación y Selección de Software de Código Abierto**

La Calificación y Selección de Software de Código Abierto (QSOS) es un método desarrollado por *Atos-Origin Company* en el 2006, se compone de cuatro etapas: definición, evaluación, calificación y selección. En el 2013 se publicó una versión más actualizada del método. En el primer paso se definen los elementos de la tipología: software, licencia y comunidad. En la tipología de software se agrupan dos categorías: análisis de madurez del proyecto y análisis de cobertura funcional donde se definen 4 categorías con 16 criterios. Las licencias se clasifican en 3 categorías, donde sus valores son sí o no o parcial. En la tipología de comunidad se clasifican en 5 categorías (QSOS, 2013).

El segundo paso es evaluar el software de código abierto, donde se debe evaluar la versión del software

teniendo en cuenta: nombre, versión, información, descripción y un análisis detallado de sus características. A los criterios se le asigna una puntuación de 0 a 2 (0- funcionalidad no cubierta, 1- funcionalidad parcialmente cubierta, 2- funcionalidad completamente cubierta).

En la etapa calificación se define un conjunto de elementos traduciendo las necesidades y las restricciones alineadas. En este paso se realizan tres tipos de filtros: filtrar las alternativas basado en un criterio o necesidad específica, filtrar por madurez y filtrar por cubrimiento funcional.

En la última etapa se debe seleccionar el software según las necesidades de los usuarios o comparar el software del mismo tipo. El modo de selección puede ser estricto o informal. La estricta selección es hecha cuando se elimina un proceso (la eliminación del software que no fue resultado satisfactorio en el filtro de identidad, la eliminación del software que no proporciona las funcionalidades requeridas y la eliminación de un software que el criterio de madurez no coincide con la relevancia definida por el usuario). Al software que se le aplica un proceso de selección se le asigna una puntuación global ponderada, de la misma manera que la selección suelta. La selección suelta es menos estricta que la anterior porque en lugar de eliminar software que no son elegibles, los clasifica al medir la diferencia en comparación con los filtros definidos previamente. Para ello, se basa en los valores de ponderación: 0- funcionalidad no requerida, 1- funcionalidad opcional, 3- funcionalidad requerida). El factor de ponderación de madurez se basa en el grado de relevancia de cada criterio de madurez (0- criterio no relevante, 1- criterio relevante y 3- criterio crítico).

QSOS es también un proyecto gratuito y comunitario dedicado a la observación colaborativa de software libre y de código abierto. Además, propone herramientas de apoyo para la automatización del proceso. También, permite seleccionar un software según las ponderaciones, representando los requisitos especificados por el usuario. Este método ayuda a seleccionar la mejor solución de código abierto en un contexto dado.

### **8. Criterios de Evaluación para Productos de Software Libre de Código Abierto**

Los Criterios de Evaluación para Productos de Software Libre y de Código Abierto (EC-FLOSSP) es un enfoque de apoyo a la toma de decisión, desarrollado por Cruz y otros autores en el 2006. El enfoque propone actividades para la evaluación de los FOSS, tales como: escenarios de uso de los FOSS, requerimientos funcionales, criterios de requerimientos y cumplimiento de los requerimientos. Los escenarios de uso para FOSS ocurrirán en varias combinaciones y se multiplican en función de los requisitos orientados a objetivos para un producto FOSS y proyectos similares (Cruz y otros, 2006).

Los requisitos brindan la posibilidad de establecer ponderaciones para los criterios poniendo énfasis

especial en un aspecto, mientras que a otros aspectos se ignoran parcialmente. Del mismo modo, los datos específicos del proyecto se clasifican en varias categorías, donde los requisitos específicos de la compañía varían desde aspectos técnicos y funcionales hasta características económicas e incluso políticas. Las características requeridas deben enumerarse por prioridades. Los requerimientos se clasifican en funcionales, técnicos, organizacionales, legales, económicos y políticos. Los aspectos cualitativos deben organizarse en una escala de "insignificante" a "esencial". Para cada requerimiento durante la ejecución de las actividades se definen métricas que sirven de apoyo a la toma de decisiones.

Este enfoque no propone una evaluación cuantitativa del número de criterios basados en sus pesos asignados para ayudar a comparar fácilmente los resultados de evaluación cuando se evalúan piezas de software similares para un uso particular.

### **9. Modelo de Evaluación de Código Abierto**

El Modelo de Evaluación de Código Abierto (E-OSS/FS-P) es desarrollado por Wheeler en el 2006. Propone cuatro pasos de evaluación de software (Wheeler, 2006):

- Identificar: se recomienda la consulta de listas publicadas que proporcionan software valorado como maduro o con experiencia o como herramienta segura.
- Leer reseñas: consultar varios foros para conocer las ventajas y desventajas de cada solución en función de los comentarios de los usuarios.
- Comparar: comparar los atributos clave del software con las necesidades del usuario. Estos atributos son: funciones activadas, costo, cuota de mercado, soporte, mantenimiento, confiabilidad, desempeño, escalabilidad, usabilidad, seguridad, flexibilidad/personalización, interoperabilidad y cuestiones legales/licencia.
- Analizar: analizar el posible software candidatos para probarlos en una situación que represente la carga de trabajo.

### **10. Observatorio de Innovación y Transferencia Tecnológica en Software de Código Abierto**

El proyecto Observatorio de Innovación y Transferencia Tecnológica en Software de Código Abierto (OITOS) fue financiado por la región italiana de Emilia-Romagna, comenzó en 2005 con el objetivo de crear un observatorio para limitar y controlar los riesgos para la adopción de TI empresarial en soluciones de código abierto. El proyecto se construyó para evaluar proyectos de código abierto, en lugar de productos, en torno a tres categorías, destinadas a medir un conjunto de objetivos propios de interés general de forma cuantitativa: (1) Desarrollo; (2) Comunidad; (3) Transición (Cabano y otros, 2007).

La categoría desarrollo mide el trabajo realizado por los desarrolladores de proyectos, y está compuesta

por las métricas: documentación, desarrolladores, herramientas y capacidad de compilación. La categoría comunidad mide todos los recursos ofrecidos por los usuarios, las comunidades científicas y empresas privadas, y está compuesta por las métricas: visibilidad, éxito de historias, soporte y financiamiento. La categoría transición estima cuánto se puede adaptar el producto a la organización existente estructuras y configurable para su uso efectivo, y está compuesto por las métricas: configurabilidad, configuración, uso, migración e interoperabilidad.

Para la evaluación de software dependiente del contexto se agregó una nueva categoría de Tecnología. La categoría de tecnología agrupa todas las métricas relativas a las características que una empresa puede considerar importante para su contexto específico. Estas métricas generalmente miden las facetas conectadas directamente con el producto de software, nunca con la infraestructura del proyecto de código abierto. Por esta razón, calificar esas métricas es una actividad subjetiva, pero las evaluaciones están más personalizadas para su contexto tecnológico que en otras categorías.

El proceso de evaluación está compuesto por tres fases: (1) Análisis de contexto; (2) Selección preliminar; (3) Selección filtrada. Durante la fase de Análisis de contexto, se recopila información sobre una empresa para definir sus necesidades. Cada necesidad se resuelve mediante una clase de software que será estimada por el modelo. Después de la identificación de las clases de software se edita una lista que contiene posibles productos para introducir y se define un conjunto completo de métricas de evaluación. En la fase de Selección preliminar se extrae un conjunto de métricas más críticas. A cada métrica crítica se le asocia un umbral, que debe superarse para otorgarle acceso a un producto de software a la siguiente fase. Durante la Selección filtrada, los productos de software que fueron satisfactorios en la Selección preliminar se estiman una vez más según el conjunto completo de medidas definidas en la fase de Análisis de contexto. No hay umbrales asignados a las métricas: en cambio, cada métrica recibe un valor de 1 a 10 que puede modificarse mediante un factor de ponderación.

### **11. Marco para la Evaluación de Sistemas Críticos de Código Abierto**

El Marco para la Evaluación de Sistemas Críticos de Código Abierto (FOCSE) es desarrollado por Ardagna y otros autores en el 2007, se basa en un conjunto de indicadores de propósito general adecuados para evaluar marcos de código abierto en general. Sin embargo, incluye algunas métricas específicas que expresan la capacidad de las soluciones de seguridad para responder a un cambio continuo en las amenazas. Este evalúa un proyecto de código abierto en su totalidad, evaluando la composición de la comunidad, la capacidad de respuesta al cambio, la calidad del software y el soporte del usuario. Utiliza FLOSSmole como plataforma para reunir, compartir y almacenar datos que respaldan el análisis

comparativo del software de código abierto y el operador OWA desarrollado por Ronal Yager para proporcionar una estimación única de cada solución evaluada (Ardagna, 2007).

La evaluación se basa en seis macro áreas: aspectos generales, comunidad de desarrolladores, comunidad de usuarios, calidad de software, documentación y soporte de interacción, e integración y adaptabilidad con tecnologías nuevas y existentes. Independientemente del área a la que pertenecen las métricas, las métricas cuantitativas son divididas en métricas del núcleo (13 métricas) y métricas avanzadas (5 métricas). Las métricas del núcleo incluyen todas las métricas que se pueden calcular fácilmente a partir de las actuales informaciones sobre los proyectos. Las métricas avanzadas incluyen parámetros de evaluación que requieren acceso privilegiado a la comunidad de desarrolladores.

### **12. Tarjeta de Puntuación Equilibrada para Software de Código Abierto**

La Tarjeta de Puntuación Equilibrada para Software de Código Abierto (BSC-OSS) es una adaptación de la técnica BSC al OSS, desarrollado por Lavazza en el 2007. En el dominio financiero plantea que el costo del software debe ser tenido en cuenta, evaluando el esfuerzo requerido por los desarrolladores. Además, para obtener una visión completa de los problemas relacionados con los costos, se consideran las alternativas y los beneficios de adoptar OSS. En el proceso evalúa qué tan bien el OSS satisface las necesidades de los procesos de la organización (esta área se inspira en el modelo OpenBQR). En el aspecto futuro, un primer problema se refiere a qué tan bien el OSS apoya el esfuerzo realizado por la organización para estar preparado para las tareas futuras y un segundo problema se refiere a la disponibilidad de versiones futuras del OSS. La satisfacción del cliente no solo el OSS contribuye al proceso interno, sino que también ayuda a alcanzar objetivos externos de la organización (Lavazza, 2007).

### **13. Clasificación de Calidad para Negocios Abiertos**

La Clasificación de Calidad para Negocios Abiertos (OpenBQR) es un marco de evaluación desarrollado por Taibi y otros autores en el 2007. Se definen varios indicadores directos que muestran la calidad de un paquete de software que se adopta, dividido en cinco áreas diferentes: análisis de requisitos funcionales, evaluación del uso del objetivo, calidad interna, calidad externa y probabilidad de soporte en el futuro. La evaluación del uso del objetivo tiene en cuenta temas tales como: licencia, estándares, lenguaje, documentación, soporte. El proceso de evaluación se compone de tres fases: filtro de evaluación rápida, recopilación y procesamiento de datos, y traducción de los datos. Este método se basa ISO/IEC 9126: 2001 para las cualidades internas del producto y en el Open BRR para las fases (Taibi y otros, 2007).

#### **14. Evaluación de Software de Código Abierto mediante Prototipos**

La Evaluación de Software de Código Abierto mediante Prototipos (EOSS-P) es desarrollado por Carbon y otros autores en el 2007. Está compuesto por tres fases: preparación, creación de prototipos y evaluación final. En la fase de preparación se realiza un análisis inicial de los requisitos, se seleccionan los candidatos OSS y se configuran los equipos de evaluación. Para ello, se tienen en cuenta los requisitos funcionales y no funcionales del sistema de software que se debe desarrollar, las restricciones arquitectónicas, los factores de contexto, los candidatos OSS y las especificaciones OSS como productos de entrada. La fase de creación de prototipos sigue un enfoque iterativo, donde los prototipos se desarrollan en varias iteraciones. Al final de cada iteración, se realiza un paso de evaluación. Si hay varios candidatos OSS disponibles, se desarrolla un prototipo por candidato OSS en paralelo. En la última fase todos los resultados de la evaluación de la fase de anterior se consolidan y se integran en un informe de evaluación (Carbon y otros, 2007).

#### **15. Calidad en Software de Código Abierto**

La metodología Calidad en Software de Código Abierto (QualOSS) es un proyecto desarrollado por la Comisión Europea en el 2008 (Ciolkowski y Soto, 2008). Fue diseñado originalmente para apoyar la evaluación de la calidad de los proyectos FLOSS, con un enfoque en la capacidad de evolución y la solidez. Está compuesto por tres tipos de elementos interrelacionados: características de calidad, métricas e indicadores (Soto y Ciolkowski, 2009).

En el último reporte QualOSS tiene como objetivo desarrollar una metodología de evaluación para FLOSS basada en el paradigma Goal Question Metric (GQM). La metodología espera que las respuestas a las preguntas ayuden a evaluar el riesgo de integrar un componente de FLOSS. En consecuencia, sugiere que las respuestas a las preguntas utilicen una escala de 4 valores que se pueda visualizar usando colores verde, amarillo, rojo y negro, donde el verde indica que no hay riesgo (o riesgo insignificante), amarillo (pequeño riesgo), rojo (riesgo medio) y negro (riesgo alto). Para proporcionar respuestas a las preguntas, las medidas se agregan y luego se proyectan en la escala de 4 valores. Esta agregación y proyección define lo que se conoce como un indicador. Claramente, algunos indicadores pueden ser más importantes que otros para evaluar riesgo, por lo tanto, a cada indicador se le puede asignar un peso. El valor de un indicador se caracteriza además por una cobertura y el peso de la importancia del indicador para responder una pregunta (Deprez, 2010).

Este proceso de evaluación se divide en 5 tareas: (1) iniciar una evaluación, (2) configurar y planificar una evaluación, (3) recopilación y análisis de datos, (4) interpretación de resultados y (5) supervisión de una

evaluación. Las primeras 4 tareas se realizan en secuencia, mientras que la última tarea es transversal, por lo tanto, se ejecuta a lo largo de las otras tareas del proceso de evaluación.

#### **16. Observatorio de Calidad de Software para Software de Código Abierto**

El Observatorio de Calidad de Software para Software de Código Abierto (SQO-OSS) es un modelo desarrollado por Samoladas y otros autores en el 2008. Este evalúa todos los aspectos del desarrollo de OSS, tanto el producto (código) como la comunidad. Además se compone de dos pasos: (1) definición del modelo de evaluación (atributos, subatributos y métricas) y (2) definición del método de agregación (categorías de evaluación y sus perfiles). El modelo utiliza una versión simplificada del proceso Goal-Question-Metric (GQM). Para la evaluación se utilizan cuatro categorías: excelente, bueno, justo y pobre. Al tener cuatro categorías, el método de agregación requiere de la definición de tres perfiles, cada uno para las tres primeras categorías (Samoladas y otros, 2008).

#### **17. Enfoque Operativo para Seleccionar Componentes de Código Abierto**

El Enfoque Operativo para Seleccionar Componentes de Código Abierto (OA-SOSC) es desarrollado por Majchrowski y Deprez en el 2008. La metodología para seleccionar FOSS es aplicable en dos escenarios: (1) proyecto de desarrollo de software interno y (2) proyecto de desarrollo de software de consultoría. En el primer escenario el cliente y el equipo de desarrollo de software trabajan para la misma empresa y la garantía de la calidad puede ser manejada por un solo equipo de calidad. En el segundo escenario el cliente y el desarrollo del software son entidades diferentes. El enfoque propone 8 actividades donde se obtiene una decisión final de selección del FOSS (Majchrowski y Deprez, 2008).

#### **18. Modelo de Confiabilidad QualiPSO**

El Modelo de Confiabilidad es uno de los principales objetivos de la Plataforma de Calidad para Software de código abierto (QualiPSO) desarrollado por Bianco y otros autores en el 2009. Este define un conjunto de métricas para capturar los diversos componentes de confiabilidad desde diferentes puntos de vista en dos fases. En la primera fase, las diversas dimensiones de confiabilidad de los productos de software y los artefactos se identifican y describen. Estas dimensiones de confiabilidad representan el punto de vista del usuario, es decir, la percepción de la confiabilidad del usuario. Se miden cuantitativamente las características y subcaracterísticas de cada elemento del modelo conceptual. La segunda fase se compone de un modelo conceptual de confiabilidad basado en la norma ISO/IEC 9126: 2001, los factores de influencia de confiabilidad, y un conjunto de medidas que cuantifican tanto a los factores que componen la confiabilidad como a las características del producto que influyen en las cualidades de confiabilidad (Bianco y otros, 2009).

El modelo incluye una parte conceptual, que describe el significado de las cualidades, y una parte operativa (la orientada a la medición), que proporciona suficientes detalles para apoyar la evaluación cuantitativa de los diversos aspectos de confiabilidad. La última parte se define por medio de la técnica GQM para la obtención de un plan. Además utilizan la técnica BSC adoptada a los FOSS donde la medición de los diversos aspectos puede demostrar que los efectos de la decisión son equilibrados, y las consecuencias globales de la decisión coinciden con el objetivo de la organización.

### **19. Modelo de Madurez de código Abierto**

El Modelo de Madurez de código Abierto (OMM) es desarrollado por Petrinja y otros autores, está organizado en niveles, y cada nivel se basa en elementos de confianza (TWE, por sus siglas en inglés). Los elementos confiables incluidos en OMM fueron recopilados e inspirados por dos fuentes: una amplia encuesta realizada a desarrolladores, usuarios e integradores de FLOSS y en las áreas de proceso CMMI (Petrinja y otros, 2009).

El modelo comprende 25 elementos confiables agrupados en 3 niveles de madurez denominados (122 prácticas y 630 métricas): básico, intermedio y avanzado (Petrinja y otros, 2012). Los elementos confiables son características del proceso de desarrollo o del producto desarrollado que aseguran a los clientes finales la calidad del producto.

En los componentes de bajo nivel se utiliza el enfoque Métrica de la Pregunta del Objetivo (GQM). El número de los objetivos especificados varían entre uno y cinco, y depende de la complejidad del TWE al que pertenecen. Se usa un valor de umbral uniforme para todo el modelo; hay cuatro posibles valores de evaluación que son del 1 al 4 con la posibilidad de asignar un 0 para métricas que no se aplican en dominios específicos. El valor umbral de 4 significa que la implementación de los elementos solicitados se cumple más del 75%, el valor 3 significa que se cumple entre 50% y 75%, el valor 2 entre 25% y 50%, y el valor 1 que su implementación es menos del 25% de lo solicitado por la práctica.

### **20. FAME: metodología de evaluación**

La metodología Filtrar-Analizar-Medir-Evaluar (FAME) es desarrollada por Pani y otros autores en el 2010. La idea principal es que los usuarios deben evaluar qué solución entre las disponibles es más adecuada a sus necesidades, comparando factores técnicos y económicos, y teniendo en cuenta el costo total de las soluciones individuales y las salidas de efectivo. La metodología está estructurada en cuatro fases: filtrar, analizar, medir y evaluar. En la fase de filtrado se obtiene una Tarjeta de Identidad del proyecto, mediante la cual se podrá llevar a cabo una evaluación totalmente cualitativa pero altamente efectiva para las operaciones de filtrado (Pani y otros, 2010).

En la fase de análisis se comprende qué solución puede satisfacer las necesidades de la organización considerada a fin de garantizar una productividad efectiva y eficiente. Los componentes finales basados en la información son los siguientes: análisis técnico y funcional, y análisis económico y social. En la fase de medición los elementos medibles provienen de las necesidades, donde la metodología convierte las evaluaciones de las partes interesadas de alto nivel en valores técnico-funcionales y valores económico-sociales. Además, se asocia un peso y una métrica a estos elementos y es posible realizar el análisis comparativo entre las soluciones. Por último, se realiza una comparación sistemática y se asocia un peso a cada valor.

### **21. Modelo de Evaluación de Código Abierto más Simple**

El Modelo de Evaluación de Código Abierto más Simple (E-OSSEM) fue desarrollado por Houaich y Belaissaoui en el 2012, se inspira en otros modelos tales como: C-OSMM, OpenBRR, OSEM, N-OSMM, QSOSS y SQO-OSS. El propósito del modelo es, por un lado, utilizar OSS y aprovechar sus ventajas, por otro lado, permitir que los responsables de la toma de decisiones seleccionen el mejor producto sin la intervención de expertos en TI. Está constituido por 4 fases (Houaich, 2015):

- **Definición:** describir las necesidades reales de cualquier empresa que considere la adopción de un nuevo OSS, donde se verifican y analizan los siguientes aspectos: funcional, técnico y estratégico.
- **Identificación:** determinar las características generales del OSS para crear una hoja de datos que describa todos los elementos clave. Para lograr eso, se dividen en 4 grupos (Producto, Integración, Calidad y Facilidad) con indicadores para evaluar la puntuación del producto.
- **Calificación:** asignar un puntaje para cada criterio con el fin de obtener un puntaje general para facilitar la elección del software más apropiado. Para mayor precisión, la evaluación se realiza proporcionando una puntuación de 0 a 5.
- **Selección:** después de recopilar toda la información necesaria, el responsable de la toma de decisiones tiene la oportunidad de recopilar los valores numéricos asignados a cada atributo para obtener un puntaje final. Para eso, se puede clasificar los diferentes OSS evaluados por orden de relevancia, lo que significa que el software que satisfaga las necesidades obtendrá el puntaje más alto.

### **22. USQO-FOSS: Modelo de Calidad**

El Observatorio de Calidad de Software basado en la Utilización para la evaluación del Software Libre y de Código Abierto (USQO-FOSS) es desarrollado por Vijaya y otros autores en el 2017. Se basa en los modelos SQO-OSS y SEM-OSS para evaluar los factores de calidad. Donde del modelo SQO-OSS

agrega las características portabilidad y utilización, y del modelo SEM-OSS agrega compartición y utilización. Cada uno de los factores de calidad tiene atributos secundarios que especifican las características de calidad responsables de la utilización efectiva. Además, los factores de calidad se miden al medir los atributos secundarios de los factores (Vijaya y otros, 2017).

El valor de medición para la evaluación del factor en cuestión se elige entre el intervalo de escala 0 y 100. '0' representa la peor medición posible para la evaluación, '25' representa la medición bastante mala, '50' representa el estado de medición indeciso, '75' representa que la medida tiene un impacto suficientemente bueno en la utilización de FOSS, y '100' representa la mejor medida posible para la evaluación. Los datos recopilados de diferentes estrategias de grupos de trabajo son variables medidas que se analizan con el índice de utilización de la Métrica de logro de objetivos de usabilidad (UGAM, por siglas en inglés) y Índice de utilización de software de código abierto (OSSUI, por siglas en inglés).

### **23. OSSpal: encontrar y evaluar el software de código abierto**

El proyecto OSSpal es desarrollado por Wasserman y otros autores en el 2017, donde se identifican categorías para el proceso de evaluación de código abierto basado en las normas ISO/IEC 9126: 2001 y ISO/IEC 25010: 2011. OSSpal es un sucesor de la metodología OpenBRR, que combinan medidas de evaluación cuantitativas y cualitativas para software en diversas categorías (Wasserman y otros, 2017). Agrupa los proyectos en categorías, según la taxonomía de software producida anualmente por Corporación Internacional de Datos (IDC, siglas en inglés), donde se definen criterios de evaluación genéricos y criterios de evaluación dependientes de la categoría. Los visitantes del sitio pueden dejar una calificación general y una revisión escrita de los proyectos individuales, pero la calificación se basa simplemente en una escala de 1 a 5 estrellas.

Los atributos de calidad de los proyectos de FLOSS pueden clasificarse en dos categorías generales: métricas de hardware y métricas de software. Las medidas de hardware son medidas cuantificables objetivas que cubren la mayoría de los atributos en áreas de tecnología de software y proceso de desarrollo. Las métricas de software son medidas cualitativas subjetivas que cubren la mayoría de los atributos en áreas de características de software operacional, así como servicio y soporte.

#### **1.5.1 Análisis de los modelos FLOSS**

Parte del análisis de los modelos FLOSS se realiza teniendo en cuenta los elementos que propone el área DAR de CMMI nivel 3. Esta área plantea que se debe definir atributos y clasificarlos, rango de importancia de los atributos, métodos.

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

Los modelos FLOSS (ver anexos 1, 2 y 3) definen categorías o áreas para la evaluación y selección de un FLOSS, excepto los modelos de Polančič y Wheeler. Además, definen atributos desde 6 hasta 29, excepto los modelos de SpikeSource (12 categorías y métricas) y Pani (grupos y métricas).

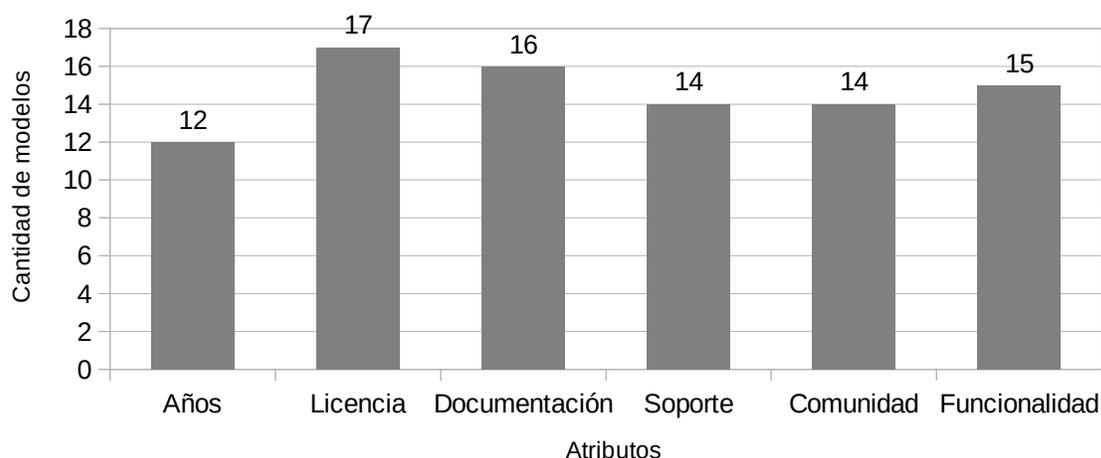
El modelo que más métricas propone es Petrinja con 630, se considera que es muy complejo de aplicar en la selección de alternativas libres durante el proceso de migración a código abierto, porque se desea que la selección sea simple y rápido, para elegir el FLOSS más idóneo para las instituciones cubanas.

De los 23 modelos solo 7 establecen importancia o prioridades, estos son: Modelo para evaluación comparativa de productos de código abierto (Polančič, 2004), OpenBRR, QSOS, OITOS, OpenBQR, QualOSS y SEM-OSS.

Teniendo en cuenta que los modelos deben establecer categorías o atributos, métricas, peso, importancia y normalización (si aplica), se obtuvo que los siguientes modelos cumplen con estos elementos: Modelo para evaluación comparativa de productos de código abierto (Polančič, 2004), N-OMMM, OpenBRR, QSOS y QualOSS.

En la Figura 2 se puede apreciar que más de la mitad de los modelos consideran que los atributos son importantes para la selección de un FLOSS.

Los modelos FLOSS no se basan en el área DAR de CMMI nivel 3 para el proceso de evaluación y selección del software, solo el modelo propuesto por Petrinja en el 2009 se basa en el modelo de calidad CMMI para el diseño de su estructura y la forma de evaluación. Sin embargo, el 70% de los modelos utilizan las características y métricas propuestas por las Normas ISO/IEC 9126-25000-15504.



**Figura 2. Cantidad de modelos que consideran los atributos importantes.** *Elaboración propia.*

Los atributos, las métricas y el nivel de importancia que definen los modelos FLOSS serán revisados y evaluados para una posterior adaptación en las instituciones cubanas. En caso que se adicionen nuevos atributos, será teniendo en cuenta las condiciones de dichas instituciones y las experiencias adquiridas de los especialistas en migración del centro CESOL durante los procesos de migración a código abierto.

### **Conclusiones parciales**

- La evaluación del software de escritorio se realiza durante el servicio de Consultoría de migración a código abierto, concluyendo que en dicha actividad no se define cómo se realiza la evaluación de las alternativas libres.
- El análisis del área DAR de CMMI nivel 3 para la toma de decisiones durante un proceso de evaluación del software de escritorio permitió determinar elementos clave que deben ser tenidos en cuenta, tales como: definición de atributos, métricas y métodos, clasificación de los atributos e importancia y el análisis de los riesgos de la organización.
- La comparación de los modelos FLOSS arrojó que cumplen con la definición de tributos, métricas y métodos, clasificación de los atributos e importancia, los modelos: Modelo para evaluación comparativa de productos de código abierto, N-OMMM, OpenBRR, QSOS y QualOSS (ver Anexos 2 y 3). La mayoría de los modelos consideran que los siguientes atributos son importantes para la selección de la alternativa libre: años, licencia, comunidad, soporte, documentación y funcionalidad (ver Anexos 4 y 5).

## **CAPÍTULO 2. ESTRATEGIA DE SELECCIÓN DE ALTERNATIVAS LIBRES**

La realización de una correcta selección de alternativas libres en un proceso de migración a código abierto permite incrementar la satisfacción de los usuarios. En el presente capítulo se presentan, de manera general, los resultados de las entrevistas realizadas a los especialistas del centro CESOL para describir cómo se realiza actualmente el proceso de selección de alternativas libres. Además se describe la estrategia de selección de alternativas libres, teniendo en cuenta las actividades y artefactos durante el proceso.

### **2.1 Diagnóstico del proceso actual**

Para la elaboración de la estrategia se procedió a realizar un diagnóstico del estado real de la selección de alternativas libres en un proceso de migración a código abierto. El diagnóstico consistió en la realización de una entrevista a 14 especialistas en procesos de migración a código abierto del Centro de Software Libre (CESOL). La misma estuvo constituida por cinco preguntas relacionadas con el número de procesos de migración en los que el entrevistado se ha visto involucrado, los roles que ha desempeñado, la manera en que se realiza la selección de alternativas libre, la forma en que los especialistas los realizan y qué elementos consideran necesarios para ser incluidos durante la ejecución de la selección de alternativas libres en un proceso de migración (ver Anexo 6: Guión de la entrevista aplicada a los especialistas en migración del centro CESOL).

De los especialistas entrevistados 7 tienen categoría docente representando el 46%. El 64% de los especialistas entrevistados ha participado en tres o más procesos de migración y el 92% por ciento ha desempeñado el rol de especialista en migración. De forma general plantearon que la selección de alternativas se realiza basado en la experiencia de aquellos que llevan a cabo el proceso de migración y solo guiándose por las buenas prácticas que son planteadas en la Guía Cubana para la Migración a Código Abierto y el Libro de Buenas Prácticas para Migración a Código Abierto. Dentro de los elementos que faltan señalan que es importante idear mecanismos, técnicas que faciliten y estandaricen el proceso. Además, que se centralice el conocimiento de los expertos en migración, así como las experiencias adquiridas en los casos de éxitos realizados, a modo de lograr que este proceso sea lo más práctico y sencillo posible.

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

Los especialistas en migración consideran que existen atributos de calidad que tienen un grado de mayor o menor importancia que otros. Por ejemplo, el recubrimiento funcional de la alternativa libre y la satisfacción de los usuarios tienen un nivel de importancia alto. En la Tabla 5 se presenta la definición del nivel de importancia de los atributos de acuerdo a los resultados de las entrevistas a los especialistas en migración. En el Anexo 7: Principales elementos de las respuestas a la entrevista aplicada a los especialistas en migración del centro CESOL se puede consultar más información de las respuestas brindadas.

**Tabla 5. Nivel de importancia por los especialistas en migración.** *Elaboración propia.*

Atributos	Nivel de importancia
Funcionalidad	Alto (A)
Satisfacción del usuario final	Alto (A)
Utilización de recursos	Medio (M)
Disponibilidad del código fuente	Alto (A)
Propietario	Bajo (B)
Popularidad	Medio (M)
Actividad de errores	Medio (M)

### 2.2 Estrategia de selección de alternativas libres

La estrategia está encaminada en apoyar a los especialistas en migración para seleccionar la alternativa libre más adecuada en un proceso de migración a código abierto de las estaciones de trabajo, con el propósito de satisfacer las necesidades de los usuarios finales en las organizaciones cubanas. Ésta se basa en modelos FLOSS y criterios establecidos por los especialistas en migración para la obtención de los atributos genéricos de la alternativa libre, en la Norma ISO/IEC 2502n (25022-25023) para la adquisición de las métricas específicas de la alternativa libre y en el área DAR de CMMI nivel 3 para la definición de las actividades durante un proceso de evaluación formal.

Además, debe cumplir con las siguientes características:

- **Ciclo iterativo:** la evaluación de las alternativas libres debe ser realizada por los especialistas en migración para cada alternativa libre, debido a los rápidos cambios en la industria del software y cumpliendo con lo establecido en el área DAR de CMMI nivel 3.
- **Sencilla:** la estrategia debe ser fácil de entender y de usar por parte de los especialistas en

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

migración, porque se describen detalladamente las actividades durante la selección de alternativas libres y los artefactos tienen comentarios para su utilización.

La estrategia de selección (ver Figura 3) de alternativas libres en el proceso de migración a código abierto está constituida por 12 actividades, divididas en cuatro etapas que complementan el proceso de selección. Las etapas son: Definición, Identificación, Evaluación y Selección.



**Figura 3. Estrategia de selección de alternativas libres.**

*Elaboración propia.*

La estructura de estrategia es la siguiente:

### **Etapas 1. Definición:**

- a) Definir las necesidades de los usuarios.
- b) Definir el objetivo de medición.
- c) Definir y clasificar los atributos para la evaluación de las alternativas libres.
- d) Definir el rango de los atributos y los métodos.
- e) Evaluar los atributos y su importancia relativa.

### **Etapas 2. Identificación:**

- a) Identificar las alternativas libres.
- b) Describir las alternativas libres.

**Etapa 3. Evaluación:**

- a) Evaluar las alternativas libres.
- b) Probar las soluciones alternativas.
- c) Evaluar los riesgos de la organización.

**Etapa 4. Selección:**

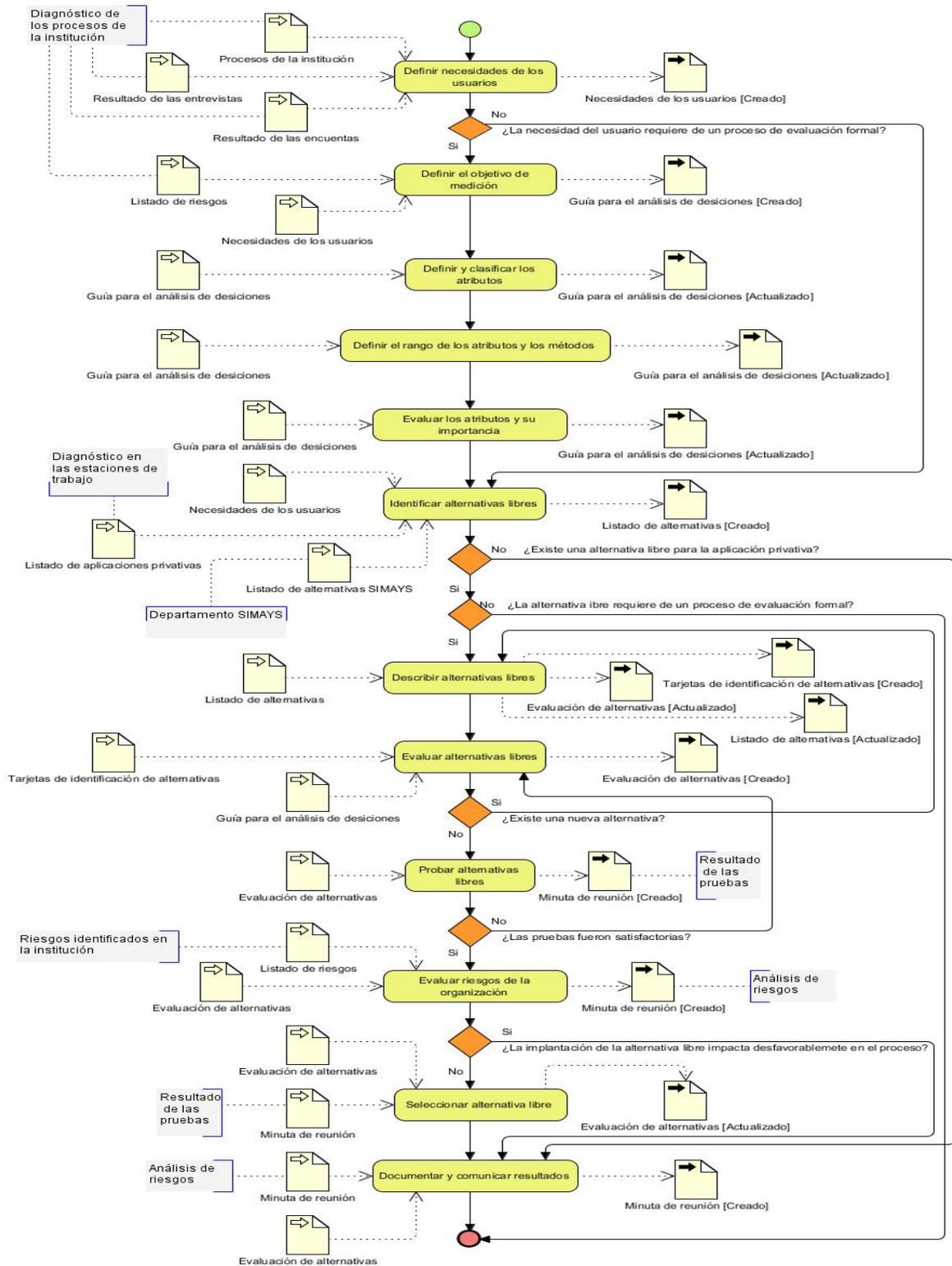
- a) Seleccionar las alternativas libres.
- b) Documentar y comunicar los resultados.

A continuación se muestra el diagrama de actividades (ver Figura 4) para la selección de alternativas libres en los procesos de migración a código abierto, reflejando los artefactos de entrada y salida en cada actividad y respetando los resultados de actividades realizadas en el proceso de migración.

Una vez ejecutadas las actividades de forma general se obtienen como salidas los siguientes productos de trabajo:

- 1. Necesidades de los usuarios:** es el artefacto que describe las necesidades de los usuarios por áreas de la institución, teniendo en cuenta los componentes: funcional, técnico y estratégico.
- 2. Guía para el análisis de decisiones:** es el artefacto que guía a los especialistas en migración para el análisis de toma de decisiones en el proceso de selección. En el documento se describe cuándo es necesario realizar un proceso de evaluación formal y se definen los atributos, métricas y métodos para la selección de las alternativas libres.
- 3. Listado de alternativas:** es el artefacto que contiene el listado de las alternativas libres por categorías para cada aplicación privativa correspondiente a la organización.
- 4. Tarjetas de identificación de las alternativas:** es el artefacto que describe las alternativas libres de cada aplicación privativa para su posterior selección.
- 5. Evaluación de alternativas:** es el artefacto que contiene la evaluación final de las alternativas libres para cada aplicación privativa correspondiente a la organización, resaltando la alternativa seleccionada.

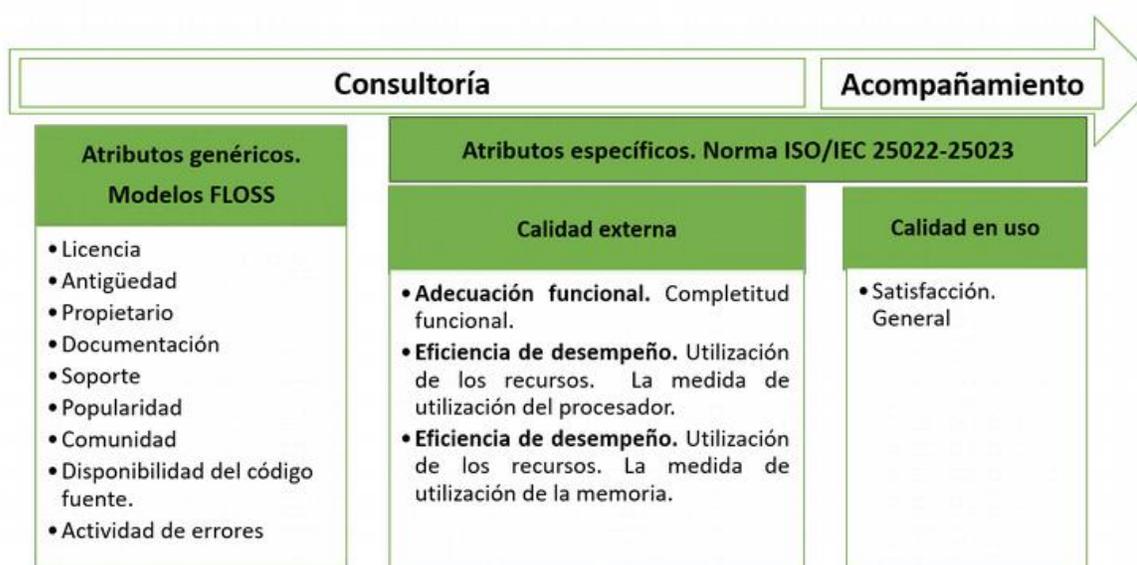
## Estrategia de selección de alternativas libres para el proceso de migración a código abierto



**Figura 4. Diagrama de actividades.** Elaboración propia.

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

Los artefactos que se obtienen como salidas en cada una de las etapas funcionan como elementos de entrada para la siguiente etapa. Durante la evaluación y selección de alternativas libres se definen atributos genéricos y específicos que son analizados durante la consultoría y acompañamiento a código abierto. Un atributo de calidad (AC) es una propiedad específica de un sistema de software que puede asumir un valor cualitativo o cuantitativo, el cual es medible u observable (Fritz, 2017). Los atributos genéricos son las características comunes de las alternativas libres, que describen lo más característico de ellas, sin entrar en detalles o especificaciones. Los atributos específicos son los requerimientos propios de la alternativa libre. Como se puede apreciar en la Figura 5 se muestra en qué momento del proceso de migración a código abierto se deben evaluar los atributos.



**Figura 5. Atributos genéricos y específicos para la selección.**

*Fuente: Normas ISO/IEC 25022: 2016 y 25023: 2017. Elaboración propia.*

La autora de la presente investigación considera que los atributos (licencia, antigüedad, documentación, soporte y comunidad) deben tenerse en cuenta en el proceso de selección de las alternativas, de acuerdo con los modelos FLOSS (ver Anexos 4 y 5). Además, como se manifiesta en los resultados de las encuestas realizadas a los especialistas en migración, es necesario evaluar la utilización de los recursos de la RAM y el procesador de la estación de trabajo a migrar a código abierto (para la asignación de la distribución cubana GNU/Linux Nova Ligerito y Nova Escritorio), que se encuentre disponible el código fuente de la alternativa libre para ser probada (cambio en código fuente para la adaptación a Nova) y

lograr la satisfacción de los usuarios. Otro atributo importante es el propietario de la alternativa, porque permite conocer si el producto es desarrollado por una persona o una institución. Los productos desarrollados por instituciones son más recomendados que los desarrollados por individuos independientes. Los atributos popularidad (aceptación de los usuarios) y actividad de errores (número de errores solucionados) son necesarios para el proceso de evaluación. Por último, el cumplimiento de las funcionalidades en la alternativa libre es fundamental en un proceso de migración a código abierto, en ocasiones una única alternativa no satisface completamente las funcionalidades de la aplicación privativa (ver Anexo 7).

### **2.2.1 Etapa 1. Definición**

La etapa de **definición** es de gran importancia, describe las necesidades reales de los usuarios finales de la organización que considere la adopción de una alternativa libre.

#### **a) Definir las necesidades de los usuarios.**

En esta actividad se emplean las encuestas resultantes de los usuarios finales y las entrevistas realizadas a los directivos de las áreas durante la consultoría de migración a código abierto. Además, se tiene en cuenta los procesos estratégicos, sustantivos y de apoyo de la institución. Es importante verificar y analizar los siguientes aspectos en el artefacto *Necesidades de los usuarios*: "funcional, técnico y estratégico". El componente funcional describe las funciones necesarias para proporcionar una lista de software que puede ayudar a satisfacer las necesidades. El componente técnico permite la adquisición de información sobre elementos secundarios que pueden contribuir al éxito de la selección del sistema, como son las prestaciones de hardware en cuanto a memoria RAM y procesador de la estación de trabajo. El componente estratégico debe tenerse en cuenta, con el fin de definir cuáles son las necesidades de los usuarios clasificados en los procesos de la institución. Se aplica la técnica de entrevista para la recopilación de información.

**Artefactos de entrada:** entrevistas con los directivos de las áreas, encuestas de los usuarios, procesos de la institución.

**Artefactos de salida:** necesidades de los usuarios (creado).

Antes comenzar con la próxima actividad es importante conocer si la necesidad del usuario requiere de un proceso de evaluación formal. Esto se refiere a que en ocasiones los usuarios solicitan aplicaciones a los especialistas en migración con fines personales y sin interferir en los procesos de la institución. Si es el caso, el especialista en migración realiza la actividad de identificación de las alternativas libres.

**b) Definir el objetivo de medición.**

En esta actividad se define el objetivo de la medición (ver Tabla 6) para el proceso de evaluación formal, donde es necesario que se tengan en cuenta los riesgos de la institución. Se define un objeto de medición en base a las necesidades de los usuarios.

**Artefactos de entrada:** listado de riesgos, necesidades de los usuarios.

**Artefactos de salida:** guía para el análisis de decisiones (creado).

Para la selección de la alternativa libre se aplica la técnica Meta-Pregunta-Métrica (Goal-Question-Metric, por sus siglas en inglés) con el propósito de definir el objeto de medición, evaluando cada alternativa libre.

**Tabla 6. Objetivo de medición.** *Elaboración propia.*

<b>No. del objetivo de medición:</b> Analizar las alternativas libres identificadas en el proceso de migración a código abierto, examinando su idoneidad, con el fin de seleccionarlas.		<b>Prioridad:</b> <i>[De 1 a 5, siendo el 5 la menor prioridad]</i>
<b>Referencia del objetivo de la medición de la que proviene:</b> Seleccionar alternativas libres		
<b>Objeto de interés:</b>	Proceso de migración a código abierto	
<b>Propósito:</b>	Analizar las alternativas libres identificadas en el proceso de migración a código abierto, a partir de los atributos genéricos y específicos, con el fin de seleccionarlas.	
<b>Perspectiva:</b>	Examinar la idoneidad de las alternativas libres identificadas desde el punto de vista de los especialistas de migración.	
<b>Ambiente:</b>	Empresa, Institución	

**c) Definir y clasificar los atributos para la evaluación de las alternativas libres.**

En esta actividad se definen y se clasifican los atributos para la evaluación de las alternativas libres. Los atributos están agrupados en dos categorías: atributos genéricos y atributos específicos. Los atributos genéricos son licencia, antigüedad, propietario, documentación, soporte, popularidad, comunidad, disponibilidad del código fuente y actividad de errores, basados en los modelos FLOSS y en los resultados de las entrevistas de los especialistas en migración. Los atributos específicos están clasificados a su vez según definen las Normas ISO/IEC 25022:2016 y 25023: 2017, tales como: calidad externa y calidad en uso. En calidad externa se evalúa la adecuación funcional de la alternativa libre en cuanto a la completud funcional. Además, se evalúa la subcaracterística utilización de recursos en la medida eficiencia de desempeño (memoria y procesador). En calidad en uso se evalúa la satisfacción general del cliente. Se

aplica la técnica de tormentas de ideas entre los especialistas en migración para la definición y clasificación de los atributos.

**Artefactos de entrada:** guía para el análisis de decisiones.

**Artefactos de salida:** guía para el análisis de decisiones (actualizado).

A continuación se describen los atributos genéricos y específicos para la selección de las alternativas libres durante el proceso de migración a código abierto:

**Licencia:** permite conocer si el producto es realmente de software libre o código abierto, a través del tipo de licencia. La licencia debe permitir modificaciones y trabajos derivados, y debe permitir que se distribuyan bajo los mismos términos que la licencia del software original (OpenSource, 2007). El mejor tipo de licencia será aquella que garantice que sus trabajos derivados continúen bajo la misma licencia y que no cambie a licencias privativas. Una licencia permisiva es simple, y es el tipo de licencia de código abierto más básico: permite hacer lo que quiera con el software, siempre y cuando se cumpla con los requisitos de aviso. Las licencias *copyleft* agregan requisitos a las licencias permisivas (MasLinux, 2017). *Copyleft* es un método general para hacer un programa de software libre y requiere que todas las versiones modificadas y extendidas del él sean también software libre (FSF, 2018). Se recomienda usar la licencia *copyleft* más fuerte dependiendo del propósito de software, por ejemplo utilizar la versión más reciente de la Licencia Pública General de GNU (GPL) (FSF, 2018). En el Anexo 8: Licencias más populares en aplicaciones de código abierto se puede apreciar las licencias más populares en aplicaciones de código abierto (OpenSource, 2007), y si estas son permisivas o *copyleft* (MasLinux, 2017).

**Antigüedad:** cantidad de días en que el producto fue liberado por primera vez para el uso de los usuarios finales. Los productos que han estado disponibles por largos períodos de tiempo tienden a ser más maduros (Golden, 2004).

**Propietario:** permite conocer si el producto es desarrollado por una persona o una institución. Los productos desarrollados por instituciones son más recomendados que los desarrollados por individuos independientes.

**Documentación:** el grado de disponibilidad de la documentación debe ser tenido en cuenta, aquellas aplicaciones poco documentadas por lo general son complejas y poco usadas. La existencia de foros de discusión, listas de correo, manuales de uso y de desarrollo son elementos a tener en cuenta. Además, de libros y artículos publicados.

**Soporte:** existen varios niveles de soporte, los más populares para productos de fuentes abiertas son el soporte comunitario y profesional. Se debe tener en cuenta que exista un adecuado soporte comunitario. Este atributo está vinculado a la existencia de una comunidad.

**Popularidad:** se refiere a la cantidad de estrellas de calidad del código fuente de la alternativa libre definida por los usuarios. Las aplicaciones más usadas, poseen por lo general una mayor aceptación de los usuarios (búsqueda en Google Trends para ayudar a determinar el valor de esta variable).

**Comunidad:** se refiere a la cantidad de colaboradores de la alternativa libre. Las comunidades activas garantizarán la mantenibilidad en el tiempo del producto y mejores tiempos de respuesta ante posibles problemas.

**Disponibilidad del código fuente:** permite descargar el código fuente desde el sitio para su posterior uso.

**Actividad de errores:** cuando se utiliza una aplicación que no es desarrollada por nuestro centro es importante que exista una activa solución de errores por parte de la comunidad de desarrollo.

**Adecuación funcional/Complejidad funcional:** permite conocer si la alternativa libre cuenta con la función específica por el usuario.

**Eficiencia de desempeño/Utilización de los recursos/La media de utilización del procesador:** permite conocer cuánto requiere la alternativa libre del procesador para su instalación.

**Eficiencia de desempeño/Utilización de los recursos/La media de utilización de la memoria:** permite conocer cuánto requiere la alternativa libre de la memoria para su instalación.

**Satisfacción general:** es el grado en que se satisfacen las necesidades del usuario cuando un producto o sistema se utiliza en un contexto de uso establecido (ISO/IEC 25022, 2016). Este atributo se evalúa durante el acompañamiento, luego que están migradas todas las estaciones de trabajo con las alternativas libres instaladas.

***d) Definir el rango de los atributos y los métodos.***

En esta actividad se define el rango de los atributos de evaluación. En los atributos genéricos (ver Tabla 7) y específicos (ver Tabla 8) los valores están entre 0 y 1, y entre 1 y 5, en dependencia de la evaluación del atributo. En los atributos específicos los valores significan que la alternativa cumple funcionalmente con las necesidades de los usuarios (calidad externa), qué distribución instalar teniendo en cuenta el consumo

***Estrategia de selección de alternativas libres para el proceso de migración a código abierto***

de los recursos (calidad externa) y cuál es el grado de satisfacción de los usuarios (calidad en uso). Se aplica la técnica de tormentas de ideas entre los especialistas en migración para la definición del rango de los atributos y los métodos.

**Artefactos de entrada:** guía para el análisis de decisiones.

**Artefactos de salida:** guía para el análisis de decisiones (actualizado).

**Tabla 7. Medidas de los atributos genéricos.** *Elaboración propia.*

Atributos genéricos		
Nombre	Descripción	Función de medición
Licencia	¿Cuál es el tipo de licencia de la alternativa libre?	$X = n$ $n =$ (1: desconocida, 2: aprobada, 3: más popular y permisiva, 4: más popular y copyleft menos fuerte, 5: más popular y copyleft más fuerte)
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	$1 \leq X \leq 5$ . Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
Nombre	Descripción	Función de medición
Antigüedad	¿Cuántos años tiene la alternativa libre desde su lanzamiento?	$X = n$ $n =$ (1: menos de 3 meses, 2: entre 3 meses y 1 año, 3: entre 1 y 2 años, 4: entre 2 y 3 años, 5: más de 3 años)
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	$1 \leq X \leq 5$ . Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
Nombre	Descripción	Función de medición

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Propietario	¿Cuál es el propietario de la alternativa libre?	X = n n = (1: persona, 3: institución no reconocida, 5: institución reconocida)
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	1 <= X <= 5. Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>
Documentación	¿Existe documentación de la alternativa libre?	X = n n = (1: no posee, 3: poca, 5: abundante y actualizada) <b>Nota:</b> Debe existir publicación de libros, manuales técnicos y de usuarios, blog, fórum, listas de contacto, comentarios en el código fuente.
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	1 <= X <= 5. Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>
Soporte	¿Cuáles son los niveles de soporte de la alternativa libre?	X = n n = (1: no posee, 3: soporte comunitario, 5: soporte profesional)
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	1 <= X <= 5. Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Popularidad	¿Cuál es la cantidad de estrellas de calidad del código fuente de la alternativa libre?	$X = n$ $n = (1: \text{no posee}, 2: \text{menos de } 50, 3: \text{entre } 50 \text{ y } 250, 4: \text{entre } 251 \text{ y } 350, 5: \text{más de } 350)$
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	$1 \leq X \leq 5$ . Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>
Comunidad	¿Cuál es la cantidad de colaboradores de la comunidad de la alternativa libre?	$X = n$ $n = (1: \text{no posee}, 2: \text{menos de } 50, 3: \text{entre } 50 \text{ y } 250, 4: \text{entre } 151 \text{ y } 250, 5: \text{más de } 250)$
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	$1 \leq X \leq 5$ . Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>
Disponibilidad del código fuente	¿Se encuentra disponible el código fuente?	$X = n$ $n = (1: \text{no}, 5: \text{si})$
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	$1 \leq X \leq 5$ . Normalmente, cuanto mayor sea el valor es mejor.	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración	
<b>Nombre</b>	<b>Descripción</b>	<b>Función de medición</b>
Actividad de errores	¿Qué proporción de las funciones erróneas están resueltas?	$X = 1 - A/B$ $A = \text{Número de funciones erróneas}$

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

		resueltas B = Número de funciones erróneas Donde B > 0.
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	0 <= X <= 1. Normalmente, cuanto menor sea el valor es mejor.	
<b>Tipo de medida:</b>	Tamaño funcional	
<b>Recursos utilizados:</b>	Sitios web, Especialista en migración, Computadora	

**Tabla 8. Medidas de los atributos específicos.** Fuente: Normas ISO/IEC 25022:2016 y 25023: 2017. Elaboración propia

Adecuación funcional. Calidad externa		
Nombre	Descripción	Función de medición
Complejidad funcional	¿Qué proporción de las funciones especificadas existe de acuerdo a las necesidades de los usuarios?	$X = 1 - A/B$ A = Número de funciones ausentes B = Número de funciones especificadas Donde B > 0.
<b>Método:</b>	Inspección	
<b>Valor deseado:</b>	0 <= X <= 1. Normalmente, cuanto menor sea el valor es mejor.	
<b>Tipo de medida:</b>	Tamaño funcional	
<b>Recursos utilizados:</b>	Necesidades del usuario, Aplicación, Especialista en migración, Computadora	
Utilización de recursos. Calidad externa		
Nombre	Descripción	Función de medición
La media de utilización del procesador	¿Cuánto tiempo del procesador se utiliza para ejecutar un determinado conjunto de tareas en comparación con el tiempo de operación?	$X = \sum_{i=1}^n (A_i/B_i)/n$ A <sub>i</sub> = Tiempo de procesador realmente utilizado para ejecutar un conjunto dado

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

		de tareas en la observación i $B_i$ = Tiempo de operación para realizar las tareas en observación i n = Número de observaciones
<b>Método:</b>	Medir el rendimiento del procesador	
<b>Valor deseado:</b>	0 <= X <= 1. Normalmente, cuanto menor sea el valor es mejor	
<b>Tipo de medida:</b>	Tiempo y Cuenta	
<b>Recursos utilizados:</b>	Necesidades del usuario, Aplicación, Especialista en migración, Computadora	
<b>NOTA 1:</b> El tiempo de observación se refiere al tiempo medido externamente por el observador, utilizando un reloj externo, por ejemplo, tiempo para terminar una transacción o una tarea de usuario.		
Nombre	Descripción	Función de medición
La media de utilización de la memoria	¿Cuánto de la memoria se utiliza para ejecutar un determinado conjunto de tareas en comparación con la memoria disponible?	$X = \sum_{i=1}^n (A_i / B_i) / n$ $A_i$ = Tamaño de la memoria realmente utilizada para realizar un determinado conjunto de tareas para el procesamiento de i-ésima muestras $B_i$ = Tamaño de memoria disponible para realizar las tareas durante el procesamiento de la i-ésima muestras n = Número de muestras procesadas
<b>Método:</b>	Medir el rendimiento de la memoria	
<b>Valor deseado:</b>	0 <= X <= 1. Normalmente, cuanto menor sea el valor es mejor	
<b>Tipo de medida:</b>	Recurso utilizado y Cuenta	
<b>Recursos utilizados:</b>	Necesidades del usuario, Aplicación, Especialista en migración, Computadora	
<b>Atributos específicos. Calidad en uso</b>		

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Nombre	Descripción	Función de medición
Satisfacción general	¿Cuál es la satisfacción general del usuario con las alternativas libres?	$X = \sum A_i$ $A_i =$ Respuesta a una pregunta
<b>Método:</b>	Cuestionario	
<b>Valor deseado:</b>	<p><math>k \leq X \leq n</math>. Normalmente, cuanto mayor sea el valor es mejor</p> <p><math>n</math> = la cantidad de preguntas por el máximo de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados</p> <p><math>k</math> = la cantidad de preguntas por el menor de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados</p>	
<b>Tipo de medida:</b>	Puntaje	
<b>Recursos utilizados:</b>	Usuarios finales, Especialista en migración	

**e) Evaluar los atributos y su importancia relativa.**

En esta actividad se evalúan los atributos y su importancia. Los atributos específicos de calidad externa (completitud funcional) y en uso tienen importancia alta, y los atributos específicos de calidad externa (utilización de recursos) tienen importancia media. Los atributos genéricos que tienen importancia alta son: licencia, disponibilidad del código fuente y comunidad. El resto de los atributos genéricos tienen importancia media excepto la antigüedad que es baja. Se aplica la técnica de tormentas de ideas entre los especialistas en migración para la definición de la importancia relativa de los atributos.

**Artefactos de entrada:** guía para el análisis de decisiones.

**Artefactos de salida:** guía para el análisis de decisiones (actualizado).

La importancia de los atributos genéricos y específicos se muestra distribuida de la siguiente manera. Si se realiza el servicio de consultoría junto al servicio de migración se utilizan las ponderaciones de la Tabla 9. Si sucede lo contrario, es decir, que los servicios se ejecutan como proyectos independientes las ponderaciones a tomar en cuenta se pueden apreciar en la Tabla 10.

**Tabla 9. Ponderación en porcentaje de los atributos (Consultoría y Migración juntos).** *Elaboración propia.*

Atributos	Nivel de importancia	Ponderación	Motivo
-----------	----------------------	-------------	--------

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Atributo genéricos			
Licencia	Alto	3 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar el tipo de licencia del software para su utilización.
Antigüedad	Bajo	1 %	Se califica con valor de importancia <b>Bajo</b> porque se debe evaluar los años de vida del software.
Propietario	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar el tipo de propietario del software para su utilización.
Documentación	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar para que los usuarios finales pueden utilizar la documentación de la alternativa.
Soporte	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar el tipo de soporte del software para su utilización.
Popularidad	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar la cantidad de usuarios que utilizan el software.
Comunidad	Alto	3 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar la cantidad de desarrolladores del software para su utilización.
Disponibilidad del	Alto	3 %	Se califica con valor de importancia

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

código fuente			<b>Alto</b> porque es muy necesario evaluar que el código fuente se encuentre disponible.
Actividad de errores	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar la actividad de los errores del software.
<b>Subtotal</b>		<b>20 %</b>	
<b>Atributo específicos</b>			
<b>Calidad externa</b>			
Complejidad funcional	Alto	50 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar que las funciones especificadas cubran todas las necesidades determinadas por el usuario.
<b>Calidad externa</b>			
La media de utilización del procesador	Medio	2,5 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar que se utilice adecuadamente el procesador.
La media de utilización de la memoria	Medio	2,5 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar que se utilice adecuadamente la memoria.
<b>Subtotal</b>		<b>5 %</b>	
<b>Calidad en uso</b>			
Satisfacción general	Alto	25 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar que las funciones especificadas cubran todas las

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

			necesidades determinadas por el usuario.
<b>Total</b>		<b>100 %</b>	

**Tabla 10. Ponderación en porcentaje de los atributos (Consultoría y Migración separados).** *Elaboración propia.*

Atributos	Nivel de importancia	Ponderación	Motivo
<b>Atributo genéricos (Servicio de Consultoría)</b>			
Licencia	Alto	3 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar el tipo de licencia del software para su utilización.
Antigüedad	Bajo	1 %	Se califica con valor de importancia <b>Bajo</b> porque se debe evaluar los años de vida del software.
Propietario	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar el tipo de propietario del software para su utilización.
Documentación	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar para que los usuarios finales pueden utilizar la documentación de la alternativa.
Soporte	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar el tipo de soporte del software para su utilización.
Popularidad	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar la cantidad de usuarios que utilizan

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

			el software.
Comunidad	Alto	3 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar la cantidad de desarrolladores del software para su utilización.
Disponibilidad del código fuente	Alto	3 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar que el código fuente se encuentre disponible.
Actividad de errores	Medio	2 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar la actividad de los errores del software.
<b>Subtotal</b>		<b>20 %</b>	
<b>Atributo específicos</b>			
<b>Calidad externa (Servicio de Consultoría)</b>			
Complejidad funcional	Alto	75 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar que las funciones especificadas cubran todas las necesidades determinadas por el usuario.
<b>Calidad externa (Servicio de Consultoría)</b>			
La media de utilización del procesador	Medio	2,5 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar que se utilice adecuadamente el procesador.
La media de utilización de la memoria	Medio	2,5 %	Se califica con valor de importancia <b>Medio</b> porque es necesario evaluar que se utilice adecuadamente la

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

			memoria.
<b>Subtotal</b>		<b>5 %</b>	
<b>Total</b>		<b>100 %</b>	
<b>Calidad en uso (Servicio de Migración--Acompañamiento--)</b>			
Satisfacción general	Alto	100 %	Se califica con valor de importancia <b>Alto</b> porque es muy necesario evaluar que las funciones especificadas cubran todas las necesidades determinadas por el usuario.
<b>Total</b>		<b>100 %</b>	

### 2.2.2 Etapa 2. Identificación

En la etapa de **identificación** se determinan las características generales de las alternativas libres para crear una tarjeta de identificación que describa los elementos clave de la misma.

**a) Identificar las alternativas libres.**

En esta actividad se realiza un levantamiento de todas las posibles alternativas libres para cada aplicación privativa teniendo en cuenta las necesidades de los usuarios identificadas y el listado de alternativas general del departamento SIMAYS. Se aplica la técnica de recopilación de información para identificar las alternativas.

**Artefactos de entrada:** necesidades de los usuarios, listado de aplicaciones privativas y listado de alternativas SIMAYS.

**Artefactos de salida:** listado de alternativas (nuevo).

**b) Describir las alternativas libres.**

En esta actividad se describen las alternativas libres (ver Tabla 11) una vez que son identificadas. Para ello, se debe realizar el levantamiento de información en sitios, portales. En caso de existir una nueva alternativa libre durante la etapa evaluación se regresa a esta actividad y se actualiza el listado de alternativas. Se aplica la técnica de recopilación de información para describir las alternativas libres identificadas.

**Artefactos de entrada:** listado de alternativas.

**Artefactos de salida:** tarjetas de identificación de alternativas (nuevo), listado de alternativas (actualizado).

**Tabla 11. Tarjeta de identificación de la alternativa libre.** *Elaboración propia.*

Elemento	Descripción
Nombre:	<i>Nombre del software</i>
Descripción:	<i>Descripción del software</i>
Versión:	<i>Versión del software</i>
Url:	<i>Dirección del sitio del software</i>
Tipo:	<i>Tipo de software (Según las categorías de la distribución cubana GNU/Linux Nova Liger y Escritorio)</i>
Año:	<i>Año de la última liberación estable.</i>
Licencia:	<i>Licencia del software</i>
Idioma:	<i>Idioma del software (Español, requerido)</i>
Lenguajes:	<i>Lenguajes utilizados para el desarrollo</i>
Disponibilidad del código fuente:	<i>Disponibilidad del código fuente (Si o No)</i>
Nivel de criticidad:	<i>Nivel de criticidad del software para los procesos de la institución (Alto / Medio / Bajo). Los procesos de la institución son: estratégicos, sustantivos y de apoyo.</i>

### 2.2.3 Etapa 3. Evaluación

En la etapa de **evaluación** se analizan las alternativas libres teniendo en cuenta los atributos genéricos y específicos, y se realizan las pruebas o simulaciones pertinentes. Además, se analizan los riesgos asociados con la implementación de la alternativa libre en la organización a migrar. Para la evaluación es necesario conocer el tipo de servicio (Consultoría y Migración) que se está realizando.

#### **a) Evaluar las alternativas libres.**

En esta actividad se evalúan las alternativas libres teniendo en cuenta los atributos genéricos y específicos de las alternativas. Las informaciones relevantes de la alternativa libre están descritas en las tarjetas de identificación. Se aplica la técnica de tormentas de ideas para la evaluación de las alternativas libres.

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

**Artefactos de entrada:** tarjetas de identificación de alternativas, guía para el análisis de decisiones.

**Artefactos de salida:** evaluación de alternativas (nuevo).

Como se puede apreciar en la Tabla 12 y Tabla 13 para la evaluación de las alternativas libres se requiere de la normalización de los atributos genéricos y específicos.

**Tabla 12. Normalización de atributos (Consultoría y Migración juntos).** *Elaboración propia*

Atributo	Valor deseado	Normalización	Ponderación Total
Licencia	1 <= X <= 5	Si x = 1 ponderación 0,5 % Si x = 2 ponderación 1 % Si x = 3 ponderación 1,5 % Si x = 4 ponderación 2 % <b>Si x = 5 ponderación 3 %</b>	3 %
Antigüedad	1 <= X <= 5	Si x = 1 ponderación 0,2 % Si x = 2 ponderación 0,4 % Si x = 3 ponderación 0,6 % Si x = 4 ponderación 0,8 % <b>Si x = 5 ponderación 1 %</b>	1 %
Propietario	1 <= X <= 5	Si x = 1 ponderación 0,5 % Si x = 3 ponderación 1 % <b>Si x = 5 ponderación 2 %</b>	2 %
Documentación	1 <= X <= 5	Si x = 1 ponderación 0 % Si x = 3 ponderación 1 % <b>Si x = 5 ponderación 2 %</b>	2 %
Soporte	1 <= X <= 5	Si x = 1 ponderación 0 % Si x = 3 ponderación 1 % <b>Si x = 5 ponderación 2 %</b>	2 %
Popularidad	1 <= X <= 5	Si x = 1 ponderación 0 % Si x = 2 ponderación 0,5 % Si x = 3 ponderación 1 % Si x = 4 ponderación 1,5 %	2 %

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

		Si $x = 5$ ponderación 2 %	
Comunidad	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 2$ ponderación 0,5 % Si $x = 3$ ponderación 1 % Si $x = 4$ ponderación 2 % Si $x = 5$ ponderación 3 %	3 %
Disponibilidad del código fuente	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 5$ ponderación 3 %	3 %
Actividad de errores	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 0,5 % Si $0,25 < x \leq 0,50$ ponderación 1 % Si $0 < x \leq 0,25$ ponderación 1,5 % Si $x = 0$ ponderación 2 %	2 %
Complejidad funcional	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 5 % Si $0,25 < x \leq 0,50$ ponderación 25 % Si $0 < x \leq 0,25$ ponderación 40 % Si $x = 0$ ponderación 50 %	50 %
La media de utilización del procesador	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 0,5 % Si $0,25 < x \leq 0,50$ ponderación 1 % Si $0 < x \leq 0,25$ ponderación 2 % Si $x = 0$ ponderación 2,5 %	2,5 %
La media de utilización de la memoria	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 0,5 % Si $0,25 < x \leq 0,50$ ponderación 1 % Si $0 < x \leq 0,25$ ponderación 2 % Si $x = 0$ ponderación 2,5 %	2,5 %
Satisfacción general	$k \leq X \leq n$ $n =$ la cantidad de preguntas por el	$X = (n / \text{cantidad de encuestados}) /$ cantidad de preguntas Si $x < 3$ ponderación 0 %	25 %

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

	máximo de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados  k = la cantidad de preguntas por el menor de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados	(Insatisfactorio) Si $3 \leq x < 4,75$ ponderación 15 % (Satisfactorio) Si $4,75 \leq x \leq 5$ ponderación 25 % (Muy satisfactorio)	
<b>Total</b>			<b>100 %</b>

**Tabla 13. Normalización de atributos (Consultoría y Migración separados).** *Elaboración propia*

Atributo	Valor deseado	Normalización	Ponderación Total
<b>Servicio de Consultoría</b>			
Licencia	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0,5 % Si $x = 2$ ponderación 1 % Si $x = 3$ ponderación 1,5 % Si $x = 4$ ponderación 2 % Si $x = 5$ ponderación 3 %	3 %
Antigüedad	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0,2 % Si $x = 2$ ponderación 0,4 % Si $x = 3$ ponderación 0,6 % Si $x = 4$ ponderación 0,8 % Si $x = 5$ ponderación 1 %	1 %

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Propietario	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0,5 % Si $x = 3$ ponderación 1 % Si $x = 5$ ponderación 2 %	2 %
Documentación	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 3$ ponderación 1 % Si $x = 5$ ponderación 2 %	2 %
Soporte	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 3$ ponderación 1 % Si $x = 5$ ponderación 2 %	2 %
Popularidad	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 2$ ponderación 0,5 % Si $x = 3$ ponderación 1 % Si $x = 4$ ponderación 1,5 % Si $x = 5$ ponderación 2 %	2 %
Comunidad	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 2$ ponderación 0,5 % Si $x = 3$ ponderación 1 % Si $x = 4$ ponderación 2 % Si $x = 5$ ponderación 3 %	3 %
Disponibilidad del código fuente	$1 \leq X \leq 5$	Si $x = 1$ ponderación 0 % Si $x = 5$ ponderación 3 %	3 %
Actividad de errores	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 0,5 % Si $0,25 < x \leq 0,50$ ponderación 1 % Si $0 < x \leq 0,25$ ponderación 1,5 % Si $x = 0$ ponderación 2 %	2 %
Complejidad funcional	$0 \leq X \leq 1$	Si $0,75 < x \leq 1$ ponderación 0 % Si $0,50 < x \leq 0,75$ ponderación 10 % Si $0,25 < x \leq 0,50$ ponderación 30 % Si $0 < x \leq 0,25$ ponderación 65 % Si $x = 0$ ponderación 75 %	75 %

**Estrategia de selección de alternativas libres para el proceso de migración a código abierto**

La media de utilización del procesador	$0 \leq X \leq 1$	<p>Si <math>0,75 &lt; x \leq 1</math> ponderación 0 %</p> <p>Si <math>0,50 &lt; x \leq 0,75</math> ponderación 0,5 %</p> <p>Si <math>0,25 &lt; x \leq 0,50</math> ponderación 1 %</p> <p>Si <math>0 &lt; x \leq 0,25</math> ponderación 2 %</p> <p><b>Si <math>x = 0</math> ponderación 2,5 %</b></p>	2,5 %
La media de utilización de la memoria	$0 \leq X \leq 1$	<p>Si <math>0,75 &lt; x \leq 1</math> ponderación 0 %</p> <p>Si <math>0,50 &lt; x \leq 0,75</math> ponderación 0,5 %</p> <p>Si <math>0,25 &lt; x \leq 0,50</math> ponderación 1 %</p> <p>Si <math>0 &lt; x \leq 0,25</math> ponderación 2 %</p> <p><b>Si <math>x = 0</math> ponderación 2,5 %</b></p>	2,5 %
<b>Total</b>			<b>100 %</b>
<b>Servicio de Migración --Acompañamiento--</b>			
Satisfacción general	<p><math>1 \leq X \leq n</math></p> <p>n = la cantidad de preguntas por el máximo de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados</p> <p>k = la cantidad de preguntas por el menor de los puntajes (1 menor puntaje y 5 mayor) por la cantidad de encuestados</p>	<p><math>X = (n / \text{cantidad de encuestados}) / \text{cantidad de preguntas}</math></p> <p>Si <math>x &lt; 3</math> ponderación 0 % (Insatisfactorio)</p> <p>Si <math>3 \leq x &lt; 4,75</math> ponderación 75 % (Satisfactorio)</p> <p><b>Si <math>4,75 \leq x \leq 5</math> ponderación 100 % (Muy satisfactorio)</b></p>	100 %
<b>Total</b>			<b>100 %</b>

**b) Probar las soluciones alternativas.**

En esta actividad se realizan simulaciones, modelados, prototipos y pilotos, según sea necesario, para probar las soluciones alternativas. Siempre los especialistas en migración deben realizar las pruebas pilotos.

**Artefactos de entrada:** evaluación de alternativas.

**Artefactos de salida:** minuta de reunión (creado,--resultado de las pruebas--).

**c) Evaluar los riesgos de la organización.**

En esta actividad se analizan los riesgos asociados con la implementación de la alternativa libre en la organización a migrar. Es importante que el especialista en migración conozca y transmita los riesgos en los procesos de la institución (estratégicos, sustantivos y de apoyo). En caso de que la implantación de la alternativa libre impacte desfavorablemente en la institución no será seleccionada y se comunicarán los resultados. Se aplica la técnica de tormentas de ideas para evaluar los riesgos de la organización.

**Artefactos de entrada:** listado de riesgos, evaluación de alternativas.

**Artefactos de salida:** minuta de reunión (creado, --análisis de riesgos--).

**2.2.4 Etapa 4. Selección**

En la etapa de **selección** las aplicaciones seleccionadas como alternativas libres más adecuadas son aquellas que tienen un valor igual o superior al 75 % cuando se ejecutan los servicios de Consultoría y Migración por separados, de lo contrario sería un valor igual o superior al 85 %, garantizando como mínimo el 60 % del sumatorio de los atributos específicos. En la Tabla 14 se pueden apreciar los porcentajes de las ponderaciones finales durante cada servicio, resaltando el cambio de los porcentajes de los atributos específicos completitud funcional y satisfacción general cuando se realiza el servicio integral.

**Tabla 14. Porcentaje de ponderaciones finales.** *Elaboración propia*

Atributos	Porcentaje de ponderación Consultoría	Porcentaje de ponderación Migración	Porcentaje de ponderación Consultoría y Migración
Genéricos	20 %	No evaluado	20 %
Específicos	75 %		50 %
	5 %		5 %

### Estrategia de selección de alternativas libres para el proceso de migración a código abierto

		100 %	25 %
<b>Total</b>	<b>100 %</b>	<b>100 %</b>	<b>100 %</b>

#### **a) Seleccionar las alternativas libres.**

En esta actividad se selecciona la alternativa libre más idónea, donde esta satisface las necesidades de los usuarios, resaltando en el listado la alternativa seleccionada. Para ello, se debe tener en cuenta los resultados de las pruebas.

**Artefactos de entrada:** evaluación de alternativas, minuta de reunión (--resultado de las pruebas--).

**Artefactos de salida:** evaluación de alternativas (actualizado).

#### **b) Documentar y comunicar los resultados.**

En esta actividad se documentan y se comunican los resultados del proceso de selección de las alternativas libres en el proceso de migración a código abierto. Para ello, se debe tener en cuenta el análisis de los riesgos.

**Artefactos de entrada:** evaluación de alternativas, minuta de reunión (--análisis de riesgos--).

**Artefactos de salida:** minuta de reunión.

#### **Conclusiones parciales**

- Las entrevistas realizadas a los especialistas en migración permitieron conocer que estos realizan la selección de alternativas libres en el proceso de migración a código abierto basados en sus experiencias individuales.
- El conocimiento de los expertos en migración debe centralice, así como las experiencias adquiridas, a modo de lograr que la selección sea lo más práctico y sencillo posible, y que esté documentado para futuros especialistas.
- La estrategia de selección brinda a los especialistas en migración un gran apoyo durante el proceso de migración a código abierto, obteniendo la alternativa idónea para su posterior instalación en las estaciones de trabajo de los usuarios finales.

## **CAPÍTULO 3. VALIDACIÓN DE LA ESTRATEGIA**

La validación es la encargada de exponer la pertinencia de cada acción ejecutada en el proceso de investigación científica y de los mecanismos lógicos que dan coherencia al diseño. En el presente capítulo se describe el proceso y resultado de la validación de la estrategia de selección de alternativas libres como parte de la evaluación de la misma. Se muestran los resultados de la aplicación del criterio de expertos, a través del método Delphi, a expertos en procesos de migración a código abierto y modelos FLOSS. Se describe el resultado del cuestionario aplicado a los usuarios del MINCOM y se realiza un estudio de caso para seleccionar la alternativas más idónea para la herramienta Microsoft Office Word.

### **3.1 Validación de la estrategia a través del criterio de expertos**

El primer método aplicado fue el criterio de expertos en su variante Delphi (Moráguez, 2006), con el objetivo de determinar la aplicabilidad y el nivel de influencia de la estrategia y de realizar una evaluación de la misma. Se define la aplicabilidad como la capacidad de que sea práctico, posible de utilizar y de adaptarse a las características de las organizaciones, en este caso del proceso de migración. El nivel de influencia está relacionado con el efecto de la estrategia de selección de alternativas libres sobre el proceso de migración.

Para la selección de los expertos se confeccionó un listado de 10 profesores y especialistas con varios años de experiencia en el desarrollo de procesos de migración a código abierto y que, las características de los expertos demuestran que poseen los conocimientos para valorar la propuesta. Se tomaron en consideración los siguientes aspectos: título universitario, categoría docente y científica, años de experiencia en el desarrollo de procesos de migración, haber desempeñado el rol de especialista en migración durante un proceso de migración a código abierto, el nivel de dominio sobre el tema que se encuesta y las fuentes de argumentación.

Para la determinación del coeficiente de competencia de los expertos (**K**) se utilizó la fórmula  $K = (Kc + Ka) * 0,5$ . Donde **Kc** representa el coeficiente de conocimiento que tiene el experto acerca del tema, y se calcula a partir de su propia valoración dentro de una escala del 0 (mínimo conocimiento) al 10 (total conocimiento) multiplicada por 0,1. En el Anexo 10 se muestra la Tabla 22 con la autovaloración de cada experto y el coeficiente de conocimiento correspondiente (autovaloración/10). En la Tabla 15 se muestra

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

un resumen de los valores de **Kc** obtenidos.

**Tabla 15. Resumen de la ubicación de los expertos según Kc.** *Elaboración propia.*

Coeficiente de conocimiento	<b>1.00</b>	<b>0.9</b>	<b>0.8</b>
Cantidad de expertos	1	2	7

Por su parte **Ka** representa el coeficiente de argumentación o fundamentación de los criterios del experto, resultado de la suma de los puntos dados por cada experto en las fuentes de argumentación definidas en la Tabla 16.

**Tabla 16. Grado de influencia de las fuentes de argumentación.** *Fuente: Moráquez, 2006*

No	Fuentes de Argumentación	Alto (A)	Medio (M)	Bajo (B)
1	Estudios teóricos realizados por usted.	0.30	0.20	0.10
2	Experiencia adquirida durante su vida profesional.	0.50	0.40	0.30
3	Conocimiento de investigaciones y/o publicaciones nacionales e internacionales.	0.05	0.04	0.03
4	Conocimiento propio sobre el estado del tema de investigación.	0.05	0.04	0.03
5	Actualización en cursos de posgrado, diplomados, maestrías, doctorado.	0.05	0.04	0.03
6	Intuición	0.05	0.04	0.03

Los valores de **Ka** correspondientes a cada experto se encuentran registrados en la Tabla 23 del Anexo 10. Finalmente se calcularon los coeficientes de competencias (**K**) según la fórmula antes descrita y se definieron los niveles de competencia según los intervalos que se muestran en la Tabla 17.

**Tabla 17. Intervalos para definir la competencia de un experto.** *Fuente: Moráquez, 2006*

Nivel de competencia		
Alto (A)	Medio (M)	Bajo (B)
$1 \geq K \geq 0.8$	$0.8 > K \geq 0.5$	$K < 0.5$

En la Tabla 24 del Anexo 10 se muestran los valores de **Kc**, **Ka**, y **K** para cada experto, donde se refleja que el 100% de los expertos obtuvo un alto nivel de competencia. Se escogieron los 10 expertos teniendo en cuenta que su nivel de competencia sobrepasaba el valor de 0,80. De los 10 expertos elegidos dos son extranjeros, el 30% posee el título académico de Máster mientras que el 30% ostenta el grado científico de Doctor en Ciencias y el 90% de los expertos posee categoría docente. El promedio de años de experiencia en el desarrollo de procesos de migración es de ocho años y el 70% ha participado en más de cinco procesos de migración a código abierto. Además, el 70% de los expertos posee gran experiencia en el rol de especialista en migración.

### **3.1.1 Valoración de la estrategia de selección por los expertos seleccionados**

Para la validación de la estrategia por los expertos se diseñó un cuestionario que se muestra en el Anexo 11: Encuesta criterios de los expertos, el cual les fue enviado en conjunto al material donde se explicaba la estrategia de selección de alternativas libres propuesta. Se presentaron los diez aspectos definidos para valorar la estrategia, cada uno de los cuales cada experto midió con una valoración de Muy adecuado (MA), Bastante adecuado (BA), Adecuado (A), Poco adecuado (PA) e Inadecuado (I), además se permitió dar algún criterio u opinión acerca de la estrategia.

Los resultados de la evaluación realizada por los expertos a cada uno de los aspectos propuestos aparecen en la Tabla 25. Sobre la base de la tabla anterior, se determina la frecuencia absoluta por aspectos (ver Tabla 26) y luego la distribución de frecuencia acumulada de cada aspecto (ver Tabla 27). A partir de la Tabla 27 del Anexo 10, se calcula la distribución de frecuencias relativas acumuladas de cada aspecto (ver Tabla 28).

En la Tabla 18 se realiza el análisis estadístico final donde se calcula: los percentiles de la distribución normal estándar correspondientes a cada una de las frecuencias relativas acumuladas (que se consideran una aproximación de la probabilidad acumulada), la suma algebraica de todos los percentiles anteriores, los puntos de corte (media de los percentiles de cada categoría evaluativa), la suma algebraica de los percentiles dividida por el producto de la cantidad de aspectos sometidos a consulta y la cantidad de categorías evaluativas empleadas(N), la media de los percentiles de cada aspecto sometido a consulta (P), la diferencia (N-P) para cada aspecto analizado (filas). Cada uno de los resultados que se obtienen en

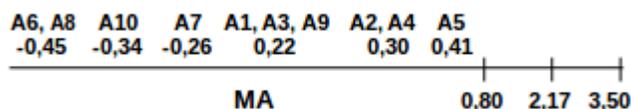
*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

la columna (N-P) se comparan con los puntos de corte y se determina en qué categoría evaluativa se encuentra cada aspecto sometido a consulta de los expertos.

**Tabla 18. Cálculo de los puntos de corte y escala de los aspectos.** *Elaboración propia*

Aspectos	MA	BA	A	PA	P	N-P	Clasificación
A1	0,84	1,28	3,50	3,50	2,28	<b>0,22</b>	<b>MA</b>
A2	0,52	1,28	3,50	3,50	2,20	<b>0,30</b>	<b>MA</b>
A3	0,84	1,28	3,50	3,50	2,28	<b>0,22</b>	<b>MA</b>
A4	0,52	1,28	3,50	3,50	2,20	<b>0,30</b>	<b>MA</b>
A5	0,52	0,84	3,50	3,50	2,09	<b>0,41</b>	<b>MA</b>
A6	1,28	3,50	3,50	3,50	2,95	<b>-0,45</b>	<b>MA</b>
A7	0,52	3,50	3,50	3,50	2,76	<b>-0,26</b>	<b>MA</b>
A8	1,28	3,50	3,50	3,50	2,95	<b>-0,45</b>	<b>MA</b>
A9	0,84	1,28	3,50	3,50	2,28	<b>0,22</b>	<b>MA</b>
A10	0,84	3,50	3,50	3,50	2,84	<b>-0,34</b>	<b>MA</b>
<b>Puntos de corte</b>	<b>0,80</b>	<b>2,12</b>	<b>3,50</b>	<b>3,50</b>	<b>N = 2,50</b>		

Como se puede observar en la Figura 6 se ubican en una recta numérica todos los puntos de corte para cada una de las categorías y los resultados (N-P) para cada uno de los aspectos sometidos a consulta. De ahí que se concluya que todos los aspectos puestos a consideración de los expertos se evalúan de Muy adecuados, lo que resulta muy bueno debido a que proviene de un grupo de personas especialistas en el tema.



**Figura 6. Recta numérica con puntos de corte y clasificación correspondiente.** *Elaboración propia*

Luego se procedió a determinar el grado de consenso de los expertos con los resultados mostrados anteriormente, para lo que se llevaron los resultados a una escala del 1 al 5, donde 5 es Muy adecuado, 4 es Bastante adecuado, 3 es Adecuado, 2 es Poco Adecuado y 1 es Inadecuado. Se confeccionó entonces la matriz correspondiente y se determinó el coeficiente de concordancia (**C**) para cada uno de los

***Estrategia de selección de alternativas libres para el proceso de migración a código abierto***

aspectos, a partir de la expresión  $C=100*(1-Ds/Xm)$  donde **Ds** es la desviación estándar que se

calcula mediante la fórmula  $Ds=\sqrt{1/(n-1)\sum_{i=0}^{10}(Xi-Xm)^2}$  y **Xm** equivale a la media del criterio de los

expertos por indicador y se calcula mediante la fórmula  $Xm=(\sum_{i=0}^{10}CEi)/10$ . Cada coeficiente de concordancia debe tener un valor superior a 75, los valores para cada indicador se muestran en la Tabla 19.

**Tabla 19. Coeficiente de concordancia por aspectos.** *Elaboración propia*

Número de experto	Aspectos									
	1	2	3	4	5	6	7	8	9	10
1	5	5	4	5	5	3	5	5	5	5
2	4	5	5	3	5	5	4	5	5	5
3	5	3	5	5	5	5	5	5	4	5
4	5	5	3	5	4	5	5	5	5	4
5	5	3	3	5	5	4	5	5	5	5
6	5	5	5	4	5	5	5	5	5	5
7	4	4	5	5	5	5	5	5	5	5
8	5	5	5	5	5	4	5	5	5	5
9	5	5	4	5	3	5	5	5	5	5
10	4	5	5	5	5	5	5	5	5	4
<b>Xm</b>	4,7	4,5	4,4	4,7	4,7	4,6	4,9	5,0	4,9	4,8
<b>Ds</b>	0,6	0,85	0,84	0,68	0,68	0,7	0,37	0,0	0,3	0,42
<b>C = 100 * (1-Ds/Xm)</b>	<b>87,23</b>	<b>81,11</b>	<b>80,91</b>	<b>85,53</b>	<b>85,53</b>	<b>84,78</b>	<b>93,47</b>	<b>100,0</b>	<b>93,47</b>	<b>91,25</b>

A partir de los resultados del análisis de concordancia (ver Tabla 19), se estableció que en la población de los 10 expertos seleccionados, los 10 aspectos evaluados de acuerdo con las puntuaciones de las categorías arriba descritas, alcanzan la condición de Muy adecuado y de Bastante adecuado y se obtuvo un grado de concordancia superior a 80 en todos los aspectos, por lo que se considera que los resultados obtenidos son válidos y fundamentan los criterios dados por los expertos.

Esto corrobora que el coeficiente de concordancia total (**Ct**) de la estrategia, que se calcula como se

muestra en la Tabla 20, obtiene un 100% de concordancia, debido a que no se registraron votos negativos. Se consideran votos negativos aquellos que se refieren a un aspecto con la categoría de Poco adecuado o Inadecuado.

**Tabla 20. Coeficiente de concordancia total de la estrategia.**

Votos negativos	Votos totales	Coeficiente de concordancia total
Vn	Vt	$Ct = (1 - Vn / Vt) * 100$
0	100	100

El consenso entre los expertos respecto a la evaluación de la estrategia de selección de alternativas libres permite confirmar su aplicabilidad y correcto nivel de influencia durante la ejecución de un proceso de migración. Este proceso de validación permitió perfeccionar y enriquecer la propuesta a partir de los criterios establecidos por los expertos seleccionados. De los resultados obtenidos se puede interpretar que:

- La estrategia propuesta es de mucha importancia durante un proceso de migración a código abierto, y su correcta aplicación depende en gran medida de un aumento en el grado de aceptación por parte de los usuarios finales en dicho proceso.
- Las actividades definidas evita que los especialistas en migración presenten alguna dificultad a la hora de realizar las tareas durante el proceso de migración.
- La elaboración de los artefactos garantiza apoyo a los especialistas en migración, contribuyendo a la calidad de los procesos en el centro CESOL.
- Los atributos genéricos y específicos garantizan la selección de una alternativa más adecuada en cada estación de trabajo, aumentando el grado de satisfacción de los usuarios finales en las instituciones.

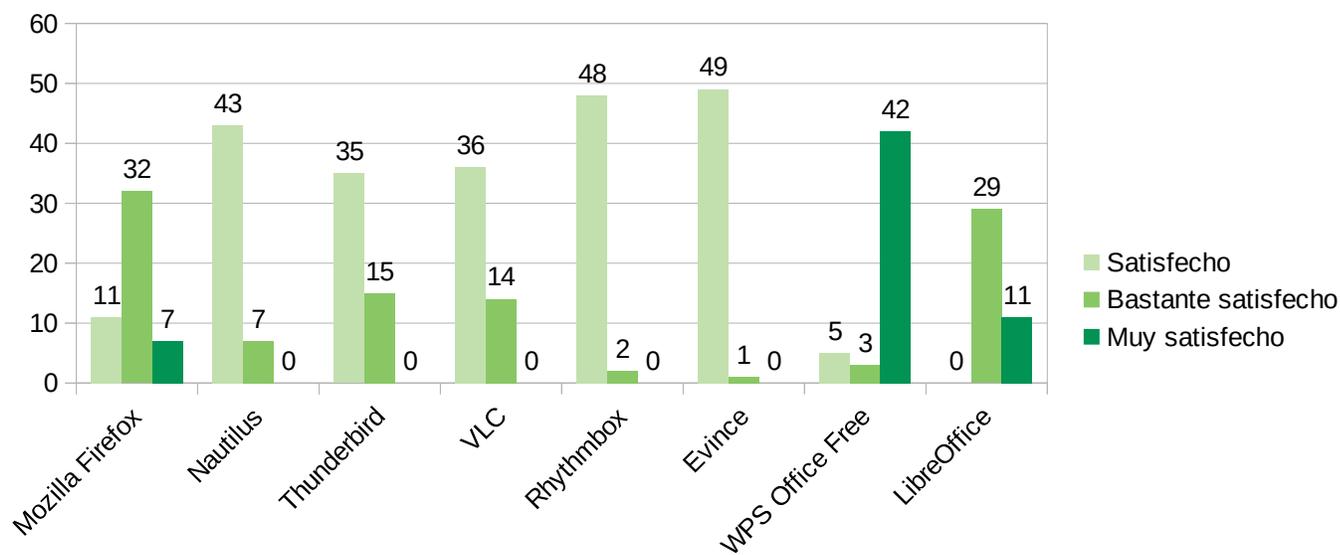
### **3.2 Resultados del cuestionario de satisfacción de los usuarios en el MINCOM**

En aras de validar la variable independiente “satisfacción de los usuarios” se aplicó un cuestionario de satisfacción (ver Anexo 12: Cuestionario aplicado para determinar nivel de satisfacción) a los usuarios finales del Ministerio de Comunicaciones (MINCOM) con el objetivo de conocer el grado de satisfacción de estos con algunas de las aplicaciones libres fundamentales de la distribución cubana GNU/Linux Nova Escritorio 6.0.

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

El Ministerio de Comunicaciones es el organismo encargado de proponer, y una vez aprobada, dirigir y controlar la política del Estado y el Gobierno para el Sistema Único de Comunicaciones del País (MINCOM, 2018). En el 2017 se realizó una migración a código abierto en la red de navegación del ministerio, como se puede apreciar en Anexo 14: Acta de aceptación de la Migración en el MINCOM que se migraron 249 estaciones de trabajo, de las cuales 20 eran computadoras de bajas prestaciones. Actualmente, el MINCOM tiene contratado el servicio de soporte con la Universidad donde se solicitó la instalación de una nueva distribución debido a que se han realizado mejoras tecnológicas y existen aplicaciones desactualizadas, lo que se decidió realizar una actualización de la distribución cubana GNU/Linux Nova Escritorio.

El cuestionario fue aplicado a una muestra de 50 trabajadores del MINCOM distribuido en diferentes áreas del ministerio. Como se puede apreciar en la Figura 7, se muestra el resultado del nivel de satisfacción de los usuarios con las alternativas libres Mozilla Firefox, Nautilus, Thunderbird, VLC, Rhythmbox, Evince, WSP Office Free y LibreOffice. Las aplicaciones con mayor grado de satisfacción fueron Mozilla Firefox y WSP Office Free. En el caso de LibreOffice hubo 10 votos de cuales, 8 eran pocos insatisfechos y 2 insatisfechos.



**Figura 7. Nivel de satisfacción por alternativas.** *Elaboración propia*

Por lo general, más del 90 % de los trabajadores en el MINCOM han trabajado con Microsoft Office

durante más de 5 años. Esto conlleva a un elemento desfavorable, por lo que se hace necesario trabajar en la sensibilización para fortalecer los conocimientos antes de introducir las tecnologías de código abierto en el ministerio. Demostrando en la Figura 7 que los usuarios se sienten muy satisfechos con la alternativa libre WSP Office Free debido a su semejanza con Microsoft Office.

Comparando los resultados obtenidos de la validación del estudio de caso respecto a migraciones realizadas anteriormente, se refleja un incremento en el grado de satisfacción de los usuarios; debido a que existen cambios en la aceptación con las alternativas libres de la distribución cubana GNU/Linux Nova propuestas por los especialistas en migración. La alternativa libre que demuestra un aumento en la aceptación de los usuarios en el MINCOM es el paquete ofimático LibreOffice Writer. Además, la técnica que se utiliza para medir el grado de satisfacción de los usuarios es la encuesta, como se propone en la estrategia teniendo en cuenta los criterios establecidos por los usuarios.

### **3.3 Estudio de caso: Alternativa libre para la herramienta privativa Microsoft Office Word**

Para ilustrar la pertinencia de la estrategia de selección de alternativas libres se propone realizar el proceso de selección de la alternativa libre más adecuada para la herramienta Microsoft Office Word 2016. En este caso se debe buscar el equivalente libre de la herramienta Microsoft Office Word 2016 para el Ministerio de Comunicaciones (MINCOM). En la red interna del ministerio los usuarios emplean Microsoft Office Word para realizar las actividades siguientes:

*1) Realizar informes sobre el cumplimiento de los planes:*

- RF1- Trabajo con estilos de páginas y carácter.
- RF2- Manejo de tablas.
- RF3- Trabajo con herramientas de formato.
- RF4- Generación de gráficos de barras y pastel.
- RF5- Copiar desde el Microsoft Excel hacia el Word tablas de forma directa.

*2) Realizar cartas en formato electrónico para un grupo determinado de usuarios:*

- RF6- Trabajo con estilos de páginas y carácter.
- RF7- Trabajo con las herramientas de formato.
- RF8- Empleo de la utilidad de combinación de correspondencia.

*3) Imprimir documentos:*

- Empleo de la utilidad de vista previa para ajustar las características de impresión.
- Imprimir directamente.

En la red navegación del ministerio se estableció utilizar la distribución cubana GNU/Linux Nova Escritorio o Ligero, lo que se necesita realizar las funcionalidades anteriores y opcionalmente la impresión de los documentos sobre tecnologías libres. A continuación se aplica cada actividad de la estrategia de acuerdo a lo recopilado anteriormente para cuando los servicios de Consultoría y Migración se realizan por separados. Normalmente, siempre se realiza un servicio de Consultoría para recopilar la información necesaria y posteriormente se ejecuta el servicio de Migración. En la Tabla 2 y Tabla 3 del capítulo 1 de la presente investigación se hace referencia a cuándo y en qué actividad aplicar la estrategia.

### **3.3.1 Definición**

En esta etapa se definieron las necesidades de los usuarios del MINCOM con respecto al software privativo Microsoft Office Word 2016.

#### **a) Definir las necesidades de los usuarios.**

Luego de la aplicación de las encuestas y las entrevistas realizadas en las áreas durante las actividades anteriores de la consultoría de migración a código abierto se constituye el artefacto “*Necesidades de los usuarios*”. Este artefacto contiene las funcionalidades de *Realizar informes sobre el cumplimiento de los planes*, *Realizar cartas en formato electrónico para un grupo determinado de usuarios* e *Imprimir documentos*. Esta última funcionalidad no se requiere de un proceso de evaluación formal debido a que en el ministerio las impresoras están conectadas a la red interna, pero se buscará una solución para 3 usuarios que lo solicitaron personalmente sin afectar los procesos estratégicos, sustantivos y de apoyo por el MINCOM. Las computadoras tienen 1 GB de RAM y más, excepto 20 estaciones de trabajos.

#### **b) Definir el objetivo de medición.**

Los especialistas en migración con el apoyo del artefacto “Guía para el análisis de decisiones” definen el objetivo de medición. Para el presente estudio de caso es analizar las alternativas libres del software privativo Microsoft Office Word 2016 en el proceso de migración a código abierto, examinando su idoneidad, con el fin de seleccionarlas.

#### **c) Definir y clasificar los atributos para la evaluación de las alternativas libres.**

El artefacto “Guía para el análisis de decisiones” define y clasifica los atributos para la evaluación de las alternativas libres. Los atributos pueden ser mejorados por los especialistas expertos en migración y se

podrán añadir nuevos atributos si es necesario pero respetando la propuesta.

**d) Definir el rango de los atributos y los métodos.**

El artefacto “Guía para el análisis de decisiones” define el rango los atributos para la evaluación de las alternativas libres. Esta actividad puede ser mejorada por los especialistas expertos en migración, siempre y cuando la evaluación sea sencilla y esté bien documentada.

**e) Evaluar los atributos y su importancia relativa.**

El artefacto “Guía para el análisis de decisiones” evalúa los atributos y su importancia relativa. La importancia puede ser mejorada por los especialistas expertos en migración, solo que los atributos específicos completitud funcional y satisfacción de los usuarios siempre deben tener mayor porcentaje de ponderación de importancia sobre los restante atributos.

### **3.3.2 Identificación**

En esta etapa se determinaron las características generales de las alternativas libres identificadas LibreOffice Writer, Abiword y WPS Office Free.

**a) Identificar las alternativas libres.**

En el levantamiento de las posibles alternativas libres para la aplicación privativa se tuvo en cuenta las necesidades de los usuarios identificadas y el listado de alternativas general del departamento SIMAYS. Donde se obtuvo que LibreOffice Writer, Abiword y WPS Office Free son alternativas libres para el software privativo Microsoft Office Word.

**b) Describir las alternativas libres.**

La información recopilada para la descripción de las tarjetas de identificación de las alternativas fue tomada de los sitios oficiales. Las tarjetas de identificación de LibreOffice Writer, Abiword y WPS Office Free se pueden apreciar en Tabla 29, Tabla 30 y Tabla 31.

### **3.3.3 Evaluación**

En la etapa se evaluaron las alternativas libres LibreOffice Writer, Abiword y WPS Office Free teniendo en cuenta los atributos genéricos y específicos.

**a) Evaluar las alternativas libres.**

## Estrategia de selección de alternativas libres para el proceso de migración a código abierto

A partir de las funcionalidades empleadas en MINCOM en el manejo de Microsoft Office Word se procede a seleccionar (ver Tabla 21) la alternativa de código abierto más adecuada a esta solución teniendo en cuenta los atributos definidos en el capítulo anterior. La información fue recopilada de los sitios oficiales de las aplicaciones tales como: LibreOffice Writer (<https://es.libreoffice.org/>), WPS Office Free (<https://www.wps.com>) y Abiword (<https://www.abisource.com/>).

Se evaluaron las alternativas libres LibreOffice Writer, Abiword y WPS Office Free teniendo en cuenta toda la información recopilada. Como se puede apreciar en Tabla 1 se muestran los resultados de la evaluación. Los servicios fueron ejecutados de forma independiente, por lo que se realizó la evaluación inicialmente de las alternativas libres sin el análisis de la satisfacción general de los usuarios.

**Tabla 21. Alternativa libre para Microsoft Office Word.** *Elaboración propia.*

NE: no evaluado

Atributos	LO Writer		Abiword		WPS Office Free	
<b>Genéricos. Servicio de Consultoría</b>						
Licencia	4	2	5	3	1 <sup>2</sup>	0,5
Antigüedad	5	1	5	1	5	1
Propietario	5	2	5	2	5	2
Documentación	5	2	3	2	3	2
Soporte	5	2	3	1	5	2
Popularidad	5	3	2	0,5	1	0
Comunidad	5	3	2	0,5	1	0
Disponibilidad del código fuente	5	3	5	3	1	0
Actividad de errores	0,4	1	0,7	0,5	1	0
<b>Subtotal</b>		<b>19 %</b>		<b>13,5 %</b>		<b>7,5 %</b>
<b>Específicos. Calidad externa. Servicio de Consultoría</b>						
Compleitud funcional	0	<b>75 %</b>	0,4	<b>30 %</b>	0	<b>75 %</b>
<b>Específicos. Calidad externa. Servicio de Consultoría</b>						
La media de utilización del procesador	0,8	0	0,2	2	0,6	0,5
La media de utilización del memoria	0,12	2	0,03	2	0,08	2

2 La licencia de WPS Office no está clara, en su web oficial se ofrece gratuitamente, pero sin código fuente disponible.

### *Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

<b>Subtotal</b>		<b>2 %</b>		<b>4 %</b>		<b>2,5 %</b>
<b>Total</b>		<b>96 %</b>		<b>47, 5%</b>		<b>85 %</b>
<b>Específicos. Calidad en uso. Servicio de Migración</b>						
Satisfacción general	3,2	<b>75 %</b>	NE	<b>NE</b>	4,85	<b>100 %</b>

Durante el servicio de Consultoría, en Tabla 21 se puede constatar que Abiword podría haber sido un excelente candidato para su instalación en Nova Ligero, debido al poco uso de los recursos de hardware del procesador y la memoria, pero esta herramienta no cumple (30 %) con los requisitos especificados y por consiguiente la alternativa libre es rechazada. Además, el sumatorio de los atributos genéricos de la alternativa libre WPS Office Free es muy mala, pero no debe ser rechazada hasta finalizar la evaluación porque cumple con un 75 % de los requisitos especificados.

En el servicio de Migración se constató que los usuarios del MINCOM estaban 100 % satisfecho con la alternativa WPS Office Free, sin embargo un 75 % con LibreOffice Writer.

#### ***b) Probar las soluciones alternativas.***

Las soluciones LibreOffice Writer y WPS Office Free fueron probadas en la distribución cubana Nova Escritorio 5,0 donde los resultados fueron satisfactorios durante ambos servicios. En el ministerio existen 20 estaciones de trabajo con Nova Ligero, los resultados también fueron satisfactorios.

#### ***c) Evaluar los riesgos de la organización.***

En el servicio de Consultoría se evaluó el riesgo de proponer la alternativa libre WPS Office Free sin la disponibilidad de un código fuente. Esto implica que los especialistas del centro CESOL no pueden brindar continuidad al producto. Además, como existen algunos usuarios insatisfecho con la alternativa LibreOffice Writer se planificaron capacitaciones a los usuarios.

### **3.3.4 Selección**

En la etapa se seleccionaron las alternativas libres LibreOffice Writer y WPS Office Free como alternativas más idóneas.

#### ***a) Seleccionar las alternativas libres.***

## *Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

En Tabla 21 se puede constatar que las alternativas LibreOffice Writer y WPS Office Free tienen 96 % y 85 % respectivamente entre el sumatorio de los atributos genéricos y específicos, excepto el atributo de calidad en uso. Ambas, alternativas fueron seleccionadas porque poseen excelentes porcentajes en la evaluación.

### ***b) Documentar y comunicar los resultados.***

En el Plan de Migración del MINCOM del servicio de Consultoría quedan identificadas las alternativas WPS Office Free y LibreOffice Writer como alternativas libres seleccionadas reflejando los riesgos de la selección. Posteriormente, en el servicio de Migración fueron instaladas las aplicaciones seleccionadas.

### **Conclusiones parciales**

- La aplicación del criterio de expertos, a través del método Delphi, a expertos en modelos FLOSS y en procesos de migración a código abierto permitió corroborar la aplicabilidad y correcto nivel de influencia de la estrategia durante la ejecución de un proceso de migración.
- La aplicación del cuestionario para conocer el grado de satisfacción de los usuarios del MINCOM respecto a las alternativas libres corroboró que los usuarios del ministerio están satisfechos con las alternativas libres instaladas en Nova Escritorio y Nova Ligerio.
- La aplicación de la estrategia de selección de alternativas libres propuesta, a través del estudio de caso, reflejó que los usuarios están más satisfechos con la alternativa WPS Office Free.

## **CONCLUSIONES GENERALES**

El desarrollo de la presente investigación permitió arribar a las siguientes conclusiones:

- El estudio realizado a los modelos y metodologías FLOSS, al proceso de migración a código abierto en Cuba, a las Normas ISO/IEC 25022: 2016 y 25023: 2017, y el área de proceso DAR de CMMI nivel 3 permitieron determinar las etapas, actividades, atributos, métricas, métodos y artefactos que conforman la estrategia de selección de alternativas libres.
- El diagnóstico realizado a la ejecución de la evaluación del software de escritorio, a través de la aplicación de una entrevista a un grupo de especialistas en migración del centro CESOL, permitió identificar la inexistencia de atributos, métricas y métodos para la selección de las alternativas libres más idóneas en el proceso de migración de código abierto.
- La estrategia de selección de alternativas libres está conformada por cuatro etapas con un conjunto de acciones que posibilitan una correcta planificación y ejecución de la evaluación del software de escritorio durante un proceso de migración a código abierto.
- La aplicación del criterio de expertos en su variante Delphi permitió validar a la estrategia de selección de alternativas libres de muy adecuada para ser utilizada en un proceso de migración a código abierto.
- La aplicación de la estrategia de selección de alternativas libres propuesta, a través del estudio de caso, reflejó que los usuarios están más satisfechos con la alternativa WPS Office Free debido a su semejanza con la aplicación privativa Microsoft Office Word, y que estos de manera general están satisfechos con las alternativas libres propuestas en el MINCOM.

## **RECOMENDACIONES**

Para dar continuidad a la presente investigación se recomienda:

- Desarrollar un sistema inteligente para el proceso de selección de alternativas libres, compuesto por una base de conocimientos que se nutre a partir de los resultados de la selección de alternativas libres en procesos de migraciones anteriores y de un proceso de vigilancia tecnología en el área de los FLOSS.
- Profundizar en el atributo específico de calidad externa completitud funcional para la selección de las alternativas libres en la etapa definición de la estrategia.

## REFERENCIAS BIBLIOGRÁFICAS

ADEWUMI, A., MISRA, S. y OMOREGBE, N. 2013. *A Review of Models for Evaluating Quality in Open Source Software*. International Conference on Electronic Engineering and Computer Science. IERI Procedia 4, pp 88 – 92.

AKBARI, M. y HASHEMI, S. 2014. *Evaluation of Free/Open Source Software Using OSMM Model Case Study: WebGIS and Spatial Database*. ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No.11 , ISSN : 2322-5157. [http:// www.ACSIJ.org](http://www.ACSIJ.org)

ARDAGNA, C.A., DAMIANI, E., y FRATI, F. 2007. *FOCSE: An OWA-based Evaluation Framework for OS Adoption in Critical Environments*. Proc. Third IFIP WG 2.13 International Conference on Open Source Systems, Limerick, Ireland, pp. 3-16.

CARBON, R., CIOLKOWSKI, M., HEIDRICH, J., JOHN, I., y MUTHIG, D. 2007. *Evaluating Open Source Software through Prototyping*, in St.Amant, K., and Still, B. (Eds.) Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives (Information Science Reference), pp. 269-281.

CMMI-DEV, v1.3. 2010. *Mejora de los procesos para el desarrollo de mejores productos y servicios*.

CRUZ, D., WIELAND, T., y ZIEGLER, A. 2006. *Evaluation criteria for free/open source software products based on project analysis*. Software Process: Improvement and Practice, 11(2)

CABANO, M., MONTI, C., y PIANCASTELLI, G. 2007. *Context-Dependent Evaluation Methodology for Open Source Software*. Proc. Third IFIP WG 2.13 International Conference on Open Source Systems (OSS 2007), Limerick, Ireland, pp. 301-306.

DEPREZ, J. 2010. *QualOSS Final Activity Report*. [en línea]. [Consulta: 7 marzo 2018]. Disponible en: [https://www.cetic.be/IMG/pdf/WP7\\_QualOSS\\_Final\\_Activity\\_Report\\_Submitted.pdf](https://www.cetic.be/IMG/pdf/WP7_QualOSS_Final_Activity_Report_Submitted.pdf)

DUIJNHOUWER, F., y WIDDOWS, C. 2003. *Open Source Maturity Model', Capgemini Expert Letter*. [https://www.google.com.cu/url?](https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0ahUKEwiMi_Lcs7rZAhXEmuAKHZ12AZw4ChAWCDIwAg&url=https%3A%2F%2Fwww.ordina.nl%2F~%2Fmedia%2Ffiles%2Fordinanl%2Fblogs%2Fopen%2520source%2520maturity%2520model%2520v2.pdf&usg=AOvVaw3EhGwJA1lxKoKFv3sDnhqp)

[sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0ahUKEwiMi\\_Lcs7rZAhXEmuAKHZ12AZw4ChAWCDIwAg&url=https%3A%2F%2Fwww.ordina.nl%2F~%2Fmedia%2Ffiles%2Fordinanl%2Fblogs%2Fopen%2520source%2520maturity%2520model%2520v2.pdf&usg=AOvVaw3EhGwJA1lxKoKFv3sDnhqp](https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0ahUKEwiMi_Lcs7rZAhXEmuAKHZ12AZw4ChAWCDIwAg&url=https%3A%2F%2Fwww.ordina.nl%2F~%2Fmedia%2Ffiles%2Fordinanl%2Fblogs%2Fopen%2520source%2520maturity%2520model%2520v2.pdf&usg=AOvVaw3EhGwJA1lxKoKFv3sDnhqp)

FREE SOFTWARE FOUNDATION. 2018. *What is free software?* The Free Software Definition. <https://www.gnu.org/philosophy/free-sw.html>.

FRITZ E., MONTEJANO G., y GARCÍA P. 2017. *Selección de Atributos de Calidad de Datos en Sistemas de Gestión de Aprendizaje bajo la Familia de Normas ISO/IEC 25000*. XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). ISBN: 978-987-42-5143-5. p. 627-631.

GLOTT R., GROVEN A.-K., HAALAND K. y TANNENBERG A. 2010. *Quality Models for Free/Libre Open Source Software Towards the "Silver Bull"*. s.l.: IEEE Xplore Digital Library, 2010. Pp 439–446.

GOLDEN, B. 2004. *Succeeding with Open Source*.

GONZÁLEZ, D., ALEMÁN, P., DÍAZ, L., y ACOSTA, E. 2007. *Impacto de la informatización en la sociedad y estomatología cubanas*. Rev Cubana Estomatol [online]. vol.44, n.2 [citado 2018-01-30], pp. 0-0. ISSN 1561-297X. Disponible en: <[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S0034-75072007000200006&lng=es&nrm=iso](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0034-75072007000200006&lng=es&nrm=iso)>.

GORHAN F., HETTINGER J., SCHULZ J. y WOLTER M. 2012. *Development of a Model Evaluating the Maturity of Open Source Software*.

HOUAICH, Y., y BELAISSAOUI, M. 2015. *Towards a New Evaluation Model to Improve Open Source Software - Application in Moroccan SMEs*. International Journal of Computer Science and Mobile Computing, Vol.4 Issue.7, pg. 364-374. ISSN 2320–088X

ISO/IEC 25022:2016. *Ingeniería de sistemas y software - Calidad de sistemas y software - Requisitos y Evaluación (SQuaRE) - Medición de la calidad en uso*.

ISO/IEC 25023:2017. *Ingeniería de software y sistemas - Requisitos de la calidad y evaluación de software y sistemas (SquaRE) - Medición de la calidad del producto de software y del sistema*.

KOPONEN, T., y HOTTI, V. 2004. *Evaluation framework for open source software*. Proc. Software Engineering and Practice (SERP), Las Vegas, Nevada, USA.

LAVAZZA, L. 2007. *Beyond Total Cost of Ownership: Applying Balanced Scorecards to Open-Source Software*. Proc. International Conference on Software Engineering Advances (ICSEA) CapEsterel, French Riviera, France, pp. 74-74.

MAJCHROWSKI, A., y DEPREZ, J. 2008. *An operational approach for selecting open source components in a software development project*. Proc. 15th European Conference, Software Process Improvement

(EuroSPI), Dublin, Ireland.

MARTÍNEZ, G. 2017. *Software libre. Para la Educación y la Pequeña Empresa. Facultad de Ingeniería de Sistemas Computacionales*,. Universidad Tecnológica de Panamá. Marzo.

MASLINUX. 2017. *¿Cuántos tipos de licencias de software abiertas existen?* [en línea]. [Consulta: 29 mayo 2018]. Disponible en: <http://maslinux.es/cuantos-tipos-de-licencias-de-software-abiertas-existen/>

MORÁGUEZ, A. 2006. *El método Delphi*. El método Delphi [en línea]. [Consulta: 4 junio 2018]. Disponible en: <http://www.gestiopolis.com/el-metodo-delphi/>.

MINCOM. *Misión del Ministerio de Comunicaciones*. [en línea]. [Consulta: 20 junio 2018]. Disponible en: <http://www.mincom.gob.cu/misi%C3%B3n>.

OPENBRR. 2005. *Business Readiness Rating for Open Source. BRR 2005 - RFC 1*. SpikeSource, the Center for Open Source Investigation at Carnegie Mellon West, and Intel Corporation. [https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CMU\\_US/C050728W.pdf](https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CMU_US/C050728W.pdf)

OPENSOURCE. [en línea]. [Consulta: 8 mayo 2018]. Disponible en: <https://opensource.org/licenses>

PANI, F., CONCAS, G., SANNA, D., y CARROGU, L. 2010. *The FAME Approach: an assessing methodology*. Proceedings of the 9th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS. ISBN: 978-954-92600-2-1

PÉREZ, Y. 2008. *Metodología para la migración a software libre de las universidades del Ministerio de Educación Superior (MES)*. Tesis de grado. La Habana: Universidad de las Ciencias Informáticas.

PÉREZ, Y., y otros. 2009. *Guía Cubana de Migración a plataformas de Código Abierto* [en línea]. S.l.: s.n. [Consulta: ]. Disponible en: [http://uptparia.edu.ve/documentos/software-libre/guia\\_cubana\\_0.2.pdf](http://uptparia.edu.ve/documentos/software-libre/guia_cubana_0.2.pdf)

PÉREZ, Y., y otros. 2014. *Buenas prácticas para la migración a código abierto*. Centro de Software Libre. Universidad de las Ciencias Informáticas. Cuba.

PÉREZ, Y. 2015. *Estrategia para la migración a aplicaciones de código abierto*. Centro de Software Libre. Universidad de las Ciencias Informáticas. Cuba.

PETRINJA, E., SILLITTI, A., y SUCCI, G. 2010. *Comparing OpenBRR, QSOS, and OMM Assessment Models*. International Federation for Information Processing. AICT 319, pp. 224–238.

PETRINJA, E., NAMBAKAM, R., y SILLITTI, A. 2009. *Introducing the OpenSource Maturity Model*. Proc.

ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (FLOSS '09), Vancouver, Canada, pp. 37-41.

PETRINJA, E., y SUCCI, G. 2012. *Assessing the Open Source Development Processes Using OMM*.

Pierra, A. 2011. *Nova, distribución cubana de GNU/Linux. Reestructuración estratégica de su proceso de desarrollo*.

POLANCIC, G., y HORVAT, R.V.2004. *A Model for Comparative Assessment Of Open Source Products*. Proc. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, USA.

QSOS. 2013. *Qualification and Selection of Open Source software*.

SAMOLADAS, I., GOUSIOS, G., SPINELLIS, D., y STAMELOS, I. 2008. *The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation*. Proc. Fourth IFIP WG 2.13 International Conference on Open Source Systems (OSS 2008), Milano, Italy.

SORIA, I. 2017 *Apropiación tecnológica y cuerpos femeninos: el camino del código abierto*. [en línea]. [Consulta: 3 abril 2018]. Disponible en: <https://www.researchgate.net/publication/317057585>.

SOTO, M., y CIOLKOWSKI, M. 2009. *The QualOSS Process Evaluation: Initial Experiences with Assessing Open Source Processes*. European Conference on Software Process Improvement. EuroSPI, CCIS 42, pp 105–116.

TAIBI, D., LAVAZZA, L., y MORASCA, S. 2007. *OpenBQR: a framework for the assessment of OSS*. Proc. Third IFIP WG 2.13 International Conference on Open Source Systems (OSS 2007), Limerick, Ireland, pp. 173-186.

VIJAYA, P., Chander, S. y Raju, G. 2017. *USQO-FOSS quality model: Utilization based Software Quality Observatory for evaluation of Free and Open Source Software*. FREE AND OPEN SOURCE SOFTWARE CONFERENCE (FOSSC-17). ISSN: 1813-419X

WHEELER, D.A. *How to Evaluate Open Source Software / Free Software (OSS/FS) Programs*. [en línea]. [Consulta: 6 marzo 2018]. Disponible en: [https://www.dwheeler.com/oss\\_fs\\_eval.html](https://www.dwheeler.com/oss_fs_eval.html)

WOODS, D., y GULIANI, G. 2005. *Open Source for the Enterprise: Managing Risks Reaping Rewards*.

## ANEXOS

### Anexo 1: Listado de modelos FLOSS

No.	Modelos FLOSS	Autor(es)	Año
1.	Capgemini Open Source Maturity Model (C-OSMM)	Duijnhouwer, Widdows	2003
2.	Evaluation Framework for Open Source Software (EF-OSS)	Koponen, Hotti	2004
3.	A Model for Comparative Assessment of Open Source Products (MCA-OSP)	Polančič, Horvat	2004
4.	Navica Open Source Maturity Model (N-OSMM)	Golden	2005
5.	Woods and Guliani's OSMM (OSMM)	Woods, Guliani	2005
6.	Open Business Readiness Rating (OpenBRR)	SpikeSource	2005
7.	Atos Origin Method for Qualification and Selection of Open Source Software (QSOS)	Atos	2006
8.	Evaluation Criteria for Free/Open Source Software Products (EC-FLOSSP)	Cruz, Wieland, Ziegler	2006
9.	Evaluate Open Source Software / Free Software (OSS/FS) Programs (E-OSS/FS-P)	Wheeler	2006
10.	Observatory for Innovation and Technological transfer on Open Source software (OITOS)	Cabano, Monti, Piancastelli	2007
11.	Framework for OS Critical Systems Evaluation (FOCSE)	Ardagna, Damiani, Fрати	2007
12.	Balanced Scorecards for OSS (BSC-OSS)	Lavazza,	2007
13.	Open Business Quality Rating (OpenBQR)	Taibi, Lavazza	2007
14.	Evaluating OSS through Prototyping (EOSS-P)	Carbon, Ciolkowski, Heidrich, John, Muthig	2007
15.	A Comprehensive Approach for Assessing Open Source	Ciolkowski, Soto	2008

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

	Projects (QualOSS)		
16.	Software Quality Observatory for Open Source Software (SQO-OSS)	Samoladas, Gousios, Spinellis, Stamelos	2008
17.	An operational approach for selecting open source components in a software development project (OA-SOSC)	Majchrowski, Deprez	2008
18.	QualiPSo trustworthiness model (QualiPSo)	del Bianco, Lavazza	2008
19.	OpenSource Maturity Model (OMM)	Petrinja, Nambakam, Sillitti	2009
20.	The FAME Approach: an assessing methodology (FAME)	Pani, Concas, Sanna, Carrogu	2010
21.	Easiest Open Source Evaluation Model (E-OSSEM)	Houaich, Belaissaoui	2015
22.	Quality factors based on Structural Equation Model for Open Source Software (USQO-FOSS)	Vijaya, Chander, Raju	2017
23.	OSSpal: Finding and Evaluating Open Source Software (OSSpal)	Wasserman, Guo, McMillian, Qian, Wei, Xu	2017

**Anexo 2: Modelos FLOSS vs Criterios**

No.	Categorías/Áreas	Peso	Importancia	Criterios/Subcriterios
1.	4	[1;5] 5 (Excelente)	N/A	16 (13 Punto de vista)
2.	3	N/A	N/A	16
3.	N/A	N/A	100%	6/18
4.	6	[1;10]	N/A	19
5.	3	[1;3] 3 (Muy maduro)	N/A	12
6.	12	100%	1 (mayor) - 12 (menor)	N/A
7.	3	[0;2] 2	[0,1,3] 3 (Crítico)	4/16 (Software), 3 (Licencia) y 5 (Comunidad)

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

		(Completamente cubierta)		
8.	6	N/A	N/A	25
9.	N/A	N/A	N/A	13
10.	3	N/A	N/A	6
11.	6	N/A	N/A	N/A
12.	4	N/A	N/A	14
13.	5	N/A	[0;9]	N/A
14.	5	N/A	N/A	N/A
15.	4	N/A	N/A	7
16.	2	N/A	N/A	10
17.	2	N/A	N/A	4
18.	4	N/A	N/A	29
19.	3	N/A	N/A	25
20.	2	N/A	N/A	N/A
21.	4	[0;5] 5 (Excelente)	N/A	12
22.	2	N/A	N/A	10
23.	7	N/A	N/A	N/A

**Anexo 3: Modelos FLOSS vs Métricas**

No.	Métricas	Peso	Importancia	Normalización	Puntuación/ Funcionalidad
1.	N/A	N/A	N/A	N/A	N/A
2.	N/A	N/A	N/A	N/A	N/A
3.	10	[1;3] 3 (Alto afecto)	N/A	Por valores altos o valores bajos	[1;9]
4.	40	[1;10]	N/A	N/A	N/A
5.	N/A	N/A	N/A	N/A	N/A
6.	28	100%	1 (mayor) – 28	[1;5] (sin categoría)	[1;3]

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

			(menor)	5 (Excelente, más 96%)	3 (Muy importante)
7.	N/A	N/A	N/A	N/A	[0,1,3] 3 (Requerida)
8.	28	N/A	N/A	N/A	N/A
9.	N/A	N/A	N/A	N/A	N/A
10.	9	[1;10]	Más críticas	N/A	N/A
11.	18	N/A	N/A	Si	N/A
12.	N/A	N/A	N/A	N/A	N/A
13.	N/A	N/A	N/A	[1;100]	[1;100] 100 (Completamente implementada)
14.	N/A	N/A	N/A	N/A	N/A
15.	116	100%	[0;1] Importancia [-3;2] Estado	N/A	N/A
16.	43	[0;N]	N/A	N/A	N/A
17.	N/A	N/A	N/A	N/A	N/A
18.	90	100%	N/A	N/A	N/A
19.	630	[1;4]	N/A	[1;4]	N/A
20.	N/A	N/A	N/A	N/A	N/A
21.	N/A	N/A	N/A	N/A	N/A
22.	26	[1;100]	[0;5] 5 (Significativo)	N/A	N/A
23.	N/A	N/A	N/A	N/A	N/A

**Anexo 4: Modelos por atributos (tabla 1)**

Modelos	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
<b>Atributos</b>												
<b>Años</b>	x	x	x	x	x		x	x	x		x	
<b>Licencia</b>	x	x			x	x	x	x	x	x		x
<b>Puntos de venta / Marketing</b>	x			x				x	x			

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Comunidad de desarrolladores	x		x		x	x	x	x		x	x	
Jerarquías humanas	x											
Colaboración con otros productos	x											
Modularidad	x											x
Estándares	x											
Comunidad de usuarios	x	x									x	
Penetración de mercado	x											
Soporte	x			x	x	x		x	x	x	x	
Probado	x				x							
Arquitectura	x				x	x						x
Despliegue	x											
Lenguaje		x										
Líder de proyecto		x			x	x	x					x
Votos del usuario			x									
Vitalidad			x		x						x	
Popularidad			x		x		x					
Actividad de errores			x				x	x			x	
Documentación				x	x	x	x			x	x	x
Funcionalidad	x			x		x	x	x	x			
Calidad del producto				x							x	x
Formación				x								
Integración				x	x							
Usabilidad						x		x				x
Seguridad						x		x				
Rendimiento						x		x				
Escalabilidad						x		x				
Confiabilidad							x	x				x
Mantenimiento							x	x				x
Costos					x		x	x				x
Flexibilidad								x				
Interoperabilidad								x		x		
Satisfacción del cliente												x

Compartición												
--------------	--	--	--	--	--	--	--	--	--	--	--	--

**Anexo 5: Modelos por atributos (tabla 2)**

<b>Modelos</b>	<b>13.</b>	<b>14.</b>	<b>15.</b>	<b>16.</b>	<b>17.</b>	<b>18.</b>	<b>19.</b>	<b>20.</b>	<b>21.</b>	<b>22.</b>	<b>23.</b>
<b>Atributos</b>											
<b>Años</b>	x				x				x		
<b>Licencia</b>	x				x	x	x	x	x	x	x
Puntos de venta / Marketing											
<b>Comunidad de desarrolladores</b>	x		x	x	x				x	x	x
Jerarquías humanas									x		
Colaboración con otros productos											
Modularidad						x					
Estándares											
Comunidad de usuarios						x					
Penetración de mercado											
<b>Soporte</b>	x		x		x	x			x		x
Probado			x								
Arquitectura	x				x						x
Despliegue											
Lenguaje											
Líder de proyecto	x				x						x
Votos del usuario											
Vitalidad	x										
Popularidad	x				x	x					x
Actividad de errores	x				x		x				
<b>Documentación</b>	x		x	x	x	x	x	x	x		x
<b>Funcionalidad</b>	x	x		x	x	x	x		x	x	x
Calidad del producto							x				
Formación									x		
Integración							x		x		

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

Usabilidad	x	x			x	x				x	x
Seguridad	x		x		x				x	x	x
Rendimiento	x	x			x				x	x	x
Escalabilidad	x				x						x
Confiabilidad	x	x	x	x	x	x				x	
Mantenimiento	x		x	x	x	x	x				
Costos	x				x			x			
Flexibilidad		x									
Interoperabilidad						x				x	
Satisfacción del cliente	x					x					
Compartición										x	

**Anexo 6: Guión de la entrevista aplicada a los especialistas en migración del centro CESOL**

Estimado(a) Especialista: La presente entrevista tiene como objetivo conocer cómo se lleva a cabo actualmente el proceso de selección de alternativas libres al software privativo durante un proceso de migración a código abierto para determinar las principales deficiencias presentes.

**Pregunta 1:** ¿En cuántos procesos de migración se ha visto involucrado de alguna manera u otra? Menciones cuáles.

**Pregunta 2:** ¿Qué roles ha desempeñado en los mismos?

**Pregunta 3:** ¿Cómo se ha realizado la selección de alternativas libres al software privativo en los procesos de migración en los que ha participado?

**Pregunta 4:** ¿Qué tiene en cuenta al proponer las alternativas libres al software privativo?

**Pregunta 5:** ¿Qué elementos cree necesario incorporar en la selección de alternativas libres al software privativo durante un proceso de migración?

**Anexo 7: Principales elementos de las respuestas a la entrevista aplicada a los especialistas en migración del centro CESOL**

Acerca de cómo se realiza la selección de alternativas	Acerca qué se tiene en cuenta al proponer las alternativas	Elementos que se deben incorporar en la selección de
--	--	--

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

	<b>libres</b>	<b>alternativas</b>
<p>La selección de alternativas libres en los procesos de migración se ha realizado por 2 vías principales, existen distintos software previamente identificados por la comunidad de software libre y páginas especializadas para la homologación de software, otros no identificados o que no son completamente homólogos en funcionalidad con el software privativo en cuestión, se buscan alternativas de software libres que en conjunto llegan a completar las funcionalidades de este. El cumplimiento de las funcionalidades y la satisfacción del usuario es de mucha importancia durante un proceso de migración a código abierto.</p> <p>Para la selección de las alternativas, primeramente se realiza un levantamiento de los sistemas homólogos que se utilizan y las funciones que estos realizan. Dependiendo</p>	<p>Para proponer alternativas libres al software privativo, se tiene en cuenta el uso de funcionalidades parcial o total y rendimiento de la misma, así como la compatibilidad de migración de datos almacenados en caso de las aplicaciones que almacenan estos datos en bases de datos.</p> <p>Se tiene en cuenta la licencia, que satisfaga la mayor cantidad de procesos para definir la menor cantidad de aplicaciones posibles.</p> <p>Facilidad de uso para el usuario. Similitud con las herramientas privativas.</p> <p>A la hora de proponer una alternativa de aplicación de software libre al software privativo, se debe tener en cuenta que, primero cumpla con todas o al menos realice las principales operaciones que realizaba la privativa. Otro elemento importante es la usabilidad y calidad de la alternativa de cara al usuario, ya que en la mayoría de</p>	<p>En muchos de los casos de selección de las alternativas libres al software privativo durante un proceso de migración se hace muy complicado, llegar a una conclusión definitiva en cuanto a qué alternativa utilizar, pues como el software libre en un gran % es desarrollado en comunidades, es difícil cumplir las expectativas de uso y funcionalidades de algunos de ellos. Por lo que se debería tener en cuenta el uso de aplicaciones virtualizadas para la migración.</p> <p>Sencillez de uso. Interfaz intuitiva.</p> <p>EL principal elemento que creo que, no incorporar, sino mejorar, es el estudio de las alternativas. Este proceso debe ser realizado con el mayor rigor posible, porque las mismas son las herramientas a utilizar en las empresas que se migren y generalmente una buena selección de herramientas, entre otros factores, ayuda a que el resultado final sea satisfactorio.</p> <p>Que el software cumpla con las</p>

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

<p>de las funciones se realiza una búsqueda entre las herramientas que ya están definidas en el centro, sobre cuáles cumplen con la mayor cantidad de funcionalidad en correspondencia con las herramientas privativas que se van a migrar. Luego se buscan alternativas para satisfacer las necesidades que no han sido resueltas con las herramientas que se han seleccionados. Lo que ha llegado a que a un sistema privativo se le ha migrado en ocasiones con más de una alternativa, dependiendo de los procesos.</p> <p>Consultando el criterio de los expertos del Centro.</p> <p>Antes de cada migración se realizaba un estudio relacionado con las características que poseía la empresa en cuanto a las condiciones técnicas que tenían las mismas, dígame hardware y las aplicaciones que normalmente utilizaban los</p>	<p>las ocasiones este es un poco reacio al cambio. Además se tiene en cuenta el soporte que pueda tener la herramienta propuesta como alternativa, no sirve proponer una aplicación que no tenga soporte o que ya este discontinuada, por consiguiente trae problemas de seguridad y estabilidad.</p> <p>Nivel de usabilidad, capacidad para abarcar todas las funciones utilizadas en las herramientas privativas, compatibilidad con los formatos de los archivos con los que se trabaja.</p> <p>Que pueda cumplir con al menos el 80% de las funciones del software privativo. Han existido ocasiones en las que 2 alternativas libres hacen la función de 1 software privativo y eso resulta ser bastante tedioso para el usuario.</p> <p>Funcionalidades, Actividad de errores, Documentación. Disponibilidad de aplicaciones de Nova.</p>	<p>expectativas del usuario, que no sea muy difícil de usar y de ser posible que tenga una interfaz amigable. Que sea un proyecto activo. Que el repositorio pueda usar las últimas versiones.</p> <p>Presentar al usuario las propuestas para tener en cuenta su opinión en la selección.</p> <p>Definir un conjunto de indicadores medibles, que se puedan tener en cuenta para la selección de las alternativas libres ante una aplicación privativa. Por ejemplo, satisfacción del usuario, recubrimiento funcional, propietario, disponibilidad del código fuente.</p> <p>Definir un sistema de almacenamiento en la nube que contenga el catálogo de aplicaciones privativa con sus posibles alternativas. Este sistema debe estar disponible, tiene que tener la capacidad de ser consultado por cualquier aplicación o persona. Tiene que ser lo suficientemente abarcador que contenga todas las aplicaciones posibles.</p>
---	---	--

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

<p>trabajadores. Se analizaban si existían alternativas a las herramientas utilizadas y si estas reunían las condiciones como para suplirlas.</p> <p>Detectar qué aplicaciones privativas utilizaban los usuarios y realizar un estudio de homólogos en aplicaciones de software libre y seleccionar una similar.</p> <p>Hasta el momento no existe una solución cubana para la selección de alternativas libres en dependencia de las condiciones y estado de las tecnologías en las empresas, además de las actividades y tareas definidas en la institución, yo como especialista para seleccionar la alternativa primeramente visitado los directorios de software en línea en búsquedas de las aplicaciones semejantes o alternativas, pero ello no me resuelve totalmente el problema luego he tenido que instalarlas y realizar</p>	<p>Primeramente que tenga todas las funcionalidades que brinda la privativa (en ocasiones me fue necesario utilizar 2 o más para lograr reunir todas las características que contenía la privativa).</p> <p>Si existe varias que reúnen las características se utiliza la que tenga mejor soporte o la que este más actualizada, mejor proceso de actualización (saque nuevas versiones en menos tiempo, esto ayuda a tener un software con mejor acabado, menos bugs).</p> <p>Que consuma menos o se ajuste a las condiciones del hardware.</p> <p>Si existen varias que cumplen con todos los requisitos, selecciono la que sea más vistosa y la más amigable para un usuario que esta acostumbrado al buen comportamiento en este sentido de las aplicaciones de windows.</p> <p>Primeramente se tiene en cuenta</p>	<p>Una aplicación informática que teniendo en cuenta un conjunto de indicadores seleccione la alternativa mas idónea ante una aplicación privativa (automatización del proceso de selección).</p> <p>Es necesario tener en cuenta el tipo de licencia y soporte.</p> <p>Sería importante idear mecanismos, técnicas que faciliten y estandaricen este proceso. Además de que se centralice el conocimiento de los expertos en migración, así como las experiencias adquiridas en los casos de éxitos realizados, a modo de lograr que este proceso sea lo más práctico y sencillo posible.</p>
--	---	--

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

<p>pruebas en búsqueda de la más idónea para usar en la empresa.</p> <p>Se realiza un levantamiento de las tecnologías y aplicaciones propietarias que están desplegadas en la empresas teniendo en cuenta la distribución por áreas y nivel de uso de las personas que interactúan con estas. A continuación se consultan los recursos de hardware de los ordenadores de la empresa, la cantidad de personas que de forma concurrente acceden a estas aplicaciones. Secundariamente se tiene en cuenta la curva de aprendizaje que puedan tener los usuarios para aprender las nuevas tecnologías. Una vez que se tiene toda esta información se pasa analizar qué aplicaciones y tecnologías libres se pueden utilizar como alternativas libres a las privadas que están en las empresas. Este proceso se realiza de forma manual consultando la guía cubana de</p>	<p>que la alternativa libre tenga todas las funcionalidades que tenía la aplicación privativa, sino es así es necesario reunir varias soluciones que agrupen la mayor cantidad de funcionalidades.</p> <p>Se tiene en cuenta también que el consumo de recursos de la alternativa sea mínimo y compatible con los recursos de hardware donde se va a desplegar.</p> <p>Que el aprendizaje de los usuarios sobre la nueva tecnología sea en corto tiempo, que exista buena documentación sobre la implantación y uso de la aplicación.</p> <p>Lo principal a tener en cuenta es que la alternativa satisfaga las necesidades de los usuarios.</p> <p>En primer lugar, que la o las alternativas cubran completamente las funcionalidades que utiliza el usuario. Adicionalmente que sean lo más similar posible a su</p>	
---	---	--

<p>migración a código abierto la cual cuenta con un catálogo de aplicaciones libres que están autorizadas para el despliegue en las instituciones cubanas. Este catálogo de aplicaciones tiene por cada aplicación registrada las posibles alternativas libres que pueden sustituir esta en el proceso de migración.</p> <p>La alternativa de los software a utilizar son seleccionadas a partir de lo establecido en la Guía de migración, depende también de la opinión de los usuarios en cuanto su utilización y depende además de las nuevas tecnologías.</p> <p>En todos los casos se realizó un levantamiento de las herramientas y servicios utilizados, así como las funcionalidades más usadas. En base a ello se determinaba (según el criterio de expertos o migraciones anteriores) la herramienta o conjunto de ellas que pudieran cubrir</p>	<p>homóloga privativa (en cuanto a su apariencia, usabilidad y rendimiento).</p>	
---	--	--

<p>totalmente las necesidades del usuario.</p> <p>El primer paso es identificar qué soluciones libres (aplicación o conjunto de aplicaciones) permiten a los usuarios realizar su trabajo. Se busca en estudios hechos dentro del propio centro o en Internet posibles candidatas. Se comprueba que cumplen con las expectativas. Las opciones encontradas en el paso anterior se filtran teniendo en cuenta: Las prestaciones del hardware de los clientes, Usabilidad, Soporte con que cuentan, Capacidad de interoperar con otras aplicaciones, Compatibilidad con los formatos de datos de las aplicaciones privativas de manera que se pueda migrar la información. Poner a prueba con los clientes la aplicación seleccionada para validar que cumple con las necesidades.</p>		
--	--	--

## **Anexo 8: Licencias más populares en aplicaciones de código abierto**

<b>Licencias</b>	<b>Observación</b>
Apache License 2.0	(Permisiva) <a href="https://opensource.org/licenses/Apache-2.0">https://opensource.org/licenses/Apache-2.0</a>
BSD 3-Clause "New" or "Revised" license	(Permisiva) <a href="https://opensource.org/licenses/BSD-3-Clause">https://opensource.org/licenses/BSD-3-Clause</a>
BSD 2-Clause "Simplified" or "FreeBSD" license	(Permisiva) <a href="https://opensource.org/licenses/BSD-2-Clause">https://opensource.org/licenses/BSD-2-Clause</a>
GNU General Public License (GPL)	(CopyLeft) <a href="https://opensource.org/licenses/gpl-license">https://opensource.org/licenses/gpl-license</a>
GNU Library or "Lesser" General Public License (LGPL)	(CopyLeft) <a href="https://opensource.org/licenses/lgpl-license">https://opensource.org/licenses/lgpl-license</a>
MIT license	(Permisiva) <a href="https://opensource.org/licenses/MIT">https://opensource.org/licenses/MIT</a>
Mozilla Public License 2.0	(CopyLeft) <a href="https://opensource.org/licenses/MPL-2.0">https://opensource.org/licenses/MPL-2.0</a>
Common Development and Distribution License	(CopyLeft) <a href="https://opensource.org/licenses/CDDL-1.0">https://opensource.org/licenses/CDDL-1.0</a>
Eclipse Public License	(CopyLeft) <a href="https://opensource.org/licenses/EPL-1.0">https://opensource.org/licenses/EPL-1.0</a>

## **Anexo 9: Encuesta para determinar nivel de competencia de los expertos**

Estimado(a) compañero(a).

La presente encuesta forma parte de las acciones para validar la estrategia de selección de alternativas libres propuesta para un proceso de migración a código abierto. Su análisis y colaboración en cuanto a los aspectos que sometemos a su consideración serán de invaluable ayuda para el desarrollo de la investigación. Le solicitamos la mayor responsabilidad y sinceridad en la realización de la encuesta.

Se necesita primeramente que evalúe su conocimiento acerca de la migración a código abierto en

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

estaciones de trabajo, modelos FLOSS y el grado de influencia de las fuentes de argumentación, según las indicaciones que se dan a continuación. Le agradecemos de antemano por su valiosa contribución.

**Datos generales del encuestado**

Título universitario: \_\_\_\_\_

Categoría científica: \_\_\_\_\_ Categoría docente: \_\_\_\_\_

Años de experiencia en el desarrollo de procesos de migración: \_\_\_\_\_

**Instrucciones**

- Según su criterio, marque con una X en la casilla que caracteriza su nivel de conocimiento sobre los temas referidos. “0” significa total desconocimiento del tema y “10” que tiene pleno conocimiento del mismo.

0	1	2	3	4	5	6	7	8	9	10

- Entre las fuentes que le han posibilitado enriquecer su conocimiento sobre los temas, se someten a consideración algunas de ellas, para que las evalúe en las categorías de: Alto (A), Medio (M) y Bajo (B), colocando una X.

FUENTES DE ARGUMENTACIÓN	Grado de influencia de cada una de las fuentes		
	Alto (A)	Medio (M)	Bajo (B)
Estudios teóricos realizados por usted.			
Experiencia adquirida durante su vida profesional.			
Conocimiento de investigaciones y/o publicaciones nacionales e internacionales.			
Conocimiento propio sobre el estado del tema de investigación.			
Actualización en cursos de posgrado, diplomados, maestrías, doctorado.			
Intuición.			

**Anexo 10: Tablas del método Delphi**

**Tabla 22. Cálculo del coeficiente de conocimiento para los expertos.** *Elaboración propia*

Número de experto	Escala										Kc
	1	2	3	4	5	6	7	8	9	10	
1								x			0.8
2								x			0.8
3								x			0.8
4									x		0.9
5									x		0.9
6								x			0.8
7								x			0.8
8								x			0.8
9										x	1
10								x			0.8

**Tabla 23. Matriz de coeficientes de argumentación por experto.** *Elaboración propia*

Número de experto	Fuente de argumentación						Ka
	1	2	3	4	5	6	
1	0,30	0,50	0,05	0,05	0,05	0,05	1
2	0,20	0,50	0,05	0,04	0,04	0,05	0,88
3	0,20	0,50	0,03	0,05	0,04	0,05	0,87
4	0,30	0,50	0,05	0,04	0,05	0,05	0,99
5	0,30	0,50	0,05	0,04	0,05	0,05	0,99
6	0,30	0,50	0,05	0,04	0,05	0,05	0,99
7	0,20	0,50	0,04	0,05	0,05	0,05	0,89
8	0,20	0,50	0,04	0,05	0,05	0,05	0,89
9	0,30	0,50	0,05	0,05	0,05	0,05	1
10	0,20	0,50	0,04	0,05	0,04	0,05	0,88

**Tabla 24. Niveles de competencia de los expertos.** *Elaboración propia*

Número de experto	Kc	Ka	K	Nivel de competencia
1	0,8	1	0,9	Alto
2	0,8	0,88	0,84	Alto
3	0,8	0,87	0,84	Alto
4	0,9	0,99	0,95	Alto
5	0,9	0,99	0,95	Alto
6	0,8	0,99	0,9	Alto
7	0,8	0,89	0,85	Alto
8	0,8	0,89	0,85	Alto
9	1	1	1	Alto
10	0,8	0,88	0,84	Alto

**Tabla 25. Resultados de la evaluación realizada por los expertos a los aspectos propuestos.** *Elaboración propia*

Número de experto	Fuente de argumentación									
	1	2	3	4	5	6	7	8	9	10
1	MA	MA	BA	MA	MA	A	MA	MA	MA	MA
2	BA	MA	MA	A	MA	MA	BA	MA	MA	MA
3	MA	A	MA	MA	MA	MA	MA	MA	BA	MA
4	MA	MA	A	MA	BA	MA	MA	MA	MA	BA
5	MA	A	A	MA	MA	BA	MA	MA	MA	MA
6	MA	MA	MA	BA	MA	MA	MA	MA	MA	MA
7	BA	BA	MA	BA	MA	MA	MA	MA	MA	MA
8	MA	MA	MA	MA	MA	BA	MA	MA	MA	MA
9	MA	MA	BA	MA	A	MA	MA	MA	MA	MA
10	BA	MA	BA							

**Tabla 26. Frecuencia absoluta por aspectos.** *Elaboración propia*

Número de aspecto	MA	BA	A	PA	I	Total
1	8	1	1	0	0	10
2	7	2	1	0	0	10
3	8	1	1	0	0	10
4	7	2	1	0	0	10
5	7	1	2	0	0	10
6	9	1	0	0	0	10
7	7	3	0	0	0	10
8	9	1	0	0	0	10
9	8	1	1	0	0	10
10	8	2	0	0	0	10

**Tabla 27. Frecuencia acumulada por aspecto.** *Elaboración propia*

Número de aspecto	MA	BA	A	PA	I
1	8	9	10	10	10
2	7	9	10	10	10
3	8	9	10	10	10
4	7	9	10	10	10
5	7	8	10	10	10
6	9	10	10	10	10
7	7	10	10	10	10
8	9	10	10	10	10
9	8	9	10	10	10
10	8	10	10	10	10

**Tabla 28. Frecuencias relativas de las evaluaciones por aspecto.** *Elaboración propia*

Número de aspecto	MA	BA	A	PA	I
1	0,8000	0,9000	1,0000	1,0000	1,0000
2	0,7000	0,9000	1,0000	1,0000	1,0000

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

3	0,8000	0,9000	1,0000	1,0000	1,0000
4	0,7000	0,9000	1,0000	1,0000	1,0000
5	0,7000	0,8000	1,0000	1,0000	1,0000
6	0,9000	1,0000	1,0000	1,0000	1,0000
7	0,7000	1,0000	1,0000	1,0000	1,0000
8	0,9000	1,0000	1,0000	1,0000	1,0000
9	0,8000	0,9000	1,0000	1,0000	1,0000
10	0,8000	1,0000	1,0000	1,0000	1,0000

**Anexo 11: Encuesta criterios de los expertos**

Estimado(a) compañero(a).

Con la finalidad de someter a su consideración como experto(a) la estrategia de selección de alternativas libres, solicitamos su valoración sobre diferentes aspectos que a continuación se presentan. De antemano, le agradecemos su valiosa contribución.

Para expresar su evaluación, por favor, luego de analizar cuidadosamente el material que se adjunta, evalúe cada uno de los aspectos que se le presentan en la tabla, marcando con una cruz en la casilla correspondiente y teniendo en cuenta para ello el siguiente código de categorías de clasificación.

- **MA:** Muy adecuado.
- **BA:** Bastante adecuado.
- **A:** Adecuado.
- **PA:** Poco adecuado.
- **I:** Inadecuado.

No	Aspectos	MA	BA	A	PA	I
1	La estrategia propuesta, como vía para una correcta planificación de selección de alternativas libres en un proceso de migración, la valoro de forma ...					
2	Las etapas definidas en la estrategia de selección de alternativas libres para el proceso de migración las valoro de forma ...					
3	Los atributos genéricos definidos para la selección de					

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

	alternativas libres, en cuanto a la forma de evaluar y clasificar sus métricas, los valoro de forma ...					
4	Los atributos específicos definidos para la selección de alternativas libres, en cuanto a la forma de evaluar y clasificar sus métricas, los valoro de forma ...					
5	La correspondencia entre los atributos genéricos de la estrategia de selección de alternativas libres y los modelos FLOSS la valora de forma ...					
6	La correspondencia entre los actividades de la estrategia de selección de alternativas libres y el área de proceso DAR de CMMI nivel 3 la valora de forma ...					
7	Los artefactos definidos de la estrategia de selección de alternativas libres para el proceso de migración a código abierto los valoro de forma ...					
8	La correspondencia entre la estrategia de selección de alternativas libres y las etapas de un proceso de migración (preparación, ejecución y consolidación) la valora de forma ...					
9	La posibilidad de aplicación de la estrategia de selección de alternativas libres durante un proceso de migración la valora de forma ...					
10	La influencia de la estrategia de selección de alternativas libres en la correcta realización de un proceso de migración la valora de forma ...					

Si desea exponer cualquier otra opinión, por favor, exprese en el espacio disponible a continuación:

---

**Anexo 12: Cuestionario aplicado para determinar nivel de satisfacción**

Estimado(a) compañero(a):

Al contestar este cuestionario podrá dar a conocer su nivel de satisfacción con respecto a las aplicaciones

*Estrategia de selección de alternativas libres para el proceso de migración a código abierto*

de escritorio fundamentales de la distribución cubana Nova Escritorio 6.0, brindada por los especialistas del centro CESOL durante el proceso de migración a código abierto desarrollado recientemente en el ministerio. Se espera su sinceridad y se le agradece su colaboración en esta investigación.

**Instrucciones:** Marque con una X la opción deseada 1: Insatisfecho, 2: Poco satisfecho, 3: Satisfecho, 4: Bastante satisfecho y 5: Muy Satisfecho.

<b>Preguntas</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
¿Le satisface utilizar el navegador web Mozilla Firefox?					
¿Le satisface utilizar el explorador de archivos Nautilus?					
¿Le satisface utilizar el cliente de correo electrónico Thunderbird?					
¿Le satisface utilizar el reproductor de multimedia VLC?					
¿Le satisface utilizar el reproductor de música Rhythmbox?					
¿Le satisface utilizar el visor de PDF Evince?					
¿Le satisface utilizar el paquete ofimático LibreOffice?					
¿Le satisface utilizar la interfaz de usuario del procesador de textos LibreOffice Writer?					
¿Le satisface utilizar el procesador de textos LibreOffice Writer para la inserción o vinculación de gráficos, hojas de cálculo y otros objetos?					
¿Le satisface utilizar las plantillas y estilos del procesador de textos LibreOffice Writer?					
¿Le satisface utilizar el paquete ofimático WPS Writer?					
¿Le satisface utilizar la interfaz de usuario del procesador de textos WPS Writer?					
¿Le satisface utilizar el procesador de textos WPS Writer para la inserción o vinculación de gráficos, hojas de cálculo y otros objetos?					
¿Le satisface utilizar las plantillas y estilos del procesador de textos WPS Writer?					

**MUCHAS GRACIAS POR SU COLABORACIÓN**

### Anexo 13: Tarjetas de identificación de LibreOffice Writer, Abiword y WPS Office Free

*Tabla 29. Tarjetas de identificación de LibreOffice Writer. Elaboración propia*

Elemento	Descripción
Nombre:	LibreOffice Writer
Descripción:	Procesador de texto de código abierto del paquete de software LibreOffice.
Versión:	5.1.0.3
Url:	<a href="https://es.libreoffice.org/">https://es.libreoffice.org/</a>
Tipo:	Oficina
Año:	2016
Licencia:	LGPL
Idioma:	Español
Lenguajes:	C++, Java, C, CSS y Python
Disponibilidad del código fuente:	Si
Nivel de criticidad:	Alto

*Tabla 30. Tarjetas de identificación de Abiword. Elaboración propia*

Elemento	Descripción
Nombre:	Abiword
Descripción:	Procesador de texto
Versión:	3.0.2
Url:	<a href="https://www.abisource.com/">https://www.abisource.com/</a>
Tipo:	Oficina
Año:	2016
Licencia:	GPL
Idioma:	Español
Lenguajes:	C++
Disponibilidad del código fuente:	Si
Nivel de criticidad:	Alto

**Tabla 31. Tarjetas de identificación de WPS Office Free.** *Elaboración propia*

<b>Elemento</b>	<b>Descripción</b>
<b>Nombre:</b>	WPS Office Writer
<b>Descripción:</b>	Procesador de texto del paquete de software WPS Office para Linux.
<b>Versión:</b>	10.1.0
<b>Url:</b>	<a href="https://www.wps.com/linux">https://www.wps.com/linux</a>
<b>Tipo:</b>	Oficina
<b>Año:</b>	2016
<b>Licencia:</b>	Desconocida
<b>Idioma:</b>	Español
<b>Lenguajes:</b>	Delphi, C y C++
<b>Disponibilidad del código fuente:</b>	No
<b>Nivel de criticidad:</b>	Alto

**Anexo 14: Acta de aceptación de la Migración en el MINCOM**



**Acta de aceptación**

**ACTA DE ACEPTACIÓN**

En cumplimiento del **Convenio de colaboración** Ministerio de Comunicaciones y en función de la ejecución del proyecto: Asesoría y Soporte para la migración a código abierto de la red de navegaciones para el MINCOM, se hace entrega del producto que se relaciona a continuación: Red de navegación del ministerio.

La red de navegación cuenta con 256 computadoras, de estas 20 son clientes ligeros. Todas las computadoras fueron migradas excepto 8: 5 son del nodo, 2 están pendientes a solicitud de mejora y 1 por el usuario.

La Parte Cliente, luego de haber revisado el producto de trabajo determina que acepta al mismo.

Entrega 	Recibe 
Nombre y apellidos: Yoandy Pérez Villazón  Cargo: Director del Centro de Software Libre Centro CESOL Facultad 1	Nombre y apellidos: Alexander Rubiera Castillo  Cargo: Jefe del Centro de Comunicaciones

Representante Parte Suministradora Centro de Software Libre

Nombre y Apellidos: María Leisy González Carrera

Cargo: Jefa de proyecto

Firma: 



Fecha: 07/02/2017