

Algoritmo para la Evaluación de la Calidad Sistémica del Software

Luis E. Mendoza

María Pérez

Anna C. Grimán

Teresita Rojas

Laboratorio de Investigación en Sistemas de Información (LISI)
Departamento de Procesos y Sistemas, Universidad Simón Bolívar
Caracas - Venezuela.
+58 (212) 906.40.17

lmendoza@usb.ve

movalles@usb.ve

agriman@usb.ve

trojas@usb.ve

RESUMEN

La necesidad de garantizar Sistemas de Software de calidad en una competencia abierta y global ha motivado que a nivel mundial se propongan un conjunto de modelos para evaluar su calidad. Los Sistemas de Software pueden ser considerados como productos o servicios que responden a las características propias de cada organización y/o negocio y a las necesidades de sus consumidores. Un modelo de calidad representa estos requerimientos y necesidades. Junto a su formulación se deben también desarrollar los pasos a seguir para su aplicación. En este artículo se presenta un algoritmo para la aplicación del **MO**dulo **S**istémico de **C**alidad (MOSCA) para la estimación de la calidad de los Sistemas de Software que garantiza la relación sinérgica entre las características de la empresa y las necesidades del usuario, de tal manera de guiar el proceso de evaluación y garantizar la calidad sistémica. Adicionalmente, a través de un caso de estudio, se describe la aplicación del algoritmo a dos empresas venezolanas desarrolladoras de Sistemas de Software, comprobándose así que el algoritmo presentado, en conjunto con MOSCA, constituye una herramienta efectiva de análisis y estimación de la Calidad Global Sistémica, pues se analizan aspectos del producto, del proceso y su relación con el medio ambiente.

Palabras claves

Algoritmo de evaluación, Evaluación de software, Calidad de software, Calidad sistémica, Proceso de evaluación, Caso de estudio.

1. INTRODUCCIÓN

La calidad es un término que ha adquirido gran relevancia con el paso del tiempo, ya que es considerada como uno de los principales activos con los que cuenta un país para mejorar su posición competitiva global [6]. Para conseguir calidad del software es esencial establecer un programa de medidas a tomar con respecto a los proveedores o desarrolladores. Es también importante utilizar los modelos y métodos apropiados para controlar el proceso de desarrollo.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar

los objetivos del proceso de desarrollo, debido a que la calidad del primero va a depender, entre otros aspectos, de éstos. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto. Según Pressman [12] la calidad del software es *“la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”*.

La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software. Sin embargo, la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después [4].

El presente artículo tiene por objetivo mostrar la formulación y aplicación de un algoritmo que permite la operacionalización del **MO**dulo **S**istémico de **C**alidad (MOSCA) [8] para la Estimación de la Calidad de los Sistemas de Software (SS), desarrollado en el Laboratorio de Investigación en Sistemas de Información (LISI) de la Universidad Simón Bolívar de Venezuela, de tal manera de guiar el proceso de evaluación de la calidad sistémica del software, soportada por los conceptos de la Calidad Total Sistémica [2,13].

En las próximas secciones se describirá brevemente MOSCA, así como sus bases conceptuales. Posteriormente, se propone el algoritmo para la aplicación de MOSCA, objeto de esta investigación. Seguidamente, a través de un caso de estudio, se mostrará su aplicación a dos empresas desarrolladoras de productos de software en Venezuela. Por último, se enumerarán las conclusiones.

2. CALIDAD GLOBAL SISTÉMICA Y LOS MODELOS DE CALIDAD

Callaos y Callaos [2,13] proponen un concepto de calidad del software en el cual están involucrados tanto características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de Calidad del Software. Este enfoque es considerado también por Dromey [3] y, particularmente reforzado por Voas [14], cuando se refiere al Triángulo de la Certificación de la Calidad del Software.

La definición de calidad sistémica en el desarrollo de los Sistemas

de Información consta de cuatro (4) tipos de Calidades: Eficiencia del Producto, Efectividad del Producto, Eficiencia del Proceso y Efectividad del Proceso, considerando las dimensiones del Cliente y del Usuario. Esta división se justifica en un sentido, porque un proyecto incluye tanto la eficiencia como la efectividad y en el otro, porque el Sistema concebido (el producto) es diferente al Sistema de las actividades humanas (el proceso) mediante el cual el Sistema-Producto es diseñado [2]. Según Callaos, la calidad global no es la suma de las calidades parciales, sino el compromiso entre todo el conjunto de calidades que conlleve a un óptimo global con cierto sacrificio de los óptimos parciales [2].

Para establecer con mayor detalle los atributos que garantizan la calidad del proceso y del producto basados en el enfoque de Calidad Sistemática, el Laboratorio de Investigación en Sistemas de Información (LISI) de la Universidad Simón Bolívar propuso dos modelos de calidad con enfoque sistémico: uno enfocado hacia el producto [9,10] y otro hacia el proceso [11]. El primer modelo enlaza las ideas del Estándar ISO/IEC 9126 [7] para el producto y del Modelo de Dromey [3]. El resultado es un conjunto de atributos que deben estar presentes en un sistema de calidad y los factores que influyen a éstos, los cuales se agrupan en cuatro (4) grupos representados por las cuatro (4) dimensiones del enfoque de Callaos [9,10]. El segundo modelo desarrollado en LISI, integra el enfoque de calidad sistémica con las características presentes en el modelo de procesos SPICE [5], adaptado a la realidad venezolana, conformado por una jerarquía de 5 niveles [11]: Ciclos de Vida, Categorías, Procesos, Principios, y Bases Prácticas, que son un conjunto de directrices a ser ejecutadas por la organización para lograr alcanzar un principio. Este modelo garantiza el balance entre la eficiencia y la efectividad del proceso de desarrollo a través de una propuesta

equilibrada de prácticas bases [11].

Ahora bien, aunque los modelos indicados anteriormente están inspirados en la Calidad Global Sistemática [2,13], ninguno soporta el concepto de calidad del software por sí solo, ya que se tiene que la calidad del software no es algo que depende de una sola característica en particular, sino que obedece al compromiso de todas sus partes [2]. Es por ello que este enfoque es el que sirve como eje en la construcción de MOSCA [8], tomando lo mejor de cada uno de los modelos descritos anteriormente, y que se expone a continuación.

3. MODELO SISTÉMICO DE CALIDAD (MOSCA)

MOSCA [8] es un modelo que integra los modelos de calidad nombrados anteriormente, considerándolos como sub-modelos de éste. Fundamentalmente, tal como indica Humphrey [6], la calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estas dos calidades; tener modelos separados capaces de medir individualmente la calidad de un producto o de un proceso de software no garantiza la relación sistémica que debe estar presente entre ellos. Esto se debe a que la naturaleza de los sistemas no puede ser dividida en partes, sino que debe existir una interdependencia y colaboración entre las partes (proceso y producto) para que el mismo sea visto como un todo [8].

La estructura de MOSCA (ver Figura 1) consta de 4 niveles los cuales son explicados brevemente a continuación [8].

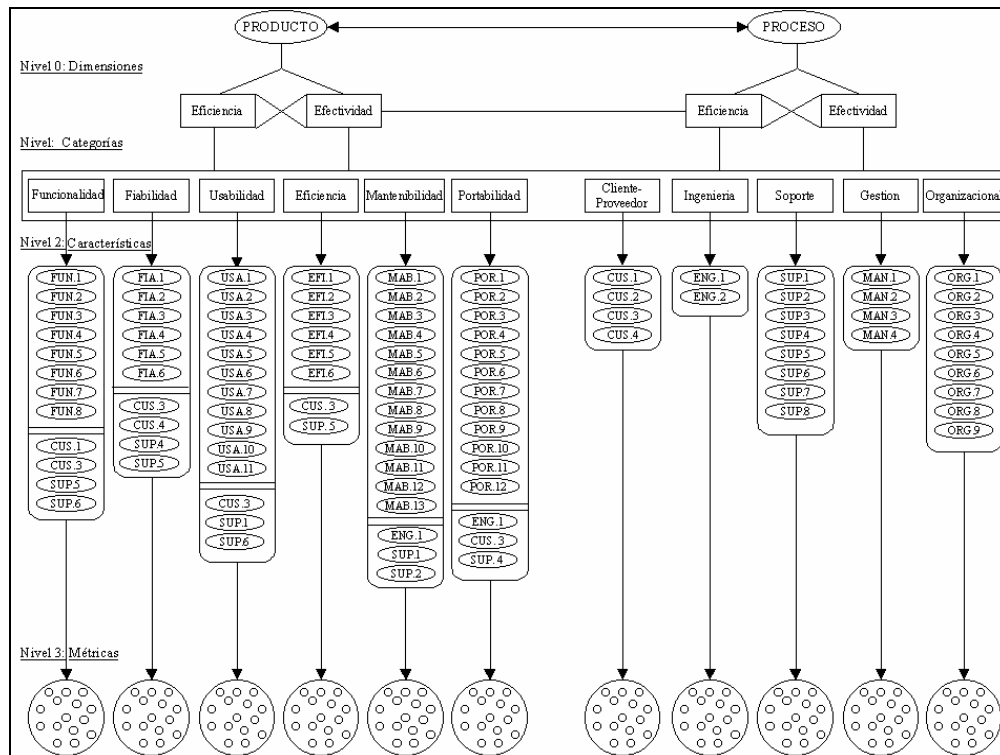


Figura 1. Estructura de MOSCA [Mendoza et al., 2001].

a) **Nivel 0: Dimensiones.** Eficiencia del proceso, Efectividad del proceso, Eficiencia del producto y Efectividad del producto son las cuatro dimensiones propuestas en el prototipo de modelo. Sólo un balance y una buena interrelación entre ellas garantizan la calidad Sistemática global de una organización.

b) **Nivel 1: Categorías.** Se contemplan 11 categorías:
Producto: Funcionalidad (FUN), Fiabilidad (FIA), Usabilidad (USA), Eficiencia (EFI), Mantenibilidad (MAB) y Portabilidad (POR)
Proceso: Cliente-Proveedor (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN) y Organizacional (ORG).

Esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a que sector o sub-modelo pertenecen.

c) **Nivel 2: Características.** Cada categoría tiene asociado un conjunto de características (56 asociadas al producto y 27 al proceso de desarrollo), las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso. Entre las características asociadas a cada categoría del producto, se proponen en el modelo MOSCA, una serie de características del proceso (ver Figura 1). Esto se debe, a que algunas características de la calidad del proceso, impactan directamente en las categorías del producto al igual que ciertas características de la calidad del producto definen categorías del proceso. *Esto ayuda a precisar que si una vez medidas las características asociadas a una categoría en particular del producto, arroja resultados no deseados, se pueden analizar las características de la calidad del proceso asociadas a esa categoría del producto para encontrar las posibles causas [8].*

d) **Nivel 3: Métricas.** Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistemática. Dada la cantidad de métricas asociadas a cada una de las características que conforman MOSCA (587 en total) [8], éstas no serán presentadas en el presente artículo.

En resumen, MOSCA [8] consta de un total de 11 categorías (6 pertenecientes al producto y 5 al proceso de desarrollo), 83 características (56 asociadas al producto y 27 al proceso de desarrollo) y un total de 587 métricas. En este sentido, la aplicación de un modelo como MOSCA puede llegar a ser complejo sino se cuenta con una guía que permita conducir este proceso; es decir, aplicar MOSCA para evaluar la calidad sistemática de un producto de software y de la organización que lo desarrolló.

Sobre la base de las ideas anteriores, el trabajo que se presenta a través de este artículo dota a las organizaciones desarrolladoras de productos de software interesadas en evaluar su nivel de calidad sistemática, de una herramienta fácil y cómoda para lograr este objetivo.

4. DESCRIPCIÓN DEL ALGORITMO PARA APLICAR MOSCA

El algoritmo contempla tres (3) fases. Se deberá evaluar la calidad del producto de software como también la calidad del proceso de

desarrollo del mismo.

4.1. Fase 1: Calidad del producto de software con un enfoque sistémico.

A través de la ejecución de esta fase, se evalúa la calidad del producto del software; para tal fin, se siguen 4 actividades, las cuales son descritas a continuación.

1) Estimar la calidad de la funcionalidad del producto. Siempre y en todos los casos se debe medir primero la categoría Funcionalidad del producto. Si cumple con el 75% de las características necesarias que se proponen para esta categoría (ver Tabla 3), entonces se procede a la segunda actividad. Si el producto no cumple con la categoría Funcionalidad, la evaluación finaliza; es decir, el sub-modelo del proceso no deberá ser evaluado. Esto se debe a que la categoría Funcionalidad es la más importante dentro de la estimación de la calidad, ya que identifica la capacidad del mismo para cumplir las funciones para las que fue fabricado. Además, como nota importante, se brindarán al cliente las causas del por qué la Funcionalidad no pudo ser satisfecha y el nivel de calidad resultó ser nulo.

2) Instanciación del sub-modelo del producto. En esta actividad el cliente o dueño del SS debe seleccionar dos (2) categorías de las cinco (5) restantes del sub-modelo del producto, aquellas que consideran que su producto de software debe cumplir y que desea que sean evaluadas. Esta actividad se realiza usando técnicas de entrevistas y de *Brainstorming* entre cliente y evaluador. Luego, se deberán evaluar cada una de las categorías seleccionadas por el cliente. Cabe destacar en este momento que el algoritmo recomienda trabajar con un máximo de tres (3) características del producto (incluyendo la funcionalidad), ya que, si se seleccionan más de tres (3) características del producto, algunas de ellas podrían entrar en conflicto. En este sentido, Bass et al. [1] indican que la satisfacción de los atributos de calidad *“nunca pueden ser satisfechos aisladamente. La satisfacción de cualquier atributo de calidad puede tener un efecto, algunas veces positivo y algunas veces negativo, en la satisfacción de otros atributos de calidad”*.

3) Estimación de calidad para cada categoría. Para las dos (2) categorías seleccionadas previamente, se debe:

- Aplicar las métricas propuestas en el sub-modelo del producto para las categorías seleccionadas.

- Normalizar los resultados de las métricas a una escala del 1 al 5. La normalización de los resultados es llevada a cabo de acuerdo a la Tabla 2. Para la normalización se seleccionó esta escala debido a que la mayor parte de las métricas se formularon sobre la base de escalas tipo Likert, la cual presenta cinco (5) opciones para cada métrica, que son traducidas numéricamente a una puntuación que va del 1 (menor grado de satisfacción) al 5 (mayor grado de satisfacción), frente a la métrica.

- Verificar que el 75% de las métricas se encuentran dentro de los valores óptimos (mayor o igual a 4) para cada una de

sus características. Si no se cumple el 75% de las métricas asociadas, entonces esta característica tendrá calidad nula. En caso de cumplir el 75% de las métricas asociadas, esta característica habrá sido satisfecha. Para el caso donde la pregunta correspondiente a la métrica es respondida por varias personas, entonces el valor de esa métrica será la mediana de la población de respuestas. Si esta población contiene únicamente 2 muestras, entonces se tomará la de menor valor. (Esto aplica para las preguntas que son respondidas por varios desarrolladores y/o para las preguntas que aparecen en las encuestas tanto del líder como del desarrollador).

Tabla 2. Normalización de las métricas [9,10].

Tipo de métrica	Valor	Valor normalizado
Likert	1	1
	2	2
	3	3
	4	4
	5	5
Tasa	$0 \leq n < 0.25$	1
	$0.25 \leq n < 0.50$	2
	$0.50 \leq n < 0.75$	3
	$0.75 \leq n < 1$	4
	$n = 1$	5
Flag	0	1
	1	5
Porcentaje (%)	$0\% \leq n < 25\%$	1
	$25\% \leq n < 50\%$	2
	$50\% \leq n < 75\%$	3
	$75\% \leq n < 90\%$	4
	$90\% \leq n \leq 100\%$	5

- Evaluar la categoría. Una categoría es satisfecha si el número de características es altamente satisfecho, es decir, la Tabla 3 presenta el número de características mínimas satisfechas que debe tener cada categoría del producto para que ésta pueda ser satisfecha. Cabe destacar que los números de características por categoría del producto indicados en la Tabla 3 fueron determinados tomando en cuenta que para que una categoría sea satisfecha, al menos el 75% de sus características son altamente satisfechas; de esta manera se garantiza coherencia y consistencia en relación a los niveles de aceptabilidad establecidos por el modelo. Luego se continúa con la actividad 4.

Tabla 3. Número mínimo de características que deben ser satisfechas por cada categoría.

Categorías del producto	Número de características mínimas que deben ser satisfechas
Funcionalidad	6
Fiabilidad	5
Usabilidad	8
Eficiencia	5
Mantenibilidad	11
Portabilidad	9

4) Estimar la calidad del producto partiendo de las categorías evaluadas. Para poder estimar la calidad del producto de

software se presenta la Tabla 4, en la cual se relacionan el nivel de calidad con las categorías satisfechas. En este punto es preciso recordar que si no se satisface la categoría Funcionalidad el algoritmo finaliza y la calidad del producto de software será nula. La Tabla 4 fue formulada partiendo del hecho de que si un producto de software cumple con el objetivo para el cual fue creado (Funcionalidad), éste tendrá una calidad Básica. Los niveles de calidad restantes atienden a la cantidad de categorías seleccionadas para la evaluación que el producto satisfaga; es decir, si satisface sólo una de las categorías seleccionadas, además de la Funcionalidad, tendrá un nivel de calidad Intermedio, mientras que si satisface todas las categorías seleccionadas, tendrá un nivel de calidad Avanzado.

Una vez terminada la evaluación del producto, y sólo en caso de que este obtenga al menos un nivel de calidad básica, se procederá a evaluar la calidad del proceso a través del sub-modelo del mismo.

Tabla 4. Nivel de calidad del producto con respecto a las categorías satisfechas para el producto.

Funcionalidad	Primera categoría evaluada	Segunda categoría evaluada	Nivel de calidad del producto de software
Satisfecha	No satisfecha	No satisfecha	Básico
Satisfecha	Satisfecha	No satisfecha	Intermedio
Satisfecha	No satisfecha	Satisfecha	Intermedio
Satisfecha	Satisfecha	Satisfecha	Avanzado

4.2. Fase 2: Calidad del proceso de desarrollo de software con un enfoque sistémico.

A través de la ejecución de esta fase, se evalúa la calidad del proceso de desarrollo del software; para tal fin, se siguen 4 actividades, las cuales son descritas a continuación.

- 1) Determinar el porcentaje de respuestas 'N/A' (No Aplica) contestadas por los encuestados, por cada categoría. Si este porcentaje es alto (superior a un 11%), entonces se debe analizar la aplicabilidad del instrumento de medición y el algoritmo se para. En caso contrario, se continúa con la segunda actividad de esta fase.
- 2) Determinar el porcentaje de respuestas 'N/S' (No Sabe) contestadas por los encuestados, por cada categoría. Para analizar el nivel de divulgación de información, si dicho porcentaje es alto (superior al 15%), esto indicará que existe un alto nivel de desinformación en relación a las actividades asociadas a la categoría en cuestión. Si el porcentaje es menor, se continúa con el algoritmo.
- 3) Determinar el grado de satisfacción de cada categoría. Para cada categoría del sub-modelo del proceso de desarrollo se debe determinar:

- El porcentaje de responder 'SI' a la(s) pregunta(s) asociadas a una métrica en particular.

- La frecuencia asociada a cada métrica, cuyo valor será la mediana de los valores calculados en el paso anterior.
- La frecuencia de cada característica.
- La frecuencia de todas las categorías del sub-modelo.

Una categoría será satisfecha si la frecuencia calculada es altamente satisfecha. La Tabla 5 muestra el número de características mínimas que cada categoría debe cumplir para ser altamente satisfecha. Al igual que para el sub-modelo del producto, los números de características por categoría del proceso indicados en la Tabla 5 fueron determinados tomando en cuenta que para que una categoría sea satisfecha, al menos el 75% de sus características son altamente satisfechas; de esta manera se garantiza nuevamente coherencia y consistencia en relación a los niveles de aceptabilidad establecidos por el modelo.

Tabla 5. Número mínimo de características que deben ser satisfechas por cada categoría para el proceso.

Categorías del proceso	Número de características mínimas que deben ser satisfechas
Cliente-Proveedor	3
Ingeniería	2
Soporte	6
Gestión	3
Organizacional	7

4) Estimar la calidad del proceso partiendo de las categorías evaluadas. Sobre la base de las categorías satisfechas, los niveles de calidad son:

- **Nivel de Calidad Básico.** Es la mínima calidad requerida. Se satisfacen las categorías Cliente-Proveedor e Ingeniería.
- **Nivel de Calidad Intermedio.** Ésta no sólo satisface las categorías del Nivel de calidad Básico, sino que, además, satisface las categorías Soporte y Gestión.
- **Nivel de Calidad Avanzado.** Satisface todas las categorías.

Cabe destacar en este momento que, a diferencia del sub-modelo de calidad del producto, el sub-modelo de calidad del proceso no se instancia debido primordialmente a que, tal como lo afirma Callaos y Callaos [2], “*el Sistema de Software construido (el producto) es diferente al Sistema de las actividades humanas (el proceso) mediante el cual el Sistema-Producto es diseñado*”. En otras palabras, el proceso tiene implicaciones organizacionales, técnicas y humanas, que son propias al proyecto de construcción del SS, y que influyen directamente sobre este. Es importante destacar que el sub-modelo del proceso resume las actividades típicas que coexisten en el desarrollo de SS, mientras que en el caso del sub-modelo del producto no todas las características pueden coexistir. Estudiar todo el proceso de desarrollo del SS estimado permite obtener una visión sistémica de éste, así como brindar observaciones más completas a la empresa.

Una vez estimada la calidad del proceso, se continúa con la última fase propuesta.

4.3. Fase 3: Integración de las mediciones de los sub-modelos de la calidad del producto y la calidad del proceso.

En esta fase se realiza la “integración” de la medición del producto y de la medición del proceso, para obtener la estimación de la calidad sistémica. Los niveles de calidad sistémica se proponen en la Tabla 6.

Tabla 6. Nivel de Calidad Sistémica Global a partir del nivel de Calidad del Producto y el nivel de Calidad del Proceso.

Nivel de Calidad Producto	Nivel de Calidad Proceso	Calidad Sistémica
Básico	-	Nulo
Básico	Básico	Básico
Intermedio	-	Nulo
Intermedio	Básico	Básico
Avanzado	-	Nulo
Avanzado	Básico	Intermedio
Básico	Intermedio	Básico
Intermedio	Intermedio	Intermedio
Avanzado	Intermedio	Intermedio
Básico	Avanzado	Intermedio
Intermedio	Avanzado	Intermedio
Avanzado	Avanzado	Avanzado

Como puede observarse en la Tabla 6, esta propuesta obedece a la necesidad de mantener un equilibrio entre las distintas dimensiones de la Calidad de los SS; es por ello que, la Calidad del Producto de Software tiene igual peso que la Calidad del Proceso de Desarrollo de Software. Se considera que la aplicación del modelo permitirá ajustar con mayor precisión este “equilibrio”.

La integración de las medidas de calidad de los sub-modelos, estima la calidad sistémica como una balanza; es decir, si el nivel de calidad de uno de los sub-modelos es menor que el nivel del otro sub-modelo, entonces la balanza no estará estable y por ello se inclinará hacia el nivel de menor calidad. Esto se debe a que si la Calidad del Producto de Software o la Calidad del Proceso de Desarrollo no cumplen con las características necesarias para tener un nivel más alto de calidad, implicará directamente que la calidad sistémica tampoco cumpla con las características necesarias para tener un nivel de calidad superior.

Cabe destacar que se realizó un prototipo del algoritmo de aplicación de MOSCA, el cual está 100% funcional. Las Figuras 2 y 3 presentan dos de las interfaces del prototipo donde se muestra el informe final de la ejecución del algoritmo.

Form1

INFORME FINAL

Nivel de Calidad Sistemica
El nivel de calidad sistémico que presenta la empresa dada es:
CALIDAD BASICA

Nivel de Calidad del Producto de Software
Debido a que el producto de software presenta un nivel de calidad:
CALIDAD INTERMEDIO

Nivel de Calidad del Proceso de Desarrollo
Debido a que el proceso de desarrollo presenta un nivel de calidad:
CALIDAD BASICA

Para obtener las características del producto y del proceso que deben ser mejoradas oprimir el botón SIGUIENTE. Si desea salir del sistema oprima el botón SALIR

SALIR SIGUIENTE

Figura 2. Interfaz que muestran el resultado final de la ejecución del algoritmo de MOSCA.

Form1

INFORME FINAL

Características del producto de software que deben ser mejoradas:

- MAB.7 Encapsulamiento
- MAB.14 Parametrizado
- MAB.2 Capacidad de cambio
- USA.3 Documentado

Características del proceso de desarrollo que deben ser mejoradas:

- CUS.2 Proceso de Suministro
- SUP.3 Proceso de Aseguramiento de la Calidad
- SUP.7 Proceso de Auditoria
- ORG.8 Proceso de Medición

IMPRIMIR SALIR

Figura 3. Interfaz que indica las características del producto y del proceso que deben ser mejoradas.

Una vez definido el algoritmo de aplicación de MOSCA, fue necesario probarlo para establecer su idoneidad y facilidad de uso; esto permitió ajustarlo según los resultados. Es por ello que en la siguiente sección se describe la aplicación del algoritmo a un caso de estudio.

5. CASO DE ESTUDIO: APLICACIÓN DEL ALGORITMO DE EVALUACIÓN

El caso de estudio incluyó dos (2) organizaciones venezolanas desarrolladoras de productos de software que expresaron su disposición a participar en el estudio. En ambas empresas se seleccionó el producto de software emblemático de cada organización y que, además, su desarrollo fuese de reciente fecha. Para el caso de la Empresa A, se seleccionó un sitio Web, a través del cual se pueden realizar operaciones de compra con tarjeta de crédito, consultar precios y obtener información de la misma empresa. Para el caso de la Empresa B, se seleccionó una

aplicación que permite la administración y control de la ejecución de los procesos funcionales de una organización.

Se aplicaron dos tipos de cuestionarios. El primero enfocado al producto y dirigido a tres grupos de evaluación: el Líder del Proyecto, Desarrolladores-Analistas y Usuarios. El segundo tipo de cuestionario está enfocado al proceso de desarrollo y va dirigido a dos grupos de evaluación: el Líder del Proyecto y los Desarrolladores-Analistas. Ambos cuestionarios se desarrollaron a partir de las métricas que conforman MOSCA y presentan un alto grado de confiabilidad. Específicamente, el cuestionario del producto fue validado utilizando la medida de estabilidad (confiabilidad por test-retest) y arrojó resultados positivos (0.7 y 0.68), lo que indica que el cuestionario es altamente confiable. Para el caso del cuestionario del proceso fue validado utilizando la prueba de confiabilidad del coeficiente Alfa-Cronbach, resultando también ser positivo (0.659) y demostrando un alto grado de confiabilidad.

Los datos de las dos empresas fueron analizados tomando en cuenta: (1) las categorías del producto seleccionadas por la empresa junto con el evaluador, (2) las categorías del proceso, y (3) las características del producto y del proceso.

En primer lugar, se analizan los datos referentes al producto. Las categorías seleccionadas (aparte de *Funcionalidad*) por ambas empresas fueron: *Mantenibilidad* y *Usabilidad*. Se debe recordar que según el algoritmo del modelo MOSCA, la empresa debe seleccionar exactamente 2 categorías que identifiquen a su producto de software.

La Empresa A seleccionó la categoría *Usabilidad*, ya que el sitio Web debe ser un producto atractivo, entendible y fácil de utilizar para los usuarios del mismo. Lo más importante de esta aplicación es su front-end, por lo cual el mismo debe cumplir los requerimientos de la categoría *Usabilidad*. La otra categoría seleccionada fue *Mantenibilidad*, ya que el producto debe ser actualizado constantemente y por ello debe tener la capacidad de ser modificado sin ningún problema.

La Empresa B seleccionó la categoría *Usabilidad*, ya que su producto está destinado a diferentes tipos de usuarios y la dificultad en el uso del mismo debe ser mínima. Además, esta aplicación debe ser atractiva, ya que el éxito de la misma, dependerá del grado de satisfacción de los usuarios. La otra categoría seleccionada fue *Mantenibilidad*, ya que el producto de software está en constante desarrollo y debe ser capaz de aceptar cualquier tipo de modificaciones. Además, la aplicación debe ser actualizada y aceptar la inclusión de nuevos módulos sin necesidad de cambiar estructuralmente la misma.

5.1. Resultado parcial: Nivel de Calidad del producto de software.

Funcionalidad. En la Figura 4 se muestran los porcentajes alcanzados por el producto de software para ambas empresas, en cuanto a los requerimientos de calidad asociados a la *Funcionalidad*.

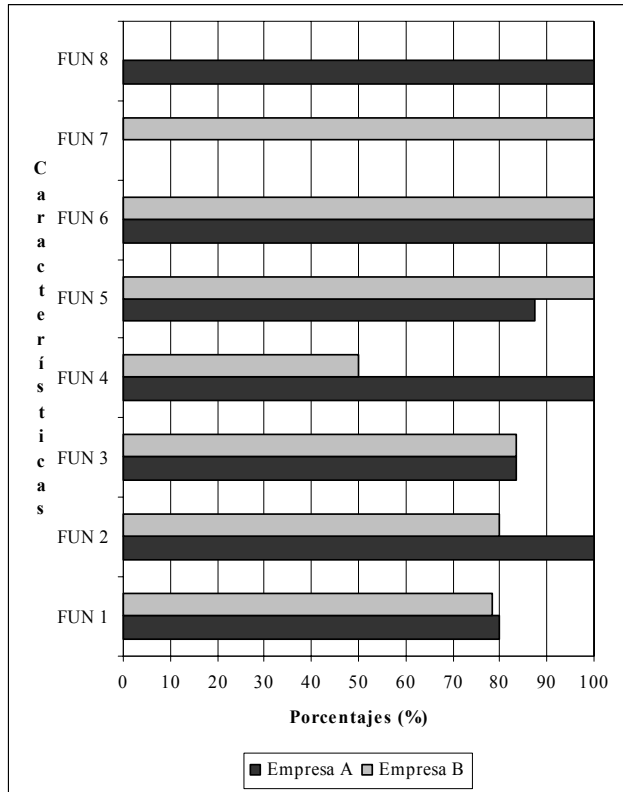


Figura 4. Porcentajes de satisfacción de los productos frente a las características de la *Funcionalidad*.

Como se puede observar en la Figura 4, para la Empresa A, siete (7) de las ocho (8) características de la categoría funcionalidad son altamente satisfechas; es decir, cumplen al menos un 75% de las métricas asociadas. Sólo la característica FUN 7 (*Encapsulado*) no fue satisfecha. Se debe recordar que si seis (6) de las características del producto de software son altamente satisfechas, entonces la categoría Funcionalidad está presente en el producto de software evaluado. En conclusión, como el número de características altamente satisfechas es siete (7), entonces **el producto de la Empresa A cumple con la categoría *Funcionalidad***. Paralelamente, para la Empresa B, sólo la característica FUN 4 (*Seguridad*) y la característica FUN 8 (*Especificado*) no son altamente satisfechas (ver Figura 4); es decir, no cumplen el 75% de las métricas asociadas. Se concluye entonces que **el producto de software de la Empresa B cumple con esta categoría**, ya que seis (6) de sus características son altamente satisfechas.

Usabilidad. La Figura 5 muestra los porcentajes alcanzados por las características asociadas a la categoría *Usabilidad* de los productos de software evaluados.

Según la Figura 5, **el producto de software evaluado de la Empresa A cumple con la categoría *Usabilidad***, ya que ocho (8) de las características son altamente satisfechas y ese es el número mínimo que propone el algoritmo de aplicación del modelo MOSCA para que la categoría Usabilidad se cumpla en un producto de software. Únicamente las características USA 7 (*Efectivo*) y USA 9 (*Documentado*) no fueron altamente

satisfechas, de hecho están ausentes en la categoría del producto.

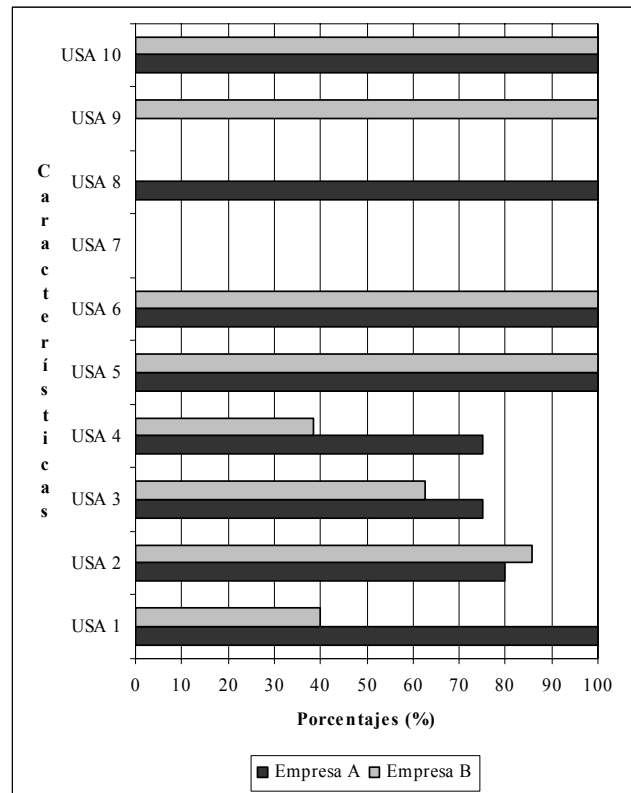


Figura 5. Porcentajes de satisfacción de los productos frente a las características de la *Usabilidad*.

Para el caso del producto de la Empresa B, las características USA 2 (*Capacidad de aprendizaje*), USA 5 (*Completo*), USA 6 (*Consistente*), USA 9 (*Documentado*) y USA 11 (*Auto-descriptivo*), según la Figura 5, son altamente satisfechas, mientras que las demás características no cumplen el 75% de las métricas asociadas. Esto indica que sólo 5 de las 11 características son altamente satisfechas, lo que implica que **el producto de software de la Empresa B no cumple con la categoría *Usabilidad***.

Mantenibilidad. Para analizar la *Mantenibilidad* que presenta el producto de software de ambas empresas, se tiene la Figura 6, en la cual se muestra los porcentajes alcanzados por las características asociadas a la categoría *Mantenibilidad*.

Para el caso de la categoría *Mantenibilidad*, se puede observar en la Figura 6 que sólo ocho (8) de las características el producto de la Empresa A asociadas a dicha categoría son altamente satisfechas, lo que implica que **el producto de software de la Empresa A no cumple con la categoría *Mantenibilidad***. Esto se debe a que según el algoritmo de aplicación de modelo MOSCA, para que un producto de software cumpla con la categoría *Mantenibilidad* debe tener al menos 11 de las 14 características altamente satisfechas (75%). Nótese que la característica MAB 11 (*Descriptivo*) está ausente en el producto de software. Por otra parte (ver Figura 6), las características MAB 1 (*Capacidad de análisis*), MAB 3 (*Estabilidad*), MAB 8 (*Atributos de madurez del software*), MAB 11 (*Sistema de estructura de información*),

MAB 11 (*Descriptivo*), MAB 13 (*Estructural*), no son altamente satisfechas por el producto de software de la Empresa B. En conclusión, **el producto de la Empresa B no es un producto mantenible; es decir, no cumple con la categoría *Mantenibilidad***, ya que sólo siete (7) características son altamente satisfechas.

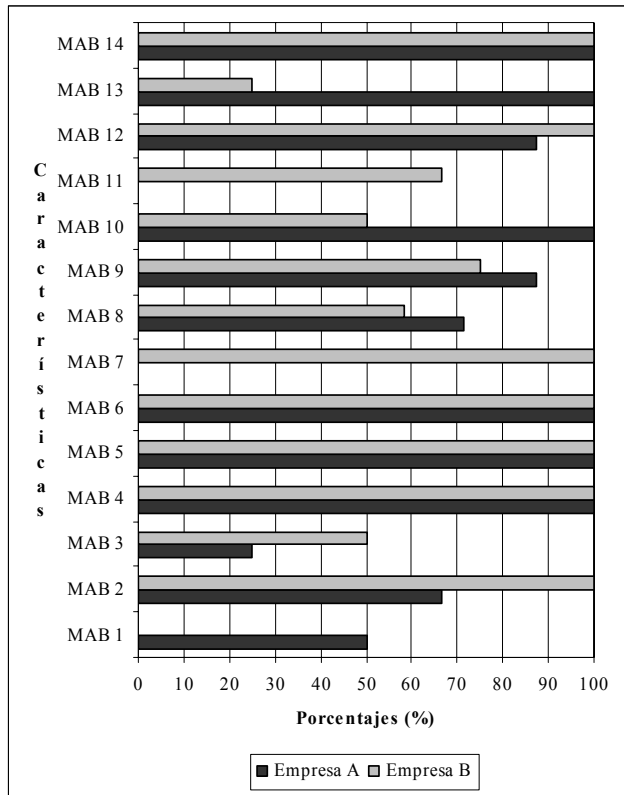


Figura 6. Porcentajes de satisfacción de los productos frente a las características de la *Mantenibilidad*.

Siguiendo con los estándares del algoritmo de aplicación del modelo MOSCA, se concluye que **el producto de software de la Empresa A presenta un nivel de calidad *Intermedio***, ya que cumple con dos (2) de las tres (3) categorías instanciadas, específicamente con la categoría *Funcionalidad* y *Usabilidad*, presentando deficiencias en la categoría *Mantenibilidad*. Por su parte, **el producto de software de la Empresa B presenta un nivel de calidad *Básico***, ya que únicamente cumple con la categoría *Funcionalidad*.

5.2. Resultado parcial: Nivel de Calidad del proceso de desarrollo.

Una vez obtenido el nivel de calidad del producto de software de ambas empresas, se debe evaluar el proceso de desarrollo de ambos productos para poder medir el nivel de calidad sistémico de las dos (2) empresas. Por ello, se presenta a continuación el análisis de los datos referentes al proceso de desarrollo de las empresas, concretamente las categorías *Cliente-Proveedor*, *Ingeniería*, *SopORTE*, *Gestión* y *Organizacional*.

Cliente-Proveedor. La Figura 7 muestra los porcentajes de las características asociadas a la categoría *Cliente-Proveedor* que presentan los procesos de desarrollo evaluados.

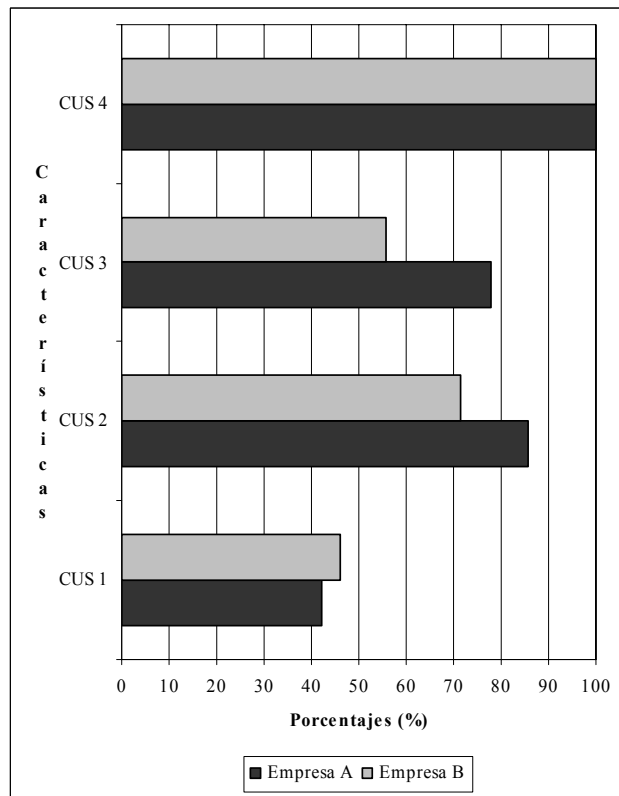


Figura 7. Porcentajes de satisfacción de las empresas frente a las características de *Cliente-Proveedor*.

Como se puede observar en la Figura 7, tres (3) de las cuatro (4) características de la categoría *Cliente-Proveedor* de la Empresa A, son altamente satisfechas. Esto indica que esta categoría se cumple en el proceso de desarrollo de la Empresa A, debido a que el algoritmo de aplicación del modelo MOSCA establece que si tres (3) de las cuatro (4) características de la categoría *Cliente-Proveedor* son altamente satisfechas, entonces **la categoría *Cliente-Proveedor* esta presente en el proceso de desarrollo de la Empresa A**. Únicamente la característica CUS 1 (*Proceso de adquisición del sistema o producto de software*) no fue altamente satisfecha por la Empresa A. Por otro lado, como se puede observar en la Figura 7, la única característica que es altamente satisfecha por la Empresa B en esta categoría, es CUS 4 (*Proceso de operación*); es decir, las características CUS 1 (*Proceso de adquisición del sistema o producto de software*), CUS 2 (*Proceso de suministro*) y CUS 3 (*Elicitación de requerimientos*) no fueron altamente satisfechas. Esto indica que **la categoría *Cliente-Proveedor* no se cumple en el proceso de desarrollo de la Empresa B**.

Ingeniería. Para analizar la categoría *Ingeniería* del proceso de desarrollo de ambas empresas, se presenta la Figura 8, donde se pueden observar el porcentaje obtenido por las características de dicha categoría.

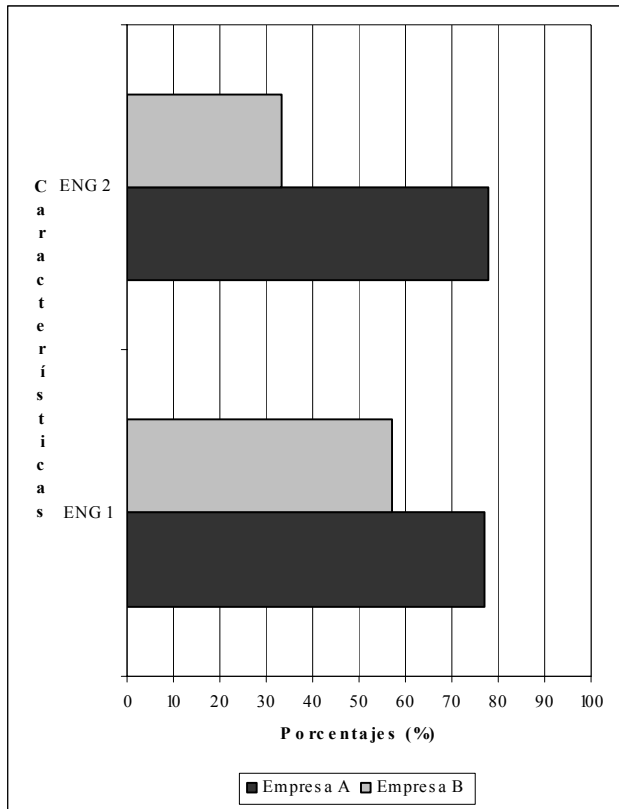


Figura 8. Porcentajes de satisfacción de las empresas frente a las características de Ingeniería.

Según la Figura 8, las dos características de la categoría Ingeniería ENG 1 (*Proceso de desarrollo*) y ENG 2 (*Proceso de mantenimiento de software y sistemas*), son altamente satisfechas en la Empresa A; es decir, ambas características cumplen con al menos un 75% de las métricas asociadas. Esto indica que **esta categoría se cumple en el proceso de desarrollo de la Empresa A**. Para el caso de la categoría Ingeniería del proceso de desarrollo de la Empresa B, se puede observar en la Figura 8 que ninguna de las dos características son altamente satisfechas; es decir, ni ENG 1 (*Proceso de desarrollo*) ni ENG 2 (*Proceso de mantenimiento de software y sistemas*) fueron cumplidas. En ese sentido, se puede afirmar que **la categoría Ingeniería no está presente en el proceso de desarrollo de la Empresa B**.

Soporte. La Figura 9 muestra el porcentaje obtenido por las características de la categoría *Soporte* asociados a los procesos de desarrollo evaluados.

Según el algoritmo de aplicación del modelo MOSCA, para que se cumpla la categoría *Soporte* en un proceso de desarrollo, se debe tener al menos seis (6) características altamente satisfechas. En el caso de la Empresa A, como se puede observar en la Figura 9, sólo cuatro (4) características son altamente satisfechas: SUP 4 (*Proceso de verificación*), SUP 5 (*Proceso de validación*), SUP 6 (*Proceso de revisión conjunta*) y SUP 8 (*Proceso de resolución de problemas*). Por esta razón, **el proceso de desarrollo de la Empresa A no cumple con la categoría Soporte en su proceso de desarrollo**. Además se puede observar que la característica

SUP 7 (*Proceso de auditoría*) no es satisfecha. Por otro lado, se puede observar en la Figura 9 que **la categoría Soporte no está presente en el proceso de desarrollo de la Empresa B**, ya que ninguna de sus características es altamente satisfecha.

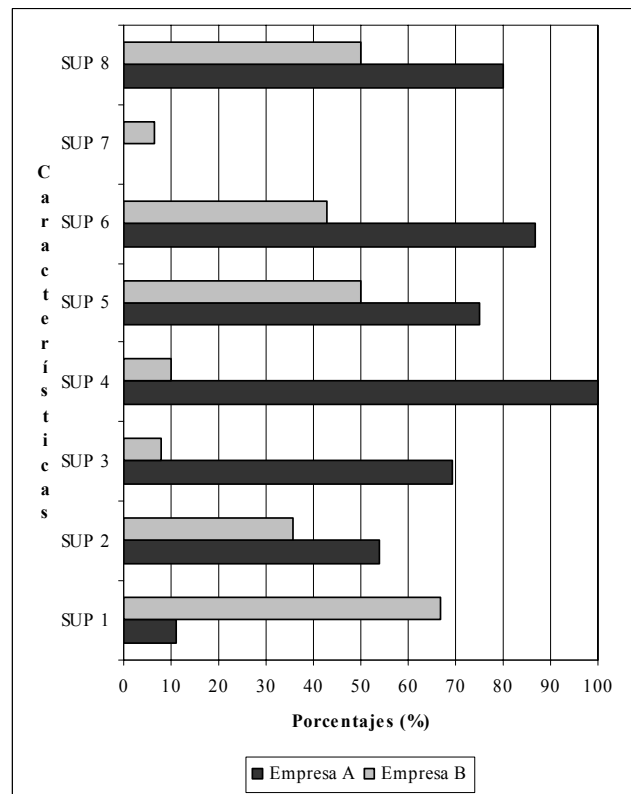


Figura 9. Porcentajes de satisfacción de las empresas frente a las características de Soporte.

Gestión. Para poder analizar el grado de satisfacción de la categoría *Gestión* en el proceso de desarrollo de ambas empresas, se presenta la Figura 11, la cual muestra el porcentaje obtenido por las características de dicha categoría.

La Figura 11 muestra que **la categoría Gestión en el proceso de desarrollo de la Empresa A no se cumple**, ya que para que dicha categoría esté presente en un proceso de desarrollo (según el algoritmo de aplicación de MOSCA), al menos tres (3) de las cuatro (4) características deben estar altamente satisfechas; para el caso de la Empresa A, sólo una (1) característica es altamente satisfecha: MAN 1 (*Proceso de gestión*). De igual manera, para el caso de la Empresa B (ver Figura 11), la única característica altamente satisfecha es MAN 1 (*Proceso de gestión*). Por esta razón, **esta categoría también está ausente en el proceso de desarrollo de la Empresa B**.

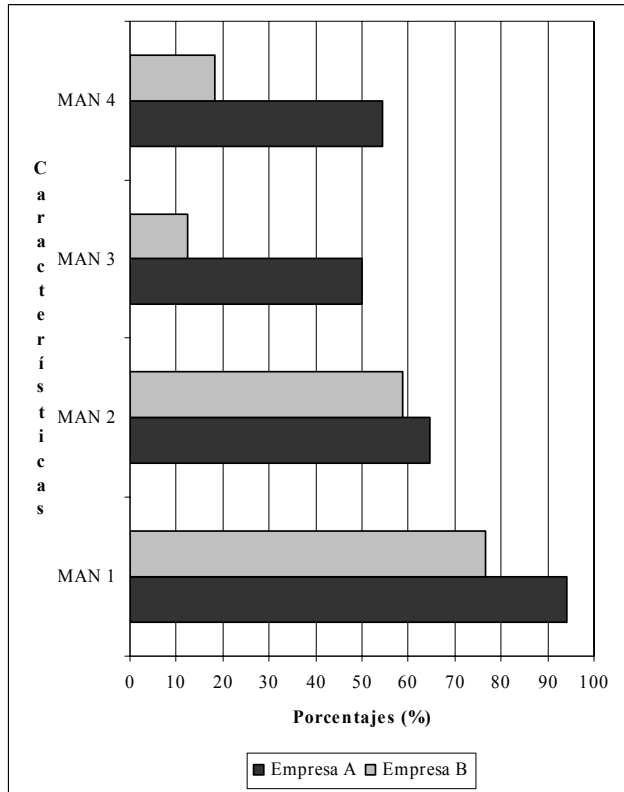


Figura 11. Porcentajes de satisfacción de las empresas frente a las características de *Gestión*.

Organizacional. La Figura 12 muestra el porcentaje obtenido por las características de la categoría *Organizacional* asociada al proceso de desarrollo de ambas empresas.

Como se puede observar en la Figura 12, sólo cinco (5) características de la categoría *Organizacional* son altamente satisfechas por la Empresa A, lo que indica que **el proceso de desarrollo de la Empresa A no cumple con la categoría *Organizacional***. Para que un proceso de desarrollo cumpla con la categoría *Organizacional*, debe tener al menos siete (7) características altamente satisfechas. Por otro lado, como se puede observar en la Figura 12, para el caso del proceso de desarrollo de la Empresa B, tan sólo una (1) característica es altamente satisfecha: ORG 7 (*Proceso de infraestructura*); por esa razón, **la categoría *Organizacional* no está presente en el proceso de desarrollo de la Empresa B**.

Analizando los resultados anteriores se tiene que **el proceso de desarrollo de la Empresa A tiene un nivel de calidad *Básico***, ya que cumple únicamente con las categorías *Cliente-Proveedor* e *Ingeniería*, siendo éstas las necesarias para obtener este nivel de calidad. Por otro lado, se concluye que **el proceso de desarrollo de la Empresa B tiene un nivel de calidad *Nulo***, ya que no cumple con las categorías *Cliente-Proveedor* e *Ingeniería*, que son las mínimas para optar por un nivel de calidad *Básico*.

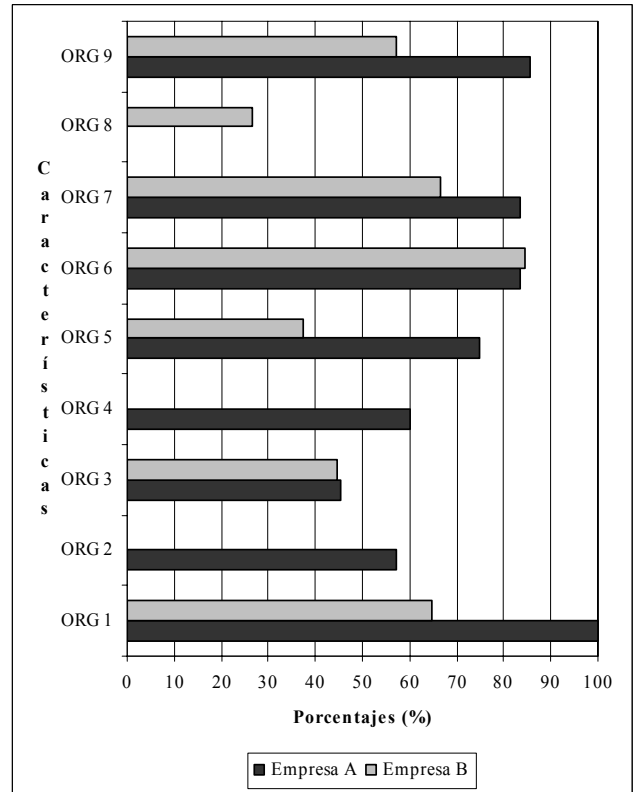


Figura 12. Porcentajes de satisfacción de las empresas frente a las características de *Organizacional*.

5.3. Resultado final: Nivel de Calidad Sistémico.

Tomando en cuenta que el producto de software de la Empresa A obtuvo un nivel de calidad *Intermedio* y sabiendo, además, que el nivel de calidad del proceso de desarrollo fue *Básico*, entonces se puede concluir, siguiendo los estándares que describe el algoritmo del modelo MOSCA, que **el nivel de calidad sistémico que presenta la Empresa A es *Básico***. Cabe acotar que los procesos que según el modelo MOSCA impactan en la categoría *Mantenibilidad* no son satisfechos, con excepción de ENG 1 (*Proceso de desarrollo*) que apenas satisface el mínimo requerido por el modelo. Estos procesos son SUP 1 (*Proceso de documentación*), SUP 2 (*Proceso de gestión de configuración*), MAN 3 (*Proceso de gestión de calidad*), SUP 3 (*Proceso de aseguramiento de la calidad*) y SUP 7 (*Proceso de auditoría*). Esto explica el nivel de calidad expuesto por la categoría *Mantenibilidad*.

Por otra parte, **el nivel de calidad sistémico que presenta la Empresa B es *Nulo***. Esto se debe a que tanto el producto de software como el proceso de desarrollo del mismo, presentan grandes deficiencias. Esencialmente se pudo observar que el proceso de desarrollo no sigue ningún tipo de metodología ni de controles, tanto internos como externos. Se debe destacar que los procesos que impactan sobre las categorías del producto que no fueron satisfechas, *Mantenibilidad* y *Usabilidad*, obtuvieron niveles de calidad muy bajos, lo cual explica el nivel de calidad

final del producto de software de la Empresa B. Los procesos que impactan sobre la *Mantenibilidad* del producto son: ENG 1 (*Proceso de desarrollo*), SUP 1 (*Proceso de documentación*) y SUP 2 (*Proceso de gestión de configuración*); nótese que ninguno de estos procesos es altamente satisfecho. Además, los procesos que impactan sobre la categoría *Usabilidad* son: CUS 3 (*Proceso de licitación de requerimientos*), SUP 1 (*Proceso de documentación*), SUP 6 (*Proceso de revisión conjunta*). Como se pudo observar, ninguno de estos procesos es altamente satisfecho. Por último, se puede observar que ninguno de los procesos que impactan sobre todas las categorías del producto es altamente satisfecho en el proceso de desarrollo de la Empresa B. Estos procesos son: MAN 3 (*Proceso de gestión de calidad*), SUP 3 (*Proceso de aseguramiento de la calidad*), SUP 7 (*Proceso de auditoría*).

6. CONCLUSIONES

Para estimar la calidad de los SS se hace necesario estimar la calidad del software como producto y como proceso. Para ello se formulan los Modelos de Calidad. En este trabajo se propone un algoritmo que estima la calidad en base al Modelo MOSCA. Este algoritmo constituye una herramienta efectiva de análisis y estimación de la Calidad Global Sistémica, ya que se analizan aspectos del producto, del proceso, su relación entre ellos y su relación con el medio ambiente.

A través de la aplicación del algoritmo, el modelo MOSCA especifica los procesos que se deben mejorar en la empresa y las características que no son satisfechas por el producto de software desarrollado. De esa manera, el modelo no sólo arroja un nivel de Calidad, sino que también explica cómo se obtuvo dicho nivel y propone las acciones que se deben tomar para mejorar los procesos así como las características del producto que no se están cumpliendo. Además, sirve como una auditoría para las empresas evaluadas, ya que expone las características del producto y del proceso que deben ser mejoradas.

El algoritmo se aplicó en dos empresas venezolanas. Las empresas evaluadas mostraron una alta aceptación, además que se enriquecieron con los resultados obtenidos, sus respectivas explicaciones y las mejoras que debían llevar a cabo para mejorar su Calidad Sistémica. Los resultados obtenidos han confirmado la percepción de las personas involucradas con ambos productos, lo cual respalda el nivel de confiabilidad del algoritmo aplicado, así como que éste soporta fielmente el objetivo del modelo MOSCA.

Hasta los momentos se ha probado la aplicabilidad e idoneidad del algoritmo propuesto y del modelo MOSCA; sin embargo, para que el mismo tenga un carácter más genérico y de mayor aceptabilidad dentro de la comunidad interesada en el área, en este momento se está aplicando MOSCA, en conjunto con el algoritmo, a otros SS en distintos dominios.

AGRADECIMIENTOS

Esta investigación fue financiada por el Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT) de la República Bolivariana de Venezuela, a través del proyecto S1-2000000437. Los autores desean dar las gracias a J. Martínez por su valiosa colaboración en la culminación de esta investigación.

REFERENCIAS

- [1] Bass, L., Clements, P., and Kazman, R. *Software Architecture in Practice*, SEI Series in Software Engineering, Addison-Wesley Longman, Inc., Reading, Massachusetts, 1998.
- [2] Callaos, N. and Callaos, B. "Designing with Systemic Total Quality", in *International Conference on Information Systems – ISAS'96*, N. Callaos and B. Sánchez (eds.), International Institute of Informatics and Systemics, Orlando, 1993, pp. 548-560.
- [3] Dromey, R., "Cornering the Chimera", *IEEE Software* (13:1), 1996, pp. 33-43.
- [4] Humphrey, W. S. *Managing the Software Process*, Addison-Wesley Pub. Co., London, UK, 1998, p. 7.
- [5] International Organization for Standardization, *Software Process Assessment*, TR 15504, on-line, WG 11: Software Process Assessment, <http://www.sqi.gu.edu.au/spice/>. 1999.
- [6] Ivancevich, J.; Lorenzi, P.; Skinner, S.; Crosby, P. "Management Quality and Competitiveness". Second Edition. IrWin/McGraw-Hill, Boston, Massachusetts, 1997, p. 237.
- [7] JTC 1/SC 7. *ISO/IEC 9126:1991 Information technology - Software product evaluation - Quality characteristics and guidelines for their use*, JTC 1 Organization, Montréal, Québec, 1991.
- [8] Mendoza, L.; Pérez, M.; Rojas, T. *Modelo Sistémico para Estimar la Calidad de los Sistemas de Software (MOSCA)*. *Acta Científica Venezolana* (52:3), 2001, p. 435, <http://www.lisi.usb.ve/publicaciones/ASOVAC%202001/Mosca.zip>
- [9] Ortega, M. *Modelo de Calidad del producto de Software con un enfoque sistémico*, Universidad Simón Bolívar, Caracas, 2000, pp. 123-125.
- [10] Ortega, M., Pérez, M. A., and Rojas, T. "A Model for Software Product Quality with a Systemic Focus," in *4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000*, N. Callaos and B. Sánchez (eds.), International Institute of Informatics and Systemics, Orlando, 2000, pp. 395-401, <http://www.lisi.usb.ve/publicaciones/calidad2.zip>.
- [11] Pérez, M. A., Rojas, T., Mendoza, L., and Grimán, A. "Systemic Quality Model for System Development Process: Case Study", in *Seventh Americas Conference on Information Systems – AMCIS 2001*, D. Strong and D. Straub (eds.), Association for Information Systems, Boston, 2001, pp. 1297-1304, <http://www.lisi.usb.ve/publicaciones/calidad11.zip>.
- [12] Pressman, R. *Ingeniería del Software. Un enfoque práctico*, 5ta edición, McGraw-Hill Interamericana, Madrid, España, 2002, p. 48.
- [13] Rojas, T., and Pérez, M., "Improvement in the Development of Information Systems by increasing its Process Effectiveness", in *InterSymp '95 - 5th International Symposium on Systems Research, Informatics and Cybernetics*, N. Callaos y B. Sánchez (eds.), International Institute of Informatics and Systemics, Baden-Baden, 1995, pp. 36-42.
- [14] Voas, J., "Software Quality's Eight Greatest Myths", *IEEE Software* (16:5), September/October, 1999, pp. 740-745.