

MODELO DE CALIDAD (MOSCA⁺) PARA EVALUAR SOFTWARE DE SIMULACIÓN DE EVENTOS DISCRETOS

GLADYS RINCÓN¹, MARÍA PÉREZ², SARA HERNÁNDEZ¹

¹Departamento de Procesos y Sistemas, Universidad Simón Bolívar Caracas – Venezuela

²Departamento de Procesos y Sistemas – LISI, Universidad Simón Bolívar Caracas – Venezuela
grincon@usb.ve; movalles@usb.ve; sara_usb@hotmail.com

MARINELLY ALVAREZ³

³ INTEVEP, S.A. Centro de Investigación y Apoyo Tecnológico. Filial de Petróleos de Venezuela, S.A.
marinellyalvarez@cantv.net

RESUMEN

El auge, en los últimos años, de la técnica de simulación ha traído como consecuencia un incremento de este tipo de software en el mercado. La complejidad y dificultad inherente a la evaluación de cualquier clase de software, aunado al abanico de posibilidades que existen de Software de Simulación de Eventos Discretos (SSED), representan un dilema para las organizaciones a la hora de una selección adecuada. Por lo tanto esta investigación propone desarrollar un modelo de especificaciones de calidad que permite evaluar SSED aplicables al área de logística de suministro, manejo, transporte y distribución de hidrocarburos en la industria petrolera. Como producto de esta investigación se generó el Modelo Sistemático de Calidad (MOSCA⁺), identificándose 40 criterios y 131 subcriterios que soportan el proceso de evaluación de SSED en esta área particular de aplicación. En este trabajo se describe en detalle el modelo propuesto MOSCA⁺ y los resultados obtenidos de su utilización para la evaluación de SSED en una organización que presta servicios de consultoría en el área de logística de la industria petrolera.

I. INTRODUCCIÓN

Evaluar software es una tarea compleja que en ocasiones puede conducir a las organizaciones a tomar decisiones inadecuadas. Según Rojas *et al* (2001), la utilización de enfoques que tienden a subestimar o ignorar aspectos importantes para la selección y uso de la tecnología que compran, puede conducir a escoger software que no se ajusta a las necesidades o intereses específicos de la organización. Además, destacan lo difícil que es diseñar un conjunto de criterios adecuados para evaluar todas los softwares y medir sus beneficios.

Por otra parte, Nikoukaran, Hlupic y Paul (1998) indican que la técnica de simulación ha tomado un auge creciente en un amplio rango de aplicaciones, incrementándose el número de software de alta calidad con fines específicos que se encuentran en el mercado. Asimismo, Hlupic (1999) afirma que el conjunto de herramientas de Software de Simulación de Eventos Discretos (SSED) disponibles conforman un abanico de posibilidades que representan un dilema para las organizaciones a la hora de una selección adecuada.

Nikoukaran *et al* (1998) explican que en la selección de software de simulación, éstos se evalúan bien sea por sus propios méritos o comparándolos con otros, pero en cualquier caso se requiere una lista de criterios para hacer una selección adecuada.

Asimismo, agregan, que una vez identificados, los criterios deben ser estructurados en términos de un modelo de decisión, capaz de maximizar la racionalidad en la transferencia de la tecnología correspondiente.

Sobre la base de lo anterior, los SSED no son ajenos a la complejidad inherente asociada a la selección de herramientas de computación, por lo tanto cabe la pregunta ¿Cómo evaluar los SSED que más se adecuen a los fines de una organización?.

Se debe tomar en cuenta, que utilizando un proceso formal de evaluación de software se reduce la posibilidad de adquirir productos inadecuados o innecesarios, y por tanto, se sugiere usar métodos de evaluación que permitan medir factores claves en el producto a adquirir. (Rojas *et al*, 2000).

Olson sostiene que múltiples criterios requieren seguir ciertos pasos para identificar objetivos, ordenarlos jerárquicamente, y luego medir cómo se comportan las alternativas disponibles con cada criterio (citado por Nikoukaran *et al* (1998))

El objetivo de este trabajo es proponer un modelo de especificaciones de calidad que permita evaluar SSED, a ser empleado en logística de suministro, manejo, transporte y distribución de hidrocarburos en la industria petrolera.

El presente trabajo está estructurado en las siguientes secciones: en la primera se explica a manera de antecedente el *Modelo Sistemico de Calidad* (MOSCA). Después se detalla la propuesta realizada a partir de MOSCA, denominada MOSCA⁺. Luego se expone la aplicación de MOSCA⁺ para evaluar *SSED* aplicado al estudio de un caso y por último se presentan las conclusiones correspondientes.

II. MODELO SISTÉMICO DE CALIDAD (MOSCA)

En la búsqueda de trabajar con un sistema organizado que permita ordenar la información sobre *SSED* y, que conduzca a obtener una herramienta que sea útil para evaluar este tipo de software, se aplicó el modelo propuesto por Mendoza *et al* (2002), *Modelo Sistemico de Calidad* (MOSCA), cuyo propósito es estimar la calidad sistémica dentro de una organización desarrolladora de software.

En la Figura 1 se muestra el diagrama del Modelo Sistemico de Calidad (MOSCA). En esta Figura se observa como el modelo MOSCA está constituido por

dos submodelos (el submodelo del *Producto* y el submodelo del *Proceso*).

De acuerdo con los objetivos previstos en la evaluación, del modelo MOSCA se puede tomar el submodelo del *Producto*, el submodelo del *Proceso*, o ambos inclusive. El primero de ellos, se utiliza para evaluar software ya elaborado, mientras que el segundo se emplea cuando además, se requiere evaluar el proceso de su desarrollo. A continuación se explican el resto de los niveles que conforman el modelo MOSCA:

Nivel 0: Dimensiones. Las cuatro dimensiones propuestas por MOSCA son: *Eficiencia* y *Efectividad* del *Proceso*, *Eficiencia* y *Efectividad* del *Producto*.

Nivel 1: Categorías. Este nivel contempla 11 categorías, de las cuales 6 pertenecen al *Producto* y 5 al *Proceso* de desarrollo. En las Tablas 1 y 2, se muestran las definiciones de las categorías pertenecientes al *Producto* y al *Proceso*, respectivamente.

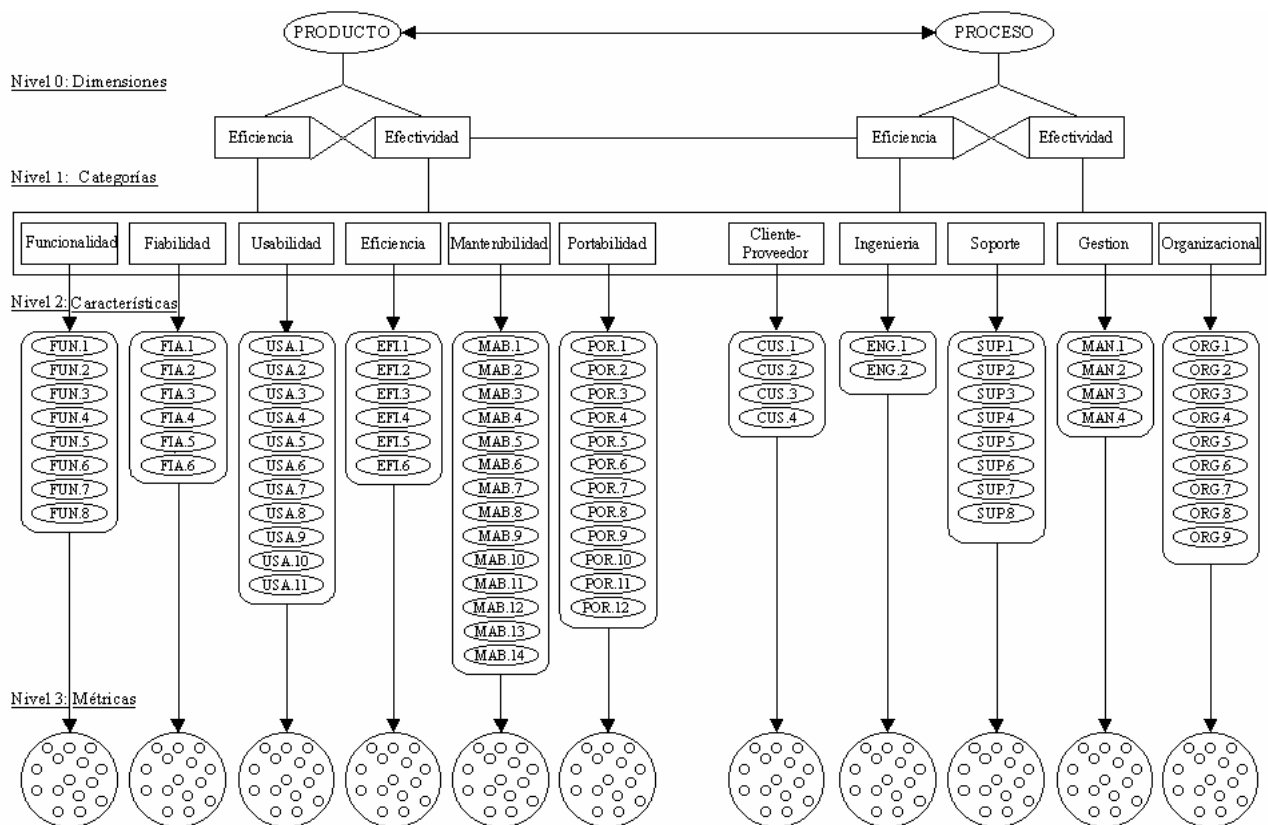


Figura 1: Diagrama del Modelo Sistemico de Calidad (MOSCA)

Nivel 2: Características. MOSCA plantea que cada categoría tiene asociada un conjunto de características, las cuales definen las áreas claves que se deben satisfacer para lograr asegurar y controlar la calidad del *Producto* y/o del *Proceso*.

Nivel 3: Métricas. Cada característica posee una serie de métricas que están relacionadas con las cualidades que se desean evaluar del software y/o del proceso de su desarrollo.

III. MODELO SISTÉMICO DE CALIDAD - MOSCA⁺ - UNA PROPUESTA

Para identificar cuáles aspectos de los propuestos por MOSCA (Figura 1) son útiles para evaluar *SSED*, se realizó un análisis de los niveles que conforman el modelo MOSCA y a partir de allí, se definieron los

niveles de esta aplicación considerando los planteamientos que se explican a continuación:

- **Selección del submodelo:** Se seleccionó únicamente el submodelo del *Producto*, excluyendo el submodelo del *Proceso*, ya que el objetivo de esta investigación es evaluar *SSED* ya elaborados.
- **Selección de la dimensión (Nivel 0):** Las dimensiones asociadas al submodelo del *Proceso* (eliminado en el paso anterior) fueron descartadas. También se omitió la dimensión *Eficiencia* del *Producto*, ya que para evaluarla se requiere el código fuente del software, el cual no estaba disponible para esta evaluación.

Tabla 1: MOSCA - Categorías para el submodelo del Producto

<i>Categoría</i>	<i>Definición</i>
<i>Funcionalidad (FUN)</i>	Es la capacidad de la herramienta para proveer funciones que cumplan con necesidades específicas o implícitas, cuando es utilizada bajo ciertas condiciones.
<i>Fiabilidad (FIA)</i>	Es la capacidad del producto para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas.
<i>Usabilidad (USA)</i>	Se refiere a la capacidad del producto para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
<i>Eficiencia (EFI)</i>	Es la capacidad del producto para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas.
<i>Mantenibilidad (MAN)</i>	Es la capacidad del software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones ante cambios del ambiente, requerimientos y especificaciones funcionales.
<i>Portabilidad (POR)</i>	Es la capacidad del producto para ser transferido de un ambiente a otro.

Tabla 2: MOSCA - Categorías para el submodelo del Proceso

<i>Categoría</i>	<i>Definición</i>
<i>Cliente - Proveedor (CUS)</i>	Está conformada por procesos que impactan directamente al cliente, apoya el desarrollo y la transición del Software hasta el cliente, y provee la correcta operación y uso del producto o servicio de software.
<i>Ingeniería (ENG)</i>	Consiste en procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el Sistema y su documentación.
<i>Soporte (SUP)</i>	Consta de procesos que pueden ser empleados por cualquiera de los procesos (incluyendo a los de soporte) en varios niveles del ciclo de vida de adquisición.
<i>Gestión (MAN)</i>	Consiste en procesos que contienen prácticas de naturaleza genérica, que pueden ser utilizadas por cualquier personaje que dirija algún tipo de proyecto o proceso, dentro de un Ciclo de Vida de Primario
<i>Organizacional (ORG):</i>	Esta relacionado con procesos que establecen las metas comerciales de la organización y desarrollan bienes (valores) de proceso, producto y recurso, que ayudarán a la organización a alcanzar sus metas en los proyectos.

- **Selección de las categorías (Nivel 1):** Mendoza *et al* (2002) indica que la categoría *FUNCIONALIDAD* es la más importante para la estimación de la calidad, debido a que identifica la capacidad del software para cumplir con las funciones para las que fue fabricado. Por tal razón, MOSCA exige que se evalúe esta categoría y propone que de las cinco categorías restantes del submodelo del *Producto* (Tabla 1), se seleccionen otras dos categorías que el software deba cumplir y que por lo tanto ameriten ser evaluadas, en este caso: *USABILIDAD* y *EFICIENCIA*.

La selección de la categoría *USABILIDAD* se sustentó en la importancia que tiene para la organización y los usuarios, el poder contar con un software que sea fácil de aprender, entender y utilizar a fin de minimizar los tiempos y los costos de entrenamiento, desarrollo y mantenimiento de los modelos (Umeda *et al*, 1997).

La categoría *EFICIENCIA* se escogió porque es fundamental conocer con cuánta celeridad se pueden obtener los resultados, su compatibilidad con los sistemas de software y hardware existentes en la organización así como los requerimientos extras para la instalación del software, los cuales involucrarían costos adicionales.

- **Selección de las características (Nivel 2):** Una vez definidas las categorías a ser evaluadas, se analizan el conjunto de características de MOSCA asociadas a cada categoría, excluyendo aquellas que carecen de pertinencia en el contexto de la evaluación. A continuación se indican las características seleccionadas y las causas de exclusión de las otras categorías:

FUNCIONALIDAD: Las características seleccionadas para esta categoría son: *Ajuste a los propósitos (APR)*, *Interoperabilidad (INT)* y *Seguridad (SEG)*, antes FUN.1, FUN.3 FUN.4, respectivamente. Las características *FUN.5* a la *FUN.8* se descartaron ya que están asociadas a la dimensión *Eficiencia del Producto*, excluida en la definición del *Nivel 0*.

También, se eliminó la característica *Precisión (FUN.2)*, debido a la dificultad para medir la fidelidad de los resultados del modelo respecto a datos reales (Shannon, 1975). En virtud de la naturaleza aleatoria de las variables que caracterizan los modelos de simulación estocásticos, éstos generan salidas que son también aleatorias y por ende sólo representan un estimado de las verdaderas características del sistema (Law *et al*, 1991).

USABILIDAD: Las características seleccionadas para esta categoría son: *Facilidad de comprensión* (antes

USA.1) y *Capacidad de aprendizaje* (antes *USA.2*) convertidas en una sola llamada *Facilidad de comprensión y aprendizaje (FCA)*, además de *Interfaz Gráfica (IGR)*, antes *USA.3* y *Operabilidad (OPR)*, antes *USA.4*. Se descartaron las características desde la *USA.5* a la *USA.10*, ya que están asociadas a la dimensión *Eficiencia del Producto*, excluida en la definición del *Nivel 0*.

EFICIENCIA: Esta categoría fue evaluada a través de *Comportamiento en el tiempo (CTI)*, antes *EFI.1* y *Utilización de recursos (URR)*, antes *EFI.2*. En esta categoría se excluyeron las características *EFI. 3* a la *EFI. 6*, debido a que están asociadas a la dimensión *Eficiencia del Producto*, excluida en la definición del *Nivel 0*.

- **Selección de las métricas (Nivel 3).** MOSCA sugiere que, al medir la calidad del producto, a cada característica se le asocien métricas que estén relacionadas con las cualidades o atributos de la herramienta que se desean evaluar (Mendoza *et al*, 2002). Sin embargo, Kitchenham (1996) sostiene que cuando los atributos escogidos para la evaluación son conceptos complejos, éstos pueden ser descompuestos en ítems más simples, los cuales, si se desea, pueden seguir siendo sucesivamente descompuestos. No obstante, agrega que el exceso de atributos puede traer como consecuencia invertir mucho tiempo en la evaluación. Por lo tanto, es necesario hacer un balance entre la profundidad de la evaluación, el nivel de confianza deseado y las dificultades prácticas del proceso.

Ahora, la complejidad conceptual de los sistemas logísticos de suministro, manejo, transporte y distribución de hidrocarburos en la industria petrolera, están caracterizados por procesos en los cuales interactúan una serie de factores que producen fenómenos probabilísticos y ocurrencias de eventos discretos (Falconer, *et al*, 1998), acompañados de operaciones continuas.

Esta situación exige que los *SSED* aplicados a esta industria sean capaces de reconocer tanto sistemas continuos como discretos. Por esta razón es imperativo incluir, además de los atributos que comúnmente caracterizan los eventos discretos, aspectos relacionados con la capacidad para operar de forma directa con procesos continuos.

La gran cantidad de rasgos a evaluar en los *SSED* sumado a la complejidad de los sistemas de logística de este tipo de industria, dirigió esta investigación a realizar modificaciones en MOSCA con el objetivo

de agrupar dichos rasgos de una forma coherente y fácil de manipular y entender.

Para tal efecto, se replanteó la manera de evaluar las características escogidas a través de un número de criterios, los cuales a su vez, se subdividen en subcriterios. Este planteamiento está avalado por Kitchenham (1996).

La referida reformulación, denominada MOSCA⁺, se diferencia del modelo original, en que incorpora un nivel adicional de subcriterios. Desde el punto de

vista conceptual, estos subcriterios corresponden a las métricas del modelo MOSCA, ya que ambos están relacionados con los atributos de calidad del software que se desean evaluar. De esta manera, el Nivel 3 está constituido por *criterios* y el Nivel 4, por *subcriterios*.

En la Figura 2 se presenta de forma detallada el diagrama del Modelo Sistemico de Calidad, denominado MOSCA⁺.

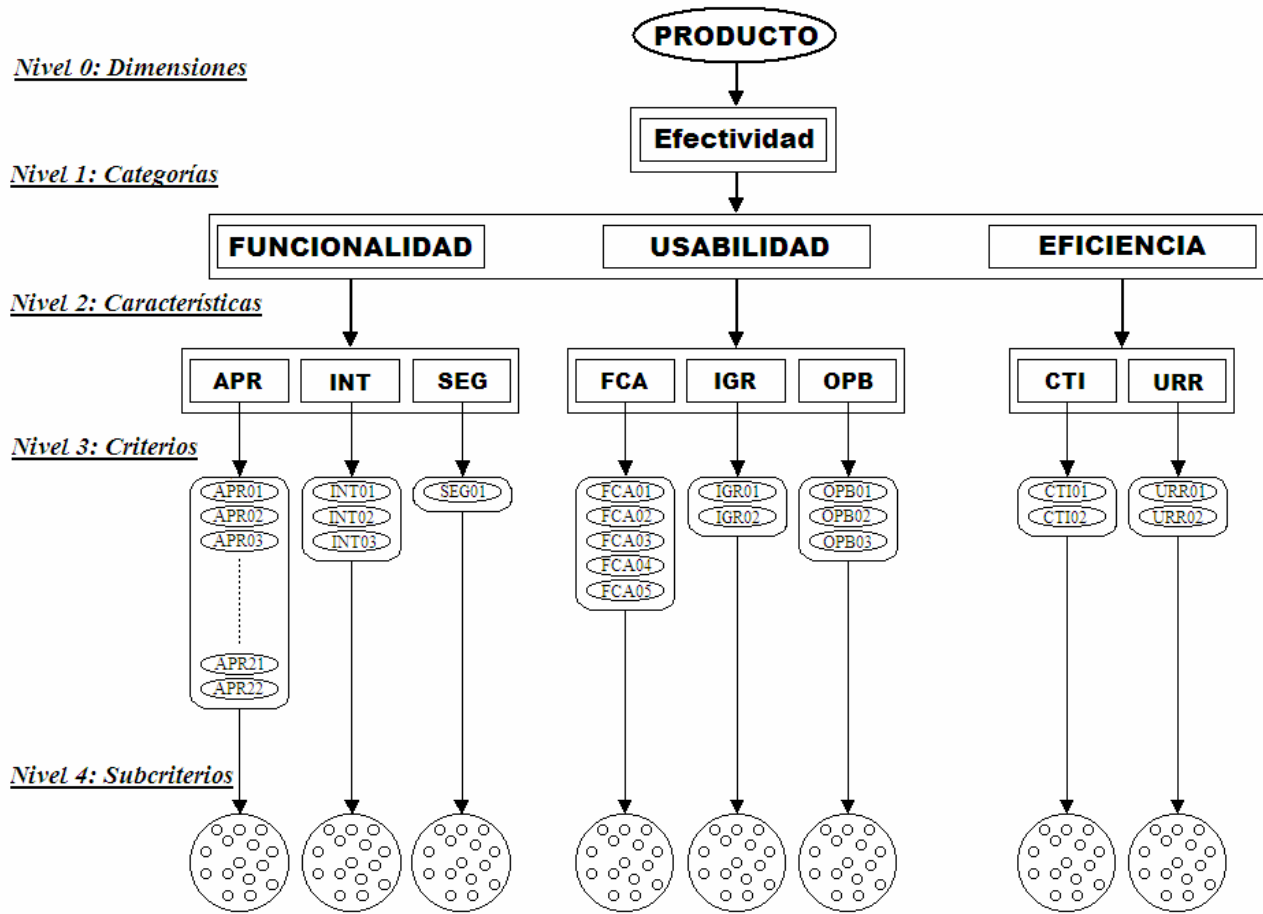


Figura 2: Diagrama del Modelo Sistemico de Calidad MOSCA⁺

IV. APLICACIÓN DE MOSCA+ - ESTUDIO DE UN CASO

La identificación de los rasgos del *SSED*, realizada en atención a los aportes de Nikoukaran, Hlupic y Paul (1998) y Hlupic (1999), y a la información recopilada a partir de literatura publicitaria, documentación técnica de las diferentes herramientas y experiencias

previas realizadas en la industria petrolera, permitió elaborar la propuesta de los criterios y subcriterios que evalúan *SSED*.

La Tabla 3 agrupa los criterios propuestos (Nivel 3), organizados de acuerdo con la característica que evalúan y la categoría a la cual pertenecen.

Tabla 3: Categorías, Características y Criterios propuestos por MOSCA⁺

<i>Categorías</i>	<i>Características</i>	<i>Código</i>	<i>Criterios</i>
FUNCIONALIDAD	Ajuste a los propósitos (APR)	APR 01	Datos de entrada
		APR 02	Clonación de elementos
		APR 03	Modularidad
		APR 04	Recursos Lógicos
		APR 05	Lenguajes de programación
		APR 06	Números aleatorios
		APR 07	Distribuciones de probabilidad
		APR 08	Reloj de simulación
		APR 09	Entidades
		APR 10	Colas
		APR 11	Operaciones
		APR 12	Fluidos
		APR 13	Tuberías
		APR 14	Tanques
		APR 15	Experimentación
		APR 16	Análisis estadístico
		APR 17	Análisis de costos
		APR 18	Grabación
		APR 19	Generación de reportes
		APR 20	Gráficos
		APR 21	Imágenes e íconos
		APR 22	Animación
INTEROPERABILIDAD (INT)	Interoperabilidad (INT)	INT 01	Sistema Operativo
		INT 02	Intercambio de datos
		INT 03	Uso de modelo por terceros
SEGURIDAD (SEG)	SEG 01	Medidas de protección	
USABILIDAD	Facilidad de comprensión y aprendizaje (FCA)	FCA 01	Tiempo de aprendizaje
		FCA 02	Facilidad de navegación
		FCA 03	Terminología
		FCA 04	Ayuda y documentación
		FCA 05	Soporte y entrenamiento
	Interfaz Gráfica (IG)	IGR01	Interfaz Windows y mouse
		IGR 02	Apariencia en pantalla
	Operabilidad (OPB)	OPB 01	Versatilidad de ejecución
		OPB 02	Interacción
OPB 03		Multitarea	
EFICIENCIA	Comportamiento en el tiempo (CTI)	CTI 01	Control de la velocidad de la animación
		CTI 02	Rapidez de compilación
	Utilización de recursos (URR)	URR 01	Requerimientos de hardware
		URR 02	Requerimientos de software

Los criterios de la categoría *FUNCIONALIDAD* se detallan a continuación organizados por características:

Característica: Ajuste a los propósitos (APR). Evalúa si el software es capaz de proveer funciones apropiadas según tareas específicas del usuario (Mendoza *et al*, 2002)

- **Criterio: APR01 Datos de entrada.** Facilidad para introducir los datos de entrada de un modelo. Los datos de entrada son los valores de todas las variables y factores que se requieren para definir un modelo, los cuales deben ser especificados por el usuario antes de ejecutar la simulación.
- **Criterio: APR02 Clonación de elementos.** Creación de elementos idénticos a partir de un elemento original (elementos, componentes básicos de los modelos de simulación).
- **Criterio: APR03 Modularidad.** Facilidad para desarrollar modelos de sistemas complejos a partir de varios subsistemas más simples, denominados módulos.
- **Criterio: APR04 Recursos Lógicos.** Facilidad para describir la operación del sistema representado en el modelo.
- **Criterio: APR05 Lenguajes de programación.** Capacidad de reconocer códigos de lenguajes de programación general, facilitando el acceso a funciones especiales a través del llamado a rutinas externas.
- **Criterio: APR06 Números aleatorios.** Facilidad que tiene el usuario para controlar la generación de números aleatorios. Se evalúa si se especifica la semilla, si se selecciona entre un conjunto de secuencias predefinidas o se si se selecciona entre números aleatorios regulares o antitéticos.
- **Criterio: APR07 Distribuciones de probabilidad.** Facilidad relacionada con distribuciones de probabilidad. Se evalúa si se selecciona el tipo de distribución o se ajusta a distribuciones empíricas.
- **Criterio: APR08 Reloj de simulación.** Relacionado con las diferentes formas en que se puede mostrar en pantalla el reloj de simulación (analógico o digital).
- **Criterio: APR09 Entidades.** Asociado a los objetos dinámicos en la simulación. Se evalúa si las entidades pueden ser finitas o infinitas, la forma en que estas ingresan y el tiempo en que ingresa la primera entidad.
- **Criterio: APR10 Colas.** Relacionado con el lugar donde se ubica una entidad cuando ésta no puede avanzar en la simulación. Se evalúa la cantidad y tipo de entidades que ingresan a la cola, así cómo éstas se organizan en la cola.
- **Criterio: APR11 Operaciones.** Facilidad para describir operaciones como prestación de servicio y planificación de averías o limpiezas de los equipos.
- **Criterio: APR12 Fluidos.** Relacionado con procesos continuos. Evalúa la capacidad para representar directamente fluidos en un modelo.

- **Criterio: APR13 Tuberías** Relacionado con procesos continuos. Facilidad para representar directamente tuberías en un modelo.
- **Criterio: APR14 Tanques.** Relacionado con procesos continuos. Facilidad para trabajar directamente con tanques de almacenamiento.
- **Criterio: APR15 Experimentación.** Facilidades para realizar experimentos de simulación. Se evalúa si se especifica tiempo de calentamiento, generación de replicas, análisis de sensibilidad y optimización del modelo.
- **Criterio: APR16 Análisis estadístico.** Capacidad para proporcionar medidas estadísticas sobre los parámetros y variables del modelo. Se evalúa si se calculan medidas estadísticas y si se realizan análisis estadístico de las diferentes replicas de simulación.
- **Criterio: APR17 Análisis de costos.** Relacionado al cálculo de los costos de operación que tiene el sistema representado en el modelo.
- **Criterio: APR18 Grabación.** Relacionado con el proceso de grabación del modelo de simulación.
- **Criterio: APR 19 Generación de reportes.** Capacidad para elaborar reportes automáticamente y reportes que se acoplen a las necesidades particulares del usuario.
- **Criterio: APR 20 Gráficos.** Facilidad para presentar los resultados de la simulación a través de diferentes tipos de gráficos.
- **Criterio: APR 21 Imágenes e Iconos.** Concerniente a la capacidad para crear, modificar, importar o almacenar imágenes e iconos.
- **Criterio: APR 22 Animación.** Relativo a la animación de un modelo de simulación.

Característica: Interoperabilidad (INT). Evalúa si el software es capaz de interactuar con uno o más sistemas (Mendoza *et al*, 2002)

- **Criterio: INT01 Sistema Operativo.** Referido al ambiente de operación que soporta el software.
- **Criterio: INT02 Intercambio de datos.** Relacionado con el intercambio de datos entre el software y otras aplicaciones, como: hojas de cálculo y archivos de texto.
- **Criterio: INT03 Uso de modelos por terceros.** Facilidades que brinda la empresa licenciante para que terceras personas puedan ejecutar o cambiar parámetros de modelos ya construidos, sin necesidad de adquirir la versión completa del paquete.

Característica: Seguridad (SEG). Evalúa si el software es capaz de proteger información, limitando el acceso a personas o sistemas autorizados (Mendoza *et al*, 2002).

- **Criterio: SEG01 Medidas de protección.** Relativo a la protección a los modelos mediante alguna clave de seguridad.

Los criterios de la categoría *USABILIDAD* se detallan a continuación organizados por características:

Característica: Facilidad de comprensión y aprendizaje (FCA). Evalúa la capacidad de facilitar al usuario el entendimiento del software y la forma en que puede ser utilizado y habilitado para el aprendizaje de la aplicación (Mendoza *et al*, 2002).

- **Criterio: FCA01 Tiempo de aprendizaje.** Tiempo requerido para que un nuevo usuario se familiarice con el uso del software y sea capaz de construir un modelo.
- **Criterio: FCA02 Facilidad de navegación.** Facilidad para la búsqueda de comandos y funciones.
- **Criterio: FCA03 Terminología.** Debido que no existe una terminología común en el área de simulación de eventos discretos y a que esto influye sobre el tiempo de aprendizaje (Nikoukaran *et al*, 1998), se propone un criterio que evalúe si los términos usados por el software son de fácil comprensión por parte del usuario.
- **Criterio: FCA04 Ayuda y documentación.** Facilidad para asistir al usuario en el aprendizaje y uso del software.
- **Criterio: FCA05 Soporte y entrenamiento.** Relativo al soporte y entrenamiento ofrecido por las empresas licenciantes.

Característica: Interfaz gráfica (IGR). Asociada a los atributos del software que lo hacen más atractivo al usuario, tales como color y diseño gráfico (Mendoza *et al*, 2002).

- **Criterio: IGR01 Interfaz Windows y mouse.** Evalúa si su manejo se puede realizar de la misma manera en que se hace cuando se opera en ambiente Windows.
- **Criterio: IGR02 Apariencia en pantalla.** Facilidad para presentar la distribución en pantalla, el color y la organización de los elementos del modelo.

Característica: Operabilidad (OPB). Evalúa si el software es capaz de habilitar al usuario a operarlo y controlarlo (Mendoza *et al*, 2002).

- **Criterio: OPB01 Versatilidad de ejecución.** Facilidad para controlar el proceso de ejecución de un modelo.
- **Criterio: OPB02 Interacción.** Relativo a la comunicación entre herramienta y usuario. Evalúan la

forma cómo se solicitan los datos, si una corrida se puede interrumpir

- **Criterio: OPB03 Multitarea.** Relacionado con la versatilidad del programa para la realización de operaciones simultáneas.

Los criterios de la categoría *EFICIENCIA* se detallan a continuación organizados por características:

Característica Comportamiento en el tiempo (CTI). Evalúa si el software es capaz de proveer respuestas y tiempos de procesamiento apropiados bajo condiciones específicas (Mendoza *et al*, 2002).

- **Criterio: CTI01 Control de la velocidad de la animación.** Relativo al control de la velocidad de animación en la simulación, cuando ésta se encuentra activada.
- **Criterio: CTI02 Rapidez de compilación.** Referido a la rapidez con que el software crea una versión ejecutable del modelo.

Característica Utilización de recursos (URR). Evalúa si el software utiliza cantidades apropiadas de los recursos cuando ejecuta sus funciones bajo condiciones específicas (Mendoza *et al*, 2002).

- **Criterio: URR01 Requerimientos de hardware.** Relativo a la posibilidad de instalación y ejecución del software en las computadoras de la organización.
- **Criterio: URR02 Requerimientos de software.** Evalúa la necesidad de instalar un software adicional al *SSED* para que éste pueda operar.

Los subcriterios que miden la calidad del software se operacionalizan a través de preguntas que permiten pulsar la calidad de los criterios, así como garantizar la objetividad y precisión de la evaluación. De este planteamiento resultaron 131 subcriterios que estiman la calidad de los 40 criterios identificados, donde cada uno tiene asociado entre 1 y 8 métricas, que calibran la calidad del *SSED*. Estos 131 subcriterios están distribuidos de la siguiente manera: 86 para los criterios de la categoría *FUNCIONALIDAD*, 39 para los criterios que conforman la categoría *USABILIDAD* y 6 para la categoría *EFICIENCIA*.

La definición de obligatoriedad (mandatoriedad) de algunos subcriterios y el grado de importancia asignado a los subcriterios deseables (no obligatorios), se determinó mediante una encuesta aplicada a expertos vinculados al área de interés en la industria petrolera, docencia y consultoras.

La Figura 3 presenta los resultados obtenidos de la aplicación del modelo propuesto (MOSCA⁺) para la

evaluación de *SSED* en una organización que presta servicios de consultoría, para apoyar la toma de decisiones en el área de logística de la industria petrolera. Los cuatro *SSED* que se muestran en esta Figura fueron preseleccionados de una lista larga de *SSED* disponibles en el mercado, de acuerdo a los objetivos establecidos por este tipo de organización, los cuales especificaban el *área de aplicación y uso*, utilización de *paquetes* y no lenguajes de simulación, capacidad de modelar *sistemas híbridos* y cumplimiento de los criterios mandatorios establecidos para este caso de estudio a través de la encuesta aplicada a los expertos.

La *Tasa de Calidad* resultante fue calculada mediante la división de la puntuación total obtenida por el *SSED* para cada categoría, entre la puntuación total máxima que pudiera haber alcanzado el *SSED* en dicha categoría.

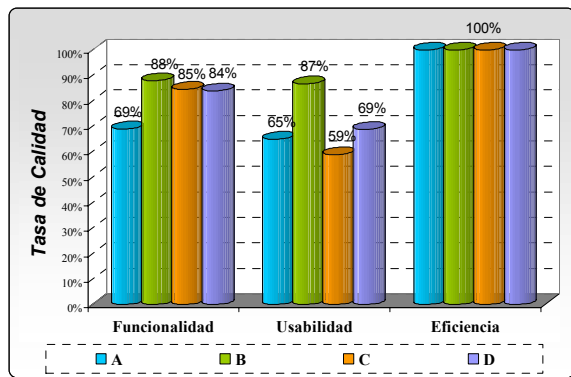


Figura 3: Tasas de Calidad por categorías

Los resultados reflejan que el modelo aplicado permite diferenciar claramente las fortalezas y debilidades de estos *SSED*, apoyando efectivamente la toma de decisiones con base en las necesidades particulares de esta organización

V. CONCLUSIONES

El modelo MOSCA⁺ identificó 40 criterios y 131 subcriterios que el proceso de evaluación de *SSED*. Tal conjunto de criterios y subcriterios permitirá soportar el proceso de selección de este tipo de software aplicado a la industria petrolera y proporcionar objetividad a la toma de decisión asociada a dicha selección.

VI. AGRADECIMIENTO

Esta investigación fue financiada por el Decanato de Investigación de la Universidad Simón Bolívar a

través del proyecto *Herramientas de Soporte en Logística para la Industria Petrolera* (DI-CAI-004-02).

VII. REFERENCIAS BIBLIOGRÁFICAS

- BANKS, J., CARSON, J. NELSON, B. (1996). *Discrete-Event System Simulation*. Prentice-Hall, New Jersey
- FALCONER, D. GUY, B. (1998). *Trully Optimal Offsites*. The Chemical Engineer, Marzo, pp. 28-33.
- GOTTFRIED, B. (1984). *Elements of Stochastic Process Simulation*. Prentice Hall, New Jersey.
- HLUPIC, V. (1999). *Discrete-Event Simulation Software: What the Users Want*. SIMULATION, 73:6, 362-370
- KELTON, W., SADOWSKI, R., SADOWSKI, D. (1998). *Simulation in Arena*. McGraw Hill, Boston
- KITCHENHAM, B. (1996). *DESMET: A method for Evaluating Software Engineering methods and tools*. Technical report TR 96-09. Department of Computer Science, University of Keele, Inglaterra. <http://www.keele.ac.uk/depts/cs/se/e&m/techreps.htm>.
- LAW, A., KELTON, D. (1991). *Simulation Modeling and Analysis*. McGraw-Hill, New York
- MENDOZA, L., PÉREZ, M. GRIMÁN, A., ROJAS, T. (2002). *Algoritmo para la Evaluación de la Calidad Sistemica del Software*. Anales de las 2das. Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2002). Salvador, Brasil. Noviembre de 2002. pp. 1-11.
- NAYLOR, T., BALINTFY, D., BURDICK, D. y CHU, K. (1966). *Computer Simulation Techniques*. Wiley. New York.
- NIKOUKARAN, J., HLUPIC, V. PAUL, R. (1998). *Criteria for simulation software evaluation*, en Proceedings of the 1999 Winter Simulation Conference, eds. D. Medeiros, E. Watson, J. Carson y M. Manivannan, pp. 399-406, Department of Information Systems and Computing of Brunel University, Reino Unido
- ROJAS, T., MENDOZA, L. PÉREZ, M. (2001). *Indicadores Organizacionales para comparación de herramientas CASE en Venezuela*. Revista de la Facultad de Ingeniería de la U.C.V., Vol. 16, N° 1, pp 95-112
- ROJAS, T., PÉREZ, M., GRIMÁN, A., ORTEGA, M. Y DIAZ, A. (2000). *Modelo de decisión para soportar la selección de herramientas CASE*. Revista de la Facultad de Ingeniería de la U.C.V., Vol. 15, N° 2, pp 117-144
- SHANNON, R. (1975). *Systems Simulation: The Art and Science*. Prentice Hall, New Jersey
- UMEDA, S. Y JONES, A. (1997). *Simulation in Japan:*

State-of-the-art update. Technical report, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Technology Administration.