

Modelo de Calidad para la Especificación de un Software de Simulación de Satélites

Quality Model for the Specification of a Satellite Simulation Software

Solangel García, *Investigador, ABAE.*

Resumen—El presente artículo propone una herramienta para la estimación de la calidad de un software de simulación de Satélites, usando como referencia el Software de Simulación del Satélite Sucre, este modelo permitirá aceptar o rechazar el software durante las fases de implementación y prueba, ya que evalúa si los requerimientos generados y exigidos durante la fase de diseño se encuentran en el software en funcionamiento. Este instrumento de medición permite validar si el software presenta características de calidad, no solo a nivel de las funcionalidades del sistema sino también en términos de tiempo de respuestas, precisión y exactitud de la data y la información. El modelo de calidad que se presenta está definido siguiendo un Modelo Sistémico de Calidad (MOSCA), adaptado al área espacial y a un simulador de satélite. La presencia de las características de calidad representará una mejora significativa en el desempeño de las operaciones en órbita del futuro satélite, especialmente en el entrenamiento de actuales y futuros operadores satelitales, ingenieros de diseño e investigadores.

Palabras Claves – modelo de calidad, métrica, simulación.

Abstract—This paper proposes a tool for the estimation of the quality of a satellite simulation software, using as reference the Sucre Satellite Simulation Software, this model will allow to accept or reject the software during the implementation and testing phases, as it evaluates If the requirements generated and required during the design phase are in the running software. This measuring instrument allows validate if the software presents quality characteristics, not only in terms of the functionalities of the system but also in terms of response time and accuracy of data and information. The quality model presented is defined according to a Systemic Quality Model (MOSCA), adapted to the space area and to a satellite simulator. The presence of quality characteristics will represent a significant improvement in the performance of the future satellite's orbiting operations, especially in the training of current and future satellite operators, design engineers and researchers

Este artículo fue enviado al II Congreso Venezolano de Tecnología Espacial en el mes de agosto de 2017.

S. García pertenece a la Unidad de Desarrollo e Innovación Tecnológica. Dirección de Investigación e Innovación. Estación Terrena de Control Satelital (ETCS). Agencia Bolivariana para Actividades Espaciales (ABAE). El Sombrero, Venezuela. (e-mail: sgarcia@abae.gob.ve).

Index Terms – quality of model, metrics, simulation.

I. INTRODUCCIÓN

La Agencia Bolivariana para Actividades Espaciales (ABAE) es un organismo autónomo adscrito el Ministerio de Ciencia, Tecnología y Educación Universitaria, creado en enero de 2008 con la finalidad de regir todo lo relativo al desarrollo de políticas espaciales y uso pacífico del espacio ultraterrestre [1]. A pesar de ser una institución nueva, actualmente lleva varios proyectos en materia espacial en ejecución y cuenta con dos satélites en órbita, el Satélite Simón Bolívar que ofrece servicios de Telecomunicaciones y el Satélite Miranda que ofrece servicio de imágenes para la observación de la tierra, se está diseñando y ensamblando el tercer satélite conocido como el Satélite Sucre, también de observación de la Tierra, en un convenio bilateral entre en la República Popular China y la República Bolivariana de Venezuela. Los dos primeros satélites tienen asociado un software de simulación de operaciones satelitales, para garantizar el éxito de las mismas, además de proporcionar bondades de entrenamiento a los nuevos operadores satelitales venezolanos. Por otro lado, ya finalizado el desarrollo del software de simulación del Satélite Sucre, permitiendo una revisión exhaustiva de funcionamiento y desempeño antes de su aceptación, puesto que ahora se cuenta con más experiencia en el manejo y gestión de simuladores de satélites. En este sentido, se crean las siguientes interrogantes que se convierten en el punto de partida de este trabajo de investigación: ¿Cómo se puede validar y verificar un software de Simulación de Satélites y su funcionamiento?, ¿Exactamente que parámetros debemos medir para garantizar un software de calidad?

El objetivo de esta investigación es proponer un modelo de calidad para medir durante la implementación y la corrida del simulador, si cumple con características de calidad tanto de requerimientos funcionales como no funcionales.

Este artículo contiene 3 secciones: en la primera sección se explica concepto de calidad, calidad del software y un resumen del estado del arte de los modelos de calidad desde sus inicios hasta los más actuales. En la segunda se explica brevemente el modelo de calidad usado como referencia a la propuesta, y en la tercera se propone el modelo de calidad

adaptado al software de simulación del Satélite Sucre y explica la forma de implementarlo. Este artículo cierra con sus respectivas conclusiones y recomendaciones.

II. CALIDAD, CALIDAD DEL SOFTWARE Y MODELOS DE CALIDAD

A. Calidad

Desde décadas pasadas, existen diversas definiciones de calidad; grandes maestros de la calidad industrial como W.E Deming propusieron la idea de calidad como conformidad a requisitos y confiabilidad en el funcionamiento (1900). J. Juran propone: “Calidad es adecuación (del producto) al uso” Esta definición incluye las características del producto que permiten obtener la satisfacción del usuario y, además, supone la ausencia de deficiencias (1904). P. Crosby asume el concepto de Juran, pero destaca también prevención: “Cero defectos” o “Hacerlo bien a la primera” (1900) [2]. Para conceptos más recientes tenemos el de Wikipedia (Calidad, s.f.), la definen como [3]:

(...) conformidad relativa con las especificaciones, a lo que al grado en que un producto cumple las especificaciones del diseño, entre otras cosas, mayor su calidad o también como comúnmente es encontrar la satisfacción en un producto cumpliendo todas las expectativas que busca algún cliente, siendo así controlado por reglas las cuales deben salir al mercado para ser inspeccionado y tenga los requerimientos estipulados por las organizaciones que hacen certificar algún producto.

Hoy en día, la calidad se puede considerar como un concepto multidimensional (referida a muchas cualidades), sujeta a restricciones (por ejemplo, al presupuesto disponible) y ligada a compromisos aceptables (por ejemplo, a plazos de fabricación), incluso se puede considerar que no es totalmente subjetiva (porque ciertos aspectos pueden medirse) ni totalmente objetiva (ya que existen cualidades cuya evaluación solo puede ser subjetiva); así pues, la calidad no es absoluta, es multidimensional. Además, la calidad suele ser transparente cuando está presente, pero es fácilmente reconocible cuando está ausente (por ejemplo, si el producto falla). [4]

B. Calidad del Software

Ahora, el concepto de calidad aplicado al software es justo la labor de la Ingeniería del Software, que consiste en establecer las características del producto, la aplicación de métodos, herramientas, revisiones técnicas efectivas y una fuerte gestión y medición, que luego se confirma en las pruebas, la calidad de software se define como [5]:

(..) 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 (p.132).

C. Modelos de Calidad

Para evaluar la calidad de software existen algunos estándares y modelos que proponen una serie de características y sub-características presentes en los productos,

las cuales permiten determinar de manera precisa el nivel de calidad del software en cada una de ellas.

El análisis a estas características es importante porque siempre van a estar presentes en los productos de software; definir y especificar cada una de ellas depende de las exigencias creadas al momento de desarrollar el producto. Los modelos de calidad se presentan, orientados al producto y orientados al proceso. Sin embargo, para efectos de esta investigación, se nombran solo los modelos orientados al producto, que existen desde décadas atrás, los cuales son:

Modelo de Calidad de McCall [6]: describen una serie de factores de calidad en base a criterios y métricas de calidad. Abarca tres áreas de trabajo (Operación del producto, Revisión del Producto y Transición del Producto)

Modelo de Calidad de Boehm [7]: “este es uno de los mejores modelos de la calidad del software, ya que toma en cuenta las necesidades de los usuarios”.

FURPS [5]: la Hewlett Packard desarrolló un conjunto de factores de Calidad del Software identificado con el acrónimo FURPS: funcionalidad, facilidad de uso, confiabilidad, performance y facilidad de soporte. Además, plantea dos categorías de requerimientos, las cuales son: Requerimientos funcionales (F) en donde se especifican funciones que el sistema debe ser capaz de realizar y Requerimientos no funcionales (URPS). Usability (facilidad de uso), reliability (confiabilidad), performance y supportability (facilidad de soporte), que describen atributos del sistema.

Norma ISO-9126-3 [8]: estándar relacionado con la calidad interna del producto de software. Se describen métricas que miden los atributos internos del software relacionados al diseño y a la codificación. Las características en el estándar ISO/IEC 9126-3 son: funcionabilidad (conveniencia, precisión, interoperabilidad, conformidad, seguridad), Fiabilidad (madurez, tolerancia a fallas, recuperabilidad), usabilidad (entendibilidad, facilidad para entender, operabilidad), eficiencia (comportamiento en los tiempos, comportamiento en los recursos), mantenibilidad (capacidad de analizar, capacidad para modificar), portabilidad (adaptabilidad, facilidad para instalación, conveniencia, capacidad de ser reemplazado).

Modelo Sistemico de Calidad (MOSCA) [9]: este modelo desarrollado por el Laboratorio de Investigación de Sistemas de Información (LISI), en la Universidad Simón Bolívar “tiene como finalidad especificar la calidad de los sistemas de software. Integra un modelo de calidad del producto y el modelo de calidad del proceso de desarrollo, y además está soportado por los conceptos de la Calidad Total Sistemica”. También cuenta con un algoritmo para su aplicación y un total de 679 métricas para medir la calidad sistémica del producto del software.

Cada uno de estos modelos representa un posible esquema de trabajo para estimar la calidad de un producto de software, por esta razón, el objetivo principal de esta investigación es proponer un modelo instanciado en características de calidad, en donde la más importante a evaluar sea la Funcionalidad, ya que, permite determinar si el software cumplirá con las necesidades para el cual fue adquirido. Este modelo

referencial a utilizar puede ser MOSCA, ya que, integra todas las categorías de FURPS y la Norma ISO-9126, además cuenta con un algoritmo de ejecución que facilita la aplicación del modelo, por tal motivo, se descarta el Modelo de McCall y Boehm, por ser los más antiguos y con poca información disponible para su aplicación

III. MODELO SISTÉMICO DE CALIDAD DE SOFTWARE (MOSCA)

MOSCA consta de cuatro niveles: dimensiones, categorías, características y métricas. Además, proporciona el algoritmo para evaluar la calidad del producto, también aplica para el proceso, pero no se tomará en cuenta en esta investigación. El algoritmo es un conjunto de pasos procedimentales que se realizan para ejecutar el modelo y estimar la calidad de software [10]. Los niveles de MOSCA son los siguientes [11]:

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

Nivel 1: Categorías. Se contemplan seis (6) categorías pertenecientes al producto:

Funcionalidad (FUN), Fiabilidad (FIA), Usabilidad (USA), Eficiencia (EFI), Mantenibilidad (MAB) y Portabilidad (POR).

Nivel 2: Características. Cada categoría tiene asociado un conjunto de características (56 asociadas al producto, las cuales definen las aéreas claves a satisfacer para lograr, asegurar y controlar la calidad del producto).

Nivel 3: Métricas. Para cada característica propone una serie de métricas utilizadas para medir la calidad sistémica. Se pauta un total de 587 métricas.

Para la aplicación de MOSCA existe el siguiente algoritmo [12]:

1. Estimación de la calidad del producto. Inicialmente, se debe medir la categoría de funcionalidad del producto. Si esta cumple con el 75% de las características necesarias propuestas para esta categoría, se prosigue con el siguiente paso.

2. Instanciación del sub-modelo del producto. De las cinco categorías restantes, se seleccionan dos. El algoritmo recomienda trabajar con un máximo de tres categorías, pues si se seleccionan más, podrían presentar conflictos.

3. Estimación de la calidad para cada categoría. Para las dos categorías seleccionadas en el paso anterior se debe: (a) aplicar las métricas propuestas en el sub-modelo del producto para las categorías seleccionadas; (b) verificar que el 75% de las métricas están dentro de los valores óptimos (mayor o igual a tres) para cada una de sus características; y (c) evaluar la categoría. Para que una categoría sea satisfecha, al menos el 75% de sus características deben ser altamente satisfechas. Esto garantiza coherencia y consistencia con los niveles de aceptación establecidos por el modelo.

4. Estimación de la calidad del producto partiendo de las categorías evaluadas. Si no se satisface la categoría funcionalidad, el algoritmo finaliza y la calidad del producto de software será nula. Si un producto cumple con los objetivos para los cuales se creó (funcionalidad), tendrá una calidad básica. Si satisface solo una de las categorías seleccionadas, además de la funcionalidad, tendrá un nivel de calidad

intermedio; si satisface todas las categorías seleccionadas, tendrá un nivel avanzado.

IV. PROPUESTA DEL MODELO DE CALIDAD PARA EL SOFTWARE DE SIMULACIÓN DEL SATÉLITE SUCRE (VRSS-2)

Según el algoritmo de MOSCA, se debe seleccionar 3 categorías del producto, estando siempre presente la más importante como la funcionalidad, las dos categorías restantes para el caso del simulador será la usabilidad, puesto que el simulador debe ser fácil de aprender y utilizar por los operadores y durante los entrenamientos y la fiabilidad motivado a que los resultados ofrecidos deben ser exactos y precisos. Dado que el alcance de este artículo está orientado a un producto en específico. Para cada una de estas categorías se seleccionó un subconjunto de características y para estas, un subconjunto de métricas; que en algunos casos fue necesario añadir nuevas métricas. A continuación, se expone en:

Según [13], la categoría **FUNCIONALIDAD (FUN)**, es la habilidad del sistema para realizar las tareas que se le han encomendado, está relacionada con una buena distribución de la carga de trabajo entre los componentes del sistema y con la buena coordinación de los mismos. La medición de la funcionalidad en el simulador nos permitirá validar la especificación de sus requerimientos de sistema y los requerimientos funcionales. Para esta categoría se escogieron las características y sub-características tales como:

1.- Características: Ajuste a los propósitos (APR) Evalúa si el software es capaz de proveer funciones apropiadas según las tareas específicas del usuario [11]. Se definieron las siguientes sub-características:

Sub-característica Procesamiento de Tele-comandos (APR01): Capacidad del simulador de procesar los Tele-comandos del satélite real.

- Métrica 1: ¿El simulador presenta limitaciones para enviar un tele-comando (TC) como en el satélite real?
- Métrica 2: ¿El simulador procesa la cantidad exacta de TC igual que el satélite real?
- Métrica 3: ¿Los TC son consistente con los del satélite real?
- Métrica 4: ¿Los TC tardan 10 segundos o menos en ejecutarse?
- Métrica 5: ¿Los TC son validados antes de ejecutarse?
- Métrica 6: ¿El simulador especifica un mensaje de error cuando un TC no se ejecuta?
- Métrica 7: ¿Se muestra una lista de TC pre-establecida con diferenciación de TC directos e indirectos?

Sub-característica Despliegue de Telemetría (APR02): Capacidad del simulador de presentar los parámetros de telemetría de acuerdo al satélite real.

- Métrica 8: ¿La telemetría (TM) está estructurada como en el Software de Despliegue de Telemetría (GIDS) del Satélite Real?
- Métrica 9: ¿El formato de la TM es acorde al satélite real?

- Métrica 10: ¿La TM es mostrada en todos los formatos como el satélite real?
- Métrica 11: ¿La TM muestra la cadena de datos?
- Métrica 12: ¿La TM se presenta en forma ordenada?
- Métrica 13: ¿Se refleja el cambio de TM cuando se ejecuta un TC?
- Métrica 14: ¿Se genera la trama de TM basada en la palabra de datos?
- Métrica 15: ¿Capacidad para realizar gráficos (Curve) de la TM?

Sub-característica Control Orbital (APR03): Evalúa si el simulador provee todas las facilidades para el seguimiento del satélite en su trayectoria orbital a través de las dos herramientas complementarias.

- Métrica 16: ¿Presencia del software de propagación orbital?
- Métrica 17: ¿Propagación de la órbita de acuerdo a los parámetros orbitales iniciales?
- Métrica 18: ¿La diferencia de la variación de los elementos orbitales con respecto al satélite real es menor del 1% de precisión?
- Métrica 19: ¿El Software de propagación incluye los efectos de las fuerzas perturbadoras (Tierra, Sol, Luna)?
- Métrica 20: ¿Facilidades de control de la propagación de la órbita hacia delante o hacia atrás?
- Métrica 21: ¿Las efemérides trabaja con orbitas bajas?
- Métrica 22: ¿Existe suficiente información sobre la posición de los Satélites a partir de condiciones iniciales de la órbita?
- Métrica 23: ¿El simulador muestra el conjunto de datos que procesa la Estación Terrena de Control para soportar el seguimiento de la antena?
- Métrica 24: ¿Capacidad de procesamiento de otros tipos de orbitas bajas?
- Métricas 25: ¿Procesamiento de datos manual o automática?
- Métricas 26: ¿Presencia de plan de mantenimiento orbital?
- Métrica 27: ¿Capacidad de planificación de maniobras?
- Métrica 28: ¿Realiza cálculos de ajustes de impulso para los tres ejes del satélite?
- Métrica 29: ¿Es posible almacenar los parámetros calculados en una base de datos?
- Métrica 30: ¿Los resultados obtenidos (eventos) almacenados contiene el vector velocidad y el cambio de actitud?
- Métrica 31: ¿Es posible verificar si la maniobra calculada satisface los objetivos de la misión?
- Métrica 32: ¿Es posible predecir la trayectoria del Satélite?
- Métrica 33: ¿La propiedad de la masa inicial y final son parte de los resultados del cálculo?
- Métrica 34: ¿El simulador muestra el cálculo del

combustible?

- Métrica 35: ¿Es posible actualizar la base de datos los resultados del cálculo de combustible una vez finalizada la simulación de la maniobra?
- Métrica 36: ¿Es posible actualizar la base de datos del simulador con parámetros de una maniobra real?

Sub-característica Simulación en Modo de tiempo real. APR05: Facilidades del software para simular con tiempos aproximados al real.

- Métrica 37: ¿Es posible cargar un escenario en tiempo real?
- Métrica 38: ¿Es posible guardar y desplegar datos de telemetría en tiempo real?
- Métrica 39: ¿Es posible importar un TC en tiempo real?
- Métrica 40: ¿Es posible inyectar una falla al modelo de simulación en tiempo real
- Métrica 41: ¿Es posible suspender una Simulación?
- Métrica 42: ¿Es posible parar una simulación?
- Métrica 43: ¿Acelerar una simulación es posible en tiempo real, es decir, se puede ir a un tiempo especificado?

Sub-característica Simulación en Modo aceleración APR05: Simulación en Modo aceleración. Facilidades del software para simular con capacidades de aceleración.

- Métrica 44: ¿Es posible que el operador pueda acelerar 2 veces el tiempo actual durante la simulación en este modo?
- Métrica 45: ¿Es posible que el operador acelerar 4 veces el tiempo actual durante la simulación en este modo?
- Métrica 46: ¿Capacidad de modificación de velocidad?
- Métrica 47: ¿Es posible importar fallas y TC en este modo?
- Métrica 48: ¿Fallas y TC con etiquetas de tiempo son soportados en este modo?
- Métrica 49: ¿Es posible usar este modo con el modo de reproducción?
- Sub-característica: APR05 Simulación en Modo de reproducción. Facilidades del software para simular con capacidades de reproducción accediendo a la data almacenada directamente.
- Métrica 50: ¿Permite el cambio de velocidad o tiempo de reproducción?
- Métrica 51: ¿Es posible bajar la reproducción de la simulación?

Sub-característica Subsistemas APR04: Subsistemas Capacidad de realizar operaciones de los diferentes subsistemas del satélite.

- Métrica 52: ¿Realiza operaciones del Subsistema de Manejo de Datos a Bordo (OBDH)?
- Métrica 53: ¿Realiza operaciones en modo seguro?
- Métrica 54: ¿Realiza operaciones de modo de

emergencia?

- Métrica 54: ¿Realiza operaciones del Subsistema de Potencia Eléctrica (PSS)?
- Métrica 55: ¿Realiza operaciones del Subsistema Control de Actitud y Orbita (AOCS)?
- Métrica 56: ¿Realiza operaciones del Control Térmico?
- Métrica 57: ¿Realiza operaciones del Subsistema de Transmisión de Datos (DTS)?
- Métrica 58: ¿Realiza operaciones del Subsistema del Control Térmico (TCS)?

2.- **Característica Precisión (PRE):** Es la capacidad del producto de software para proveer los resultados correctos [12].

Sub-característica Resultados (PRE01): Es la capacidad de medir la fidelidad de los resultados de la simulación con respecto a los datos reales.

- Métrica 59: ¿Grado de aproximación de los resultados del modelo de propagación con respecto a otro software comercial del mismo fin?
- Métrica 60: ¿Nivel de exactitud de los elementos orbitales al propagar una órbita?
- Métrica 61: ¿Al ejecutar un procedimiento se muestra la TM acorde al satélite real?
- Métrica 62: ¿Provee la capacidad de generar reportes y log de la simulación?

3.- **Característica Seguridad (SEG):** La seguridad, según [13], mide la capacidad del sistema para proteger datos e información de accesos no autorizados mientras les proporciona acceso a las personas y al sistema que sí están autorizadas. (p. 147).

Sub-característica Integridad de los Datos o mensajes SEG01: Capacidad del simulador de verificar la integridad de los mensajes, datos y la configuración de archivo.

- Métrica 63: ¿El simulador permite verificar si los mensajes, los datos y los archivos están completos o no son corruptos?

Sub-característica Autenticación de usuario SEG03: consiste en asegurar que un actor (un usuario o un computador remoto) es quien dice ser; mediante claves de accesos, claves de tiempo, certificaciones digitales, identificación biométrica que proporcionan la autenticación.

- Métrica 64: ¿El simulador permite la identificación del operador al sistema mediante un nombre (usuario) establecido previamente?
- Métrica 65: ¿El simulador permite la identificación del administrador del sistema?
- Métrica 66: ¿El simulador permite el uso de claves de acceso o alguna otra herramienta de seguridad?

La categoría **USABILIDAD (USA):** “Es la facilidad con la que un usuario puede manejar y acceder al sistema de información con garantías de éxito. En realidad, esta categoría permite estudiar el grado de elegancia y claridad con que se ha diseñado el sistema, para facilitar la interacción con el mismo” [14]. En el simulador de satélite abarca los requerimientos de interfaz gráfica, despliegue de telemetría y procesamiento de comandos entre otros que especifican la interacción y

visualización directa con la interfaz del simulador, se debe chequear si es de forma amigable, fácil de entender y además que propicie el aprendizaje.

4.- **Característica Interfaz Gráfica (ING):** Está asociada a los atributos del sistema que lo hacen más atractivo al usuario, en el caso del simulador, evalúa si el tratamiento de la telemetría y los Telecomando es igual al satélite real, operaciones vitales para el éxito de la simulación.

Sub-característica Datos de entrada (ING01): Facilidad para introducir los datos de entrada, estos deben ser especificados por el operador antes de ejecutar la simulación.

- Métrica 67: ¿Nivel de complejidad para introducir parámetros iniciales en la simulación?
- Métrica 68: ¿La visualización de la TM se presenta como en el satélite real?
- Métrica 69: ¿La visualización de los TC se presenta como en el satélite real?
- Métrica 70: ¿La interfaz gráfica del usuario (GUI) es amigable?

Sub-característica Manejo de escenarios (APR5): Facilidades para realizar experimentos de simulación (normal, emergencia, tiempo real, aceleración, reproducción y en cualquier subsistema).

- Métrica 71: ¿Es posible crear nuevos escenarios de simulación?
- Métrica 72: ¿Existe restricciones para realizar una simulación?
- Métrica 73: ¿Es posible operar el simulador sin ningún entrenamiento previo en el software?
- Métrica 74: ¿El simulador presenta limitaciones a nivel de operación específicamente para realizar un procedimiento?
- Métrica 75: ¿Facilidad de controlar la simulación?
- Métrica 76: ¿Es posible mostrar el estatus de una simulación?

5.- **Característica Facilidad de Comprensión (FDC):** Evalúa la capacidad de facilitar al usuario el entendimiento del software [7]. Es importante que el simulador pueda comprenderse rápidamente.

Sub-característica Dificultad de uso del Simulador (FDC01): Evalúa el grado de complejidad del uso del simulador de acuerdo al nivel de diferencia en la operación con el satélite real.

- Métrica 77: ¿Nivel de complejidad del software?
- Métrica 78: ¿Es indispensable el uso de ayudas?
- Métrica 79: ¿Es indispensable el manual de simulación?

Sub-característica Facilidad de Navegación (FDC02): Evalúa la facilidad para ubicar las funcionalidades del simulador.

- Métrica 80: ¿Las barras de navegación son entendibles?
- Métrica 81: ¿La barra de navegación está completa?
- Métrica 82: La barra de navegación es consistente con el simulador?
- Métrica 83: ¿Están definidas claramente las

funciones del simulador?

- Métrica 84: ¿La ayuda están fácilmente ubicables?

La categoría **EFICIENCIA (EFI)**: Capacidad del producto de software para proveer un rendimiento apropiado relativo a la cantidad de recursos utilizados, bajo condiciones específicas [9].

6.- Característica Comportamiento del Tiempo (CDT):

Evalúa si el software es capaz de proveer las respuestas y tiempos de procesamiento apropiados bajo condiciones específicas [10].

Sub-característica Control de la velocidad de la simulación CDT01: Mide los niveles de control de velocidad o aceleración de la simulación

- Métrica 85: ¿Es posible controlar la velocidad de la simulación?
- Métrica 86: ¿Es posible iniciar, pausar y finalizar una simulación?
- Sub-característica: CDT02 Rapidez de Ejecución. Mide el tiempo que tarda en procesar una simulación.
- Métrica 87: ¿Los tiempos de respuestas en el procesamiento son iguales al satélite real?
- Métrica 88: ¿La capacidad máxima de duración de una simulación es igual o menor a 72 horas?
- Métrica 89: ¿Ante un reinicio o falla en el software el sistema está disponible en un máximo de 10 minutos?
- Métrica 90: ¿La sincronización del tiempo es menor a 30 segundos?
- Métrica 91: ¿Los mensajes de alertas aparecen en menos de 5 segundos?

7.- Característica Utilización de Recursos (UDR):

Evalúa si el software utiliza cantidades apropiadas de recursos cuando ejecuta sus funciones bajo condiciones específicas [10].

Sub-característica Requerimientos de Hardware (UDR01): Evalúa la posibilidad de instalación de hardware adicional para la mejora del desempeño.

- Métrica 92: ¿El software se limita por la capacidad de la memoria RAM?
- Métrica 93: ¿El cambio de algún componente mejora los tiempos de respuestas?
- Métrica 94: ¿Los controles de velocidad dependen del hardware instalado?

Sub-característica Requerimientos de Software (UDR): Evalúa la necesidad de instalar un software adicional para que el simulador pueda operar correctamente.

- Métrica 95: ¿Se requiere de algún software adicional para mejorar el rendimiento del simulador?

Una vez seleccionadas las categorías, características y sub-características y las métricas para medir la calidad del software de acuerdo al modelo de MOSCA, estas métricas se especifican a través de preguntas que deben ser completadas una vez que el software demo de simulación este operativo y

en funcionamiento. A continuación, se muestra un a propuesta de posibles respuestas para cada métrica.

- Presencial: Esta métrica indica si un atributo está presente en el software. Puede ser medido: sí = (4) y no = (0).

- Nivel: Se utiliza para indicar un grado de esfuerzo, habilidad, etc., cuando se da una medida subjetiva dentro de una escala de valores. Es una variable que puede tomar el siguiente rango de valores:

1.- 0 (muy bajo), 1 (bajo), 2 (medio), 3 (alto), 4 (muy alto).

2.- 0 (no está definido), 1 (poco definido), 2 (medianamente definido), 3 (casi todo definido), 4 (completamente definido).

3.- 0 (nunca), 1 (pocas veces), 2 (algunas veces), 3 (casi siempre), 4 (siempre).

Finalmente se debe verificar que el 75% de las métricas se encuentren igual o mayor a 3, para cada uno de las características y sub-características, si no se cumple la calidad será nula, si cumple la siguiente tabla muestra el nivel de calidad alcanzado por el software.

TABLA I
Nivel de calidad del producto [9]

Funcionalidad	Usabilidad	Eficiencia	Nivel de Calidad
Satisfecha	No satisfecha	No satisfecha	Básico
Satisfecha	No satisfecha	Satisfecha	Básico
Satisfecha	Satisfecha	No satisfecha	Intermedio
Satisfecha	Satisfecha	Satisfecha	Avanzado

En esta tabla se representa el posible nivel de calidad alcanzado una vez evaluadas las características de calidad.

El instrumento de medición propuesto consta de cuatro niveles que se resumen a continuación:

1.- Nivel 0: Dimensiones. Producto (Simulador del Satélite Sucre).

2.- Nivel 1: Categorías. Se contemplan 3 en total.

3.- Nivel 2: Características. Se cuenta con 7 en total.

4.- Nivel 3: Sub-características. Se cuentan con 18 en total.

5.- Nivel 4: Métricas. Para cada característica se proponen una serie de métricas utilizadas para medir la calidad del producto. Se cuentan con 95 en total.

V. CONCLUSIONES Y RECOMENDACIONES

Se propone en este artículo un instrumento de medición, basado en un modelo de calidad, para validar y verificar el funcionamiento del futuro software de simulación del Satélite Sucre, este modelo también es de propósito general, ya que, puede ser adaptado a cualquier software de simulación de satélite, proporcionando una valiosa herramienta para la estimación de la calidad, puesto que será aplicado por los operadores satelitales durante la fase de prueba del simulador.

Por otro lado, con esta medición se pretende probar el software y todos los futuros simuladores dentro de la ABAE. Finalmente se recomienda una extensión de métricas para que el modelo sea lo más efectivo y preciso posible.

REFERENCIAS

- [1] Agencia Bolivariana para Actividades Espaciales (ABAE). Consulta 25 de Julio 2016. www.abae.gob.ve
- [2] García, Solangel. Evaluación de la Calidad de la Arquitectura de Software. Universidad de Los Andes. Mérida-Venezuela. [Tesis en línea]. <http://pgcom.ing.ula.ve/SitioPGCOMP/publicaciones.php>
- [3] Calidad. (s.f.). En Wikipedia. Consultado el 12 de Octubre de 2016 en: <http://es.wikipedia.org/wiki/Calidad>
- [4] Piattini, M.; Calvo, J.; Cervera, J. y Fernández, L. (2007). Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. España, Madrid. Editorial RA-MA.
- [5] Pressman, R. Ingeniería del Software. Enfoque Práctico (5ª ed.). España, Madrid: McGraw Hill.
- [6] McCall, J.; Richards, P. y Waters, G. Factors in Software Quality (vol. III). New York: Rome.
- [7] Omaña, M. Modelo de Calidad basado en características para la selección de un sistema de Gestión de Aprendizaje. Tesis de Postgrado. USB. Venezuela. Caracas. [Documento en línea]. Consultado el 29 de octubre de 2016 en: <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AR7606.pdf>
- [8] Grimán, A. Evaluación Arquitectónica de la calidad de software. Trabajo de ascenso no publicado. Universidad Simón Bolívar, Caracas.
- [9] Mendoza, L.; Pérez, M. y Grimán, A. Prototipo de Modelo Sistemático de Calidad (MOSCA) del Software. Computación y Sistemas [Artículo en línea], 8 (3). Consultado el 18 de febrero de 2016 en: http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_21.pdf
- [10] Seminario de Calidad de Software. Universidad de Los Andes. Postgrado de Computación. María A. Pérez. Noviembre (2009)
- [11] Luis E Mendoza, María A. Pérez y Anna C. Grimán. Prototipo de Modelo Sistemático de Calidad (MOSCA) del Software. Universidad Simón Bolívar. Departamento de Procesos y Sistemas. Caracas, Venezuela (2002).
- [12] Pessagno, L, Domínguez. K, Rivas. L. Modelo de calidad para herramientas FLOSS que dan apoyo al modelado de procesos de negocio.) [Documento en línea]. Consultado el 19 de Octubre de 2016 en: https://www.researchgate.net/publication/255632144_Modelo_de_calidad_para_herramientas_FLOSS_que_dan_apoyo_al_modelado_de_procesos_del_negocio
- [13] Bass, L.; Clements, P.; Kazman, R. Software Architecture in Practice (3a ed.). Boston: Addison Wesley Pub. SEI Series in Software Engineering.
- [14] Pérez, J. y Campanero, A. Desarrollo de Software y de las Arquitecturas Software. [Documento en línea]. UNED. España, Madrid. Consultado el 19 de noviembre de 2012 en: <http://www.issi.uned.es/doctorado/softwarech/DEA/Trabajos%2010/Trabajo%20de%20Investigacion%20-%20Juan%20Antonio%20Perez-Campanero.pdf>



Solangel García nacida en Acarigua en Venezuela, el 18 de abril del año 1978, egresó como Ingeniero de Sistemas del Instituto Universitario Politécnico “Santiago Mariño”, en Mérida, en el año 2001. Realizó un Entrenamiento Profesional de Satélites, en el Instituto Shenzhou en Beijing en China en el año 2008. Realizo estudios de Maestría en Ciencias de la Computación en la Universidad de Los Andes en Mérida en el año 2014. Realizo un Diplomado On-line en Ingeniería de Procesos en el Centro de Extensión de la Universidad José Antonio Páez en el año 2015. Actualmente cursando estudios de Maestría en Gerencia de los Sistemas de Información en la Universidad Internacional del Caribe. En el 2001 ingreso a laborar en la Red de Datos de la Universidad de Los Andes en Mérida como técnico y ayudante de Administración de Servicios de Internet. En el 2006 ingresa al Centro Nacional de Tecnologías de información en Caracas como Administradora de Servicios de Internet. En el 2008 ingresa a la Agencia Bolivariana para Actividades Espaciales como operadora e Ingeniero Satelital del Satélite Simón Bolívar encargada de velar por la salud y el correcto desempeño en órbita, especialmente en el área de la carga útil de comunicaciones. En el 2016 es promovida como investigador en el área de desarrollo e innovación tecnológica para la fabricación de pequeños satélites y como encargada de los simuladores de los satélites VRSS-1 y VRSS-2.