EVALUACIÓN DE HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN EN LA FORMACIÓN INGENIERIL

L. S. Vargas Pérez[1]
A. F. Gutiérrez Tornés[2]
A. M. Soto Hernández[3]

RESUMEN

En el mundo globalizado se requiere contar con herramientas de software que faciliten el desarrollo de aplicaciones rápidas para generar sistemas de información. La formación de los Ingenieros Informáticos debe incluir métodos que permitan su evaluación para complementar su capacitación ante los retos globales. Se propone una metodología propia para realizar la evaluación técnica de las herramientas para generar sistemas de información en ambientes visuales. Con el uso de esta metodología es posible evaluar y seleccionar, de manera eficaz y fácil, las herramientas más adecuadas. Los casos de estudio elegidos para las pruebas de las herramientas RAD son las plataformas visuales comerciales Visual Studio.Net, el Eclipse y el Net Beans. El proceso de prueba se realizó con usuarios expertos y con novatos. Prácticamente, en los casos de estas tres plataformas, se obtuvieron resultados de validación sin diferencias significativas en el aspecto técnico, por lo que serían otros aspectos los que influirían en una decisión para su elección.

ANTECEDENTES

Para mejorar cualquier software se requiere evaluar sus atributos. Para ello se utiliza un conjunto de medidas y métricas significativas, que proporcionan indicadores como estrategia de evaluación técnica de la calidad del producto establecida. Este proceso se implementa con una metodología para valorar los requerimientos específicos que el producto de software debe cumplir. Es importante que las mediciones sean fáciles de calcular e interpretar en su ejecución y se eliminen ambigüedades. Algunas veces la definición académica de las características de calidad no permite medirlas directamente, por lo cual se requiere establecer métricas que las correlacionen objetivamente con un producto de software (Pressman, 2010).

El primer paso en la evaluación de un software es la determinación de las propiedades relevantes de calidad a considerar mediante un modelo de calidad específico. Éste identifica las características de calidad y sus interrelaciones con los elementos en que se descompone, para facilitar la evaluación cualitativa y cuantitativa del producto. De acuerdo con Gutiérrez (1999, 2003), generalmente se presentan en formato de árbol de estructura jerárquica donde aparecen las características en su nivel más alto, las subcaracterísticas en el intermedio y los atributos en el más bajo.

En la formación de los estudiantes de ingeniería, para enfrentar los retos globales, es primordial conocer las diferentes metodologías que permitan la evaluación de todo tipo de herramientas, que les ayude a medir y a elevar la calidad en sus productos y procesos; se requiere contar, también, con herramientas de software que faciliten el desarrollo de aplicaciones rápidas para generar sistemas de información estandarizados, y que permitan ser evaluados para determinar su calidad.

[1] Profesor investigador, Instituto Tecnológico de Ciudad Madero. laura_silvia_vargas@gmail.com
[2] Profesor investigador, Universidad Autónoma de Guerrero. afgutierrez@uagro.mx
[3] Profesor investigador, Instituto Tecnológico de Ciudad Madero. sotohana@gmail.com

Tipos de medidas. El tipo de medición requerido depende del propósito de la evaluación. Si el propósito primario es entender y corregir deficiencias, muchas de las mediciones que se realizan son para monitorear y controlar las mejoras. Si se trata de comparar la calidad de un producto con productos alternativos o contra requerimientos, es importante que la especificación de la evaluación se base en un modelo de calidad preciso, métodos de medición, escalas, y rango de niveles para cada métrica, como la norma ISO/IEC 9241 (ISO, 2008).

A nivel internacional surgen periódicamente herramientas para facilitar la creación de nuevos sistemas de información, lo que implica realizar evaluaciones sobre la calidad de estas herramientas para una selección adecuada en determinado entorno de desarrollo.

Tal es el caso de la evaluación y selección de las herramientas conocidas como RAD (*Rapid Application Development*) que forman parte de un IDE (*Integrated Development Environment* -Entorno o Ambiente de Desarrollo Integrado- que constituye el marco de trabajo de los programadores de aplicaciones para crear sistemas de información.

Objetivo. La formación de Ingenieros Informáticos debe incluir métodos que les permitan evaluar aquellas herramientas que van a utilizar para el desarrollo de nuevos sistemas de información o aplicaciones. Para ello, se propone el aprendizaje del modelo MECHDAV (MECRAD en inglés) que permite hacer un análisis comparativo de diferentes tipos de herramientas para el desarrollo de aplicaciones en ambientes visuales, mediante una propuesta de métricas de calidad (Vargas *et al.*, 2008).

Estado del arte y trabajos relacionados

Desde hace algunos años se han estado proponiendo diversos tipos de modelos de calidad que se han convertido en estándares: ISO/IEC 25000 (ISO, 2005), ISO/IEC 14598 (ISO, 1998), ISO/IEC 9126 (ISO, 1997), así como las IEEE 1061, IEEE 610, (1992 y 1994). Estos modelos son muy útiles pero a su vez muy genéricos, por lo que deben adaptarse para su utilización práctica.

La gran mayoría de los trabajos asociados con esta temática se enfoca a la evaluación de la calidad en el desarrollo de software, como los de Cochea (2009), Olsina (2006), Piattini (2006), Moreno (2004). En este trabajo se desarrolló la metodología denominada MECHDAV utilizando los modelos y estándares ya mencionados tomando características, subcaracterísticas y atributos distintas de cada uno (Gutiérrez, 2003, 1999), además de otras cualidades del Inventario de Medidas de Usabilidad del Software SUMI (Human Factors Research Group, 2000)

METODOLOGÍA

Se generó una metodología propia con modelos, procesos, técnicas y herramientas, que permite hacer cuadros comparativos para realizar la evaluación técnica de las herramientas RAD. El modelo propuesto se basa en la fusión: adopción y adaptación de las contribuciones señaladas más arriba, para conformar la arquitectura del modelo. Es necesario precisar que las aplicaciones utilizadas para probar la metodología son productos

comerciales por lo que no se dispone de la información concerniente a su código fuente. Lo anterior implica que no se tomaron en cuenta las métricas internas.

Proceso de evaluación

Las actividades de evaluación en el modelo propuesto son las que se indican en la Figura 1, donde se observa que aquellos atributos relacionados con la calidad en uso tiene una importancia singular, ya que es desde el punto de vista del usuario. La relación de la calidad en uso con las otras características de calidad del software depende del tipo de usuario, como aquel para quien la calidad en uso es principalmente el resultado de la evaluación de la funcionalidad, la confiabilidad, la usabilidad, la eficiencia y la reusabilidad (Gutiérrez, 1999, 2003; ISO, 2005).

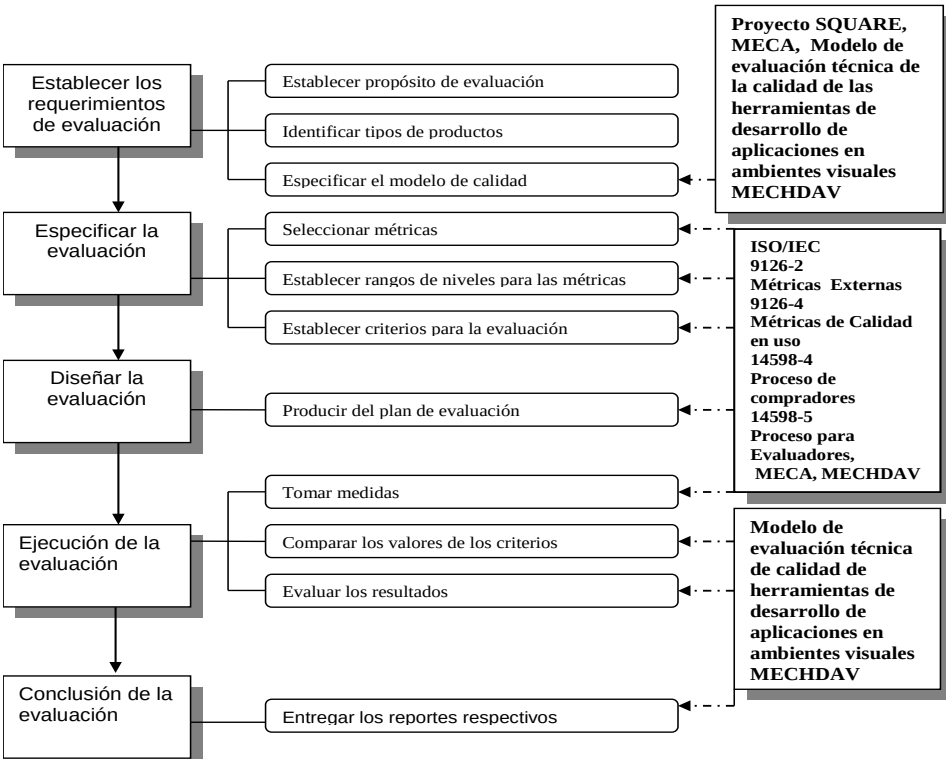

Figura 1. Proceso de evaluación para el modelo propuesto MECHDAV.

Compactación del modelo MECHDAV

El modelo MECHDAV está compuesto por cualidades específicas del *software* (características), subcaracterísticas, atributos y métricas, los cuales se agrupan en dos continentes: el modelo completo y el compacto, para facilitar su enseñanza y comprensión. El uso del modelo completo se recomienda para usuarios expertos, profesores y alumnos de semestres avanzados de ingeniería y posgrado que desarrollan aplicaciones de *software*. En la Figura 2 se muestra el modelo MECHDAV completo.

Para los usuarios novatos y los alumnos de los primeros semestres de ingeniería se recomienda el modelo compacto. Aunque uno de los propósitos de la aplicación de este modelo de evaluación técnica es la de proporcionar un medio de comparación objetivo

entre diversos productos de software dentro de un ambiente visual para cualquier tipo de usuario (sea experto o principiante).

Características /Subcaracterísticas / Atributo / Métrica

1.1.1.1 Funcionalidad / *Completitud* / Contenido total / métrica
1.2.1.1 Funcionalidad / *Consistencia* / Uniformidad del formato de los elementos funcionales métrica
1.2.2.1 Funcionalidad/ *Consistencia* / Uniformidad de retorno al procesamiento / métrica
1.2.3.1 Funcionalidad/ *Consistencia* /Uniformidad del vocabulario y de la simbología utilizada /métrica
1.3.1.1 Funcionalidad/ *Corrección* / Correspondencia de las descripciones con los objetos métrica
1.3.2.1 Funcionalidad/ *Corrección* / Funcionamiento correcto / métrica
1.3.3.1 Funcionalidad/*Corrección* / Utilización correcta del idioma / métrica
1.4.1.1 Funcionalidad/ *Interoperabilidad* / Intercambiabilidad de datos / métrica
1.4.2.1 Funcionalidad/*Interoperabilidad/* Intercambiabilidad en componentes e interfaces métrica
1.5.1.1 Funcionalidad/ *Normalización* / Normalización de la simbología / métrica
1.5.2.1 Funcionalidad / *Normalización/* Normalización del vocabulario / métrica
2.1.1.1 Confiabilidad / Madurez / Frecuencia de fallas / métrica
2.2.1.1 Confiabilidad / *Recuperabilidad* / Opciones de recuperabilidad / métrica
2.3.1.1Confiabilidad /*Tolerancia de errores o fallos* /Procesamiento degradado /métrica
2.3.2.1 Confiabilidad/ *Tolerancia de errores o fallos* / Tratamiento de errores / métrica
3.1.1.1 Usabilidad / *Aprendibilidad* / Demo / métrica
3.1.2.1 Usabilidad / *Aprendibilidad* / Eficacia del Demo / métrica
3.1.3.1 Usabilidad / *Aprendibilidad* / Tutorial / métrica
3.1.4.1 Usabilidad / *Aprendibilidad* / Eficacia del Tutorial / métrica
3.1.5.1 Usabilidad / *Aprendibilidad* / Documentación / métrica
3.2.1.1 Usabilidad / *Atractividad* / Interacción atractiva / métrica
3.2.2.1 Usabilidad / *Atractividad* / Recuperación exitosa / métrica
3.2.3.1 Usabilidad / *Atractividad* / Tiempo de operación / métrica
3.3.1.1 Usabilidad / *Comprensibilidad* / Ayuda en línea / métrica
3.3.2.1 Usabilidad / *Comprensibilidad* / Interfaz de usuario adecuada / métrica
3.3.3.1 Usabilidad / *Comprensibilidad* / Terminología acorde al usuario / métrica
3.4.1.1 Usabilidad / *Difusión* / Amplitud / métrica
3.4.2.1 Usabilidad / *Difusión* / Frecuencia de operación / métrica
3.5.1.1 Usabilidad / *Operabilidad/* Operabilidad de las ayudas / métricas
3.5.2.1 Usabilidad / *Operabilidad* / Utilidad de las ayudas / métrica
4.1.1.1 Eficiencia / *Escalabilidad* / Disponibilidad / métrica.
4.2.1.1 Eficiencia / *Uso de los recursos* / Eficiencia en los recursos / métrica
4.3.1.1 Eficiencia / *Uso del tiempo* / Eficiencia en tiempo / métrica
5.1.1.1 Portabilidad / *Instalabilidad* / Módulo de instalación / métrica
5.1.2.1 Portabilidad / *Instalabilidad* / Documentación del módulo de instalación / métrica
5.1.3.1 Portabilidad / *Instalabilidad* / Módulo de configuración / métrica.
5.1.4.1 Portabilidad / *Instalabilidad* / Documentación del módulo de configuración / métrica.
5.2.1.1 Portabilidad / *Adaptabilidad* / Independencia del hardware / métrica.
5.2.2.1 Portabilidad / *Adaptabilidad* / Independencia del ambiente de software / métrica.
6.1.1.1 Calidad en uso / *Eficacia* / Eficacia de tareas / métrica.
6.1.2.1 Calidad en uso / *Eficacia* / Rendimiento de tareas / métrica.
6.2.1.1 Calidad en uso / *Productividad* / Eficiencia relativa al usuario / métrica.
6.2.2.1 Calidad en uso / *Productividad* / Proporción productiva / métrica.
6.3.1.1 Calidad en uso / Satisfacción / efectos psicológicos favoritos de usuario: SUMI métricas.

Figura 2. Modelo MECHDAV completo.

Definición de las métricas y de la escala evaluativa

Las evaluaciones se realizan midiendo los atributos a los que se les asigna una métrica correspondiente. El resultado, el valor medido, puede ser mapeado sobre una escala. Este valor no muestra en sí mismo el nivel de satisfacción de los requerimientos. Para este propósito la escala se divide en rangos que definen el grado de satisfacción.

El modelo utiliza dos maneras para hacerlo: 1) dividir la escala en dos categorías como son insatisfactorio y satisfactorio; 2) dividir la escala en cinco niveles o categorías denominadas A, B, C, D y E. Observar la Figura 3.

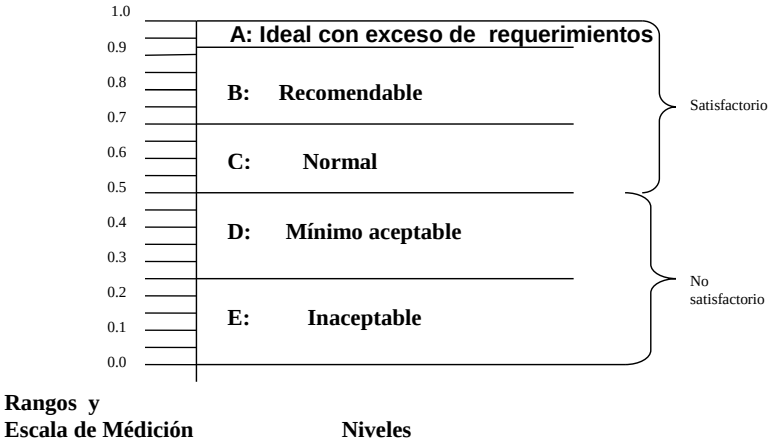

Figura 3. Rangos y escala de medición para las métricas.

Dado que en la norma IEEE 610 (1994) se define la métrica como "una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado", para usar la segunda forma se propone un conjunto de rangos de escala numérica donde se califica cada una de las métricas. La evaluación se realiza llenando un cuestionario donde, al seleccionar una respuesta, se determina el valor de la métrica, asociada a la escala presentada en la Tabla 1.

Tabla 1. Interpretación de rangos de niveles de métricas.

VALOR	% CUMPLIMIENTO	SIGNIFICADO / INTERPRETACIÓN	RANGO
1.0	90 - 100	Excelente / Siempre	A
0.8	70 - 89	Satisfactorio / Casi siempre	B
0.6	50 - 69	Aceptable / Regularmente	C
0.4	30 - 49	Deficiente / En ocasiones	D
0.0	0 - 29	Inaceptable / Nunca o raras veces	E

Para respaldar este modelo se han desarrollado y documentado 44 métricas, tal como aparece en el formato de la Figura 4, más 11 que corresponden a la característica: Calidad en Uso; subcaracterística: Satisfacción; atributo: Efectos psicológicos de usuario.

Las descripciones y aplicaciones de las métricas se enfocan en: la observación del comportamiento del software en cuanto a la diferencia entre los resultados de ejecución actuales y la especificación de los requerimientos (una vista de prueba y validación de calidad); las ocurrencias inesperadas del comportamiento en el tiempo o utilización de recursos durante la operación del software; las instrucciones directas al usuario para que realice una tarea específica, tomando nota de ciertos indicadores (por ejemplo: tiempo, número de ocurrencias de algún evento, etc.); y los cuestionamientos directos al usuario, en los cuales sólo se determina la existencia (presencia o ausencia) de la herramienta y/o la característica a analizar (atributos esenciales).

Característica: **3. USABILIDAD.**
Subcaracterística: *3.5 Operabilidad.*
Atributo: 3.5.2 Operabilidad de las ayudas.
Métrica: 3.5.2.1 Relación de resultados exitosos mediante la facilidad de operación de la ayudas del software.
Objetivo: Determinar la relación del esfuerzo realizado por el usuario para obtener primeros resultados exitosos mediante la facilidad de operación de las ayudas del software.
Método: Analizar los componentes de cada herramienta para determinar la relación de resultados exitosos mediante la facilidad de operación de las ayudas del software.
Fórmula: X= A/B
Evaluación: E(x)={ (0,0), (0.4, 40), (0.6, 60), (0.8, 80), (1,100)}
***Medidas*:** A= Número de primeras tareas cuyos objetivos son completados exitosamente, utilizando las ayudas.
B= Número de tareas intentadas por el usuario.
***Interpretación*:** Guía interactiva (intentos de usuario en tareas exitosas)
0 < = X < = 1; lo más cercano a 1 es lo mejor.
Fuente de referencia: *nueva de* MECHDAV, ISO 9126.

Figura 4. Ejemplo de la documentación de una de las métricas utilizadas.

Plasmar los resultados tanto parciales como totales de la evaluación de la calidad del software no es tarea fácil, por lo que se deben elegir formatos simples y comprensibles para conseguir una valoración rápida y confiable de la calidad de sus diferentes representaciones. Los casos de estudio elegidos para las pruebas de las herramientas RAD son las plataformas visuales comerciales Visual Studio.Net (Microsoft, 2006), Net Beans (Sun Microsystems, 2004-2010) y Eclipse (Eclipse Foundation, 2005), en diferentes versiones. Estos paquetes de software no requieren la compra de las licencias, porque dos son productos de código abierto (Net Beans y Eclipse), son gratuitos y sin restricciones de uso; el único producto que es de licencia comercial para su venta (Visual Studio.Net), es otorgado a las instituciones educativas, para uso de sus estudiantes y docentes, de manera gratuita, durante el periodo escolar.

DISCUSIÓN DE LOS RESULTADOS

Se realizaron diez pruebas para los ambientes visuales estudiados. Seis de ellos se realizaron por usuarios expertos (maestros y alumnos del posgrado del ITCM) y cuatro por principiantes (alumnos de ingeniería en sistemas computacionales del ITCM). Las evaluaciones las realizaron los alumnos de la licenciatura de Ingeniería en Sistemas Computacionales (usuarios inexpertos), profesores y alumnos del posgrado de la Maestría en Ciencias de la Computación (usuarios expertos).

Los resultados obtenidos por la evaluación técnica durante todo el proceso de medición, para cada una de las herramientas RAD -Visual Studio.Net, Net Beans y Eclipse- se muestran en las figuras 5, 6, 7 y 8. Para este fin se eligieron formatos tales como listas de comprobación (*checklist*), tablas simples de relación y matrices de control.

En la Figura 5 se muestra un ejemplo de una fracción de la matriz de control que se utiliza para obtener los resultados de la evaluación de cada una de las herramientas objetos de estudio. Cuando se obtienen los valores respectivos de la evaluación del software elegido,

se genera el reporte final de la evaluación, en el que se plasman los resultados definitivos y el porcentaje de cumplimiento. A su vez, se proporciona un esquema donde se muestran los puntos donde el software obtiene un buen nivel de calificación de calidad, con qué criterio lo alcanza y además se brinda una recomendación final acerca de su aceptación o rechazo.

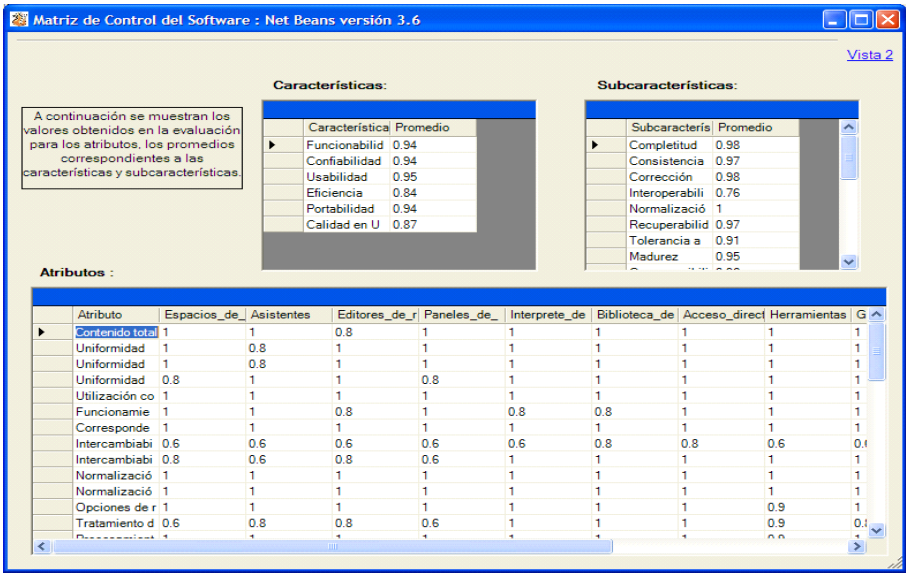

Figura 5. Ejemplo de matriz de control que se genera con la evaluación del ambiente Visual Net Beans, versión 3.6.

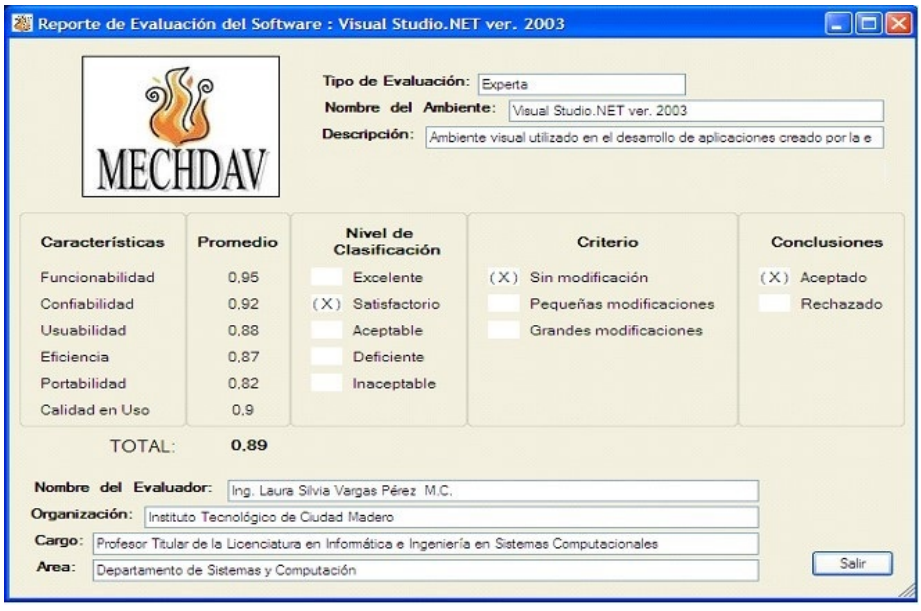

Figura 6. Reporte técnico final de la evaluación del ambiente Fuente: Visual Studio.Net, versión 2003.

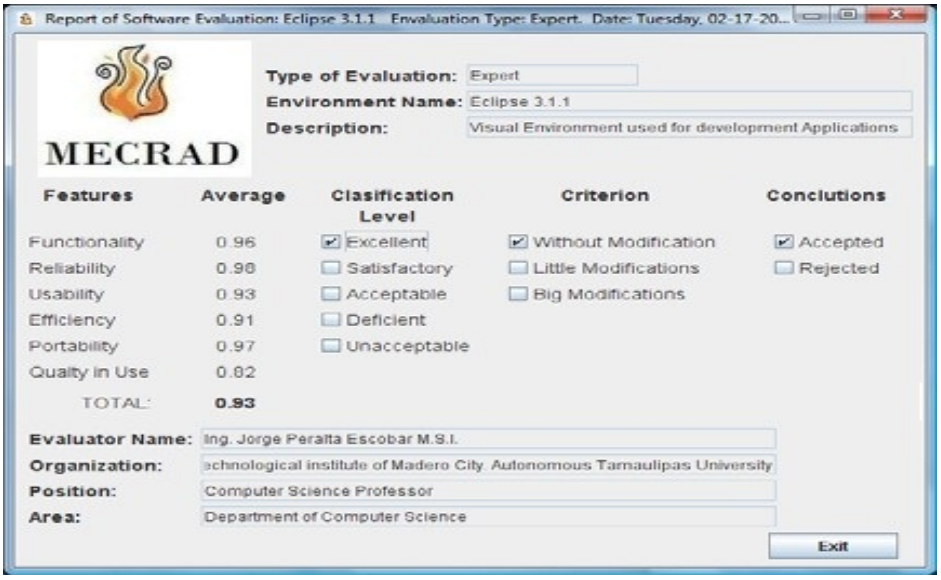

Figura 7. Reporte técnico final de la evaluación del sistema en inglés
Fuente: Estudiante de posgrado del ambiente Eclipse 3.1.1.

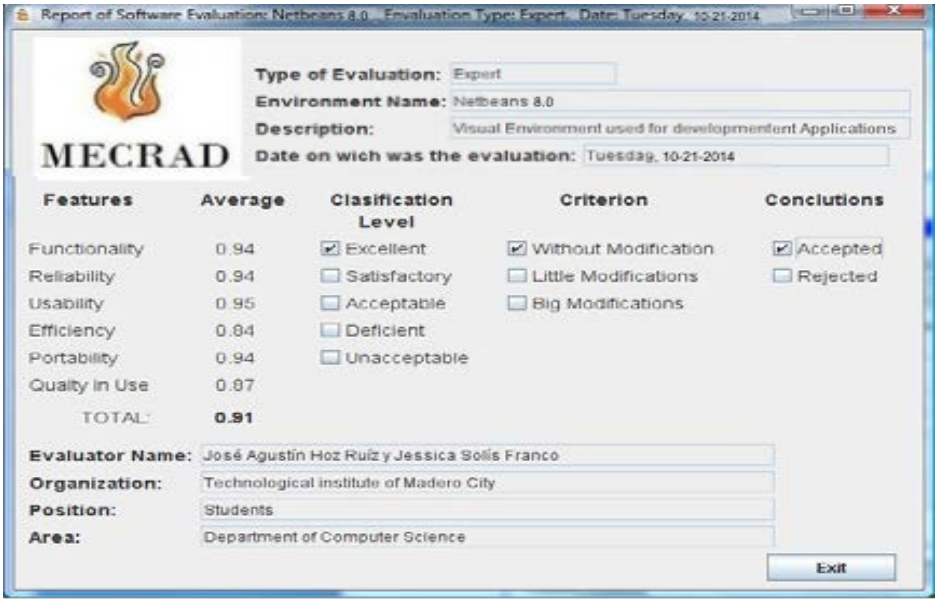

Figura 8. Reporte técnico final de la evaluación del sistema en inglés
Fuente: Estudiante de posgrado del ambiente Netbeans 8.0.

CONCLUSIONES

De acuerdo con los resultados obtenidos en las pruebas realizadas a los ambientes de desarrollo integrado (IDE) y sus herramientas RAD, seleccionados como casos de estudio, se llegó a las conclusiones que se presentan a continuación:

El Visual Studio.Net (Microsoft, 2006) obtuvo una evaluación general promedio de 0.89 (89%) entre los usuarios principiantes y un puntaje de 0.88 (88%) entre los expertos, dentro del rango de valores propuesto, por lo que el nivel de clasificación de calidad alcanzado es satisfactorio sin recomendaciones, ya que no requiere de modificaciones en su diseño -sólo actualizaciones- y por lo tanto es aceptado ampliamente. Vale la pena comentar que una de sus mayores debilidades se ubica en la característica de portabilidad, lo cual es comprensible dada su dependencia de la plataforma Windows de Microsoft (Figura 5).

El Net Beans (Sun Microsystems, 2004-2010) y el Eclipse (Eclipse Foundation, 2005), obtuvieron una evaluación general promedio de 0.92 (92%) y 0.93 (93%), respectivamente, entre los usuarios expertos que lo evaluaron. El nivel de clasificación de calidad alcanzado es excelente, sin recomendaciones, ya que no requiere de modificaciones en su diseño (sólo actualizaciones) por lo que se recomienda ampliamente; aunque se puede comentar que se podría pensar que su mayor debilidad se ubica en su calidad en uso, dado que para la utilización de ambos es requisito indispensable el dominio del lenguaje Java, pero como se evalúan las herramientas RAD, no influye en el resultado.

Debido a la similitud de los resultados de la evaluación técnica, para decidir cuál seleccionar se deben tomar en cuenta otros criterios, también importantes, como son los costos, la posibilidad de interacción con otros sistemas, la posible inserción en la plataforma en la que se van a desarrollar los sistemas, etc.

Como trabajo futuro se recomienda hacer revisiones periódicas a MECHDAV para su actualización, así como tratar de introducir el modelo de evaluación técnica de las herramientas de desarrollo de aplicaciones en los ambientes visuales de los sitios *web* y en ambientes de otro tipo.

BIBLIOGRAFÍA

Cochea T. (2009). Métrica de Calidad de los sistemas de información – aplicación en la certificación de calidad de un Sistema de una empresa del sector hidrocarburífero. *Artículos de Tesis de Grado – ICM, Escuela Superior Politécnica del Litoral*, Guayaquil, Ecuador.

Eclipse Foundation (2005). *Eclipse Entorno de Desarrollo Integrado IDE libre de código abierto.* Versión 3.1.1. Compañía Eclipse Foundation. Sitio WEB: http://www.eclipse.org

Gutiérrez, A. (2003). *Modelo de evaluación para el aseguramiento de la calidad del Software, Modelo MECA.* México: Instituto Politécnico Nacional.

Gutiérrez, A. (1999). *Metodología para el aseguramiento de la calidad del Software MACS.* México: Instituto Politécnico Nacional.

Human Factors Research Group. (2000). *SUMI: Software Usability Measurement Inventory, European Directive on Minimum Health and Safety Requirements for Work with Display Screen Equipment.* Londres: Taylor & Francis Ltd.

IEEE. (1994). *Software Engineering Standards Collection. Standard Glossary of Software Engineering Terminology*. *IEEE, STD 610.* Institute of Electrical and Electronics Engineers.

IEEE. (1992). *Software Quality Metrics Methodology. IEEE 1061 Standard for a Software Quality Metrics Methodology*. Institute of Electrical and Electronics Engineers.

ISO. (1997). *ISO/IEC 9126. Software Product Evaluation – Part 2: External metrics. Part 4: Quality in use metrics*--. International Organization for Standardization.

ISO. (1998). *ISO/IEC 14598:1998. Information technology -- Software product evaluation – Part 4: Process for evaluators, Part 5: Process for acquirers*. International Organization for Standardization.

ISO. (2008). *ISO 9241-151. Ergonomics of human-system interaction -- Part 151: Guidance on World Wide Web user interfaces and industry standards for user centered design.* International Organization for Standardization.

ISO. (2005). *ISO/IEC 25000 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE.* International Organization for Standardization.

Microsoft (2006). *Visual Studio.Net 2003. Entorno de Desarrollo Integrado IDE para sistemas operativos Windows.* Versión 7.1. Microsoft Visual Studio. Microsoft Company. Sitio WEB: https://www.visualstudio.com/es-es/visual-studio-homepage-vs.aspx

Moreno, M. (2004). Aplicación de las Métricas de Calidad del Software en la Evaluación Objetiva de Gramáticas Independientes de Contexto Inferidas. *I Simposio Avances en Gestión de Proyectos y Calidad del Software* (pp. 209-220). Salamanca: Universidad de Salamanca.

Olsina, L. (2006). Medición y Evaluación de Calidad en Uso: Un Caso de Estudio para una Aplicación E-Learning. *9º Taller Iberoamericano de Ingeniería de Requisitos y Ambientes de Software* (pp. 317-330). La Plata: Centro Latinoamericano de Estudios en Informática.

Piattini, M. (2006). Métricas para la Evaluación de Modelos de Proceso de Negocio. *9º Taller Iberoamericano de Ingeniería de Requisitos y Ambientes de Software* (pp. 419-432). La Plata: Centro Latinoamericano de Estudios en Informática.

Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico*. Séptima Edición. Madrid: McGraw Hill/Interamericana.

Sun Microsystems (2004-2010). *NetBeans Entorno de Desarrollo Integrado IDE libre de código abierto.* Versión 3.6., Versión 8.0. Sun Microsystems / Oracle Corporation. Sitio web: https://netbeans.org/

Vargas-Pérez, L. S., Gutiérrez-Tornés, A. F., Felipe- Riverón, E. M. (2008). MECRAD: Model and Tool for the Technical Quality Evaluation of Software Products in Visual Environment. *3rd International Multi-Conference on Computing in the Global Information Technology* (pp. 107-112). Athens: IEEE Computer Society.

Vargas-Pérez, L. S., Gutiérrez-Tornés, A. F., Felipe-Riverón, E. M. (2012). *Sistema para evaluar herramientas RAD*. Registro ante SEP INDAUTOR con número de registro público 03-2012-100211292400-01. Ciudad de México, México.