

DEPÓSITO LEGAL ZU2020000153

ISSN 0041-8811

E-ISSN 2665-0428

Revista de la Universidad del Zulia

Fundada en 1947
por el Dr. Jesús Enrique Lossada



Ciencias del
Agro,
Ingeniería
y Tecnología

Año 16 N° 45

Enero - Abril 2025

Tercera Época

Maracaibo-Venezuela

Selección de herramientas para ambientes de desarrollo integrado habilitados con inteligencia artificial

Giovanna Virginia Cianfaglione *

Frank Alejandro Caicedo **

María Pérez ***

Dinarle Ortega ****

RESUMEN

El avance tecnológico ha impulsado el crecimiento acelerado de herramientas de software tipo Entornos de Desarrollo Integrados (IDE) habilitadas por Inteligencia Artificial (IA), ofreciendo a los desarrolladores oportunidades ilimitadas de abordar problemas, automatizar tareas y potenciar la creatividad en diversas áreas, desde el análisis de datos hasta la generación de contenido, lo que hace que su selección no sea una tarea fácil. El objetivo de este artículo es evaluar tres herramientas tipo IDE habilitadas con IA tales como, Visual Studio Code, NetBeans y Apple Xcode, usando el método de Evaluación de Análisis de Características de DESMET, con sus respectivas métricas obtenidas a partir del método Goal Question Metric (GQM). La identificación del conjunto de características, están fuertemente orientadas en las características de los estándares ISO 25010 y ISO 25059. De esta manera, se cuenta con un mecanismo objetivo y preciso para seleccionar la herramienta IDE habilitadas con IA, más adecuada a las necesidades de cada proyecto de software.

PALABRAS CLAVE: Entornos de Desarrollo Integrado, Análisis de características, ISO 25010, ISO 25059, Inteligencia Artificial.

*Escuela de Ingeniería de Sistemas, Universidad Metropolitana, Caracas (Venezuela) ORCID: <https://orcid.org/0009-0004-8839-6034> E-mail: cianfaglione.giovanna@correo.unimet.edu.ve

** Escuela de Ingeniería de Sistemas, Universidad Metropolitana, Caracas (Venezuela) ORCID: <https://orcid.org/0009-0007-7257-8564> E-mail: frank.caicedo@correo.unimet.edu.ve

*** Escuela de Ingeniería de Sistemas, Universidad Metropolitana, Caracas (Venezuela) ORCID: <https://orcid.org/0009-0000-9288-5763> E-mail: maperez@unimet.edu.ve

**** Escuela de Ingeniería Informática, Universidad Católica Andrés Bello (Venezuela) ORCID: <https://orcid.org/0000-0002-6503-352X> E-mail: dortega@ucab.edu.ve

Recibido: 17/09/2024

Aceptado: 20/11/2024

Selection of Tools for AI-Enabled Integrated Development Environments

ABSTRACT

Technological advancements have driven the rapid growth of software tools such as Integrated Development Environments (IDEs) enabled by Artificial Intelligence (AI), offering developers unlimited opportunities to tackle problems, automate tasks, and enhance creativity in various areas, from data analysis to content generation, making their selection not an easy task. The objective of this article is to evaluate three AI-enabled IDE tools, such as Visual Studio Code, NetBeans, and Apple Xcode, using the DESMET Feature Analysis Evaluation method, with their respective metrics obtained from the Goal Question Metric (GQM) method. The identification of the feature set is strongly oriented by the characteristics of the ISO 25010 and ISO 25059 standards. In this way, an objective and precise mechanism is provided to select the most suitable AI-enabled IDE tool for the needs of each software project.

KEY WORDS: Integrated Development Environments, Feature Analysis, ISO 25010, ISO 25059, Artificial Intelligence.

Introducción

En un mundo cada vez más digitalizado, donde el software se ha convertido en una parte fundamental de la tecnología, la calidad del mismo es un aspecto crucial (Techopedia, 2023). Desde el funcionamiento de sistemas críticos hasta el entretenimiento personal, la confianza en la precisión, seguridad y eficacia del software, son esenciales. Según Amazon (2023), “un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software, de manera eficiente”.

En este sentido, la elección adecuada de las herramientas IDEs, es fundamental para la calidad de los productos finales. En particular, en el enfoque de desarrollo de software, como DevSecOps, los IDEs, integran funcionalidades esenciales como la edición de código, pruebas automáticas, y empaquetado de software, lo que asegura un flujo de trabajo más eficiente y menos propenso a errores (Amazon, 2023).

Adicionalmente, los IDEs soportan la generación de código limpio y eficiente, reduciendo así la probabilidad de errores y mejorando la mantenibilidad del software, de esta manera, facilitan el aprendizaje y la adaptación de nuevos desarrolladores al entorno de trabajo, mejorando así la productividad y eficiencia del equipo (Red Hat, 2023).

La integración de la Inteligencia Artificial (IA) en los IDEs, facilita además, la generación automatizada de código, la identificación de errores mejoran la comprensión y mantenimiento del código, entre otros, ofreciendo así, sugerencias basadas en descripciones en lenguaje natural, reduciendo el tiempo y esfuerzo necesarios para desarrollar software y asegurando productos de software más robustos y eficientes (Finio y Downie, 2024).

En este trabajo se evalúan tres herramientas de tipo IDE, utilizando el método de Análisis de Características con estudios de caso; el cual se usa para evaluar las tecnologías de software a través de un conjunto de métricas rigurosamente definidas, que permiten una valoración detallada y objetiva de cada una de las herramientas (Kitchenham, 1998).

Luego de aplicar el método de evaluación, se obtienen 188 métricas, que aplicadas bajo tres técnicas diferentes permiten al ingeniero de software identificar con precisión y rigor, las debilidades y fortalezas de los IDEs habilitados con IA, que se ofrecen actualmente en el mercado y facilitar la toma de decisión, al momento de seleccionar estas herramientas.

1. Contexto del Problema

En estos tiempos existen enfoques de desarrollo de software tales como DevSecOps que impulsan el uso intensivo de herramientas, las cuales deben ser seleccionadas rigurosamente porque además de la cantidad numerosa existente y en constante crecimiento, éstas deben estar alineadas con las condiciones del equipo de desarrollo y del problema a resolver. En este contexto, las herramientas tipo IDE desempeñan un papel crucial, ya que permiten a los desarrolladores iniciar la programación de nuevas aplicaciones, de manera rápida y eficiente, usando el autocompletado inteligente y la generación automatizada de código, minimizando así, la escritura de largas secuencias de caracteres, permitiendo a los desarrolladores centrarse más en la lógica y el diseño del software (Red Hat 2023).

La incorporación de la Inteligencia Artificial (IA) en estas herramientas, mejora significativamente la facilidad, rendimiento y productividad en la creación de software porque acelera el proceso de programación mediante la generación de código y sugerencias de funciones completas basadas en descripciones en lenguaje natural. Además, la IA puede predecir las modificaciones que se pudieran realizar en el código, proporcionando finalizaciones inteligentes que agilizan la escritura del mismo, especialmente cuando se trabaja con código de otros desarrolladores (Red Hat 2023) (Finio y Downie, 2024).

Klaxoon (2024) y Orta (2024) coinciden en que la elección de las herramientas dependerá del tamaño, la complejidad y el presupuesto del proyecto; así como de la cultura organizacional, razón por la cual, se debe contar con un mecanismo que permita seleccionar dichas herramientas. (Software Development (linkedin), 2024), (SW Team, 2024). En este sentido, una mala elección de las herramientas, puede causar problemas de eficiencia y calidad (Orta, 2024; Valdovinos, 2023).

El objetivo es evaluar las herramientas IDE siguiendo el Método de Análisis de Características con Estudio de Caso, para seleccionar herramientas de Entorno de Desarrollo Integrado (IDE) habilitadas con IA. Ahora bien, para la identificación de estas características, resultaron claves las normas ISO 25010 para la calidad de software y la ISO 25059 para la calidad de la IA, garantizando de esta manera una comprensión y definición de términos estándares entre los involucrados. En cuanto a la primera norma, define con precisión las características que debe cumplir un producto de software de calidad. En cuanto a la segunda norma, define las características de calidad necesarias de los componentes de IA antes de su lanzamiento al mercado (NormasISO, 2024). Este trabajo está inserto en una investigación en progreso.

Considerando todo lo mencionado y analizado, surge la pregunta: ¿Cómo podemos hacer un análisis de características con base en los estándares ISO 25010 e ISO 29059 para evaluar herramientas IDE habilitadas con IA?

2. Análisis de características por estudio de caso de herramientas IDE

El Método Análisis de Características por Estudio de Caso se usa para evaluar y seleccionar tecnologías de software, siguiendo estas actividades (Kitchenham, 1998): *Primero*, se definen los objetivos y características clave, para ello se formula el modelo de análisis basado en características. *Segundo*, se eligen las tecnologías candidatas y se identifican los estudios de caso representativos. *Tercero*, se recopilan los datos cualitativos y los cuantitativos aplicando las métricas que operacionalizan el modelo. *Cuarto*, los resultados se analizan para comparar el desempeño de cada tecnología, identificando fortalezas y debilidades. y *Quinto*, se toma una decisión informada sobre qué tecnología adoptar, documentando el proceso y los resultados. Este enfoque asegura una evaluación detallada y objetiva, basada en el rendimiento real (Kitchenham, 1998).

La aplicación de este método de evaluación conduce a ajustes en las métricas para mejorar su precisión y relevancia, asegurando que no sólo fueran teóricamente sólidas, sino también prácticas y útiles en situaciones reales.

2.1 *Primero*, definición de los objetivos y características clave orientados por los estándares ISO 25010 y ISO 25059

Siguiendo los pasos propuestos por Kitchenham (1998); la formulación de este modelo de análisis incluyó una revisión exhaustiva de la literatura relacionada con herramientas IDE habilitadas con IA y el enfoque de desarrollo DevSecOps. Esta revisión se centró en el estudio de diversas herramientas IDE, con el objetivo de identificar sus características de calidad.

Estas características se seleccionaron con base en las funcionalidades y no funcionalidades que comúnmente ofrecen las herramientas de tipo IDE. Por lo que la instanciación del modelo de análisis consta de cinco características esenciales definidas según la norma ISO 25010, 1: *adecuación funcional*, con un enfoque específico en las subcaracterísticas *completitud funcional* y *corrección funcional* en el contexto de gestión de interfaz, edición de código y depuración; 2: *flexibilidad*, con énfasis en la sub-característica de *adaptabilidad*; 3: *compatibilidad*, particularmente en la subcaracterística *interoperabilidad*; 4: *eficiencia*, evaluada a través del *comportamiento temporal*; y 5: *capacidad de iteración*, centrada en la subcaracterística *operabilidad*.

En cuanto a las habilidades relacionadas con la IA, se adoptaron las directrices del estándar ISO 25059, una extensión de la norma ISO/IEC 25010 que aborda aspectos adicionales presentes en los sistemas de IA (ISO 25000, 2022). De este estándar, se seleccionaron dos características; 1: *seguridad*, con un enfoque en la sub-característica de *intervenibilidad*; y 2: *usabilidad*, particularmente en la sub-característica de *controlabilidad del usuario*.

Es oportuno señalar, que la característica *adecuación funcional* es fundamental, especialmente la sub característica de *completitud funcional*, en el caso de las IDEs, la corrección funcional, es particularmente necesaria, dado que se trata de elaborar código, lo cual exige una alta precisión, al momento de la generación de código sin errores (López y Noguera, 2016; Hernández, Velasco y Benítez-Guerrero, 2016).

En cuanto a cumplir con la característica de *flexibilidad*, especialmente en términos de *adaptabilidad*, le permite a la herramienta de tipo IDE, ajustarse a diferentes entornos de desarrollo y requisitos sin cambios significativos. De igual forma, la característica de *compatibilidad*, particularmente su *interoperabilidad*, permite a la herramienta trabajar en conjunto con otros sistemas y herramientas de desarrollo, lo cual es especialmente relevante en un enfoque DevSecOps (Red Hat, 2023; Software Development, 2024).

Asimismo, la característica de eficiencia, especialmente su comportamiento temporal, asegura que las acciones del usuario se ejecuten sin demoras, que las tareas complejas como la compilación y ejecución se realicen de manera ágil, y que el manejo de proyectos grandes no degrade el rendimiento. La capacidad de iteración, especialmente su operabilidad, permite a los desarrolladores aprender rápidamente su uso, realizar tareas de manera eficiente y adaptar la herramienta a sus necesidades (Software Development (LinkedIn), 2024; Cook, Samalam y Unewisse, 2021). La seguridad, especialmente en términos de intervenibilidad, permite una rápida identificación y corrección de fallos, asegurando que los datos y modelos de IA estén protegidos contra accesos no autorizados, y facilitando la recuperación tras incidentes, minimizando riesgos asociados a errores y vulnerabilidades. Por otro lado, la usabilidad del componente IA, no sólo facilita una experiencia de usuario más intuitiva, sino que también reduce la curva de aprendizaje, optimizando la interacción con la herramienta y mejorando la eficiencia en el desarrollo y ajuste de modelos de IA (Hernández, Velasco y Benítez-Guerrero, 2016; Cook, Samalam y Unewisse, 2021).

Luego de determinar las características a evaluar mediante el análisis de estudios de caso, se procedió a su operacionalización. Para la operacionalización, se siguió el método Goal Question Metric (GQM) (Estrada y Maya 2016), este método ofrece un enfoque sistemático para definir y medir la calidad del software mediante la formulación de métricas específicas basadas en los objetivos a alcanzar (Genially, 2021). Para el modelo propuesto, la meta es el logro de las sub-características identificadas en el modelo instanciado. Estas métricas están diseñadas para capturar datos precisos sobre aspectos específicos del software, permitiendo una evaluación integral de cómo cada herramienta IDE cumple con los estándares establecidos.

2.2. *Segundo*, selección de las tecnologías candidatas, serán los estudios de caso

Para la selección de los casos de las Herramientas tipo IDE, que se les hará el Análisis de Características, fueron: *primero*, la popularidad y la adopción de estas herramientas en la comunidad de desarrolladores. Dado que el uso extendido es un indicador de la calidad y la confiabilidad de una herramienta, por lo que se optó por la búsqueda de IDEs, con una amplia base de usuarios y una comunidad activa que pueda brindar soporte y recursos.

Segundo, la compatibilidad y el soporte de plataformas. En un entorno de desarrollo diverso, donde los equipos pueden utilizar diferentes sistemas operativos, es fundamental que las herramientas IDE seleccionadas puedan funcionar sin problemas en los principales entornos (Windows, macOS, Linux).

Tercero, la funcionalidad y las características de las herramientas IDE. Se aseguró de que cubrieran las necesidades específicas del proyecto, como el editor de código, el depurador, el administrador de proyectos, el control de versiones, la refactorización y las pruebas unitarias, entre otras.

Cuarto, la facilidad de iteración, se desea IDEs con una interfaz de usuario intuitiva y amigable.

Y quinto, criterios como la extensibilidad y la personalización, el rendimiento y la estabilidad, el soporte y la documentación, la integración con herramientas y ecosistemas, y el costo y el licenciamiento, también se consideraron durante el proceso de selección.

De las herramientas encontradas en la literatura, que cumplieron con todos estos criterios y por ende seleccionadas fueron: Visual Studio Code, Netbeans y Apple XCode (Apple Inc, s.f; Extensions, s.f y Netbeans, s.f).

2.3 *Tercero*, recopilación de los datos cualitativos y los cuantitativos aplicando las métricas que operacionalizan el modelo

La aplicación de las métricas a estas tres herramientas, se realizó con base en la aplicación de tres técnicas, a continuación se describe la técnica y los resultados obtenidos.

Técnica 1. La primera técnica consiste en obtener el valor de las métricas establecidas en el modelo de análisis, el cual cuenta con un total de 188 métricas. Cada métrica cuenta con una escala de 1 a 5 puntos, por lo que la puntuación máxima posible sería de 940 puntos (188 métricas x 5 puntos por métrica). Aplicando esta técnica, se procedió a sumar las

puntuaciones obtenidas por cada herramienta tipo IDE evaluada con base en los estándares de calidad ISO 25010 y ISO 25059, planteados en dicho modelo.

Técnica 2. La segunda técnica aplicada se denomina la técnica del 100% de cumplimiento. Esta técnica consiste en determinar, primero por cada sub-característica y luego por cada característica del modelo de análisis, cuál de las herramientas cumple con la puntuación máxima. Es decir, para que una herramienta tenga una sub-característica presente, debe obtener una puntuación de 5 puntos en cada una de las métricas correspondientes a esa sub-característica.

Técnica 3. Por último, se aplicó la tercera técnica llamada el 75% del cumplimiento. Esta técnica se asemeja a la del 100% del cumplimiento, pero con la diferencia de que, para que una sub característica o característica esté presente para una herramienta IDE, puede obtener 3, 4 o 5 puntos en sus métricas

2.4 *Cuarto*, análisis de los resultados para comparar el desempeño de cada tecnología, identificando fortalezas y debilidades.

Luego de aplicar las métricas a las tres herramientas, según las tres técnicas fueron las siguientes.

Técnica 1. Obsérvese en Figura 1 y Tabla 1 que la herramienta con la mayor puntuación es Apple Xcode, seguida por Visual Studio Code y, en tercer lugar, Netbean.

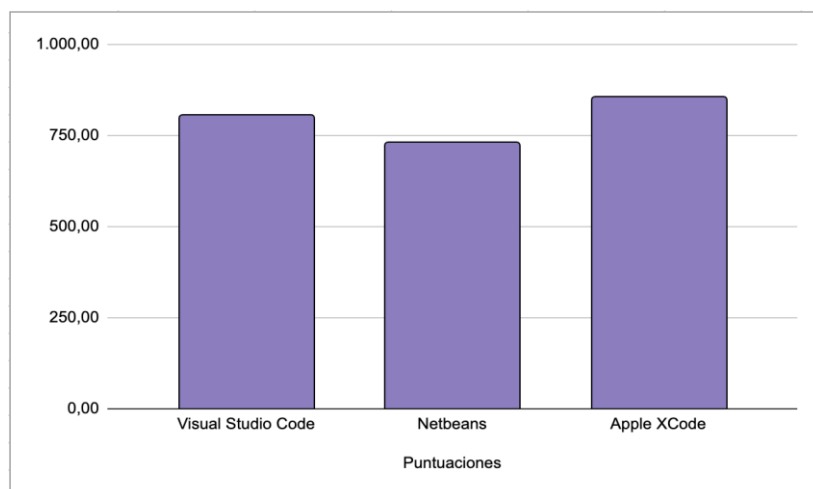


Figura 1. Resultado de las puntuaciones de cada una de las herramientas, basadas en la suma total. Fuente. Elaboración propia

Tabla 1: Puntaje de cada una de las herramientas, basadas en fuente propia

Puntuaciones	
Visual Studio Code	807,00
Netbeans	731,00
Apple XCode	855,00

Fuente. Elaboración propia

Después de observar la tabla 1, se destaca que Apple Xcode es la herramienta IDE más sobresaliente en comparación con las demás evaluadas. Xcode logró obtener las mejores puntuaciones en muchas de las métricas establecidas para las diferentes sub características evaluadas. En contraste, tanto Visual Studio Code como NetBeans presentaron debilidades notables en varias áreas, incluyendo la depuración de código, el análisis de interfaz e interoperabilidad. Estos puntos de debilidad indican áreas en las que estas herramientas podrían mejorar para igualar el rendimiento superior de Xcode.

Técnica 2. La Figura 2 y la Tabla 2 muestran los resultados.

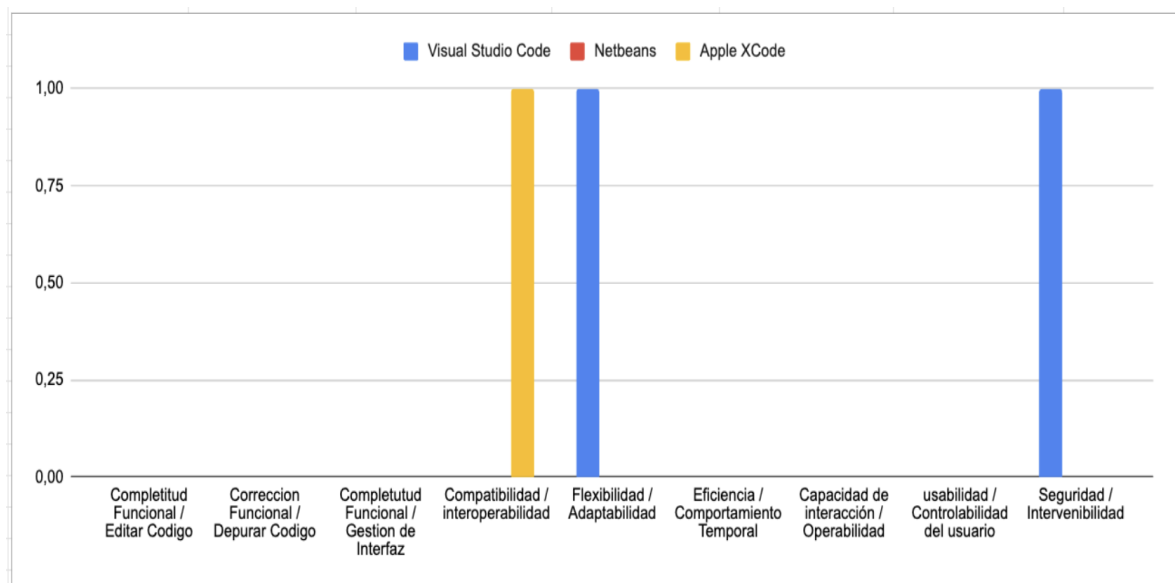


Figura 2. Distribución del Cumplimiento al 100% de las Herramientas IDE según las Normas ISO 25010 e ISO 25059. Fuente. Elaboración propia

Tabla 2: Evaluación de Cumplimiento al 100% de Características y Subcaracterísticas de las Normas ISO 25010 e ISO 25059 por Herramientas IDE

	Compleitud Funcional / Editar Código	Corrección Funcional / Depurar Código	Compleitud Funcional / Gestión de Interfaz	Compatibilidad / interoperabilidad	Flexibilidad / Adaptabilidad	Eficiencia / Comportamiento Temporal	Capacidad de interacción / Operabilidad	usabilidad / Controlabilidad del usuario	Seguridad / Intervenibilidad
Visual Studio Code	0	0	0	0	1	0	0	0	1
Nelbeans	0	0	0	0	0	0	0	0	0
Apple XCode	0	0	0	1	0	0	0	0	0

Fuente: Elaboración propia

Con esta técnica se encontró que la herramienta IDE Visual Studio Code cumple con la característica de flexibilidad, específicamente con la sub-característica de adaptabilidad y además cuenta en la parte de la IA con la característica denominada seguridad, especialmente la sub-característica intervenibilidad. Por otro lado, la herramienta IDE Apple Xcode cumple con la característica de compatibilidad, especialmente con la sub-característica de interoperabilidad.

Nótese que Visual Studio Code presenta debilidades en la sub característica de la completitud funcional en la edición de código a través de diversas métricas. Por otro lado, Visual Studio Code no cumplió con las características de usabilidad de su componente IA, siendo más específico en la sub-característica de controlabilidad del usuario. Finalmente, en lo que respecta a la sub característica de capacidad de interacción en la operabilidad, Visual Studio Code podría mejorar.

Por otro lado, con respecto a la herramienta de desarrollo de Apple, Xcode, cuenta con la característica de compatibilidad, especialmente con la sub-característica de interoperabilidad. Sin embargo, como toda herramienta, no es perfecta y, a pesar de contar con esa característica al 100%, estuvo cerca de obtener otras características, pero debe mejorar ciertos parámetros.

Además, estuvo cerca de obtener la característica de completitud funcional con la sub-característica de gestión de interfaz. Por otro lado, Apple XCode no cumplió con las características de usabilidad, siendo más específico en la sub-característica de controlabilidad del usuario. También presentó debilidades en la controlabilidad del componente de IA, en particular en la capacidad de los usuarios de intervenir y modificar el comportamiento del componente de IA. Por último, estuvo cerca de obtener la característica

de eficiencia, especialmente la sub-característica de comportamiento temporal más no logró el 100%.

La herramienta NetBeans no logró cumplir con el requisito de obtener una puntuación de 5 puntos en ninguna de las características evaluadas.

Técnica 3. La Figura 3 y Tabla 3, muestran los resultados.

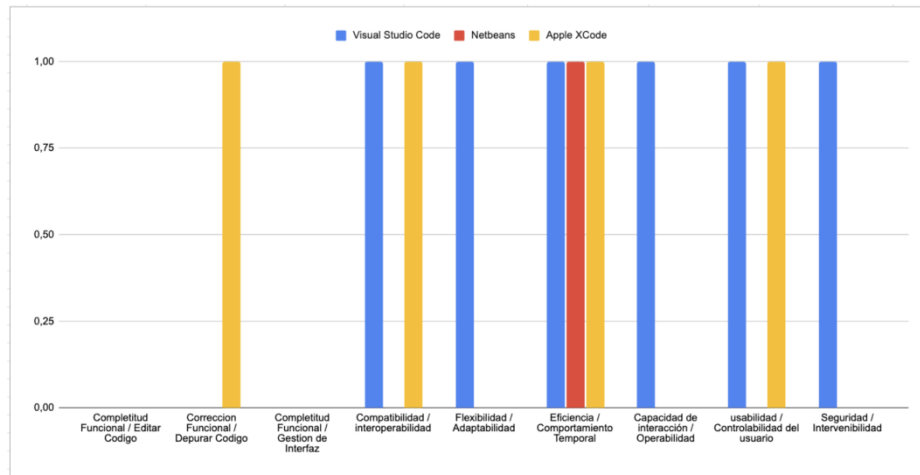


Figura 3. Distribución del Cumplimiento al 75% de las Herramientas IDE según las Normas ISO 25010 e ISO 25059. Fuente. Elaboración propia

Tabla 3: Evaluación de Cumplimiento al 75% de Características y Subcaracterísticas de las Normas ISO 25010 e ISO 25059 por Herramientas IDE

	Complejidad Funcional / Editar Código	Corrección Funcional / Depurar Código	Complejidad Funcional / Gestión de Interfaz	Compatibilidad / interoperabilidad	Flexibilidad / Adaptabilidad	Eficiencia / Comportamiento Temporal	Capacidad de Interacción / Operabilidad	usabilidad / Controlabilidad del usuario	Seguridad / Intervenibilidad
Visual Studio Code	0	0	0	1	1	1	1	1	1
NetBeans	0	0	0	0	0	1	0	0	0
Apple XCode	0	1	0	1	0	1	0	1	0

Fuente. Elaboración propia

Para esta técnica, la herramienta IDE Visual Studio Code logra cumplir con: la característica de flexibilidad, específicamente con la sub-característica de adaptabilidad, la característica de compatibilidad, especialmente con la sub-característica de interoperabilidad, la característica de eficiencia, específicamente con la sub-característica de comportamiento temporal, la característica de capacidad de interacción, específicamente con la sub-característica de operabilidad y con la característica asociada con IA de Usabilidad especialmente con la subcaracterística de Controlabilidad del usuario, además de lo ya logrado, con la técnica anterior. Los resultados del análisis de Visual Studio Code,

demuestra que esta herramienta está muy cerca de lograr todas las características propuestas, en el modelo propuesto.

Por otro lado, la herramienta Netbeans logra cumplir, bajo esta técnica, con la característica de eficiencia, específicamente con la sub-característica de comportamiento temporal.

Por último, la herramienta Apple Xcode cumple con la característica de corrección funcional, específicamente con la sub-característica de depurar código con la característica de compatibilidad, especialmente con la sub-característica de interoperabilidad, y la característica de eficiencia, específicamente con la sub-característica de comportamiento temporal. Finalmente, Apple XCode al igual que Visual Studio Code cumple con la característica asociada con IA de Usabilidad, especialmente con la sub característica de Controlabilidad del usuario.

2.5. *Quinto*, documentación del proceso y los resultados

Luego de aplicar el método de análisis de características a tres estudios de casos, se pudo constatar con precisión las debilidades y fortalezas de las tres herramientas IDEs seleccionadas. Intuitivamente se esperaba este resultado, pero gracias al modelo de análisis propuesto, se identifica con precisión dónde fallan y cual es su potencial, por ende, sus oportunidades de mejoras. Los tres casos de estudio analizados fueron: Apple Xcode, Visual Studio Code y NetBeans.

La primera técnica aplicada, que consistió en sumar las puntuaciones de 188 métricas establecidas, reveló que Apple Xcode es la herramienta más destacada, mientras que Visual Studio Code y NetBeans presentaron áreas de mejora en depuración de código, análisis de interfaz e interoperabilidad.

En cuanto a la segunda técnica del 100% de cumplimiento permitió identificar que Visual Studio Code sobresale en flexibilidad y seguridad en IA, aunque tiene debilidades en usabilidad y controlabilidad, mientras que Apple Xcode mostró compatibilidad e interoperabilidad, pero necesita mejorar en usabilidad y eficiencia. NetBeans, por su parte, no logró cumplir con ninguna característica al 100%.

Y por último, la tercera técnica del 75% de cumplimiento indicó que Visual Studio Code se acerca a cumplir con todas las características, NetBeans mejora en eficiencia, y Apple Xcode mantiene fortalezas en depuración y eficiencia.

Conclusiones

Entre las conclusiones, se presentan:

La selección de la herramienta IDE adecuada para el desarrollo de software es una decisión crítica que puede afectar significativamente la productividad y la calidad del producto final, por ello se aplica el método Análisis de Características por Estudio de casos.

Para identificar las características se formula un modelo de análisis de características, instanciado a partir de las normas ISO 25010 e ISO 25059.

El modelo propuesto contiene las características completitud funcional, corrección funcional, completitud funcional, flexibilidad, eficiencia, capacidad de interacción usabilidad y seguridad y se operacionaliza con un total de 188 métricas, formuladas según el método GQM.

El método Análisis de Características por Estudio de Caso se aplicó para tres estudios de caso: Apple Xcode, Visual Studio Code y NetBeans, se siguieron tres técnicas de estimación, con las métricas propuestas.

La evaluación de la calidad de las herramientas IDE Visual Studio Code, NetBeans y Apple Xcode mediante las tres técnicas revela que cada herramienta tiene fortalezas y debilidades particulares.

Apple Xcode sobresale en términos de puntuación total de métricas y muestra un cumplimiento sólido en compatibilidad especialmente en la sub-característica de interoperabilidad, aunque necesita mejoras en completitud funcional y seguridad especialmente en intervenibilidad sobre el componente de IA.

Visual Studio Code demuestra ser altamente flexible y eficiente, con buen cumplimiento en varias características bajo la técnica del 75% y en la del 100%, pero presenta áreas de mejora en depuración de código, comportamiento temporal y completitud Funcional específicamente en el aspecto de gestión de Interfaz.

NetBeans muestra la necesidad de mejoras significativas, especialmente en adecuación funcional, eficiencia y usabilidad y seguridad sobre el componente de la IA, aunque presenta

un cumplimiento razonable en eficiencia (comportamiento temporal) bajo la técnica del 75%.

Recomendaciones

Entre las recomendaciones, se propone:

Aplicar el método de análisis de características a una variedad más amplia de herramientas utilizadas en el enfoque DevSecOps. Esto permitirá evaluar su eficacia y precisión en diferentes entornos y con diversas tecnologías.

Aplicarlo además en distintos ámbitos y contextos organizacionales dentro del desarrollo de software, a fin de constatar su beneficio como una herramienta de apoyo en la toma de decisiones, respecto a la selección y evaluación de herramientas.

Referencias

Amazon. (2023). ¿Qué es un entorno de desarrollo integrado (IDE)? <https://aws.amazon.com/es/what-is/ide/#:~:text=El%20IDE%20permite%20a%20los,pruebas%20de%20integraci%C3%B3n%20m%C3%A1s%20complejas.>

Apple Inc. (s.f). Xcode. Apple Developer. Retrieved August 22, 2024, from <https://developer.apple.com/xcode/>

Cook, S; Samalam, A y Unewisse, M. (2021). A metrics framework to facilitate integration of disaggregated software development. Shoalgroup.com. <https://shoalgroup.com/wp-content/uploads/2021/07/Cook-Samalam-Unewisse-Metrics-to-Drive-Software-Integration-2021-Ver-1.pdf>

Estrada, F y Maya, E. (2016, marzo 6). Metodología GQM. SlideShare. <https://www.slideshare.net/slideshow/metodologa-gqm-59163173/59163173>

Extensions, L. M. A. (s.f). Visual Studio Code - code editing. Redefined. Visualstudio.com. Retrieved August 22, 2024, from <https://code.visualstudio.com/>

Finio, M., & Downie, A. (2024, octubre 7). IA en el desarrollo de software. Ibm.com. <https://www.ibm.com/es-es/think/topics/ai-in-software-development>

Genially. (2021, enero 14). MODELO GQM. <https://view.genially.com/5fff8bada3ac614547499ed4/horizontal-infographic-timeline-modelo-gqm>

Hernández, Y. J., Velasco-Elizondo, P., & Benítez-Guerrero, y. E. (2016). Evaluando Adecuación Funcional y Usabilidad en Herramientas de Composición desde la Perspectiva

del Usuario Final.
<https://pdfs.semanticscholar.org/31a9/3046d6ff53b975761ac911e1c4441e862a81.pdf>

ISO 25000. (2022). ISO/IEC 25059. <https://iso25000.com/index.php/normas-iso-25000/iso-25059>

ISO 25010. (2022). Iso25000.com. <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

Kitchenham, B., DESMET. A method for evaluating Software Engineering methods and tools. Computing & Control Engineering Journal: 8(3), 120 – 126 (1997)

Klaxoon. (2024). *¿Cómo elegir las herramientas adecuadas para la gestión de proyectos?* <https://klaxoon.com/es/insight/como-elegir-las-herramientas-adecuadas-para-la-gestion-de-proyectos>

López, C y Noguera, W. (2016). Evaluación De Calidad Externa Con La Norma Iso/Ec 25000 Al "Sistema De Gestión De Documentos Y Pqrd" De La Gobernación De Narino. <https://sired.udenar.edu.co/7685/1/91834.pdf>

Normas ISO. (2023, septiembre 24). ISO 25010: La norma ISO 25010 puede ayudarte a mejorar la calidad del software y la satisfacción de tus usuarios. Normasiso.org. <https://normasiso.org/norma-iso-25010/>

NetBeans, A. (s.f.). Apache NetBeans releases. Apache.org. Retrieved August 22, 2024, from <https://netbeans.apache.org/front/main/download/>

Orta, M. (2024, abril 20). *Los Mejores IDEs para PHP: Una Guía Completa para Desarrolladores.* Marco Orta; Marco Orta - Web Developer. <https://ortamarco.me/blog/mejores-ides-para-php-desarrolladores>

RedHat. (2023, julio 31). *¿Qué es y para qué sirve un IDE?* <https://www.redhat.com/es/topics/middleware/what-is-ide>

Red Hat. (2023). Entornos de Desarrollo Integrado (IDE): Flexibilidad y Adaptabilidad. Recuperado de <https://www.redhat.com/en/topics/ide>

Software Development. (2024). La Importancia de la Interoperabilidad en las Herramientas de Desarrollo. Recuperado de <https://www.softwaredevelopment.com/interoperability-ide>

Software Development. (2024, abril 25). *¿Qué sistemas de control de versiones proporcionan la integración más fluida con los IDE más populares?* LinkedIn.com. <https://es.linkedin.com/advice/1/what-version-control-systems-provide-most-octhc?lang=es>

Software Development. (2024, marzo 3). How can software product metrics help you make data-driven decisions? LinkedIn.com. Recuperado el 6 de agosto de 2024, de <https://www.linkedin.com/advice/1/how-can-software-product-metrics-help-you-ipxze>

SW Team. (2024, marzo 11) Ides: Qué Son, Para Qué Sirven Y Los Más Populares <https://www.swhosting.com/es/blog/ides-que-son-para-que-sirven-y-los-mas-populares>

Techopedia.com. (2023). Importancia del Desarrollo del Software en la Tecnología. <https://www.techopedia.com/>

Valdovinos, G. (2023, mayo 17). *Herramientas y metodologías ágiles en el desarrollo de software: cómo optimizar la productividad del equipo*. LinkedIn.com. <https://es.linkedin.com/pulse/herramientas-y-metodolog%C3%ADas-%C3%A1giles-en-el-desarrollo-de-valdovinos>

Conflicto de interés

Los autores de este manuscrito declaran no tener ningún conflicto de interés.

Declaración ética

Los autores declaran que el proceso de investigación que dio lugar al presente manuscrito se desarrolló siguiendo criterios éticos, por lo que fueron empleadas en forma racional y profesional las herramientas tecnológicas asociadas a la generación del conocimiento.

Copyright

La *Revista de la Universidad del Zulia* declara que reconoce los derechos de los autores de los trabajos originales que en ella se publican; dichos trabajos son propiedad intelectual de sus autores. Los autores preservan sus derechos de autoría y comparten sin propósitos comerciales, según la licencia adoptada por la revista

Licencia Creative Commons

Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-Compartir Igual 4.0 Internacional



REVISTA DE LA UNIVERSIDAD DEL ZULIA, Fundada el 31 de mayo de 1947

UNIVERSIDAD DEL ZULIA, Fundada el 11 de septiembre de 1891