

BÚSQUEDA DE LA CALIDAD EN LAS HERRAMIENTAS TIPO IDE HABILITADAS CON IA: MODELO DE DOMINIO

SEARCH FOR QUALITY IN AI-ENABLED IDE TOOLS: DOMAIN MODEL

GIOVANNA CIANFAGLIONE

cianfaglione.giovanna@correo.unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela)

FRANK CAICEDO

frank.caicedo@correo.unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela)

MARÍA PÉREZ

maperez@unimet.edu.ve
Universidad Metropolitana, Caracas (Venezuela)

DINARLE M ORTEGA

dortega@ucab.edu.ve
Universidad Católica Andrés Bello (Venezuela)

Resumen

El avance tecnológico ha impulsado el crecimiento 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. En este artículo, se especifica un modelo de dominio de herramientas tipo IDE habilitadas con IA considerando la norma ISO 25059 como base fundamental en la búsqueda de un modelo de estimación de calidad para este tipo de herramientas. Este modelo de dominio se construye mediante un análisis exhaustivo, revisión bibliográfica, estudio de los plugins más relevantes y utilizando el lenguaje estándar UML, en particular el del diagrama de clases, debido a su riqueza semántica para formularlo de manera precisa y comprensible. Es importante reconocer que este modelo puede presentar ciertas limitaciones asociadas a la información disponible, proporcionada por fabricantes y fuentes bibliográficas. Este modelo, proporciona una estructura estandarizada que facilita la comunicación y contribuye al éxito de un proyecto objetivo, al facilitar la comprensión de realidades complejas propias de la calidad en las herramientas



habilitadas por IA, especialmente en el contexto de las IDE.

Palabras Claves: DevSecOps, Entorno de desarrollo integrado (IDE), herramientas habilitadas IA, calidad, ISO 25059.

Abstract

Technological advancement has driven the growth of Artificial Intelligence (AI)-enabled Integrated Development Environment (IDE) software tools, offering developers unlimited opportunities to address issues, automate tasks, and enhance creativity across various domains, from data analysis to content generation. This article specifies a domain model for AI-enabled IDE tools considering ISO 25059 as a fundamental basis in the quest for a quality estimation model for such tools. This domain model is constructed through comprehensive analysis, literature review, examination of the most relevant plugins, and utilizing the standard UML language, particularly that of the class diagram, due to its semantic richness for formulating it precisely and understandably. It's important to acknowledge that this model may have certain limitations associated with available information provided by manufacturers and bibliographic sources. Nonetheless, this model provides a standardized structure that facilitates communication and contributes to the success of a target project by aiding in the understanding of complex realities inherent in AI-enabled tools' quality, especially in the context of IDEs.

Key Words: DevSecOps, Integrated Development Environment (IDE), AI-enabled tools, quality, ISO 25059.

RECIBIDO: 09-01-2024 / ACEPTADO: 11-07-2024 / PUBLICADO: 30-09-2024

Cómo citar: Cianfagione et al. (2024). Búsqueda de la calidad en las herramientas tipo IDE habilitadas con IA: Modelo de dominio. *Anales*, 40, 15 - 32. <https://doi.org/10.58479/acbf.2024.111>

CONTENIDO

Resumen	15
Abstract	16
1. Introducción	19
2. Contexto	19
2.1 Proceso de Desarrollo de Software	20
2.2 DevSecOps	20
2.3 IDE	21
2.4 Herramientas Habilitados por IA	21
2.5 ISO 25010 e ISO 25059	22
3. Modelo de Dominio	23
4. Plug - Ins	26
5. Conclusiones	28
6. Recomendaciones	28
7. Referencias Bibliográficas	29

1. 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, se considera un aspecto importante a tomar en cuenta (Techopedia, 2023). Desde el funcionamiento de los sistemas críticos hasta el entretenimiento personal, la confianza en la precisión, seguridad y eficacia del software, es crucial. En este sentido, este artículo tiene como objetivo proponer un modelo de dominio del contexto de los IDE (Entorno de Desarrollo Integrado) habilitados por IA, considerando la norma ISO 25059, respectivamente (ISO 25000, 2022). La calidad es un aspecto complejo puesto que el modelo de dominio promueve su comprensión. Ante situaciones complejas, los ingenieros de software necesitan usar algunas técnicas que les ayuden a entender mejor la problemática que están analizando antes de iniciar con la implementación del software. En este caso, el uso del modelo de dominio sirve para identificar y representar los conceptos del dominio de problema y apoyar así, en su análisis (Jardiel, 2015).

Por ello este artículo se divide en tres secciones, además de la Introducción, Conclusiones y Recomendaciones. Se comienza con la sección Contexto en la que se abordan las definiciones cruciales para la propuesta y creación del modelo de dominio, este se encuentra en la segunda sección, cómo se modela cada una de las relaciones de las definiciones representadas con un diagrama de clases cumpliendo las normas del lenguaje UML y cómo cada una de ellas es crucial para el desarrollo de un producto de software de calidad. En la siguiente sección se reflexiona acerca de lo encontrado en la literatura sobre los plugins de IA que existen hoy día en el mercado para la codificación, integración u otras funciones que puedan realizar las herramientas IDE y sean de gran ayuda para los programadores en la actualidad. Esta sección incluye una tabla que contiene cada plugin con sus características más relevantes y cómo cada una de estas se relacionan con la normas ISO 25059 según las características y subcaracterísticas propuestas por esta norma.

Esta es una investigación en progreso que busca formular un modelo para estimar la calidad para este tipo de herramientas habilitadas con IA y así establecer y prescribir las características y subcaracterísticas que deben estar presentes en dichos sistemas. Contar con un modelo de calidad facilita criterios técnicos que proporcionan orientaciones y referencias a los interesados, al momento de seleccionarlas.

2. Contexto

Cuatro son los tópicos analizados para reflexionar sobre la realidad de la calidad en las herramientas de software tipo IDE habilitadas con IA.

2.1 Proceso de Desarrollo de Software

Con base en los autores ITE Corp (2022) y Prieto (2023), el proceso de desarrollo de software es un conjunto de pasos o etapas que se siguen para crear un producto de software, tales como recopilación, análisis y especificación de requisitos; diseño y desarrollo; pruebas, instalación y despliegue; mantenimiento y soporte. El seguimiento de éstos permite desarrollar programas de software eficientes, seguros y útiles para los usuarios e implica planificar, realizar y gestionar eficientemente un proyecto para ejecutarlo con éxito y lograr que cumpla con el objetivo para el cual fue diseñado. Por otro lado, IBM (s/f) también sostiene que “El desarrollo de software se refiere a un conjunto de actividades informáticas dedicadas al proceso de creación, diseño, despliegue y compatibilidad del software.”

Se concluye entonces, que el proceso de desarrollo de software es un proceso estructurado y ordenado de actividades que consta de varias etapas o pasos, las cuales abarcan desde su concepción hasta el mantenimiento del mismo. En otras palabras, se trata de un conjunto de actividades interrelacionadas y con un orden acordado que tienen como objetivo final la creación de un producto de software que satisfaga las necesidades de los usuarios. Por último, con referencia a la eficiencia, seguridad y utilidad del software, los cuales son claves para el desarrollo del mismo, implican una adecuada planificación, ejecución y gestión del proyecto para garantizar que cumpla con los objetivos establecidos y sea beneficioso para los usuarios (Maida y Paciencia, 2015).

2.2 DevSecOps

Dynatrace (2024), indica que “DevSecOps es un marco de colaboración que amplía el impacto de DevOps al agregar prácticas de seguridad al proceso de desarrollo y entrega de software. Éste resuelve la tensión entre los equipos de DevOps que desean lanzar software rápidamente y los equipos de seguridad que priorizan la seguridad por encima de todo”. Además señala que al integrar principios y prácticas de seguridad de aplicaciones en el desarrollo y operaciones de software, esto le permite a los equipos poder ofrecer nuevos servicios y desarrollo de software, a una velocidad mucho mayor sin comprometer la seguridad de las aplicaciones (Dynatrace, 2024). También, RedHat (2023) señala que “DevSecOps significa desarrollo, seguridad y operaciones, definido como un enfoque que aborda la cultura, la automatización y el diseño de plataformas, e integra la seguridad, como una responsabilidad compartida durante todo el ciclo de vida de la TI”.

Por otro lado, según AWS (2023), “DevSecOps es la práctica de integrar las pruebas de seguridad en cada etapa del proceso de desarrollo de software. Esta incluye herramientas y procesos que fomentan la colaboración entre los desarrolladores, los especialistas en seguridad y los equipos de operaciones para crear un software que sea eficiente y seguro”. El objetivo de éste, se centra en ayudar a los equipos de desarrollo a abordar los problemas de seguridad de forma eficaz, de esta manera contribuye a una alternativa para suplantar a las antiguas prácticas de seguridad del software que no se ajustaban a las rápidas actualizaciones y a los ritmos de plazos más ajustados. De esta manera DevSecOps se convirtió en una

transformación cultural que hace de la seguridad una responsabilidad compartida para todos los desarrolladores de software (AWS, 2023).

Nótese entonces que DevSecOps es una filosofía, un marco de trabajo y un conjunto de prácticas que busca mejorar la colaboración y la comunicación entre los equipos de desarrollo de software y operaciones y seguridad de TI. Su objetivo principal es acortar el ciclo de vida del desarrollo de software, agilizar la entrega de aplicaciones y mejorar la calidad del producto final, además el uso de estas prácticas se basa en tres pilares fundamentales personas, procesos y tecnología con el fin de desarrollar y entregar softwares de forma más rápida, fiable y eficiente. En otro sentido, DevSecOps, según Ingenio (2023) se basa en estos cuatro principios: es una cultura y filosofía, automatiza todo el proceso de desarrollo, usa el desarrollo de enfoque Lean (Tecnologías-información, 2018), por último busca medir todo el proceso y compartir. A su vez, contempla este conjunto de etapas: prueba, planificación, entrega, monitoreo, configuración, retroalimentación, operar, construir, desplegar y codificar. Entre las herramientas a utilizar en el proceso de desarrollo, en particular en la etapa “code” y “build” de DevSecOps destacan las herramientas IDE.

2.3 IDE

AWS (2023), afirma que “Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar software en una aplicación fácil de usar. Así como los escritores utilizan editores de texto y los contables, hojas de cálculo, los desarrolladores de software utilizan IDE para facilitar su trabajo”. Por otro lado, RedHat (2013) y Datascientest (2022), definen una IDE como un sistema de software integrado para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI).

Por lo que, una IDE se puede definir como una aplicación de software que proporciona un conjunto de herramientas para facilitar el desarrollo de software que ayuda a los programadores a realizar su código más eficiente. Su objetivo principal es aumentar la productividad de los programadores al integrar en una sola interfaz gráfica de usuario (GUI) que combina diversas capacidades, como la edición, creación, prueba y empaquetado de software, en una sola aplicación fácil de usar. Es decir, la misma es una herramienta esencial para cualquier programador que busca aumentar su productividad y la mejora de la calidad de su trabajo; las cuales pueden tener integrado algún tipo de componente del tipo IA.

2.4 Herramientas Habilitados por IA

RedHat (2024), indica que “la inteligencia artificial (IA) se refiere en general a los procesos de Ciencias de la Computación y los algoritmos estadísticos que son capaces de simular y mejorar la inteligencia humana. En otras palabras, la IA describe sistemas capaces de adquirir conocimiento y aplicar información para resolver problemas”. A su vez, las herramientas habilitadas con IA, son sistemas que incluyen datos y componentes que implementan algoritmos

que imitan el aprendizaje y resolución de problemas, tienen características inherentemente diferentes a los sistemas de software solos (Horneman, et al, 2019).

Los eventos que registramos pueden estar relacionados con la seguridad, entonces un sistema habilitado de IA combina los conceptos de sistemas habilitados, con los procesos de Ciencias de la Computación y algoritmos estadísticos de la Inteligencia Artificial. A su vez, esta infraestructura tecnológica está configurada para adquirir, procesar y analizar datos con el fin de tomar decisiones inteligentes, realizar predicciones y automatizar las tareas (Roymo, 2023). Dicho esto se puede decir que un sistema habilitado de Inteligencia Artificial (IA) (Sistema Enable IA) se refiere a una infraestructura tecnológica configurada con los componentes, software y recursos necesarios para aprovechar y utilizar capacidades de IA en sus funciones y operaciones. Estos sistemas están diseñados para aplicar algoritmos y técnicas de IA con el objetivo de simular y mejorar la inteligencia humana, adquiriendo conocimiento y aplicando información para resolver problemas y tratar de asegurar la calidad del software.

2.5 ISO 25010 e ISO 25059

Agencia (2021), afirma que “la calidad de software está en relación directa con el cumplimiento de los requerimientos formulados por el usuario, de tal forma que si un programa no cumple con alguno de éstos, es un software de baja calidad”. Por otro lado, Pérez (2019), afirma que “La calidad en el desarrollo de software es un aspecto crucial para cualquier proyecto de software exitoso. El objetivo de cualquier proyecto es garantizar la calidad de los resultados y satisfacer las necesidades del cliente y de las demás partes interesadas. Para lograr esto, es importante planificar la calidad de la entrega y del proceso.”

En conclusión, se considera la calidad de software como un concepto fundamental para el éxito de cualquier proyecto de desarrollo de software. Se refiere al grado en que el software cumple con los requisitos establecidos por el cliente y las demás partes interesadas, y a su capacidad para satisfacer sus necesidades (Rodríguez, 2016). A su vez, la calidad del software es un factor esencial para el éxito de cualquier proyecto de desarrollo de software, mediante el cumplimiento de los estándares de la ISO 25010 e ISO 25059 (ISO 25000, 2022).

Según la página de ISO 25010 (2022) el estándar ISO 25010, representa “un modelo de calidad, el cual es fundamental para la evaluación del producto debido a que establece el sistema central en el que se basa”. También señalan que “en este modelo se determinan las características de calidad que se consideran a la hora de evaluar las propiedades de un producto de software determinado cuya calidad se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios, aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.”

Por otro lado, Ormeño (2019), indica que “ISO 25010 es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software”, compuesto de 8 características y 31 subcaracterísticas que se relacionan con las

propiedades estáticas del software y las propiedades dinámicas del sistema informático. El modelo es aplicable tanto a los sistemas informáticos como a los productos de software”.

Por otra parte, según la página de ISO 25000 (2022), el estándar ISO 25059 “es una extensión del modelo de calidad de la norma ISO/IEC 25010 que se enfoca en abordar los aspectos adicionales que se presentan en los sistemas de IA. A medida que las organizaciones incorporan cada vez más sistemas de IA, se ha vuelto necesario adaptar el modelo de calidad existente incorporando las características específicas de estos sistemas”. Según, ITeh Standard Preview (2023), este estándar es “un documento que describe un modelo de calidad para sistemas de IA y es una extensión específica de la aplicación de los estándares SQuaRE. Las características y subcaracterísticas detalladas en el modelo proporcionan un lenguaje y terminología consistentes para especificar, medir y evaluar la calidad de los sistemas de IA” y se utilizan como un conjunto de criterios de calidad con los cuales se pueden comparar los requisitos de calidad establecidos para determinar su completitud (Iso/iec 25059:2023, 2023).

Se concluye entonces que las normas ISO 25010 e ISO 25059 desempeñan un papel fundamental en la formulación de un modelo de calidad para las herramientas habilitadas por IA, el cual establecerá y prescribirá las características y subcaracterísticas que deben estar presentes en herramientas habilitadas por IA. La norma ISO 25050, cuenta con nuevas características y cada una de ellas cuenta con subcaracterísticas. Se tiene como primera característica la Funcionalidad, la cual tiene como subcaracterísticas adaptabilidad, pertinencia funcional, corrección funcional y completitud funcional. La característica de Seguridad, cuenta con intervisibilidad, no repudio, autenticidad, responsabilidad, integridad y confidencialidad. La Fiabilidad cuenta con 5 subcaracterísticas las cuales son: capacidad de recuperación, robustez, madurez, disponibilidad y tolerancia a fallos (Iso/iec 25059:2023, 2023).

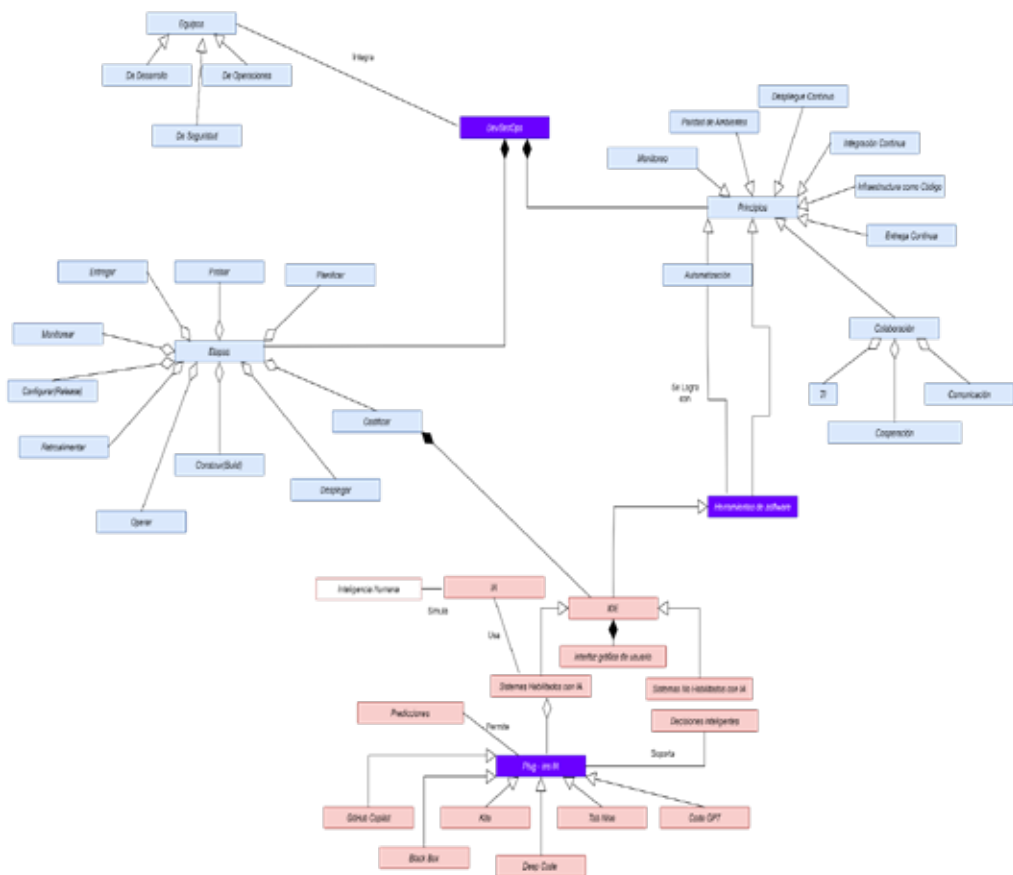
A su vez, la Capacidad de Interacción cuenta con reconocibilidad de la adecuación, operatividad, accesibilidad, protección frente errores de usuarios, estética, transparencia y controlabilidad del usuario. La Protección cuenta con cinco subcaracterísticas, las cuales son: restricción operativa, identificación de riesgos, protección ante fallos, advertencia de peligro e integración segura. Se cuenta con otra característica la cual es Mantenibilidad, la cual; cuenta con capacidad de ser probado, capacidad de análisis, confidencialidad, modularidad y capacidad de ser modificado. También se tiene la Portabilidad que como subcaracterística tiene adaptabilidad, facilidad de instalación, capacidad de ser reemplazado y escalabilidad. La Eficiencia es otra característica que cuenta con comportamiento temporal, utilización de recursos y capacidad. Como última característica, se tiene la Compatibilidad que cuenta con la interoperabilidad y la coexistencia. Estas características y subcaracterísticas proporcionan un marco de evaluación completo para medir y evaluar la calidad de un producto de software, permitiendo asegurar que cumple con los requisitos establecidos (ISO 25000, 2022 y ISO 25010, 2022).

De acuerdo a lo planteado anteriormente, los tópicos explicados pudieran ser presentados como un modelo de dominio expresado visualmente con un diagrama de clases de UML.

3. Modelo de Dominio

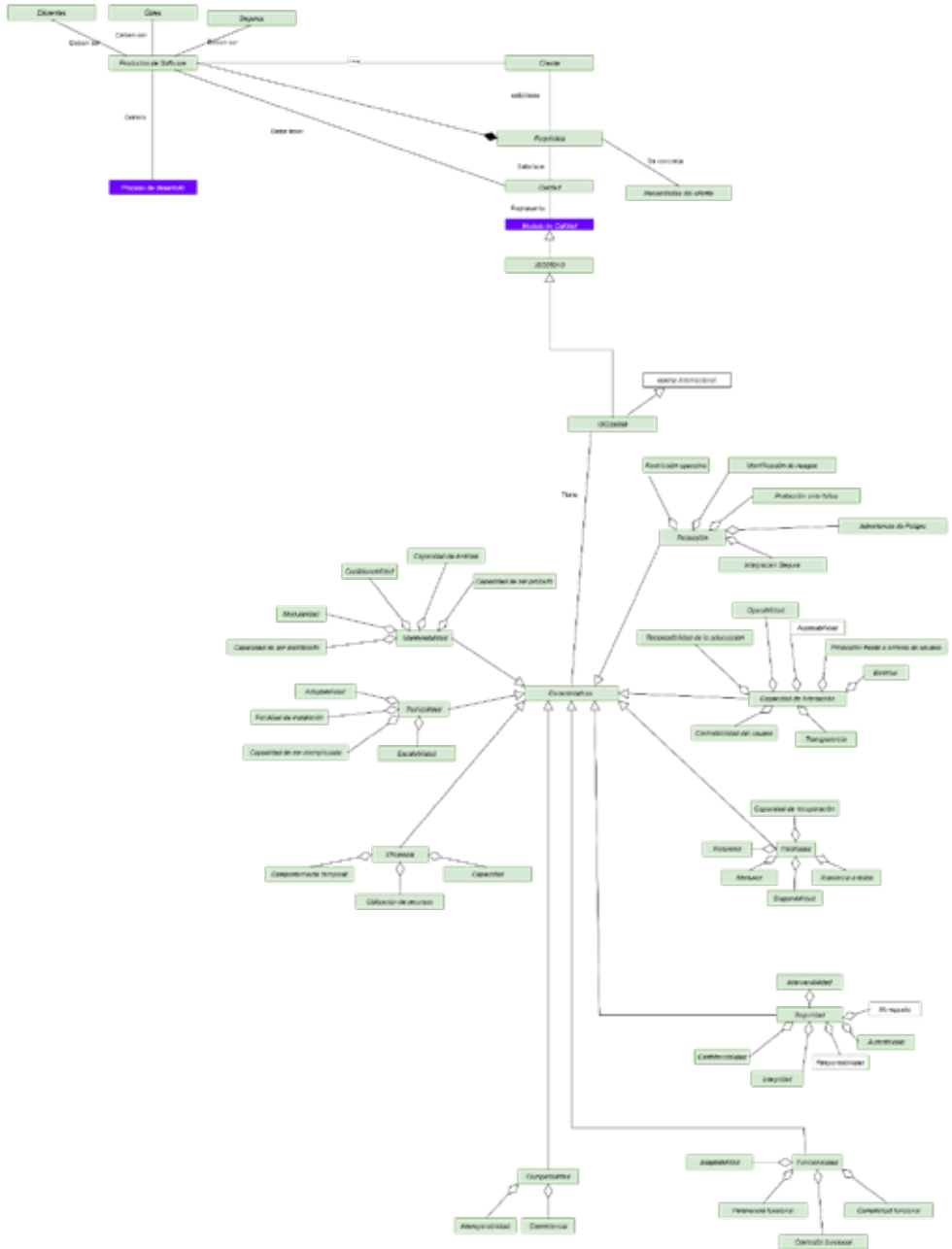
Por consiguiente, se puede apreciar a continuación en el Modelo de Dominio propuesto y elaborado con un Diagrama de Clase expresado en el lenguaje estándar UML, que él se destacan tres mundos como son DevSecOps, ISO y Modelos de calidad, y Herramientas habilitadas con IA-Plugins. A continuación, se muestran esos tres mundos, identificándose en ellos, los conceptos revisados en la literatura y las relaciones propuestas; obsérvese que se identificaron además los conceptos que integran estos tres mundos (las clases de color morado oscuro) (Figuras 1, 2 y 3). Estas clases vinculan los tres mundos descritos.

Figura 1: Primer mundo DevSecOps



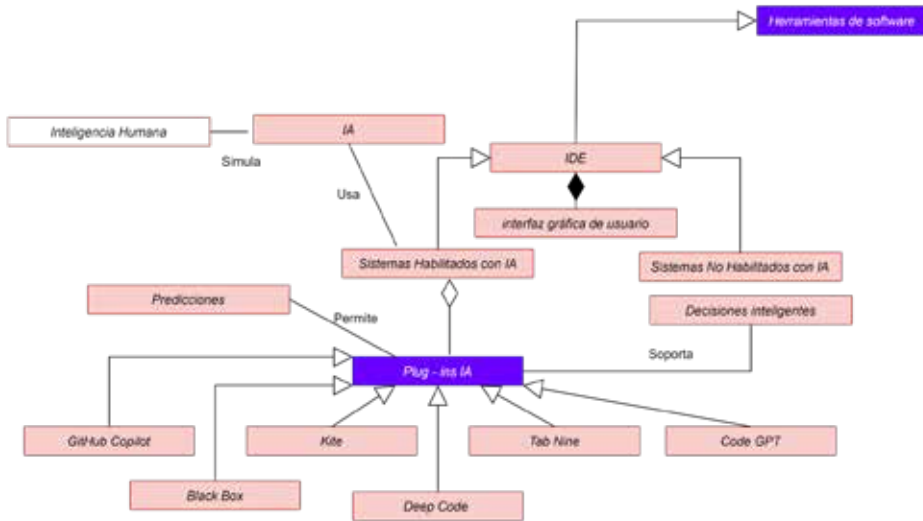
Los conceptos vinculantes encontrados aquí son: DevSecOps y Herramientas de Software, a través del principio Automatización.

Figura 2: Segundo Mundo ISO y Modelos de Calidad



Aquí los conceptos vinculantes de este mundo con DevSecOps son: Proceso de Desarrollo y Modelo de Calidad, dado que DevSecOps es una filosofía sobre el proceso de desarrollo.

Figura 3: Tercer Mundo Herramienta Habilitadas con IA-Plugins



Este mundo se vincula directamente con DevSecOps a través del concepto Herramientas de Software.

El modelo de dominio, es entonces la integración de estos tres mundos, dada su complejidad, se presentó en tres partes; sin embargo, gracias a él se recibe la complejidad del tema tratado, y ya se identifican relaciones que permiten precisar las características y subcaracterísticas que deberían estar presente en las herramientas de software habilitadas con IA tipo IDE. Una pieza fundamental en este tipo de herramientas son los *Plug-ins* que habilitan con IA.

4. Plug - Ins

Según Correa (2024), los *plug-ins* de IA para IDEs se han convertido en herramientas cada vez más importantes para los programadores. Estos *plug-ins* aprovechan el poder de la IA para mejorar la productividad del desarrollador, la calidad del código, la seguridad y la experiencia general de desarrollo. Por ende, en el ámbito del desarrollo de productos de software destacan distintos *plug-ins* que son los más comunes, que se utilizan actualmente y los que interoperan con distintas IDEs como Visual Studio Code, Eclipse, Sublime Text y Atom.

En referencia a la ayuda de la codificación y desarrollo más eficiente del código por parte de los programadores destacan: Github Copilot según Fernandez (2023), es un sistema de ayuda creado por GitHub que te permite escribir el código en tiempo real, ya que a medida que el programador realiza el código GitHub Copilot le sugiere el código referente a lo que se esté desarrollando. Por otro lado, Fernandez (2024), define a Black Box como una herramienta de IA que ayuda a encontrar y usar fragmentos de código de manera eficiente. De esta manera, mientras el programador esté codificando tendrá sugerencias para poder completar tareas que se estén realizando.

A su vez, Fernandez et al. (2019) sostiene que Kite es otro *plug-in* que destaca en el proceso de codificación, ya que es un complemento de IA que se coloca en el editor de código el cual tiene la función de autocompletado como los que se mencionaron anteriormente. Adicionalmente, TabNine al igual que GitHub Copilot y Kite funciona generando código u ofreciendo sugerencias de código en tiempo real, realizando un análisis del contexto del código y proporcionando el completado automático.

Para cerrar, es de suma importancia el análisis de la seguridad, errores y vulnerabilidades que puedan presentarse en el transcurso del desarrollo, ahí es cuando, Deep Code según Correa (2024) destaca, ya que permite analizar el código para mejorar la calidad y seguridad del software realizando un análisis exhaustivo utilizando técnicas de aprendizaje continuo para detectar vulnerabilidades, errores o malas prácticas del código. Finalmente, Stork (2024) afirma que Code GPT tiene como objetivo potenciar la habilidad para codificar y agilizar el flujo de trabajo, realizando preguntas sobre problemas con el código que se esté trabajando.

El siguiente link muestra un resumen de los plugins hasta ahora analizados, en la búsqueda de cuáles son las características y subcaracterísticas según ISO 29059 que están presentes en ellos.

<https://doi.org/10.6084/m9.figshare.25648779.v2>

Después de analizar la Tabla, se observa una tendencia en la presencia de la sub-característica de Interoperabilidad, dentro de la característica de compatibilidad, en la mayoría de los *plugins* evaluados, como Github Copilot, Kite, DeepCode, TabNine y Code GPT. Sin embargo, el único *plugin* que no presenta esta sub-característica es Black Box. Además, se observa que para la característica de Funcionalidad cada uno de ellos tiene diferentes subcaracterísticas asociadas. Los plugins Github Copilot, Kite, Code GPT y TabNine cuentan con la sub-característica de Pertinencia Funcional dentro de la característica de Funcionalidad. Por otro lado, Black Box, Kite, DeepCode, Code GPT y TabNine tienen la sub-característica de Adaptabilidad Funcional. El plugin Code GPT y Kite también presentan la sub-característica de Corrección Funcional.

En cuanto a la Capacidad de Interacción, se observa: Kite y TabNine tienen la sub-característica de Reconocibilidad de la Adecuación, mientras que DeepCode cuenta con las subcaracterísticas de Operabilidad y Controlabilidad del usuario. Por último, la característica de Fiabilidad está presente únicamente en el *plugin* Code GPT. Con apoyo a esto se puede decir

que los *plugins* evaluados difieren en cuanto a las subcaracterísticas presentes dentro de cada una de las características evaluadas, lo que evidencia las distintas capacidades y enfoques de cada uno de ellos.

5. Conclusiones

Entre las conclusiones, se presentan:

Se hizo una revisión bibliográfica del contexto relacionado con las herramientas habilitadas con IA tipo IDE, del proceso de desarrollo DevSecOps, tomando en cuenta cual puede ser el aporte de un modelo de calidad inspirado en las normas ISO 25059 para la estimación de sus calidad.

El desarrollo de software es un proceso complejo que exige la colaboración de diversos equipos y la integración de múltiples prácticas. En este contexto, la adopción de herramientas habilitadas con IA en el marco de DeSecOps emerge como una estrategia prometedora para optimizar la eficiencia y la calidad del software desarrollado. DevSecOps, como filosofía, integra la seguridad en el ciclo de vida del software, promoviendo la colaboración entre equipos de desarrollo, operación y seguridad. Las herramientas habilitadas con IA, por su parte, ofrecen un abanico de posibilidades para automatizar tareas, mejorar la eficiencia y optimizar la calidad del software. La norma ISO 25059 proporciona un marco de referencia sólido para evaluar la calidad de herramientas habilitadas con IA tipo IDE.

Se propone un modelo de dominio elaborado con el lenguaje UML según la notación de diagramas de clase, el cual permite interpretar analizar y reflexionar sobre el contexto de las herramientas habilitadas con IA tipo IDE bajo la filosofía de DevSecOps en la búsqueda de un modelo de para estimar la calidad que de este tipo de herramientas habilitadas con IA, así como lo son las IDEs.

El análisis de la tabla de plugins de IA revela una variedad de capacidades y enfoques. La interoperabilidad está presente en la mayoría de los plugins, permite su integración con otras herramientas de desarrollo. La funcionalidad, ofrece subcaracterísticas como pertinencia funcional, adaptabilidad funcional y corrección funcional . La capacidad de interacción también presente, facilita el reconocimiento de la adecuación de las sugerencias, así como su operabilidad y controlabilidad del usuario. Finalmente, la fiabilidad, ofrece características que garantizan la confiabilidad de las sugerencias. La elección del *plug-in* ideal dependerá de las necesidades específicas del proyecto, considerando las diversas capacidades y enfoques que cada uno ofrece.

6. Recomendaciones

Entre las recomendaciones, se proponen las siguientes:

Realizar una revisión más exhaustiva de los *plug-ins* y de las herramientas habilitadas con IA tipo IDE a fin de profundizar y proponer un modelo de estimación de la calidad para ellas.

Operacionalizar a través de la propuesta de un conjunto de métricas que permitan al desarrollador evaluar las diferentes herramientas en función de las características y subcaracterísticas presentes en este modelo de calidad.

7. Referencias Bibliográficas

- AWS (2023). ¿Qué es DevSecOps? <https://aws.amazon.com/es/what-is/devsecops/#:~:text=building%20the%20software.-.What%20does%20DevSecOps%20stand%20for%3F,they%20are%20building%20software%20applications.>
- AWS. (2023). ¿Qué es un entorno de desarrollo integrado (IDE)?. Amazon.com. <https://aws.amazon.com/es/what-is/ide/>
- Agencia, F. (2021, August 1). Calidad del Software. Salud Electrónica. <https://saludelectronica.com/calidad-del-software/>
- Correa, J. (2024, abril 15). AI for developers: Guía completa de AI para desarrolladores de software. Developer Blog. <https://develohero.io/ai-for-developers>
- Datascientest. (2022, septiembre 2). IDE : ¿Qué es un Entorno de Desarrollo Integrado? Formación en ciencia
- Dynatrace. (27, marzo 2024). DevSecOps—Development, security and operations https://www.dynatrace.com/monitoring/solutions/devsecops-1/?utm_source=google&utm_medium=cpc&utm_term=devsecops&utm_campaign=us-appsec-application-security&utm_content=none&utm_campaign_id=11810003001&gclid=Cj0KCCQjwztOwBhD7ARIsAPDKnkChL4ZftHqjOpxPwYUoA5pBUaLT4EGlPRYbsLjsXFp-BNOMQWqfh7caAmRVEALw_wcB&mrasn=1160682.1439354.4uUpreyi
- Fernandez, E., Pardo, L., & Ferzzola, M. (2019, February 8). Kite: Autocompletado de código basado en AI. Neoteo.com. <https://www.neoteo.com/kite-autocompletado-de-codigo-basado-en-ai/>
- Fernández, Y. (2024, March 27). Blackbox AI: qué es y qué herramientas ofrece esta inteligencia artificial que te ayuda a programar completando código. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/blackbox-que-que-herramientas-ofrece-esta-inteligencia-artificial-que-te-ayuda-a-programar-completando-codigo>
- Fernández, Y. (2023, February 16). Qué es Copilot de GitHub y cómo funciona esta inteligencia artificial que te ayuda a programar. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/que-copilot-github-como-funciona-esta-inteligencia-artificial-que-te-ayuda-a-programar>
- Horneman, A, Mellinger A and Ozkaya I. (2019). AI Engineering: 11 Foundational Practices. Pittsburgh: Carnegie Mellon University Software Engineering Institute, 2019. https://resources.sei.cmu.edu/asset_files/WhitePaper/2019_019_001_634648.pdf

- IBM.com. (s/f) ¿Qué es el desarrollo de software?. <https://www.ibm.com/es-es/topics/software-development>
- Ingenio, J. (2023). Software Architects Handbook: Become a successful software architect by implementing effective architecture concepts. Editorial Packt.
- 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>
- Iso/iec 25059:2023. (2023). ISO. <https://www.iso.org/standard/80655.html>
- ITeH Standard Preview (2023). Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality model for AI systems. <https://cdn.standards.iteh.ai/samples/80655/168addf09e0a4d8181b9172dc7404fab/ISO-IEC-25059-2023.pdf>
- Jardiel, L. (2015, enero 27). El Modelo de Dominio. Blogspot.com. <https://modeladodesoftware.blogspot.com/2015/01/9-el-modelo-de-dominio.html>
- Maida, E y Paciencia, J. (2015). Metodologías de desarrollo de software. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- ITE Soluciones by ITE Corp. (2022, noviembre 9). ¿Cuál es el proceso de desarrollo de software? LinkedIn.com. <https://es.linkedin.com/pulse/cu%C3%A1-es-el-proceso-de-desarrollo-software-itesolucionesmx>
- Ormeño, N. (2019, mayo 15). ISO 25010 y el Desarrollo de Software. <https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341>
- Pérez, J. B. (2019). 3digits – Tenemos una solución – Tecnologías de la Información. <https://www.3digits.es/blog/gestion-de-la-calidad-en-el-desarrollo-de-software.html>
- Prieto, E. (2023, noviembre 16). ¿Cuáles Son Las Etapas Del Desarrollo De Software?. <https://global.tiffin.edu/noticias/cuales-son-las-etapas-del-desarrollo-de-software>
- Red Hat (2024). Inteligencia artificial open source a su favor. <https://www.redhat.com/es/products/ai>
- Redhat.com. (2013, julio 31). ¿Qué es y para qué sirve un IDE?. <https://www.redhat.com/es/topics/middleware/what-is-ide>
- RedHat.com. (2023, marzo 15). ¿Qué es DevSecOps?. <https://www.redhat.com/es/topics/devops/what-is-devsecops>
- Rodríguez, A. (2016, marzo 23). ¿Cómo se evalúa la calidad de una IDE?. Blogspot.com. <https://blog-idee.blogspot.com/2016/03/como-se-evalua-la-calidad-de-una-ide.html>

Roymo, D. (2023, julio 31). La IA y la automatización de procesos: aumenta eficiencia y reduce costes. GAMCO, SL. <https://gamco.es/ia-automatizacion-de-procesos-eficiencia-reducir-costes/> <https://www.stork.ai/es/ai-tools/codegpt-efbb2>

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

Tecnologías-información. (2018) Método Lean de desarrollo de Software. <https://www.tecnologias-informacion.com/metodo-lean.html>

