

Evaluación de la mantenibilidad de los sistemas híbridos (clásico-cuánticos)

Ana Díaz Muñoz^{1,2}, Moisés Rodríguez Monje^{1,2},
José A. Cruz-Lemus², Mario Piattini Velthuis²

¹ AQCLab Software Quality 13500 Puertollano (Ciudad Real), España

²Instituto de Tecnologías y Sistemas de Información y Escuela Superior de Informática,
Universidad de Castilla-La Mancha 13071 Ciudad Real, España

{adiaz, mrodriguez}@aqclab.es,
{joseantonio.cruz, mario.piattini}@uclm.es

Abstract. *The remarkable development of quantum computing arose in response to the need to address problems intractable for conventional computers. However, it is essential to emphasize that quantum computing will not replace traditional computing but will coexist in hybrid systems. Similar to the scenario observed in the conventional software domain, evaluating the quality of these emerging hybrid systems becomes mandatory, with a particular focus on their maintainability to ensure their scalability. In this paper, we introduce a pioneering methodological and technological framework comprising a set of properties and metrics and a set of automated tools designed for hybrid software maintainability.*

Resumen. *El notable desarrollo de la computación cuántica surge en respuesta a la necesidad de abordar problemas que resultan intratables para las computadoras convencionales. No obstante, es fundamental destacar que la computación cuántica no reemplazará a la computación tradicional, sino que ambas coexistirán en sistemas híbridos. Similar al escenario observado en el ámbito del software convencional, se vuelve obligatorio evaluar la calidad de estos sistemas híbridos emergentes, con un enfoque particular en su mantenibilidad para asegurar su escalabilidad. En el presente trabajo, introducimos un marco metodológico y tecnológico pionero que comprende un conjunto de propiedades y métricas, así como una serie de herramientas automatizadas diseñadas para la mantenibilidad del software híbrido.*

1. Introducción

La computación cuántica ha emergido como un área clave de investigación y desarrollo, prometiendo revolucionar diversas disciplinas al ofrecer velocidades de cálculo notablemente superiores a las de las computadoras convencionales, gracias a la explotación de los principios de la mecánica cuántica [Bernhard 2019]. No obstante, este avance plantea importantes desafíos, especialmente en el desarrollo de software capaz de aprovechar eficientemente estas capacidades cuánticas. Una solución a esto ha sido la creación de software híbrido, que integra componentes tradicionales y cuánticos para potenciar las soluciones ofrecidas.

Con la expansión de la computación cuántica en áreas como la inteligencia artificial y la medicina, se ha vuelto crucial asegurar la calidad de estos desarrollos [Piattini et al. 2020]. La calidad del software se evalúa mediante estándares internacionales como el ISO/IEC 25010 [ISO 2011], que forma parte de la familia de ISO/IEC 25000 [ISO 2014] y se enfoca en distintos aspectos de la calidad del producto software, incluida la mantenibilidad, que es esencial para garantizar la capacidad de un software de ser modificado y mejorado de manera eficiente [Rodríguez and Piattini 2014]. La mantenibilidad incluye subcaracterísticas como la analizabilidad, que se refiere a la facilidad con la que el software puede ser entendido.

Aunque se están desarrollando prácticas para el mantenimiento del software cuántico, persisten desafíos significativos debido al soporte inmaduro de herramientas y la falta de infraestructura adecuada para adaptar prácticas ágiles al desarrollo de este tipo de software [Moll et al. 2021, Khan et al. 2023]. Además, en el desarrollo ágil de software, los análisis estáticos de código son esenciales para identificar mejoras potenciales de forma continua [Gheorghe-Pop et al. 2020].

Debido a esta importante necesidad, este artículo presenta una solución innovadora que aborda el desafío de evaluar el software híbrido (clásico-cuántico). La investigación se centra en el desarrollo de un modelo y un entorno, formado por distintas herramientas, para medir y evaluar la analizabilidad del software híbrido, lo que representa un paso crucial para lograr una calidad integral en los productos software híbridos.

En respuesta a estos retos, este artículo propone una solución novedosa centrada en evaluar la calidad del software híbrido. Presenta un modelo y un conjunto de herramientas diseñadas para medir y evaluar la analizabilidad del software híbrido, lo que constituye un avance clave hacia la mejora de la calidad integral de estos productos software. El trabajo concluye resaltando las contribuciones de esta investigación y esbozando direcciones futuras para continuar avanzando en este campo.

2. Mantenibilidad del software híbrido

Los modelos de calidad del software, como los definidos por el estándar ISO/IEC 25010, son esenciales para evaluar la calidad de los productos software, adaptándose a sus necesidades y requisitos específicos. Este estándar identifica ocho características clave de la calidad, siendo una de ellas la mantenibilidad, que engloba la capacidad del software para ser modificado de manera fácil y eficiente, ya sea para corregir fallos o adaptarlo a nuevos requerimientos [Rodríguez et al. 2015]. Dentro de la mantenibilidad, la analizabilidad juega un papel crucial, ya que un software bien estructurado y documentado facilita significativamente su comprensión y modificación a lo largo de su ciclo de vida.

Las propiedades para la evaluación de la analizabilidad del software híbrido se han diseñado considerando todos los aspectos distintivos que caracterizan este tipo de sistemas. El enfoque se centra en la representación de los componentes clásicos y cuánticos, así como en la capacidad de realizar análisis estáticos en el entorno de evaluación.

2.1. Propiedades de analizabilidad para el software cuántico

Las propiedades de calidad que influyen en la capacidad de análisis del software cuántico se han centrado principalmente en el kit de desarrollo Qiskit, ya que es el utilizado por un mayor número de usuarios de la comunidad cuántica. Éstas son las siguientes:

- Incumplimiento de reglas de codificación Qiskit (QiCR) - La existencia de código que no cumple con las convenciones y normas establecidas para la programación cuántica. En este contexto, aún no se ha establecido una guía que defina las reglas de codificación específicas para Qiskit, aunque existen contribuciones recientes en este ámbito [Chen et al. 2023].
- Anchura del circuito (CW) - La cantidad de cúbits necesarios para representar y manipular la información en un circuito cuántico. Un circuito más ancho puede exigir más recursos y ser más complicado de analizar.
- Profundidad del circuito (CD) - Número de capas que componen un circuito cuántico. Una mayor profundidad puede incrementar la complejidad del circuito y, por ende, dificultar su comprensión.
- Complejidad de las puertas del circuito (CCG) - Evalúa el número y tipo de puertas cuánticas utilizadas en un circuito, según las contribuciones de Qiskit [Qiskit-Contributors 2024]. La dificultad de entender y modificar el circuito aumenta con puertas más complejas aplicadas sobre un mayor número de cúbits.
- Instrucciones condicionales (CI) - Las operaciones realizadas en un circuito cuántico basándose en los resultados de mediciones previas sobre los cúbits. Esto agrega complejidad y puede requerir un análisis más meticuloso y detallado para comprender el circuito.
- Complejidad ciclomática cuántica (QCC) - La complejidad de un programa cuántico en términos del número de posibles caminos lógicos a través del código cuántico [Kumar 2023]. Una mayor complejidad ciclomática puede indicar un circuito más difícil de analizar.
- Operaciones de medición (MO) - Las instrucciones que permiten extraer información sobre el estado de uno o más cúbits. Son críticas en puntos específicos porque representan la interacción entre el software clásico y el cuántico; una elección inadecuada de los puntos de medida puede afectar la analizabilidad del algoritmo.
- Operaciones de inicialización y reinicio (IRO) - Las operaciones que preparan el estado inicial de los cúbits antes de la ejecución del circuito y las que los restablecen a un estado conocido después de la ejecución. Comprender cómo se establecen y restablecen los cúbits puede complicar la comprensión del circuito.
- Cúbits auxiliares (AQ) - Los cúbits adicionales para realizar determinadas operaciones en un circuito. Estos cúbits auxiliares influyen directamente en la complejidad del algoritmo y requieren un análisis adicional para comprender cómo afectan al comportamiento del circuito.

De forma análoga a la evaluación realizada sobre el software clásico y sus propiedades, se evalúan las propiedades cuánticas de Qiskit utilizando la función por perfiles. Este procedimiento se basan en la experiencia de evaluación de la mantenibilidad por parte de AQCLab a lo largo de los últimos 25 años, tal y como se puede consultar en [Verdugo et al. 2024].

2.2. Métricas de analizabilidad para el software híbrido

Los métodos de cálculo, las métricas y las propiedades de análisis mencionadas están vinculados en el proceso de evaluar la calidad del software híbrido. Los métodos de cálculo constituyen los procedimientos y algoritmos empleados para medir cuantitativamente los diversos aspectos de la capacidad de análisis; ofrecen una base cuantitativa

para evaluar y comparar la aptitud del software para ser analizado y comprendido de manera efectiva. Las métricas, por su parte, son indicadores numéricos derivados de estos métodos, proporcionando medidas específicas de las propiedades de análisis. Con estas métricas en su lugar, es posible obtener una comprensión objetiva y cuantificable de la capacidad de análisis del software híbrido, identificando áreas que puedan requerir ajustes o mejoras.

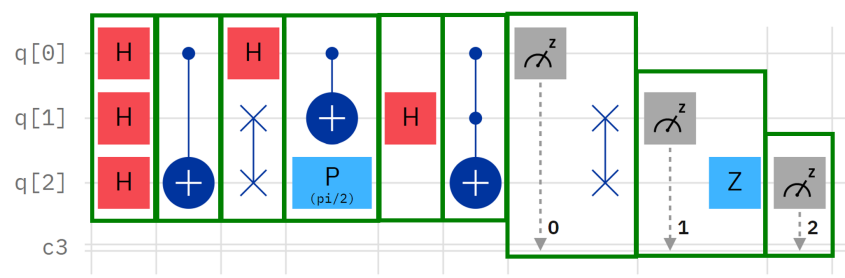
En el estudio de [Díaz et al. 2024], se presentó una versión inicial de las métricas para evaluar la capacidad de análisis de los sistemas híbridos, las cuales han sido refinadas mediante diversas validaciones realizadas sobre productos reales durante este trabajo.

A continuación, se detalla un ejemplo de procedimiento de cálculo.

2.2.1. Calcular la profundidad del circuito

La propiedad de la profundidad del circuito se relaciona directamente con la cantidad máxima de operaciones aplicadas sobre un mismo cúbit en el circuito cuántico. Se muestra un ejemplo de cálculo de la profundidad de un circuito cuántico en la Figura 1, donde se obtiene un resultado de 9 para el valor de su profundidad.

Figura 1. Representación de la profundidad del circuito



Para evaluar esta propiedad sobre el circuito cuántico se definen niveles y se utiliza una función por perfiles que indica la cantidad máxima aceptada de circuitos para cada nivel de calidad.

Una vez que los diversos circuitos han sido clasificados según sus profundidades, se procede a evaluar su distribución mediante métricas base. Estas métricas se dividen en tres niveles, siendo el nivel 3 el menos crítico y el nivel 1 el más significativo:

- NC_CD1: Número de circuitos con una profundidad del circuito de nivel 1.
- NC_CD2: Número de circuitos con una profundidad del circuito de nivel 2.
- NC_CD3: Número de circuitos con una profundidad del circuito de nivel 3.
- NCIR: Número total de circuitos.

El cálculo de estas métricas base, definidas para la propiedad de profundidad del circuito, se realiza en función de los umbrales especificados en la Tabla 1.

A partir de estas métricas base, se calculan las siguientes métricas derivadas:

- DC_CD1: Densidad de circuitos con profundidad del circuito de nivel 1.
- DC_CD2: Densidad de circuitos con profundidad del circuito de nivel 2.

Tabla 1. Niveles para la evaluación de la profundidad del circuito

| Rangos | | | |
|---------|---|-------|----------------------------------|
| Niveles | 1 | >17 | Profundidad del circuito mala |
| | 2 | 11-17 | Profundidad del circuito regular |
| | 3 | 1-10 | Profundidad del circuito buena |

Niveles de calidad

- DC_CD3: Densidad de circuitos con profundidad del circuito de nivel 3.

Estas métricas derivadas se obtienen dividiendo el número de circuitos de cada nivel por el número total de circuitos evaluados, de forma similar al cálculo de las densidades para la propiedad de la anchura del circuito.

Una vez obtenidas estas tres densidades, se determina un valor de calidad normalizado para la propiedad mediante la función por perfiles. En la función por perfiles para la profundidad del circuito se definen los rangos mostrados en la Tabla 2 (consultar [Verdugo et al. 2024]). Para cada nivel de calidad, se establece el umbral máximo aceptado de densidad de circuitos. Cabe destacar que la tabla no incluye el nivel de anchura más deseable, que corresponde a una buena anchura de los circuitos.

Tabla 2. Niveles y rangos para la evaluación de la profundidad del circuito

| | | Niveles | | |
|--------|---|---------|----|----------|
| | | 1 | 2 | |
| Rangos | 0 | - | - | 0 |
| | 1 | 20 | 40 | (0-33) |
| | 2 | 15 | 30 | [33-66) |
| | 3 | 10 | 20 | [66-100) |
| | 4 | 5 | 15 | 100 |

Niveles de calidad

3. Entorno para la evaluación de la analizabilidad del software híbrido

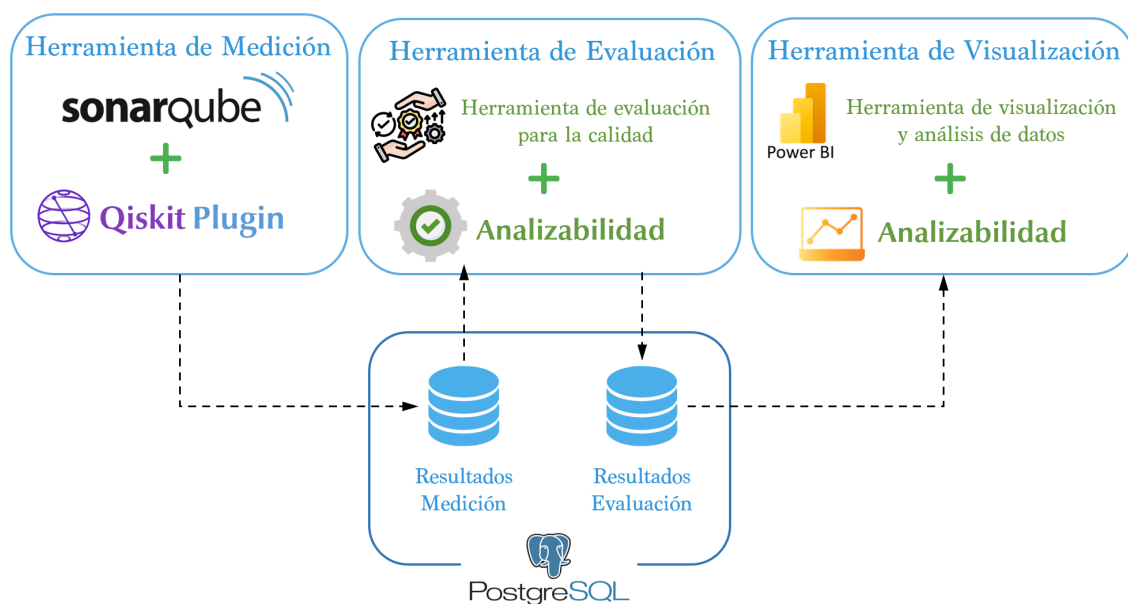
En este apartado presentamos el entorno tecnológico desarrollado y diseñado para medir, evaluar y visualizar los resultados de la analizabilidad del software híbrido.

Sobre la Figura 2 se proporciona una representación de la arquitectura de las distintas tecnologías. Este entorno se encuentra compuesto de tres herramientas: la herramienta de medición, la herramienta de evaluación y la herramienta de consulta y análisis de datos. La arquitectura modular y bien definida del entorno facilita una interacción fluida entre estas herramientas, lo que permite llevar a cabo una evaluación eficiente de la capacidad de análisis del software híbrido. Además, la elección de PostgreSQL como punto central de almacenamiento y comunicación entre las herramientas asegura la integridad y disponibilidad de los datos a lo largo de todo el proceso de evaluación.

Principalmente, hemos optado por utilizar SonarQube¹ en el entorno de desarrollo para el componente de medición, basándonos en dos ventajas clave. En primer lugar,

¹<https://www.sonarsource.com/products/sonarqube/>

Figura 2. Arquitectura del entorno tecnológico para la evaluación de la analizabilidad



destaca su capacidad para ser compatible con una amplia gama de lenguajes de programación, lo que lo convierte en una solución idónea para entornos híbridos que emplean diversas tecnologías.

En segundo lugar, la flexibilidad que ofrece SonarQube al incorporar extensiones o *plugins* personalizados resulta fundamental. Esta característica nos brinda la capacidad de ajustar la herramienta según las necesidades específicas de los sistemas híbridos desarrollados con Qiskit. Hemos creado un 'Qiskit Plugin' que integra las métricas propuestas para evaluar la analizabilidad del software híbrido. Gracias a esta funcionalidad, podemos adaptar las mediciones de la plataforma de manera personalizada para satisfacer los requisitos específicos de nuestro proyecto.

La evaluación del software híbrido se basa en una herramienta compuesta por un archivo XML, que detalla propiedades, métricas, métodos de cálculo y niveles de calidad, y un programa en Python. Este último automatiza la aplicación de los criterios definidos en el XML a los resultados de medición del código, calculando el nivel de analizabilidad del software, tanto en sus componentes clásicos como cuánticos, y del sistema híbrido en su conjunto.

Finalmente, la utilidad destinada al examen y exploración de datos juega un papel fundamental en la extracción de los resultados de la evaluación desde la base de datos PostgreSQL. Este instrumento posibilita la ejecución de consultas y análisis de datos de manera eficaz, suministrando datos significativos sobre la analizabilidad del software híbrido al encargado de la evaluación o auditor de software.

4. Conclusiones

La búsqueda de excelencia en el desarrollo de software cuántico se enfrenta a desafíos significativos debido a las características únicas de este tipo de software, que pueden hacerlo propenso a errores y difícil de mantener sin las prácticas de ingeniería

de software cuántico adecuadas. Esto podría impedir que el software alcance su máximo potencial [Akbar et al. 2023].

Este artículo presenta un entorno tecnológico diseñado para medir y evaluar la analizabilidad del software híbrido (clásico-cuántico), proporcionando valores cuantitativos de analizabilidad siguiendo pautas específicas. Se están realizando pruebas adicionales por auditores de software para verificar la funcionalidad y aplicabilidad del entorno.

La herramienta de medición desarrollada ha sido aplicada en investigaciones anteriores, como la realizada en [Alvarado-Valiente et al. 2023], donde se empleó para evaluar métricas en 16 algoritmos cuánticos, incluidos los algoritmos de Shor [Shor 1994] y Grover [Gover 1994], contribuyendo a la identificación de mejoras en la producción e implementación de servicios cuánticos específicos.

La integración de la medición en un entorno DevOps para realizar evaluaciones automáticas es un objetivo futuro, que eventualmente incluirá la incorporación de más herramientas a este pipeline. Este trabajo no solo sienta las bases para evaluaciones de calidad más exhaustivas y precisas en el futuro, sino que también abre la puerta a la incorporación de nuevas ideas de desarrollo, incluida la expansión del entorno de evaluación para cubrir otras características de calidad del software y para abarcar otras plataformas de desarrollo híbridas como Q#.

Agradecimientos

Esta investigación ha contado con el apoyo de Q-SERV (Quantum Service Engineering: Development Quality, Testing and Security of Quantum Microservices) financiado por el Ministerio de Ciencia e Innovación y los Fondos FEDER (PID2021-124054OB-C32); QU-ASAP (Prototipo de modernización del software quantum) financiado por el Ministerio de Ciencia e Innovación y los Fondos NextGenerationEU (PDC2022-133051-I00); y ayudas para la realización de proyectos de investigación aplicada, en el marco del Plan Propio de Investigación, cofinanciadas por el Fondo Europeo de Desarrollo Regional (Feder): UNION (2022-GRIN-34110) y MESIAS (2022-GRIN-34202)

Referencias

- Akbar, M., Khan, A., and Rafi, S. (2023). A systematic decision-making framework for tackling quantum software engineering challenges. *Automated Software Engineering*, 30(22).
- Alvarado-Valiente, J., Romero-Álvarez, J., Díaz, A., Rodríguez, M., García-Rodríguez, I., Moguel, E., Garcia-Alonso, J., and Murillo, J. M. (2023). Quantum services generation and deployment process: A quality-oriented approach. *QUATIC: The International Conference on the Quality of Information and Communications Technology*, 1871.
- Bernhard, C. (2019). *Quantum Computing for Everyone*. The MIT Press.
- Chen, Q., Câmara, R., Campos, J., Souto, A., and Ahmed, I. (2023). The smelly eight: An empirical study on the prevalence of code smells in quantum computing. pages 358–370, Melbourne, Australia: IEEE.
- Díaz, A., Rodríguez, M., and Piattini, M. (2024). Towards a set of metrics for hybrid (quantum/classical) systems maintainability. *Journal of Universal Computer Science (J.UCS)*, 30(1):25–48.

- Gheorghe-Pop, I.-D., Tcholtchev, N., Ritter, T., and Hauswirth, M. (2020). Quantum devops: Towards reliable and applicable nisq quantum computing. *IEEE Globecom Workshops*, pages 1–6.
- Gover, L. K. (1994). A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, page 212–219.
- ISO (2011). Iso/iec 25010. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Retrieved from *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*.
- ISO (2014). Iso/iec 25000. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE*.
- Khan, A., Akbar, M., Ahmad, A., Fahmideh, M., Shameem, M., Lahtinen, V., Waseem, M., and Mikkonen, T. (2023). *2023 IEEE International Conference on Quantum Software (QSW)*, chapter Agile Practices for Quantum Software Development: Practitioners’ Perspectives, pages 9–20. IEEE.
- Kumar, A. (2023). Formalization of structural test cases coverage criteria for quantum software testing. *International Journal of Theoretical Physics*, 62:1–16.
- Moll, N., Barkoutsos, P., and Benjamin, S. (2021). Quantum development beyond qiskit. *npj Quantum Information*, 7(1):1–7.
- Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J., Serrano, M., González, G. H., Guzmán, I., Andrés, C., Polo, M., Murina, E., Jiménez, L., Marqueño, J., Gallego, R., Tura, J., Phillipson, F., Murillo, J., Niño, A., and Rodríguez, M. (2020). The talavera manifesto for quantum software ingenieering and programming. Talavera de la Reina.
- Qiskit-Contributors (2024). Qiskit: Circuit library.
- Rodríguez, M. and Piattini, M. (2014). Software product quality evaluation using iso/iec 25000. *ERCIM News*, 99.
- Rodríguez, M., Piattini, M., and Fernández, C. (2015). A hard look at software quality: Pilot program uses iso/iec 25000 family to evaluate, improve and certify software products. *Quality Progress*, 48(9):30–36.
- Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134.
- Verdugo, J., Oviedo, J., Rodríguez, M., and Piattini, M. (2024). Connecting research and practice for software product quality certification: a 25-year journey. *Accepted to IEEE Software*.