Una revisión de herramientas para Programación Cuántica

Luciano Marrero[1], Verena Olsowy [1], Juan Fernández Sosa[1], Fernando Tesone, Leonardo Corbalán[1], Pablo Thomas[1], Patricia Pesado[1]

[1] Instituto de Investigación en Informática LIDI
Facultad de Informática - Universidad Nacional de La Plata – Argentina
Centro Asociado Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

{lmarrero, volsowy, jfernandez, ftesone, corbalan, pthomas, ppesado}@lidi.info.unlp.edu.ar

Abstract. Con el objetivo de lograr computadoras con mayor velocidad de procesamiento, se ha incrementado la escala de integración para incluir más transistores en un mismo espacio físico, fabricando microchips cada vez más pequeños. Sin embargo, existe un límite en el cual éstos dejan de funcionar correctamente. Cuando se llega a la escala de nanómetros, los electrones se escapan de los canales por donde deben circular debido a un fenómeno conocido como "efecto túnel", asociado a la física cuántica. Actualmente, la computación tradicional se encuentra limitada porque ha alcanzado escalas difíciles de seguir reduciendo. En este contexto, desde hace varios años, algunas empresas han invertido en el desarrollo de tecnología cuántica con el objetivo de construir sus propias computadoras cuánticas. La evolución de la computación cuántica en los últimos años representa un cambio radical para muchos aspectos de la computación binaria. Este trabajo presenta una revisión de las herramientas actuales para la programación cuántica, con el objetivo de destacar los aspectos más relevantes según su contexto de aplicación en el ámbito informático.

Keywords: programación cuántica, lenguajes de programación cuántica, simuladores cuánticos

1 Introducción

El concepto de computación cuántica surgió a principios de la década de 1980 con Paul Benioff, quien realizó simulaciones sobre algunos principios fundamentales de la mecánica cuántica en ordenadores binarios (máquinas de Turing). Entre 1981 y 1982, Richard Feynman propuso utilizar fenómenos de la física cuántica para resolver problemas computacionales de manera más rápida. A partir de la década de 1990, empezaron a aparecer los primeros algoritmos cuánticos, aplicaciones cuánticas y

máquinas capaces de realizar cálculos cuánticos. Mediante comparaciones, se demostró la ventaja de una computadora cuántica sobre una binaria. Entre los algoritmos cuánticos más destacados se encuentra el algoritmo de Shor, desarrollado por Peter Shor, el cual sería teóricamente capaz de calcular los factores primos de números a una velocidad mucho mayor que cualquier computadora tradicional. A fines de la década de 1990, se realizaron algunas aplicaciones prácticas de lo que hasta entonces era solo una teoría [1, 2].

En 1998, apareció la primera máquina cuántica con 2 qubits (bits cuánticos), presentada en la Universidad de Berkeley, California. Posteriormente, surgieron nuevas máquinas cuánticas con el objetivo de aumentar la cantidad de qubits y ejecutar algoritmos cuánticos [1, 2].

El resto del trabajo se organiza de la siguiente manera: en la sección 2 se introducen los conceptos básicos de computación cuántica, luego se describen los lenguajes de programación, y posteriormente la computación cuántica en la nube y los simuladores cuánticos en línea. Finalmente, se presentan las conclusiones y trabajo futuro.

2 Conceptos Básicos de Computación Cuántica

La computación cuántica es un área de investigación que estudia algoritmos y sistemas utilizando las propiedades de la física cuántica. Se basa en los principios de la superposición y el entrelazamiento cuántico para el desarrollo de computadoras que poseen características distintas a las computadoras binarias. Estas máquinas son capaces de almacenar más estados por unidad de información y, de esta manera, operar con algoritmos de forma más eficiente que una computadora binaria. En este contexto, a continuación se describen brevemente algunos conceptos importantes para la computación cuántica.[2]

Qubit: un qubit, o bit cuántico, es la unidad básica de información en la computación cuántica, análoga al bit en la computación binaria. Mientras que un bit puede ser 0 o 1, un qubit puede estar en una superposición de ambos estados simultáneamente. En computación binaria, los bits se agrupan en conjuntos de 8, esto se denomina bytes. La cantidad de estados que un conjunto de bits puede representar se calcula como 2^n, donde n es el número de bits. Los qubits, al poder representar múltiples estados a la vez, permiten que la computación cuántica pueda manejar y procesar información de manera exponencialmente más eficiente que las computadoras binarias [3].

Superposición: es una propiedad que permite a un qubit estar en múltiples estados al mismo tiempo. En una computadora binaria, un bit sólo puede tener valor 0 o 1 en un momento dado. En una computadora cuántica, un qubit puede tener valor 0, 1 o cualquier combinación de ambos simultáneamente. Esta capacidad de estar en múltiples estados a la vez permite a las computadoras cuánticas procesar una enorme cantidad de información en paralelo [2].

Entrelazamiento: es un fenómeno donde dos o más qubits se vinculan de tal manera que el estado de uno de ellos está directamente relacionado con el estado del

otro, sin importar la distancia que los separe. Esto significa que una operación realizada en un qubit afectará instantáneamente a su qubit entrelazado. El entrelazamiento permite a las computadoras cuánticas realizar operaciones complejas de manera más eficiente, ya que los cambios en un qubit pueden influir en otros qubits de forma predecible y simultánea [2, 3].

Las propiedades de superposición y entrelazamiento otorgan a las computadoras cuánticas gran capacidad de procesamiento para resolver problemas que en una computadora binaria demandaría un tiempo considerablemente mayor.

Puertas lógicas cuánticas: Los microprocesadores en las computadoras binarias, operan a través de puertas (o compuertas) lógicas como AND, OR, XOR, NOT, NAND, NOR y XNOR. Una computadora cuántica opera de manera similar pero utiliza puertas lógicas cuánticas CNOT, Pauli, Hadamard, Toffoli y SWAP, entre otras [3].

Estas puertas lógicas cuánticas se representan mediante matrices. Para calcular el resultado en la salida de una operación entre puertas lógicas cuánticas, se realiza el producto matricial de una puerta determinada por el vector que representa el estado actual del computador cuántico [3].

A*lgoritmos cuánticos:* consisten en aplicar operaciones (puertas lógicas cuánticas) que operan en los estados de superposición y entrelazamiento de los qubits. La estructura general de un algoritmo cuántico comienza con la inicialización, donde se preparan los qubits en un estado inicial, generalmente el estado base $|0\rangle$. A continuación, se aplica una serie de puertas lógicas cuánticas, por ejemplo Hadamard, Pauli-X, Pauli-Y, Pauli-Z, para manipular los qubits y crear una superposición y/o entrelazamiento de estados. Luego, se implementan transformaciones cuánticas específicas para resolver el problema. Finalmente, se mide el estado final de los qubits. La medición colapsa la superposición a uno de los posibles resultados, proporcionando así la solución al problema. [1,2,3]

3 Lenguajes de Programación Cuántica

Los lenguajes de programación cuántica son diseñados específicamente para escribir algoritmos ejecutables en computadoras cuánticas. Este tipo de lenguajes de programación opera con qubits y aprovecha fenómenos como la superposición, el entrelazamiento y la interferencia. Los algoritmos cuánticos implican la programación de circuitos cuánticos mediante la relación de puertas cuánticas. Aunque para pequeños ejemplos o prototipos se suele utilizar una notación gráfica, este método se vuelve impráctico a medida que los circuitos se vuelven más complejos [4].

A lo largo del tiempo han surgido diversos lenguajes de programación cuántica [4] [5]. Algunos de ellos no son considerados lenguajes de programación autónomos sino más bien librerías o lenguajes embebidos que pueden ser invocados utilizando lenguajes de programación de propósito general, como por ejemplo Python [4]. Similar a los lenguajes de programación tradicionales, los lenguajes de programación cuántica presentan diferentes grados de abstracción al programador: los de bajo nivel están orientados a una programación directa sobre el circuito cuántico, mientras que aquellos de un nivel de abstracción alto ocultan detalles complejos del

hardware cuántico y las operaciones de bajo nivel. Los lenguajes de programación cuántica de alto nivel resultan más adecuados para introducir a los profesionales informáticos tradicionales en el campo de la programación cuántica.

En la tabla 1 se presentan trece lenguajes de programación cuántica identificados en este estudio, cada uno acompañado de su fecha de creación y el lenguaje de programación tradicional en el que se basa. Esta lista se compone de varios lenguajes de programación seleccionados en [4], enfocándose en aquellos creados a partir de 2016 y que ofrecen un alto nivel de abstracción.

Tabla 1. Lenguajes de programación cuántica en estudio

Año	Nombre	Lenguaje base	Referencia
2016	FJQuantum	Java	[9]
2016	ProjectQ	Python	[6]
2016	pyQuil	Python	[8]
2016	qPCF	PCF	[7]
2017	Qiskit	Python	[10]
2018	Blackbird	Python	[14]
2018	Cirq	Python	[13]
2018	IQu	Idealized Algol	[7]
2018	QuantumOptics.jl	Julia	[11]
2018	Q#	C#	[12]
2018	Strawberry Fields	Python	[14]
2020	OpenQL	C++ / Python	[15]
2020	Silq	Python	[16]

Se analizan cada uno de los lenguajes en función de las siguientes categorías: facilidad de aprendizaje, ecosistema y herramientas y el uso en educación. Los resultados se muestran en la tabla 2 y se detallan a continuación.

3.1 Facilidad de aprendizaje

En este punto se evalúa lo accesible y comprensible que resulta el lenguaje de programación cuántica para los desarrolladores que desean iniciarse en la programación cuántica. Se considera la disponibilidad de documentación y recursos educativos, así como la curva de aprendizaje del lenguaje base, la comunidad y el soporte asociados a éste.

La tabla 1 deja en evidencia que la mayoría de los lenguajes de programación cuántica mencionados están basados en Python. Este tipo de lenguaje de gran popularidad y comunidad permite que la transición de los programadores clásicos hacia la programación cuántica sea más suave. Lenguajes como Q#, que se basa en C#, también ofrece facilidad de aprendizaje debido a su integración con herramientas conocidas como Visual Studio y la gran cantidad de documentación que se encuentra disponible en línea.

Analizando otros lenguajes de programación base como el caso de Julia, para el lenguaje QuantumOptics.jl, se puede presentar una curva de aprendizaje más pronunciada para aquellos que no están familiarizados con dicho lenguaje base. La comunidad de Julia, aunque creciente, no es tan extensa como la de Python, lo que puede limitar el acceso a recursos y soporte.

Para finalizar este análisis en el contexto de este trabajo, la facilidad de aprendizaje puede ser considerada como *Alta*, *Media* o *Baja*.

3.2 Ecosistema y herramientas

El ecosistema y las herramientas disponibles para los lenguajes de programación cuántica juegan un papel fundamental en la facilidad de uso y adopción. Este criterio tiene en cuenta la variedad de frameworks, simuladores, entornos de desarrollo integrados (IDEs), entre otros recursos que apoyan el desarrollo de software cuántico.

Un ecosistema *robusto o extenso* incluye un entorno completo para desarrollar, probar y ejecutar algoritmos cuánticos. Por ejemplo Qiskit ofrece diferentes componentes que permiten ejecutar algoritmos cuánticos en hardware real [10]. Q#, propiedad de Microsoft, posee un kit de desarrollo (QDK) que se integra a Visual Studio y facilita el desarrollo y depuración de los algoritmos cuánticos [19]. Cirq es otro caso de un lenguaje de programación que proporciona herramientas para la simulación y la integración con hardware cuántico de Google.

Aquellos que presentan un ecosistema *moderado*, como el caso de OpenQL y QuantumOptics.jl, también posibilitan simular y ejecutar algoritmos cuánticos, pero su ecosistema no es tan amplio como los anteriores.

Por último los ecosistemas *limitados*, como es el caso de qPCF, FJQuantum, IQu y Blackbird, están enfocados principalmente en el ámbito académico y experimental. Silq al ser uno de los lenguajes más recientes de este estudio, se encuentra en una fase de desarrollo temprano con un ecosistema emergente.

3.3 Uso en educación

Este criterio evalúa si el lenguaje y los recursos que tiene asociados son adecuados para la enseñanza de la computación cuántica. Para esto se tiene en cuenta por ejemplo la disponibilidad de material didáctico, ejemplos prácticos, ejercicios y entornos de aprendizaje interactivos.

PojectQ, pyQuil, Qiskit, Q# y Cirq son las mejores opciones para este caso ya que cuentan con un amplio soporte educativo que incluye tutoriales, ejemplos prácticos, ejercitación, etc. Por ejemplo Qiskit ofrece un conjunto de guías, tutoriales y cursos para fomentar el aprendizaje de la programación cuántica utilizando Qiskit [17] [18]. Para el lenguaje Q#, existe una serie de tutoriales interactivos [19] que cubren algoritmos de diferentes niveles de complejidad. Además, la utilización de un entorno familiar como Visual Studio facilita el aprendizaje. Cirq ofrece tutoriales detallados y una guía de aprendizaje en su sitio web con ejemplos prácticos [13].

En el caso de QuantumOptics.jl, Strawberry Fields, OpenQL y Silq poseen recursos educativos pero no son tan extensos como los anteriores, por lo menos al momento de realizar este estudio. Por último, qPCF, FJQuantum, IQu y Blackbird tienen recursos educativos limitados, en parte porque son lenguajes utilizados en contextos académicos y de investigación especializada con pocos recursos prácticos disponibles.

Considerando las definiciones previas, se califica el uso en educación como *extenso*, *moderado* o *limitado*.

Tabla 2. Análisis de lenguajes de programación cuántica

Nombre	Facilidad de aprendizaje	Ecosistema y herramientas	Uso en educación
ProjectQ	Alta	Extenso	Extenso
qPCF	Media	Limitado	Limitado
pyQuil	Alta	Extenso	Extenso
FJQuantum	Media	Limitado	Limitado
Qiskit	Alta	Extenso	Extenso
lQu	Media	Limitado	Limitado
QuantumOptics.jl	Media	Moderado	Moderado
Q#	Alta	Extenso	Extenso
Cirq	Alta	Extenso	Extenso

Strawberry Fields	Alta	Extenso	Moderado
Blackbird	Media	Limitado	Limitado
OpenQL	Media	Moderado	Moderado
Silq	Media	Limitado	Moderado

4 Computación Cuántica en la Nube y Simuladores Cuánticos en Línea

La computación cuántica en la nube o *Quantum Cloud Computing* (QCC) y los simuladores cuánticos en línea son dos herramientas que están democratizando el acceso a la computación cuántica en el mundo y abriendo un abanico de posibilidades para la investigación, el desarrollo y la innovación en diversos campos.

Aunque la computación cuántica promete superar las prestaciones de las computadoras clásicas, la implementación de sistemas cuánticos a gran escala aún presenta desafíos técnicos y económicos significativos. La QCC aborda estas limitaciones al combinar la potencia de la computación cuántica con la flexibilidad y escalabilidad de la computación en la nube.

4.1 Ventajas de la Computación Cuántica en la Nube

A continuación se enuncian algunas de las ventajas que presenta la Computación Cuántica en la Nube.

Accesibilidad universal: La QCC elimina las barreras geográficas y económicas, permitiendo que usuarios de todo el mundo accedan a recursos cuánticos de manera remota a través de internet.

Reducción de costos: Al evitar la necesidad de adquirir y mantener hardware cuántico especializado, la QCC ofrece una solución más rentable para la mayoría de los usuarios.

Escalabilidad: La infraestructura en la nube permite escalar los recursos cuánticos de forma dinámica, adaptándose a las necesidades de cada proyecto.

Colaboración: Facilita la colaboración entre investigadores y equipos de trabajo, permitiendo compartir recursos y conocimientos de manera fluida.

4.2 Arquitectura de la Computación Cuántica en la Nube

La arquitectura de un sistema QCC típico comprende los siguientes componentes:

Interfaz en la nube: Permite a los usuarios interactuar con los recursos cuánticos de forma remota a través de un portal web o una aplicación.

APIs: Brindan acceso al software como servicio (SaaS), abstrayendo la complejidad del hardware subyacente y facilitando la programación de tareas cuánticas.

Orquestación del entorno de ejecución cuántico: Gestiona la asignación de recursos cuánticos a las tareas en ejecución, optimizando el uso de los recursos

disponibles.

Distribución de recursos: Distribuye los datos y las tareas entre los diferentes nodos del sistema cuántico para maximizar el rendimiento.

Almacenamiento: Brinda un espacio seguro para almacenar datos cuánticos y resultados de las ejecuciones.

Conectividad: Facilita la comunicación entre los componentes del sistema y con los usuarios, asegurando una transmisión de datos confiable y segura.

Computadoras cuánticas: Procesan las tareas cuánticas en centros de datos remotos, utilizando tecnologías como superconductores o trampas de iones.

En la actualidad, la QCC se encuentra en sus primeras etapas, y la interconexión entre las computadoras cuánticas y los usuarios se basa principalmente en las tecnologías clásicas de Internet y la informática. Esto da lugar a la aparición de modelos híbridos de QCC, donde los poderosos procesadores cuánticos se combinan con la infraestructura clásica de redes y procesamiento de datos [20]. En este contexto, los simuladores cuánticos en línea adquieren una relevancia fundamental como componentes esenciales de este ecosistema.

4.3 El Rol de los Simuladores Cuánticos en Línea

Los simuladores cuánticos en línea han surgido como una herramienta fundamental para el aprendizaje y la experimentación en computación cuántica. Estos entornos accesibles permiten a usuarios de todo el mundo, incluso aquellos sin experiencia previa en programación o computación, familiarizarse con los conceptos fundamentales de esta disciplina emergente. En este sentido, los simuladores cuánticos en línea ofrecen diversas ventajas:

Accesibilidad: Eliminan barreras geográficas y económicas, permitiendo el acceso a la computación cuántica desde cualquier lugar con conexión a internet.

Facilidad de uso: Ofrecen interfaces amigables e intuitivas, incluso para usuarios sin experiencia previa en programación o computación.

Escalabilidad: Permiten escalar la complejidad de los problemas simulados, posibilitando un avance gradual en el aprendizaje de áreas más avanzadas.

Aprendizaje práctico: Brindan la oportunidad de poner en práctica los conocimientos adquiridos, ejecutando algoritmos cuánticos y observando su comportamiento en un entorno simulado.

Puente hacia la computación cuántica real: Sirven como paso previo para ejecutar los algoritmos desarrollados en simuladores en computadoras cuánticas reales.

4.4 Criterios de selección de simuladores cuánticos en línea

Los simuladores cuánticos en línea ofrecen una plataforma ideal para el aprendizaje, la experimentación y el desarrollo de habilidades en computación cuántica, abriendo un abanico de posibilidades para investigadores, estudiantes, profesionales y empresas.

En este contexto, la elección del simulador cuántico en línea adecuado puede ser decisivo para maximizar la experiencia de aprendizaje y alcanzar los objetivos específicos de cada usuario. A continuación, en la Tabla 3 se presenta un análisis

comparativo de cuatro de los simuladores cuánticos en línea más representativos: IBM Quantum Composer [21], Google Quantum Playground [22], Amazon Braket [23] y Quantum Inspire [24], destacando sus características clave y criterios de selección. Todos ellos son ampliamente reconocidos dentro de la comunidad cuántica por su capacidad para satisfacer las necesidades tanto de principiantes como de usuarios experimentados

Tabla 3. Análisis de simuladores cuánticos en línea

Caracte-rísticas	**Simuladores**			
	IBM Quantum Composer	**Google Quantum Playground**	**Amazon Braket**	**Quantum Inspire**
Proveedor	IBM	Google	Amazon	QuTech
Facilidad de uso	Interfaz intuitiva (arrastrar y soltar) Tutoriales interactivos Documentación extensa	Interfaz visualmente atractiva Experimentos gamificados Simulaciones en 3D	Entorno de desarrollo basado en la nube Integración con AWS	Interfaz gráfica intuitiva Experimentos predefinidos Visualización de circuitos y resultados
Recursos educativos	Amplia biblioteca de algoritmos y tutoriales. Cursos en línea. Comunidad activa	Tutoriales interactivos. Guías paso a paso. Recursos para educadores	Documentación detallada. Ejemplos de código. Soporte de la comunidad	Documentación extensa. Ejemplos de código. Comunidad activa
Lenguajes de progra-mación	Python, Qiskit	JavaScript	Cirq, Qiskit, Forest	QASM
Integración	IBM Q	Google Cloud	AWS	Spin-2, Ion Trap
Orientación	Principiantes, estudiantes, investigadores	Principiantes, estudiantes, educadores	Usuarios experimentados, profesionales, empresas	Principiantes, estudiantes, investigadores
Criterios de selección	Facilidad de uso, amplia gama de recursos, alineación con IBM Q	Experiencia de aprendizaje atractiva, enfoque en la comprensión conceptual, introducción a principios básicos	Flexibilidad para ejecutar algoritmos en diferentes plataformas cuánticas, integración con herramientas de AWS, orientación a análisis y visualización de datos	Facilidad de uso, experimentación directa con circuitos cuánticos, integración con hardware cuántico

IBM Quantum Composer: Este simulador se destaca por su facilidad de uso, su amplia gama de recursos educativos y su alineación con la plataforma de computación cuántica IBM Q. Su interfaz intuitiva, basada en el paradigma de arrastrar y soltar,

permite a los usuarios crear y modificar circuitos cuánticos de manera sencilla, sin necesidad de conocimientos previos de programación. Además, IBM Quantum Composer ofrece una extensa biblioteca de algoritmos predefinidos, tutoriales interactivos y cursos en línea que guían a los usuarios paso a paso en el aprendizaje de la computación cuántica. A esto se suma la estrecha integración con la plataforma IBM Q, lo que permite a los usuarios ejecutar sus algoritmos simulados en hardware cuántico real, brindándoles una experiencia práctica y valiosa.

Google Quantum Playground: Con su interfaz visualmente atractiva, experimentos gamificados y simulaciones en 3D, Google Quantum Playground se convierte en una herramienta ideal para principiantes y estudiantes que buscan una introducción lúdica y atractiva a los conceptos básicos de la computación cuántica.

Amazon Braket: Dirigido a usuarios experimentados y profesionales, Amazon Braket se presenta como una plataforma integral para el desarrollo y la ejecución de algoritmos cuánticos. Su entorno de desarrollo basado en la nube ofrece flexibilidad y escalabilidad, permitiendo ejecutar algoritmos en diferentes plataformas cuánticas, incluyendo las de IBM, Rigetti y Oxford Quantum Circuits. Además, Braket se integra a la perfección con las herramientas de análisis y visualización de datos de AWS, facilitando el procesamiento y la comprensión de los resultados obtenidos. Esta potente combinación lo convierte en una herramienta ideal para empresas y profesionales que buscan explorar las aplicaciones prácticas de la computación cuántica en áreas como la optimización financiera, la logística y la cadena de suministro.

Quantum Inspire: Desarrollado por QuTech, Quantum Inspire ofrece una interfaz gráfica intuitiva que permite a los usuarios diseñar y ejecutar circuitos cuánticos de forma directa. Incluye una biblioteca de experimentos predefinidos y herramientas de visualización para analizar los resultados. Su principal atractivo reside en la posibilidad de integrar los algoritmos simulados con hardware cuántico real, lo que lo convierte en una herramienta ideal para aquellos que buscan dar un paso más allá de la simulación y experimentar con la computación cuántica real.

5 Conclusiones

En este trabajo se ha explorado el campo emergente de la computación cuántica, destacando la capacidad única de los qubits para operar en estados de superposición y entrelazamiento. Estas propiedades permiten a las computadoras cuánticas resolver problemas de manera exponencialmente más eficiente que las computadoras clásicas.

Se ha realizado una revisión general de los principales lenguajes de programación cuántica, subrayando su papel decisivo en la accesibilidad y la educación. El análisis incluye aspectos como la facilidad de aprendizaje, el ecosistema y las herramientas disponibles, así como su aplicación en entornos educativos.

La computación cuántica en la nube ha sido identificada como una herramienta transformadora que elimina barreras geográficas y económicas al permitir el acceso remoto a recursos cuánticos escalables.

La disponibilidad de lenguajes de programación cuántica, la computación en la nube y los simuladores de sistemas cuánticos actúan como un puente esencial entre los conceptos teóricos y su aplicación práctica, permitiendo a los estudiantes y profesionales traducir ideas abstractas en implementaciones concretas. Los simuladores en línea, en particular, juegan un papel fundamental en el aprendizaje y desarrollo de esta tecnología. Éstos permiten a los desarrolladores experimentar con algoritmos sin la necesidad de acceso a hardware costoso, allanando el camino para la innovación y el descubrimiento.

La revisión demuestra que el campo de la computación cuántica, a pesar de su complejidad inherente, se está volviendo cada vez más accesible para quienes estén interesados en ella. La proliferación de herramientas gratuitas y recursos en línea ha democratizado significativamente el aprendizaje y la experimentación en este campo. A medida que las plataformas cuánticas se vuelvan más sofisticadas y accesibles, se espera que nuevas aplicaciones y descubrimientos impulsen avances significativos en la ciencia y la tecnología.

Las conclusiones sugieren un futuro prometedor para la computación cuántica, donde la colaboración y la investigación interdisciplinaria jugarán roles decisivos. Del análisis aquí presentado se deduce la importancia de integrar la computación cuántica en las currículas de informática. La facilidad de aprendizaje de muchas de estas herramientas hace que sea un momento ideal para introducir estos conceptos en la educación superior.

En resumen, es recomendable aprovechar estos recursos accesibles para explorar el mundo de la computación cuántica. La barrera de entrada nunca ha sido más baja y el potencial de innovación nunca ha sido más alto. Aunque el impacto pleno de esta tecnología aún está por materializarse, la computación cuántica representa un cambio de paradigma que promete transformar radicalmente la capacidad para abordar problemas complejos. Sin duda, estos avances marcarán el futuro de la informática, abriendo nuevas fronteras en la resolución de desafíos computacionales que antes parecían insuperables.

7 Bibliografía

1. Moret Bonillo, V. Principios fundamentales de computación cuántica. Departamento de Computación. Facultad de Informática.
2. https://www.xataka.com/ordenadores/computacion-cuantica-que-es-de-donde-viene-y-que-ha-conseguido
3. https://www.xataka.com/investigacion/ordenadores-cuanticos-explicados-como-funcionan-que-problemas-pretenden-resolver-que-desafios-deben-superar-para-lograrlo
4. Serrano, M. A., Cruz-Lemus, J. A., Perez-Castillo, R., & Piattini, M. (2022). Quantum software components and platforms: Overview and quality assessment. ACM Computing Surveys, 55(8), 1-31.
5. Dwivedi, K., Haghparast, M., & Mikkonen, T. (2024). Quantum software engineering and quantum software development lifecycle: a survey. Cluster Computing, 1-19.
6. Steiger, D. S., Häner, T., & Troyer, M. (2018). ProjectQ: an open source software framework for quantum computing. Quantum, 2, 49.

7. Paolini, L., Piccolo, M., & Zorzi, M. (2019). QPCF: Higher-order languages and quantum circuits. Journal of Automated Reasoning, 63, 941-966.
8. Koch, D., Wessing, L., & Alsing, P. M. (2019). Introduction to coding quantum algorithms: A tutorial series using pyquil. arXiv preprint arXiv:1903.05195.
9. Feitosa, S. S., Vizzotto, J. K., Piveta, E. K., & Du Bois, A. R. (2016). FJQuantum–a quantum object oriented language. Electronic Notes in Theoretical Computer Science, 324, 67-77.
10 Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C. J., Lishman, J., Gacon, J., ... & Gambetta, J. M. (2024). Quantum computing with Qiskit. arXiv preprint arXiv:2405.08810.
11. Krämer, S., Plankensteiner, D., Ostermann, L., & Ritsch, H. (2018). QuantumOptics. jl: A Julia framework for simulating open quantum systems. Computer Physics Communications, 227, 109-116.
12. Svore, K., Geller, A., Troyer, M., Azariah, J., Granade, C., Heim, B., ... & Roetteler, M. (2018, February). Q# enabling scalable quantum computing and development with a high-level dsl. In Proceedings of the real world domain specific languages workshop 2018 (pp. 1-10).
13. Google Quantum AI. (s.f.). Cirq. Accedido en junio de 2024, de https://quantumai.google/cirq
14. Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., & Weedbrook, C. (2019). Strawberry fields: A software platform for photonic quantum computing. Quantum, 3, 129.
15. Khammassi, N., Ashraf, I., Someren, J. V., Nane, R., Krol, A. M., Rol, M. A., ... & Almudever, C. G. (2021). OpenQL: A portable quantum programming framework for quantum accelerators. ACM Journal on Emerging Technologies in Computing Systems (JETC), 18(1), 1-24.
16. Bichsel, B., Baader, M., Gehr, T., & Vechev, M. (2020, June). Silq: A high-level quantum language with safe uncomputation and intuitive semantics. In Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (pp. 286-300).
17. IBM Quantum. (s.f.). IBM Quantum Experience. Accedido en junio de 2024, de https://learning.quantum.ibm.com/
18. IBM Quantum. (s.f.). IBM Quantum Documentation. Accedido en junio de 2024, de https://docs.quantum.ibm.com/
19. Microsoft. (s.f.). Introducción a los katas en Q# y QDK. Accedido en junio de 2024, de https://learn.microsoft.com/es-es/azure/quantum/tutorial-qdk-intro-to-katas
20. Nguyen, H. T., Krishnan, P., Krishnaswamy, D., Usman, M., & Buyya, R. (2024). Quantum Cloud Computing: A Review, Open Problems, and Future Directions. arXiv preprint arXiv:2404.11420.
21. IBM Quantum Composer. Accedido en junio de 2024, https://quantum.ibm.com/composer
22. Google Quantum Playground. Accedido en junio de 2024, https://www.quantumplayground.net
23. Amazon Bracket. Accedido en junio de 2024, https://aws.amazon.com/es/braket
24. Quantum Inspire . Accedido en junio de 2024, https://www.quantum-inspire.com