



# UNIVERSIDAD COMPLUTENSE MADRID

Proyecto de Innovación

Convocatoria 2020/2021

Nº de proyecto 398

Título del proyecto:

Enseñanza de coMputación cuántica Práctica pAra esTudiantes de Informática:  
Arquitectura y programación (EMPATIA)

Nombre del responsable del proyecto: Guillermo Botella Juan

Departamento: Arquitectura de Computadores y Automática

## 1. Objetivos propuestos en el presente proyecto:

La computación cuántica es ya una realidad y se está afianzando cada año. La llegada de la primera computadora cuántica comercial, la IBM Q System One, nos ha demostrado que esta disciplina ha alcanzado una madurez difícil de imaginar hace no más de cinco años [1]. Pero este no es el único hito que hemos presenciado recientemente. IBM anunció la primera computadora cuántica comercial de 53 qubits y un artículo de Google [2-3] refleja que esta compañía podría haber logrado la "supremacía cuántica". 2019 y 2020 fueron años muy prolíficos en términos de computación cuántica, y seguimos viendo un avance en 2021 gracias, sobre todo, al impulso de empresas como IBM, Google o Microsoft, y, también en menor medida, Amazon y Alibaba, dos grandes gigantes comerciales [4-5].

Los gobiernos de países desarrollados, particularmente Estados Unidos, la Unión Europea y China están financiando el trabajo en el área con la preocupación de que las computadoras cuánticas puedan dar al país que llegue primero una gran ventaja competitiva. Por ejemplo, la Ley de Iniciativa Cuántica Nacional de EEUU de 2019 (fecha de petición del presente proyecto de innovación docente) autorizó una financiación de \$ 1.2 mil millones durante los próximos 5-10 años. En el marco de la formación de los nuevos ingenieros TIC de la UCM, se hace pues necesaria una capacitación en programación de computadores cuánticos para aprovechar este nuevo paradigma de procesamiento de información y resultar competitivo en este escenario.

### 1.1 Computación cuántica en educación:

En este contexto, este nuevo paradigma de computación es un desafío para los estudiantes de ingeniería informática y disciplinas STEM. Se ha afirmado que es posible enseñar de manera efectiva computación cuántica a estudiantes universitarios sin conocimientos en física haciendo uso de los conocimientos de hardware y software estudiados a lo largo de la carrera, así como los conocimientos básicos de matemáticas elementales. Con estos postulados en mente, proponemos el presente proyecto basado en una suite de diferentes entornos de programación cuántica (tanto en simulación como entornos reales) basados en contenedores (dockers), para instalar en equipos de escritorio y placas de bajo coste y consumo como las Raspberry Pi 4, Odroid o similar. De esta forma que se podrá empaquetar el código de aplicación junto con sus dependencias en forma aislada, lo que permite a los usuarios poder ejecutar diferentes versiones de la misma aplicación cambiando una serie de parámetros. Esta característica proporciona escalabilidad, portabilidad y estandarización. De esta manera se puede implementar un entorno basado en cloud y ofrecer QAAS (Quantum as a Service). Los contenedores permiten una velocidad de procesamiento más rápida en comparación con las máquinas virtuales [6]. El lenguaje de entrada (Python) será común a todos los entornos. La programación se realizará en Jupyter Notebook [7], que es un entorno de código abierto basado en web con documentos que combinan tutoriales, la ejecución de código, texto, etc. Este entorno que hemos elegido para propósitos docentes se puede utilizar para la gestión de datos abierta (limpieza, transformación, visualización de datos), tareas estadísticas o aprendizaje automático, entre otros.

Cuando se necesite, estos contenedores se ejecutarán en la nube haciendo uso de instancias AWS proveyendo al estudiante una plataforma ágil y configurable. Los entornos considerados

para su evaluación, comparación y preparación serán los principales: Qiskit de IBM [8], Cirq de Google [9], PyQuil de Rigetti [10], Q # de Microsoft [11] y ProjectQ de ETH [12].

## 1.2 Objetivos del proyecto y propuestas de valor del mismo

Tenemos cuatro objetivos para este proyecto que serán abordados por un equipo formado por PDIs, alumnos, PAS y personas vinculadas al mundo empresarial.

Objetivo 1- Creación de un framework de trabajo basado en contenedores que puedan ejecutar instancias paralelas (tanto en simulación como en computadores reales) de los principales entornos de computación cuántica existentes: Qiskit de IBM, Cirq de Google, PyQuil de Rigetti , Q # de Microsoft y ProjectQ de ETH. Este marco de trabajo está inicialmente pensado para probarlo en una nueva asignatura optativa ofertada en varios grados de Informática (Grado en Ing. De Computadores, Grado en Ing. Informática, Grado Ing. Software, Doble Grado en Matemáticas e Ing. Informática) para el curso 2020-2021 y cuyo nombre es “ APCC (Arquitectura y Programación de Computadores Cuánticos)” [20].

Objetivo 2- Creación de una metodología docente de Computación Cuántica práctica para estudiantes TICs, sin requisitos adicionales de física o matemáticas. Esta metodología estará basada en varias etapas e intentará aunar la computación cuántica con la formación básica previa que tienen los estudiantes de TICs.

Objetivo 3- Creación de una suite de prácticas tanto para ejecución en computadores reales como para simulación. Estas prácticas estarán cuidadosamente seleccionadas para la explicación de conceptos relevantes dentro de este paradigma de computación. El nivel de dificultad será incremental.

Objetivo 4- Esta última parte evaluará el sistema construido y se planteará futuras extensiones. Se evaluarán los diferentes entornos de programación y se hará uso de la métrica de Halstead [19] para medir la complejidad y dificultad de codificación de circuitos cuánticos desde el punto de vista docente y del alumno. También se realizará una evaluación de la plataforma y obtención de estadísticas. Por último se abordará una diseminación científico-técnica, escritura de informes y planteamiento de alguna publicación en función de los resultados obtenidos.

## 2. Objetivos alcanzados:

Todos los objetivos alcanzados se han considerado en el entorno de la asignatura optativa “Arquitectura y Programación de Computadores Cuánticos (APCC)” [20] ofertada en varios grados de Informática (Grado en Ing. De Computadores, Grado en Ing. Informática, Grado Ing. Software, Doble Grado en Matemáticas e Ing. Informática) para el curso 2020-2021.

Objetivo 1- Se ha creado un entorno de trabajo basado en contenedores que pueden ejecutar instancias paralelas (tanto en simulación como en computadores reales) de los principales entornos de computación cuántica existentes: Qiskit de IBM, Cirq de Google, PyQuil de Rigetti, Q # de Microsoft y ProjectQ de ETH. Este framework funciona en placas tipo Raspberry 4 de forma reversible, tanto como cliente ligero de dockers (y ejecución por ende en la nube a través de instancias tipo AWS o similares), y también como servidor de docker, desplegando localmente los entornos requeridos, bibliotecas y códigos asociados. Este esquema se viene ilustrado en la figura 1.

Objetivo 2- Se ha elaborado una metodología docente de Computación Cuántica práctica para estudiantes TICs, sin requisitos adicionales de física o matemáticas. Esta metodología estará basada en cuatro etapas conceptuales y se puede esquematizar según la figura 2.

- Introducción y conceptos preliminares (I).
- Arquitectura, Tecnología e Interfaces de programación (II).
- Lenguaje de programación de alto nivel (basado en Python) y QASM (Quantum assembler) (III).
- Algoritmos relevantes para probar en este nuevo paradigma de computación que pongan de manifiesto sus principales ventajas e inconvenientes (IV).
- Computación cuántica VS Computación de altas prestaciones: Simuladores de Computación cuántica y su relación con el HPC. (V)

Objetivo 3- se han creado una suite de prácticas tanto para ejecución en computadores reales como para simulación. Estas prácticas se han seleccionado para la explicación de conceptos relevantes dentro de este paradigma de computación:

- Teletransportación cuántica.
- Deutsch-Josza, algoritmo de búsqueda simple [13] .
- Bernstein-Vazirani, algoritmo resolviendo un oráculo [14] .
- Simon, algoritmo basado en subgrupo oculto [15] .
- Transformada cuántica de Fourier [16] .
- Estimación de fase cuántica, una primitiva para muchos algoritmos cuánticos .

- Grover, algoritmo de búsqueda en bases de datos [17] .
- Shor, algoritmo para factorización [18] .

Objetivo 4- Esta última parte se ha evaluado el sistema construido y se planteado futuras extensiones. Se han evaluado los diferentes entornos de programación y se ha hecho uso de la métrica de Halstead [19] para medir la complejidad y dificultad de codificación de circuitos cuánticos desde el punto de vista del docente y del alumno. También se realizará una evaluación de la plataforma y obtención de estadísticas. Los resultados de este objetivo se esquematizan en las figuras 3-6 (localizadas en la sección de Desarrollo de esta memoria). Por último se ha abordado la diseminación científico-técnica, escritura de informes y planteamiento de alguna publicación en función de los resultados obtenidos. La metodología, resultados y análisis crítico así como las figuras de la presente memoria están extraídos de las siguientes publicaciones derivadas que se incluyen en la bibliografía final:

[22] Ginés Carrascal, Guillermo Botella, and Alberto A. Del Barrio. 2020. Comparison of container-based platforms for quantum computing simulation. In Proceedings of the 2020 Summer Simulation Conference (SummerSim '20). Society for Computer Simulation International, San Diego, CA, USA, Article 14, 1–14.

[23] Carrascal, G., del Barrio, A.A. & Botella, G. First experiences of teaching quantum computing. *J Supercomput* 77, 2770–2799 (2021). [https://doi.org/10.1007/s11227-020-03376-](https://doi.org/10.1007/s11227-020-03376-x)

x

### **3. Metodología empleada:**

La metodología desarrollada durante el desarrollo de este Proyecto de Innovación Docente ha sido la siguiente:

(I) Selección de nuevos dispositivos de procesamiento. Raspberry Pi 4, como sistema multicore de bajo coste y consumo apropiado para alojar los contenedores. Como alternativa PC o laptops con suficientes recursos. Se usan instancias en la nube tipo Amazon Web Service o similar.

(II) Selección de entornos de computación cuántica. Evaluación de complejidad del desarrollo de código en los principales entornos disponibles. (Qiskit de IBM, Cirq Google, PyQuil de Rigetti, Q # de Microsoft y ProjectQ de ETH). Como métrica de partida se usa la métrica de Halstead [19] que nos da idea del esfuerzo de código requerido. Como estímulo de entrada se usará entre otros el sumador de Vedral [21]. Se realiza también la usabilidad de los entornos mencionados.

(III) Construcción de los contenedores mediante tecnología docker. Se ha usado un repositorio tipo github o similar para la correcta organización y etiquetado de los mismos. Un esquema de la topología del mismo se puede apreciar en la figura 1.

(IV) Construcción de un front-end que parametrize, seleccione y configure nuestro sistema de desarrollo. Este front-end ejecutará el despliegue paralelo los contenedores tanto en la nube mediante instancias AWS como de forma local en las placas tipo Raspberry 4, evaluando los recursos disponibles.

(V) Evaluación del sistema según la metodología y suite de prácticas descritas en el apartado anterior. Preparación de guías de usuario inclusivas para configurar los proyectos en el entorno desarrollado teniendo en cuenta los diferentes escenarios (Ejecución usando instancias AWS, ejecución local).

(VI) Validación por parte de alumnado, profesorado y miembros del PIMCD. Medida de satisfacción de usuario final. Extensión y adaptación según el escenario Covid-19 encontrado. Disseminación de resultados con informes técnicos y/o artículo para la comunidad educativa. Debido a que el lenguaje que se usará (Python) está muy extendido y arraigado y otras disciplinas (Matemáticas, Física, Ingeniería), planteamiento de portabilidad de este proyecto a otras carreras afines. El entorno construido cumplirá los estándares de accesibilidad para personas con visión limitada (texto, fuentes, imágenes, análisis de contraste, listas, columnas, tablas, hipervínculos, etc. así como creación de contenidos accesibles multimedia).

### **4. Recursos humanos:**

Tal y como se ha puesto de manifiesto en la descripción del presente proyecto docente, este proyecto de 10 participantes desglosados en 3 docentes de la Facultad de Informática, 4 alumnos (que cursan Máster en la fac de Informática y el Doctorado en la misma), 1 Personal de Administración y Servicios (Técnico especialista de la FDI) y 2 personas vinculadas a la empresa (Mathworks e IBM) .

## 5. Desarrollo del proyecto:

Se adjunta el esquema de contenedores propuesto en la figura 1. Se usan los principales entornos disponibles. (Qiskit de IBM, Cirq de Rigetti, PyQuil de Google, Q # de Microsoft y ProjectQ de ETH).

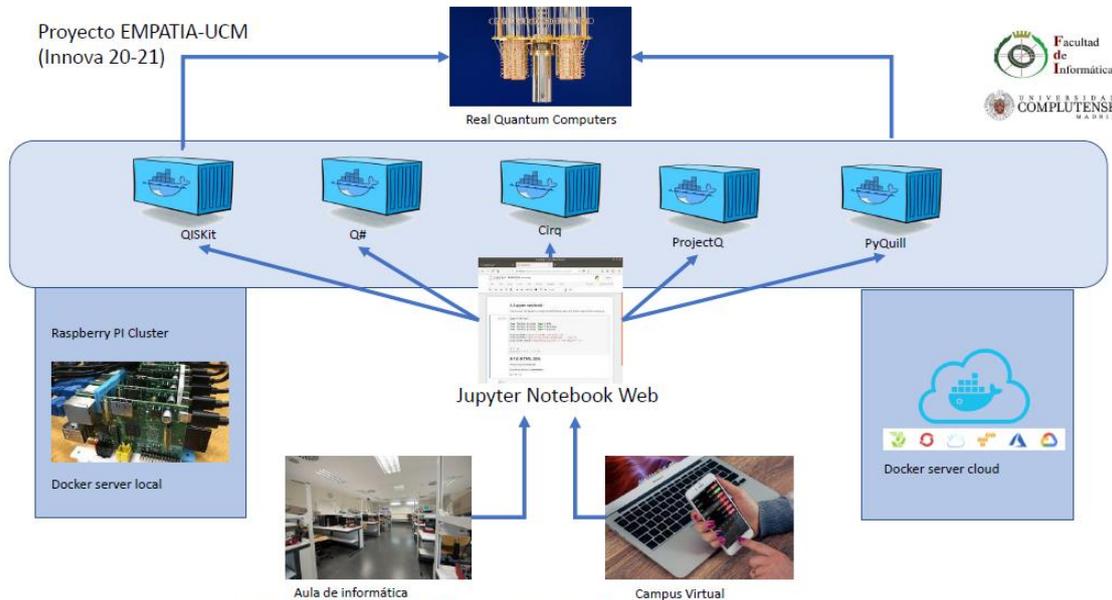


Figura 1. Arquitectura del sistema propuesto.

Adicionalmente las metodologías de enseñanza propuesta y aplicada en la asignatura "Arquitectura y Programación de Computadores Cuánticos (APCC)" se ilustra en la figura 2.

Proyecto EMPATIA-UCM  
(Innova 20-21)

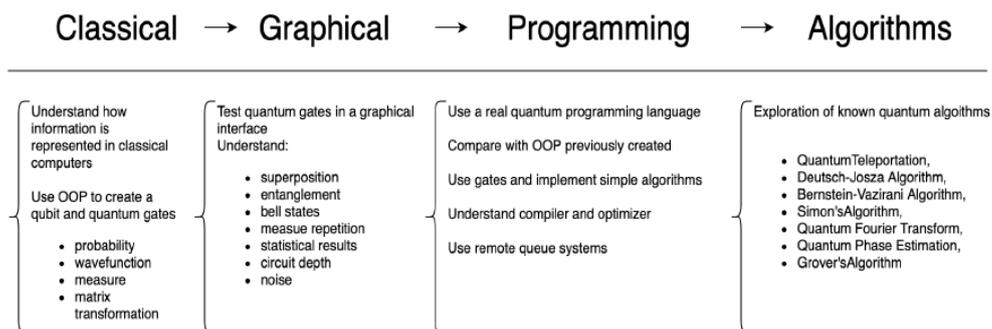


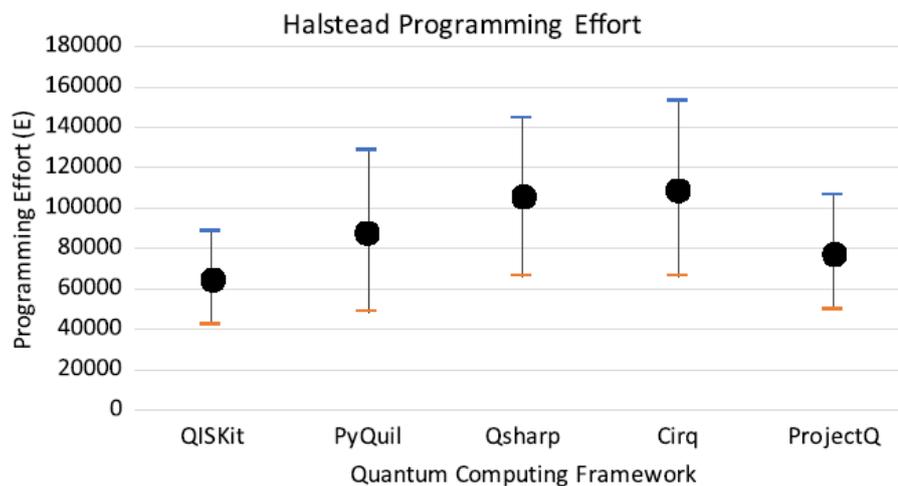
Figura 2. Metodología Propuesta. Extraída de: Carrascal, G., del Barrio, A.A. & Botella, G. First experiences of teaching quantum computing. *J Supercomput* (2020). <https://doi.org/10.1007/s11227-020-03376-x>

La métrica de Halstead para medir la dificultad de codificar los programas queda plasmada en las siguientes Figuras 3-4.

Figura 3

	Qiskit	PyQuil	MQDK	Cirq	ProjectQ
Program length ( $L$ )	430.33	503.00	475.67	537.50	487.50
Program vocabulary ( $V$ )	62.00	67.00	59.67	66.00	66.00
Volume ( $VL$ )	2561.15	3050.04	2807.17	3248.86	2949.18
Difficulty ( $D$ )	24.97	27.98	36.89	33.00	25.83
Effort ( $E$ )	65,190.84	88,124.73	105,740.82	109,373.31	77,473.72

Figura 4.



Figuras 3-4. Aplicación de parámetro de las métricas de Haslthead al sumador de Vedral.  
 Extraída de: Carrascal, G., del Barrio, A.A. & Botella, G. First experiences of teaching quantum computing. J Supercomput (2020). <https://doi.org/10.1007/s11227-020-03376-x>

Como se desprende de las figuras (3-4) y estudios anteriores, se sugiere que Qiskit (IBM) puede ser un lenguaje idóneo debido a la curva de aprendizaje y el bajo esfuerzo respecto a otras alternativas. Por lo tanto el grueso de las prácticas se diseñarán para que se use este entorno de programación.

Por otro lado, se han diseñado encuestas docentes cuyo resultado se plasma en las figuras (5-7), en ellas se puede ver un alto nivel de satisfacción respecto a los conceptos explicados e interés sobre los temas relacionados. La UCM, se ha visto beneficiada al disponer de un proyecto piloto que evalúe estas técnicas en la primera asignatura ofertada de computación cuántica práctica su adecuación paulatina a las diferentes asignaturas de ciencia y tecnología.

Figura 5

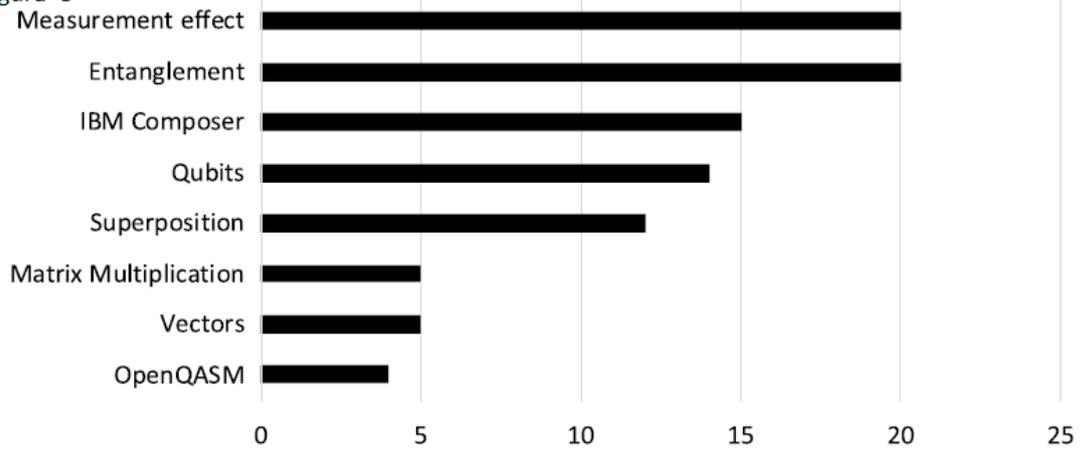


Figura 6

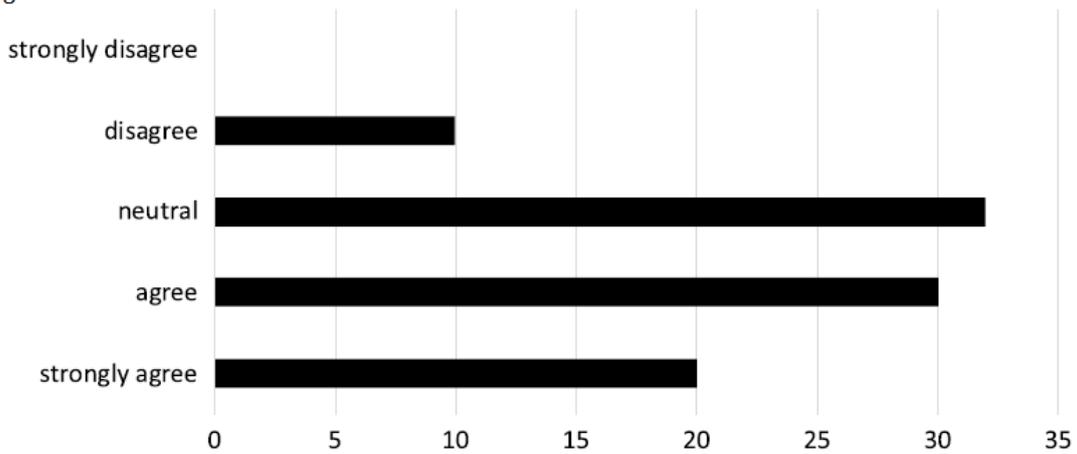
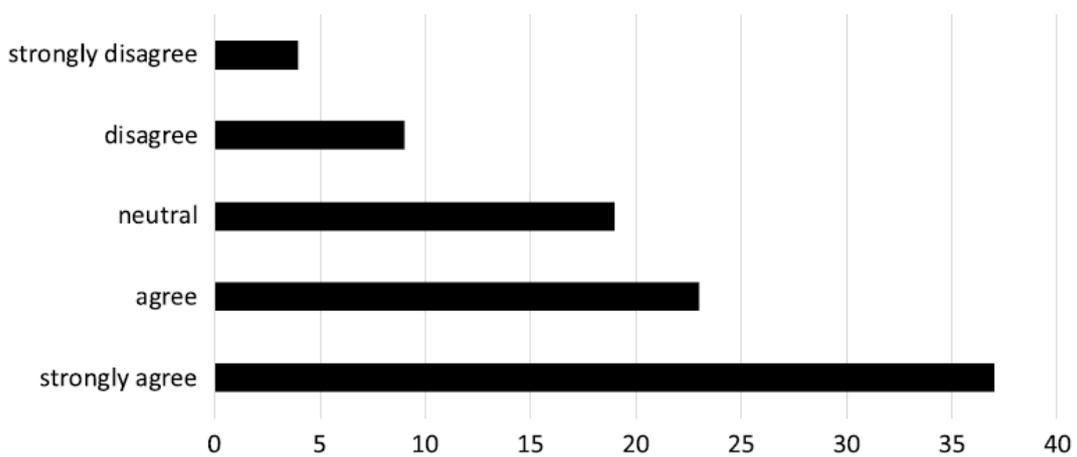


Figura 7



Figuras 5-7. Resultados de las encuestas aplicadas. Temas que se han encontrado más interesantes por parte de alumnado (Figura 5). Opiniones acerca de si las herramientas computacionales ayudan a comprender la naturaleza cuántica (Figura 6.). Opiniones sobre la atención que despierta este tema. (Figura 7).

Extraída de: Carrascal, G., del Barrio, A.A. & Botella, G. First experiences of teaching quantum computing. *J Supercomput* (2020). <https://doi.org/10.1007/s11227-020-03376-x>

## Bibliografía del Proyecto innova “EMPATIA” 2020-2021

- [1] R. B. Griffiths, *Consistent Quantum Theory*. Cambridge University Press, 2003.
- [2] “Quantum Mechanics, Interpretations of | Internet Encyclopedia of Philosophy.” <https://www.iep.utm.edu/int-qm/> (accessed Apr. 18, 2020).
- [3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [4] “IBM Unveils World’s First Integrated Quantum Computing System for Commercial Use - Jan 8, 2019.” <https://newsroom.ibm.com/2019-01-08-IBM-Unveils-Worlds-First-Integrated-Quantum-Computing-System-for-Commercial-Use>. (acceso 21/10/2021)
- [5] Association for Computing Machinery [ACM] / Institute of Electrical and Electronics Engineers [IEEE], “Computer Engineering Curricula 2016,” in *Computer Engineering Curricula 2016. CE2016. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, New York, Estados Unidos: IEEE, ACM., 2016.
- [6] B. Varghese, L. T. Subba, L. Thai, and A. Barker, “Container-Based Cloud Virtual Machine Benchmarking,” in *2016 IEEE International Conference on Cloud Engineering (IC2E)*, Apr. 2016, pp. 192–201, doi: 10.1109/IC2E.2016.28.
- [7] “Project Jupyter.” <https://www.jupyter.org> (acceso 21/10/2021).
- [8] “Qiskit.” <https://qiskit.org/> (acceso 21/10/2021).
- [9] “Announcing Cirq: An Open Source Framework for NISQ Algorithms,” *Google AI Blog*. <http://ai.googleblog.com/2018/07/announcing-cirq-open-source-framework.html> (acceso 21/10/2021).
- [10] “Welcome to the Docs for the Forest SDK! — pyQuil 2.16.0 documentation.” <http://docs.rigetti.com/en/stable>(acceso 21/10/2021).
- [11] “Quantum Development Kit | Microsoft,” *Microsoft Quantum - US (English)*. <https://www.microsoft.com/en-us/quantum/development-kit> (acceso 21/10/2021).
- [12] damian\_projectq, “ProjectQ,” *ProjectQ*. <https://projectq.ch/> (acceso 21/10/2021)
- [13] D. Deutsch and R. Jozsa, “Rapid Solution of Problems by Quantum Computation,” *Proceedings of the Royal Society of London Series A*, vol. 439, pp. 553–558, Dec. 1992, doi: 10.1098/rspa.1992.0167.
- [14] Ethan. Bernstein and Umesh. Vazirani, “Quantum Complexity Theory,” *SIAM J. Comput.*, vol. 26, no. 5, pp. 1411–1473, Oct. 1997, doi: 10.1137/S0097539796300921.
- [15] D. R. Simon, “On the Power of Quantum Computation,” *SIAM Journal on Computing*, vol. 26, pp. 116–123, 1994.
- [16] Coppersmith, Don. “An approximate Fourier transform useful in quantum factoring”, IBM Research Report RC19642 ,; R. Cle.” (1994).
- [17] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *arXiv:quant-ph/9605043*, Nov. 1996, (acceso 21/10/2021). [Online]. Available: <http://arxiv.org/abs/quant-ph/9605043>.
- [18] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Nov. 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.
- [19] Chang Z, Song R, Sun Y (2018) Validating halstead metrics for scratch program using process data. In: 2018 IEEE Int. Conference on Consumer Electronics-Taiwan (ICCE-TW), May 2018, pp 1–5. <https://doi.org/10.1109/icce-china.2018.8448607>
- [20] [Online] <https://www.ucm.es/estudios/grado-ingenieriadecomputadores-plan-804999> (acceso 21/10/2021)
- [21] Vedral V, Barenco A, Ekert A (1996) Quantum networks for elementary arithmetic operations. *Phys Rev A* 54(1):147–153. <https://doi.org/10.1103/PhysRevA.54.147>
- [22] Ginés Carrascal, Guillermo Botella, and Alberto A. Del Barrio. 2020. Comparison of container-based platforms for quantum computing simulation. In *Proceedings of the 2020 Summer Simulation Conference (SummerSim '20)*. Society for Computer Simulation International, San Diego, CA, USA, Article 14, 1–14.
- [23] Carrascal, G., del Barrio, A.A. & Botella, G. First experiences of teaching quantum computing. *J Supercomput* 77, 2770–2799 (2021). <https://doi.org/10.1007/s11227-020-03376-x>