

Entornos de Desarrollo

Actividad de Investigación: Computación y Software Cuánticos

Introducción: Willow, Heron y Qiskit

La computación cuántica es un campo interdisciplinar que combina principios de la Física Cuántica, las Matemáticas y la Informática con el objetivo de desarrollar lo que a efectos prácticos es el siguiente nivel y el futuro inmediato de todo lo que esté relacionado con la computación, que es el aprovechamiento de las propiedades únicas de los sistemas cuánticos, que procederemos a comentar en breve.

En el momento en que este documento está siendo redactado, Google acaba de anunciar el lanzamiento del primer chip cuántico, Willow, que podría multiplicar la rapidez y eficiencia de un superordenador en una escala infinitamente mayor a la que supuso el salto de los primeros ordenadores a los que conocemos ahora. De acuerdo con el propio fundador y líder de Google Quantum AI, Hartmut Neven, “puede reducir errores de forma exponencial a medida que escalamos utilizando más qubits. Esto elimina un desafío clave en la corrección de errores cuánticos perseguida en este campo durante casi 30 años”

“Willow rindió en cinco minutos a un estándar computacional que le tomaría a uno de los superordenadores más veloces de hoy 10 septillones (es decir, 10²⁵) años - un número que sobrepasa la edad del Universo”.

Por su parte, IBM ha anunciado en su propia web el lanzamiento del procesador Heron y el software de programación Qiskit:

“...Un ordenador cuántico capaz de arrancar circuitos cuánticos de hasta 5.000 operaciones de dos Qubits (...) Gracias a los más recientes avances en hardware, middleware y software, Heron ahora puede llevar a cabo cálculos precisos empleando circuitos con 5.000 puertas de dos Qubits”

“Qiskit Functions es un servicio de programación que permite acceso a hardware y software cuántico de alto rendimiento a un nivel más alto de abstracción”.

Podemos detectar, por tanto, un patrón común en ambas noticias: la palabra “Cuántico” y “Qubits” como parte del vocabulario común en el nuevo paradigma de la ciencia computacional que está por llegar.

¿Qué son los Qubits? Características principales y diferencias con el bit

A grandes rasgos, el Qubit (Quantum Bit) o Cúbit es la unidad básica de información de la computación cuántica, del mismo modo que el bit clásico es la unidad mínima de información en la computación tradicional. El término Qubit es atribuido al físico teórico estadounidense Benjamin Schumacher, quien en 1994 lo definió tal como acabamos de mencionar.

Mientras que un bit es un valor binario que sólo admite dos valores, verdadero (1) o falso (0), el Qubit puede estar en una superposición de ambos estados al mismo tiempo, permitiendo realizar cálculos de forma paralela. Hablando en términos muy técnicos, un qubit puede representarse como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

donde α y β son amplitudes complejas que satisfacen $|\alpha|^2 + |\beta|^2 = 1$. Las probabilidades de medir el qubit en el estado $|0\rangle$ o $|1\rangle$ son $|\alpha|^2$ y $|\beta|^2$, respectivamente.

Hablando en plata, podemos en cambio decir que si un Bit es una moneda, esta solo puede ser cara o cruz. Un Qubit sería una moneda girando, y su valor solo se decide a la hora de “mirar” la moneda, o medir el Qubit.

Características:

1) La idea de “medir” el Qubit implica obligar al sistema cuántico a decidir un valor 0 o 1. Esta cualidad se conoce como **superposición**, una combinación 50/50 de ambos estados. Dado que un bit clásico solo puede ser un 0 o un 1 en un momento dado, la cantidad de combinaciones numéricas que puedes representar es mucho más limitada que con un Qubit:

- Si tuviéramos 3 bits en un ordenador clásico, solamente podríamos representar uno de estos números a la vez: 000, 001, 010, 111, etc en cada paso del cálculo.
- En cambio la superposición de un Qubit permitiría representarlos todos al mismo tiempo en un mismo cálculo, permitiendo llevar a cabo una infinidad de estados o combinaciones posibles en un solo cálculo.

Si utilizáramos un rompecabezas como alegoría, el bit iría probando con cada pieza una por una hasta ir encajándolas todas. El qubit probaría todas las posibles combinaciones a la vez. Esto no significa que pueda obtener todos los resultados simultáneamente, aunque mediante el uso de técnicas avanzadas como la interferencia cuántica, podemos manipular las probabilidades para que el resultado correcto sea más probable.

Esta es la clave del salto masivo que la computación cuántica supone de la tradicional, siendo que, por ejemplo, 10 qubits pueden procesar 1024 estados posibles al mismo tiempo.

2) Entrelazamiento :

El entrelazamiento cuántico consiste en la dependencia de dos partículas cuánticas entre sí de manera independiente a la distancia existente entre estas. Esto implica que al medir el estado de un Qubit, conocemos al instante el estado del otro que está entrelazado con este.

Este fenómeno tiene lugar no debido a una transmisión de la información, sino a **una correlación intrínseca en los estados** de ambas partículas relacionadas.

En términos cuánticos, esto sería como tener dos qubits entrelazados en el estado:

$$|\psi\rangle = \frac{1}{\sqrt{2}}$$

$$(|01\rangle + |10\rangle)$$

$$(|01\rangle + |10\rangle)$$

Aquí:

- $|01\rangle$ significa que el primer qubit está en $|0\rangle$ y el segundo en $|1\rangle$.
- $|10\rangle$ significa lo contrario: el primer qubit está en $|1\rangle$ y el segundo en $|0\rangle$.

Si mides el primer qubit y obtienes $|0\rangle$, inmediatamente sabes que el segundo qubit debe estar en $|1\rangle$, sin necesidad de medirlo explícitamente.

Si usáramos de nuevo la metáfora de las monedas, podríamos imaginar que si una moneda dentro de una caja se encontrara en “cara”, esa misma moneda en otra caja, sin importar la distancia, estaría automáticamente en “cruz”, y viceversa. La manera en que ambos estados son desplazados de uno a otro mediante el entrelazamiento, es conocida como “teletransporte cuántico”, una de las aplicaciones clave en el campo de la computación cuántica.

En lo que a nosotros concierne, la razón por la que esta característica del Qubit es tan importante en la computación es que puede utilizarse para transferir información entre diferentes partes de un computador cuántico o incluso entre computadoras cuánticas separadas físicamente de forma segura.

El entrelazamiento también juega un papel importante en la corrección de errores cuánticos. Los qubits son extremadamente sensibles a las perturbaciones ambientales, lo que puede llevar a errores en los cálculos. Sin embargo, mediante técnicas de codificación cuántica que utilizan el entrelazamiento, es posible detectar y corregir estos errores sin medir directamente el estado de los qubits, lo que preserva la información cuántica.

3) Interferencia cuántica :

Tratando de explicarlo en los términos más simples posibles, es el proceso mediante el cual las probabilidades asociadas con diferentes trayectorias o estados cuánticos se combinan de manera constructiva o destructiva, afectando el resultado final de una medición. Este fenómeno permite que los computadores cuánticos exploren múltiples posibilidades simultáneamente y amplifiquen las soluciones correctas mientras suprimen las incorrectas.

La interferencia cuántica puede ser:

- Constructiva : Cuando las amplitudes de probabilidad se suman, aumentando la probabilidad de ciertos resultados.
- Destructiva : Cuando las amplitudes de probabilidad se cancelan, disminuyendo la probabilidad de ciertos resultados.

La manera en que esta característica se comporta se puede ilustrar con el ejemplo técnico de la doble rendija: Si se disparan partículas a través de una doble rendija, estas no pasarían por una de ellas sino que interferirían consigo mismas para pasar por ambas al mismo tiempo.

La interferencia cuántica es lo que permite a los algoritmos cuánticos concentrar la probabilidad en las soluciones correctas y minimizarla en las incorrectas. De todos estos algoritmos, el más conocido de ellos es el **Algoritmo de Glover**.

Conocido como la mejor aplicación práctica de la interferencia cuántica en la resolución de problemas, este algoritmo está diseñado para buscar un elemento específico en una base de datos no ordenada y manipular de manera sistemática de todas las probabilidades posibles haciendo uso de la interferencia cuántica. Lo interesante de este algoritmo es que explota también las cualidades de superposición y entrelazamiento al iniciar con todos los qubits en una superposición uniforme de todos los posibles estados. Esto nos sirve para entender un poco mejor las complejidades de las relaciones que las características del Qubit guardan entre sí.

El de Glover no es el único algoritmo a destacar que explote la interferencia, existen otros como el de **Shor** (para identificar los períodos de funciones periódicas, lo que permite factorizar números enteros grandes en tiempo polinómico), el de **Deutsch-Jozsa** (determina si una función es constante o balanceada necesitando una sola evaluación) o el de **HHL** (resuelve sistemas de ecuaciones lineales de manera mucho más rápida que los métodos clásicos)

Para finalizar, con respecto a la interferencia en ordenadores clásicos, hay tres diferencias a destacar, siendo la **naturaleza de las ondas** mecánicas o electromagnéticas (en los ordenadores cuánticos estas ondas son abstractas y representan amplitudes de probabilidad), la necesidad de un **control más preciso** y el **colapso de superposición**.

La interferencia cuántica tiene algunas limitaciones:

Desfasamiento (decoherencia) : Las interacciones con el entorno pueden perturbar las amplitudes de probabilidad, degradando la interferencia y causando errores en los cálculos.

Ruido cuántico : Los errores introducidos por el ruido en los sistemas cuánticos pueden dificultar mantener la coherencia necesaria para la interferencia efectiva.

Escalabilidad : A medida que aumenta el número de qubits, mantener la interferencia precisa se vuelve más difícil debido a la creciente complejidad del sistema.

4) Colapso de la función de onda

Como dijimos antes, si un Qubit fuese una moneda girando, medir un sistema cuántico implica colapsar el sistema para forzar dicha moneda a ser cara o cruz.

Este “giro” de moneda se conocería como estado de **superposición** (combinación de múltiples estados posibles), y forzarla a adoptar un estado es lo que conocemos como colapso de la función de onda.

No obstante, este colapso es aleatorio y sus posibilidades son de 50/50, ya que depende de las amplitudes cuánticas del estado antes de la medición. Esto significa que no podemos observar directamente la superposición, ya que en el momento de colapsar ya no se encuentra en este estado.

En cuanto al tema de la computación cuántica, esta cualidad es crucial en la corrección de errores cuánticos, ya que nos permite medir un bit para detectar y corregir fallos sin destruir toda la información contenida en el sistema. Además, nos permite implementar algoritmos cuánticos donde la medición final determina el resultado del cálculo.

¿Cómo son los primeros lenguajes de programación cuántico?

Existen varios lenguajes de programación cuánticos como **Ocean**, **Cirq** o **Q#**. No obstante, el que por ahora parece ser el más consistente y popular es **Qiskit**.

Desarrollado por IBM, hemos mencionado anteriormente Qiskit como una de las plataformas más populares para la programación en el ámbito de la computación cuántica. Sin embargo, hay que tener en cuenta que Qiskit, más que un lenguaje de programación en sí mismo, es un ecosistema de código abierto que proporciona una biblioteca basada en Python con las herramientas necesarias para la creación, simulación y ejecución de circuitos cuánticos.

Qiskit posee una interfaz Python intuitiva y es compatible con el hardware cuántico. Además, ofrece un conjunto de módulos adicionales para extender sus capacidades de optimización y aprendizaje de máquina, junto con una comunidad bastante activa.

A continuación, un ejemplo de programación en Qiskit:

```
from qiskit import QuantumCircuit, transpile, assemble, Aer, execute
from qiskit.visualization import plot_bloch_multivector, plot_histogram

# Crear un circuito cuántico con 2 qubits y 2 bits clásicos
qc = QuantumCircuit(2, 2)

# Aplicar una puerta Hadamard al primer qubit
qc.h(0)

# Entrelazar los dos qubits con una puerta CNOT
qc.cx(0, 1)

# Medir los qubits
qc.measure([0, 1], [0, 1])

# Simular el circuito
simulator = Aer.get_backend('qasm_simulator')
job = execute(qc, simulator, shots=1000)
result = job.result()
counts = result.get_counts(qc)

# Mostrar los resultados
print(counts)
```

La estructura de Qiskit se compone de los siguientes módulos:

Qiskit Terra es el núcleo que contiene las herramientas básicas necesarias para construir circuitos cuánticos y ejecutarlos en hardware cuántico real o simulado. También incluye funcionalidades para definir operaciones cuánticas, medir resultados y optimizar circuitos. Ofrece además una API (Interfaz de Programación de Aplicaciones) que permite manipular y visualizar circuitos cuánticos fácilmente.

Qis Aer se centra en la simulación de circuitos cuánticos con el fin de modelar distintos tipos de comportamientos. Esto permite analizar posibles escenarios de comportamientos no esperados en circunstancias no ideales y aprender a depurar algoritmos cuánticos.

Qiskit Ignis ofrece métodos para cuantificar y corregir errores debido a factores como el ruido ambiental, con el fin de aumentar la fiabilidad de los cálculos cuánticos.

Qiskit Aqua proporciona aplicaciones cuánticas listas para usar en diversos campos, como química, optimización y aprendizaje automático. Esto permite a los usuarios implementar algoritmos cuánticos avanzados sin tener que preocuparse demasiado por los detalles internos del hardware.

Qiskit IBM Quantum Provider permite conectar al usuario con los dispositivos cuánticos reales de [IBM Quantum](#) y almacenar y recuperar su trabajo en procesadores en nube.

Qiskit Metal permite a los usuarios diseñar y simular dispositivos cuánticos físicos, módulo de especial utilidad para investigadores.

¿Hay alguna especie de IDE ya para cuántica?

En el anterior apartado hemos hablado de **Qiskit** siendo el lenguaje de programación cuántica más popular (ciertamente no el único ni necesariamente el mejor, pero sin duda el que por ahora parece ser el estándar más aceptado), y por consistencia comentaremos el IDE desarrollado por IBM como parte del ecosistema Qiskit y para su utilización, **IBM Quantum Lab**.

Diseñado específicamente para trabajar con hardware cuántico real y simulaciones, es una plataforma basada en la nube y uno de los IDEs más completos para la programación cuántica. Sus características más importantes son:

Interfaz web: Accesible desde cualquier navegador moderno.

Notebooks Jupyter integrados: Permite escribir y ejecutar código Python con Qiskit directamente en el navegador.

Acceso a hardware cuántico: Facilita el envío de trabajos a procesadores cuánticos reales de IBM Quantum.

Simuladores avanzados: Incluye simuladores cuánticos para probar algoritmos antes de ejecutarlos en hardware real.

Visualización de circuitos: Ofrece herramientas para dibujar y analizar circuitos cuánticos gráficamente.

Colaboración: Permite compartir notebooks y proyectos con otros usuarios.

Además, es un entorno de fácil acceso que no requiere la instalación de software local, y que se integra directamente con el hardware cuántico de IBM, abundante en recursos educativos y tutoriales.

No obstante requiere una conexión a Internet constante y posee algunas limitaciones en términos de personalización del entorno.

Otros entornos de desarrollo dignos de mencionar son Microsoft Quantum Development Kit, Quirk, Forest SDK (que emplea el lenguaje PyQuil), Google Cirq (utiliza Colab), Quantum Computer Playground o XACC (eXtreme-scale ACCelerator)

¿Qué es un QPath?

Un QPath es un término algo vago que puede emplearse para describir varios aspectos de la computación cuántica, que desglosaremos a continuación.

Ruta Cuántica (Quantum Path): Representa una de las múltiples trayectorias posibles en un sistema cuántico, teniendo en cuenta que las partículas cuánticas no siguen trayectorias clásicas definidas sino que existen en superposiciones de múltiples caminos posibles.

En algunos algoritmos, como el anteriormente mencionado de Glover, una ruta cuántica puede referirse a secuencias de estados cuánticos que buscan elementos en bases de datos no estructuradas.

También puede referirse a la ruta óptima para transmitir qubits entre nodos de una red.

Circuitos Cuánticos (Quantum Circuit Paths): Se refiere a las conexiones lógicas entre puertas cuánticas dentro de un circuito. Estas deben aplicarse en un orden específico debido a sus dependencias temporales. En este sentido, el Qpath es la secuencia óptima de operaciones para minimizar errores o maximizar eficiencia.

Rutas en Grafos Cuánticos: En problemas de grafos cuánticos, donde los nodos y aristas son manipulados mediante algoritmos cuánticos, un "QPath" podría referirse a una ruta específica en el grafo. Esto sería relevante de cara a solucionar problemas de ruteo : En redes de transporte o comunicación, un "QPath" podría ser la ruta óptima y más eficiente encontrada mediante un algoritmo cuántico.

Plataformas o Herramientas Cuánticas: Es posible que "QPath" sea el nombre de una herramienta, plataforma o framework específico desarrollado para facilitar algún aspecto de la programación o simulación cuántica. Tal puede ser el caso de un entorno de desarrollo integrado (IDE) o un simulador cuántico que permite diseñar y probar circuitos cuánticos. (En contextos educativos, "QPath" podría referirse a una guía o camino estructurado para aprender computación cuántica, similar a cómo otros términos como "learning path")

Temperatura ideal de un ordenador cuántico y cómo lograrlo:

Debido a la fragilidad del comportamiento cuántico y su alta sensibilidad a factores ambientales (de los que ya hemos hecho mención anteriormente) que pueden causar el problema conocido como **decoherencia** (lo que provoca la destrucción de la información almacenada en los qubits), estos ordenadores necesitan trabajar a una temperatura cercana al **cero absoluto en grados Kelvin**.

Esta es la única manera de preservar las partículas cuánticas, reducir el ruido térmico, preservando la coherencia y minimizando los errores.

La estrategia actual para lograr esto es desarrollar computadoras cuánticas que puedan operar a temperaturas más cercanas a la temperatura ambiente . [Aunque esto sigue siendo un desafío considerable](#), hay avances prometedores en varios frentes:

Superconductores de alta temperatura crítica : Si se descubren [nuevos materiales superconductores que funcionen a temperaturas más altas](#), podrían revolucionar la computación cuántica.

Qubits topológicos y de spin : Estos qubits ofrecen una vía hacia sistemas más robustos frente al ruido térmico.

Miniaturización y optimización : Avances en la fabricación de dispositivos cuánticos podrían reducir la sensibilidad al ruido y permitir operaciones a temperaturas más altas.

También debemos tener en cuenta estrategias de cara a la corrección y minimización de errores con el fin de desarrollar que sacrifiquen estabilidad a cambio de resistencia a las fluctuaciones térmicas.

¿Qué es un algoritmo cuántico?

Anteriormente, hablando de las características del qubit cuántico, hemos mencionado varios algoritmos cuánticos, entre ellos el de Grover. Un algoritmo cuántico es un conjunto de instrucciones diseñado específicamente para ser ejecutado en una computadora cuántica, aprovechando los principios fundamentales de la mecánica cuántica que ya hemos comentado en los apartados anteriores. Estos algoritmos están pensados para resolver problemas que serían extremadamente difíciles o ineficientes de abordar con computadoras clásicas. Como también hemos dicho ya, estos algoritmos trabajan con qubits en lugar de bits, con todas las ventajas que esto conlleva de cara a la resolución de problemas y la transmisión de información o la realización de cálculos.

Volvamos brevemente sobre los algoritmos cuánticos más conocidos:

Algoritmo de Shor

Propósito : Factorizar números enteros grandes en sus factores primos.

Impacto : Es especialmente relevante para la criptografía, ya que muchos sistemas de cifrado modernos (como RSA) dependen de la dificultad de factorizar números grandes.

Ventaja cuántica : El algoritmo de Shor puede resolver este problema en tiempo polinómico, mientras que los mejores algoritmos clásicos requieren tiempo exponencial.

Algoritmo de Grover

Propósito : Buscar un elemento específico en una base de datos no estructurada.

Impacto : Aunque no ofrece una ventaja exponencial como Shor, reduce el tiempo de búsqueda.

Aplicaciones : Útil en problemas de optimización, inteligencia artificial y bases de datos.

Algoritmo de Deutch-Jozsa

Propósito : Determinar si una función dada es constante o balanceada.

Impacto : Demostró por primera vez que un algoritmo cuántico podía resolver un problema con menos consultas que cualquier algoritmo clásico.

Significado teórico : Fue uno de los primeros ejemplos que mostraron la potencial superioridad de los algoritmos cuánticos.

Simulación cuántica

Propósito : Simular sistemas cuánticos complejos, como moléculas y materiales.

Impacto : Permitirá modelar reacciones químicas y propiedades físicas que son imposibles de simular con precisión en computadoras clásicas.

Aplicaciones : Desarrollo de nuevos medicamentos, materiales avanzados y catalizadores.

Algoritmo de Variational Quantum Eigensolver (VQE)

Propósito : Resolver problemas de optimización y encontrar los valores propios mínimos de operadores cuánticos.

Impacto : Es útil en química cuántica para calcular energías de enlace molecular y otras propiedades físicas.

Característica : Combina técnicas clásicas y cuánticas para funcionar en hardware cuántico actual con limitaciones.

Algoritmo QAOA (Quantum Approximate Optimization Algorithm)

Propósito : Resolver problemas de optimización combinatoria.

Impacto : Tiene aplicaciones en logística, planificación y finanzas.

Característica : Está diseñado para funcionar en computadoras cuánticas de "media escala" con ruido.

Bibliografía:

Google Blog. (2023). *Google Willow Quantum Chip* .

<https://blog.google/technology/research/google-willow-quantum-chip/>IBM. (s.f.). *Qubit* . IBM Topics.

<https://www.ibm.com/es-es/topics/qubit>

EL PAÍS. (2023). *Google presenta Willow, un chip cuántico que resuelve en 5 minutos una tarea que un superordenador tardaría cuatrillones de años* . Tecnología.

<https://elpais.com/tecnologia/2023/09/14/actualidad.html>

IBM Quantum Computing Blog. (2022). *IBM Quantum delivers on 2022 100x100 performance challenge* .

<https://research.ibm.com/blog/ibm-quantum-performance-challenge>

IBM Quantum Computing Blog. (2023). *Announcing the Qiskit Functions Catalog* .

<https://research.ibm.com/blog/qiskit-functions-catalog>

QuantumPath. (s.f.). *Introducción al desarrollo de software cuántico con QPath* . QuantumPath. Nota: Este parece ser un título personalizado; si no hay información adicional, puedes usar "s.f." para indicar que no se especifica la fecha.

The Quantum Insider. (2022, julio 28). *State of Quantum Computing Programming Languages in 2022* .

<https://thequantuminsider.com/2022/07/28/state-of-quantum-computing-programming-languages-in-2022/>

The Quantum Insider. (2023, marzo 24). *Quantum Computing Challenges* .

<https://thequantuminsider.com/2023/03/24/quantum-computing-challenges/>

The Quantum Insider. (2023, agosto 3). *How Would Room-Temperature Superconductors Change Quantum Computing?*

<https://thequantuminsider.com/2023/08/03/how-would-room-temperature-superconductors-change-quantum-computing/>

MLearningLab. (2022, noviembre 16). *Algoritmos cuánticos y la estructura necesaria* .

<https://mlearninglab.com/2022/11/16/algoritmos-cuanticos-y-la-estructura-necesaria/>