



BAHIA SOFTWARE

CTS-2022-0054

Arquitectura

Miguel Boubeta
Sergio Figueiras
Román López
Luis Lorenzo

15/05/2023



*Unha maneira
de facer Europa.*



Fondos Europeos



Despregamento dunha infraestrutura baseada en tecnoloxías cuánticas da información que permita impulsar a I+D+i en Galicia.

Apoiar a transición cara a una economía dixital.

Operación financiada pola Unión Europea, a través do FONDO EUROPEO DE DESENVOLVEMENTO REXIONAL (FEDER) como parte da resposta da Unión á pandemia da COVID-19

Tabla de contenidos

1. Introducción	7
2. Arquitectura global	9
2.1 Frontales	10
2.2 Sistema de HPC	11
2.3 Recursos cuánticos	11
2.4 Acceso a recursos externos	12
3. Retos	13
3.1 Control de acceso	13
3.2 Gestión de colas	13
3.2.1 Problemática actual	13
3.2.2 Propuesta y mejoras	14
3.3 Interfaces de comunicación	15
3.3.1 Comunicaciones mediante las interfaces de red	15
3.3.2 Comunicaciones mediante un bus	18
3.4 Integración con recursos externos	19
3.5 Representación de los circuitos cuánticos	23
3.6 Representación de los resultados	29
3.7 Gestión de la contabilidad	44
4. Conclusiones	47
4.1 Control de acceso	49
4.2 Gestión de colas	49
4.3 Interfaz de comunicación vía driver	50
4.4 Interfaz de comunicaciones en red	50
4.5 Integración con recursos externos	50
4.6 Representación de los circuitos	51
4.7 Plataforma analítica de los resultados	51
4.8 Representación de los resultados	51
4.9 Gestión de la contabilidad	52

Lista de figuras

Figura 1: Historia de la computación cuántica.....	7
Figura 2: Arquitectura propuesta para la integración de las QPUs en una infraestructura HPC.....	9
Figura 3: Representación de las distintas capas del Modelo OSI.....	15
Figura 4: Gráfica de la evolución de tiempos de respuesta mediante un API REST.	16
Figura 5: Gráfica de la evolución de tiempos de respuesta mediante un socket.....	16
Figura 6: Propuesta de la comunicación de un programa híbrido (clásica más cuántica).	17
Figura 7: Pila de llamadas desde un worker a la QPU.....	18
Figura 8: Arquitectura propuesta para la integración de los proveedores cloud.....	20
Figura 9: Creación de una cuenta para un usuario.....	21
Figura 10: Editar cuenta.....	22
Figura 11: Ejecución de trabajo cuántico.	22
Figura 12: Ejemplo de representación de un circuito cuántico.	23
Figura 13: Ejemplo de aplicación de la técnica del quantum circuit cutting.	25
Figura 14: Ejemplo de representación de un circuito cuántico mediante DAGs. Fuente de información: https://doi.org/10.21105/joss.03968	27
Figura 15: Ejemplo de definición de un DAG con Apache Airflow. Fuente de información: https://medium.com/apache-airflow/apache-airflow-2-0-tutorial-41329bbf7211	29
Figura 16: Representación de los bits (izquierda) y cúbits (derecha).....	30
Figura 17: Ejemplo de gráfico de barras obtenido con el método plot_histogram.	31
Figura 18: Ejemplo de representación de la matriz de estado con el método plot_state_city.	32
Figura 19: Ejemplo de representación de la matriz de estado con el método plot_state_hinton.	33
Figura 20: Ejemplo de representación de la matriz de estado con el método plot_state_paulivec.....	33
Figura 21: Ejemplo de representación del estado cuántico con el método plot_state_qsphere.	34

Figura 22: Ejemplo de representación del estado de todos los cúbits con el método plot_bloch_multivector.....	35
Figura 23: Ejemplo de representación de los resultados de un circuito cuántico de 10 cúbits.	36
Figura 24: Ejemplo de visualización del estado de un sistema durante la ejecución del algoritmo de teletransportación cuántica.....	37
Figura 25: Diseño visual de VENUS.....	38
Figura 26: Ejemplo de visualización del árbol binario de esferas de Bloch.....	39
Figura 27: Ejemplo de visualización con GraphStateVis.....	39
Figura 28: Ejemplo de visualización de Virtual Lab.....	40
Figura 29: Propuesta de representación gráfica.	42
Figura 30: Propuesta de visualización para la representación de resultados. Fuente de información: https://arxiv.org/pdf/2302.04919.pdf	44
Figura 31: Diagrama Pert.....	49

Lista de tablas

Tabla 1: Modo de gestión de la contabilidad de los principales proveedores de servicios cuánticos 46

Lista de acrónimos

- DAG *Grafo acíclico dirigido*
- FPGA *Matriz de puerta programable en campo*
- GPU *Unidad de procesamiento gráfico*
- HPC *Computación de alto rendimiento*
- LDAP *Protocolo ligero de acceso a directorios*
- PCI *Interconexión de componentes periféricos*
- QPU *Unidad de proceso cuántica*
- SSH *Shell segura*

1. Introducción

El término de mecánica cuántica apareció a principios del siglo XX para describir aquellos fenómenos que tienen lugar debido a la interacción de partículas y ondas subatómicas. A partir de estos primeros estudios tuvo lugar la primera revolución cuántica que permitió el desarrollo, por ejemplo, del láser o los semiconductores.

La computación cuántica constituye un nuevo paradigma de la computación que se basa en las propiedades de la mecánica cuántica, como la superposición o el entrelazamiento, para realizar cálculos de una manera mucho más eficiente y rápida. A diferencia de la computación clásica que usa los bits para representar la información, la mecánica cuántica usa los bits cuánticos o **cúbits**.

A pesar de los importantes avances alcanzados en los últimos años, se trata de una disciplina todavía emergente (ver Figura 1 para más detalles¹) y son muchos los desafíos que deben superarse para que la tecnología pueda desarrollar todo su potencial.

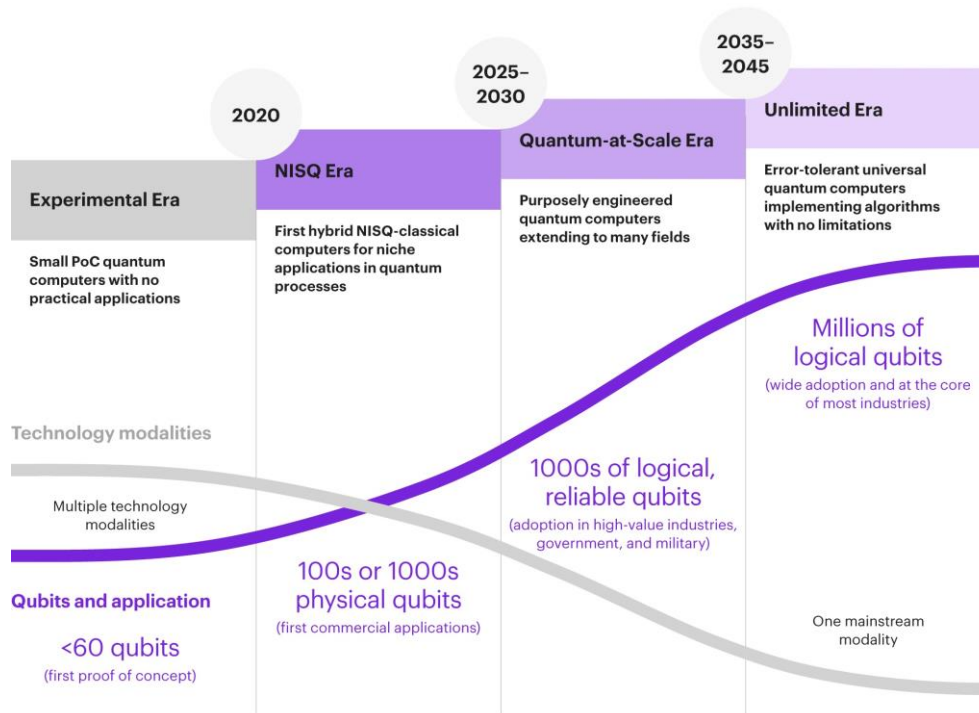


Figura 1: Historia de la computación cuántica.

Uno de los principales desafíos a los que se enfrenta la computación cuántica es la **escalabilidad**. Las computadoras cuánticas actuales manejan un número reducido de

¹ Historia de la computación cuántica. Consultado con fecha de 09/05/2023 en <https://www.kearney.com/industry/technology/article/-/insights/quantum-computing-why-you-should-care>

cúbits y esto supone una restricción importante para la resolución de problemas complejos, donde se podrían requerir de cientos o incluso miles de cúbits. Otro desafío destacable es la inmadurez de la algoritmia. A diferencia de la computación clásica que se basa en la lógica booleana, la computación cuántica se basa en los algoritmos cuánticos. Estos se encuentran aún en una fase muy temprana y suponen un cambio de paradigma, pues difieren de los clásicos tanto en términos de complejidad como de funcionalidad. Además, dado que para su implementación es necesario un importante conocimiento de los conceptos de la física cuántica, sería de utilidad el desarrollo de un lenguaje de programación adaptado y específico que permita abstraer parte de la complejidad subyacente. En computación cuántica los algoritmos se expresan mediante **circuitos cuánticos**. Aunque se trata de una forma extendida de codificar los algoritmos cuánticos, presenta importantes problemas de escalabilidad, ya que su representación se vuelve inmanejable cuando el circuito crece verticalmente (número de cúbits) o horizontalmente (número de puertas cuánticas). En estos casos, el diagrama se vuelve incomprensible y las tareas de depuración se vuelven más tediosas. Por otra parte, la representación de resultados presenta también un desafío importante. La dimensión del espacio de salida crece exponencialmente con el número de cúbits dificultando tanto el procesamiento como la interpretación de los resultados. Así, suponiendo que tenemos un sistema de electrones en cualquiera de 40 posiciones, se tienen un total de 2^{40} configuraciones posibles. Este comportamiento exponencial supone un importante reto, tanto para la representación e interpretación de resultados como para el intercambio de información.

A este conjunto de desafíos técnicos propios de la disciplina habría que sumar otros asociados a la integración de una infraestructura de computación cuántica en un centro de supercomputación como puede ser el Centro de Supercomputación de Galicia (CESGA). Entre ellos, se encuentran el control de acceso, la gestión de colas o la monitorización y gestión de la contabilidad asociada al uso de la infraestructura. Por otra parte, a nivel investigación, puede ser muy interesante poder hacer análisis de benchmarking de las diferentes tecnologías cuánticas para analizar debilidades y fortalezas, o incluso complementar la infraestructura del centro de supercomputación si sus especificaciones suponen una limitación a la hora de desarrollar nuevos algoritmos cuánticos. Para ello, es importante que la infraestructura del centro de supercomputación sea suficientemente flexible para poder integrarse con otros proveedores de servicios de computación cuántica, tales como Amazon Braket, Azure Quantum o IBM Quantum.

En las siguientes secciones se detalla la propuesta de arquitectura para la integración de una computadora cuántica en una infraestructura de supercomputación como puede ser la del CESGA. En la Sección 2 se presenta una visión global de la arquitectura del sistema, especificando los diferentes frontales de acceso a la plataforma y la interacción entre los

distintos recursos, tanto clásicos como cuánticos. Por otra parte, en la Sección 3 se detalla la solución a cada uno de los retos planteados, esto es, el control de acceso, la gestión de colas, las interfaces de comunicación, la integración con recursos externos, la representación de circuitos cuánticos y resultados, y la gestión de la contabilidad. Finalmente, en la Sección 4 se recopilan las principales conclusiones obtenidas durante el estudio.

2. Arquitectura global

A lo largo de esta sección se da una visión general de la propuesta de arquitectura para una plataforma de HPC (High Performance Computing) junto con recursos de Quantum Computing. A su vez, se hace una pequeña comparativa con la arquitectura típica de un supercomputador argumentando los cambios en el diseño de la arquitectura y referenciándolos en posteriores secciones del presente documento.

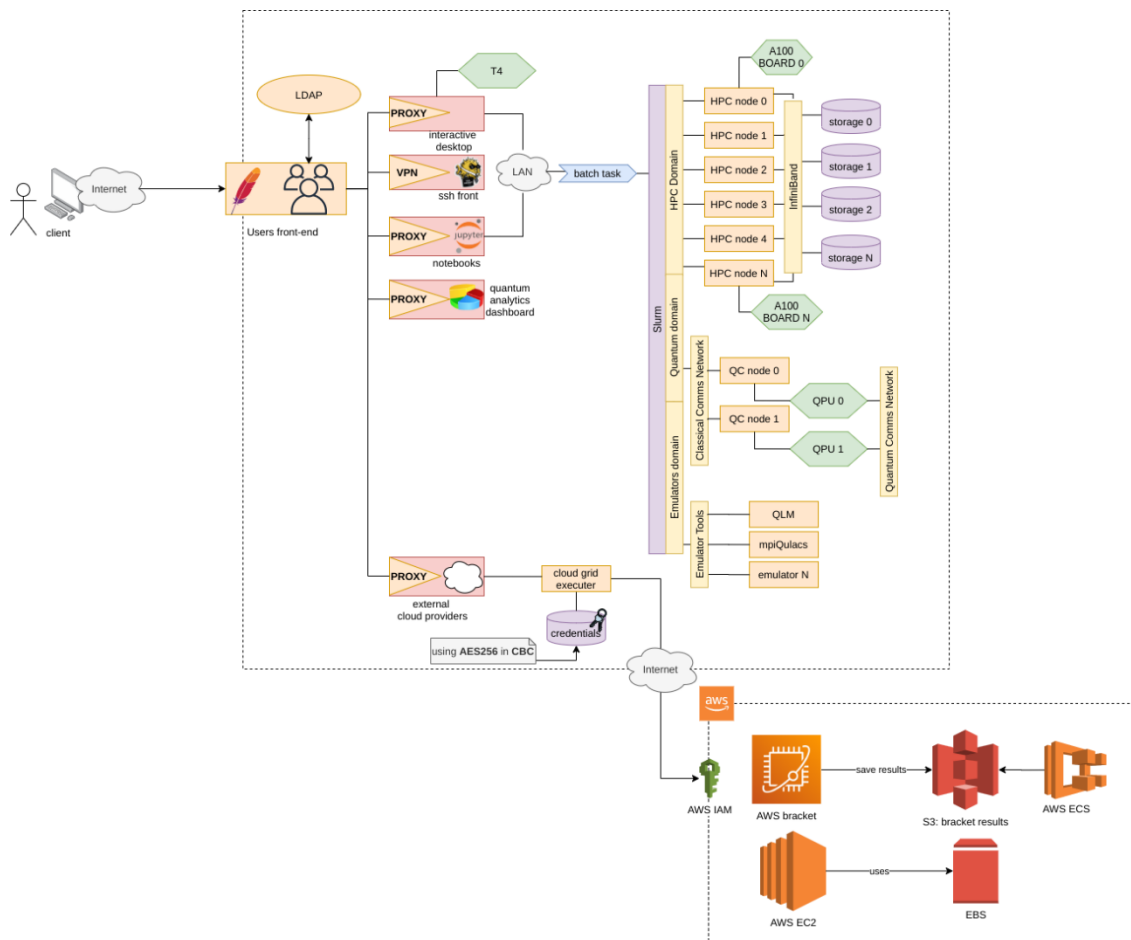


Figura 2: Arquitectura propuesta para la integración de las QPUs en una infraestructura HPC.

En la Figura 2 se puede apreciar la arquitectura propuesta, donde el usuario accede al catálogo de productos disponibles mediante un portal de usuarios, el cual hace uso del LDAP para la autenticación.

Una vez autenticado puede acceder a los servicios que se detallan en el siguiente apartado mediante un proxy, además, de poder acceder a un nodo frontal de SSH mediante una conexión VPN.

Desde los distintos frontales de la plataforma se podrán emitir trabajos en batch al gestor de colas Slurm para cualquiera de sus dominios.

2.1 Frontales

El usuario final de esta plataforma tendrá acceso a cinco frontales distintos, dependiendo del tipo de tareas que quiera ejecutar:

- **Escritorio interactivo:** es un recurso compartido entre los usuarios de la plataforma. Este servicio dispone de tarjetas Nvidia T4 no solo para ejecutar el sistema gráfico sino para poder realizar también ejecuciones con CUDA bajo demanda. La parte de computación es suplida mediante un nodo compartido del sistema de HPC. La forma de acceso a este recurso es mediante un proxy que de acceso al frontal web para controlar el escritorio y, además de la ejecución bajo demanda de pequeñas tareas, se permite el envío de trabajos más complejos al clúster de HPC.
- **Frontal SSH:** máquina de recursos suficientes para realizar tareas de compilación y pruebas que se utiliza para el envío de trabajos al sistema de HPC y la comprobación de los resultados. La forma de acceso a estas máquinas es mediante el uso de un cliente VPN.
- **Jupyter Notebooks:** al igual que el escritorio interactivo, el acceso a este recurso requiere de un proxy. Este sistema de cuadernos permite la creación y edición de scripts (en diversos lenguajes) para su posterior ejecución sobre el sistema HPC donde, además, se podrán analizar los resultados mediante librerías como (matplotlib, pandas, etc.). Como el resto de los frontales web que conforman esta arquitectura se requiere de un proxy de acceso.
- **Quantum Analytics Dashboard:** herramienta especialmente diseñada para el análisis de los resultados, ejecutados sobre las QPU y los distintos emuladores que conforman esta plataforma. Las aproximaciones que se detallan en la Sección 3.6 han sido extraídas y analizadas mediante el estudio del estado del arte de diversas

áreas de conocimiento. Este recurso hace uso no solo del proxy para el acceso, sino que consume directamente del sistema de almacenamiento del entorno de HPC para poder leer, escribir y transformar los datos de los volúmenes de los usuarios de la plataforma.

- **External Cloud Providers:** frontal que permite dar de alta cuentas en sistemas clouds externos, editar recursos y la gestión de tareas de cómputo relacionadas con dichos proveedores.

2.2 Sistema de HPC

El sistema de HPC para la computación clásica seguirá una aproximación similar a la implementada en cualquier superordenador, es decir, un sistema Slurm que gestione los recursos de la corporación y las solicitudes de los usuarios. Tras este sistema se encontrarían una serie de computadores, aceleradores hardware (GPUs, co-procesadores, FPGAs, etc.) y un sistema de almacenamiento conectados a una red de baja latencia.

La principal diferencia de la arquitectura propuesta en este documento con una implementación de un superordenador convencional es que el sistema de almacenamiento y el clúster HPC no estarían divididos dependiendo de si la tarea es clásica o cuántica. Además de lo citado previamente, no se puede acceder mediante una API directamente a las QPUs, sino que todas las peticiones tienen que ir mediante el Slurm. En el caso de las ejecuciones híbridas se hará uso del driver para las comunicaciones mediante un bus en lugar de una API, tal y como se detalla en la Sección 3.3.

2.3 Recursos cuánticos

En comparativa con una presente distribución de un supercomputador, el planteamiento que se hace en este documento es partiendo de que los emuladores cuánticos se encuentran todos al mismo nivel y son gestionados mediante el uso y las reglas de Slurm. No solo se tendrán en cuenta varios emuladores, sino que también cabe la posibilidad de tener que gestionar varios computadores cuánticos a la vez, siendo estos de distintos tipos (p. ej. uno basado en superconductores y otro en trampa de iones). Para ello se tendrá no solo que añadir las QPUs al sistema de colas, sino que se tendrán que crear reglas específicas para cuando se quiera realizar computación híbrida (clásica más cuántica). Por último, se deben tener en cuenta dos factores importantes para el escalado de la plataforma:

- **Segmentación de la QPU:** es posible que algunas de las tareas que entren en el planificador de trabajos no requieran toda la capacidad de cómputo, de manera que al menos dos tareas puedan ejecutarse a la vez en una misma QPU.
- **Comunicaciones entre las QPUs:** de forma opuesta a lo que se ha explicado en el punto anterior, es posible que una tarea requiera más potencia de la que ofrece una sola QPU y haya que combinar varios de estos recursos (al igual que se hace con las CPUs con MPI o las GPUs de Nvidia con nvidia). Para esta solución se plantean dos vías: (1) mediante un sistema de comunicaciones clásicas o (2) mediante un sistema de comunicaciones cuánticas. La opción (1) es poco recomendable por la inyección de ruido a los resultados. Por otra parte, la opción (2) constituye la aproximación deseable ya que garantiza el entrelazamiento cuántico entre las dos o más QPUs y asegura la calidad de los resultados.

2.4 Acceso a recursos externos

El escalado, el acceso a los recursos o las soluciones propietarias, pueden ser la motivación que pueden llevar a los usuarios de un centro de supercomputación a utilizar proveedores cloud, como AWS, Google o Azure en lugar de los recursos que se ponen a disposición en dicha corporación. Por ello, se debe contemplar la integración de las diversas soluciones previamente explicadas con estos sistemas de terceros. A lo largo de la Sección 3.4 del presente documento se detallan y analizan que productos hay en el mercado para hacer frente a este requerimiento y que solución se ha optado.

En líneas generales, este servicio es el encargado de gestionar las cuentas y recursos de todos los usuarios de la plataforma en los distintos proveedores de servicios cloud. Para ello, estas claves deberán estar almacenadas de forma segura en una base de datos mediante el uso de cifrado simétrico.

Este servicio contará con un sistema de colas propio para la gestión de trabajos, teniendo en cuenta los límites de cada uno de los usuarios y mediante las distintas APIs de los proveedores de servicios se emitirán los trabajos haciendo uso de los servicios de autenticación y autorización como, por ejemplo, IAM de AWS.

3. Retos

3.1 Control de acceso

La propuesta actual toma de base un escenario de supercomputación genérico como el del BSC o el CESGA. De igual manera, solo hay dos interfaces de entrada:

- Un nodo de login SSH al cual se accede mediante una conexión VPN.
- Un portal de usuario abierto para autenticar a los usuarios del sistema.

Desde dicho portal de usuarios se dará acceso, mediante un proxy, a la siguiente gama de servicios:

- Escritorios interactivos.
- Notebooks de Jupyter.
- Sistema de análisis de resultados de circuitos cuánticos.

Tanto con los escritorios como los notebooks se pueden realizar ejecuciones de computación clásica y emitir trabajos en batches, ya sean clásicos, cuánticos o híbridos. El sistema de analítica de resultados utilizaría los resultados que el usuario haya generado en las ejecuciones para, mediante una serie de dashboards predefinidos, poder sacar valor y mayor interpretabilidad a los resultados. Finalmente, el conglomerado de sistemas de emulación permite el despliegue de distintos servicios adicionales como terminales SSH o notebooks de Jupyter para exploraciones y experimentaciones.

3.2 Gestión de colas

A lo largo de este punto se estudia y analiza la situación actual de un sistema de colas en un clúster HPC y se detallan y proponen una serie de mejoras para la integración de QPUs en dicho sistema.

3.2.1 Problemática actual

Al instalar una QPU en un centro de supercomputación, toda su infraestructura de cálculo estará gestionada y orquestada por 2 instancias de Slurm distintas. Estas dos instancias de Slurm serán las responsables de gestionar las distintas áreas del entorno clúster:

- Los nodos de cómputo y aceleración.
- El sistema de HPC asociado al computador cuántico.

Por otra parte, en el caso de disponer un emulador cuántico con frontal propio, se necesitará también un sistema de cargas de trabajo interna. Esta solución, deja varias incongruencias en la composición de los elementos del clúster:

- Sistemas de almacenamiento separados aumentando la dificultad de la comunicación entre ambos.
- Sistemas de HPC divididos en dos, limitando las capacidades de cómputo generales del sistema HPC, y reduciendo la capacidad de esos recursos para hacer computación híbrida.
- No hay una integración en el gestor de tareas para la computación híbrida (ya que unas son gestionadas por el frontal cuántico y otras por el Slurm).
- Sobre gestión y mantenimiento de los sistemas de tareas.

3.2.2 Propuesta y mejoras

Dadas estas limitaciones, en la presente propuesta se propone el uso de una única instancia de Slurm que gestione tanto los nodos de cómputo y aceleradoras, como la QPU. La única diferencia será el uso de los dominios Slurm.

Los dominios en Slurm se refieren a las diferentes áreas en las que se puede dividir un clúster de computación, como por ejemplo grupos de nodos, departamentos o proyectos. Estos dominios permiten asignar recursos, establecer políticas de acceso y controlar el uso de los recursos dentro del clúster, de acuerdo con las necesidades específicas de cada grupo o proyecto. Por ejemplo, un departamento puede tener su propio dominio con un conjunto de nodos dedicados exclusivamente a sus trabajos, mientras que otro departamento puede tener su propio conjunto de nodos separados y dedicados.

Además, permiten establecer cuotas de uso para los recursos, lo que ayuda a evitar que un grupo o proyecto acapare los recursos del clúster y afecte al rendimiento de otros usuarios. Además, los dominios pueden ser gestionados de forma independiente, lo que facilita la administración y el mantenimiento del clúster en general.

Por lo tanto, se tendrían que declarar dos dominios, uno para todo lo referente a la computación clásica (CPUs y GPUs), y otro para todo lo referente a las QPUs. Todos estos dominios tendrán completo acceso al sistema de almacenamiento, pues es necesario en ambos casos. De la misma forma, se tendría que declarar un dominio exclusivo para los distintos emuladores que conforman la infraestructura.

Además de los dominios previamente citados, sería interesante tener un dominio mixto en donde se integren los elementos de computación clásica y cuántica, ya que, probablemente

muchos de los problemas que se tengan que resolver a futuro requieran ejecutar un algoritmo clásico mediante condiciones cuánticas o viceversa.

3.3 Interfaces de comunicación

3.3.1 Comunicaciones mediante las interfaces de red

Muchas de las soluciones actuales de los computadores cuánticos pasan por comunicaciones mediante la capa 7 (aplicación) del modelo OSI (ver Figura 3 para más detalles²). La gran mayoría de ellas se basan en REST y sobre la capa de transporte HTTP. Para ejecuciones de pequeña complejidad (aprox. 10-12 cúbits) este método es viable, no obstante, a partir de ese umbral los tiempos y el tamaño de las respuestas se vuelven inmanejables.

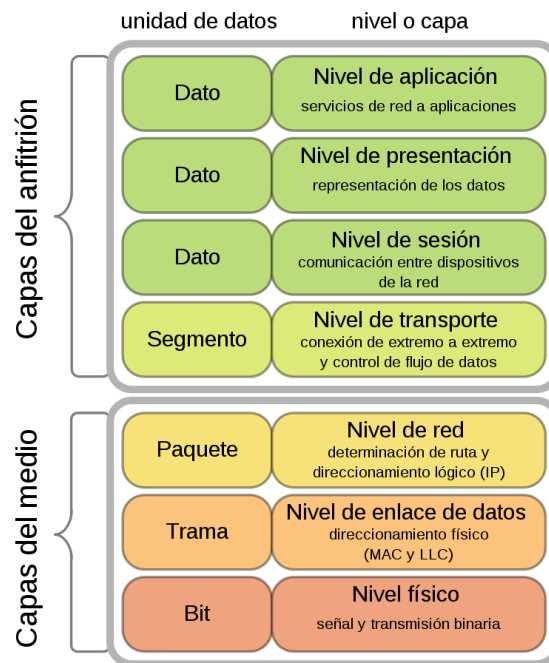


Figura 3: Representación de las distintas capas del Modelo OSI.

Para contrastar la afirmación anterior se ha implementado un pequeño benchmark para comparar los tiempos y los tamaños de las respuestas aplicando las tecnologías actuales (APIs REST) y una solución propuesta la cual funciona mediante sockets (capa 5 del modelo OSI: sesión). Para ello se realizaron un total de 16 llamadas entre el cliente y el servidor, en

² Modelo OSI. Consultado con fecha de 11/05/2023 en <https://programacionfacil.org/blog/el-modelo-osi-explicacion-sencilla/CTS-2022-0054>

las cuales se añadía 1 cúbit más a la transferencia de información y se midieron los tiempos y los tamaños de las respuestas.

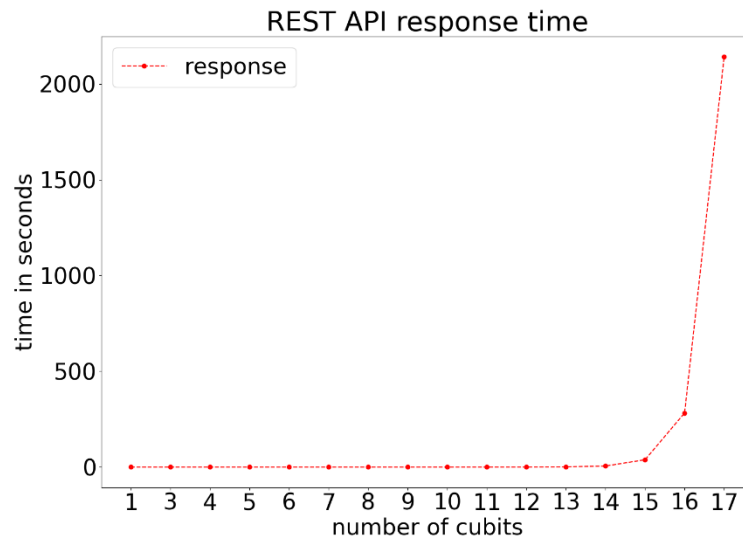


Figura 4: Gráfica de la evolución de tiempos de respuesta mediante un API REST.

En la Figura 4 se puede apreciar el comportamiento exponencial de esta nueva tecnología, observándose una progresión de los tiempos que oscila desde unas décimas de segundo (0.827 segundos para 13 cúbits) hasta más de 35 minutos en el caso de 17 cúbits. El tiempo de respuesta para 15 cúbits es de 38 segundos mientras que en el caso de 16 cúbits ronda los 5 minutos.

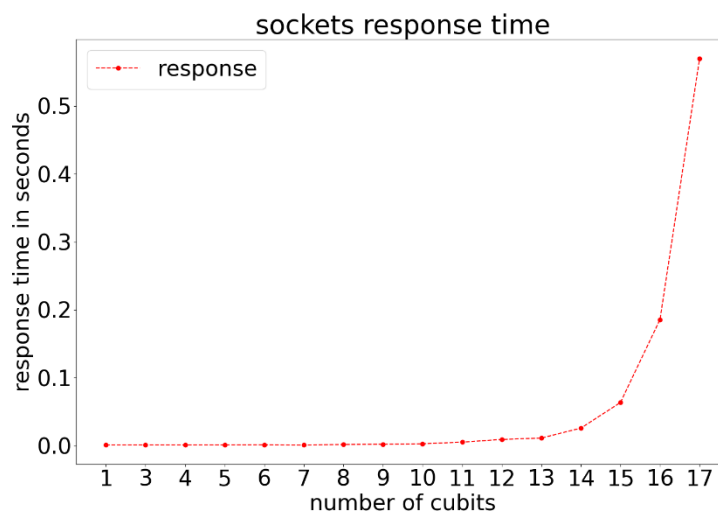


Figura 5: Gráfica de la evolución de tiempos de respuesta mediante un socket.

De igual manera, optimizando las comunicaciones con los sockets se tiene un comportamiento exponencial, pero no tan pronunciado. La principal diferencia es la disminución de los tiempos, pasando de aproximadamente 35 minutos a unos escasos 0.5 segundos (ver Figura 5 para más detalles). No obstante, es una mejora poco significativa, ya que, como se ha citado anteriormente, tiene un carácter exponencial y al realizar las mismas operaciones con 22 cúbits (4194304 bits), las comunicaciones tardan aproximadamente 15 minutos.

Analizando el tamaño de la respuesta, nos encontramos una situación muy parecida a la descrita en los tiempos de comunicación. Crecimientos exponenciales en ambos casos, siendo el caso de los sockets más ligero que en el del API REST.

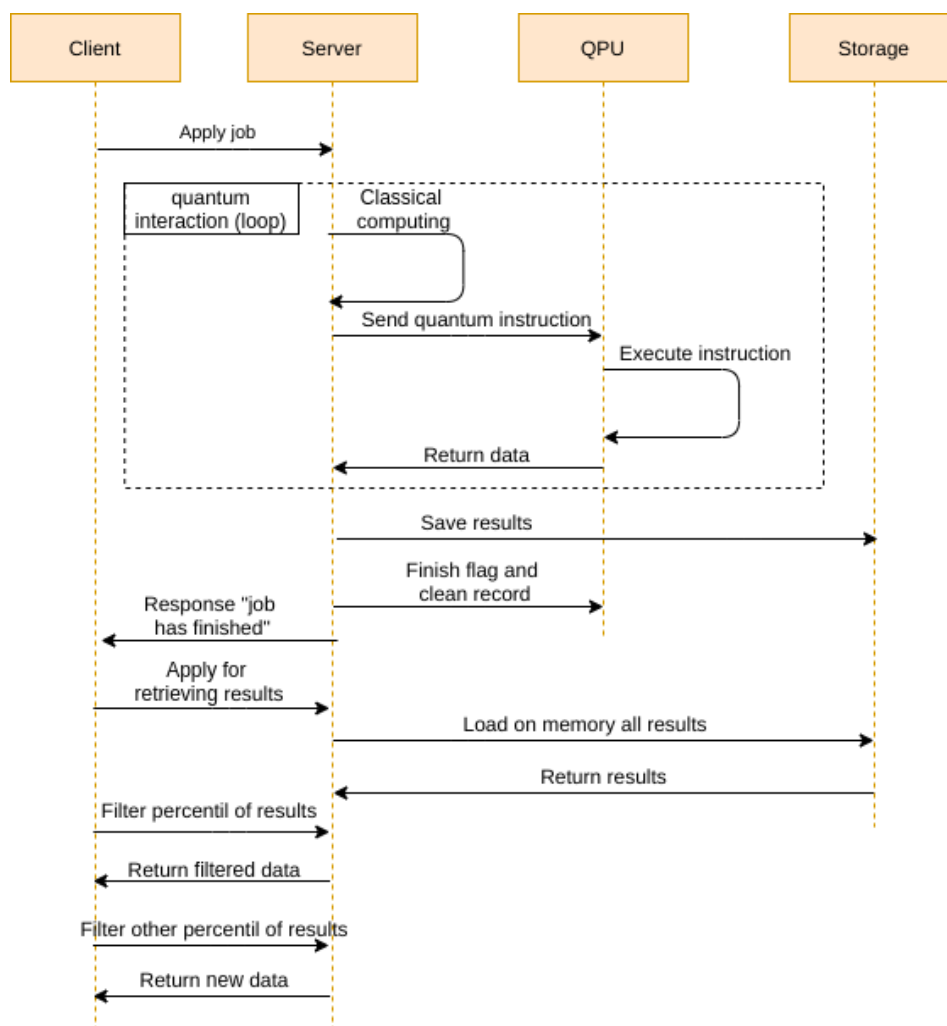


Figura 6: Propuesta de la comunicación de un programa híbrido (clásica más cuántica).

Para concluir, el uso de sockets, es decir, utilizar métodos de comunicación menos generalizables y más versátiles, permite el intercambio de información de una forma mucho más eficiente. No obstante, la naturaleza de esta tecnología no hace viable la posibilidad de recoger toda la información en un cliente y gestionarla desde ahí. Para poder realizar esta

clase de intercambios de información en fragmentos/regiones de interés el uso de sockets se adapta correctamente, puesto que se puede construir un protocolo de comunicaciones eficiente para la carga de datos en batches tal y como muestra la propuesta de la Figura 6.

3.3.2 Comunicaciones mediante un bus

La computación híbrida es un requerimiento de muchos centros de cálculo. El principal ejemplo es el uso de la CPU con la GPU mediante el uso de CUDA y otras librerías, aunque existen otros paradigmas como coprocesadores, FPGAs, etc.

Una de las líneas de trabajo a futuro, especula que el concepto de QPU u ordenador cuántico acabará siendo relegado a ser un componente más de aceleración de cálculo como los ya citados. Dado ese escenario, es necesario analizar e implementar un conjunto de controladores y directivas que permitan realizar llamadas al sistema para ejecutar unas determinadas instrucciones sobre la QPU y, a la vez, poder gestionar los resultados y las interrupciones que se puedan producir.

En el caso del kernel Linux, el bus de comunicaciones más habitual para realizar este tipo de operaciones es mediante PCI (ver Figura 7 para más detalles).

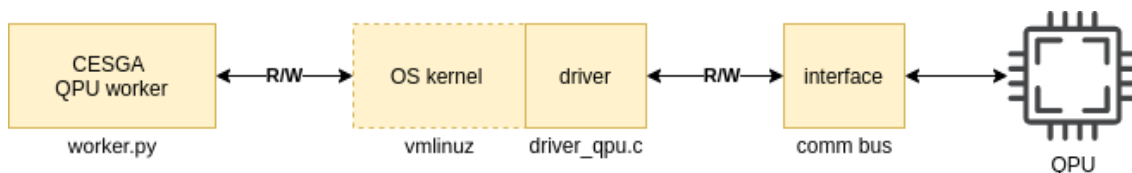


Figura 7: Pila de llamadas desde un worker a la QPU.

La implementación de un driver PCI Express para el kernel de Linux implica varios pasos, pudiendo variar algunos de ellos en base a la arquitectura y/o tipo del periférico a gestionar. No obstante, en líneas generales, estos serían los pasos a seguir:

1. **Identificar las funciones, las estructuras y la arquitectura** del dispositivo PCI Express que se desea controlar.
2. **Descargar el vmlinuz** de la versión del kernel con la que se quiera probar el controlador.
3. **Crear el archivo C o C++ que vaya a implementar la interfaz PCI** y definir los registros de solo lectura del dispositivo.
 - a. VendorId: registro de 16 bits que identifica a un fabricante de hardware.

- b. DeviceId: registro interno de 16 bits que utiliza el fabricante para identificar el dispositivo.
 - c. Class: registro de 8 o 16 bits que identifica el tipo de dispositivo conectado (p. ej. ethernet).
4. **Implementación de las structs y declarar las funciones necesarias** para leer y escribir la información entre el kernel y el periférico. Estas funciones deben de ser:
 - a. Inicialización del dispositivo.
 - b. Lectura y escritura de los registros y de la memoria.
 - c. Gestión de interrupciones.
 - d. Comunicación con el resto de los dispositivos conectado al bus PCI.
5. **Compilación.**
6. **Testing.**
7. **Registro y empaquetado del controlador.** El paquete de instalación debe incluir el driver compilado, junto con cualquier otra dependencia o documentación necesaria para su instalación y uso.
8. **Instalar el controlador.** Una vez se han implementado todas las funciones del controlador, este debe vincularse con el kernel de Linux. Luego, el controlador puede cargarse en el kernel de Linux utilizando las herramientas estándar del kernel.

3.4 Integración con recursos externos

Como ya se citó en la Sección 2.4, cabe la posibilidad que haya usuarios que requieran más capacidad de cómputo que la configuración actual de cualquier supercomputador. Por lo tanto, es necesario delegar esa parte del cómputo en proveedores externos, como bien puedan ser AWS, Azure, etc. Para la integración de la plataforma de supercómputo con dichos proveedores se estudiaron varias herramientas, entre las cuales destaca el software GridWay MetaScheduler³.

GridWay MetaScheduler es un software de programación y gestión de recursos que se utiliza para ejecutar aplicaciones de forma eficiente en entornos de computación distribuida. Es un sistema de software libre y de código abierto que está diseñado para integrarse con diferentes tipos de sistemas y recursos de computación, incluyendo

³ GridWay Metascheduler. Consultado con fecha de 11/05/2023 en <https://gridway.ucm.es/doku.php>

clústeres, grids y algunas nubes. Además, permite a los usuarios definir y ejecutar trabajos que involucren múltiples nodos de computación distribuida de manera transparente, lo que significa que los usuarios no necesitan conocer los detalles de la infraestructura subyacente para utilizar el sistema. El software proporciona una serie de características para ayudar a los usuarios a gestionar sus trabajos, incluyendo la planificación de tareas, la monitorización del rendimiento y la gestión de recursos.

No obstante, dos puntos negativos llegaron a desechar el uso de este producto para el presente requisito de la propuesta de la plataforma:

- **Falta de mantenimiento:** dicho software lleva desde marzo de 2015 sin recibir actualizaciones, por lo tanto, la implantación requeriría un doble esfuerzo no solo de mantenerlo sino de saber cómo funciona internamente.
- **No está adaptado a los productos de computación cuántico** de los proveedores cloud. Al revisar su documentación y parte de su código fuente no se hace mención a productos como, por ejemplo, AWS Braket, siendo este el principal interés en el requerimiento presente.

Dadas estas dificultades, se ha optado por diseñar un sistema de integración de trabajos (cuánticos o clásicos) desde la red interna del superordenador con el resto de las plataformas privadas.

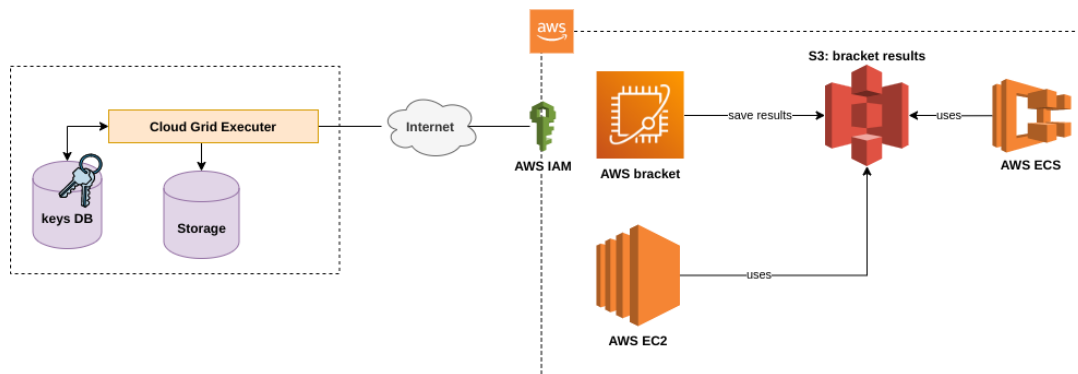


Figura 8: Arquitectura propuesta para la integración de los proveedores cloud.

El diagrama de la Figura 8 ilustra la integración con el proveedor cloud de Amazon. Dentro de la infraestructura destacan 3 componentes:

- **Cloud Grid Executer:** encargado de gestionar las tareas, no solo de cómputo sino de alta de usuarios y gestión de los créditos.
- **Keys DataBase:** base de datos donde se almacenan las claves programáticas para poder emitir peticiones a los distintos servicios cloud. Esta base de datos estaría protegida mediante cifrado simétrico AES256.

- **Sistema de almacenamiento:** arroja capacidad para guardar los resultados finales.

A continuación, se exponen una serie de casos de uso que se deberán de tener en cuenta en el escenario de una integración con, por ejemplo, Amazon Web Services.

La gestión de las cuentas de los usuarios será mediante la creación de subcuentas, donde, desde una cuenta principal del centro de supercomputación, se darán de alta y vincularán una serie de subcuentas para los usuarios, estableciendo no solo la cuenta del usuario, y sus claves, sino los productos a los que el usuario tendrá acceso y los créditos disponibles para hacer uso de dichos productos.

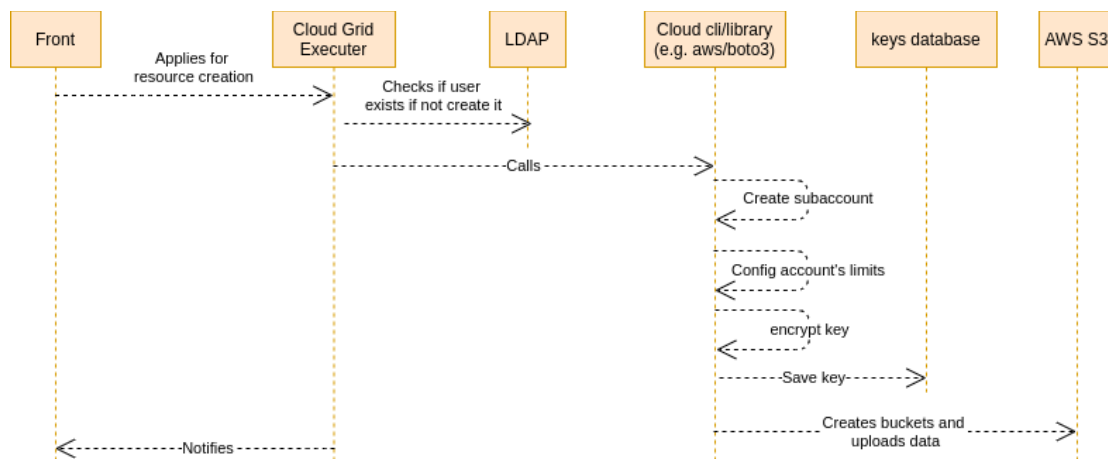


Figura 9: Creación de una cuenta para un usuario.

Tal y como muestra el diagrama de la Figura 9, un responsable daría de alta en el servicio a un nuevo usuario. Para ello, se tiene que comprobar si el usuario existe en el directorio activo del centro, si no, se crea. Una vez dado de alta en el directorio activo, se procede a crear la subcuenta y establecer los límites del usuario, es decir, a qué productos tiene acceso y cuantos créditos ha contratado o se le han proporcionado. Por último, se genera la clave programática para el usuario y se almacena en la base de datos de forma segura mediante un cifrado simétrico (AES256). Por último, se notifica al usuario de la creación de su cuenta.

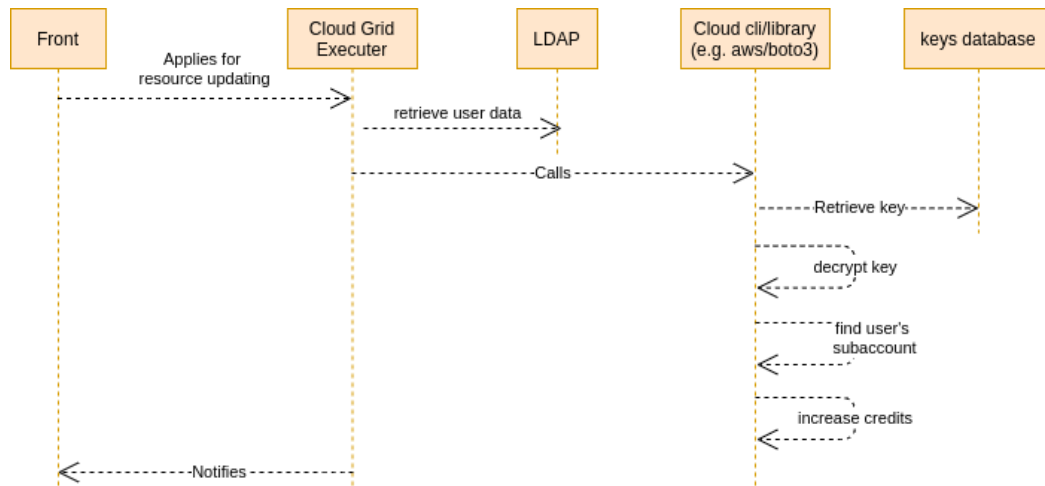


Figura 10: Editar cuenta.

En el diagrama de la Figura 10 se muestra cómo se podría aumentar el crédito de un determinado usuario existente en la plataforma. Para ello, el administrador de la plataforma tendría que recuperar la clave programática de administración de dicha cuenta y aumentar el número de créditos que se consideren. Por último, se notifica al usuario.

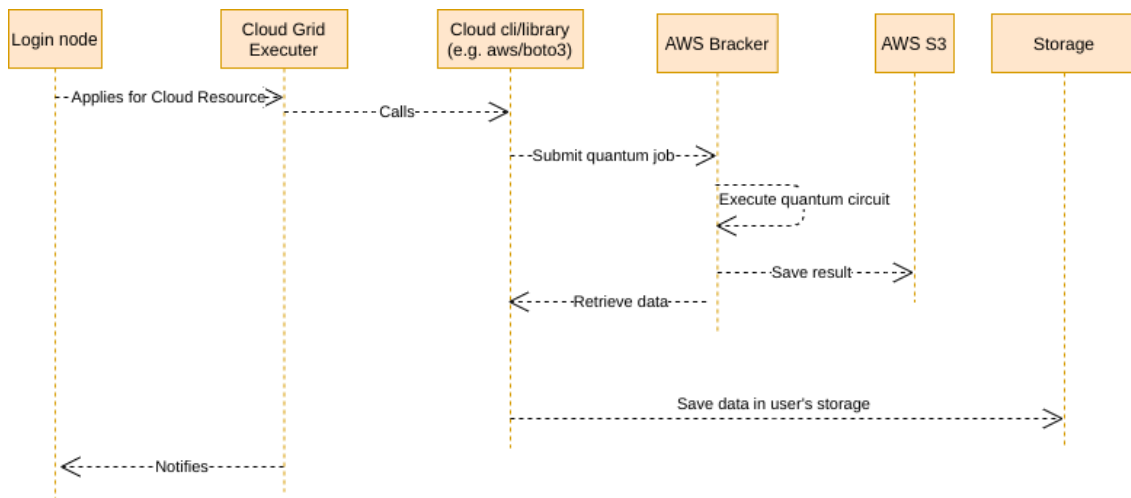


Figura 11: Ejecución de trabajo cuántico.

Finalmente, para la ejecución de un circuito cuántico, se tiene que aplicar mediante un frontal y el proceso es muy parecido a los pasos anteriores, se recupera la clave programática y se ejecuta el trabajo mediante el API de boto3 (en el caso de AWS), se ejecuta el circuito pertinente, se guardan los datos en un bucket de S3 para finalmente recuperarlos desde la red local del centro de supercomputación y almacenarlos en los directorios del usuario (ver Figura 11 para más detalles).

3.5 Representación de los circuitos cuánticos

Un circuito cuántico es un modelo de representación abstracta de las operaciones cuánticas. Cada línea horizontal representa un cúbit. En términos generales un circuito cuántico cuenta con tres componentes claramente diferenciados (ver Figura 12 para más detalles⁴):

- **Entradas:** constituyen el punto de entrada del circuito cuántico y usualmente se representan en la parte izquierda del circuito.
- **Puertas cuánticas:** conjunto de operaciones que se realizan sobre los cúbits y se representan mediante cajas.
- **Salidas:** representan los resultados finales del circuito cuántico y se obtienen después de realizar las medidas.

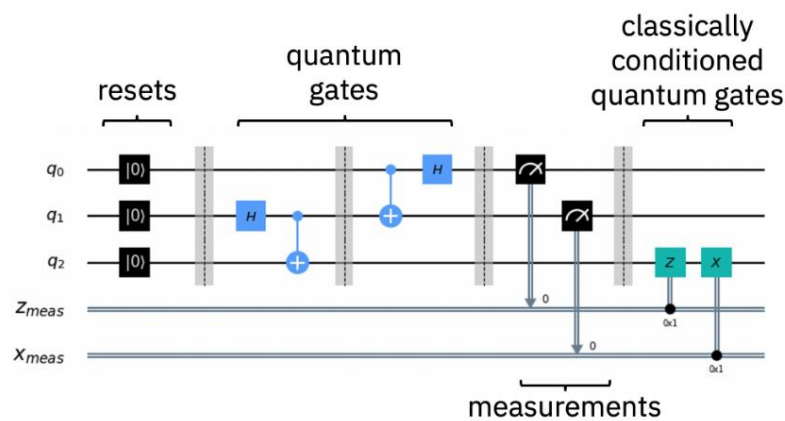


Figura 12: Ejemplo de representación de un circuito cuántico.

Las puertas cuánticas constituyen las operaciones primitivas y representan transformaciones reversibles que permiten preservar la información almacenada en los cúbits. Parte de las puertas cuánticas tienen su análoga clásica (p. ej., las puertas X o CX), mientras que otras son exclusivas del ámbito cuántico (p. ej., la puerta de Hadamard). El operador de medición, representado con el símbolo del medidor permite extraer información parcial sobre el estado del cúbit y representarlo mediante un bit clásico. Con el conjunto de puertas universales

$$H, rZ(\theta), CX$$

y las puertas de medición, se puede construir cualquier circuito cuántico.

⁴ Circuito cuántico. Consultado con fecha de 28/05/2023. Fuente de información: https://qiskit.org/documentation/locale/es_UN/qc_intro.html

Los algoritmos cuánticos pueden combinar parte cuántica y clásica. Por ejemplo, los algoritmos variacionales ejecutan circuitos cuánticos dentro de un bucle de optimización que contiene procesamiento clásico.

A pesar de ser una forma elegante e intuitiva de representar los procesos, esta representación de los circuitos cuánticos presenta problemas de escalado cuando aumenta la complejidad, tanto horizontal (aumentando el número de operadores) como verticalmente (aumentando el número de cúbits). En el primer caso, si el número de operadores es elevado, la representación del circuito se vuelve demasiado grande y puede llegar a rebasar el ancho de la pantalla, teniendo que usar la barra de desplazamiento para poder ver todos los operadores involucrados. Además, en este contexto garantizar la trazabilidad de los inputs y outputs puede llegar a ser una tarea ardua. Por otra parte, cuando el número de cúbits aumenta, usualmente también crece la complejidad del circuito, dificultando su interpretabilidad.

Para subsanar estas deficiencias, se han considerado diferentes aproximaciones. Entre ellas, destacan:

- **Quantum circuit cutting:** técnica que permite descomponer un circuito cuántico en componentes más elementales (subcircuitos).
- **Herramientas de abstracción:** permite agrupar varias puertas cuánticas en una única capa abstracta.
- **Diagramas de flujo:** constituyen un método alternativo para representar circuitos cuánticos grandes.

El quantum circuit cutting permite cortar un circuito cuántico grande en subcircuitos más pequeños (ver Figura 13 para más detalles⁵). Esta técnica, posee una doble funcionalidad. Por una parte, permite reducir el tamaño de la representación del circuito cuántico mediante la descomposición en componentes más elementales y, por otra parte, permite ejecutar un circuito cuántico complejo en múltiples dispositivos más pequeños. Esta segunda funcionalidad es de suma importancia en estos inicios de la era NISQ (Noisy Intermediate-Scale Quantum), en donde no es sencillo todavía disponer una infraestructura cuántica con un número elevado de cúbits garantizando una buena corrección de los errores. Así, el quantum circuit cutting puede suponer una alternativa idónea para poder ejecutar circuitos complejos que actualmente son inabordables desde las infraestructuras cuánticas estándares actuales.

⁵ Quantum circuit cutting. Consultado con fecha de 28/05/2023. Fuente de información: <https://dl.acm.org/doi/10.1145/3445814.3446758>

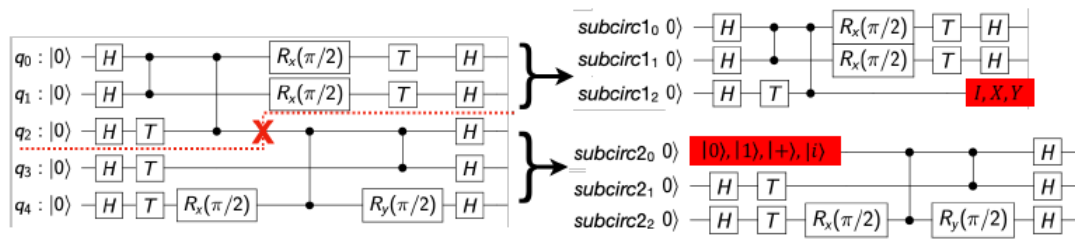


Figura 13: Ejemplo de aplicación de la técnica del quantum circuit cutting.

Actualmente, son muchos los investigadores que están centrando su trabajo en analizar la forma óptima por donde cortar un circuito cuántico para minimizar la pérdida de información y asegurar que la suma de información almacenada en todos los circuitos es próxima a la del circuito original.

En Tang et al. (2021)⁶ se propone CuQC, un enfoque computacional híbrido y altamente escalable que combina la computación cuántica y clásica para ejecutar algoritmos que, con las capacidades de cómputo disponibles hoy en día, no podrían ser resolubles mediante un único enfoque. El CuQC busca cortar los circuitos cuánticos en subcircuitos más pequeños de manera que puedan ser ejecutados en computadoras cuánticas con capacidades más limitadas. Posteriormente, mediante un procedimiento clásico se podría reconstruir la salida del circuito cuántico general.

En Xin-Chuan et al. (2021)⁷ y Lowe et al. (2023)⁸ se proponen algoritmos de optimización para determinar el lugar de corte del circuito cuántico. En el primer caso se propone un framework de optimización jerárquico bloque por bloque (QGo) que combina tres procesos:

1. **Particionamiento:** dividir un circuito cuántico en un conjunto de bloques independientes.
2. **Síntesis:** regenerar y optimizar cada bloque usando síntesis cuántica.

⁶ Tang, W., Tomesh, T., Suchara, M., Larson, J., & Martonosi, M. (2021, April). Cutqc: using small quantum computers for large quantum circuit evaluations. In Proceedings of the 26th ACM International conference on architectural support for programming languages and operating systems (pp. 473-486).

⁷ X. -C. Wu, M. G. Davis, F. T. Chong and C. Iancu, "Reoptimization of Quantum Circuits via Hierarchical Synthesis," *2021 International Conference on Rebooting Computing (ICRC)*, Los Alamitos, CA, USA, 2021, pp. 35-46, doi: 10.1109/ICRC53822.2021.00016.

⁸ Lowe, A., Medvidović, M., Hayes, A., O'Riordan, L. J., Bromley, T. R., Arrazola, J. M., & Killoran, N. (2023). Fast quantum circuit cutting with randomized measurements. *Quantum*, 7, 934.

3. **Reconstrucción:** recomponer el circuito final mediante la unión de todos los bloques.

En este caso, se testea el comportamiento del algoritmo propuesto en varios escenarios: circuitos pequeños en dispositivos reales, circuitos de tamaño medio en simulaciones con ruido y circuitos grandes de más de 60 cúbits. Por otra parte, en el segundo caso se propone un enfoque basado en teoría de grafos para la elección de los cortes y se valida el comportamiento de la propuesta aplicándolo sobre el algoritmo *Quantum Approximate Optimization Algorithm* (QAOA).

El costo de postprocesamiento clásico necesario para la reconstrucción original crece de forma exponencial con el número de cortes. Es por ello que muchos autores basan el procedimiento de corte en esta medida. Así, por ejemplo, en Harada et al. (2023)⁹ se propone un corte paralelo óptimo sin cúbits ancilla y en Chen et al. (2022)¹⁰ usan un método basado en muestreo como Markov Chain Monte Carlo (MCMC) para seleccionar cadenas de bits de alta probabilidad y evitar cálculos excesivos en la reconstrucción de la distribución de probabilidad global. Otros métodos determinan el punto de corte en base a algoritmos de clustering (Peng et al., 2020)¹¹ o redes neuronales (Sarkar and Adhikari, 2023)¹².

En cuanto a las herramientas de abstracción, en Sergioli (2021)¹³ se propone una estrategia general para obtener una representación en matriz de bloques de puertas cuánticas aplicadas a cúbits ubicados en posiciones arbitrarias sobre un estado de entrada dimensional también arbitrario.

La tercera opción (diagramas de flujo), además de constituir una forma alternativa de representar los circuitos cuánticos (diagramas de grafos), también permiten aplicar técnicas propias del ámbito para optimizar los procesos involucrados. Así por ejemplo, en Moflic et al. (2023)¹⁴ usan DAGs (Directed Acyclic Graphs) para representar los circuitos cuánticos (ver en Figura 14 un ejemplo de circuito cuántico y su correspondiente DAG) y aplican aprendizaje por refuerzo (RL por sus siglas en inglés) para la optimización de los circuitos cuánticos. Además, dado que el espacio que debe explorar un agente RL es

⁹ Harada, H., Wada, K., & Yamamoto, N. (2023). Optimal parallel wire cutting without ancilla qubits. arXiv preprint arXiv:2303.07340.

¹⁰ Chen, D., Baheri, B., Chaudhary, V., Guan, Q., Xie, N., & Xu, S. (2022). Approximate Quantum Circuit Cutting. arXiv preprint arXiv:2212.01270.

¹¹ Peng, T., Harrow, A. W., Ozols, M., & Wu, X. (2020). Simulating large quantum circuits on a small quantum computer. Physical review letters, 125(15), 150504.

¹² Sarkar, R. S., & Adhikari, B. (2023). Scalable quantum circuits for n-qubit unitary matrices. arXiv preprint arXiv:2304.14096.

¹³ Sergioli, G. A Matrix Representation of Quantum Circuits over Non-Adjacent Qudits. Int J Theor Phys 60, 515–533 (2021). <https://doi.org/10.1007/s10773-019-04051-5>

¹⁴ Moflic, I., Garg, V., & Paler, A. (2023). Graph Neural Network Autoencoders for Efficient Quantum Circuit Optimisation. arXiv preprint arXiv:2303.03280.

extremadamente grande y eso ralentiza el aprendizaje de las estrategias de optimización, los autores presentan una solución basada en autoencoders para la optimización de los circuitos cuánticos y validan la implementación en circuitos de Bernstein-Vazirani, concluyendo que se mantiene la optimización del método RL original reduciendo en un 20% el tamaño de la tabla que codifica la estrategia de optimización aprendida.

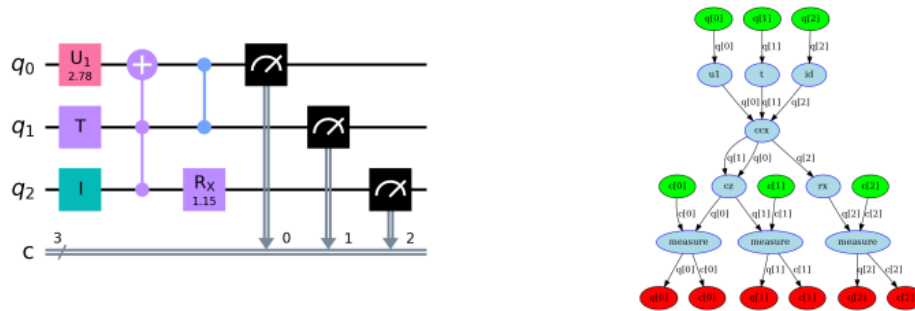


Figura 14: Ejemplo de representación de un circuito cuántico mediante DAGs. Fuente de información: <https://doi.org/10.21105/joss.03968>

Hong et al. (2022)¹⁵ proponen una estructura de datos similar a los diagramas de decisión, denominada Diagrama de Decisión de Tensor (TDD por sus siglas en inglés) que permite una representación compacta y canónica de los circuitos cuánticos. Además, se espera que los TDD desempeñen un papel importante en diversas tareas relacionadas con la automatización del diseño, tales como el chequeo de la equivalencia, la detección de errores, la síntesis, la simulación y la verificación.

La optimización de los circuitos cuánticos permite reducir su tiempo de cómputo. Un aspecto clave en muchas técnicas de optimización de circuitos cuánticos es la coincidencia de patrones. En Iten et al. (2022)¹⁶ y Chen et al. (2022)¹⁷ se proponen algoritmos de *pattern matching* que se basan en una representación del circuito cuántico en DAGs. En el primer caso, el algoritmo se ha implementado en Qiskit y se ha demostrado empíricamente que escala de manera satisfactoria. Por otra parte, en rustworkx (Treinish et al., 2022)¹⁸ se

¹⁵ Hong, X., Zhou, X., Li, S., Feng, Y., & Ying, M. (2022). A tensor network-based decision diagram for representation of quantum circuits. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 27(6), 1-30.

¹⁶ Iten, R., Moyard, R., Metger, T., Sutter, D., & Woerner, S. (2022). Exact and practical pattern matching for quantum circuit optimization. *ACM Transactions on Quantum Computing*, 3(1), 1-41.

¹⁷ Mingyu Chen, Yu Zhang, Yongshang Li, Zhen Wang, Jun Li, and Xiangyang Li. (2022). QCIR: Pattern Matching Based Universal Quantum Circuit Rewriting Framework. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22)*. Association for Computing Machinery, New York, NY, USA, Article 55, 1-8. <https://doi.org/10.1145/3508352.3549405>

¹⁸ Treinish et al., (2022). rustworkx: A High-Performance Graph Library for Python. *Journal of Open-Source Software*, 7(79), 3968, <https://doi.org/10.21105/joss.03968>

proporciona una librería de grafos de alto rendimiento y flexible inspirada en NetworkX para Python.

Otras aproximaciones como qpqc¹⁹ proponen representaciones alternativas de los circuitos cuánticos, en este caso, usando el paquete TikZ de LaTeX.

Teniendo en cuenta el análisis del estado del arte previo, se propone una herramienta flexible y altamente escalable para la visualización de los circuitos cuánticos. El propósito de la solución tiene por objetivo atajar las deficiencias de los actuales métodos de representación de los circuitos cuánticos, los cuales se vuelven tediosos y poco intuitivos cuando o bien aumenta la complejidad del circuito o bien el número de cúbits. En estos casos, el tamaño del circuito cuántico usualmente sobrepasa las dimensiones de una pantalla estándar y es preciso hacer scroll para poder verlo completamente, haciendo más difícil su interpretación y causando un efecto negativo en la experiencia de usuario.

Para ello se propone una visualización interactiva que incluye herramientas de abstracción para agrupar varias puertas cuánticas en una única capa abstracta. Esto es, se trata de una herramienta versátil que permitirá definir:

- Reglas customizadas de abstracción en donde el usuario podrá definir exactamente qué operadores cuánticos agregar en cada capa abstracta.
- Reglas automatizadas basadas en técnicas como las de circuit cutting vistas anteriormente.

La solución propuesta se deberá implementar en una librería de Python de alto nivel y con capacidad para integrarse con las principales librerías del ámbito. Las reglas de la capa de abstracción deberán fijarse en el momento de la configuración y definición del circuito cuántico desde el propio Jupyter Notebook. Las agrupaciones customizadas se podrán determinar mediante encapsulación con funciones, declarando las dependencias, o con pragmas. Para ello se propone usar un método de definición de reglas sencillo e intuitivo, pero lo suficientemente versátil como para satisfacer las necesidades de los usuarios. Como punto de partida se puede tomar una sintaxis similar a la de Apache Airflow que propone una forma sencilla y elegante para definir los DAGs. En la Figura 15 se muestra un ejemplo de un DAG de Apache Airflow. En este caso el pipeline se inicia con la tarea task_a. Luego, las tareas task_b y task_c se ejecutarían en paralelo y finalmente, una vez concluida la tarea task_c, se ejecutaría la task_d. El DAG proporcionado por Apache Airflow cuenta con un código de leyenda que permite visualizar el estado de cada tarea: queued, running, success, failed, etc. Para definir las relaciones de dependencia entre tareas, en Airflow se usa el

¹⁹ qpqc. Consultado con fecha de 04/05/2023 en <https://github.com/qpqc/qpqc>

operador >>. La sintaxis para reproducir la relación de dependencia representada en la Figura 15 es la siguiente:

```
task_a >> [task_b, task_c]
```

```
task_c >> task_d
```

Las reglas automatizadas se podrán definir incorporando un flag que haga referencia al método de circuit cutting empleado.

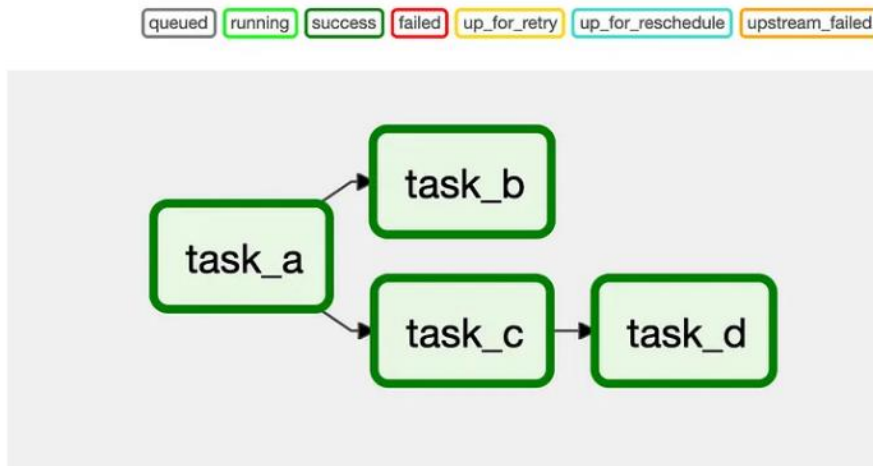


Figura 15: Ejemplo de definición de un DAG con Apache Airflow. Fuente de información: <https://medium.com/apache-airflow/apache-airflow-2-0-tutorial-41329bbf7211>

Para la representación final del circuito cuántico se propone una visualización interactiva similar a la usada en Quirk²⁰, con opción de *drill down* al clicar en la caja correspondiente para ver todas las subtareas contenidas en las distintas capas abstractas. De esta manera el usuario podrá ver tanto la versión agregada del circuito cuántico como la desagregada en el caso de querer analizar las componentes más elementales.

3.6 Representación de los resultados

Mientras el bit constituye la unidad básica de información en computación clásica, el bit cuántico o cúbit constituye la estructura de información fundamental en computación cuántica. Los cúbits poseen características propias, como la **superposición de estados** y el **entrelazamiento cuántico**, que abren nuevas formas de procesar la información.

²⁰ Quirk. Consultado con fecha de 04/05/2023 en <https://algassert.com/quirk>

En la Figura 16 se muestra una forma de representar los bits y los cúbits²¹. A diferencia del bit clásico que solo toma valores en el conjunto {0,1}, los cúbits tienen una complejidad mayor. Además de tomar los valores anteriores, puede estar en un estado de superposición cuántica de los estados 0 y 1.

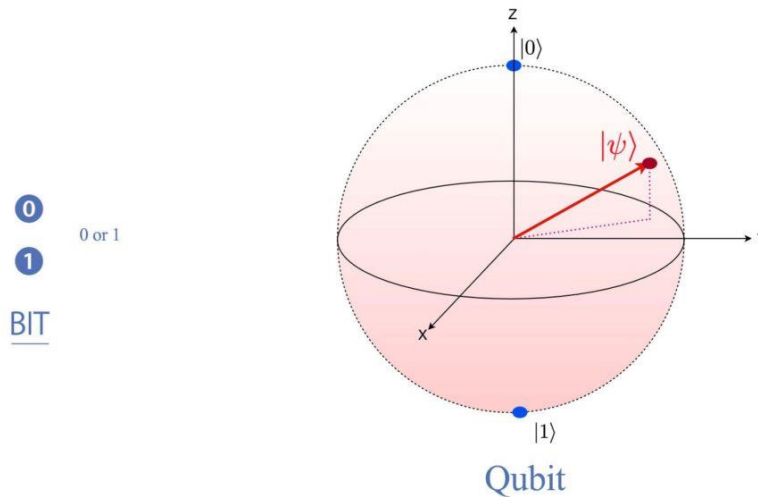


Figura 16: Representación de los bits (izquierda) y cúbits (derecha).

Para describir el estado de un cúbit se usan los vectores de estado. En el caso de un único cúbit, el vector de estado viene determinado por un vector bidimensional de norma unitaria. Este vector, contiene toda la información necesaria para describir el sistema cuántico de un cúbit. Así, un vector arbitrario $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, representa un estado cuántico si α y β son números complejos que contienen información de fase, verificando

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1}$$

Los vectores del estado cuántico,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ y } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

se denominan ket 0 y ket 1 respectivamente, y juegan un papel destacado, pues forman una base del espacio vectorial que describe el estado de un cúbit. Esto es, cualquier vector de estado $|\psi\rangle$ puede representarse como una combinación lineal de los elementos de la base

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

²¹ Representación de los bits y cúbits. Consultado con fecha de 25/04/2023. Fuente de información: <https://tecnologia.cibertux.com/2019/11/09/que-es-un-qubit-y-que-tiene-que-ver-con-la-computacion-cuantica/>

donde α y β son números complejos verificando la ecuación (1). Esta ecuación indica que en la medición se obtiene un estado o el otro. Es decir, debido a la naturaleza cuántica del cúbit, al realizar una medición, se altera el estado, rompiendo la superposición y colapsando en el estado que ha resultado de la medida y, en consecuencia, $\{\alpha, \beta\}$ se transforma en $\{0, 1\}$ de manera irreversible.

Los métodos de visualización juegan un papel importante tanto en la asimilación como en la interpretación de los resultados obtenidos en los circuitos cuánticos. Por otro lado, constituyen un apoyo esencial para comunicar los resultados obtenidos de manera efectiva y, a diferencia de otras formas de representar los datos como pueden ser las tablas, con la representación gráfica, la información se vuelve inteligible por lo que su interpretación se hace mucho más sencilla.

Los cúbits pueden representarse en 3 dimensiones mediante la esfera de Bloch²². La esfera de Bloch permite visualizar un estado cuántico de un cúbit como un vector tridimensional (ver Figura 16 para más detalles). Además de la esfera de Bloch, existen múltiples alternativas para visualizar los resultados de los circuitos cuánticos. Una de las más habituales es a través de un gráfico de barras o barplot.

La Figura 17 muestra un ejemplo de visualización de los resultados de un circuito cuántico sencillo formado por 2 cúbits mediante un diagrama de barras²³. En él se muestran los conteos de cada uno de los estados. Esto es, de los 1000 shots que se han ejecutado, en 496 se obtiene el estado 00 y en 504 el estado 11.

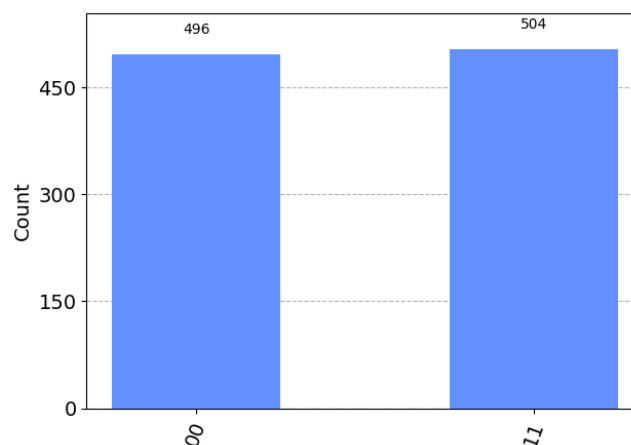


Figura 17: Ejemplo de gráfico de barras obtenido con el método `plot_histogram`.

²² Esfera de Bloch. Consultado con fecha de 25/04/2023 en https://en.wikipedia.org/wiki/Bloch_sphere

²³ Diagrama de barras obtenido a través de Qiskit. Consultado con fecha de 25/04/2023 en https://qiskit.org/documentation/tutorials/circuits/2_plotting_data_in_qiskit.html

En muchos casos puede ser interesante visualizar el estado del computador cuántico, por ejemplo, para tareas de depuración. Para ello, asumiendo que se tiene el estado real (bien sea por simulación o por tomografía), existen múltiples alternativas para visualizar el estado cuántico. Los recursos necesarios para este tipo de tareas crecen exponencialmente con el número de cúbits, por lo que estos métodos son recomendados únicamente para circuitos cuánticos de tamaño reducido. Qiskit ofrece varias alternativas para visualizar el estado de un computador cuántico. En las siguientes figuras se muestran los principales métodos de visualización del estado que ofrece la herramienta.

La Figura 18 muestra la forma estándar de representación de los estados cuánticos. El método `city` permite representar la parte real (izquierda) e imaginaria (derecha) de la matriz de estado como un diagrama de barras tridimensional. La Figura del ejemplo sugiere que la matriz de estado presenta valores estrictamente positivos únicamente en la parte real.

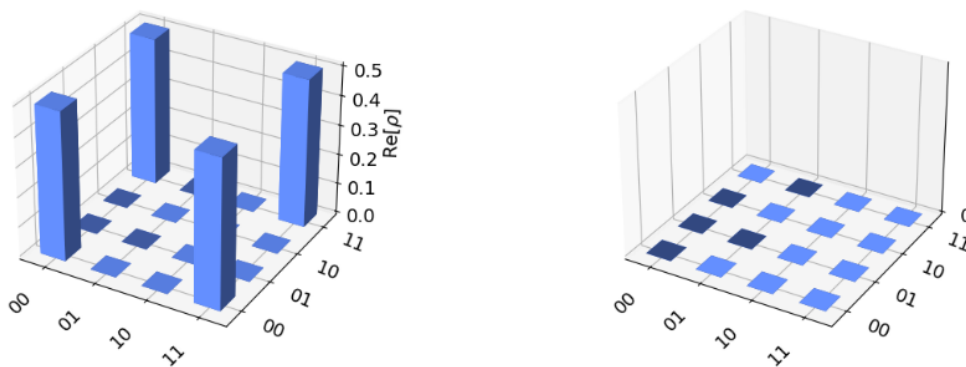


Figura 18: Ejemplo de representación de la matriz de estado con el método `plot_state_city`.

El método de Hinton permite visualizar la parte real e imaginaria de la matriz de estado de forma similar al caso anterior (método City). Tal y como se puede apreciar en la Figura 19, a diferencia del caso anterior, en el método Hinton se representa la matriz de estado mediante dos gráficas bidimensionales. Esto es, esta forma de visualización se obtiene proyectando el gráfico tridimensional del método City en dos dimensiones. Para ello, en lugar de usar las barras, la amplitud de los elementos de la matriz se establece a través del tamaño del recuadro y la dirección (positiva o negativa) se representa en base a un código de colores. Los valores positivos se representan con color blanco, los negativos con color negro y el valor 0 con color gris.

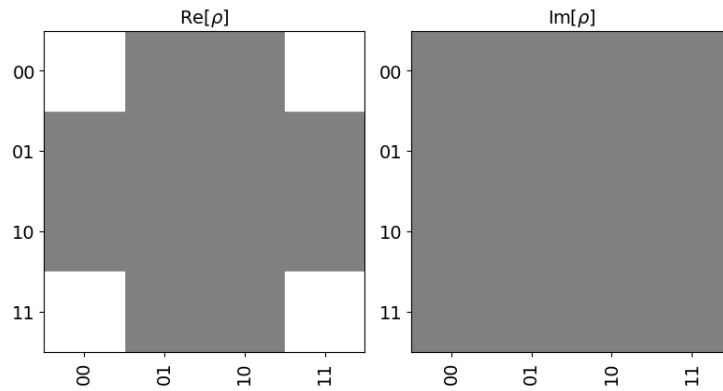


Figura 19: Ejemplo de representación de la matriz de estado con el método `plot_state_hinton`.

El método de Pauli (ver Figura 20 más detalles), se usa para representar una matriz de estado sobre las matrices de Pauli. El resultado final es un diagrama de barras *sparse* de los valores esperados, útil también para visualizar estados mixtos. El rango de los valores esperados oscila entre -1 y 1, y la altura de las barras representa los valores esperados.

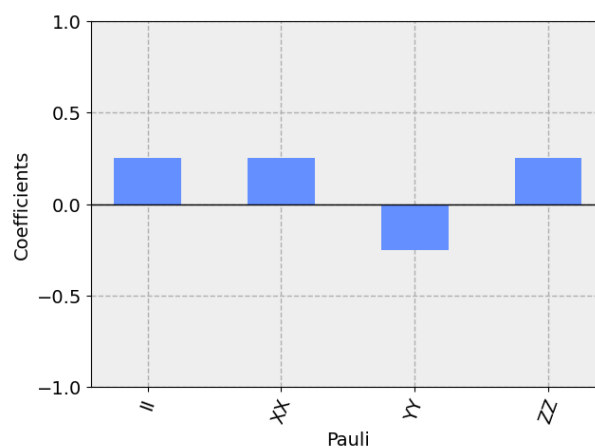


Figura 20: Ejemplo de representación de la matriz de estado con el método `plot_state_paulivec`.

La q-esfera²⁴ representa el estado de un sistema cuántico de uno o más cúbits asociando cada estado con un punto en la superficie de una esfera. A diferencia de la esfera de Bloch que ofrece una vista local del estado cuántico, la q-esfera proporciona una representación del estado cuántico desde el punto de vista global, permitiendo visualizar el estado cuántico en su totalidad. Por lo tanto, puede ser visto como una alternativa idónea para explorar aplicaciones y algoritmos cuánticos de un número reducido de cúbits. La Figura 21 muestra un ejemplo de representación de la q-esfera obtenida con el método *q-sphere*. Cada nodo se

²⁴ Q-esfera. Consultado con fecha de 26/04/2023 en <https://quantum-computing.ibm.com/composer/docs/iqx/visualizations>

representa por un punto. El radio de cada nodo es proporcional a la probabilidad de su estado base y la fase cuántica se representa mediante un código de colores que se refleja en la leyenda. Empezando en el polo norte de la q-esfera y progresando hacia el sur, cada sucesiva latitud tiene estados base con un mayor número de unos. La latitud de un estado base está determinada por su distancia de Hamming²⁵ desde el estado cero. Tal y como se aprecia en la Figura, la q-esfera permite representar de una manera compacta la información completa de un estado cuántico.

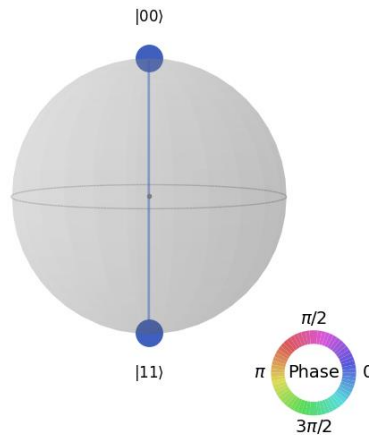
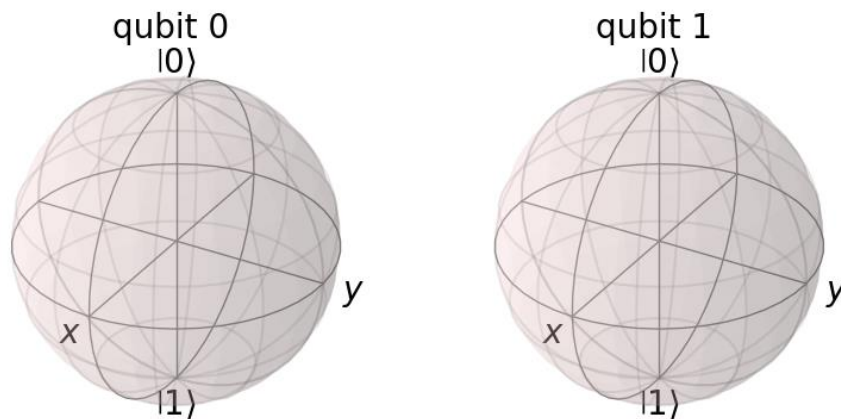


Figura 21: Ejemplo de representación del estado cuántico con el método `plot_state_qsphere`.

La Figura 22 muestra un ejemplo de representación del estado cuántico de todos los cúbits de un circuito cuántico con el método Bloch multivector. Este método permite visualizar el estado de los cúbits en superposición, el cual se destruye en el momento de la medición. Por ello, constituye una excelente opción para visualizar los resultados del simulador del vector de estados.

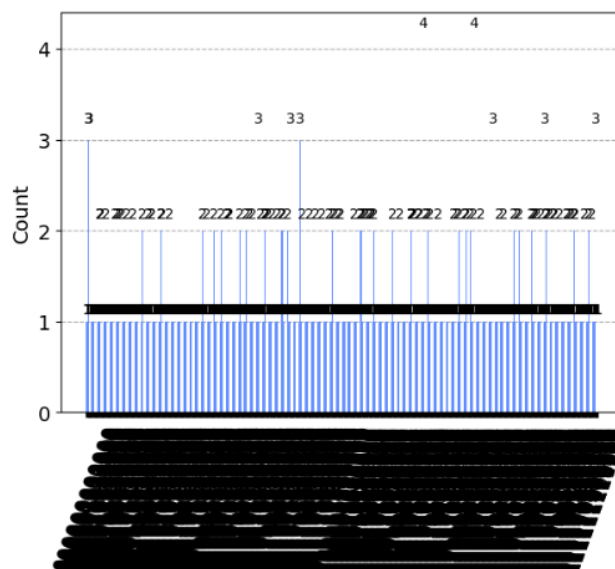


²⁵ Distancia de Hamming. Consultado con fecha de 26/04/2023 en https://en.wikipedia.org/wiki/Hamming_distance

Figura 22: Ejemplo de representación del estado de todos los cúbits con el método `plot_bloch_multivector`.

A pesar de los beneficios generales de los métodos visualización, las características y particularidades que rigen en el ámbito de la computación cuántica provocan que los métodos desarrollados hasta el momento cuenten con importantes limitaciones y no escalen a medida que el número de cúbits crece. Así, al ejecutar un circuito cuántico de n cúbits se obtienen 2^n resultados posibles. Es decir, la relación entre el número de inputs y outputs tiene un comportamiento exponencial, lo que hace que la información representada con los métodos clásicos colapse, haciendo muy difícil tanto la observación como la interpretación de la información. Por ello, los principales proveedores de servicios de computación cuántica como IBM tienen limitadas las representaciones gráficas en base a un número máximo de cúbits que depende del tipo específico de visualización²⁶. Así, por ejemplo, la visualización de los estados de un N-cúbit permite hasta un máximo de 15 cúbits, el diagrama de barras para las probabilidades permite hasta un máximo de 8 cúbits, la q-esfera hasta 5 cúbits y el vector de estados hasta 6 cúbits.

En la Figura 23 se reflejan perfectamente los problemas de escalabilidad de los métodos de representación de la información habituales. En el ejemplo se muestra la salida de un circuito cuántico de 10 cúbits. Como se puede observar, las etiquetas del barplot se solapan y no permiten ver y extraer la información relevante del gráfico.



²⁶ Visualizaciones de IBM Quantum. Consultado con fecha de 26/04/2023 en <https://quantum-computing.ibm.com/composer/docs/iqx/visualizations>

Figura 23: Ejemplo de representación de los resultados de un circuito cuántico de 10 cúbits.

Este hecho se puede acentuar aún más si las capacidades de la infraestructura cuántica son elevadas, ya que permitirá ejecutar circuitos cuánticos complejos en los que se acentúen más aun las limitaciones de los métodos de visualización tradicionales. Por ello, es importante investigar en métodos alternativos de visualización de la información que escalen de manera efectiva a medida que aumenta el número de cúbits.

En los últimos años, la comunidad científica ha abordado este problema y se han propuesto nuevas formas de visualización para acometer las limitaciones de los actuales sistemas de visualización. En Lamy (2019)²⁷ se propone una herramienta de visualización que permite representar el estado de múltiple cúbits y analizar la evolución a lo largo del tiempo de ejecución del circuito cuántico.

²⁷ Lamy, J. (2019). Dynamic Software Visualization of Quantum Algorithms with Rainbow Boxes. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - IVAPP, ISBN 978-989-758-354-4; ISSN 2184-4321, SciTePress, pages 155-163. DOI: 10.5220/0007247801550163

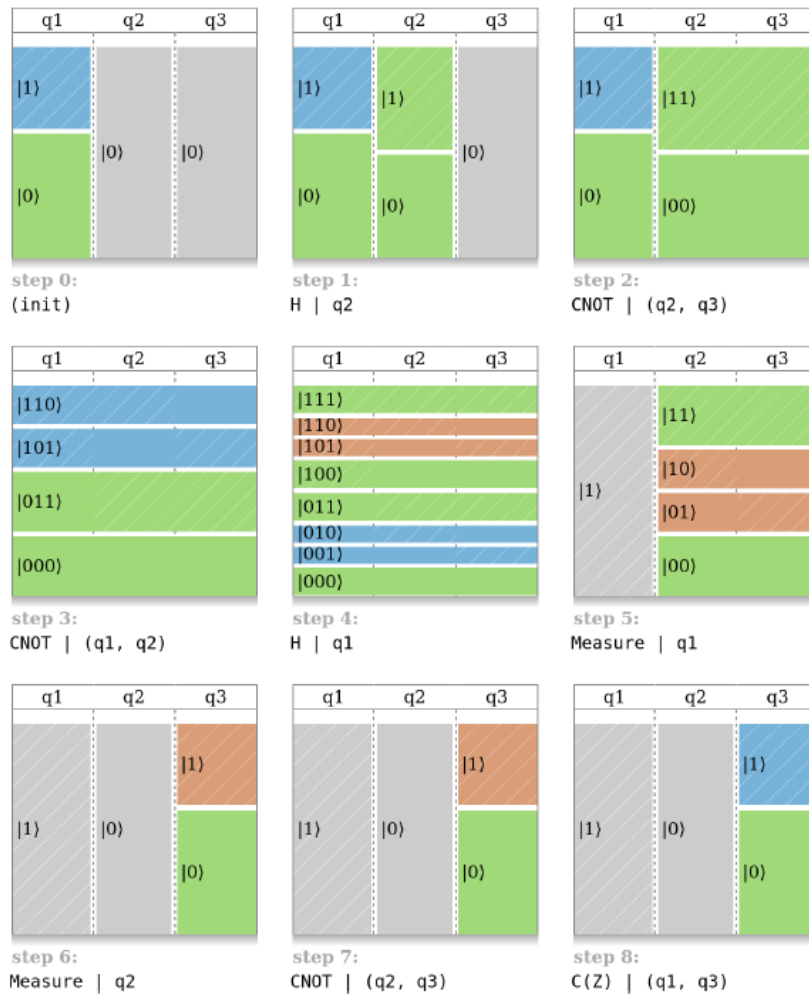


Figura 24: Ejemplo de visualización del estado de un sistema durante la ejecución del algoritmo de teletransportación cuántica.

En la Figura 24 se muestra un ejemplo de la propuesta de visualización aplicada al algoritmo de teletransportación cuántica, un algoritmo de gran relevancia en el ámbito de la criptografía. En cada paso del algoritmo, se proporciona una visualización del estado actual mediante cuadros con códigos de colores y se etiqueta la puerta aplicada. Los estados entrelazados se representan por cuadros que abarcan varios cúbits, mientras que los no entrelazados abarcan una única columna. La altura de las cajas es proporcional a la probabilidad de obtener ese estado en la medición y la fase se representa mediante un código de colores. El color gris se usa cuando no hay fase relativa, esto es, cuando el estado de los cúbits está perfectamente determinado y no hay superposición. En caso contrario, el tono del color indica la fase relativa (se toma el color verde para representar la fase de 0, es decir, cuando es igual a la fase de referencia).

Aunque el método de visualización propuesto en Lamy (2019) es muy informativo y puede ser una herramienta realmente útil para tareas de depuración, presenta importantes problemas de escalado. Esto es, se comporta bien para circuitos cuánticos pequeños que no presentan una importante complejidad, pero por la contra, a medida que los operadores de

los circuitos cuánticos y/o el número de cúbits crecen, el método de visualización propuesto presenta problemas de escalabilidad.

En Ruan et al. (2023)²⁸ se propone VENUS, una representación geométrica para la visualización del estado cuántico. Esta nueva forma de representación aporta una visualización novedosa que pretende abordar las debilidades de la esfera de Bloch para visualizar el entrelazamiento y la superposición cuántica. Al correlacionar de forma explícita las formas geométricas bidimensionales basadas en las características de la computación cuántica, se pueden representar de manera efectiva las amplitudes tanto de un solo cúbit como dos cúbits.

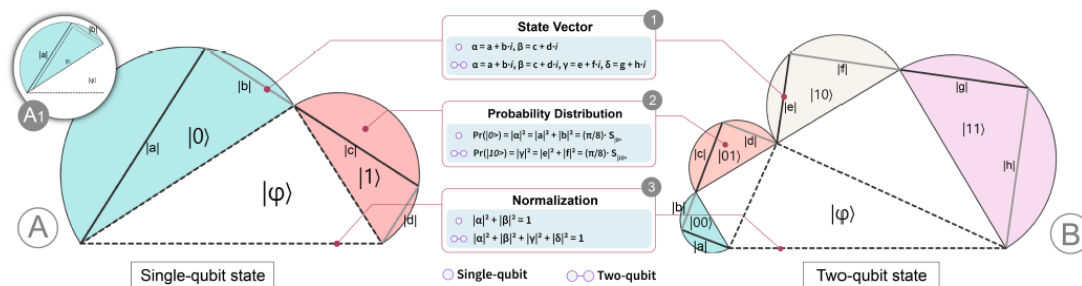


Figura 25: Diseño visual de VENUS.

La Figura 25 muestra el diseño visual de VENUS para representar el estado en el caso de un único cúbit (A) y dos cúbits (B). Los segmentos visualizan el vector de estado. La línea negra denota la parte real mientras que la línea gris denota la parte imaginaria. El área del semicírculo indica la probabilidad de medir el correspondiente estado. La longitud de la base del triángulo es 1 para garantizar la restricción de normalización.

Al igual que sucedía con el caso anterior, el método de representación geométrica propuesto en VENUS es útil para circuitos cuánticos con un número reducido de cúbits pero tiene importantes limitaciones de escalabilidad cuando se aplica a circuitos complejos.

En Bathe et al. (2013)²⁹ presentan un método de visualización denominado árbol binario de esferas de Bloch que permite representar de manera única un conjunto de estados puros arbitrarios de múltiples cúbits. El método de visualización combina la descomposición de Schmidt con la representación de la esfera de Bloch. En la Figura 26 se muestra un ejemplo

²⁸ Ruan, S., Yuan, R., Wang, Y., Lin, Y., Mao, Y., Jiang, W., ... & Guan, Q. (2023). VENUS: A Geometrical Representation for Quantum State Visualization. arXiv preprint arXiv:2303.08366.

²⁹ Barthe, A., Grossi, M., Tura, J., & Dunjko, V. (2023). Bloch Sphere Binary Trees: A method for the visualization of sets of multi-qubit systems pure states. arXiv preprint arXiv:2302.02957.

de la metodología para visualizar el efecto de cambiar un parámetro en un circuito cuántico variacional.

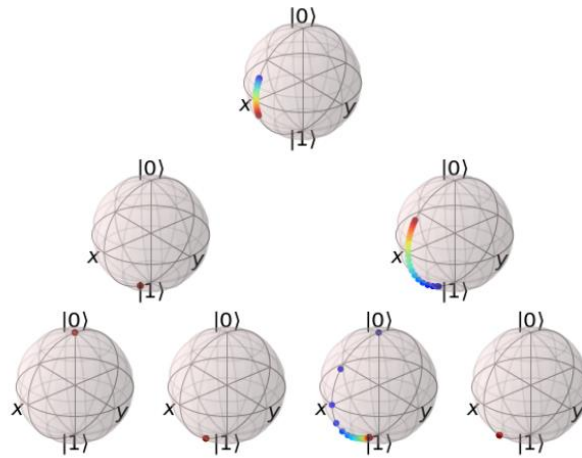


Figura 26: Ejemplo de visualización del árbol binario de esferas de Bloch.

En Miller et al. (2021)³⁰ y Migdatl et al. (2022)³¹ se proponen métodos alternativos de visualizar fenómenos cuánticos de manera intuitiva usando un enfoque interactivo. GraphStateVis es una aplicación web enfocada al análisis visual de los estados de los cúbits y sus grupos estabilizadores (ver Figura 27 para más detalles).

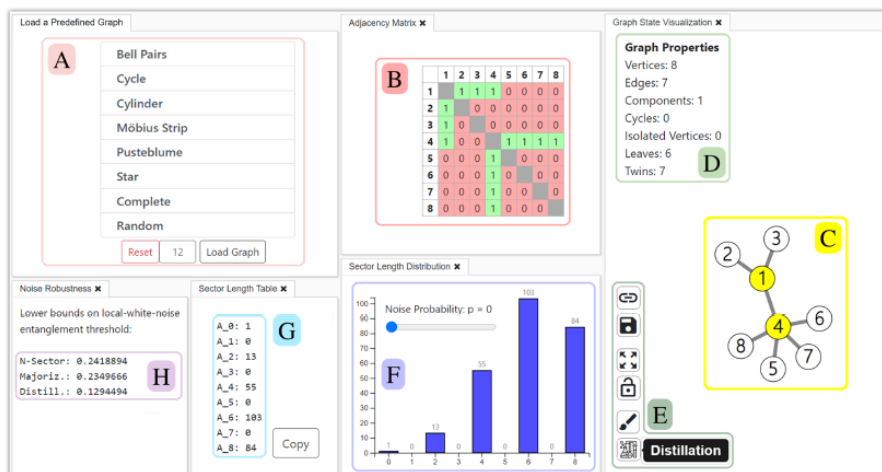


Figura 27: Ejemplo de visualización con GraphStateVis.

³⁰ Miller, M., & Miller, D. (2021, October). Graphstatevis: Interactive visual analysis of qubit graph states and their stabilizer groups. In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE) (pp. 378-384). IEEE.

³¹ Migdał, P., Jankiewicz, K., Grabarz, P., Decaroli, C., & Cochin, P. (2022). Visualizing quantum mechanics in an interactive simulation–Virtual Lab by Quantum Flytrap. Optical Engineering, 61(8), 081808-081808.

Por otra parte, Virtual Lab permite visualizar estados cuánticos entrelazados y mostrar medidas de entrelazamiento, incluyendo visualizaciones interactivas de la notación ket y una visualización similar a la de un mapa de calor de los operadores cuánticos (ver Figura 28 para más detalles).

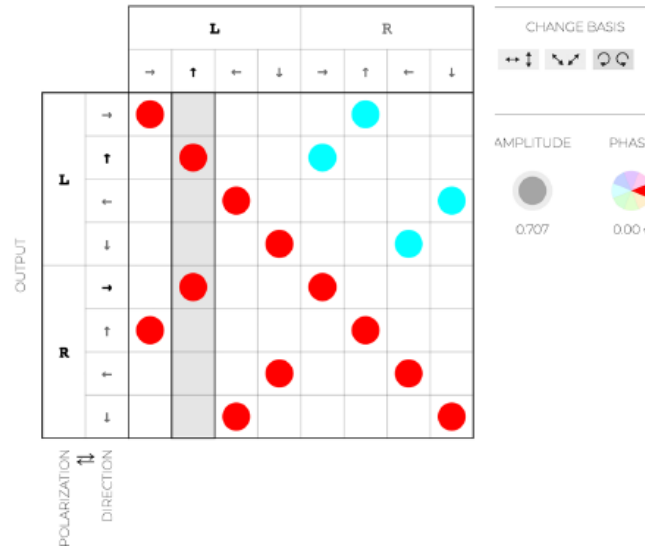


Figura 28: Ejemplo de visualización de Virtual Lab.

En Tuan et al. (2021)³² se presenta tqix, una librería open source escrita en Python para la visualización de medidas cuánticas, tomografía y metrología cuánticas. Esta librería puede usarse tanto para estados como observaciones cuánticas y posee varios back-end para la simulación en la medición cuántica.

Las metodologías revisadas facilitan el análisis e interpretación de los resultados obtenidos en los circuitos cuánticos. No obstante, muchas de ellas están especialmente enfocadas a circuitos cuánticos con un número reducido de cúbits y presentan problemas de escalado cuando el número de cúbits empleado es relativamente elevado. Por ello, se propone una metodología que combina las ventajas de los enfoques interactivos, como los propuestos en Miller et al. (2021) y Migdatl et al. (2022), con una manera alternativa de visualizar la información especialmente pensada para contextos con un gran número de cúbits.

Entre las ventajas de los métodos de visualización interactiva destacan:

³² Tuan, K. Q., & Nguyen, H. Q. (2021). tqix: A toolbox for quantum in X: X: quantum measurement, quantum tomography, quantum metrology, and others. *Computer Physics Communications*, 263, 107902.

- **Facilidad de explotación:** la visualización interactiva facilita la explotación de los datos de manera más intuitiva para descubrir patrones, relaciones o tendencias, que de otra manera podría no ser tan evidente.
- **Personalización:** los usuarios pueden ajustar los filtros y parámetros de las visualizaciones para enfocarse en aquellos datos específicos que se quieren analizar, permitiendo personalizar la presentación en base a las necesidades.
- **Experiencia atractiva:** las visualizaciones interactivas usualmente son más atractivas que las visualizaciones estáticas.
- **Mejora la comprensión:** las visualizaciones interactivas ayudan a comprender mejor los datos al permitir analizar la información desde diferentes perspectivas.

Los dashboards de resultados se implementarán en el módulo del *Quantum Analytics Dashboard*. El módulo contará con una galería de opciones de visualización interactiva tanto del ámbito de la computación cuántica (simulación cuántica y contexto real), como de propósito general (p. ej. métodos de visualización convencionales para realizar análisis estadísticos de las distribuciones de los resultados). Además, se proporcionarán diferentes plantillas que estarán disponibles para que los usuarios puedan realizar sus análisis. Se trata pues, de un sistema versátil y suficientemente flexible para que los usuarios puedan construir sus propios dashboards e incluso customizar su apariencia.

Los usuarios podrán definir los filtros que mejor se ajusten a sus necesidades para quedarse con la información más relevante y despreocuparse de la residual. Esta funcionalidad facilita la escalabilidad del sistema propuesto, ya que ayuda a combatir la explosión de combinaciones que se produce con el aumento del número de cúbits. Para garantizar el buen funcionamiento del sistema, el procesamiento de la información se realizará en el servidor, dejando la parte del cliente únicamente para mostrar la información ya procesada, y se usarán cubos OLAP (Online analytical processing) para acelerar las tareas de procesamiento de datos.

Además de las representaciones más habituales usadas en el ámbito de la computación cuántica (p. ej. las incluidas en la librería Qiskit), se incorporarán nuevas representaciones orientadas a circuitos cuánticos de gran dimensionalidad (número elevado de cúbits).

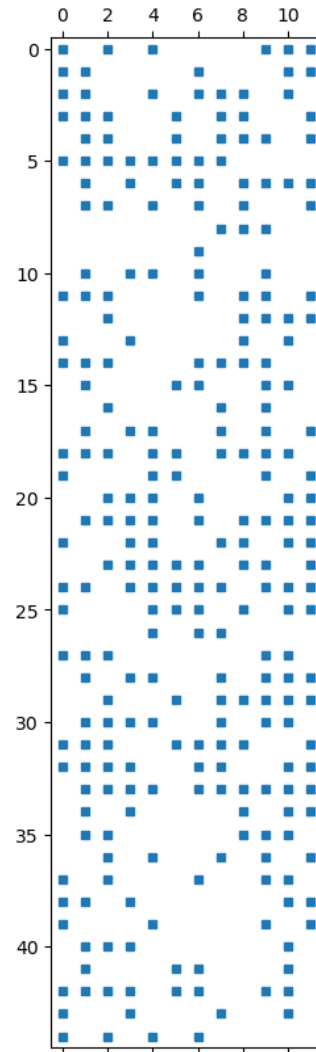


Figura 29: Propuesta de representación gráfica.

Entre las posibles representaciones alternativas se propone una nueva representación tridimensional similar a la de la Figura 29, en donde en el eje de la x se representaría el resultado de la medida para cada cúbit (en este caso 12 cúbits), en el eje de la y se representaría cada una de las 2^n posibles combinaciones y en el eje z, a diferencia del caso de la Figura en donde se representa el indicador de si ha habido coincidencia o no, se mostraría el recuento de coincidencias obtenidas al ejecutar el circuito cuántico un número determinado de shots. De esta manera, el resultado final podría ser una superficie tridimensional, un mapa de curvas de nivel o una representación similar a la de la Figura

29 distinguiendo la magnitud del eje z mediante un código de colores. Una representación similar a la de la propuesta puede encontrarse en Lubinski et al. (2023)³³.

Otra posible representación sería similar a la propuesta en Wu et al. (2023)³⁴. Esta representación (ver Figura 30 para más detalles) puede ser usada con múltiple propósito. Por un lado, puede usarse en contextos de simulación como gráfico de visualización alternativo a la esfera de Bloch, esto es, el gráfico de la figura puede verse como una esfera de Bloch en 2D (vista desde arriba), pero a diferencia de la representación tridimensional en donde es sencillo distinguir el hemisferio (norte o sur), la distinción podría realizarse mediante un código de colores. Por otro lado, este tipo de representación es también fácilmente adaptable al contexto real. En este caso, se podría adaptar ligeramente la gráfica para mostrar la misma información que en la Figura 29. Para ello, habría que crear tantos círculos concéntricos como cúbits. Las etiquetas que bordean el círculo podrían denotar las 2^n posibles combinaciones y, de igual forma que sucedía con el caso anterior, el número total de coincidencias podría reflejarse mediante un código de colores. Como se puede apreciar, este tipo de representación gráfica presenta un buen comportamiento de escalado, pudiendo ser muy útil para mostrar los resultados obtenidos en el computador cuántico.

Una vez finalice la ejecución del circuito cuántico, bien sea en el emulador o el computador cuántico, el usuario recibiría una notificación informativa sobre el estado de su trabajo y podría acceder al módulo del *Quantum Analytics Dashboard* a través del portal de usuarios. Una vez dentro, el investigador podrá hacer uso del catálogo de componentes para tomar aquellas opciones de visualización que más se adecúen a sus necesidades.

³³ Lubinski, T., Johri, S., Varosy, P., Coleman, J., Zhao, L., Necaie, J., ... & Proctor, T. (2023). Application-oriented performance benchmarks for quantum computing. IEEE Transactions on Quantum Engineering.

³⁴ Wu, D., Rossi, R., Vicentini, F., Astrakhantsev, N., Becca, F., Cao, X., ... & Carleo, G. (2023). Variational Benchmarks for Quantum Many-Body Problems. arXiv preprint arXiv:2302.04919.

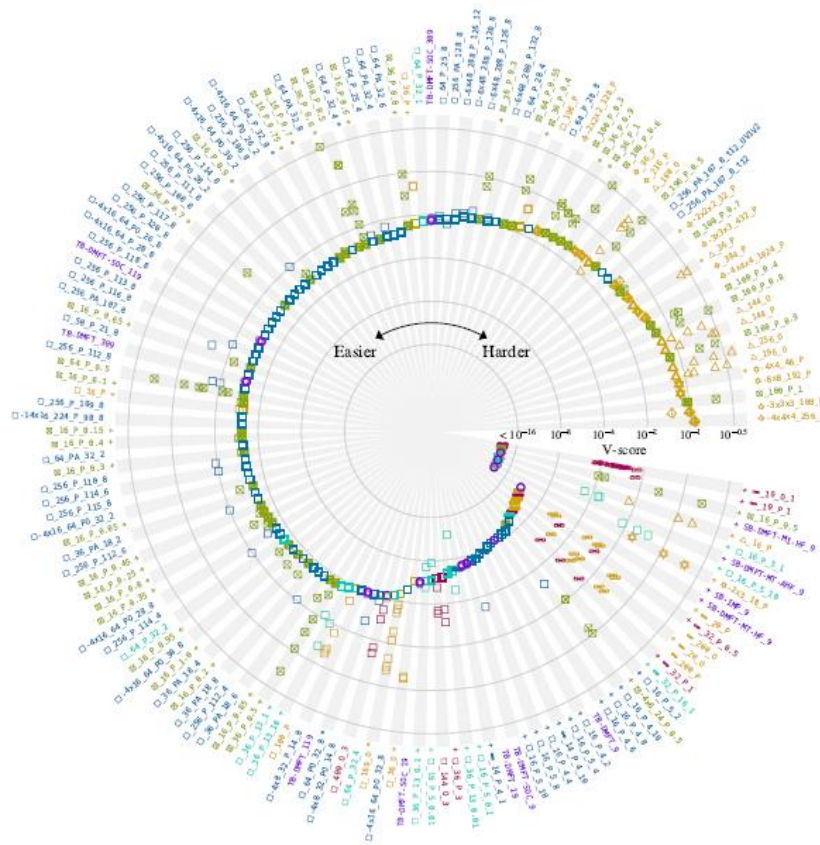


Figura 30: Propuesta de visualización para la representación de resultados. Fuente de información: <https://arxiv.org/pdf/2302.04919.pdf>

3.7 Gestión de la contabilidad

Conocer la tipología de los trabajos que se ejecutan en la infraestructura cuántica es vital, tanto desde el punto de la **contabilidad** como de la **monitorización**. La información recabada puede ser usada con múltiples propósitos, como emitir informes sobre el uso de la infraestructura, conocer cuáles son las características de los procesos y tareas ejecutadas o el consumo energético asociado. El objetivo perseguido es analizar la huella energética vinculada a la infraestructura de supercomputación y asentar las bases para avanzar en el uso de las tecnologías verdes. Por otra parte, la monitorización de procesos constituye una fuente útil de información que puede ser explotada para mejorar la gestión de los recursos disponibles.

En ambos casos, es necesario previamente definir qué KPIs son los más adecuados para este nuevo servicio de computación y ver cómo adaptarlos en el actual sistema de gestión de la contabilidad del centro de cálculo.

El sistema de gestión de la contabilidad de los entornos HPC habitualmente se basa en el consumo core/hora. Por otra parte, los entornos cloud, además del consumo core/hora usualmente también tienen en cuenta la memoria RAM y la capacidad de almacenamiento.

Para poder aportar una propuesta de valor, se ha analizado el modo de gestión de la contabilidad de los principales proveedores de servicios de computación cuántica. Más concretamente se han analizado Amazon Braket, IBM, Azure Quantum, Google Quantum AI, D-Wave y Pasqal.

En la Tabla 1 se recoge un resumen del modo de gestión de la contabilidad de los principales proveedores de servicios cuánticos. Como se puede apreciar, se observan diversas opciones que van desde modos de gestión de la contabilidad básicos basados únicamente en tiempo de cómputo hasta opciones más complejas que tienen cuenta el número de puertas, su complejidad o el número de shots.

Amazon Braket³⁵ contabiliza tanto la ejecución del circuito cuántico como su re-ejecución (“shots”) haciendo distinción en base al proveedor de hardware / tipo de QPU. En el caso de la ejecución de circuitos cuánticos sobre emuladores, solo se tiene en cuenta el tiempo de uso, de manera análoga a como se hace con las AWS EC2.

IBM ofrece sus servicios de computación cuántica a través de un catálogo de planes (open, pay-as-you-go y premium) y su gestión de la contabilidad se basa únicamente en el tiempo de cómputo.

Por otro lado, Azure y Google Quantum AI poseen diversos servicios de computación cuántica, tanto propios como de sus partners (IonQ, Quantinuum, Rigetti, etc.)³⁶. En el caso de los partners, son ellos mismos los que gestionan la contabilidad. IonQ gestiona la contabilidad en base al número de puertas, la complejidad y el número de lanzamientos (gate-shots), mientras que Quantinuum lo hace en base al HQC, una expresión matemática que depende del número de operaciones y el número de capturas ejecutadas. Por otra parte, al igual que en el caso de IBM, Rigetti, D-Wave y Pasqal gestionan la contabilidad únicamente en base al tiempo de cómputo.

PROVEEDOR	HARDWARE	CONTABILIDAD
AMAZON BRAKET	IonQ Device (IonQ)	Proveedor de hardware

³⁵ Gestión de la contabilidad de Amazon Braket. Consultado con fecha de 31/03/2023 en <https://aws.amazon.com/es/braket/pricing/>

³⁶ Gestión de la contabilidad de los partners de Microsoft Azure y Google Quantum AI. Consultado con fecha de 31/03/2023 en <https://learn.microsoft.com/es-es/azure/quantum/pricing?tabs=tabid-aria%2Ctabid-standard%2Ctabid-AQcreditsRigetti%2Ctabid-payasgo%2Ctabid-learndevelop&pivots=ide-computing>

	Lucy (OQC) Aquila (Quera) Aspen-M (Rigetti) Borealis (Xanadu)	Tarea Shot
IBM	QPU (5 y 7 cúbits) Falcon (27 cúbits) Eagle (127 cúbits) Osprey (433 cúbits)	Plan (open, pay-as-you-go, premium), basado en el tiempo de cómputo
AZURE QUANTUM	Infraestructura Azure Quantum	Plan (Learn & develop y Performance at scale), basado en tiempo de cómputo
	IonQ	Número de puertas Complejidad de las puertas Número de lanzamientos (gate-shots)
	Quantinuum	Créditos cuánticos del Sistema H (HQC)
	Rigetti	Tiempo de cómputo
GOOGLE	IonQ	Número de puertas Complejidad de las puertas Número de lanzamientos (gate-shots).
D-WAVE	2000Q Advantage	Tiempo de cómputo
PASQAL	Generation 1 Generation 2 (en desarrollo)	Tiempo de cómputo

Tabla 1: Modo de gestión de la contabilidad de los principales proveedores de servicios cuánticos.

Teniendo en cuenta el análisis del estado de arte anterior, para la gestión de la contabilidad del servicio de computación cuántica se definen los siguientes KPIs:

- Número de cúbits.

- Número de operaciones de inicialización.
- Número de operaciones de un solo cúbit.
- Tiempo promedio de operaciones de un solo cúbit.
- Número de operaciones nativas de 2 cúbits.
- Tiempo promedio de operaciones de 2 cúbits.
- Número de operaciones de dos cúbits con varios controles.
- Tiempo promedio de operaciones de 2 cúbits con varios controles.
- Número de medidas del estado.
- Número de shots.
- Tiempo de cómputo por shot.
- Consumo energético asociado al periodo de cómputo.

Es importante destacar que los KPIs propuestos son medibles únicamente bajo la propia infraestructura cuántica. En el caso de las ejecuciones de circuitos cuánticos en proveedores externos (Amazon Braket, Azure Quantum, Google Quantum AI, IBM, etc.), únicamente se podrá medir la facturación desglosada por servicios, bien sea real o mediante un sistema de créditos configurado previamente.

4. Conclusiones

La computación cuántica constituye una de las disciplinas más disruptivas de las últimas décadas. Se basa en las propiedades de la mecánica cuántica como la superposición o el entrelazamiento cuántico y permite el procesamiento de la información en paralelo. Aunque todavía es necesario hacer frente a los desafíos que supone este cambio de paradigma, los importantes avances conseguidos en los últimos años le hacen presagiar un gran futuro. La computación cuántica tiene aplicaciones en ámbitos diversos entre los que se incluyen optimización de operaciones, finanzas, farmacia, genómica, ciberseguridad y criptografía, diseño de materiales o simulación de fenómenos nucleares.

El CESGA, bajo el marco del Polo de Tecnologías Cuánticas, está centrando los esfuerzos para posicionar a Galicia como un referente en computación cuántica, tanto a nivel nacional como internacional. Desde finales del 2021, el CESGA ha incorporado a la infraestructura del Finisterrae III el primer emulador cuántico instalado en España. Este emulador está diseñado para la investigación y explotación de algoritmos que requieran de la generación

de números aleatorios. A esta infraestructura se añadirá a finales del 2023 uno de los primeros ordenadores cuánticos de Europa y el más potente de España hasta la fecha. El equipamiento cuántico contratado a Fujitsu, además del ordenador cuántico incluye un computador de altas prestaciones, un emulador cuántico y un sistema de almacenamiento.

Para poder llevar a cabo la integración del equipamiento cuántico contratado a Fujitsu en una infraestructura de supercomputación como la del CESGA, se deben abordar una serie de desafíos como el control de acceso, la gestión de colas, las interfaces de comunicación o la gestión de la contabilidad. Para ello, se presenta una propuesta de arquitectura global que permite engranar las diferentes soluciones de supercomputación disponibles en la organización, tanto clásicas como cuánticas. Se trata pues de una arquitectura flexible y modular que permite satisfacer tanto las necesidades actuales como las que puedan surgir a futuro. La arquitectura propuesta permite a los usuarios que puedan acceder al catálogo de productos disponibles (clásicos y cuánticos) mediante el portal de usuarios de la Organización, previa autorización mediante el sistema de LDAP.

Desde los distintos frontales, se podrán emitir trabajos al gestor de colas Slurm para los distintos dominios. La propuesta contempla también la integración con proveedores de servicios de computación cuántica externos como Amazon Braket, Azure Quantum o Google Cloud Platform. Con estas funcionalidades, se puede complementar la infraestructura propia de un centro de supercomputación como el CESGA, proporcionando a la comunidad científica las herramientas necesarias para poder llevar a cabo los análisis o pruebas que requiera la investigación.

Finalmente, y al margen de los desafíos que supone la integración de la infraestructura cuántica, se proporciona también una solución para abordar la gestión de la contabilidad y la representación cuántica (tanto de los circuitos cuánticos como de los resultados) en ámbitos complejos donde las representaciones habituales muestran problemas de escalado.

Para poder llevar a cabo la propuesta descrita de forma exitosa se tienen que seguir las recomendaciones a los requerimientos descritos mediante una serie de proyectos de investigación que se detallan a continuación. La ejecución de dichos proyectos puede llevarse a cabo, en algunos casos, en paralelo siguiendo las dependencias que se muestran en el diagrama Pert (ver Figura 31 para más detalles), donde se parte de la arquitectura inicial de un centro de supercomputación para llegar a la arquitectura ideal descrita en la Sección 2 del presente documento.

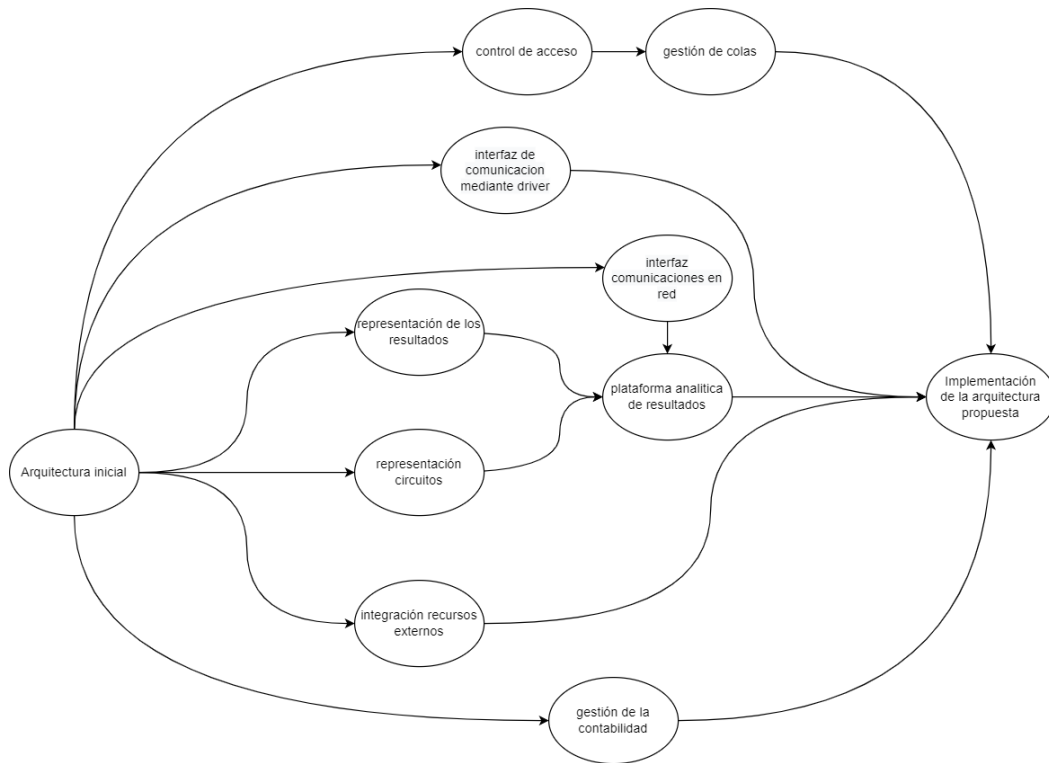


Figura 31: Diagrama Pert.

4.1 Control de acceso

Necesidad: una serie de frontales, protegidos que permitan el uso completo de los productos y servicios internos de la plataforma.

Recomendación: despliegue y protección de dichos servicios tal y como se especifica en la Sección 3.1.

Perfiles necesarios: 1 administrador de sistemas Unix, 1 experto en ciberseguridad y 1 arquitecto de sistemas.

Estimación de tiempo: 3 meses.

4.2 Gestión de colas

Necesidad: unificar bajo el mismo gestor, pero diferenciando dominios de aplicación y área los distintos recursos de cómputo.

Recomendación: desacoplar los sistemas de QPU de los sistemas HPC que suelen traer por defecto y la implementación de dominios bajo el mismo gestor de tareas.

Perfiles necesarios: 1 administrador de sistemas Unix y 1 experto en sistemas clúster.

Estimación de tiempo: 3 meses.

4.3 Interfaz de comunicación vía driver

Necesidad: reducir al máximo la latencia de comunicaciones y/o ejecución de instrucciones cuánticas entre un sistema clásico y otro cuántico.

Recomendación: implementación de un driver para el envío y recepción de instrucciones mediante un bus de comunicaciones.

Perfiles necesarios: 2 arquitectos de computadores y 2 ingenieros de firmware.

Estimación de tiempo: 15 meses.

4.4 Interfaz de comunicaciones en red

Necesidad: optimizar las comunicaciones entre los sistemas clásicos y cuánticos mediante la red.

Recomendación: implementación de un protocolo de comunicaciones sobre TCP/IP.

Perfiles necesarios: 1 experto en diseño de algoritmos y 2 ingenieros de telecomunicaciones o redes.

Estimación de tiempo: 15 meses.

4.5 Integración con recursos externos

Necesidad: integrar la infraestructura cuántica con proveedores de servicios externos.

Recomendación: implementar un SAAS que permita la integración desde el superordenador con dichos recursos, la gestión de las tareas y facturación para todas las cuentas.

Perfiles necesarios: 1 arquitecto de soluciones cloud, 1 experto en ciberseguridad y 2 DevOps.

Estimación de tiempo: 18 meses.

4.6 Representación de los circuitos

Necesidad: crear un sistema de visualización de circuitos cuánticos.

Recomendación: implementar una representación de los circuitos cuánticos interactiva incorporando herramientas de abstracción que permitan poder englobar un conjunto de puertas en una única capa abstracta.

Perfiles necesarios: 1 experto en computación cuántica y 2 expertos en UX.

Estimación de tiempo: 12 meses.

4.7 Plataforma analítica de los resultados

Necesidad: crear una plataforma que agilice el análisis de los resultados obtenidos en computación cuántica.

Recomendación: desarrollar una plataforma analítica que incorpore un catálogo de componentes para que los usuarios puedan implementar dashboards interactivos customizados que mejor se ajusten a sus necesidades.

Perfiles necesarios: 3 DevOps.

Estimación de tiempo: 12 meses.

4.8 Representación de los resultados

Necesidad: disponer un catálogo de visualización especialmente adaptado al contexto de circuitos cuánticos con un número elevado de cúbits.

Recomendación: desarrollar un sistema de visualización interactiva que permita analizar las salidas de los circuitos cuánticos. El sistema deberá incorporar tanto las representaciones habituales como nuevas formas de visualización especialmente adaptadas a contextos con un número elevado de cúbits.

Perfiles necesarios: 2 expertos en big data, 1 analista de BI y 1 experto en computación cuántica.

Estimación de tiempo: 9 meses.

4.9 Gestión de la contabilidad

Necesidad: disponer un sistema de monitorización y gestión de la contabilidad que permita conocer la tipología y el consumo energético de los trabajos que se ejecutan desde un ordenador cuántico.

Recomendación: es importante definir y medir un conjunto acotado de KPIs que ayuden a conocer las características de los circuitos cuánticos ejecutados (número de cúbits, número de puertas cuánticas, complejidad, número de shots, etc.).

Perfiles necesarios: 1 experto en computación cuántica y 1 ingeniero de software.

Estimación de tiempo: 12 meses.