

# Curso de Qiskit

## Instalación e Introducción

Pablo González

Qiskit Summer Jam Mexico, Agosto 2021



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# Introducción

Tradicionalmente, nuestras computadoras han funcionado a base de unos y ceros para realizar la infinidad de maravillas y comodidades que estos han traído; pero con el paso del tiempo el hecho de seguir usando un sistema que opera de esta manera representa una limitante en muchas tareas, ya que un bit solo puede ser 1 o 0.

En cambio, en la computación cuántica, intervienen las leyes de la mecánica cuántica, y la partícula puede estar en superposición coherente: puede ser 0, 1 y puede ser 1 y 0 a la vez (dos estados ortogonales de una partícula subatómica). Eso permite que se puedan realizar varias operaciones a la vez, según el número de cúbits. A más números de cúbits, mas exponencial es la potencia.



# Introducción a la computación cuántica

Esta idea fue propuesta por Paul Benioff, quien expuso su teoría para aprovechar las leyes cuánticas en el entorno de la computación. En vez de trabajar a nivel de voltajes eléctricos, se trabaja a nivel de cuanto.

Dotando a las mismas de una alta velocidad con la que podrían acelerar el descubrimiento de nuevos medicamentos, descifrar los sistemas de seguridad criptográfica más complejos, diseñar nuevos materiales, modelar el cambio climático y sobrecargar la inteligencia artificial, según dicen los expertos en el área.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización



# Anaconda

Una de las herramientas principales para este curso es el uso del lenguaje Python; se ha buscado simplificar mucho el uso de la programación para el área de la computación cuántica que ahora podemos simular la respuesta de un circuito cuántico con este lenguaje.

Para ello vamos a necesitar el entorno de Anaconda, el cual nos provee de diferentes ambientes de programación para el uso de Qiskit. A continuación, una rápida guía para instalar Anaconda dependiendo de nuestro sistema operativo. Pedimos a los usuarios ir a la sección 4 'Herramientas locales' para encontrar el link que los lleve al enlace de descarga.



# Anaconda dependiendo nuestro entorno

Antes de comenzar la descarga, recomendamos revisar el sistema operativo (SO por sus siglas en ingles) de nuestro equipo. Una vez seguros del SO de nuestra computadora, y de la forma de instalación (recomendando las opciones graficas), descargamos el instalador correspondiente.

## Anaconda Installers

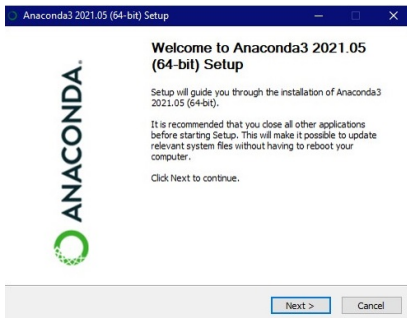
Windows 	MacOS 	Linux 
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (477 MB)	64-Bit Graphical Installer (440 MB)	64-Bit (x86) Installer (544 MB)
32-Bit Graphical Installer (409 MB)	64-Bit Command Line Installer (433 MB)	64-Bit (Power8 and Power9) Installer (285 MB)
		64-Bit (AWS Graviton2 / ARM64) Installer (413 M)
		64-bit (Linux on IBM Z & LinuxONE) Installer (292 M)





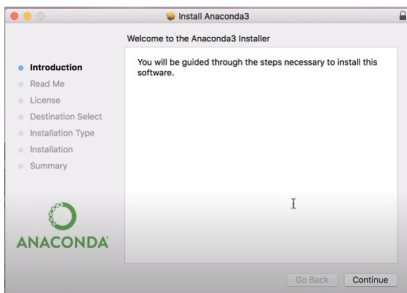
# Anaconda en Windows

Para el caso de Windows, descargaremos el instalador correspondiente. Una vez se ha descargado, nos aparecerá la siguiente ventana. Tan solo necesitas leer las instrucciones del instalador hasta terminar. Hecho esto, tendremos Anaconda instalado.



# Anaconda en macOS

El caso de macOS es un poco más sencillo que en Windows, tan solo basta con seleccionar la opción recomendada de instalador grafico (Graphical Installer) y seguir los pasos que nos dará el instalador. Este paso es recomendado para usuarios inexpertos pero el usuario es libre de usar el instalador por líneas de comando (Command Line).



# Anaconda en Linux

Para instalar Anaconda se necesita descargar la opción recomendada (la cual es la primera de la lista de descargas) y posteriormente, realizar los siguientes pasos.

**Linux** 

**Python 3.8**

64-Bit (x86) Installer (544 MB) ←

64-Bit (Power8 and Power9) Installer (285 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (413 M)

64-bit (Linux on IBM Z & LinuxONE) Installer (292 M)



# Anaconda en Linux

Accedemos a la terminal de nuestra computadora, tipiamos el siguiente comando y ejecutamos pulsando a Enter:

```
1 bash ~/Downloads/Anaconda3-2020.07-Linux-x86_64.sh
```

Hecho esto, nos desplegará el siguiente mensaje, el cual es el acuerdo de licencia de Anaconda; para aceptarlo, tipiamos `yes`

```
Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

    * Install and use the Anaconda Individual Edition (which was formerly known as Anaconda Distribution),
    * Modify and create derivative works of sample source code delivered in Anaconda Individual Edition from Anaconda's repository; and
    * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below.

Do you accept the license terms? [yes|no]
[no] >>> yes
```



# Anaconda en Linux

A continuación, nos dirá una ruta predeterminada para la instalación, recomendamos solo pulsar Enter para continuar con la descarga de complementos y paquetes, o especificar una ruta diferente. El proceso de descarga toma alrededor de 5 a 10 minutos en completarse, por lo que pedimos ser pacientes en este punto.

```
Anaconda3 will now be installed into this location:  
/home/usrx/anaconda3
```

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below



# Anaconda en Linux

Finalizado, cerramos y abrimos la terminal para tipiar 'jupyter notebook', para conocer la ruta de acceso hacia la carpeta de anaconda desde donde podrá ser ejecutada la aplicación, o tambien accedando como se muestra en la imagen, la cual es la opción recomendada, finalizando con la instalación.

```
usrx@usrx:~$ ./anaconda3/bin/anaconda-navigator
```



# Instalación de Qiskit

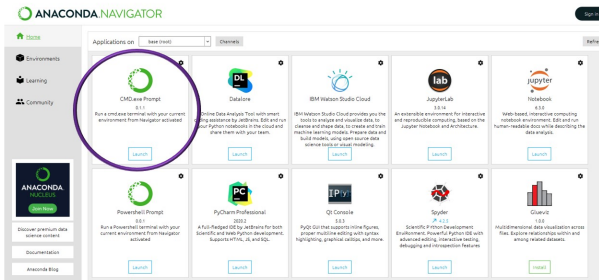
Para instalar Qiskit de manera local en nuestra computadora, basta con utilizar el nuevo entorno que hemos descargado, este procedimiento de instalación de Qiskit es el mismo para todos los sistemas operativos.

Primero, nos interesa arrancar el nuevo entorno de Anaconda, para ello puedes utilizar el buscador en Windows o macOS tipiendo 'anaconda3' para buscar el ejecutable y en Linux entrando mediante Jupyter Notebook.



# Instalación de Qiskit

Hecho esto, nos aparecerá un logo en nuestra pantalla hasta que nos aparezca el navegador de Anaconda, desde aquí podremos acceder a los múltiples ambientes de programación que hemos instalado. En este caso, nos interesa el terminal de Conda, el cual esta remarcado en la siguiente imagen.





# Instalación de Qiskit

Una vez aquí, debemos crear un entorno virtual para poder preparar toda la paquetería y módulos de Qiskit; crear un entorno también nos es útil para crear un espacio independiente a tu instalación local, con el objetivo de aislar los recursos y librerías, con este concepto podemos tener distintos entornos virtuales con diferentes versiones de Python o de una librería concreta.

Para crear nuestro ambiente, tipiamos el siguiente comando:

```
1 conda create -n "Nombre de nuestro entorno" python=3
```



# Instalación de Qiskit

El siguiente paso es activar nuestro ambiente, por lo que debemos tipiar el siguiente comando:

```
1 conda activate "Nombre de nuestro entorno"
```

Posterior a eso, nos deben salir varias instrucciones que lo que están haciendo es activar nuestro entorno, este proceso toma al rededor entre 3 y 5 minutos. Hecho esto, nuestro entorno está listo para instalar Qiskit con la siguiente instrucción:

```
1 pip install qiskit
```

Ahora solo queda esperar a terminar la instalación.



# Instalación de Qiskit

Transcurrido un tiempo de aproximadamente 3 minutos, Qiskit estará instalado en nuestro ambiente de Anaconda, pero lo que nos falta es un soporte visual para los Jupyter Notebooks con el siguiente comando:

```
1 pip install qiskit[visualization]
```

Para usuarios macOS, es recomendado usar este comando:

```
1 pip install 'qiskit[visualization]'
```



# Instalación de Qiskit

Para este curso es necesario tener las siguientes versiones de Qiskit:

```
qiskit 0.28.0
qiskit-aer 0.8.2
qiskit-aqua 0.9.4
qiskit-ibmq-provider 0.15.0
qiskit-ignis 0.6.0
qiskit-nature 0.1.3
qiskit-terra 0.18.0
```

Pudiendo ser comprobado por medio del siguiente comando:

```
1 conda list
```

o con:

```
1 pip list
```



# Instalación de complementos

En este curso vamos a necesitar librerías extras que no vienen por defecto al instalar Qiskit mediante los pasos anteriores. Para ello necesitamos tipiar los siguientes comandos:

```
1 pip install qiskit-finance
2 pip install qiskit-optimization
3 pip install qiskit-nature
4 pip install qiskit-machine-learning
```

Recordando tipiar uno por uno.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización



# Librerías en Python

Ahora que hemos instalado el paquete Qiskit en nuestro sistema, podemos empezar a escribir un pequeño código para empezar a comprender las configuraciones básicas de los sistemas de simulación de una computadora cuántica.

Pero antes de comenzar tratar código, vamos a darle una pequeña revisión a lo que es Qiskit, su uso y básicamente ¿qué es Qiskit? [kiss-kit] es un SDK (Software Development Kit) de código abierto para trabajar con computadoras cuánticas a nivel de pulsos, circuitos y módulos de aplicación. Lo que logra acelerar el desarrollo de aplicaciones cuánticas aportando toda una gama de herramientas para muchas aplicaciones distintas pero con la gran ventaja de eliminar por completo el ruido captado por un ordenador cuántico, dejando el resultado puro.



# Librerías en Python

Para lograr dicho cometido, Qiskit cuenta con una lista de simuladores que podremos llamar para realizar pruebas a nuestros circuitos, obteniendo con ello diferentes métodos para trabajar, resultados y comportamientos que son parte del funcionamiento de cada simulador. Como ejemplo serían Aer, Aqua, Terra, entre otros que serán abarcados a lo largo de este curso.

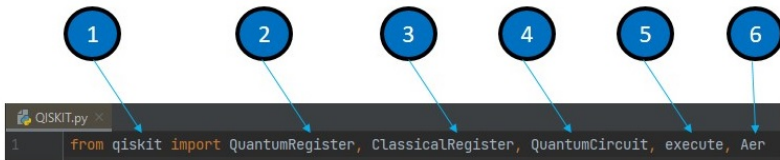
Para esta introducción estaremos manejando los simuladores de Aer mediante la IDE de PyCharm, la cual es la recomendada para trabajar esta parte del curso aunque si se desea usar otra IDE es perfectamente válido.





# Librerías en Python

Para empezar a trabajar los circuitos cuánticos con Qiskit, deberemos llamar a las librerías con las que realizaremos nuestro primer circuito cuántico, para ello debemos definir lo siguiente en nuestra IDE.



# Elementos en Qiskit

- **1.- Modulo Qiskit:** Es el modulo principal donde se alojan todas las herramientas con las que podemos disponer para realizar nuestras simulaciones.
- **2.- Librería de registros cuánticos:** Es la libreria con la que podemos agregar los cúbits que necesitemos para nuestro trabajo, a partir de aqui podremos cambiarlos de signo, rotarlos, entre muchas otras acciones hasta cambiar las magnitudes de su estado.
- **3.- Librería de registros clásicos:** Para poder revisar el comportamiento de nuestros qbits necesitaremos una forma de medir su estado, la mejor manera es utilizando bits clásicos (0 o 1) y con esta librería podremos agregar los circuitos cuánticos.



# Elementos en Qiskit

- **4.- Librería de circuitos cuánticos:** Es la librería con la que podemos agregar los circuitos cuánticos que necesitemos en nuestro trabajo a partir de los cúbits o bits clásicos registrados con anterioridad.
- **5.- Librería ejecuciones:** Con esta librería podremos realizar los trabajos de simulación seleccionando nuestros 'disparos' o 'shots' , el simulador para conseguir los resultados, entre otras cosas más.
- **6.- Librería del simulador:** Básicamente, son las características con las que ejecutaremos así como otras cosas con las que podemos dotar a nuestro circuito como un tipo de ruido que tenga nuestro sistema.



# Elementos en Qiskit

Cabe resaltar, que el estudiante siempre revise los elementos con los que estará trabajando en el circuito que desee diseñar, mediante el uso de la documentación oficial de Qiskit (<https://qiskit.org/>). Ahora, ya hemos comenzado a utilizar las librerías con las que crearemos nuestro circuito, ahora utilizaremos estas librerías para utilizar sus elementos.



# Compuertas cuánticas

La herramienta de Qiskit nos abre la puerta a las compuertas de Pauli, deben su nombre a Wolfgang Ernst Pauli, son matrices usadas en física cuántica en el contexto del momento angular intrínseco o espín. Matemáticamente, las matrices de Pauli constituyen una base vectorial del álgebra de Lie del grupo especial unitario  $SU(2)$ , actuando sobre la representación de dimensión 2.



# Compuertas cuánticas

Puesto que se basan en el momento del espín, la representación típica de estas son las siguientes:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1)$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (2)$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (3)$$



# Compuertas cuánticas

Analogamente a esto, podemos conocer la posición del espín:

$$\{|\uparrow\rangle = (1, 0); |\downarrow\rangle = (0, 1)\}$$

La alteración del estado del espín se realiza mediante las compuertas cuánticas que realizan operaciones como rotaciones, inversiones, filtros, entre otras operaciones más mediante el lenguaje de programación de Python.



# Resultados y visualización

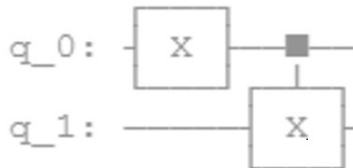
Qiskit nos ofrece una gama de simuladores y formas de visualizar de representar nuestros circuitos cuánticos. La primera opción nos ofrece varias formas de obtener resultados de maneras diferentes, ya sea por medio de shots o disparos de los resultados, o por la posición final de los vectores de los espines.





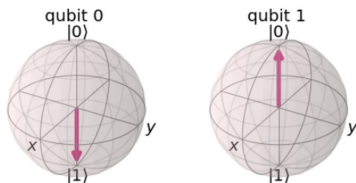
# Resultados y visualización

Además de esto, nos ofrece formas de ver nuestros circuitos desde una forma muy sencilla hasta formas más complejas y visualmente atractivas. Ejemplo:



# Resultados y visualización

Incluso podemos utilizar una forma de representar la posición final de los cúbits mediante el uso de la esfera de Bloch en una representación tridimensional.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización



# Conda

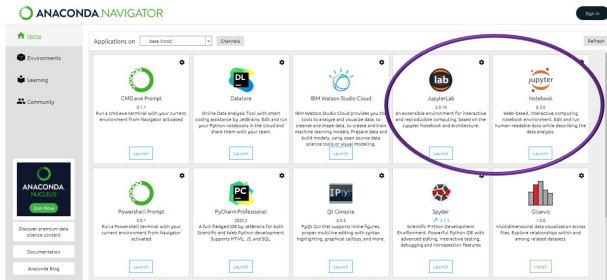
La ventaja que tenemos al trabajar con Qiskit es poder maniobrarlo ya sea en la nube o de manera local, esto nos da la posibilidad de trabajar incluso sin conexión, por lo que pedimos a los nuevos usuarios pasarse por este link para instalar Conda en su última versión.

<https://conda.io/projects/conda/en/latest/user-guide/install/index.html>



# Jupyter Notebook

Para ejecutar nuestros proyectos tenemos variadas opciones pero nuestra principal herramienta serán los notebooks (de Jupyter Notebooks) seleccionando ya sea en la imagen que ven en pantalla.



# Jupyter Notebook

Esto nos permite crear y revisar notebooks de Python en Jupyter compilando por medio de internet, pero también gracias a Conda podemos acceder de manera local para crear nuestros programas sin necesidad de entrar a internet como lo son las IDEs de PyCharm y Spyder. Tan solo requiriendo hacer clic como en el ejemplo anterior.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Anaconda y Qiskit

Instalación en Windows

Instalación en macOS

Instalación en Linux

## ③ Introducción a Qiskit

Librerías en Python

Compuertas cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# IBM Quantum Experience

Incluso con todo lo anterior, si necesitamos más herramientas para trabajar en computación cuántica o requerimos aplicaciones donde la computación cuántica es una herramienta fundamental, tenemos en la nube diferentes opciones para trabajar de manera gratuita con la computación cuántica. Nuevamente, son simuladores de alta precisión debido al ruido que estas máquinas llegan a capturar.

Ejemplo de esto es la IBM Quantum Experience donde podremos trabajar en tiempo real con una de las múltiples computadoras cuánticas que nos ofrece la compañía. Aprendamos un poco como seleccionar un buen modelo para empezar a trabajar y conocer el ambiente.

Para acceder a la herramienta, hay que dirigirnos a esta página:

<https://quantum-computing.ibm.com/>





# IBM Quantum Experience

Entrando al link anterior, nos encontraremos esta página de bienvenida donde podemos interactuar par conocer un poco más sobre la forma de trabajar de la mano de IBM. Para registrarnos, debemos hacer click en el botón azul que dice IBMid.

Real quantum computers.  
Right at your fingertips.

IBM offers cloud access to the most advanced quantum computers available.  
Learn, develop, and run programs with our quantum applications and systems.

Learn more about IBM Quantum Experience

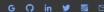


#### Graphically build quantum circuits

Start building quantum circuits right away with IBM Quantum Composer. No sign in required.

[Explore Quantum Composer](#)

Sign in to IBM Quantum

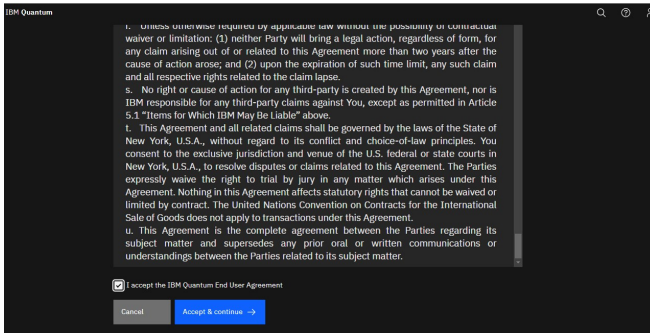


New to IBM Quantum?  
[Create an IBMid account.](#)



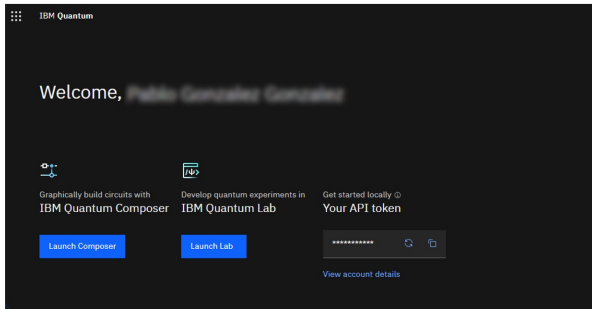
# IBM Quantum Experience

Una vez entrando, nos pedirá nuestros datos personales para poder pasar a aceptar un acuerdo de uso con la herramienta virtual de IBM, asegurarse de leer completamente el acuerdo de uso. Una vez hecho esto, podemos dar en el botón azul y seguir con la aplicación.



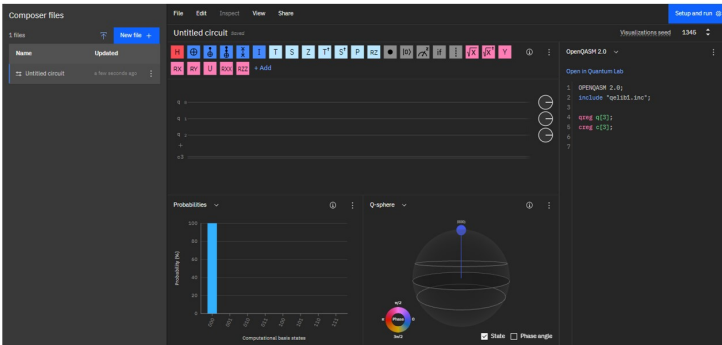
# IBM Quantum Experience

Avanzaremos hasta lo que es nuestra página de inicio de sesión de usuario, en esta se encuentran diferentes opciones como el laboratorio de IBM para escribir código, el simulador de una computadora cuántica y varias opciones más, lo que hoy nos llama es entrar a la parte del Composer, el cual es el botón azul que dice "Launch Composer".



# IBM Quantum Experience

En esta parte podemos disfrutar de una herramienta muy completa para desarrollar circuitos cuánticos, conozcamos un poco más nuestro ambiente.



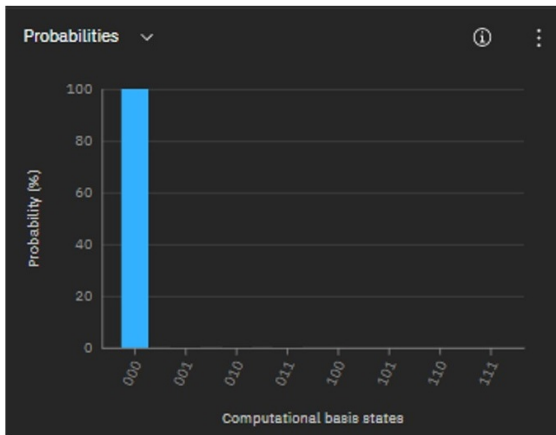
# IBM Quantum Experience

Primero que nada, tenemos nuestro diseñador de circuitos cuánticos, junto a todas las compuertas cuánticas que podremos llegar necesitar, desde las más básicas hasta las más complicadas.



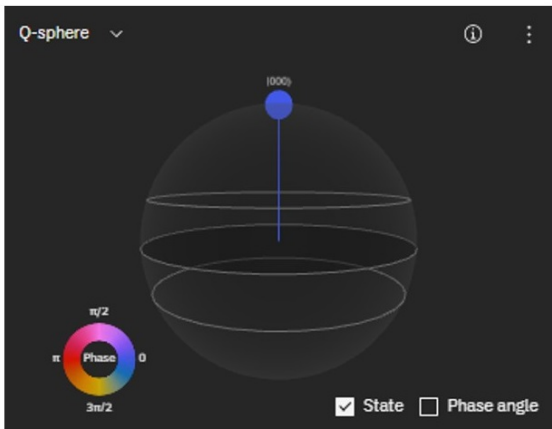
# IBM Quantum Experience

Segundo, cuando terminemos de diseñar nuestro circuito y queramos ver los resultados en forma gráfica y de porcentajes de que ocurran, tenemos esta útil grafica que nos facilita el trabajo.



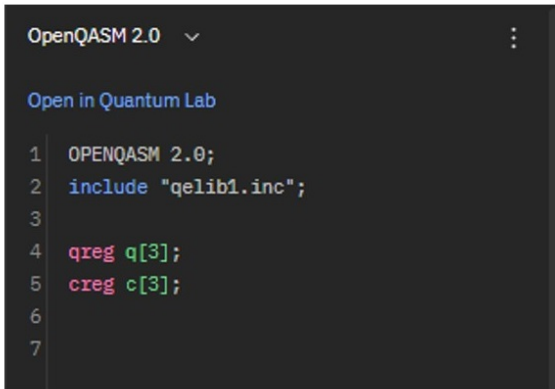
# IBM Quantum Experience

Tercero, nuestra esfera de Bloch móvil donde podemos ver la dirección de nuestros vectores en tiempo real, logrando intercalar entre estado y ángulo de fase.



# IBM Quantum Experience

Cuarto y último, si preferimos probar la entrada por código tenemos una sencilla interfaz para el trabajo, con disponibilidad para usar el lenguaje OpenQASM 2.0 o una versión de Python con Qiskit.

A screenshot of the IBM Quantum Experience code editor interface. At the top left, it says "OpenQASM 2.0" with a dropdown arrow. Below that is a blue link "Open in Quantum Lab". The code editor shows a list of line numbers from 1 to 7 on the left. The code is as follows:

```
1 OPENQASM 2.0;  
2 include "qelib1.inc";  
3  
4 qreg q[3];  
5 creg c[3];  
6  
7
```

