

# Curso de Qiskit

## Instalación e Introducción

Pablo Gonzalez

Qiskit Summer Jam Mexico, Agosto 2021



# Tabla de Contenido

## ① Introducción

## ② Instalación de Qiskit

## 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 Qiskit

## ③ Introducción a Qiskit

Librerías en Python

Compuertas Cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# Introducción

Tradicionalmente, nuestros ordenadores 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 la situación de tener que seguir usando un sistema que no permita la dualidad de situaciones ha llevado a encontrar un cierto límite con respecto a tareas más complejas de realizar que navegar por internet.

Los computadores cuánticos teóricamente tienen una ventaja significativa contra los procesos que se pueden llevar a cabo en un ordenador normal, un ejemplo es la ciberseguridad y el desarrollo de cálculos matemáticos que le llevarían años en términos de tiempo a los computadores actuales.



# Introducción a la computación cuántica

Aunque si bien, esto se escucha como ciencia ficción, los computadores actuales tienen el inconveniente de trabajarse por medio de la física cuántica, dejando en condiciones muy especiales para que estos puedan ser funcionales, ya que incluso la presencia de un observador, las variedades del ruido (ambiental, electromagnético, eléctrico, entre otros) pueden afectar a las operaciones que estas realicen.

Por tal motivo, su programación actualmente es un poco más avanzada que la realizada para los ordenadores convencionales, pero la ventaja que tenemos hoy contra hace 50 años en el pasado es que los ordenadores convencionales pueden simular de manera eficiente la respuesta real que se podría conseguir mediante un ordenador cuántico.



# Introducción a la computación cuántica

Es por ello que en este documento se estarán presentado, la forma de programar mediante herramientas en ordenadores convencionales gracias al uso de lenguajes de programación tales como Python donde es posible simular de manera precisa la respuesta de un ordenador cuántico mediante simuladores internos.

Así como también el uso de tecnologías remotas o en la nube como las son herramientas de IBM donde puede utilizarse uno de los ordenadores cuánticos reales para realizar nuestros trabajos de computo.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Qiskit

## ③ Introducción a Qiskit

Librerías en Python

Compuertas Cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# Instalación de Qiskit en Python

Para realizar una instalación directa de lo que es Qiskit y sus simuladores se deberán seguir los siguientes pasos:

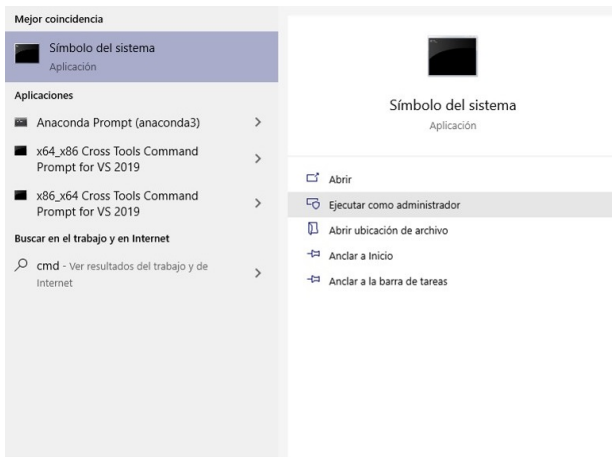
- **Abrimos el Command Prompt(CMD):** Para utilizar esta consola basta con ir a nuestro buscador (variando entre versiones de Windows) y poner ahí la palabra `cmd`.





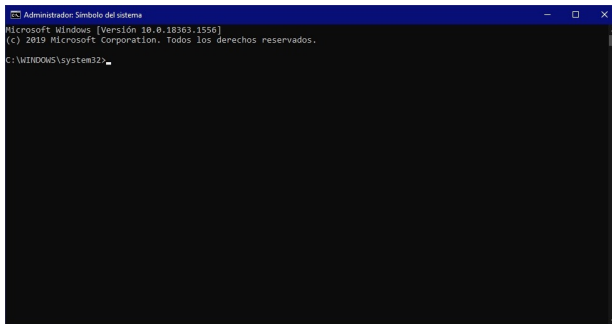
# Instalación de Qiskit en Python

Una vez hecho esto, daremos a buscar (o enter) y nos debería aparecer la siguiente ventana del buscador.



# Instalación de Qiskit en Python

Le daremos click a la opción resaltada 'Ejecutar como administrador' solo por seguridad, pero puede funcionar incluso sin los permisos de administrador. Cuando termine de cargar, nos debe aparecer la siguiente ventana.



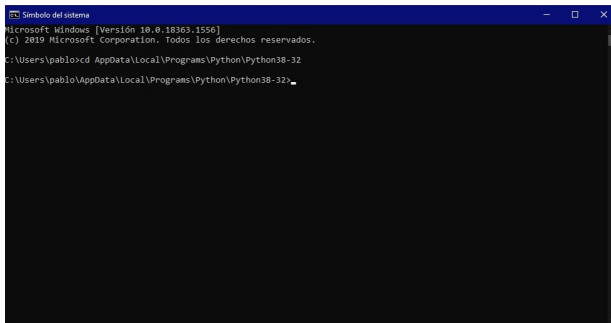
```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>
```



# Actualizar nuestra versión de pip

Ya una vez en la ventana del CMD lo que necesitamos es acceder a Python desde la carpeta de aplicaciones locales de nuestro ordenador, por ello es necesario escribir la siguiente instrucción:



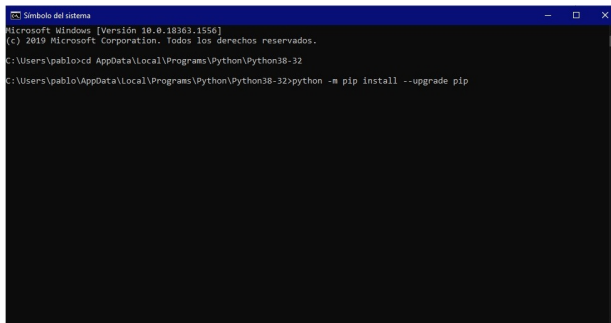
```
Simbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\pablo>cd AppData\Local\Programs\Python\Python38-32
C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>
```



# Actualizar nuestra versión de pip

Al hacer esto, hemos llamado a nuestra versión de Python desde donde estaremos actualizando la versión de pip (21.1.2) mediante el comando:



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\pablo>cd AppData\Local\Programs\Python\Python38-32

C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>python -m pip install --upgrade pip
```



# Actualizar nuestra versión de pip

Cuando demos enter sobre este comando nos debe aparecer la siguiente ventana, donde se nos actualizara la versión de pip y al final su correspondiente mensaje donde la actualización fue exitosa.

```
Simbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\pablo>cd AppData\Local\Programs\Python\Python38-32
C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>python -m pip install --upgrade pip
Collecting pip
  Downloading pip-21.1.2-py3-none-any.whl (1.5 MB)
    | 1.5 MB 1.3 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.2
    Uninstalling pip-20.2:
      Successfully uninstalled pip-20.2
Successfully installed pip-21.1.2

C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>
```



# Actualizar nuestra versión de pip

Añadido a esto para conocer la versión actual de nuestro pip basta con ejecutar el siguiente comando.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\pablo>cd AppData\Local\Programs\Python\Python38-32

C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>python -m pip install --upgrade pip
Collecting pip
  Downloading pip-21.1.2-py3-none-any.whl (1.5 MB)
    | 1.5 MB 1.3 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.2
    Uninstalling pip-20.2:
      Successfully uninstalled pip-20.2
Successfully installed pip-21.1.2

C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>pip --version
pip 21.1.2 from c:\users\pablo\appdata\local\programs\python\python38-32\lib\site-packages\pip (python 3.8)

C:\Users\pablo\AppData\Local\Programs\Python\Python38-32>
```



# Instalar Qiskit de manera local

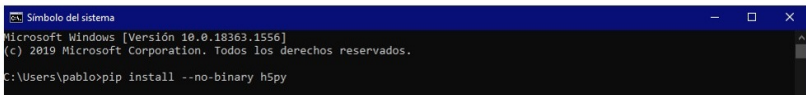
Una vez tengas la versión más actualizada de pip, podrás ingresar el siguiente comando para instalar todas las librerías con las que se trabajaran de qiskit.

```
Simbolo del sistema - pip install qiskit
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\pablo>pip install qiskit
```



# Instalar complementos de Qiskit

Puede ocurrir que durante la instalación de qiskit nos den algunos errores, para ello sugerimos seguir los pasos a continuación para darle solución a estos problemas.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\pablo>pip install --no-binary h5py
```

Esto ya que algunos complementos necesarios para la instalación de qiskit no vienen en nuestros ordenadores al instalar Python y necesitaremos instalarlos de manera manual para resolver estos conflictos.





# Prueba de instalación exitosa

Una vez haya terminado la instalación, es conveniente saber que estamos en la versión más actualizada de qiskit, para comprobar este punto te sugerimos introducir el siguiente comando.

```
1 pip list
```

Para actualizar nuestra versión de Qiskit instalada a la más reciente, sugerimos introducir el siguiente comando y revisar que las versiones más actuales (26 de Junio del 2021) sean similares a la imagen más adelante.



# Prueba de instalación exitosa

```
1      pip install qiskit[visualization] -U
2      pip install qiskit-nature -U
3      pip install --upgrade qiskit==0.27.0
```

```
qiskit                0.27.0
qiskit-aer             0.8.2
qiskit-aqua            0.9.2
qiskit-ibmq-provider   0.14.0
qiskit-ignis           0.6.0
qiskit-nature          0.1.3
qiskit-terra           0.17.4
```



# Tabla de Contenido

## ① Introducción

## ② Instalación de Qiskit

## ③ Introducción a Qiskit

Librerías en Python

Compuertas Cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube



# 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? 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

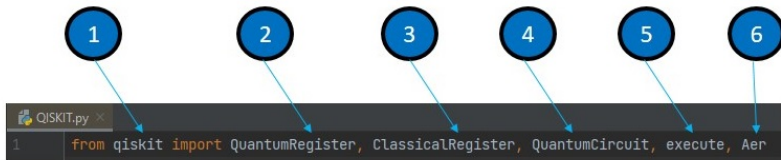
Para lograr dicho cometido, qiskit cuenta con una lista de simuladores que podremos llamar para realizar pruebas a nuestros circuitos, obteniendo con ello diferentes métodos para trabajar, resultados y comportamientos que son parte del funcionamiento de cada simulador. Como ejemplo serían Aer, Aqua, 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 qbits 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 qbits o bits clásicos registrados con anterioridad.
- **5.- Librería ejecuciones:** Con esta librería podremos realizar los trabajos de simulación seleccionando nuestros "disparos.<sup>o</sup> 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 Cuanticas

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 Cuanticas

Analogamente a esto, podemos conocer la posición del espin:

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

La alteración del estado del espin se realiza mediante las compuertas cuanticas que realizan operaciones como rotaciones, inversiones, filtros, entre otras operaciones mas mediante el leguanje de programacion de Python.



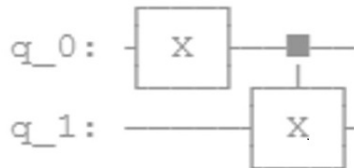
# Resultados y visualización

Qiskit nos ofrece una gama de simuladores y formas de visualizar de representar nuestros circuitos cuanticos. La primera opcion 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 espin.



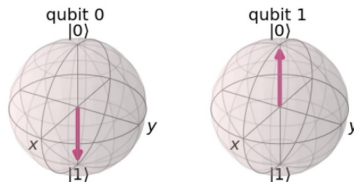
# Resultados y visualización

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



# Resultados y visualización

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



# Tabla de Contenido

## ① Introducción

## ② Instalación de Qiskit

## ③ Introducción a Qiskit

Librerías en Python

Compuertas Cuánticas

Resultados y visualización

## ④ Herramientas locales

## ⑤ Herramientas en la nube





# Conda

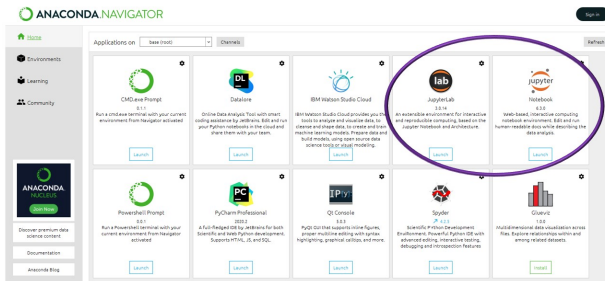
La ventaja que tenemos al trabajar con Qiskit es poder maniobrarlo ya sea en la nube o de manera local, esto nos da la posibilidad de trabajar incluso sin conexión, por lo que pedimos a los nuevos usuarios pasarse por este link para instalar conda en su última versión.

<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 a través 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 click como en el ejemplo anterior.



# Tabla de Contenido

## ① Introducción

## ② Instalación de Qiskit

## 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.



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

IBMid 

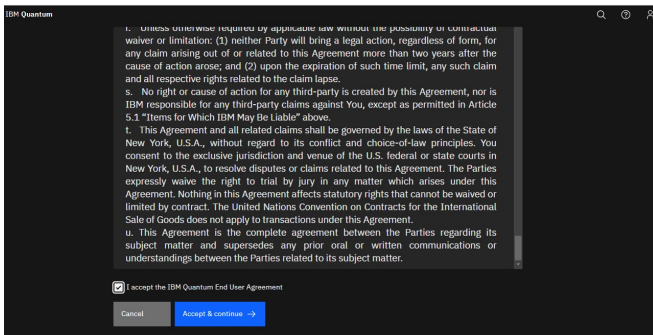
[G](#) [in](#) [in](#) [in](#) [in](#) [in](#) [in](#)

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



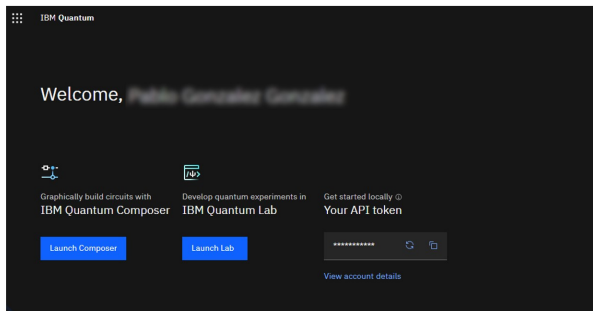
# IBM Quantum Experience

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

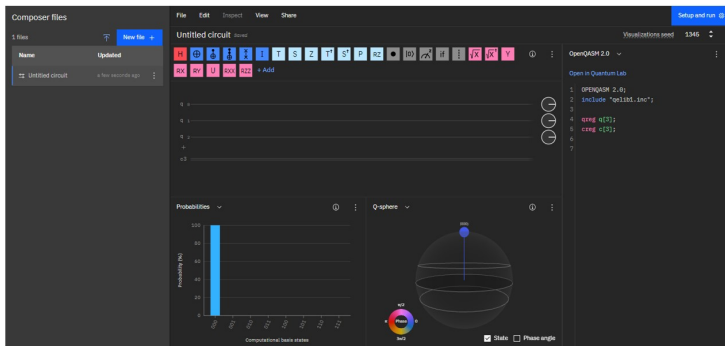
Avanzaremos hasta lo que es nuestra página de inicio de sesión de usuario, en esta se encuentran diferentes opciones como el laboratorio de IBM para escribir código, el simulador de una computadora cuántica y varias opciones más, lo que hoy nos llama es entrar a la parte del Composer, el cual es el botón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



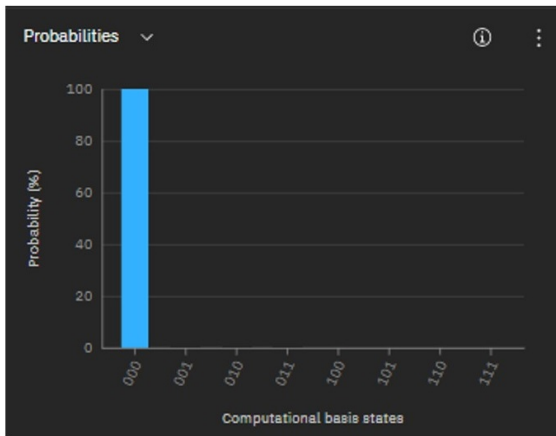
# IBM Quantum Experience

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.



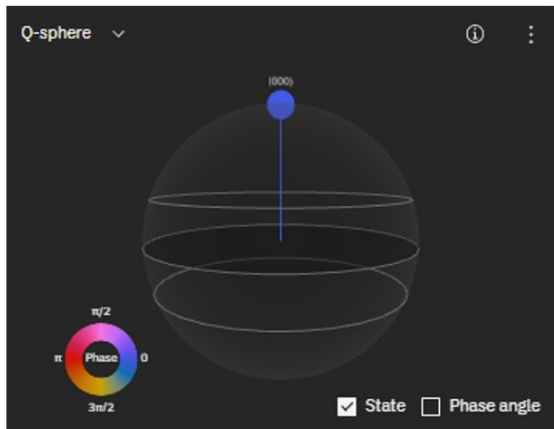
# IBM Quantum Experience

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.



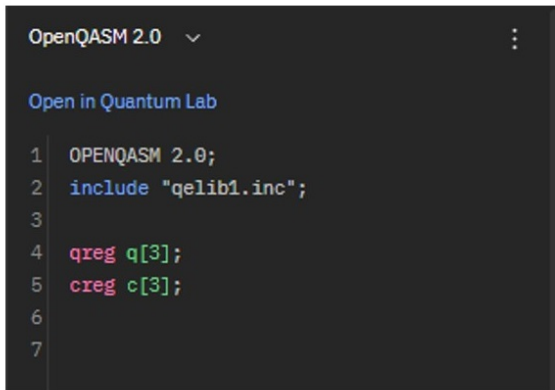
# IBM Quantum Experience

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.



The screenshot shows a code editor window titled "OpenQASM 2.0" with a dropdown arrow. Below the title is a link "Open in Quantum Lab". The editor contains the following code:

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

