

# Qiskit | Fall Fest

Venezuela (Universidad Simón Bolívar) - Reto



***Clase 2:*** *Computación cuántica al nivel de pulsos. Introducción a Qiskit Pulse.*

**M.Morgado**

Université de Strasbourg - Centre Européen de Sciences Quantiques (CESQ)

Apoyado por



# Contenido

- **Compuertas**
  - **Formalismo de circuitos**
  - **Revisión del hardware**
  - **Sistemas cuánticos**
  - **Pulsos / formas de ondas**
  - **Introducción a Qiskit Pulse**
  - **Ejemplo de compuertas**
-

# Referencias



## Qiskit textbook

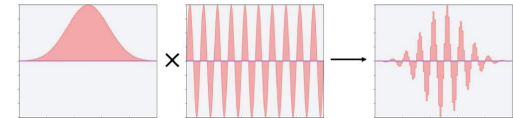
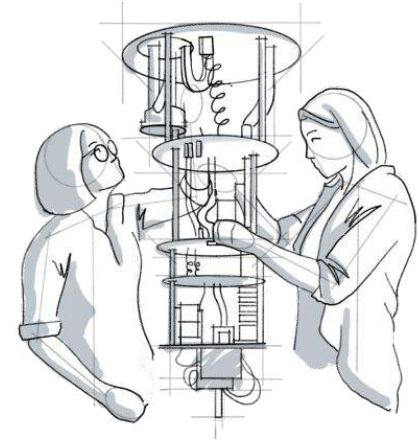
[qiskit.org/textbook/ch-quantum-hardware/index-pulses](https://qiskit.org/textbook/ch-quantum-hardware/index-pulses)

## The Role of Timing and Pulses by Thomas Alexander

<https://youtu.be/cEhaVe19TKs>

## Paper

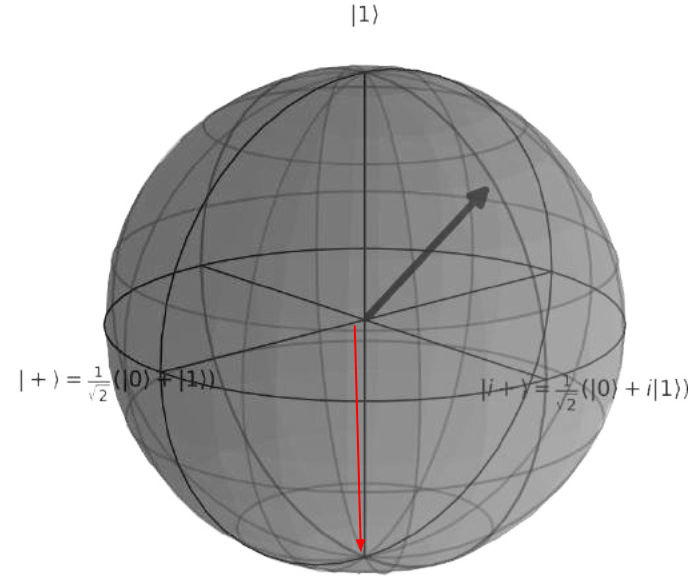
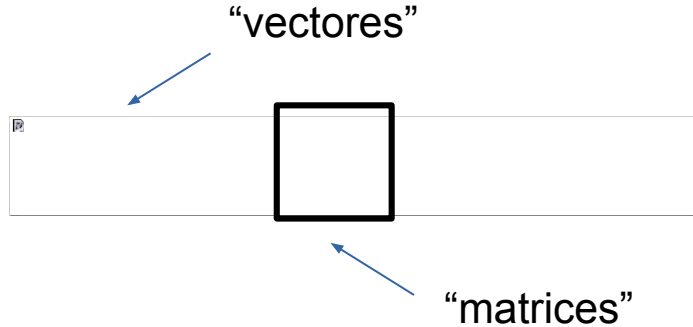
**Alexander, Thomas, et al.** "Qiskit pulse: Programming quantum computers through the cloud with pulses." Quantum Science and Technology 5.4 (2020): 044006. [arXiv: 2004.06755](https://arxiv.org/abs/2004.06755)



# Compuertas cuánticas

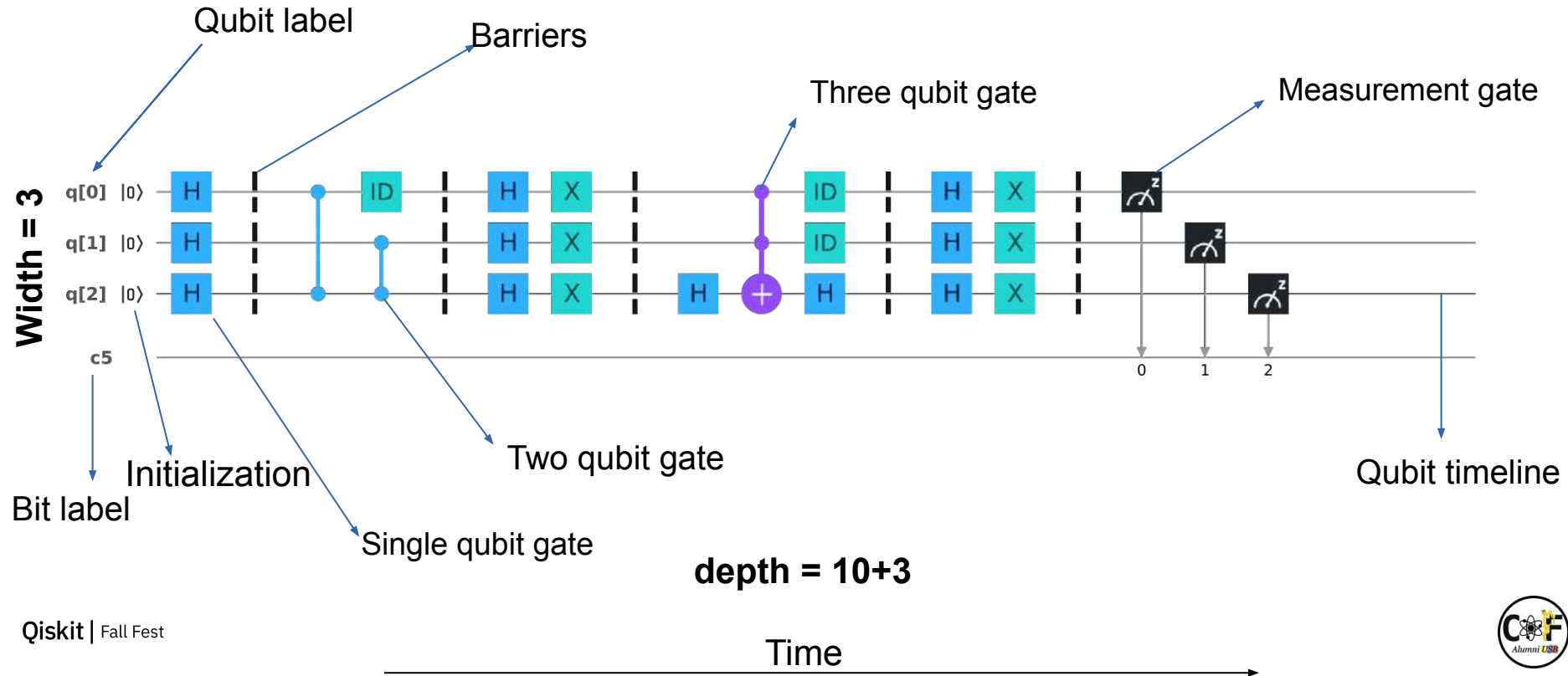


Las **compuertas lógicas cuánticas** son transformaciones unitarias (reversible) que toman un estado inicial del qubit (input) y las evolucionan (transforman) en un estado final (output).



$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle$$
$$\theta, \phi \in \mathbb{R}$$

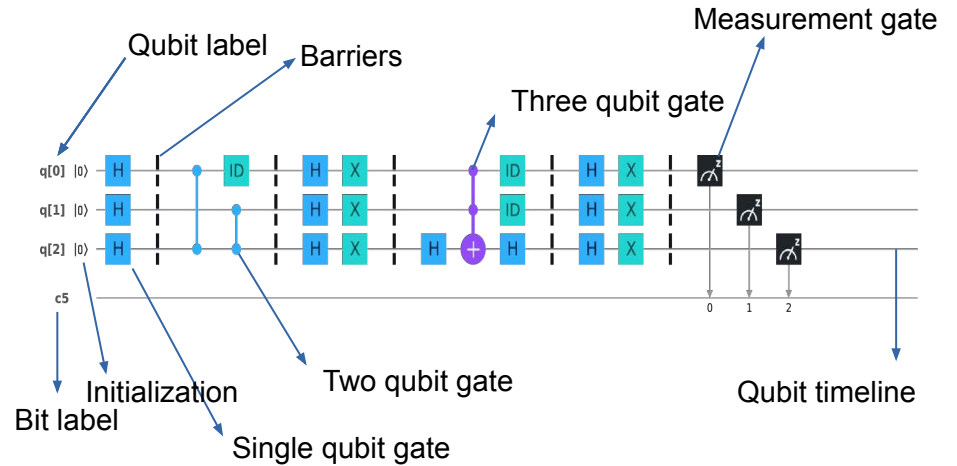
# Formalismo de circuitos



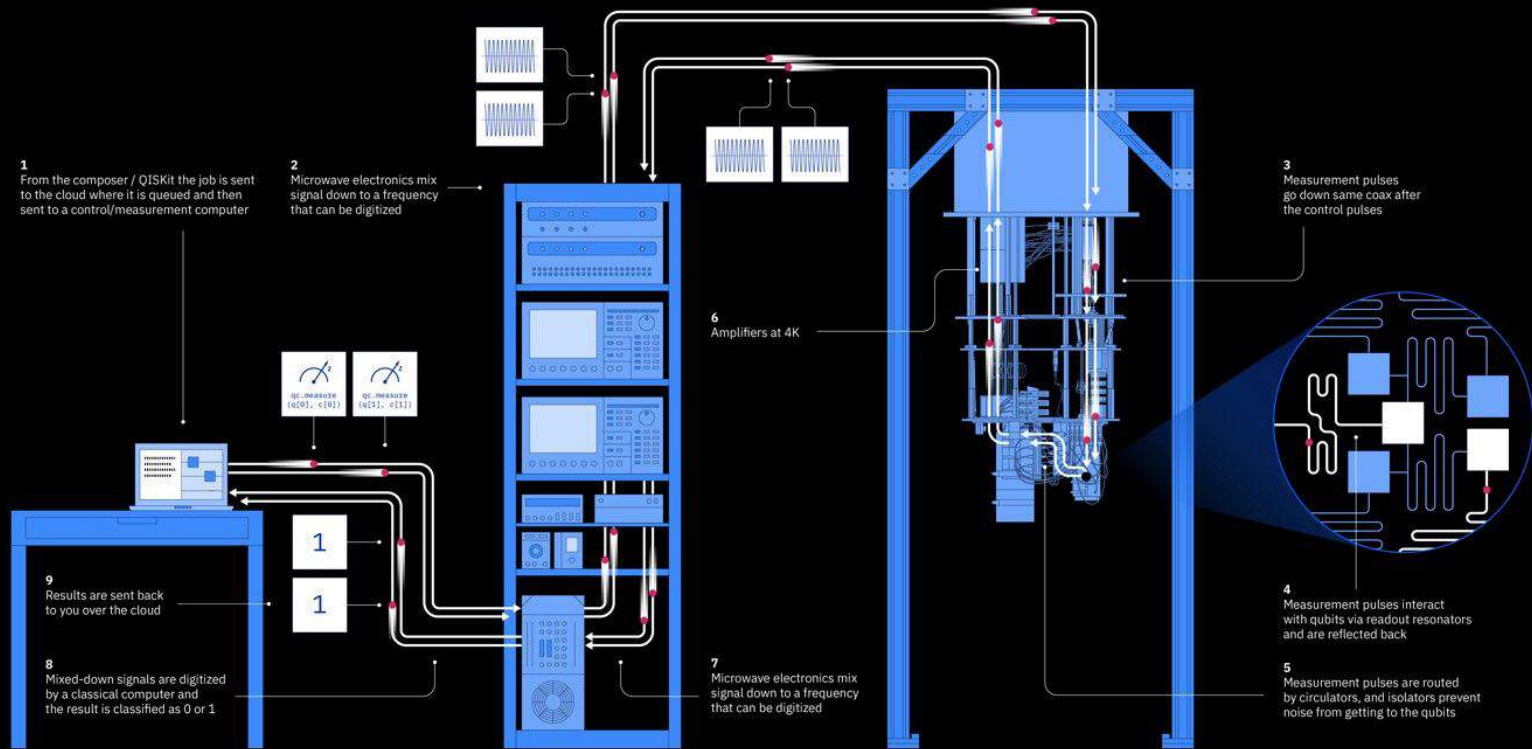
# Formalismo de circuitos



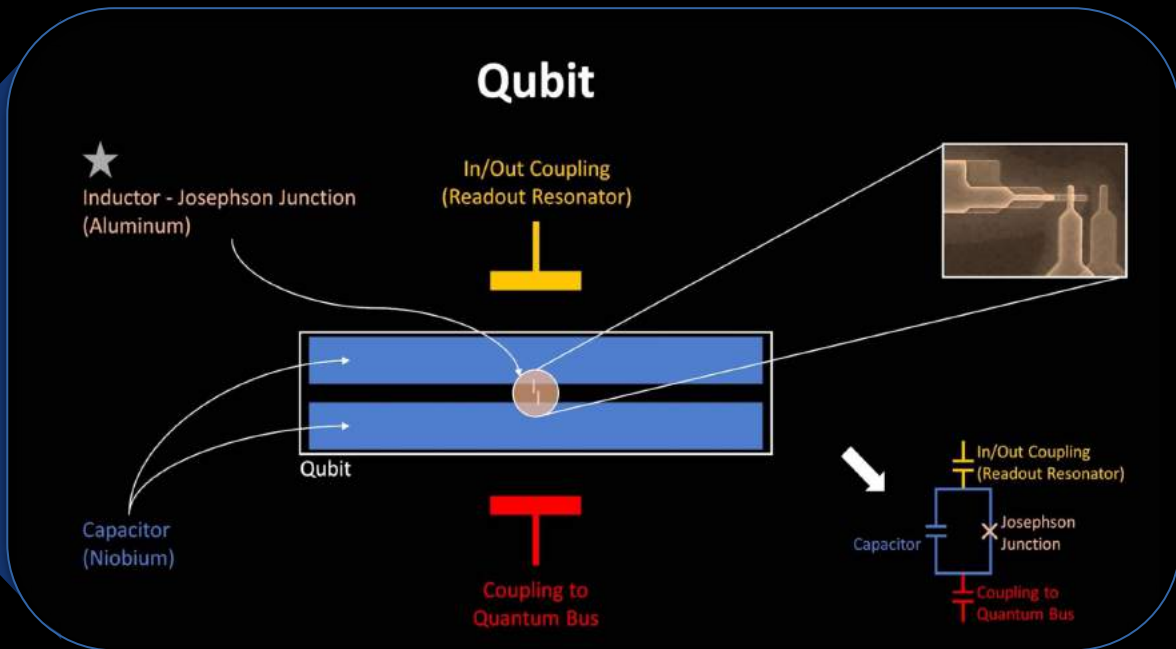
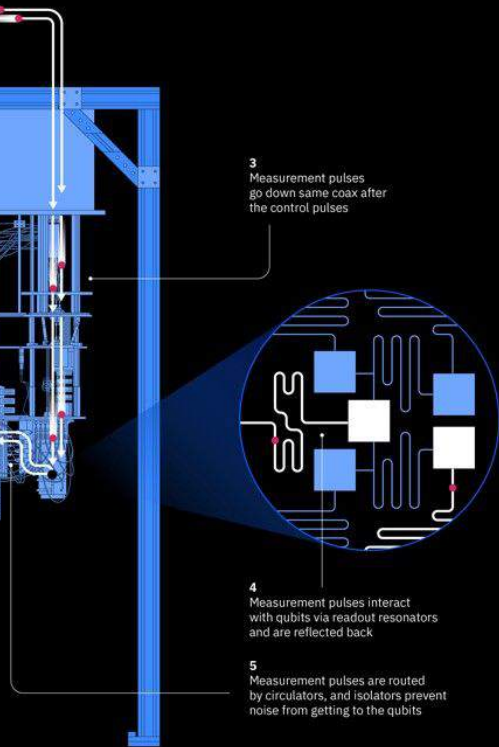
```
example.py
1 import qiskit
2 from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
3 from numpy import pi
4
5 qreg_q = QuantumRegister(3, 'q')
6 creg_c = ClassicalRegister(3, 'c')
7 circuit = QuantumCircuit(qreg_q, creg_c)
8
9 circuit.h(qreg_q[0])
10 circuit.h(qreg_q[1])
11 circuit.h(qreg_q[2])
12
13 circuit.barrier(qreg_q[0])
14 circuit.barrier(qreg_q[1])
15 circuit.barrier(qreg_q[2])
16
17 circuit.ccx(qreg_q[0], qreg_q[2])
18 circuit.h(qreg_q[0])
19 circuit.ccx(qreg_q[1], qreg_q[2])
20
21 circuit.barrier(qreg_q[0])
22 circuit.barrier(qreg_q[1])
23 circuit.barrier(qreg_q[2])
24
25 circuit.h(qreg_q[0])
26 circuit.h(qreg_q[1])
27 circuit.h(qreg_q[2])
28 circuit.x(qreg_q[0])
29 circuit.x(qreg_q[1])
30 circuit.x(qreg_q[2])
31
32 circuit.barrier(qreg_q[0])
33 circuit.barrier(qreg_q[1])
34 circuit.barrier(qreg_q[2])
35
36 circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
37 circuit.h(qreg_q[0])
38 circuit.h(qreg_q[1])
39 circuit.h(qreg_q[2])
40 circuit.x(qreg_q[0])
41 circuit.x(qreg_q[1])
42 circuit.x(qreg_q[2])
43
44 circuit.barrier(qreg_q[0])
45 circuit.barrier(qreg_q[1])
46 circuit.barrier(qreg_q[2])
47
48 circuit.h(qreg_q[0])
49 circuit.h(qreg_q[1])
50 circuit.h(qreg_q[2])
51 circuit.x(qreg_q[0])
52 circuit.x(qreg_q[1])
53 circuit.x(qreg_q[2])
54
55 circuit.barrier(qreg_q[0])
56 circuit.barrier(qreg_q[1])
57 circuit.barrier(qreg_q[2])
58
59 circuit.measure(qreg_q[0], creg_c[0])
60 circuit.measure(qreg_q[1], creg_c[1])
61 circuit.measure(qreg_q[2], creg_c[2])
62
```



# Revisión del hardware



# Revisión del hardware ⚡





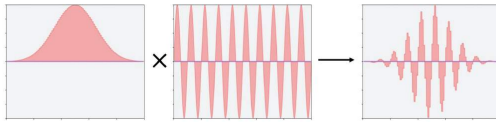
# Revisión del hardware ⚡

El hardware de control está compuesto por 3 principales tipos de dispositivos electrónicos de generación de señales:

*DDS – Direct Digital Synthesizers* (**Activo**): para producir las frecuencias continuas del qubit y los resonadores.

*AWG - Arbitrary Waveform Generators* (**Activo**): que produce las envolventes programables arbitrariamente.

*Mixer* (**Pasivos**): combinan las señales sinusoidales con las envolventes. También existen otros componentes pasivos e.g., amplificadores, divisores etc.



# Sistemas cuánticos



Resolviendo la **ecuación de Schrödinger**, bajo la acción de un Hamiltoniano, podemos describir la evolución temporal del sistema cuántico:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{\mathcal{H}} |\psi(t)\rangle$$

Esto nos lleva a obtener el **operador evolución**:

$$U = e^{i\hat{\mathcal{H}}\Delta t/\hbar}$$

# Sistemas cuánticos

Expandiendo en series de Taylor la matriz exponencial obtenemos la transformación unitaria:

$$e^{\hat{A}} = \sum_{n=0}^{\infty} \frac{\hat{A}^n}{n!}$$

Esta transformación unitaria es lo que conocemos como compuerta lógica cuántica:



# Pulsos / formas de ondas



El Hamiltoniano independiente del tiempo puede entenderse como una suma de dos términos: 1) acoplamientos entre los estados del qubit y 2) las energías de los estados del qubit.

$$\hat{\mathcal{H}} = \sum_i (\hat{\mathcal{H}}_{\Omega_i} + \hat{\mathcal{H}}_{\delta_i})$$

El Hamiltoniano dependiente del tiempo puede entenderse como un Hamiltoniano independiente del tiempo, con una función de modulación:

$$\hat{\mathcal{H}}(t) \sim f(t)\hat{\mathcal{H}}$$

Escrito de otra manera:

$$\hat{\mathcal{H}}(t) = \sum_i (\hat{\mathcal{H}}_{\Omega_i}(t) + \hat{\mathcal{H}}_{\delta_i}(t))$$

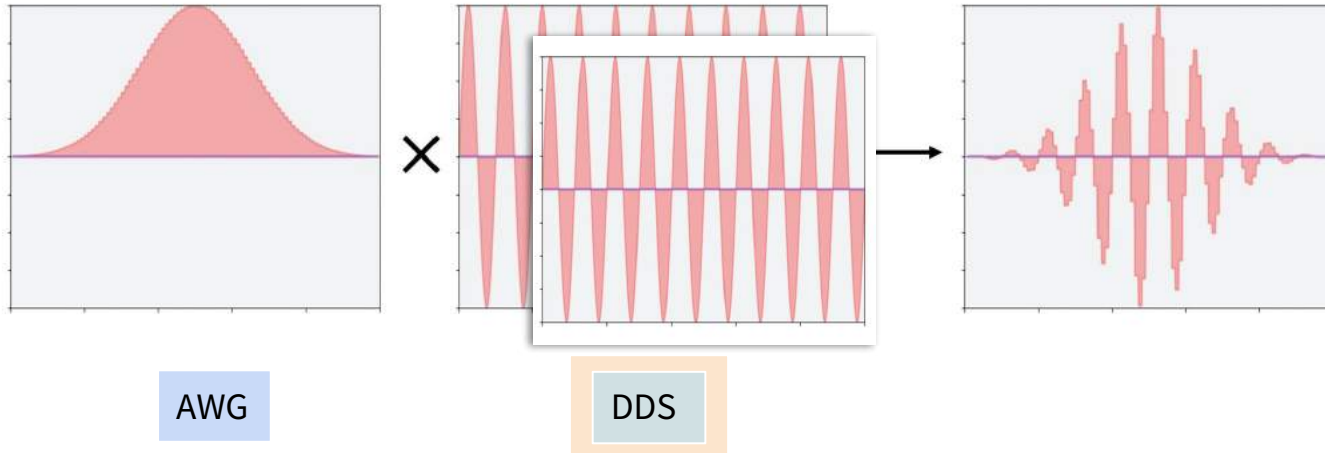
Luego de aplicar la Aproximación de Onda Rotante (RWA, en inglés), el Hamiltoniano queda:

$$\hat{\mathcal{H}}_{RWA} = \left( \frac{\omega_q - \omega_d}{2} \right) \hat{\sigma}_z + \left( \frac{\Omega}{2} \right) A(t) [\cos(\phi) \hat{\sigma}_x + \sin(\phi) \hat{\sigma}_y]$$

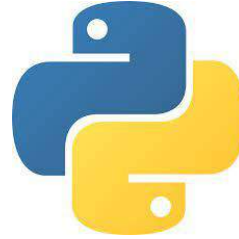
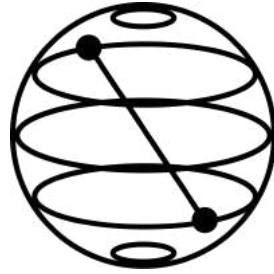
# Pulsos / formas de ondas



$$\hat{\mathcal{H}}_{\text{RWA}} = \left( \frac{\omega_q - \omega_d}{2} \right) \hat{\sigma}_z + \left( \frac{\Omega}{2} \right) A(t) [\cos(\phi) \hat{\sigma}_x + \sin(\phi) \hat{\sigma}_y]$$



# Introducción a Qiskit Pulse



<https://github.com/COFAlumni-USB/qiskit-fall-2022>

# ¿Preguntas?

