To quantum apocalypse

...and beyond...

Ekoparty #20
15 de Noviembre de 2024
Carlos Benitez

God does not play dice. (Albert Einstein)

To quantum apocalypse

...and beyond...

LAST BUT NOT LEAST

Ekoparty #20
15 de Noviembre de 2024
Carlos Benitez

God does not play dice. (Albert Einstein)

Carlos Benitez

- Ing. y Mg. de la UTN FRBA
- Investigador en procesamiento de señales acústicas submarinas.
- Director del primer Laboratorio en Seguridad Informática (Si6) en el ámbito del Estado.
- Implementación del primer SOC del Ministerio de Defensa.
- Asesor técnico de la Subsecretaría de Ciberdefensa.
- Consultor en ciberseguridad.
- Co-fundador de Platinumciber.
- Proyectos de ciberseguridad, como: SOC, Ethical Hacking, Vulnerability Assessment, Análisis forense, Análisis y Gestión de Riesgos, etc.
- Algunas publicaciones en congresos y dos patentes en USA en ciberseguridad.
- Docente de posgrado en ciberseguridad.
- Formador y mentoring de teams.
- Quantum Computing enthusiast.

Índice

Parte 0
- Preparación

Parte 1
- El apocalipsis cuántico
- Evolución tecnológica
- Tipos de computadoras cuánticas
- Proyección

Parte 2
- Conceptos básicos de qiskit
- Pasos para ir de un problema a un circuito cuántico
- Errores
- Ejemplos de problemas en ciberseguridad
- Ejecución de ejemplos e interpretación de resutados

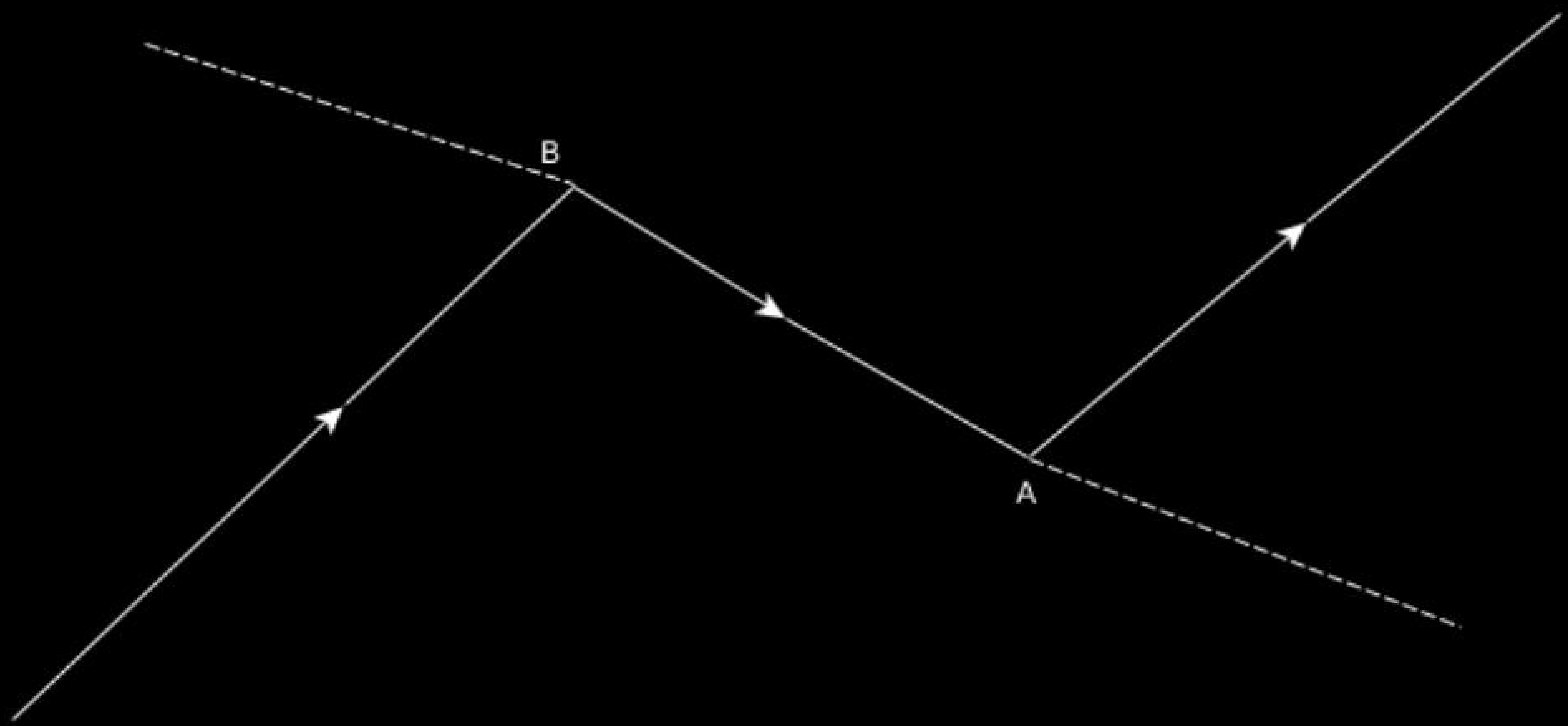
Bonus track
- ¿AI ayudando a Quantum o Quantum ayudando a AI?

Conclusiones

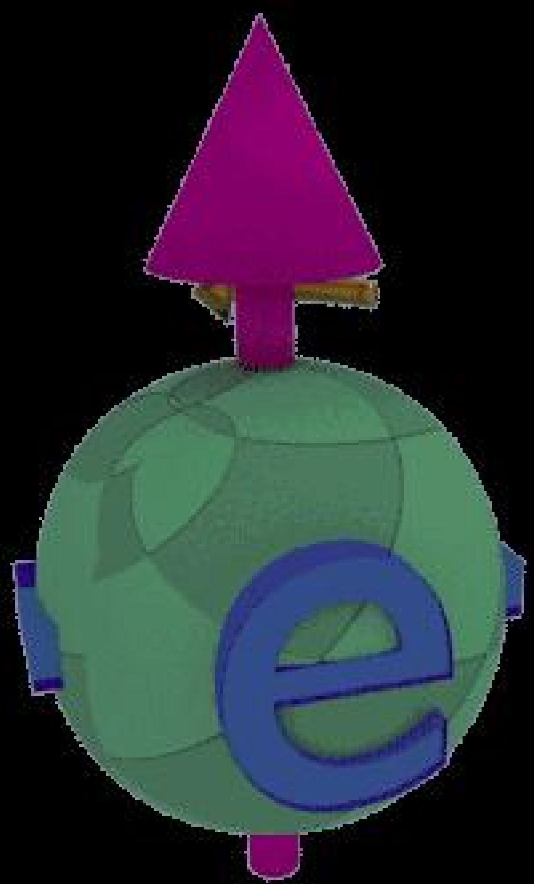
Referencias y bibliografía

Parte 0

koan 1


B
A

koan 2



$$\mu = \frac{e \cdot v \cdot r}{2}$$

- $e = 1.6 \times 10^{-19}\,\mathrm{C}$,
- $r = 1.41 \times 10^{-15}\,\mathrm{m}$.
- $\mu = \mu_B = 9.27 \times 10^{-24}\,\mathrm{A\ m}^2$

$$v = \frac{2\mu_B}{e \cdot r}$$

$$v \approx 8.26 \times 10^8\,\mathrm{m/s}$$

$$c \approx 3 \times 10^8\,\mathrm{m/s}$$

$$v \approx 2.75 \times c\,!!!!!$$

$$\mu = \frac{e \cdot v \cdot r}{2}$$

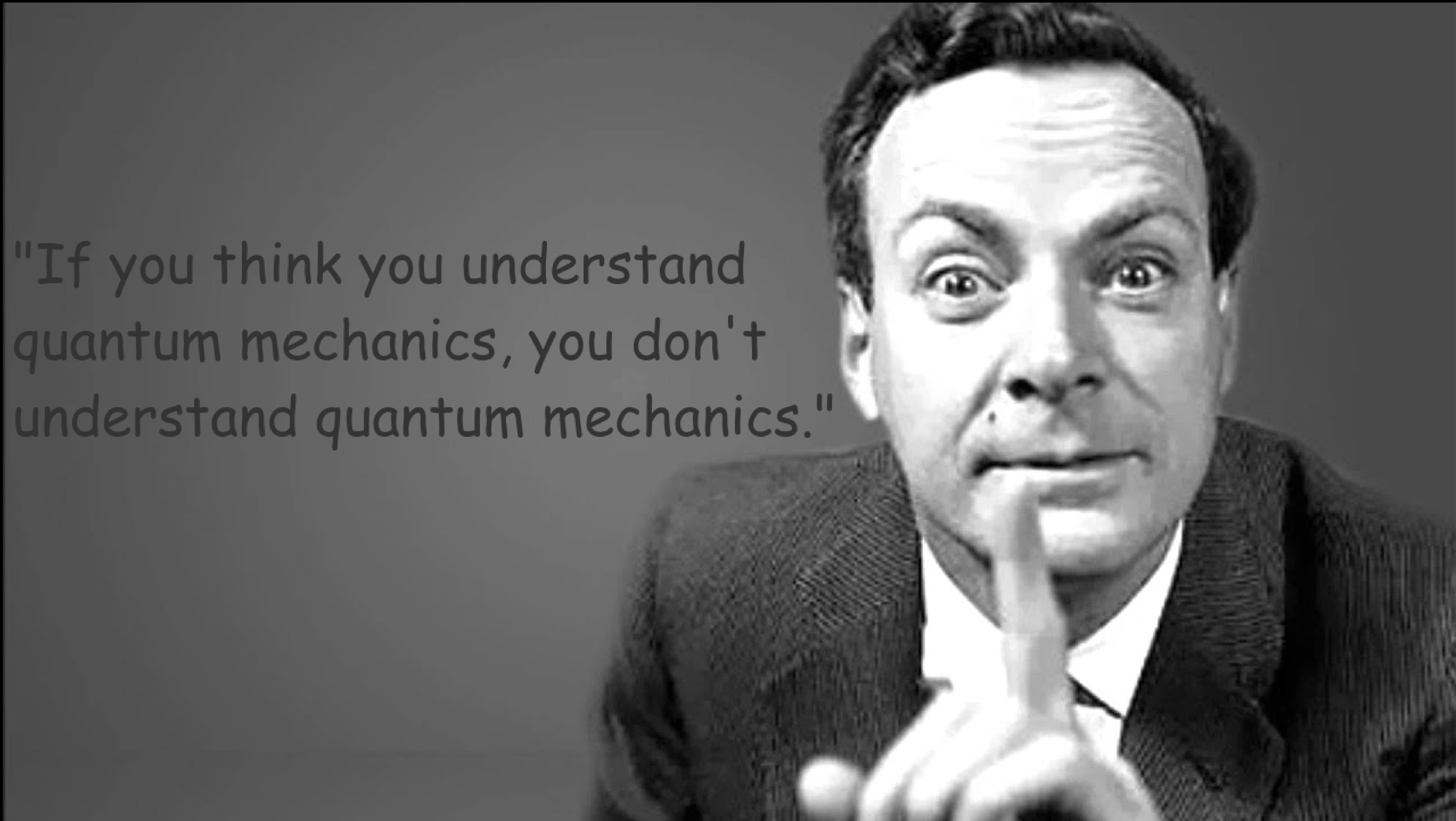
- $e = 1.6 \times 10^{-19}\,\mathrm{C}$,
- $r = 1.41 \times 10^{-15}\,\mathrm{m}$.
- $\mu = \mu_B = 9.27 \times 10^{-24}\,\mathrm{A\ m^2}$

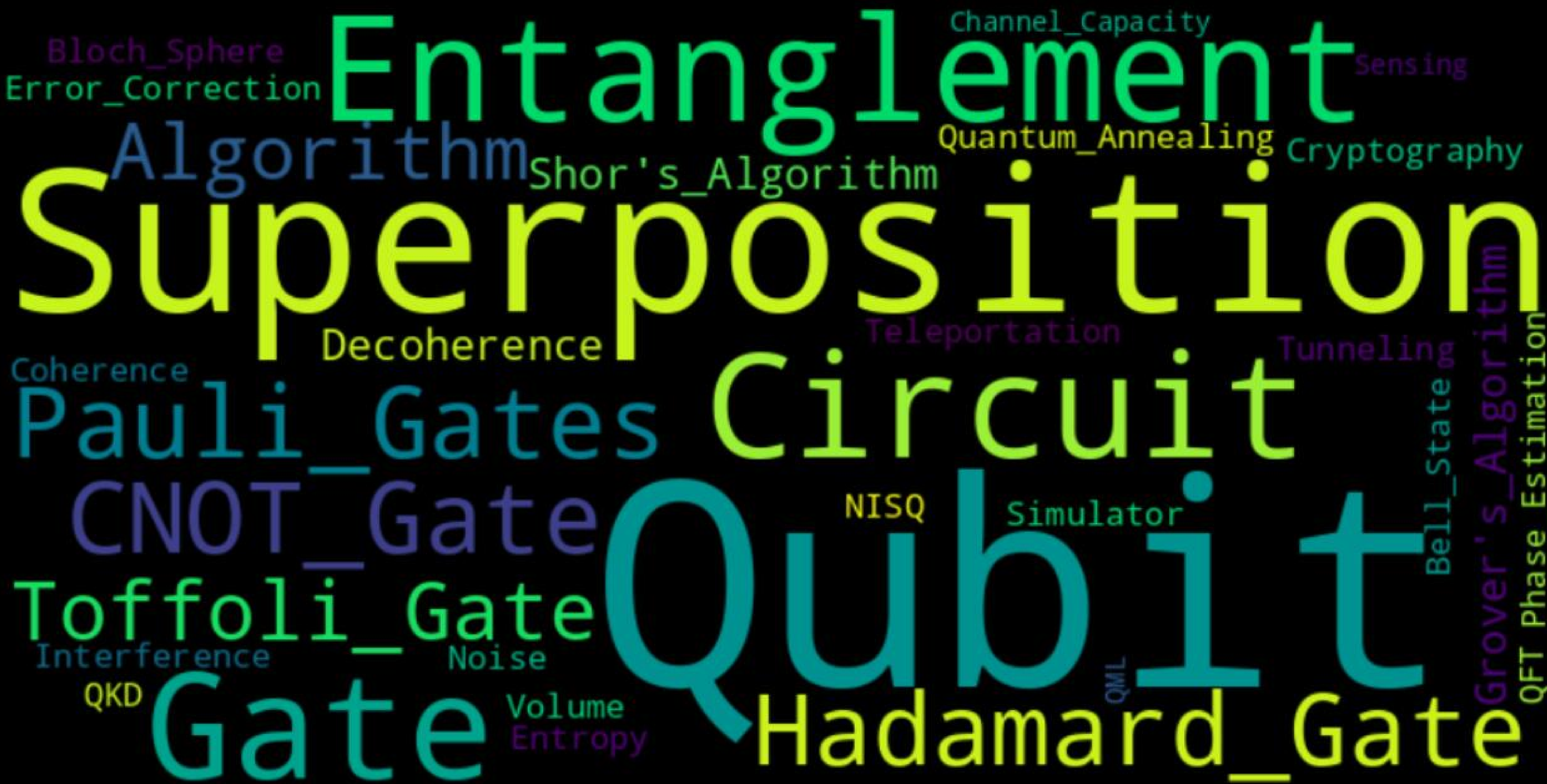
$$v = \frac{2\mu_B}{e \cdot r}$$

$$v \approx 8.26 \times 10^{8}\,\mathrm{m/s}$$

$$c \approx 3 \times 10^{8}\,\mathrm{m/s}$$

$$v \approx 2.75 \times c\,!!!!!$$


"If you think you understand
quantum mechanics, you don't
understand quantum mechanics."

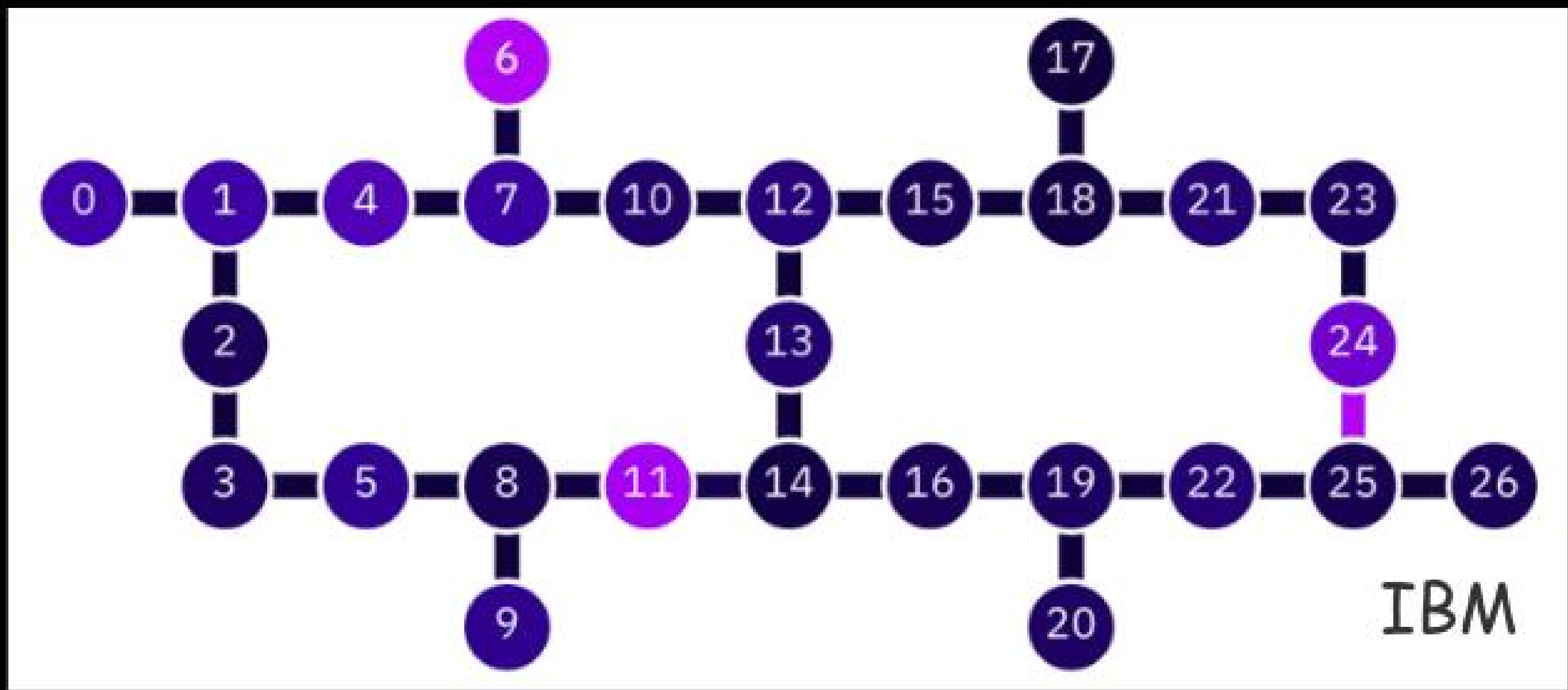


Channel_Capacity
Bloch_Sphere
Entanglement
Sensing
Error_Correction
Algorithm
Quantum_Annealing
Cryptography
Shor's_Algorithm
Superposition
Teleportation
Decoherence
Tunneling
Coherence
Pauli_Gates
Circuit
Grover's_Algorithm
QFT Phase_Estimation
Bell_State
CNOT_Gate
NISQ
Simulator
Qubit
Toffoli_Gate
Interference
Noise
QML
QKD
Gate
Volume
Hadamard_Gate
Entropy

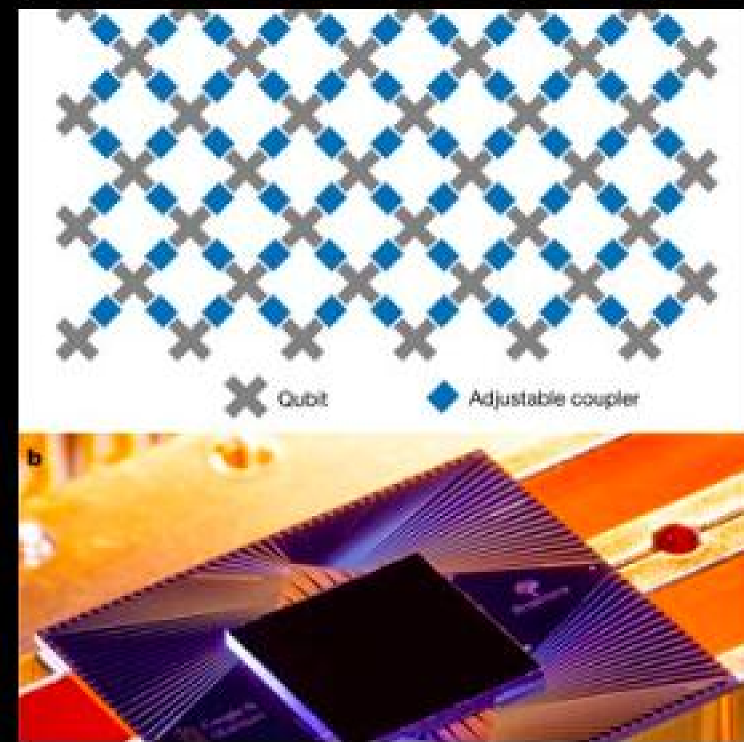
Parte 1

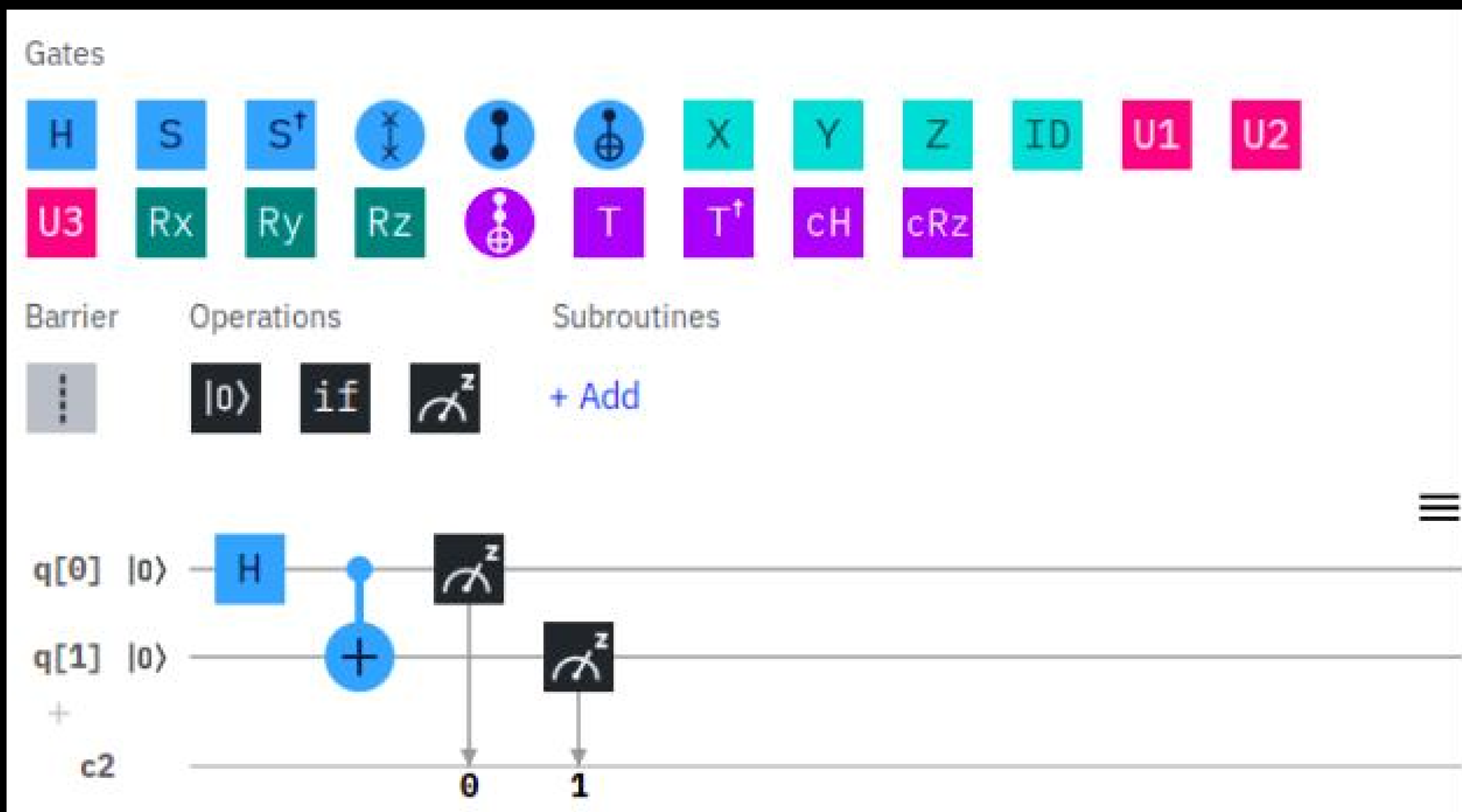
Tipos de computadoras cuánticas

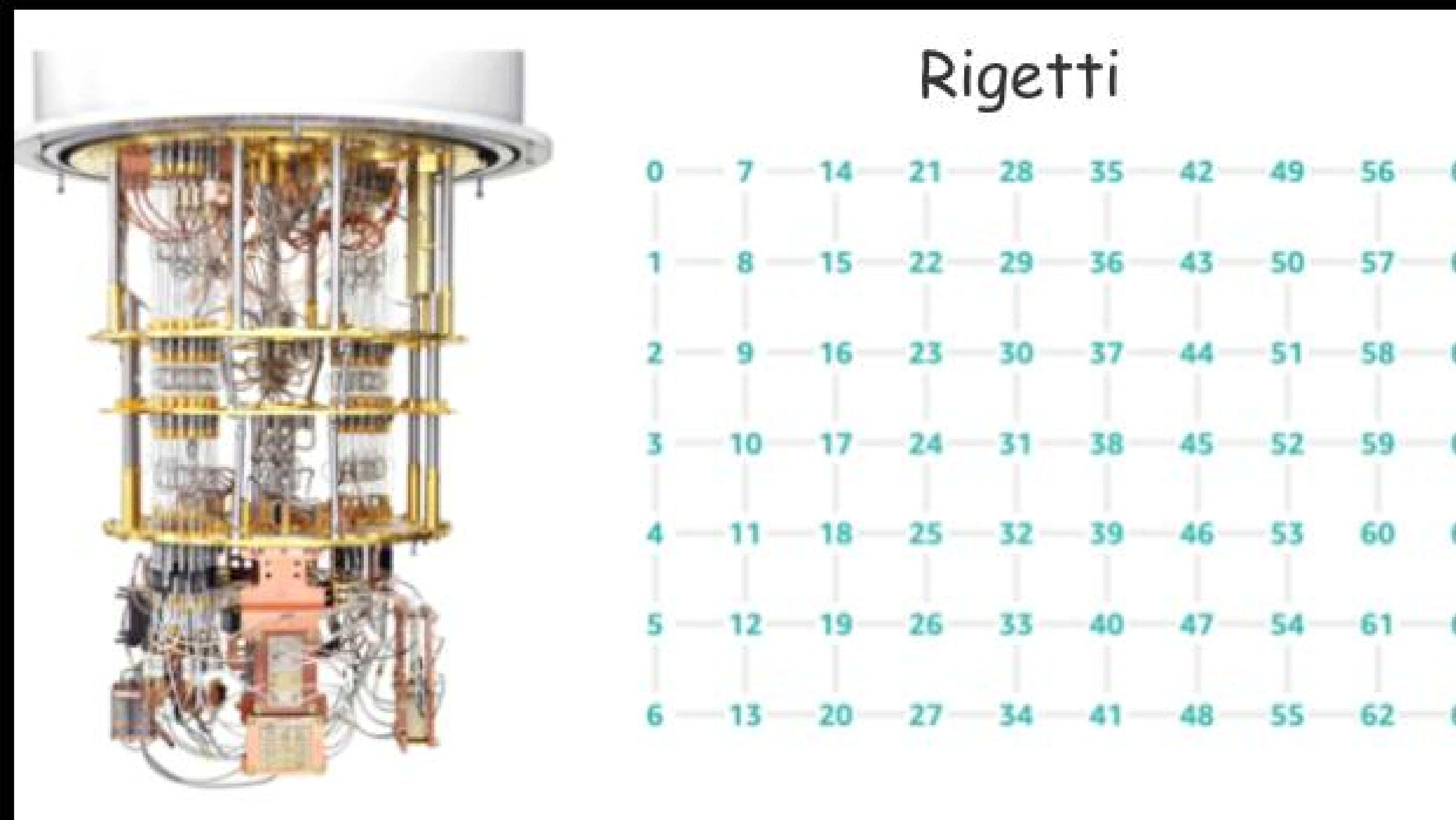
Propósitos generales



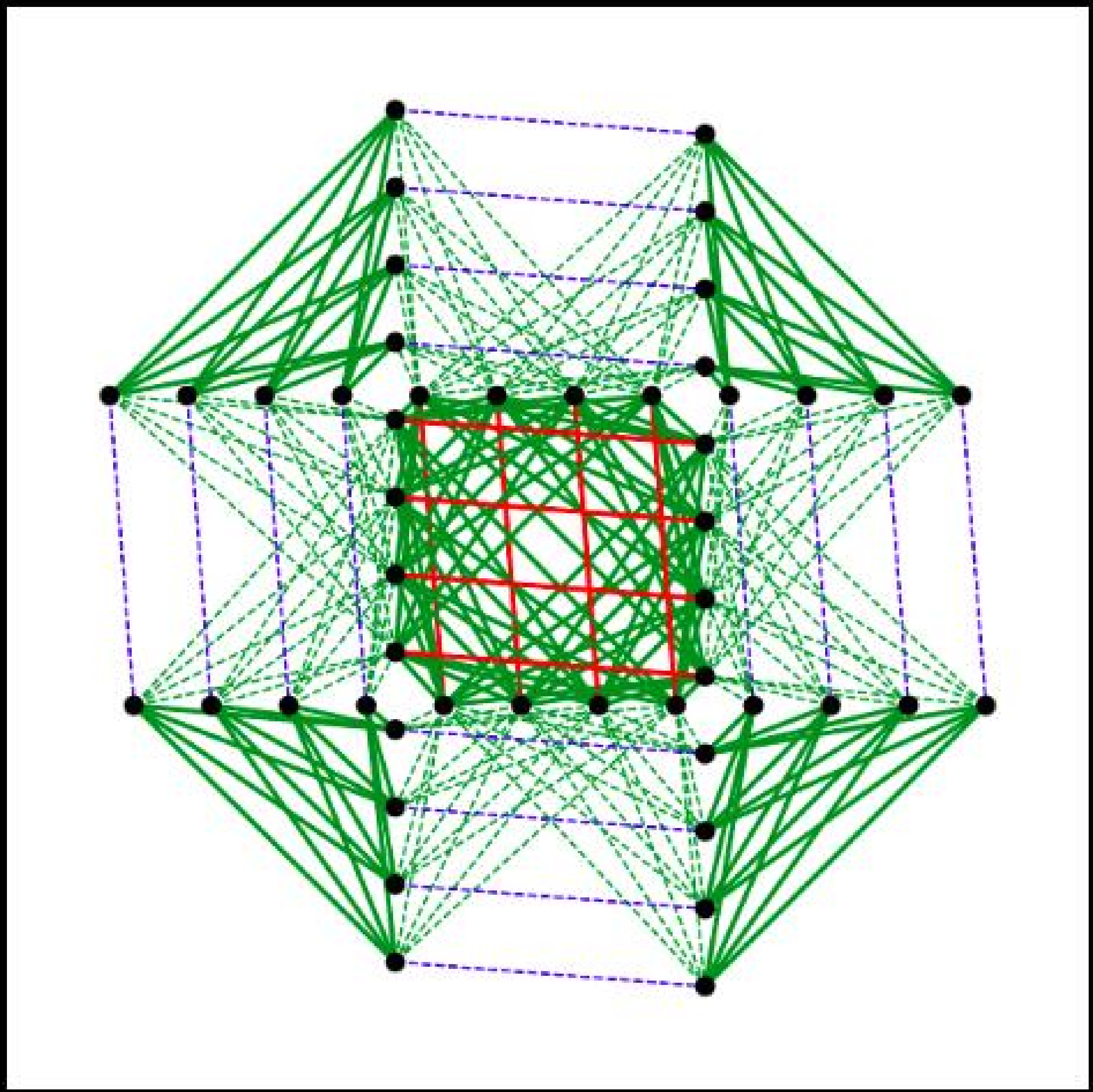
Google

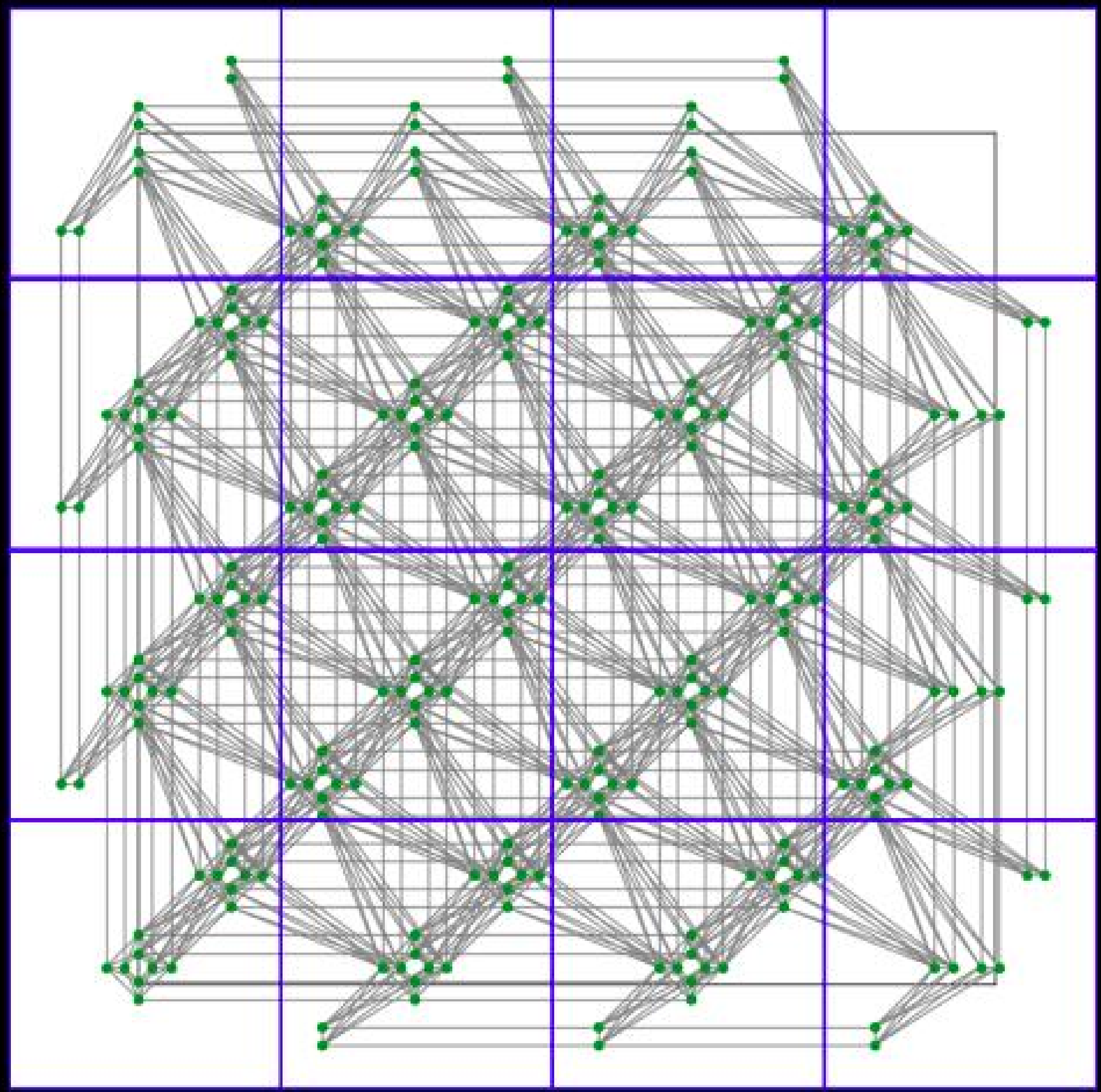






Adiabáticas



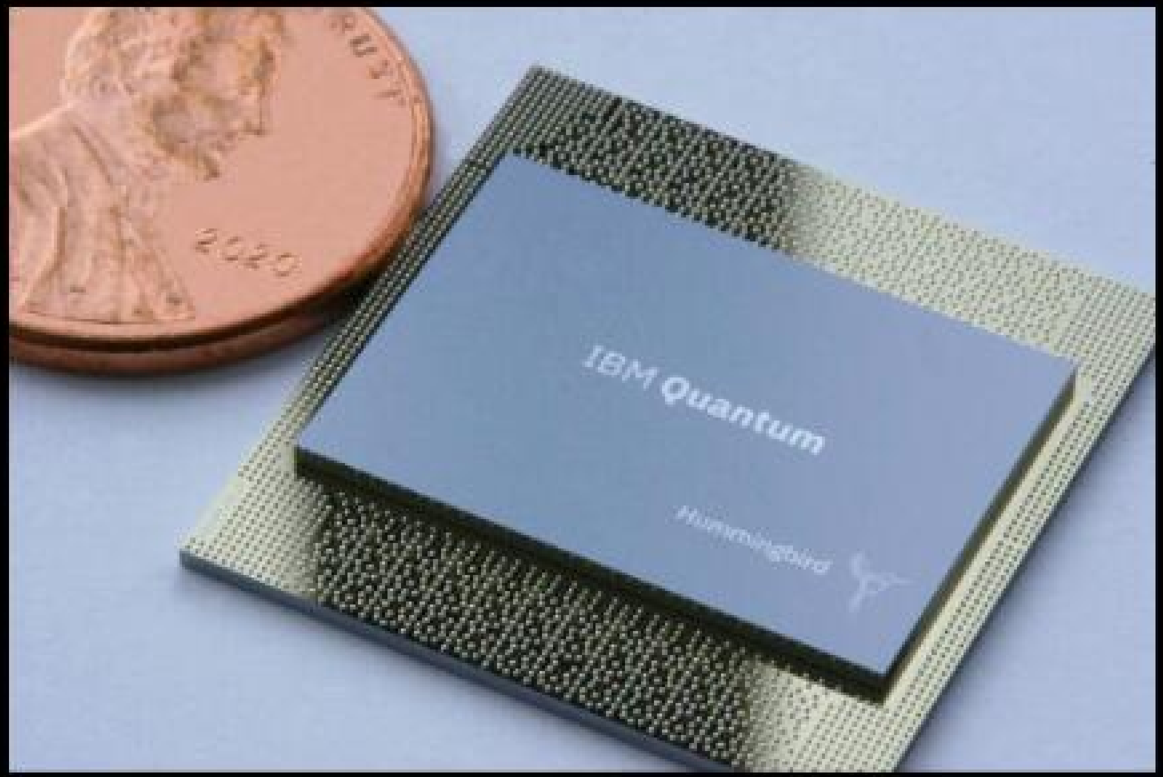


Dwave

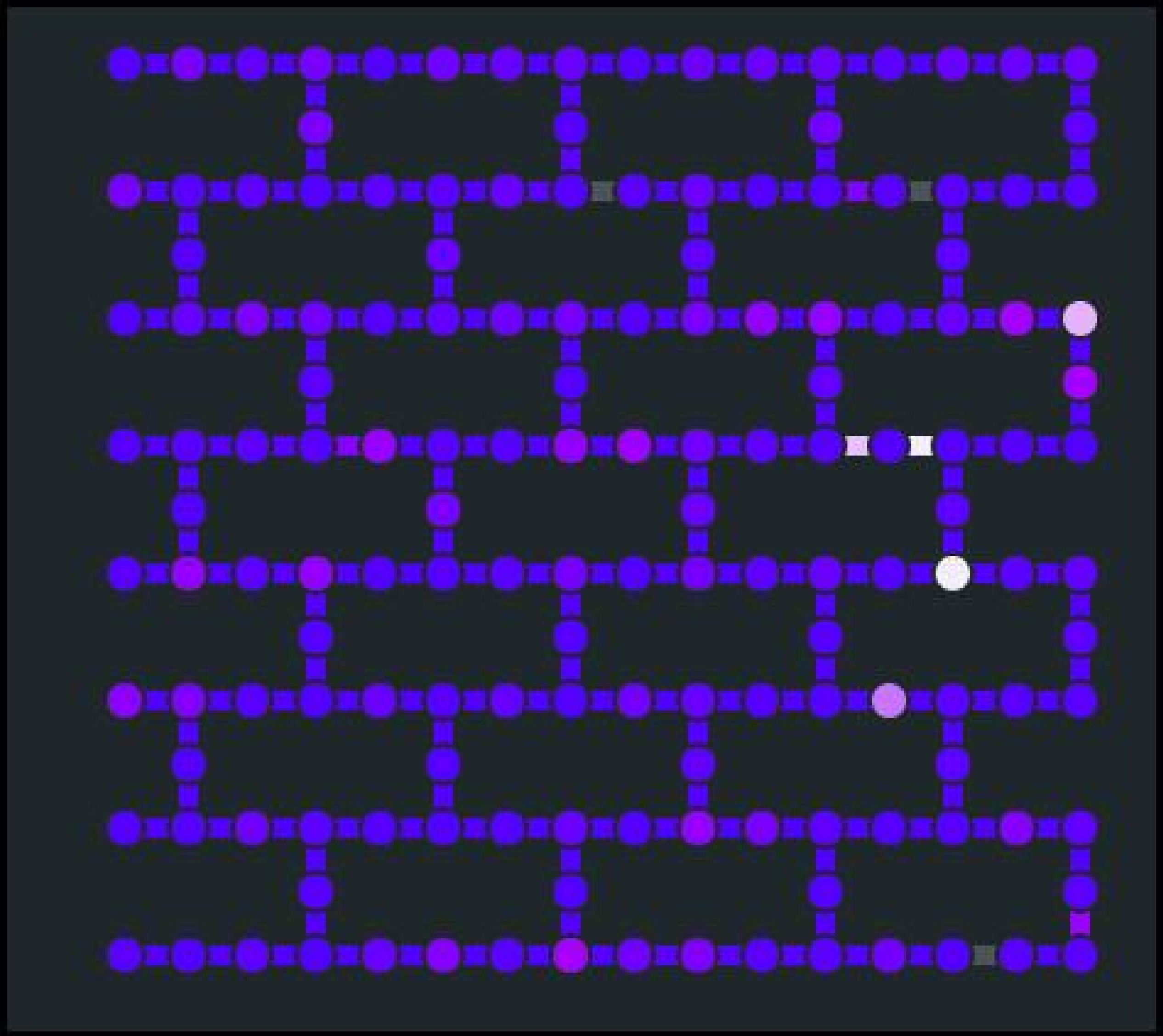
Simuladas

Evolución tecnológica

2010
experimental
1 qubit

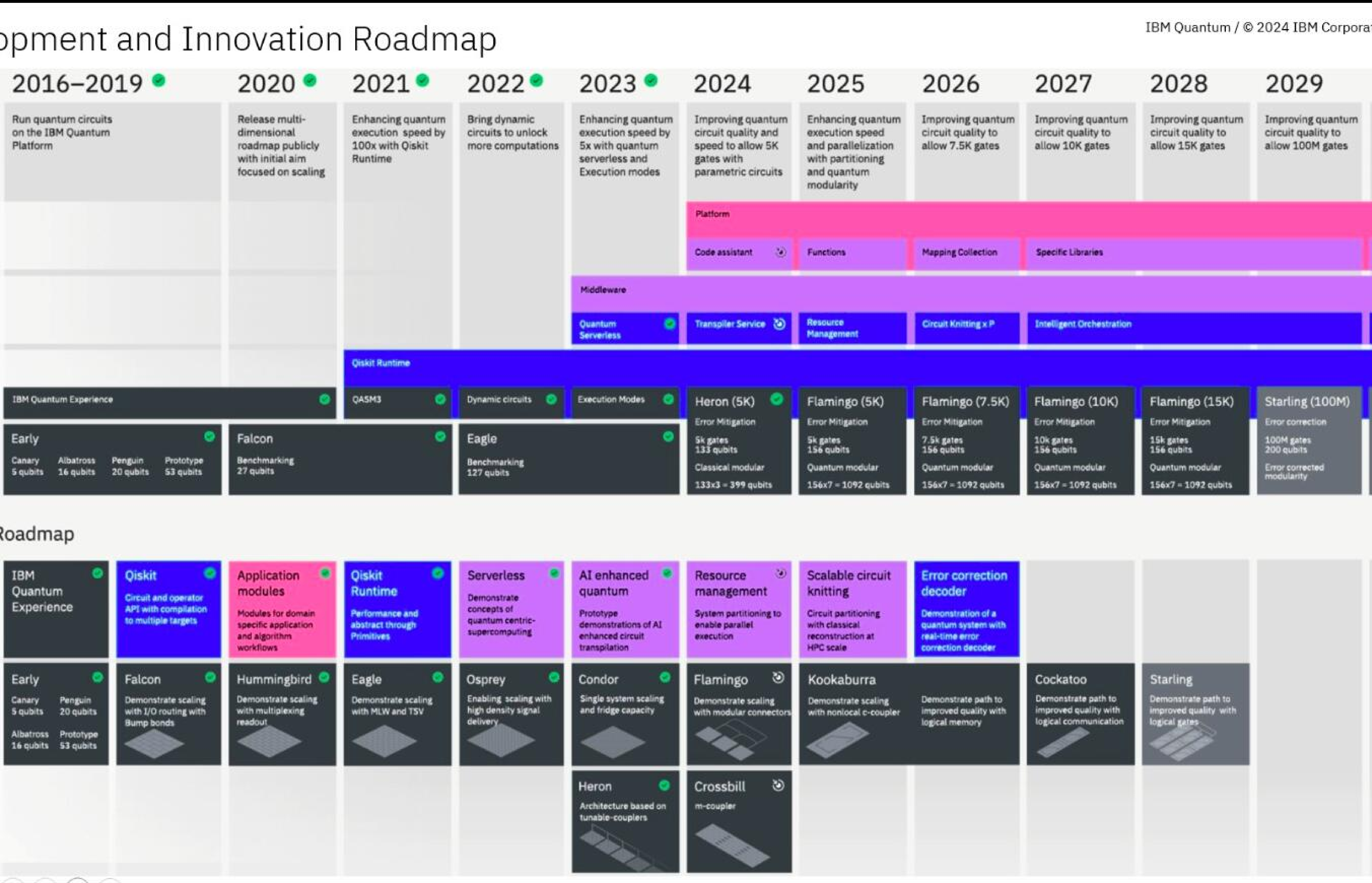


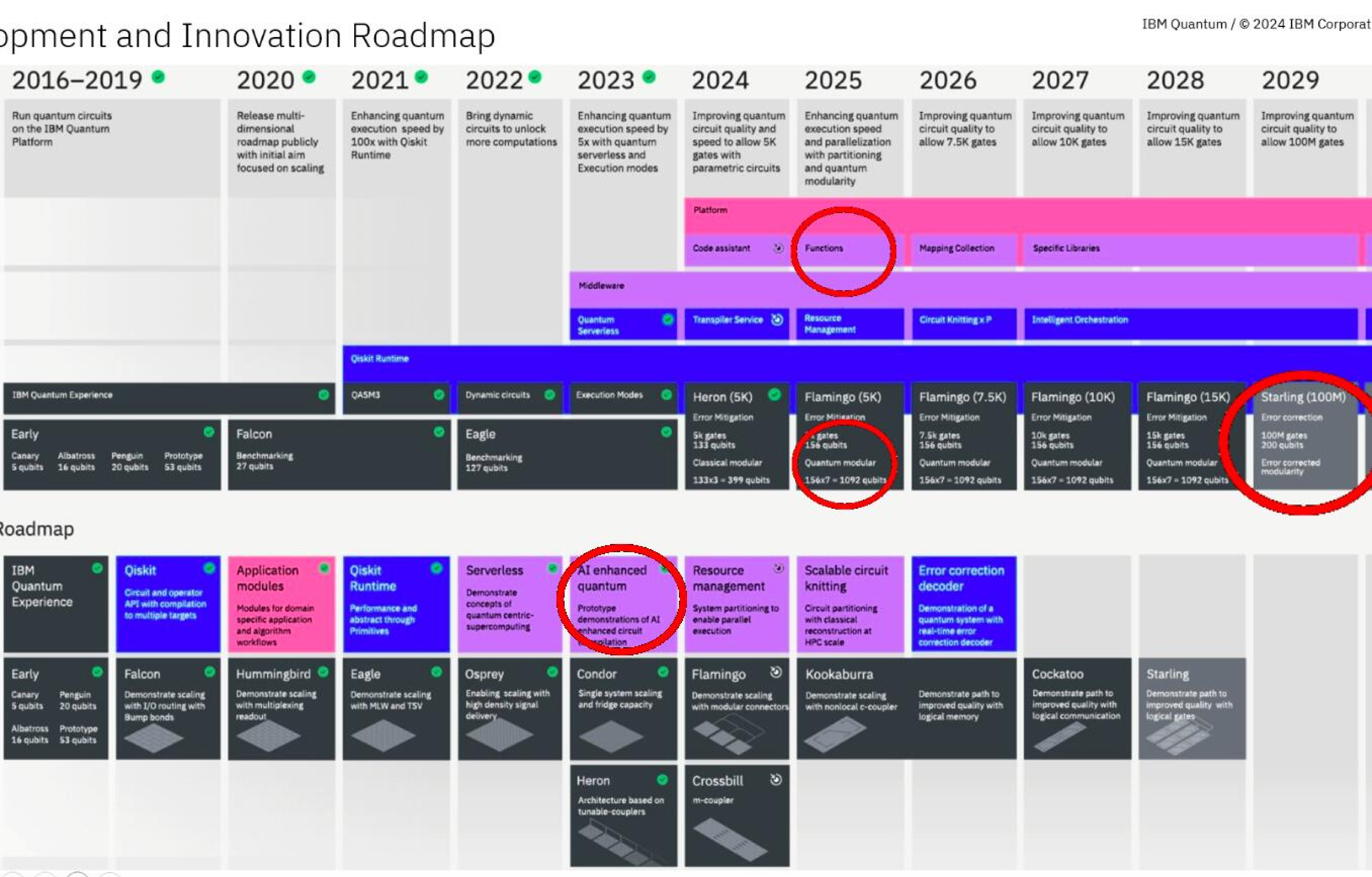
2020
hummingbird
63 qubits



2024
heron r2
156 qubits

Proyecciones

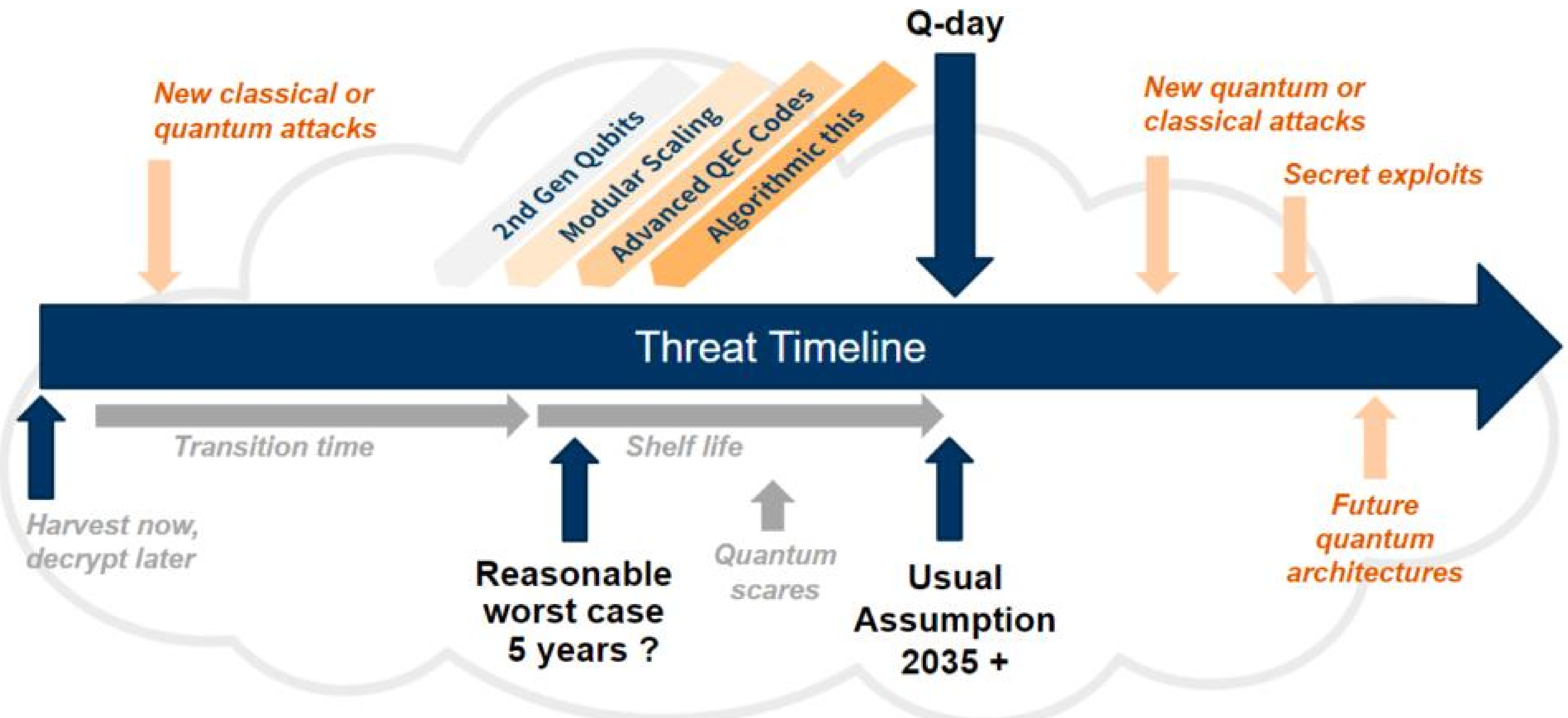

opment and Innovation Roadmap
IBM Quantum / © 2024 IBM Corporat
2016–2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
Run quantum circuits on the IBM Quantum Platform
Release multi-dimensional roadmap publicly with initial aim focused on scaling
Enhancing quantum execution speed by 100x with Qiskit Runtime
Bring dynamic circuits to unlock more computations
Enhancing quantum execution speed by 5x with quantum serverless and Execution modes
Improving quantum circuit quality and speed to allow 5K gates with parametric circuits
Enhancing quantum execution speed and parallelization with partitioning and quantum modularity
Improving quantum circuit quality to allow 7.5K gates
Improving quantum circuit quality to allow 10K gates
Improving quantum circuit quality to allow 15K gates
Improving quantum circuit quality to allow 100M gates
Platform
Code assistant
Functions
Mapping Collection
Specific Libraries
Middleware
Quantum Serverless
Transpiler Service
Resource Management
Circuit Knitting x P
Intelligent Orchestration
Qiskit Runtime
IBM Quantum Experience
QASM3
Dynamic circuits
Execution Modes
Heron (5K)
Error Mitigation
5k gates
133 qubits
Classical modular
133x3 = 399 qubits
Flamingo (5K)
Error Mitigation
5k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (7.5K)
Error Mitigation
7.5k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (10K)
Error Mitigation
10k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (15K)
Error Mitigation
15k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Starling (100M)
Error correction
100M gates
200 qubits
Error corrected modularity
Early
Canary 5 qubits
Albatross 16 qubits
Penguin 20 qubits
Prototype 53 qubits
Falcon
Benchmarking
27 qubits
Eagle
Benchmarking
127 qubits
Roadmap
IBM Quantum Experience
Qiskit
Circuit and operator API with compilation to multiple targets
Application modules
Modules for domain specific application and algorithm workflows
Qiskit Runtime
Performance and abstract through Primitives
Serverless
Demonstrate concepts of quantum centric-supercomputing
AI enhanced quantum
Prototype demonstrations of AI enhanced circuit transpilation
Resource management
System partitioning to enable parallel execution
Scalable circuit knitting
Circuit partitioning with classical reconstruction at HPC scale
Error correction decoder
Demonstration of a quantum system with real-time error correction decoder
Early
Canary 5 qubits
Penguin 20 qubits
Albatross 16 qubits
Prototype 53 qubits
Falcon
Demonstrate scaling with I/O routing with Bump bonds
Hummingbird
Demonstrate scaling with multiplexing readout
Eagle
Demonstrate scaling with MLW and TSV
Osprey
Enabling scaling with high density signal delivery
Condor
Single system scaling and fridge capacity
Flamingo
Demonstrate scaling with modular connectors
Kookaburra
Demonstrate scaling with nonlocal c-coupler
Demonstrate path to improved quality with logical memory
Cockatoo
Demonstrate path to improved quality with logical communication
Starling
Demonstrate path to improved quality with logical gates
Heron
Architecture based on tunable-couplers
Crossbill
m-coupler



opment and Innovation Roadmap
IBM Quantum / © 2024 IBM Corporat
2016–2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
Run quantum circuits on the IBM Quantum Platform
Release multi-dimensional roadmap publicly with initial aim focused on scaling
Enhancing quantum execution speed by 100x with Qiskit Runtime
Bring dynamic circuits to unlock more computations
Enhancing quantum execution speed by 5x with quantum serverless and Execution modes
Improving quantum circuit quality and speed to allow 5K gates with parametric circuits
Enhancing quantum execution speed and parallelization with partitioning and quantum modularity
Improving quantum circuit quality to allow 7.5K gates
Improving quantum circuit quality to allow 10K gates
Improving quantum circuit quality to allow 15K gates
Improving quantum circuit quality to allow 100M gates
Platform
Code assistant
Functions
Mapping Collection
Specific Libraries
Middleware
Quantum Serverless
Transpiler Service
Resource Management
Circuit Knitting x P
Intelligent Orchestration
Qiskit Runtime
IBM Quantum Experience
QASM3
Dynamic circuits
Execution Modes
Heron (5K)
Error Mitigation
5k gates
133 qubits
Classical modular
133x3 = 399 qubits
Flamingo (5K)
Error Mitigation
5k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (7.5K)
Error Mitigation
7.5k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (10K)
Error Mitigation
10k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Flamingo (15K)
Error Mitigation
15k gates
156 qubits
Quantum modular
156x7 = 1092 qubits
Starling (100M)
Error correction
100M gates
200 qubits
Error corrected modularity
Early
Canary 5 qubits
Albatross 16 qubits
Penguin 20 qubits
Prototype 53 qubits
Falcon
Benchmarking 27 qubits
Eagle
Benchmarking 127 qubits
Roadmap
IBM Quantum Experience
Qiskit
Circuit and operator API with compilation to multiple targets
Application modules
Modules for domain specific application and algorithm workflows
Qiskit Runtime
Performance and abstract through Primitives
Serverless
Demonstrate concepts of quantum centric-supercomputing
AI enhanced quantum
Prototype demonstrations of AI enhanced circuit transpilation
Resource management
System partitioning to enable parallel execution
Scalable circuit knitting
Circuit partitioning with classical reconstruction at HPC scale
Error correction decoder
Demonstration of a quantum system with real-time error correction decoder
Early
Canary 5 qubits
Penguin 20 qubits
Albatross 16 qubits
Prototype 53 qubits
Falcon
Demonstrate scaling with I/O routing with Bump bonds
Hummingbird
Demonstrate scaling with multiplexing readout
Eagle
Demonstrate scaling with MLW and TSV
Osprey
Enabling scaling with high density signal delivery
Condor
Single system scaling and fridge capacity
Flamingo
Demonstrate scaling with modular connectors
Kookaburra
Demonstrate scaling with nonlocal c-coupler
Demonstrate path to improved quality with logical memory
Cockatoo
Demonstrate path to improved quality with logical communication
Starling
Demonstrate path to improved quality with logical gates
Heron
Architecture based on tunable-couplers
Crossbill
m-coupler

El apocalipsis

Q-day demands a reasonable worst case mindset



Post quantum NIST standards

#1 - FIPS 203: CRYSTALS-Kyber (-> ML-KEM)

#2 - FIPS 204: CRYSTALS-Dilithium (-> ML-DSA)

#3 - FIPS 205: Sphincs+ (-> SLH-DSA)

(#4 - FIPS 206: FALCON (-> FN-DSA))

Parte 2

Quantum Security

Algoritmo de Shor

Algoritmo de Grover

QKD

Hackear computadoras cuánticas

....?

Hands on 1

Instalar qiskit

```
#1 - Install miniconda

#2 - Create conda environment
  conda create --name qiskitpg

#3 - Activate environment
  conda activate qiskitpg

#4 - Install pip
  conda install pip

#5- Install qiskit
  pip install qiskit

#6 - Install additional libs
  pip install matplotlib
  pip install qiskit_ibm_runtime
  pip install pylatexenc

#7 - Create a new Jupyter Notebook file
  go to vscode
```

Instalar qiskit

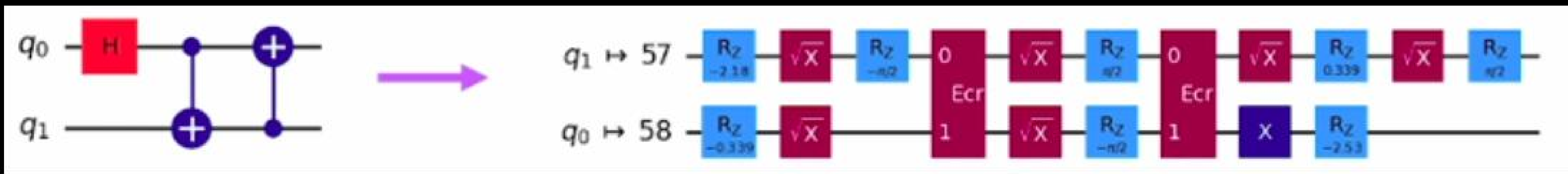
```
#8 - Select kernel (qiskitipg)
  auto install: ipykernel

#9 In the ipynb cell:
  import qiskit
  qiskit.__version__

#10 Instanciate IBM quantum services
  go to https://quantum.ibm.com
  get the token

#11 - In the ipynb
  from qiskit_ibm_runtime import QiskitRuntimeService
  service = QiskitRuntime Service(channel="ibm_quantum", token=
"XXX")
  service = QiskitRuntime
Service.save_account(channel="ibm_quantum",
token= "XXX")

#12 Connect to a real device
  backend = service.backend(name="ibm_brisbane")
  backend.num_qubits
```

Pasos para ir de un problema a un circuito cuántico

1. Map problem to quantum circuits and operators	2. Optimize circuits for target hardware	3. Execute on target hardware	4. Postprocess results

transpilation

...compilation...?



simulation

```
# Run the sampler job locally using FakeManilaV2
fake_manila = FakeManilaV2()
pm = generate_preset_pass_manager(backend=fake_manila, optimization_level=1)
isa_qc = pm.run(qc)

# You can use a fixed seed to get fixed results.
options = {"simulator": {"seed_simulator": 42}}
sampler = Sampler(backend=fake_manila, options=options)

result = sampler.run([isa_qc]).result()
```

<= 50 qubits

Errores

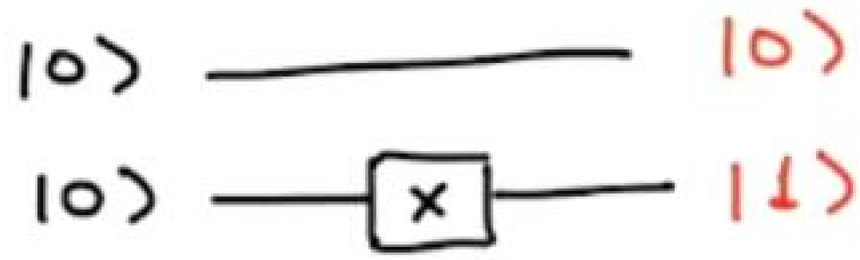
- Gate
- Decoherence
- Readout

```
backend = service.backend(name="<backend_name>")
print(backend.target)
```

Data encoding

Basis encoding

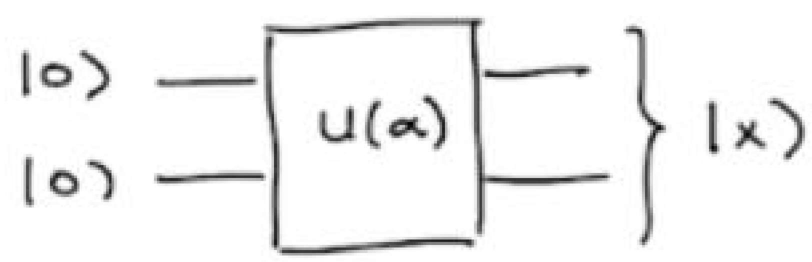
$$\overrightarrow{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 01 \\ 00 \\ 11 \end{bmatrix} = \begin{bmatrix} |11\rangle \\ |01\rangle \\ |00\rangle \\ |11\rangle \end{bmatrix}$$



Data encoding

Amplitude encoding

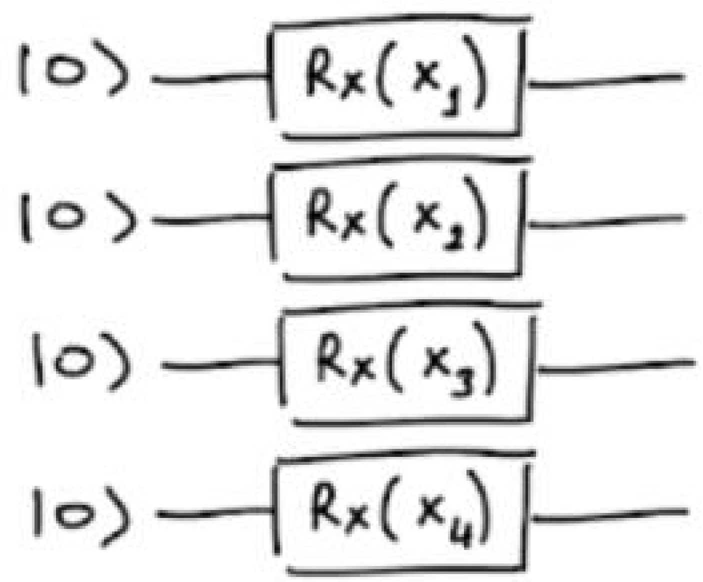
$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3/\sqrt{19} \\ 1/\sqrt{19} \\ 0/\sqrt{19} \\ 3/\sqrt{19} \end{bmatrix}$$



Data encoding

Angle encoding

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3/\sqrt{19} \\ 1/\sqrt{19} \\ 0/\sqrt{19} \\ 3/\sqrt{19} \end{bmatrix}$$



Map problem to quantum circuits and operators

Ansatz

Resultados posibles

- No se puede crear el circuito/no ejecuta
- Ejecuta, pero no hay ventaja
- Ejecuta y hay ventaja

Ejemplos de problemas en seguridad

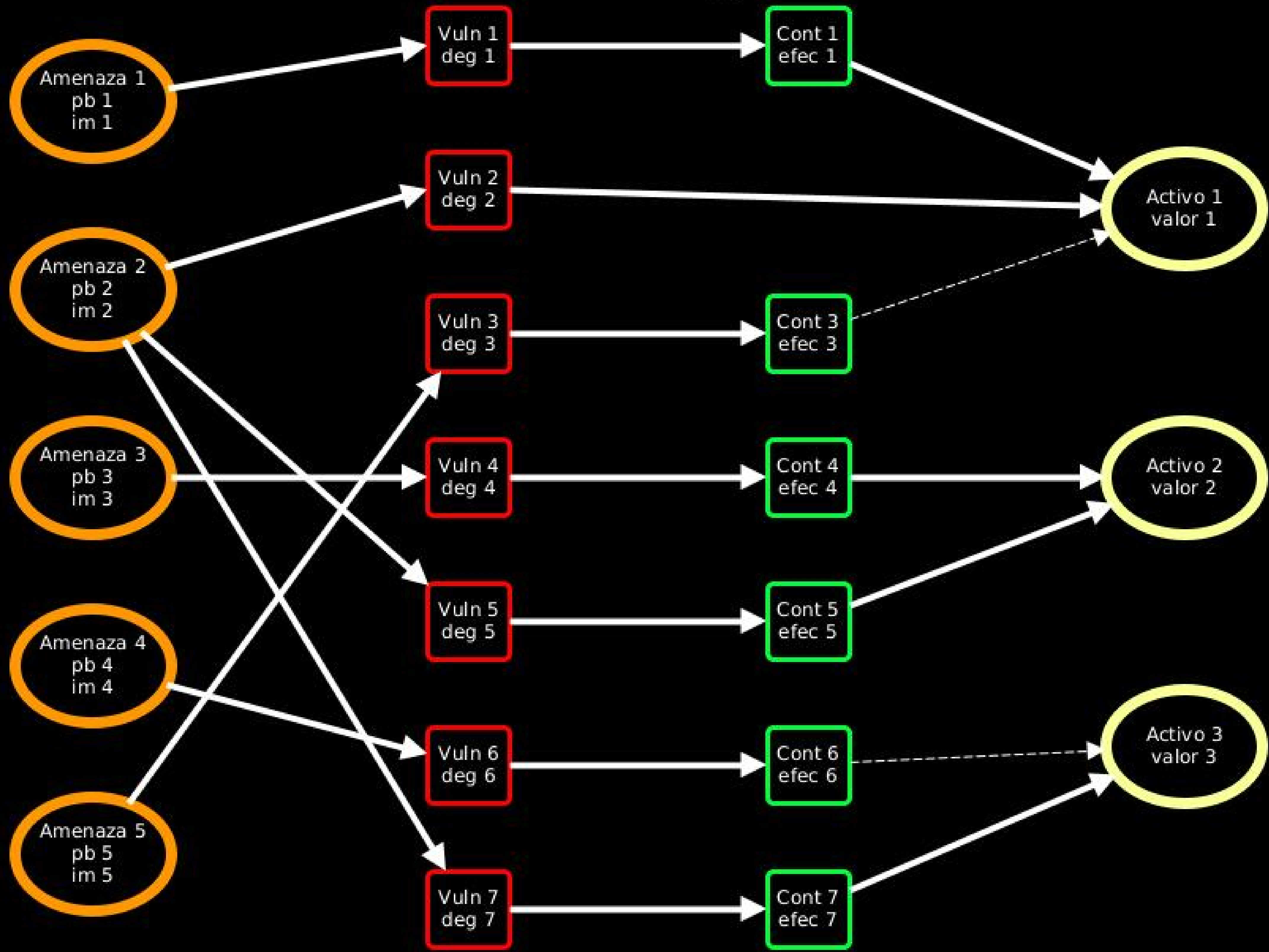
Cálculo de riesgo de IT

Riesgos de IT en una organización.

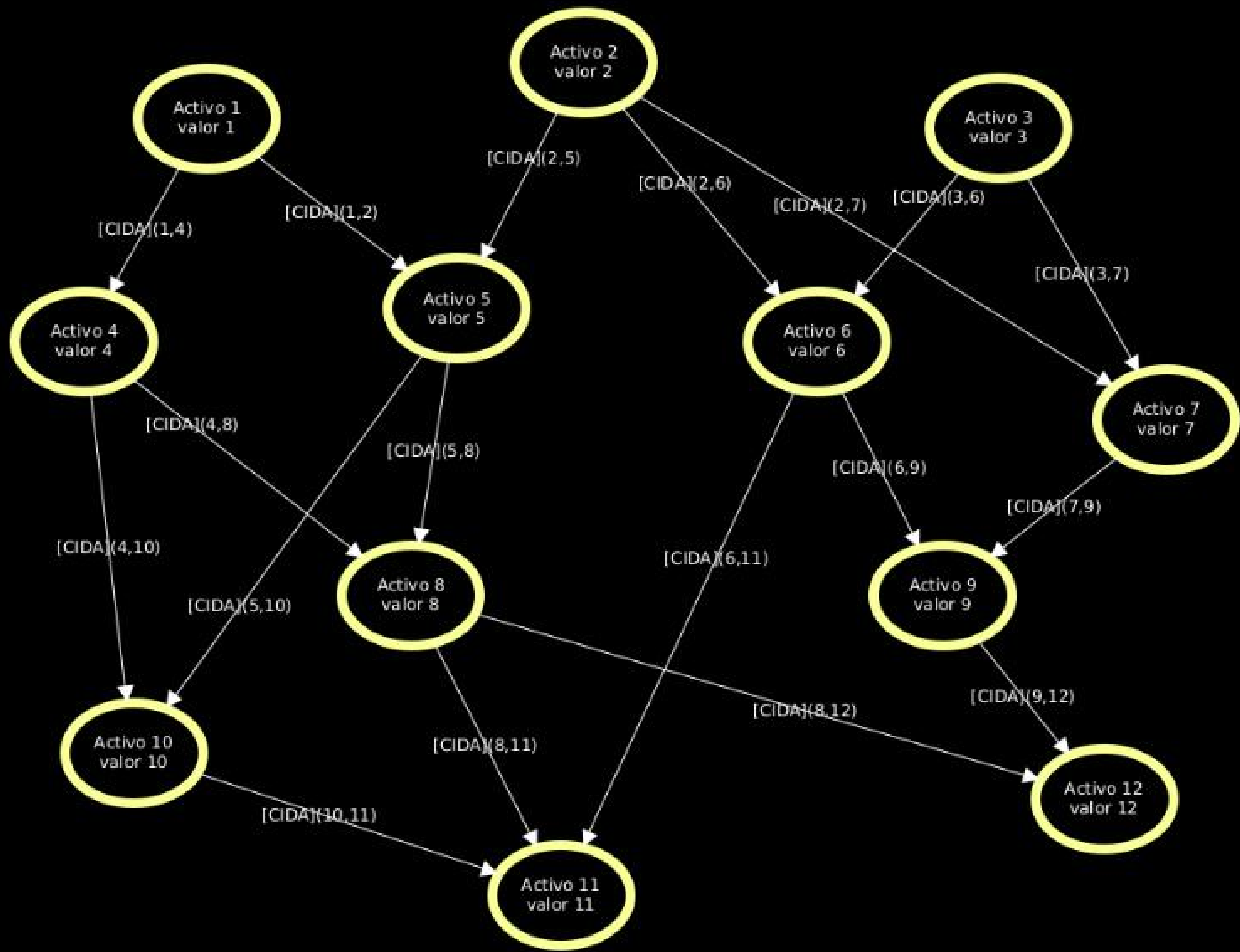
- Activos (de información) [valor]
- Amenazas [probabilidad/impacto]
- Vulnerabilidades [degradacción]
- Contramedidas [efectividad]

- Dependencias entre activos

Cálculo de riesgo de IT



Cálculo de riesgo de IT



Cálculo de riesgo de IT

Assets:	Threats:
A1: Web Server	T1: SQL Injection
A2: Database	T2: DDoS Attack
A3: File Server	T3: Data Exfiltration
A4: Application Server	T4: Phishing Attack
A5: Email Server	T5: Insider Threat
A6: Backup Server	T6: Ransomware
A7: HR Database	T7: Zero-Day Exploit
A8: Finance Database	T8: Man-in-the-Middle Attack
A9: Customer Portal	T9: Brute Force Attack
A10: Internal Network	T10: Malware Injection

Cálculo de riesgo de IT

Probability matrix:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
T1:	[0.8,	0.6,	0.0,	0.2,	0.1,	0.3,	0.4,	0.5,	0.2,	0.1]
T2:	[0.7,	0.2,	0.1,	0.0,	0.3,	0.5,	0.2,	0.4,	0.3,	0.2]
T3:	[0.1,	0.4,	0.5,	0.3,	0.2,	0.6,	0.1,	0.0,	0.4,	0.3]
T4:	[0.3,	0.1,	0.2,	0.5,	0.4,	0.2,	0.3,	0.1,	0.3,	0.5]
T5:	[0.4,	0.3,	0.2,	0.0,	0.5,	0.1,	0.4,	0.6,	0.2,	0.1]
T6:	[0.2,	0.5,	0.3,	0.1,	0.4,	0.7,	0.3,	0.2,	0.6,	0.4]
T7:	[0.6,	0.2,	0.4,	0.3,	0.0,	0.3,	0.5,	0.2,	0.1,	0.6]
T8:	[0.5,	0.4,	0.3,	0.6,	0.2,	0.3,	0.2,	0.5,	0.4,	0.2]
T9:	[0.3,	0.6,	0.4,	0.2,	0.5,	0.2,	0.6,	0.1,	0.3,	0.7]
T10:	[0.4,	0.3,	0.5,	0.4,	0.6,	0.2,	0.3,	0.4,	0.5,	0.6]

Cálculo de riesgo de IT

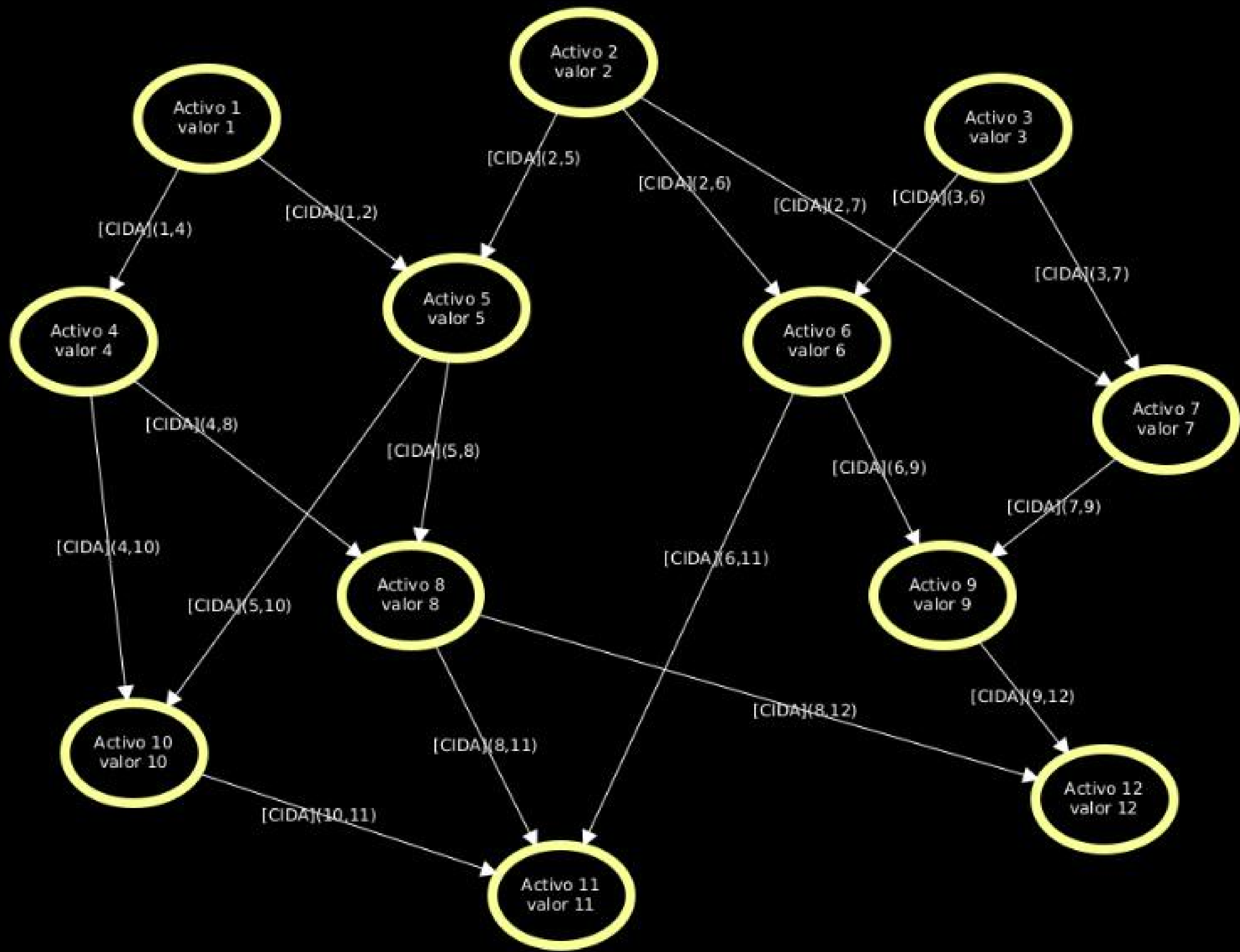
Impact matrix:

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10

T1: [0.9, 0.8, 0.3, 0.7, 0.5, 0.4, 0.6, 0.9, 0.7, 0.5]
T2: [0.6, 0.5, 0.4, 0.3, 0.7, 0.8, 0.6, 0.5, 0.8, 0.6]
T3: [0.4, 0.7, 0.8, 0.5, 0.6, 0.7, 0.4, 0.3, 0.8, 0.7]
T4: [0.7, 0.3, 0.5, 0.8, 0.4, 0.5, 0.7, 0.2, 0.6, 0.8]
T5: [0.5, 0.6, 0.4, 0.2, 0.9, 0.3, 0.8, 0.9, 0.6, 0.3]
T6: [0.3, 0.7, 0.5, 0.4, 0.6, 0.9, 0.5, 0.4, 0.8, 0.5]
T7: [0.8, 0.4, 0.6, 0.5, 0.3, 0.4, 0.9, 0.3, 0.2, 0.9]
T8: [0.5, 0.6, 0.5, 0.7, 0.5, 0.4, 0.6, 0.8, 0.7, 0.4]
T9: [0.6, 0.8, 0.6, 0.4, 0.7, 0.3, 0.7, 0.4, 0.5, 0.9]
T10: [0.7, 0.4, 0.8, 0.6, 0.9, 0.5, 0.5, 0.6, 0.8, 0.8]

Hands on 2

Cálculo de riesgo de IT



Cálculo de riesgo de IT

Dependency matrix (simplified):

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1:	[0.0,	0.3,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.2]
A2:	[0.0,	0.0,	0.0,	0.5,	0.0,	0.0,	0.0,	0.0,	0.1,	0.0]
A3:	[0.2,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.3,	0.0,	0.0]
A4:	[0.0,	0.2,	0.1,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.4]
A5:	[0.1,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.5,	0.0,	0.0]
A6:	[0.0,	0.3,	0.2,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0]
A7:	[0.0,	0.0,	0.0,	0.4,	0.0,	0.0,	0.0,	0.2,	0.0,	0.0]
A8:	[0.0,	0.1,	0.3,	0.0,	0.2,	0.0,	0.0,	0.0,	0.0,	0.0]
A9:	[0.3,	0.3,	0.0,	0.3,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0]
A10:	[0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.1,	0.0,	0.3,	0.0]

Hands on 3

Cálculo de riesgo de IT

Threat-Vulnerability matrix (degradations):

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
T1:	[0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.7,	0.0,	0.0]
T2:	[0.0,	0.3,	0.0,	0.8,	0.0,	0.0,	0.6,	0.0,	0.0,	0.0]
T3:	[0.0,	0.0,	0.2,	0.0,	0.0,	0.4,	0.0,	0.0,	0.0,	0.5]
T4:	[0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.7,	0.0]
T5:	[0.0,	0.8,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0]
T6:	[0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.6]
T7:	[0.0,	0.9,	0.7,	0.0,	0.5,	0.0,	0.0,	0.3,	0.0,	0.0]
T8:	[0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.9,	0.0,	0.0,	0.0]
T9:	[0.0,	0.0,	0.0,	0.0,	0.8,	0.0,	0.0,	0.0,	0.0,	0.0]
T10:	[0.0,	0.0,	0.6,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0]

Cálculo de riesgo de IT

Control matrix (efectiveness):

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
T1:	[0.8,	0.6,	0.0,	0.0,	0.1,	0.0,	0.4,	0.5,	0.2,	0.1]
T2:	[0.0,	0.2,	0.1,	0.0,	0.3,	0.5,	0.2,	0.4,	0.3,	0.2]
T3:	[0.0,	0.0,	0.5,	0.3,	0.2,	0.6,	0.1,	0.0,	0.4,	0.0]
T4:	[0.3,	0.0,	0.2,	0.5,	0.4,	0.2,	0.3,	0.1,	0.3,	0.0]
T5:	[0.0,	0.0,	0.2,	0.0,	0.5,	0.1,	0.4,	0.6,	0.2,	0.1]
T6:	[0.0,	0.0,	0.3,	0.1,	0.4,	0.7,	0.3,	0.2,	0.6,	0.4]
T7:	[0.6,	0.2,	0.0,	0.3,	0.0,	0.3,	0.5,	0.2,	0.1,	0.6]
T8:	[0.0,	0.4,	0.0,	0.6,	0.2,	0.3,	0.2,	0.5,	0.4,	0.2]
T9:	[0.3,	0.6,	0.4,	0.2,	0.5,	0.2,	0.6,	0.1,	0.3,	0.7]
T10:	[0.0,	0.3,	0.0,	0.0,	0.0,	0.2,	0.3,	0.4,	0.5,	0.6]

Bonus track

AI helping QC?

QC helping AI?

Conclusiones

- Apocalipsis cuántico para 2030/2035(?)
- Desarrollos para alejar la funcionalidad de las QC de la física
- Facilidad para usar algunas QC online
- Pensar en qué problemas de seguridad se pueden resolver con QC

Carlos Benitez
@ch4r1i3b
carlos<at>platinumciber.com
https://cybersonthestorm.com
https://github.com/ch4r1i3b

Ekoparty #20
15 de Noviembre de 2024
Carlos Benitez

Not only does God play dice, but... he sometimes throws them where they cannot be seen. (Stephen Hawking)

REFERENCIAS

Post Quantum NIST
https://csrc.nist.gov/projects/post-quantum-cryptography
https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards

Labs IBM Quantum
https://lab.quantum-computing.ibm.com/
https://quantum-computing.ibm.com/composer

Qiskit
https://github.com/Qiskit/qiskit
https://docs.quantum.ibm.com/
https://github.com/Qiskit/qiskit-ibm-runtime
https://www.ibm.com/quantum/ecosystem
https://docs.quantum.ibm.com/guides/install-qiskit

Richard Feynman hablando de cuántica
https://www.youtube.com/watch?v=xdZMXWmlp9g