



WORKSHOP GERCOM

PROGRAMANDO COM QISKIT

COMPUTAÇÃO QUÂNTICA

Polyana dos Santos Moraes

SUMÁRIO

- 01** COMPUTAÇÃO QUÂNTICA
- 02** IBM/QISKIT
- 03** EXECUÇÃO
- 04** OTIMIZAÇÃO
- 05** CONCLUSÃO

COMPUTAÇÃO QUÂNTICA



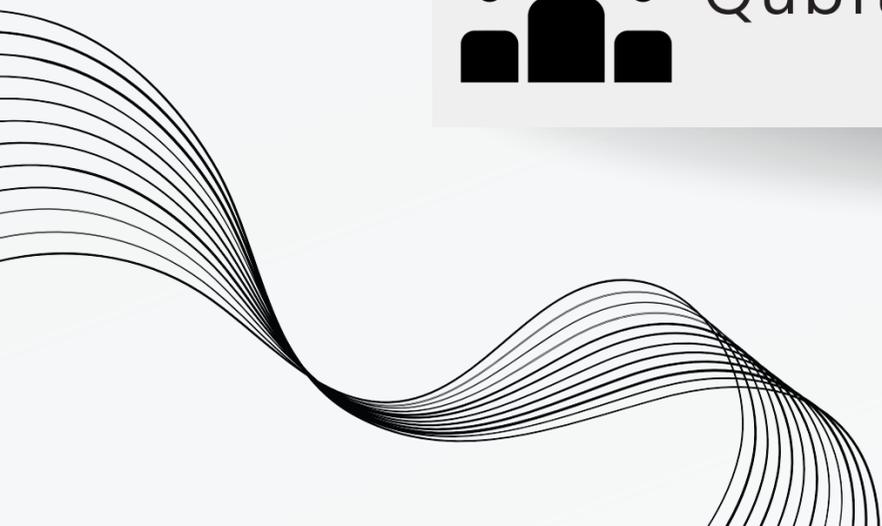
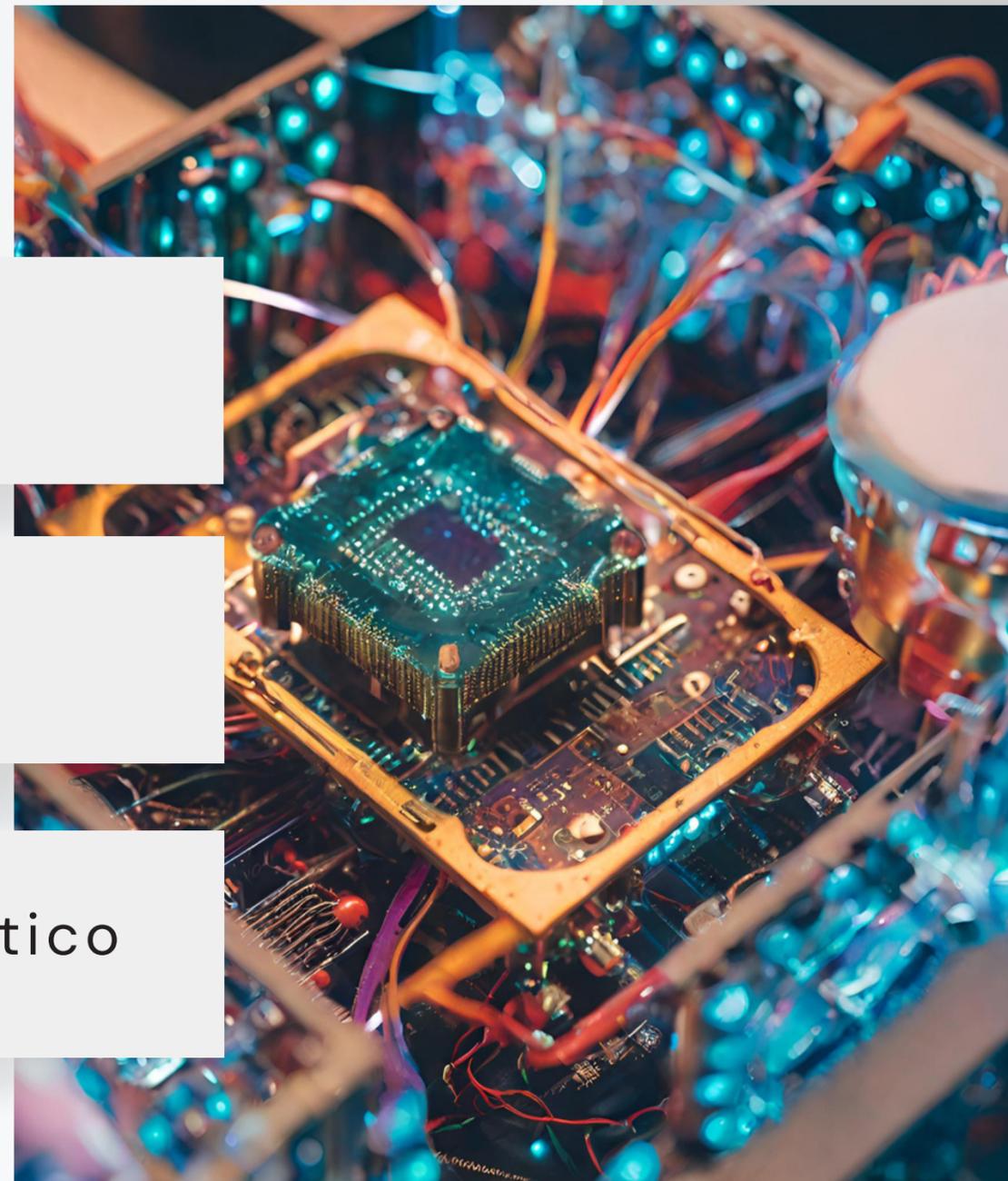
Objetivos para alcançar

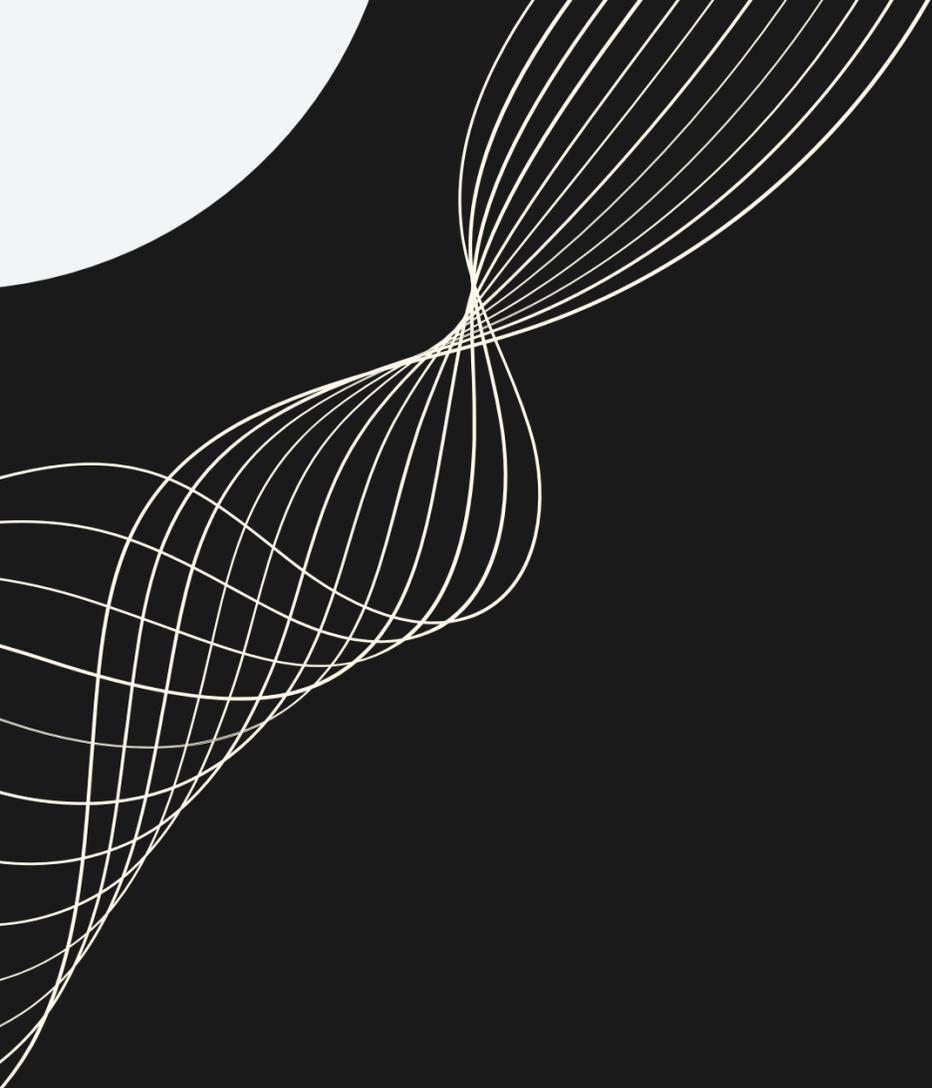


Estado de superposição



Qubit e o entrelaçamento quântico





IBM Quantum Platform | Dashboard | Compute resources | Jobs

IBM Quantum

Use our suite of applications to support your quantum research and development needs.

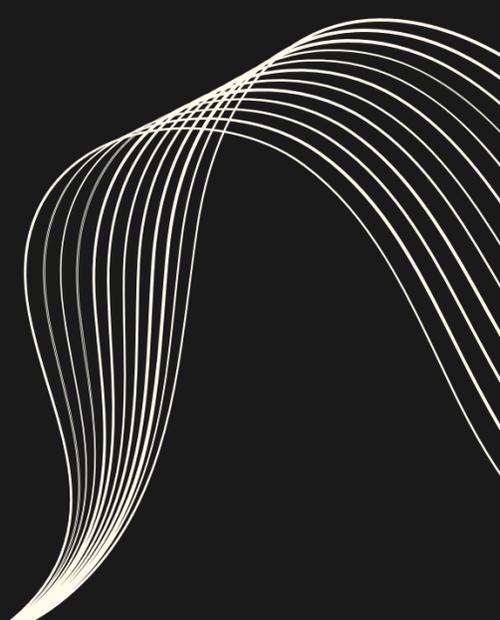
Platform

Copy your API token, track jobs, and view quantum compute resources.

Job ID	Status	Created	Completed	Compute resource
cmvtyr3605a00002m4g	Completed	About 2 mont...	About 2 mont...	ibm_qig1
cmvtyr3605a00004m0g	Completed	About 2 mont...	About 1 mont...	ibm_qig1

Realizar o cadastro na IBM Quantum, para acessar suas ferramentas.

IBM / QISKIT



Através do Qiskit, podemos ajustar ou criar um algoritmo e transformá-lo no circuito para sua execução em um hardware ou simulador na plataforma da IBM. Nesse primeiro momento, é demonstrado como acessar o servidor da IBM Quantum e a busca por um dos backend menos ocupados.

```
from qiskit_ibm_provider import IBMProvider

IBMProvider.save_account("2e433c09c9a3d034dc924cd7f280c49f7cc89518663e74863db06c8903eafcde3300afb6273405def2e71dd9864573e4f6db76c9e695f6c64fc11d67efela", overwrite=True)
IBMProvider.saved_accounts()

# ver na IBM quantum platform >> account >> manage account >> profile overvies
provider = IBMProvider()
```

Python

```
> from qiskit_ibm_provider import least_busy

devices = provider.backends(simulator=False, operational=True)
backend_least_busy = least_busy(devices)

backend_least_busy.status()
```

Python

```
[12] ...
<qiskit.providers.models.backendstatus.BackendStatus object at 0x7fb336ab2940>
```

```
name: ibm_kyoto
version: , pending jobs: 13
status: active
```

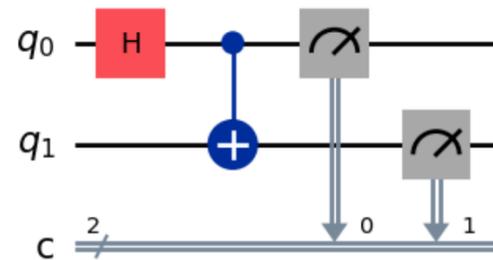


Assim, criamos uma variável chamada "backend" para armazenar o menos ocupado. Depois escrevemos nosso código e transpilamos.

```
: backend = backend_least_busy
backend

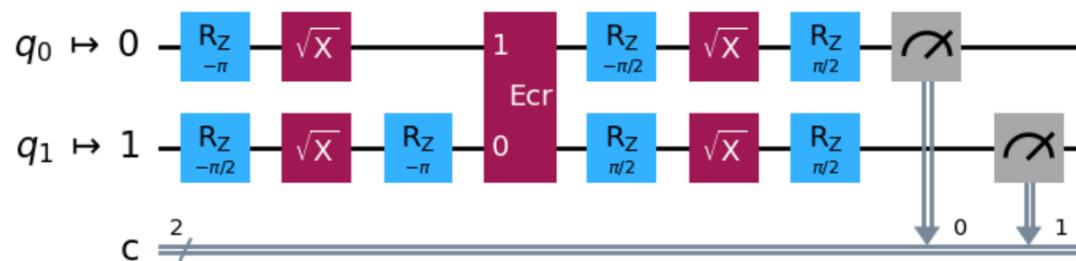
<IBMQBackend('ibm_kyoto')>

:
:
: from qiskit import QuantumCircuit, BasicAer, execute
: from qiskit.visualization import plot_histogram
:
: qc = QuantumCircuit(2, 2)
: qc.h(0)
: qc.cx(0, 1)
: qc.measure(range(2), range(2))
:
: display(qc.draw("mpl", style="iqp"))
```



```
: from qiskit import transpile
: qc_basis = transpile(qc, backend)
: qc_basis.draw(output='mpl', style='iqp', idle_wires=False)
```

Global Phase: $7\pi/4$



EXECUÇÃO

- No código ao lado, por meio do job, foi executado o circuito no backend menos ocupado da IBM.
- Demora cerca de minutos a horas para executar um job, pois existe uma fila na execução no jobs de outras pessoas.

```
#Executando o job
job = backend_least_busy.run(mapped_circuit, shots=1024, job_tags=["bell"])

[42]

job.status()

[28]
... <JobStatus.QUEUED: 'job is queued'>

job.job_id()

[29]
... 'cpn5wj t9nad0008pr4h0'
```

The screenshot shows the IBM Quantum Platform dashboard. At the top, there's a navigation bar with 'Dashboard', 'Compute resources', and 'Jobs'. Below this, a 'Monthly usage' section shows 'Used 4s' and 'Remaining 9m 56s'. A 'Recent jobs' table lists several jobs with their IDs, statuses, creation times, completion times, and compute resources. The first job is 'Queued' with an estimated wait time of 8 hours. The rest are 'Completed'. On the right, there's a 'What's new' section with product updates. At the bottom, there are sections for 'Instance systems', 'Documentation', and 'Learning'.

Job ID	Status	Created	Completed	Compute resource
crbcw0rbs5wg008apfvg	Queued Est. wait: 8 hours	15 minutes ago		ibm_osaka Queue position: 23
cr4zyc1cgeqg008xsh30	Completed	10 days ago	10 days ago	ibm_kyoto
cr4zx5wcgeqg008xsh2g	Completed	10 days ago	10 days ago	ibm_kyoto
cr1e9kedvs8g008bjcj0	Completed	15 days ago	15 days ago	ibm_kyoto
cr1e97mk5z700081wh50	Completed	15 days ago	15 days ago	ibm_kyoto

OTIMIZAÇÃO

- A otimização compreende aprimorar e achar melhores caminhos e alocações do qubit para que a execução do código seja mais eficiente e com menos erros.

```
from qiskit.providers.fake_provider import FakeCambridgeV2
backend = FakeCambridgeV2()
print(backend.target)
```

Target

Number of qubits: 28

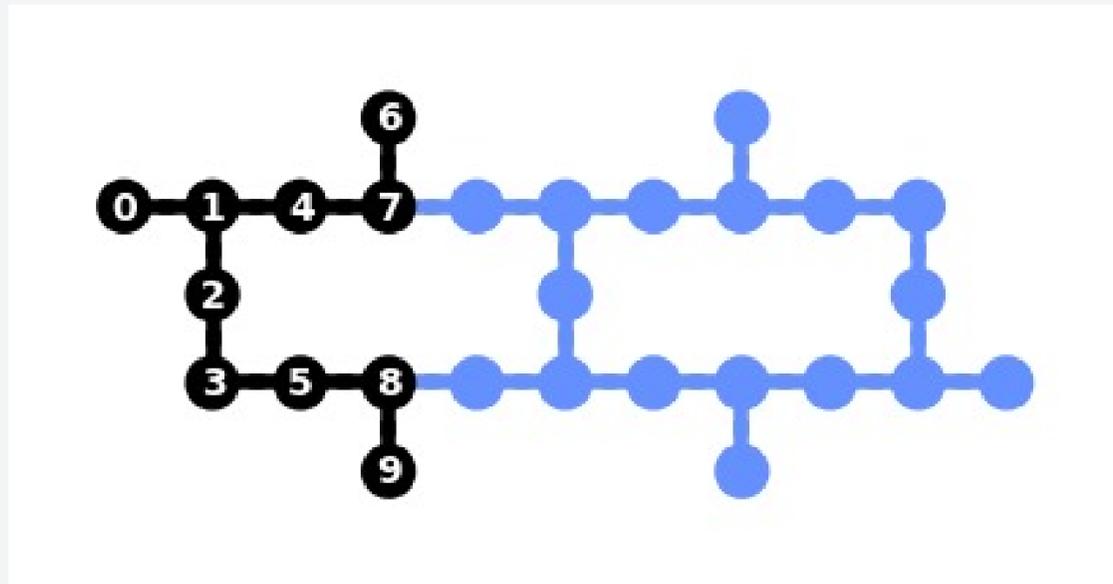
Instructions:

- Depois teremos que realizar o mesmo processo de montar nosso circuito, transpilá-lo para o backend e depois poder observar como é a sua topologia nos 4 níveis de otimização.

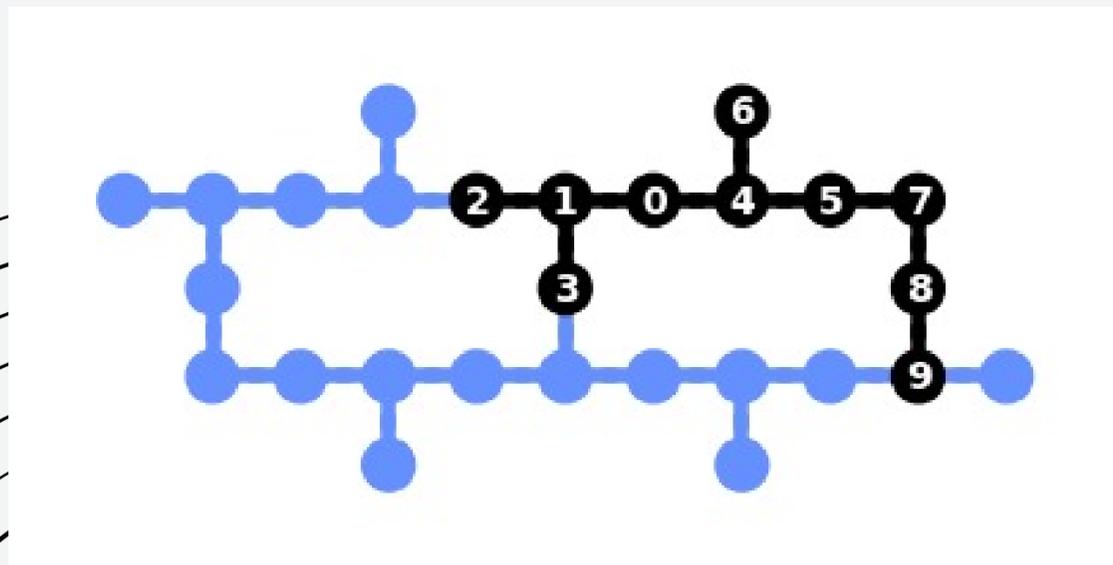
```
from qiskit.visualization import plot_circuit_layout
new_circ_lv0 = transpile(qc, backend=backend, optimization_level=0)
plot_circuit_layout(new_circ_lv0, backend)
```

Fakebackend/Simulador Cambrigde

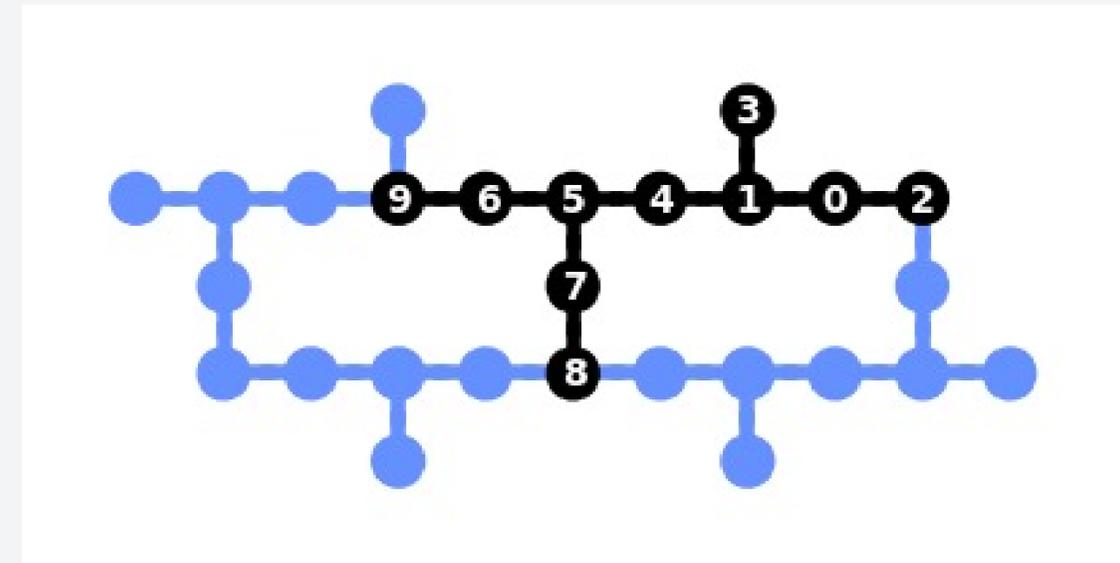
Nível 0



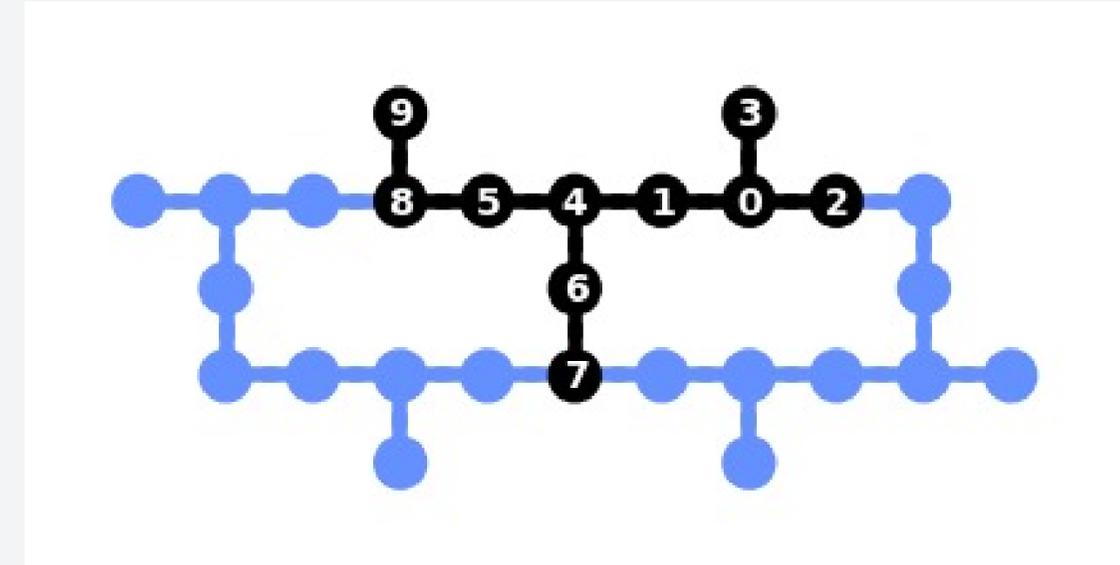
• Nível 2



• Nível 1



• Nível 3



- Para me conectar com o hardware, eu preciso seguir esses passos para escolher qual backend quero selecionar. No exemplo abaixo, foi escolhido o "ibm_osaka".

```
from qiskit_ibm_provider import IBMProvider
```

```
IBMProvider.save_account("2e433c09c9a3d034dc924cd7f280c49f7cc89518663e74863db06c8903eafcde3300afb6273405def2e71dd986457")  
IBMProvider.saved_accounts()
```

```
# pode ser necessário definir o argumento "instance"  
# ver na IBM quantum platform >> account >> manage account >> profile overvies  
provider = IBMProvider()
```

```
quantum_devices_available = provider.backends(simulator=False, operational=True)
```

```
quantum_devices_available
```

```
[<IBMBackend('ibm_brisbane')>,  
<IBMBackend('ibm_kyoto')>,  
<IBMBackend('ibm_osaka')>]
```

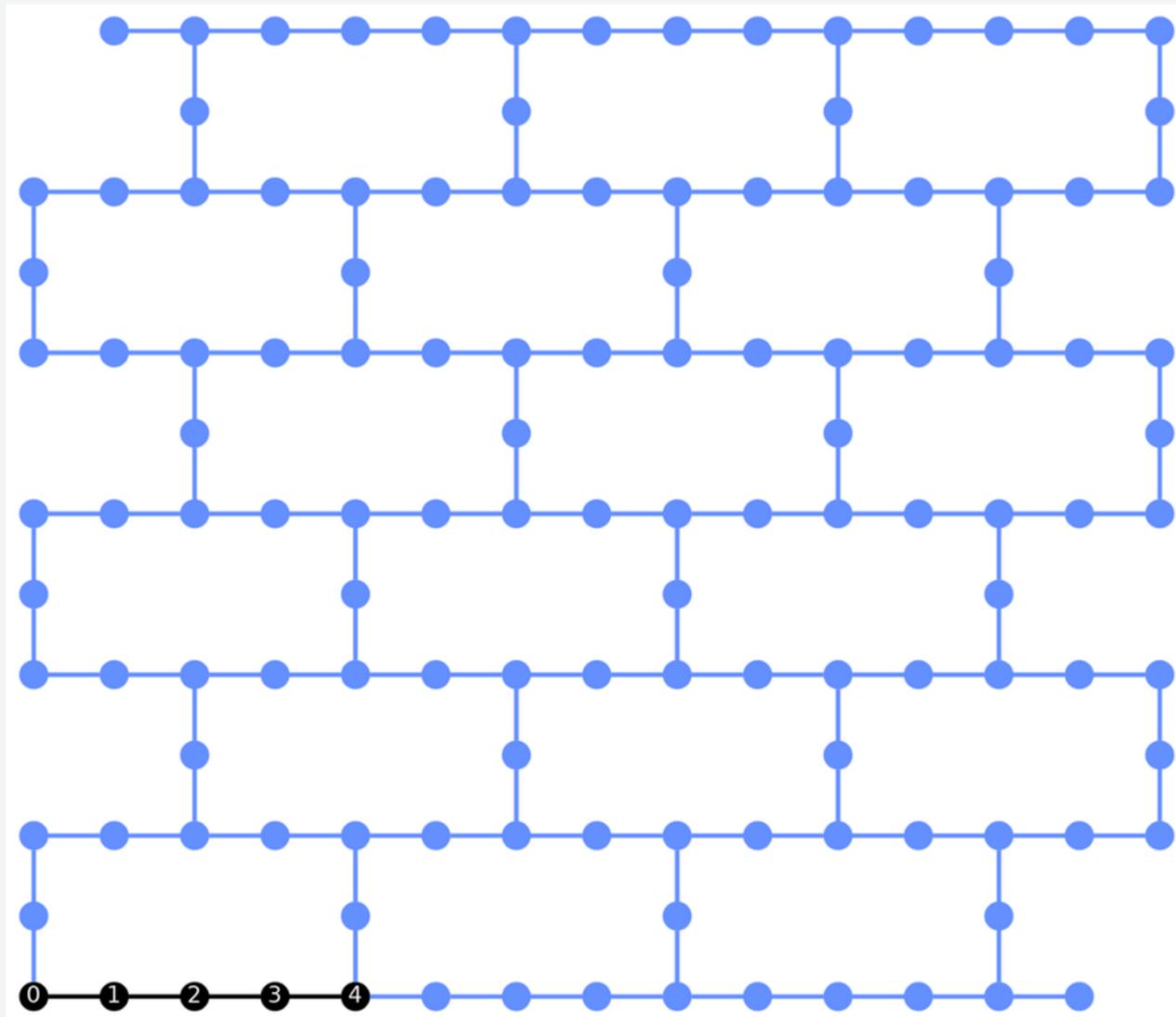
```
backend = quantum_devices_available[2]
```

```
backend
```

```
<IBMBackend('ibm_osaka')>
```

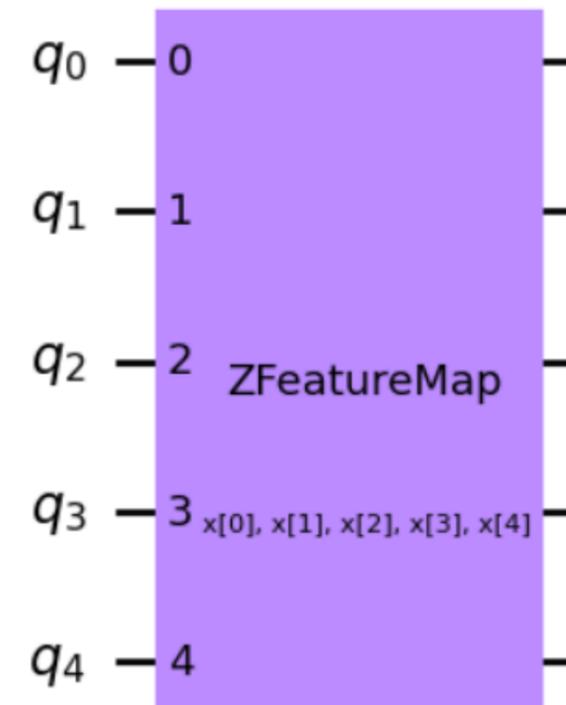
Hardware Osaka

- Utilizei outro circuito para demonstrar as otimizações.
- Os níveis de otimização
 - **Nível 0**

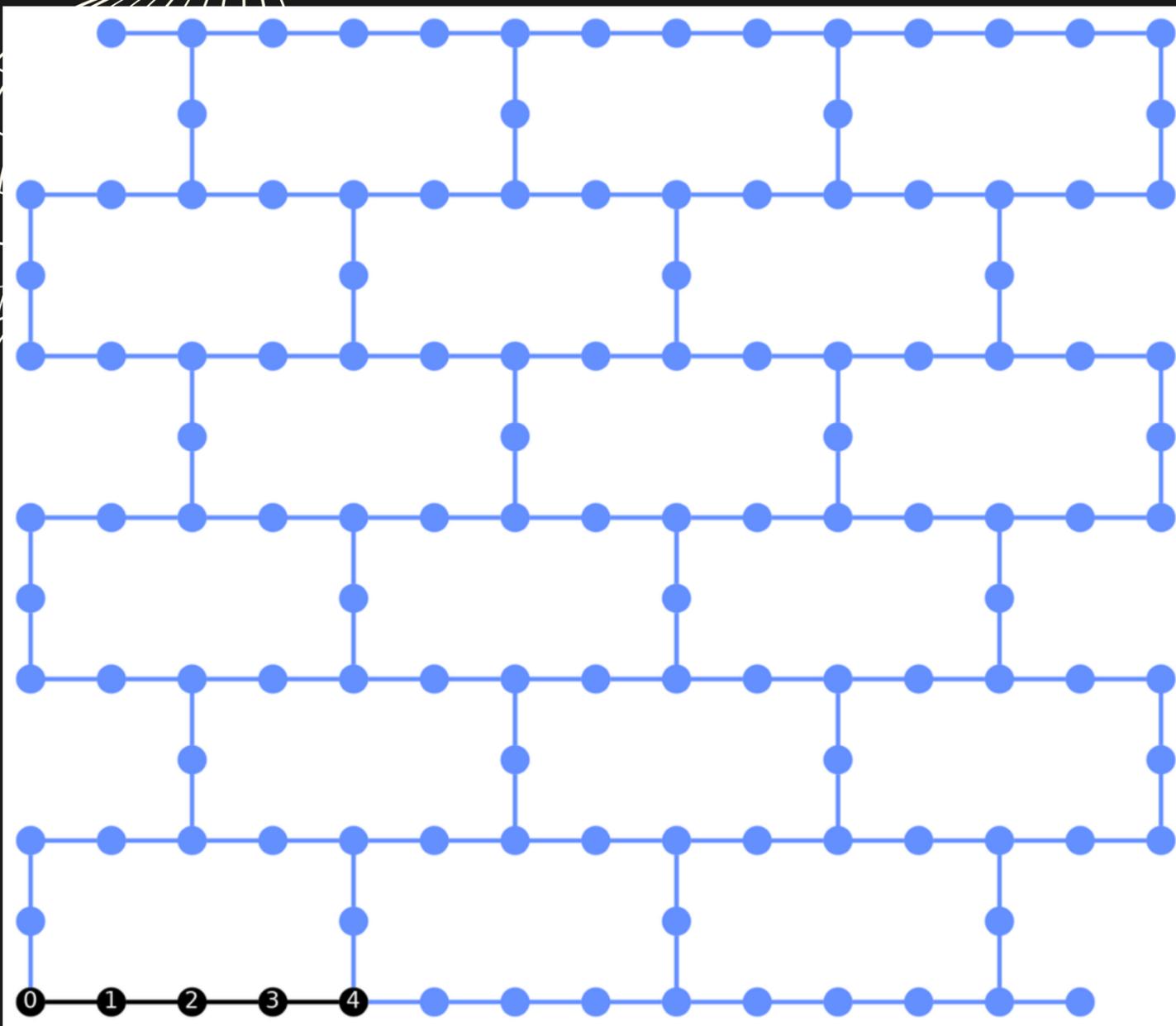


```
from qiskit.circuit.library import ZFeatureMap

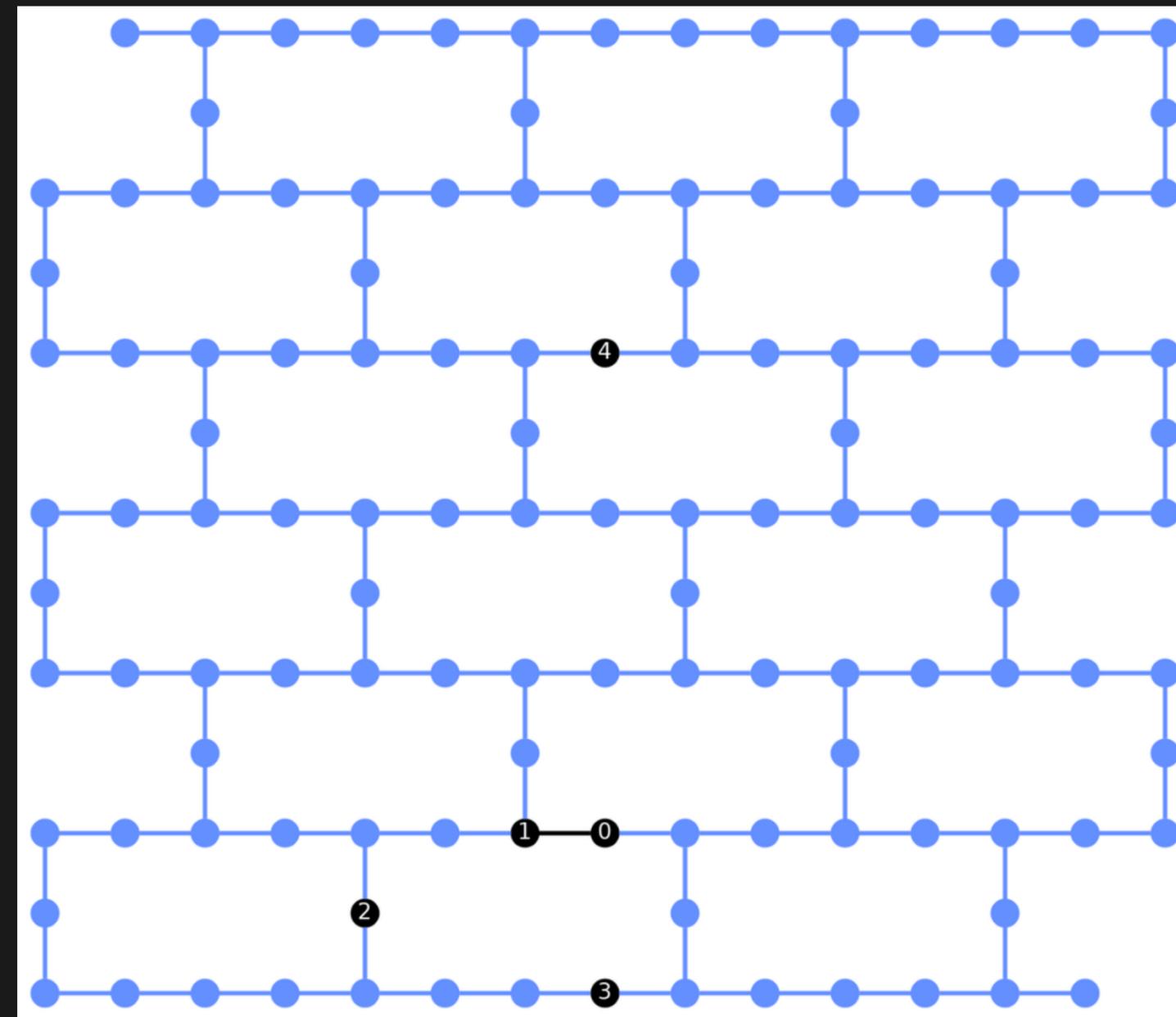
map_z = ZFeatureMap(5, reps=3, insert_barriers=True)
map_z.draw(output='mpl')
```



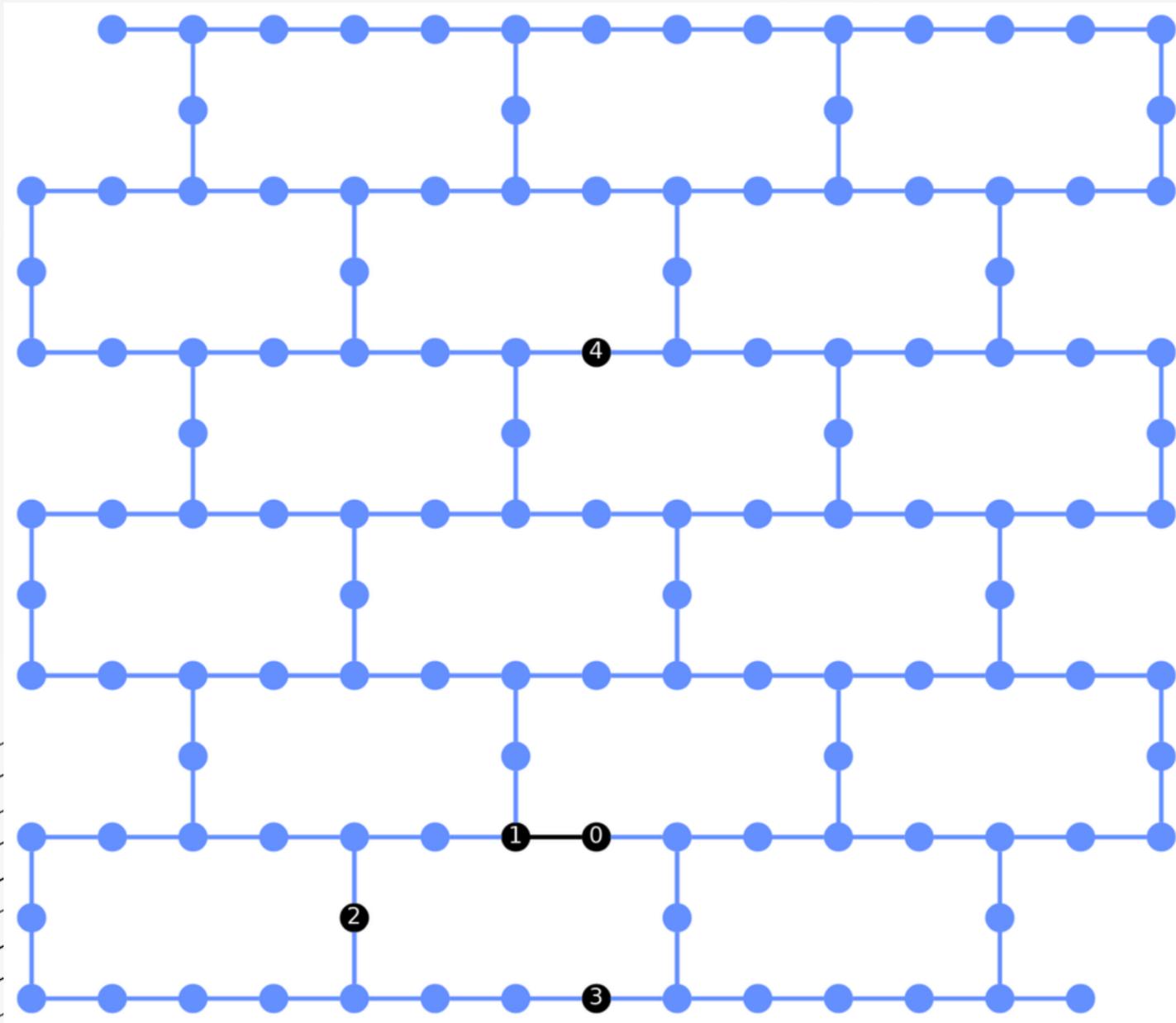
• Nível 1



• Nível 2



- **Nível 3**



Na otimização níveis mais altos geram circuitos mais otimizados, às custas de tempos de transpilação mais longos.

CONCLUSÃO

A missão da computação quântica vislumbra um mundo mais interligado e e que promova avanços em diversas esferas da existência humana.



CRIPTOGRAFIA



ÁREA DA SAÚDE



FINANCEIRA



**MUITO
OBRIGADA!!**

PERGUNTAS?

