shor

December 11, 2023

1 Algoritmo de Shor implementado en Qiskit

Para el caso N=15, A = 7 ### Autores Álvaro Cabo & Oussema El-Hatifi

1.1 Instalación de dependencias

Para poder ejecutar este proyecto, simplementes necesitamos la librería core de qiskit, con ella podremos tanto conectarnos al backend como utilizar sus algortimos built-in - Guía de easy set-up para qiskit

[ ]:
```
@rem Recomendamos crear un entrono virtual para instalar las dependencias
@rem python -m venv .venv
@rem source .venv/bin/activate (Linux)

%pip install qiskit-ibm-runtime qiskit-ibmq-provider python-dotenv matplotlib␣
↪pylatexenc
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting qiskit-ibm-runtime
  Downloading qiskit_ibm_runtime-0.17.0-py3-none-any.whl (229 kB)
                             229.7/229.7

KB 3.3 MB/s eta 0:00:00a 0:00:01
Collecting qiskit-ibmq-provider
  Downloading qiskit_ibmq_provider-0.20.2-py3-none-any.whl (241 kB)
                             241.5/241.5

KB 4.4 MB/s eta 0:00:00a 0:00:01
Requirement already satisfied: python-dotenv in /usr/lib/python3/dist-
packages (0.19.2)
Requirement already satisfied: matplotlib in
/home/varo/.local/lib/python3.10/site-packages (3.7.2)
Collecting pylatexenc
  Downloading pylatexenc-2.10.tar.gz (162 kB)
                             162.6/162.6

KB 4.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: urllib3>=1.21.1 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime) (2.0.3)
```

```
Collecting qiskit-ibm-provider>=0.7.2
  Downloading qiskit_ibm_provider-0.7.2-py3-none-any.whl (243 kB)
                             243.3/243.3
KB 4.8 MB/s eta 0:00:00a 0:00:01
Collecting qiskit>=0.44.1
  Downloading qiskit-0.45.1-py3-none-any.whl (9.6 kB)
Collecting requests-ntlm>=1.1.0
  Downloading requests_ntlm-1.2.0-py3-none-any.whl (6.0 kB)
Collecting websocket-client>=1.5.1
  Downloading websocket_client-1.7.0-py3-none-any.whl (58 kB)
                             58.5/58.5 KB
3.0 MB/s eta 0:00:00
Collecting ibm-platform-services>=0.22.6
  Downloading ibm-platform-services-0.48.0.tar.gz (300 kB)
                             300.8/300.8
KB 4.9 MB/s eta 0:00:00a 0:00:01
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Preparing metadata (pyproject.toml) … done
Requirement already satisfied: requests>=2.19 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(2.31.0)
Requirement already satisfied: python-dateutil>=2.8.0 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime) (2.8.2)
Requirement already satisfied: numpy>=1.13 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(1.25.1)
Collecting websockets>=10.0
  Using cached websockets-12.0-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (130 kB)
Collecting requests-ntlm>=1.1.0
  Downloading requests_ntlm-1.1.0-py2.py3-none-any.whl (5.7 kB)
Collecting numpy>=1.13
  Downloading
numpy-1.23.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1
MB)
                             17.1/17.1 MB
4.9 MB/s eta 0:00:0000:0100:01
Collecting qiskit-terra>=0.18.0
  Downloading
qiskit_terra-0.45.1-cp38-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(6.3 MB)
                             6.3/6.3 MB
4.1 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
```

```
/usr/lib/python3/dist-packages (from matplotlib) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages
(from matplotlib) (9.0.1)
Requirement already satisfied: fonttools>=4.22.0 in
/home/varo/.local/lib/python3.10/site-packages (from matplotlib) (4.41.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/home/varo/.local/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/home/varo/.local/lib/python3.10/site-packages (from matplotlib) (23.1)
Requirement already satisfied: cycler>=0.10 in
/home/varo/.local/lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in
/home/varo/.local/lib/python3.10/site-packages (from matplotlib) (1.1.0)
Collecting ibm-cloud-sdk-core<4.0.0,>=3.17.0
  Downloading ibm-cloud-sdk-core-3.18.1.tar.gz (57 kB)
                             57.9/57.9 KB
3.1 MB/s eta 0:00:00
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Preparing metadata (pyproject.toml) … done
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages
(from python-dateutil>=2.8.0->qiskit-ibm-runtime) (1.16.0)
Collecting ply>=3.10
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
                             49.6/49.6 KB
2.1 MB/s eta 0:00:00
Requirement already satisfied: psutil>=5 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-
terra>=0.18.0->qiskit-ibmq-provider) (5.9.6)
Collecting rustworkx>=0.13.0
  Downloading
rustworkx-0.13.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0
MB)
                             2.0/2.0 MB
4.1 MB/s eta 0:00:0000:0100:01
Collecting dill>=0.3
  Downloading dill-0.3.7-py3-none-any.whl (115 kB)
                            115.3/115.3
KB 3.4 MB/s eta 0:00:00
Collecting scipy>=1.5
  Downloading
scipy-1.11.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (36.4
MB)
                             36.4/36.4 MB
4.2 MB/s eta 0:00:0000:0100:01
Collecting symengine!=0.10.0,>=0.9
  Downloading
```

```
symengine-0.11.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
(39.4 MB)
                           39.4/39.4 MB
3.9 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: stevedore>=3.0.0 in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-
terra>=0.18.0->qiskit-ibmq-provider) (5.0.0)
Collecting sympy>=1.3
  Downloading sympy-1.12-py3-none-any.whl (5.7 MB)
                           5.7/5.7 MB
5.4 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: typing-extensions in
/home/varo/.local/lib/python3.10/site-packages (from qiskit-
terra>=0.18.0->qiskit-ibmq-provider) (4.8.0)
Requirement already satisfied: idna<4,>=2.5 in
/home/varo/.local/lib/python3.10/site-packages (from requests>=2.19->qiskit-ibm-
runtime) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/varo/.local/lib/python3.10/site-packages (from requests>=2.19->qiskit-ibm-
runtime) (3.1.0)
Requirement already satisfied: certifi>=2017.4.17 in
/home/varo/.local/lib/python3.10/site-packages (from requests>=2.19->qiskit-ibm-
runtime) (2023.5.7)
Collecting ntlm-auth>=1.0.2
  Downloading ntlm_auth-1.5.0-py2.py3-none-any.whl (29 kB)
Requirement already satisfied: cryptography>=1.3 in /usr/lib/python3/dist-
packages (from requests-ntlm>=1.1.0->qiskit-ibm-runtime) (3.4.8)
Collecting PyJWT<3.0.0,>=2.8.0
  Downloading PyJWT-2.8.0-py3-none-any.whl (22 kB)
Collecting urllib3>=1.21.1
  Downloading urllib3-2.1.0-py3-none-any.whl (104 kB)
                           104.6/104.6
KB 3.1 MB/s eta 0:00:00
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in
/home/varo/.local/lib/python3.10/site-packages (from stevedore>=3.0.0->qiskit-
terra>=0.18.0->qiskit-ibmq-provider) (5.11.1)
Collecting mpmath>=0.19
  Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
                           536.2/536.2
KB 4.8 MB/s eta 0:00:0000:0100:01
Building wheels for collected packages: pylatexenc, ibm-platform-services,
ibm-cloud-sdk-core
  Building wheel for pylatexenc (setup.py) … done
  Created wheel for pylatexenc: filename=pylatexenc-2.10-py3-none-any.whl
size=136835
sha256=04000fd8ef9f2160d4b4dbcd5323eeaee64b4636a947aaeb1c9d6e2dbabccb56
```

```
  Stored in directory: /home/varo/.cache/pip/wheels/d3/31/8b/e09b0386afd80cfc556
c00408c9aeea5c35c4d484a9c762fd5
  Building wheel for ibm-platform-services (pyproject.toml) … done
  Created wheel for ibm-platform-services:
filename=ibm_platform_services-0.48.0-py3-none-any.whl size=315032
sha256=3d497e8602428c0421b054779e7a75141e59b19973cd49307e814a498eee708e
  Stored in directory: /home/varo/.cache/pip/wheels/61/c9/b2/97eaa297a6e49d28a10
d90046f9d6e884f9ac75a72d3d5bb41
  Building wheel for ibm-cloud-sdk-core (pyproject.toml) … done
  Created wheel for ibm-cloud-sdk-core:
filename=ibm_cloud_sdk_core-3.18.1-py3-none-any.whl size=94343
sha256=c574af82e6557c7d46f363f0c63c55625a4af82c77df6de86c9792424c531178
  Stored in directory: /home/varo/.cache/pip/wheels/e7/bc/63/a250bb561941180a2b8
666bdb5bc67c67f537a660790e8901c
Successfully built pylatexenc ibm-platform-services ibm-cloud-sdk-core
Installing collected packages: pylatexenc, ply, mpmath, websockets, websocket-
client, urllib3, sympy, symengine, PyJWT, numpy, ntlm-auth, dill, scipy,
rustworkx, requests-ntlm, qiskit-terra, ibm-cloud-sdk-core, qiskit-ibmq-
provider, qiskit, ibm-platform-services, qiskit-ibm-provider, qiskit-ibm-runtime
  Attempting uninstall: urllib3
    Found existing installation: urllib3 2.0.3
    Uninstalling urllib3-2.0.3:
      Successfully uninstalled urllib3-2.0.3
  Attempting uninstall: numpy
    Found existing installation: numpy 1.25.1
    Uninstalling numpy-1.25.1:
      Successfully uninstalled numpy-1.25.1
Successfully installed PyJWT-2.8.0 dill-0.3.7 ibm-cloud-sdk-core-3.18.1 ibm-
platform-services-0.48.0 mpmath-1.3.0 ntlm-auth-1.5.0 numpy-1.23.5 ply-3.11
pylatexenc-2.10 qiskit-0.45.1 qiskit-ibm-provider-0.7.2 qiskit-ibm-
runtime-0.17.0 qiskit-ibmq-provider-0.20.2 qiskit-terra-0.45.1 requests-
ntlm-1.1.0 rustworkx-0.13.2 scipy-1.11.4 symengine-0.11.0 sympy-1.12
urllib3-2.1.0 websocket-client-1.7.0 websockets-12.0
Note: you may need to restart the kernel to use updated packages.
```

1.2 Selección de backend

Para poder ejecutar nuestro código en un ordenador cuántico, hacemos uso del *Cloud Quantum Computing* > IBM provides tools for a beginner to learn and use the quantum computer using the visual programming tool that converts the codes into interactable nodes and still provide the user the code that they make using the visual programming[14]. While Qutech's quantum computer only provides the user the tool to write the code for the quantum computer

Además contamos con una prueba gratuita de 10 minutos

[ ]:
```python
import os
from dotenv import load_dotenv
```

```
load_dotenv('.env')

from qiskit import *
from math import pi, gcd

try:
    IBMQ.enable_account(os.environ.get("API_KEY"))
except:
    print("The acount has already been enabled")

provider = IBMQ.get_provider(hub='ibm-q')

backend = provider.get_backend('ibmq_qasm_simulator')
```

```
C:\Users\34642\AppData\Local\Temp\ipykernel_19608\46384157.py:11:
DeprecationWarning: The package qiskit.providers.ibmq is being deprecated.
Please see https://ibm.biz/provider_migration_guide to get instructions on how
to migrate to qiskit-ibm-provider (https://github.com/Qiskit/qiskit-ibm-
provider) and qiskit-ibm-runtime (https://github.com/Qiskit/qiskit-ibm-runtime).
  IBMQ.enable_account(os.environ.get("API_KEY"))
C:\Users\34642\AppData\Local\Temp\ipykernel_19608\46384157.py:11:
DeprecationWarning: The qiskit.IBMQ entrypoint and the qiskit-ibmq-provider
package (accessible from 'qiskit.providers.ibmq`) are deprecated and will be
removed in a future release. Instead you should use the qiskit-ibm-provider
package which is accessible from 'qiskit_ibm_provider'. You can install it with
'pip install qiskit_ibm_provider'. Just replace 'qiskit.IBMQ' with
'qiskit_ibm_provider.IBMProvider'
  IBMQ.enable_account(os.environ.get("API_KEY"))
```

1.3 Implementación de la Transformada Cuántica de Fourier

Implementa físicamente en los cubits su valor en base computacional, no hace ningún cómputo, solo "traslada" la información a un dominio donde puede operar más fácil

Video divulgación QFT

[ ]:
```
def QFT(n):
    qft_circ = QuantumCircuit(n) # Colección de n qubits
    for i in range(n-1, -1, -1): # from n-1 to 0
        qft_circ.h(i) # Qubit en estado de superposición
        for j in range(i - 1, -1, -1):
            # Hacemos el CU1 con todos los qubits que quedan por encima
            theta = pi/(2 ** (i - j)) # Angulo de rotación
            qft_circ.cu(theta, 0, 0, 0, j, i)
    for i in range(n // 2): # Damos la vuelta a los valores
        qft_circ.swap(i, n - i - 1)
    # Convertimos el circuito en una puerta para poder integrarlo en nuestro␣
 ↪circuito de Shor
```

```
    gate = qft_circ.to_gate(label="QFT" + str(n))
    return gate
```

1.4 Puerta U(f)

Implementamos la puerta que realiza la operación `|x0> -> |xf(x)>`

Para nuestro caso, esta puerta realiza la operación `7 mod(15)` en el circuito

[ ]:
```
def _7mod15():
    circ = QuantumCircuit(8) # Colección de 8 qubits
    circ.x(4) # Flipeamos el qubit 4
    # CNOTs -> Flipeamos si el qubit de control (1º) es |1>
    circ.cx(0,5)
    circ.cx(0,6)
    circ.cx(1,4)
    circ.cx(1,6)
    for i in range(4,8):
        # CCNOTs -> Flipeamos si los dos qubits de control son |1>
        circ.ccx(0,1,i)
    gate = circ.to_gate(label="7mod15")
    return gate
```

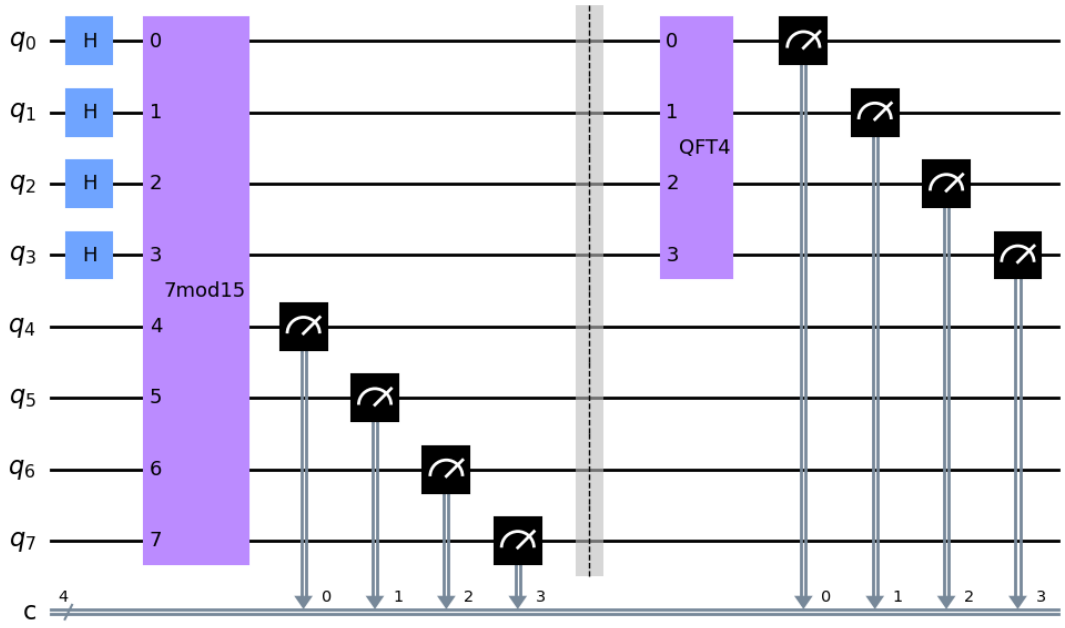
1.5 Creando el circuito -> Encontrar el periodo

Implementamos la parte cuántica del algoritmo según se describe: 1. Ponemos los 4 primeros qubits en estado de superposición 2. Ejecutamos la función `f -> Uf` (Multiplicación modular) para encontrar los posibles pares de resultados 3. Medimos los 4 últimos qubits sin que nos importe el resultado 4. Eliminamos el residuo |x0> utilizando la QFT 5. El valor al que colapse la medición de los 4 qubits restantes será una aproximación racional de `15/t`, por lo que podríamos despejar t

[ ]:
```
circ = QuantumCircuit(8,4)
circ.h(range(4)) # Ponemos los 4 primeros qubits en estado de superposición
circ.append(_7mod15(), range(8)) # Añadimos el circuito de 7mod15
circ.measure(range(4,8),range(4)) # Medimos los 4 últimos qubits sin que nos␣
 ↪importe el resultado
circ.barrier(range(8)) # Añadimos una barrera para separar los dos circuitos
circ.append(QFT(4), range(4)) # Añadimos el circuito de QFT
circ.measure(range(4), range(4)) # Medimos los 4 primeros qubits
circ.draw(output = 'mpl')
```

```
c:\Users\34642\AppData\Local\Programs\Python\Python311\Lib\site-
packages\qiskit\visualization\circuit\matplotlib.py:266: FutureWarning: The
default matplotlib drawer scheme will be changed to "iqp" in a following
release. To silence this warning, specify the current default explicitly as
style="clifford", or the new default as style="iqp".
  self._style, def_font_ratio = load_style(self._style)
```

[ ]:



1.6 Ejecutando el código en el back

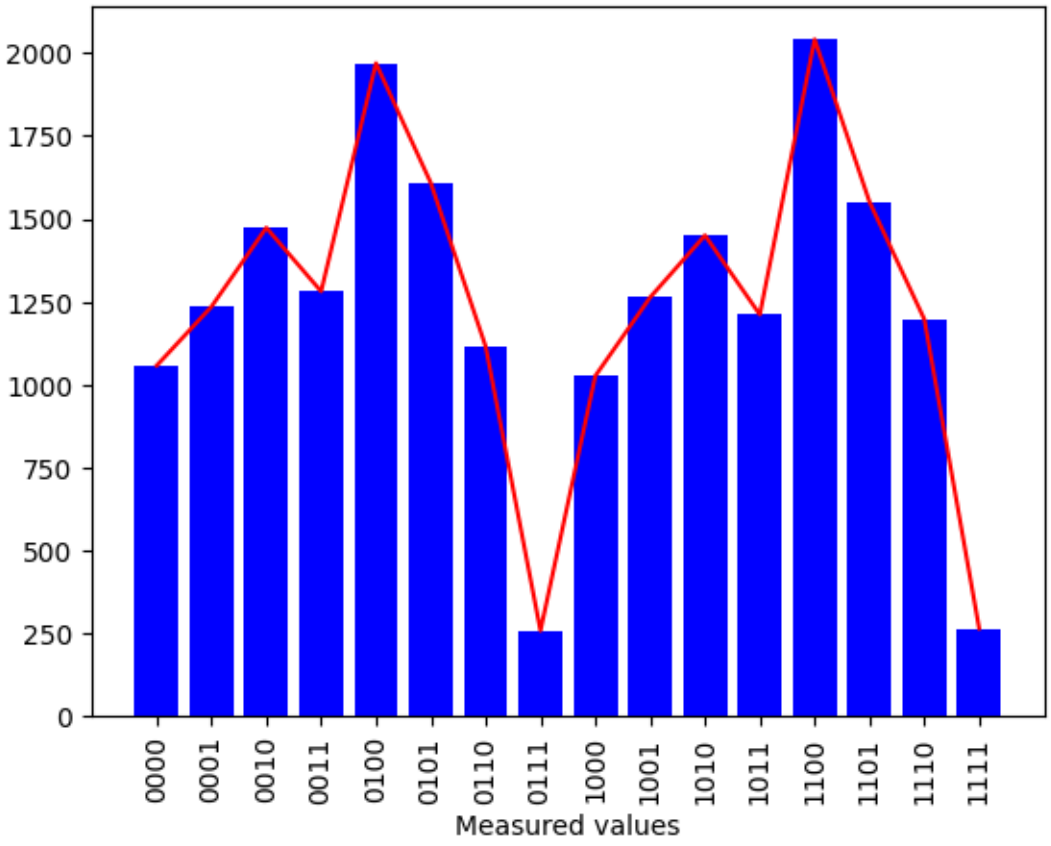
Con este resultado, obtendremos el periodo de la función

[ ]:
```
job = execute(circ, backend, shots=20000)
result = job.result()
counts = result.get_counts()

import matplotlib.pyplot as plt
sorted_counts = {k: counts[k] for k in sorted(counts.keys())}  # Sort the keys
plt.bar(sorted_counts.keys(), sorted_counts.values(), color='b')
plt.plot(sorted_counts.keys(), sorted_counts.values(), 'r-')

# Increase the space between bar key labels
plt.xticks(rotation=90)

plt.xlabel('Measured values')
plt.show()
```



1.7 Analizando los resultados

Utilimos los convergentes de la fracción continua para obtener aproximaciones racionales p/q del periodo r.

> Para encontrar valor en estos resultados, tenemos que tener en cuenta que la onda que recibimos no tiene forma sinusoidal, por lo que contiene armónicos (Valores residuales).

Aceptamos 4 y 12 como posibles soluciones, porque `7^4 mod(15) =1` y como 12 == 0 mod4, entonces podemos concluir que, siendo p*q=15:

[ ]:
```
periodo = [4,12]
resultado = -1
for i in periodo:
    p = gcd(i-1, 15)
    q =  gcd(i+1, 15)
    if p*q == 15:
        resultado = i
        break
print(f"El periodo de la funcion es {resultado}")
print(f"Factores primos: p = {p}, q = {q}")
```

```
El periodo de la funcion es 4
Factores primos: p = 3, q = 5
```