Algoritmo de Grover para el caso de 3 qubits

Javier E. Salas Catonga
Facultad de Ciencias, Universidad Nacional Autónoma de Mexico (UNAM)
Contacto: javieremilio@ciencias.unam.mx

Junio 2020

Resumen

En este trabajo se construye el algoritmo de Grover para el caso de n=3 qubits, donde la inicialización de cada qubit es cero. El algoritmo de Grover es una herramienta poderosa e importante en el computo cuántico ya que permite encontrar con alta probabilidad un elemento en una secuencia no ordenada de N datos (en este caso $N = 8$) con un tiempo del orden de $\sqrt{N}$, en notación $\mathcal{O}(\sqrt{N})$, consta principalmente de dos compuertas, U_w y U_s, que se iteran k veces a los qubits. La implemetación aquí presente se desarrolló con la herramienta *Qiskit* en Python 3, y se obtuvieron probabilidades de alrededor de 0.92 y 0.95 para cada uno de los estados cuánticos posibles dados 3 qubits, dichas probalidades se obtuvieron haciendo $k = 2$ iteraciones , lo cual comprueba que k debe ser del orden de $\mathcal{O}(\sqrt{N})$.

1. Introducción

El algoritmo de Grover es uno de los principales algoritmos de la computación cuántica y es uno de los más básico, el cual explota al máximo el principio fundamental de la superposición de estados cuánticos, mostrando la superioridad de las computadoras cuánticas sobre las computadoras clásicas. Este algoritmo se conoce por su poder de búsqueda sobre un conjunto de datos no estructurados, es decir, imaginemos que tenemos un conjunto de N datos, pero necesitamos encontrar uno de ellos en específico, si realizamos la acción de búsqueda sobre estos N datos con un algoritmo clásico debería buscarlo en $\mathcal{O}(N)$ operaciones. En un sistema cuántico dada la superposición de estados, se puede resolver examinando simultáneamente todas las posibles combinaciones. Como resultado, el estado que deseamos encontrar se puede obtener en solo $\mathcal{O}(\sqrt{N})$ pasos.

2. Marco teórico

2.1 Compuertas

Sea $\{|0\rangle, |1\rangle >, ... |N-1\rangle\}$ un conjunto de N estados cuanticos, supongamos que queremos extraer un estado específico de ese conjunto, llamemos a este elemento $|w\rangle$.

Sea $|s\rangle$ la superposicion de todos los estados:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N-1} |i\rangle \tag{1}$$

Sea $|s'\rangle$ la superposicón de todos los estados excepto el elemento $|w\rangle$

$$|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \neq w} |i\rangle \tag{2}$$

Definimos la compuerta U_s como

$$U_s = 2\,|s\rangle \langle s| - I \tag{3}$$

y a la compuerta U_w tal que

$$U_w = I - 2\,|w\rangle \langle w| \tag{4}$$

Notamos que

$$U_w\,|i\rangle = |i\rangle - 2\,|w\rangle \langle w|i\rangle \tag{5}$$

donde claramente $\langle w|i\rangle = \delta_{iw}$

Así

$$U_w\,|i\rangle = \begin{cases} |i\rangle\,, & \text{if } i \neq w \\ -\,|i\rangle\,, & \text{if } i = w \end{cases} \tag{6}$$

Es decir, la compuerta U_w le cambia el signo solo al estado $|w\rangle$, que es el estado que queremos extraer, dejando igual a todos los demás estados.

Por otra parte, podemos expresa $|s\rangle$ en términos de $|w\rangle$ y de $|s'\rangle$ como

$$|s\rangle = \frac{1}{\sqrt{N}}\,|w\rangle + \sqrt{\frac{N-1}{N}}\,|s'\rangle \tag{7}$$

Sea θ el ángulo tal que $\langle s'|s\rangle = cos(\frac{\theta}{2})$ en el plano formado por los vectores $|w\rangle$ y $|s'\rangle$, los cuales son ortogonales. De la ecuación anterior tenemos que

$$cos(\frac{\theta}{2}) = \sqrt{\frac{N-1}{N}} \tag{8}$$

De donde

$$cos^2(\frac{\theta}{2}) = 1 - \frac{1}{N} \tag{9}$$

Despejando θ

$$\theta = 2arcsin(\frac{1}{\sqrt{N}}) \tag{10}$$

Por tanto, podemos escribir $|s\rangle$ como

$$|s\rangle = sin(\frac{\theta}{2})\,|w\rangle + cos(\frac{\theta}{2})\,|s'\rangle \tag{11}$$

2.2 El algoritmo

El algoritmo consiste en alplicar las dos compuertas definidas anteriormente al vector $|s\rangle$. En primer lugar, se aplica la compuerta U_w, dadas las expresiones (6) y (11) se obtiene solamente un cambio de signo en el coeficiente de $|w\rangle$

$$U_w\,|s\rangle = -sin(\frac{\theta}{2})\,|w\rangle + cos(\frac{\theta}{2})\,|s'\rangle \tag{12}$$

Es decir la compuerta U_w refleja al vector $|s\rangle$ con respecto al vector $|s'\rangle$

Posteriormente se aplica la compuerta U_s. Tomando la expresión (3) y aplicandola a la expresion anterior tenemos

$$U_sU_w\,|s\rangle = -(sin(\frac{\theta}{2})\,|w\rangle + cos(\frac{\theta}{2})\,|s'\rangle) + 2cos^2(\frac{\theta}{2})\,|s\rangle \tag{13}$$

Lo cual se simplifica a

$$U_sU_w\,|s\rangle = cos(\theta)\,|s\rangle \tag{14}$$

Es decir el angulo entre los vectores $|s\rangle$ y $U_sU_w\,|s\rangle$ es θ lo que conlleva a concluir que la compuerta U_s realiza una reflexión al vector $U_w\,|s\rangle$ con respecto a $|s\rangle$.

Para visualizar geométricamente lo que está pasando, consideremos el siguiente diagrama del plano formado por los vectores linealmente independientes $|s'\rangle$ y $|w\rangle$.

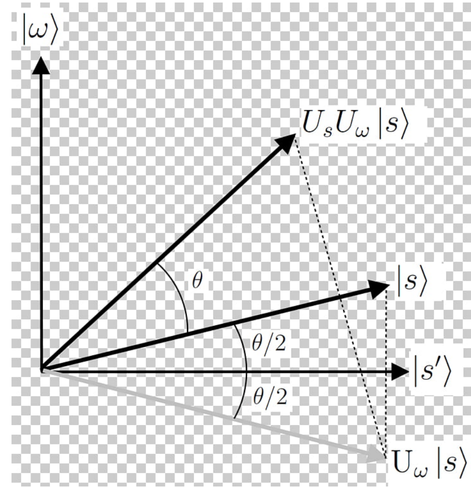

Figura 1: *Visualización geométrica del algoritmo de Grover*

Al aplicar la compuerta U_w al vector $|s\rangle$, el cual inicialmente forma un ángulo $\frac{\theta}{2}$ con el eje $|s'\rangle$, el coeficiente de $|w\rangle$ cambia de signo mientras que el de $|s'\rangle$ permanece invariante, es decir, existe una reflexión respecto al eje $|s'\rangle$. Luego, al aplicar la compuerta U_s se realiza una reflexión del vector $U_w\,|s\rangle$ con respecto a $|s\rangle$. Notemos que el estado final $U_sU_w\,|s\rangle$ no es más que una rotación del estado inicial $|s\rangle$ por un ángulo θ y se encuentra más cerca del estado $|w\rangle$ que este último.

De lo anterior, se tiene que el estado final es

$$U_sU_w\,|s\rangle = sin(\frac{3\theta}{2})\,|w\rangle + cos(\frac{3\theta}{2})\,|s'\rangle \tag{15}$$

Al aplicar nuevamente las compuertas U_w y U_s al estado final anterior, volverá a rotar un ángulo θ aproximandose más a $|w\rangle$.

$$(U_sU_w)^2\,|s\rangle = sin((2+\frac{1}{2})\theta)\,|w\rangle + cos((2+\frac{1}{2})\theta)\,|s'\rangle \tag{16}$$

Si aplicamos este procedimiento k veces obtenemos

$$(U_sU_w)^k\,|s\rangle = sin((k+\frac{1}{2})\theta)\,|w\rangle + cos((k+\frac{1}{2})\theta)\,|s'\rangle \tag{17}$$

Por otro lado, la probabilidad de obtener el estado $|w\rangle$ está dada por

$$P_{w,k} = |\,\langle w|\psi_{w,k}\rangle\,|^2 \tag{18}$$

Donde

$$\psi_{w,k} = (U_sU_w)^k\,|s\rangle = sin((k+\tfrac{1}{2})\theta)\,|w\rangle + cos((k+\tfrac{1}{2})\theta)\,|s'\rangle$$

Para que el resultado sea lo más proximo a $|w\rangle$ necesitamos que $P_{w,k} \approx 1$, es decir

$$(k + \frac{1}{2})\theta \approx \frac{\pi}{2} \tag{19}$$

De la ec. (10) tenemos que $\theta = 2arcsin(\frac{1}{\sqrt{N}})$, despejando k

$$k = \frac{\pi}{4arcsin(\frac{1}{\sqrt{N}})} - \frac{1}{2} \tag{20}$$

Por tanto, el número de iteraciones k debe ser del orden de $\mathcal{O}(\sqrt{N})$.

3. Implementación con Qiskit

Para el caso de tres qubits, es decir, estados de la forma $|w\rangle = |q_2 q_1 q_0\rangle$, donde cada $q_i \in \{0, 1\}$, tenemos una lista de $N = 8$ posibilidades de w: 000,001,010, 011,100,110,101 y 111. Para la implementación de las compuertas necesarias para el algoritmo, definidas en la sección anterior, se utilizó la paquetería Qiskit en Python 3.

3.1 Compuerta U_w

Dado que la compuerta U_w depende del estado que se quiere extraer $|w\rangle$ se contruyó una compuerta por cada estado posible. Recordando que la única función de esta compuerta es cambiar el signo del estado $|w\rangle$, por lo tanto se utilizó una compuerta $CC - Z$ el cual mediante solo una combinación exacta de los 3 qubits permitirá el cambio de signo , dicha combinación será el estado que se quiere extraer. Dado que Qiskit no cuenta con la compuerta CC-Z, entonces se usó la identidad que se muestra debajo en la Fig. 2 para construir la compuerta

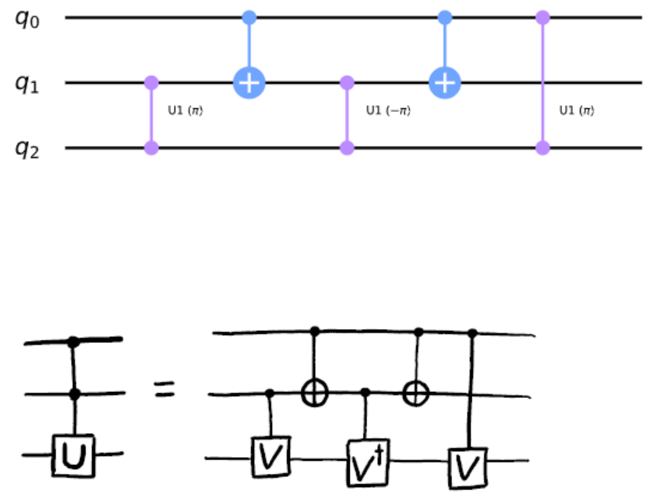

Figura 2: *Identidad CC-Z*

Donde $U = Z$ y $V = \sqrt{Z}$, que practicamente es una compuerta de desplazamiento de fase de $\varphi = \pi/2$ para V y $\varphi = -\pi/2$ para $V^\dagger$.

Además, como en Qiskit la inicialización de los qubits siempre es cero, es neceario preparar el estado $|w\rangle$ deseado, por tanto se implementan las compuertas X dependiendo el estado que en el que se deasea el cambio de signo.

Los circuitos que definen U_w para cada $|w\rangle$ se listan debajo

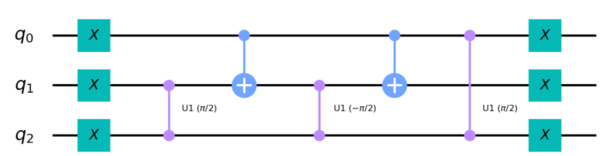

Figura 3: *Compuerta U_w para $|w\rangle = |000\rangle$*

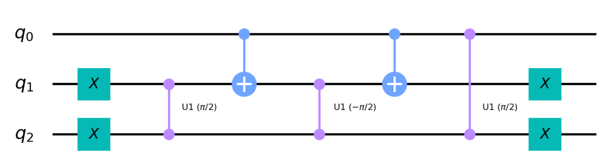

Figura 4: *Compuerta U_w para $|w\rangle = |001\rangle$*

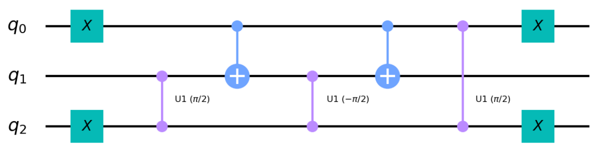

Figura 5: *Compuerta* U_w *para* $|w\rangle = |010\rangle$

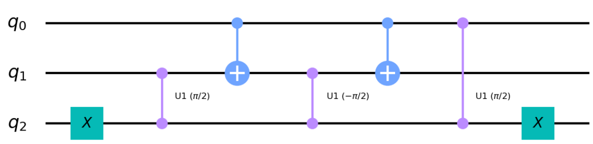

Figura 6: *Compuerta* U_w *para* $|w\rangle = |011\rangle$

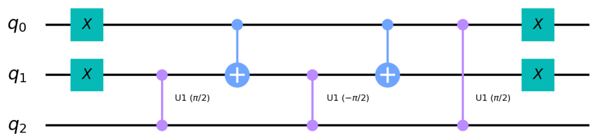

Figura 7: *Compuerta* U_w *para* $|w\rangle = |100\rangle$

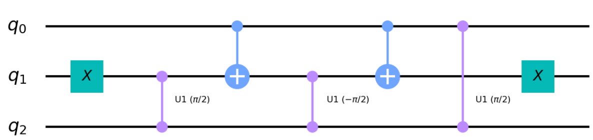

Figura 8: *Compuerta* U_w *para* $|w\rangle = |101\rangle$

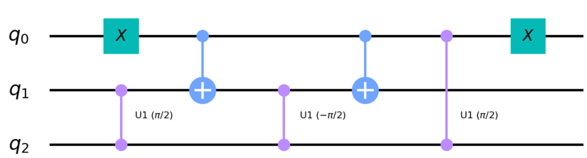

Figura 9: *Compuerta* U_w *para* $|w\rangle = |110\rangle$

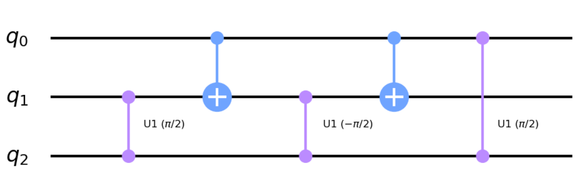

Figura 10: *Compuerta* U_w *para* $|w\rangle = |111\rangle$

3.2 Compuerta U_s e iteraciones

Por otro lado, para implementar la compuerta U_s es necesario considerar las funciones Hadamard en la siguiente identidad (para el caso de n=3 qubits):

$$|s\rangle = H \otimes H \otimes H \,|000\rangle \tag{21}$$

Recordando que el operador Hadamard es un operador unitario ($H^2 = I$), podemos expresar la compuerta U_s como

$$U_s = -H \otimes H \otimes H(I - 2\,|000\rangle\,\langle 000|)H \otimes H \otimes H \tag{22}$$

Lo que está entre paréntesis no es más que la compuerta U_w para el caso $|w\rangle = |000\rangle$, llamemos a esta compuerta U_{000}. Dado que al realizar la medición la fase global no es de importancia, podemos despreciarla, por tanto podemos eliminar el signo negativo de la expresion, obteniendo

$$U_s = H \otimes H \otimes H(U_{000})H \otimes H \otimes H \tag{23}$$

Así, la implementación en Qiskit queda como

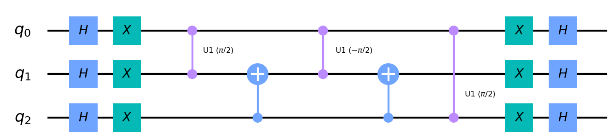

Figura 11: *Compuerta* U_s

Notemos que el circuito de la compuerta U_{000} de la figura 3 y de la Fig. 11 son equivalentes.

El número de iteraciones viene dado por la conclusion de que k debe ser del orden $\mathcal{O}(\sqrt{N})$, en este caso N=8, por tanto $K \approx 2$. Los circuitos finales para cada estado $|w\rangle$ se encuentran en el anexo final.

4. Resultados y Conclusión

Los resultados al realizar las mediciones arrojan una probabilidad entre 0.92 y 0.94 con precisión del orden de 10^{-3} para cada uno de los estados, tal y como se puede ver en los histogramas de la siguiente página. El número de iteraciones k es sumamente importante, ya que la rotación puede pasarse del estado que se desea extraer, o si son muy pocas, la rotación no puede ser suficiente y la aproximación puede ser no muy buena. Esto se pude observar en las ultimos histogramas, donde se varió k= 1,2,3 y 4, para el estado w=010.

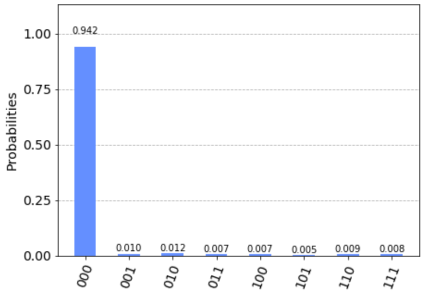

Figura 12: *Histograma para* $|000\rangle$

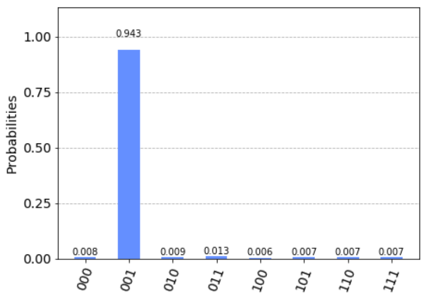

Figura 13: *Histograma para* $|001\rangle$

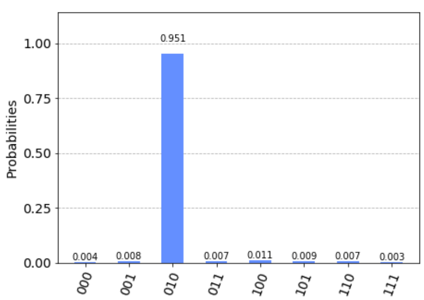

Figura 14: *Histograma para* $|010\rangle$

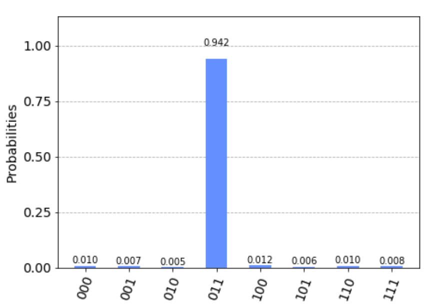

Figura 15: *Histograma para* $|011\rangle$

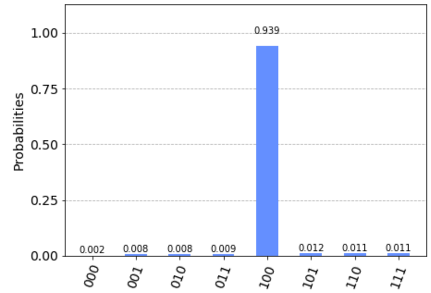

Figura 16: *Histograma para* $|100\rangle$

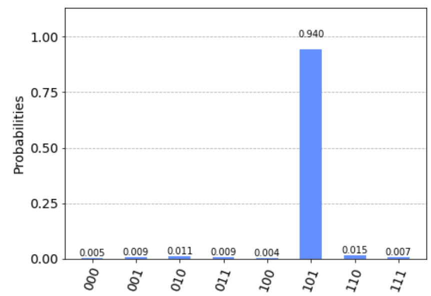

Figura 17: *Histograma para* $|101\rangle$

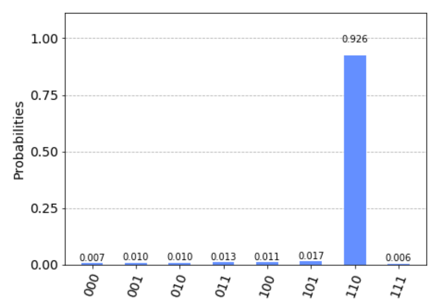

Figura 18: *Histograma para* $|110\rangle$

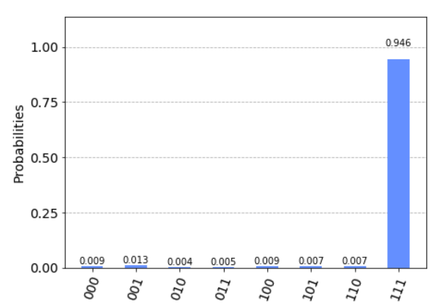

Figura 19: *Histograma para* $|111\rangle$

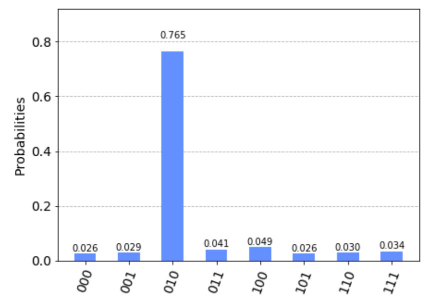

Figura 20: *Histograma para* $|010\rangle$ con k=1

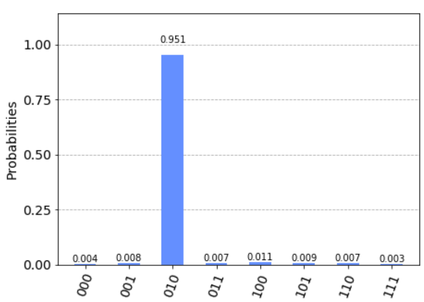

Figura 21: *Histograma para* $|010\rangle$ con k=2

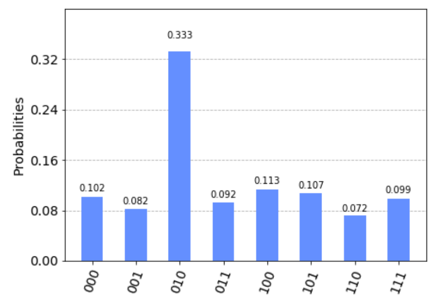

Figura 22: *Histograma para* $|010\rangle$ con k=3

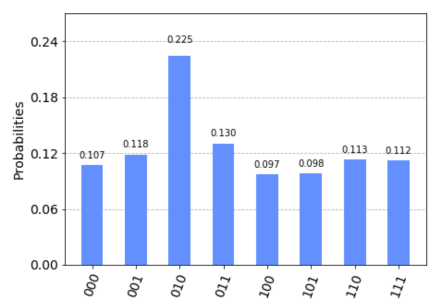

Figura 23: *Histograma para* $|010\rangle$ con k=4

Como se observa, la probabilidad decrece conforme se aleja del valor $k = 2$ de ambos lados, verificando así el orden de $\sqrt{N}$, la cual es la principal carácterística de este algoritmo y exhibe la ventaja que el cómputo cuático tiene sobre el cómputo clásico. Finalmente, el algoritmo de Grover puede generalizarse para n qubits, técnicamente el problema se reduce a econtrar las 2^n compuertas U_w ya que la compuerta U_s depende solo de una de ellas, como se mostró en la sección 3.2.

Referencias

[1] ”Notas Computación Cuática”. Luis Fernando Quezada Mata. Facultad de Ciencias UNAM, Cd. Universitaria, México 2020.

[2] ”Implementación del algoritmo de Shor y de Grover en el comuputador cuántico del IBM”. César Augusto Vega Fernández, Johan Sebastián Ramirez Celis. Escuela Colombiana de Ingenería Julio Garavito, Bogotá D.C. 2017.

[3] ”Quantum computing : from linear algebra to physical realizations”.M. Nakahara and Tetsuo Ohmi.CRC Press Taylor Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742.

Anexo: Algoritmo de Grover (circuitos finales)

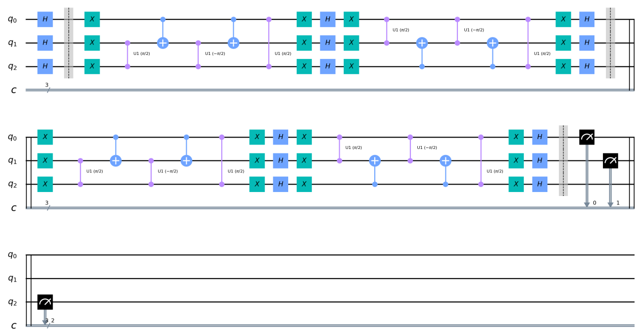

Figura 24: *Algoritmo de Grover, circuito final w=000*

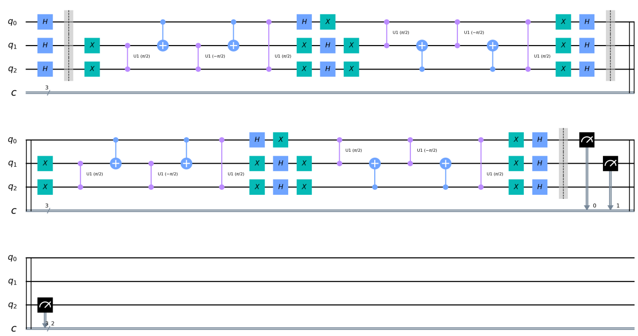

Figura 25: *Algoritmo de Grover, circuito final w=001*

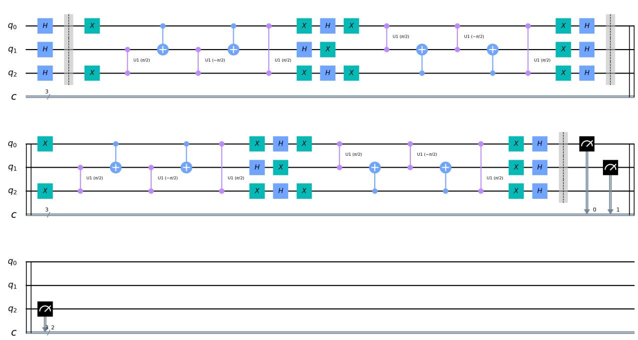

Figura 26: *Algoritmo de Grover, circuito final w=010*

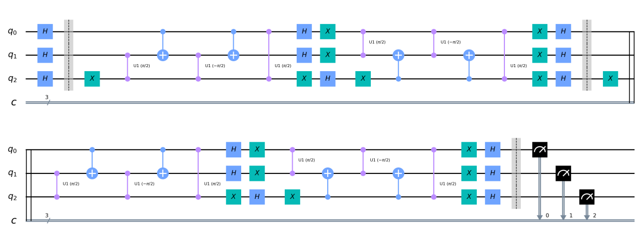

Figura 27: *Algoritmo de Grover, circuito final w=011*

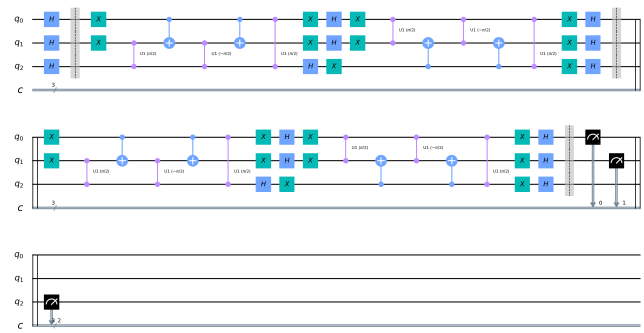

Figura 28: *Algoritmo de Grover, circuito final w=100*

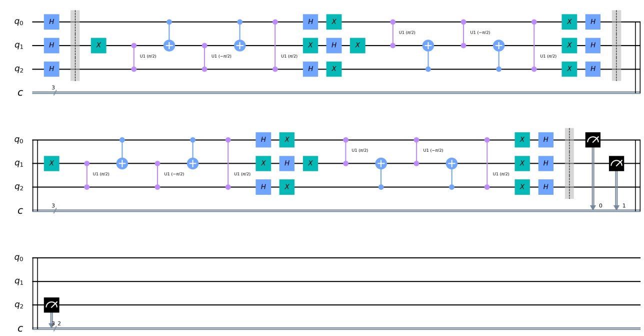

Figura 29: *Algoritmo de Grover, circuito final w=101*

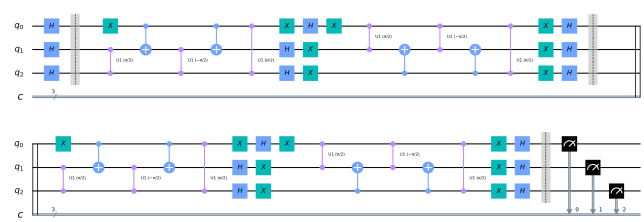

Figura 30: *Algoritmo de Grover, circuito final w=110*

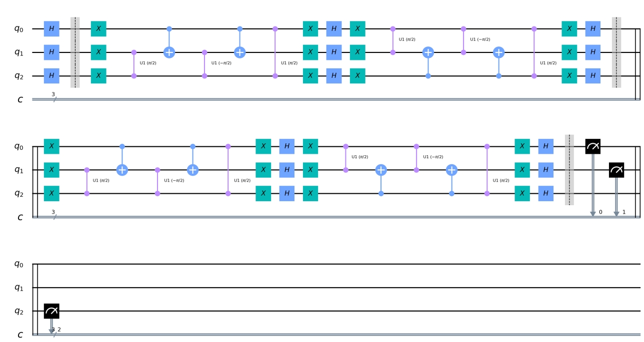

Figura 31: *Algoritmo de Grover, circuito final w=111*