Implementación del Algoritmo de Deutsch y Deutsch-Jozsa

**Juan David Martínez**

**Escuela Colombiana de Ingeniería Julio Garavito**

**Juan.mmendez@mail.escuelaing.edu.co**

**19/11/2024**

*Este reporte se entrega para cumplir con los requisitos parciales del curso CNYT: Computación Cuántica- 2020-1*

# Tabla de contenidos

[Tabla de contenidos 1](#_Toc39509128)

[1 Introducción 1](#_Toc39509129)

[2 Algoritmo de Deutsch 2](#_Toc39509130)

[2.1 Problema 2](#_Toc39509131)

[2.2 Implementando las funciones en el computador cuántico 2](#_Toc39509132)

[2.3 Implementando el algoritmo de Deutsch en un computador cuántico 2](#_Toc39509133)

[3 Algoritmo de Deutsch-Jozsa 2](#_Toc39509134)

[3.1 Problema 2](#_Toc39509135)

[3.2 Implementando las funciones en el computador cuántico 2](#_Toc39509136)

[3.3 Implementando el algoritmo de Deutsch-Josza en un computador cuántico 2](#_Toc39509137)

[4 Conclusiones 2](#_Toc39509138)

[5 Bibliografía 2](#_Toc39509139)

# Introducción

En el presente documento vamos a desarrollar los puntos propuestos para los algoritmos de Deutsch y Deutsch-Jozsa por medio de la librería qiskit, con la cual podremos simular los circuitos y así implementar estos algoritmos.

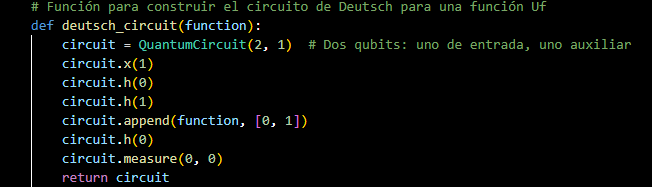
El Algoritmo de Deutsch esta diseñado para determinar si una función booleana de un solo bit de entrada es constante o balanceada, mientras que el algoritmo de Deutsch-Jozsa esta diseñado para determinar si una función es constante o balanceada, la diferencia con el otro algoritmo es que estas funciones operan sobre n bits.

En el siguiente reporte vamos a explicar los ejercicios que realizamos para los dos algoritmos usando qiskit, todo lo realizado esta contenido en un libro de jupyter.

# Algoritmo de Deutsch

## Problema

El algoritmo define si una función dada es balanceada o constante, por medio de la siguiente estructura



Para este caso tomamos un qubits de entrada y uno auxiliar, aplicamos una matriz x al qubit 1, después hadamard al quibit 0 y qubit 1, finalmente añadimos la función que definimos, esta función actúa en los dos qubits, finalmente volvemos a aplicar hadamard y medimos, este algoritmo nos definirá si la función será constante o balanceada de acuerdo con el resultado de la medición.

## Implementando las funciones en el computador cuántico

A partir de las 4 funciones definidas en la entrega del trabajo, comprobamos que el algoritmo funcionará como debería, sus resultados son los siguientes

Diagrama

Descripción generada automáticamente

Estos resultados representan que el conteo que se hicieron para cada circuito, van a dar 0 o 1 para todos los 1000 conteos que se realizaron, asumimos que cada qubit inicia en 0, por lo que los circuitos que dan como resultado 0 son los constantes mientras que los que dan 1 son balanceados.

# Algoritmo de Deutsch-Jozsa

El algoritmo de Detsch-Jozsa determina que funciones son constantes o balanceadas, la diferencia es que estas funciones tendrán n qubits por lo que es mucho más avanzado que el anterior algoritmo.

## Implementando las funciones en el computador cuántico

El siguiente código simula el circuito de Deutsch-Jozsa

Texto

Descripción generada automáticamente

A partir de una función(Oracle) y un n construimos el algoritmo como hemos visto en clase y en las diapositivas dadas, la diferencia con el algoritmo anterior son los hadamard elevados a la n que actúan sobre los estados.

## Implementando el algoritmo de Deutsch-Josza en un computador cuántico

En este caso solo realizamos la función, mas no propusimos las funciones.

# Conclusiones

Podemos concluir varias cosas

1. Podemos utilizar diferentes librerías en Python para poder simular los circuitos como si estuviésemos en un computador cuántico
2. El algoritmo de Deustch-Jozsa es una extensión del algoritmo de Deustch ya que de por si tienen el mismo objetivo pero con parámetros diferentes, uno de ellos trata de resolverlo para una función de un numero limitado de qubits mientras que la otra lo hace para n qubits.

# Bibliografía

<https://en-m-wikipedia-org.translate.goog/wiki/Deutsch–Jozsa_algorithm?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#:~:text=Deutsch%27s%20algorithm%20is%20a%20special,zero%2C%20then%20is%20constant%2C%20otherwise>

<https://docs.quantum.ibm.com/api/qiskit>