**IMPLEMENTACION DE ALGORITMO DE DEUTSCH Y DECUTSCH-JOZSA**

**JHON SEBASTIAN SOSA MUÑOZ**

**ESCUELA DE INGENIEROS JULO GARAVITO**

**JHON.SOSA-M@MAIL.ESCUELAING.EDU.CO**

**ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO**

**BOGOTÁ D.C. 08 DE MAYO 2020**

*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)

Guithub del proyecto: https://github.com/jhonSsosa/Circuitos\_tipo\_Deutch.git

# Introducción

La computación esta cada día más involucrado en nuestras vidas, de tal manera que pudimos encontrar la forma de no solo traducir fenómenos físicos en lenguaje binario, sino que pudimos hallar la forma de utilizar las propiedades cuánticas en qbits, con el objetivo de poder hacer varias operaciones a la vez, pudiendo potenciar procesos muy complicados por hacer desde un ordenador convencional.

En este proyecto se va a presentar la implementación y utilización de librerías de programación quántica tales como quikit, la cual es utilizada para generar matrices y circuitos de programación, en los que se pueden modificar los diferentes estados en los que se encuentran los qbits, utilizando ejemplos famosos como lo pueden ser las matrices Deutsch, y Deutsch Joza, para poder demostrar algunos algoritmos y propiedades famosas en las operaciones de los circuitos.

El documento se desarrollara empezando desde ejemplos de qbits con naturaleza n=2, junto a la demostración de algoritmo de Deutsch, para luego seguir con una serie de ejemplos con aridad n = 4, utilizando circuitos entre medias del algoritmo de Deutsch Joza y la demostración de esta dos con funciones constantes y balanceadas.

# Algoritmo de Deutsch

El algoritmo Deutsch-Jozsa es un [algoritmo cuántico](https://gaz.wiki/wiki/es/Quantum_algorithm)[determinista](https://gaz.wiki/wiki/es/Deterministic_algorithm) propuesto por [David Deutsch](https://gaz.wiki/wiki/es/David_Deutsch) en 1992. Se podría decir que es uno de los primeros algoritmos cuánticos, y desarrollado de tal manera que su implementación es exponencialmente más rápida que cualquier posible algoritmo clásico determinista implementado hasta ahora, y en este informe se va a demostrar el funcionamiento de este algoritmo a través de pruebas de desarrollo que utiliza dos bits de entrada y esta nos devuelve un valor binario.

* + Dibujo:



## Problema

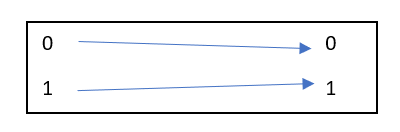
El problema propuesto en ele ejercicio consiste en poder determinar si una función es balanceada o constante.

* El algoritmo es balanceado si todos los resultados de las funciones son diferentes.
* El algoritmo es constante si la salida es un solo valor constante.

Las cuatro funciones a operar son:

Gráfico, Gráfico de líneas

Descripción generada automáticamente



Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## Implementando las funciones en el computador cuántico

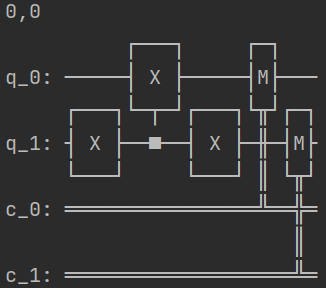
### Función 1

La función prueba que se utilizara en este caso va a ser la siguiente:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

La implementación grafica de la primera función para la correcta implementación es la siguiente:



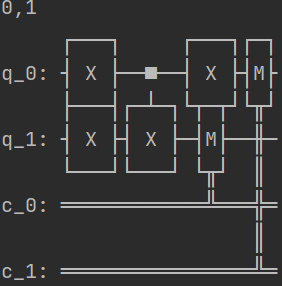
Para la comprobación de resultados, se realizaron un total de 1000 pruebas para cada una de las funciones presentes en este documento para comprobar la naturaleza de la función, dándonos como resultado el siguiente grafico:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Con las siguientes pruebas se siguen exactamente los mismos pasos:

* Dibujo:



* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Dibujo:

Imagen que contiene Diagrama

Descripción generada automáticamente

* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Dibujo:

Imagen que contiene reloj

Descripción generada automáticamente

* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

En cuanto a la realización de la matriz, se puede interpretar de tal manera que las filas serán las salidas de la función, y las columnas eran las entradas, etiquetando con cada una de las funciones posibles en la aridad n = 2, presentando a continuación la siguiente matriz:

### Función 2

La función prueba que se utilizara en este caso va a ser la siguiente:

Imagen que contiene Carta

Descripción generada automáticamente

Y tal como la primera función de la sección 2.2.1, se realizaron las pruebas correspondientes imprimiendo la grafica, recopilando resultados, y realizando su correspondiente matriz para poder realizar su correspondiente análisis.

* Dibujo:

Diagrama

Descripción generada automáticamente

* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Dibujo:

Imagen que contiene Diagrama

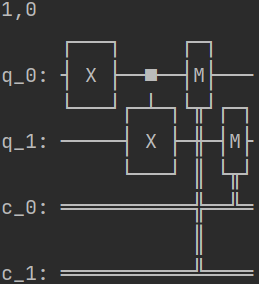
Descripción generada automáticamente

* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Dibujo:



* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Dibujo:

Diagrama

Descripción generada automáticamente con confianza media

* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Matriz correspondiente:

### Función 3

El procedimiento para esta función se debe realizar de la misma manera que se realizó en las funciones uno y dos.

* Función para operar:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* Pruebas:
  + Dibujo:

Diagrama

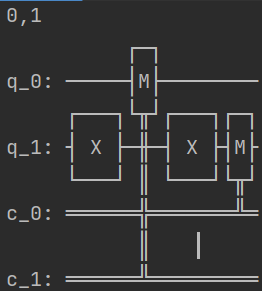
Descripción generada automáticamente

* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:

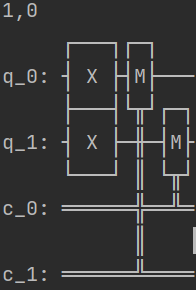


* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:



* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:

Imagen que contiene Gráfico

Descripción generada automáticamente

* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Matriz correspondiente:

### Función 4

* Función a operar:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* + Dibujo:

Diagrama, Esquemático

Descripción generada automáticamente

* + Resultado:

Gráfico, Forma, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:

Diagrama

Descripción generada automáticamente con confianza baja

* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* + Dibujo:

Diagrama

Descripción generada automáticamente

* + Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* Matriz correspondiente:

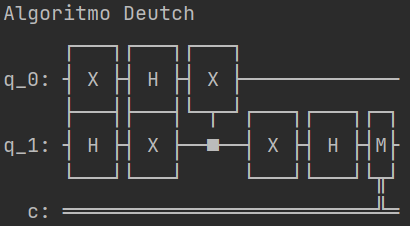
## Implementando el algoritmo de Deutsch en un computador cuántico

El análisis del algoritmo de Deutsch se realiza tomando en cuenta la naturaleza de sus resultados al dar un resultado constante de 1 o de 0, 1 siendo una función balanceada y 0 una función constante.

### Primera función

Se utiliza la función referenciada en la primera prueba de la sección 2.2, y nos damos cuenta que esta función es balanceada con la ejecución y la salida de el circuito y resultados presentados a continuación:

* Dibujo:



* Resultado:

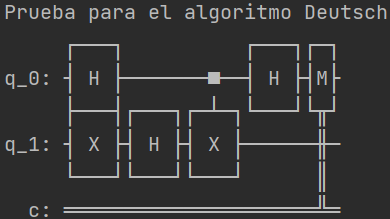
Gráfico de rectángulos

Descripción generada automáticamente con confianza baja

### Función 2

Utilizando el proceso anterior, se grafica un algoritmo el cual tiene como resultados los dibujos y resultados siguientes, con los cuales podemos concluir que la funcion es balanceada.

* Dibujo:



* Resultado:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

### Función 3

En base al circuito y el resultado de esta función, al dar las mil salidas hacia 0 se puede concluir que la función es constante:

* Dibujo:



* Resultado:

Gráfico, Forma

Descripción generada automáticamente

### Función 4

Mismo principio se aplica respecto a la función anterior, dando la conclusión de que esta función es también balanceada:

* Dibujo:

Diagrama

Descripción generada automáticamente con confianza media

* Resultados:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

# Algoritmo de Deutsch-Jozsa

Mismo mecanismo de la función anterior, solo que en vez de aridad n=2, esta utiliza n bits de entrada, y devuelve un valor de similar aridad.

## Problema

En cuanto al planteamiento de esta función, es que en esta ocasión se puede preguntar si la función es constante o balanceada, pero el proceso pasa a ser mas largo al ser de mayor longitud.

## Implementando las funciones en el computador cuántico

Se procederán a utilizar con funciones de aridad 4 para poder desarrollar el algoritmos de Deutsch-Jozsa efectivamente.

### Función 1

La implementación de estas funciones son muy parecidas a las de su antecesora, con la diferencia que el numero de n sea las que el programador considere necesarias para su desarrollo.

* Dibujo:

Diagrama, Esquemático

Descripción generada automáticamente

* Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

### Función 2

Utilizando el procedimiento de la función anterior, esto da como resultado las siguientes salidas:

* Dibujo:

Diagrama, Esquemático

Descripción generada automáticamente

* Resultados:

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

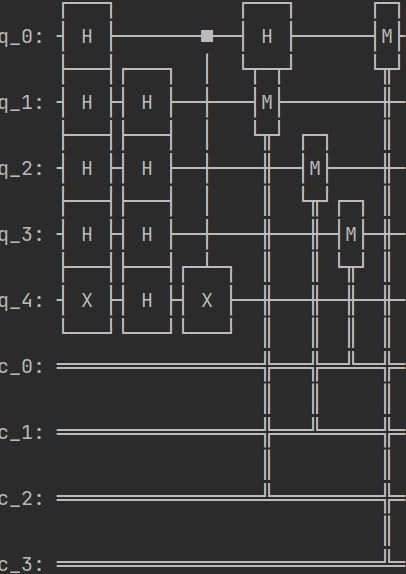
## Implementando el algoritmo de Deutsch-Josza en un computador cuántico

Utilizando las mismas funciones del enunciado anterior, y partiendo del principio de que una funcion es constante en cuanto sus resultados apunten al valor binario 0000, y es balanceada en el supuesto de que los resultados apunten a un resultado diferente.

### Función 1

Usando la primera funcion del anterior numeral, obtenemos el circuito anterior, el cual como resultado dará una funcion balanceada.

* Dibujo:



* Resultado:

Gráfico de rectángulos

Descripción generada automáticamente con confianza media

### Función 2

Siguiendo el procedimiento de la anterior función, se puede concluir que este circuito es balanceado.

* Dibujo:

Diagrama

Descripción generada automáticamente

* Resultado:

Gráfico

Descripción generada automáticamente con confianza media

# Conclusiones

Con a las pruebas ofrecidas en las considerablemente abundantes pruebas realizadas en este proyecto, se puede observar claramente el potencial de la computación cuántica en la realización y comprobación de procesos de un calibre iguales de abundantes, en relación a la rápida demostración de los procesos sin necesidad de una infraestructura de alta gama, las cuales no harían gran diferencia en la operación de las funciones antes mencionadas mediante un dispositivo promedio, pudiendo identificar las aplicaciones que este tipo de computación tendría para el rendimiento general en algunos procesos específicos que la sociedad requira.

# Bibliografía

[The Wild Project #166 ft Eduardo Sáenz de Cabezón | Teorema de Sheldon Cooper, ¿El infinito existe? - YouTube](https://www.youtube.com/watch?v=57Gts-zsb2w)

[Algoritmo de Deutsch – Jozsa - gaz.wiki](https://gaz.wiki/wiki/es/Deutsch%E2%80%93Jozsa_algorithm)