

Laboratorio de matrices hermitianas, unitarias y producto tensor

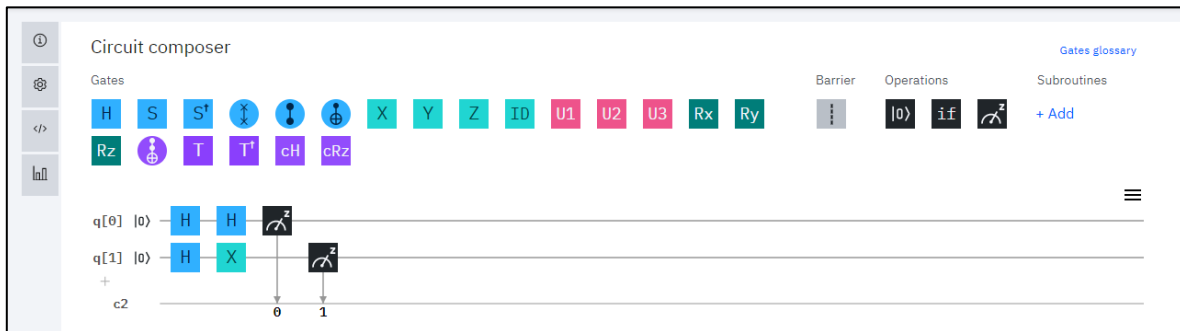


Objetivo del laboratorio

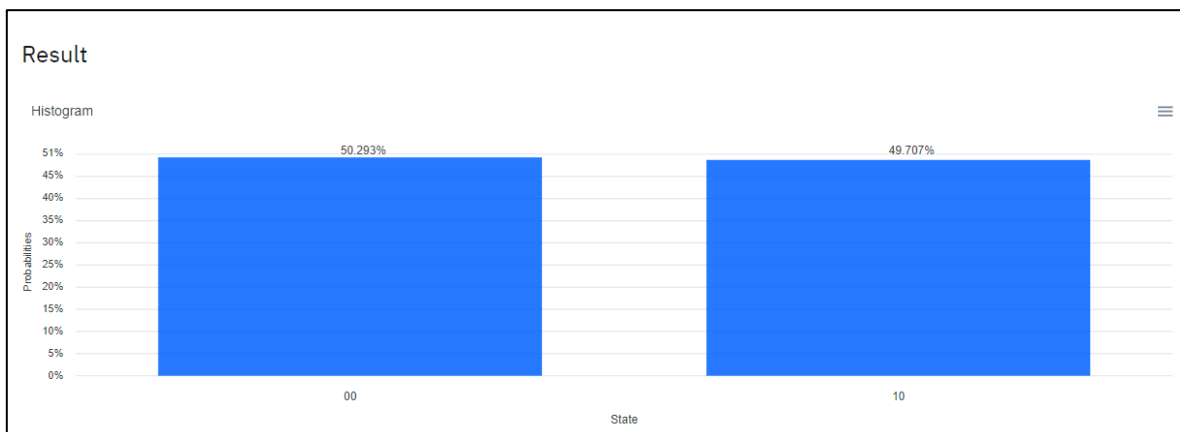
Realizar la simulación de un circuito definido en el computador de IBM, con el fin de comparar el resultado obtenido, con la librería personal, la cual implementa el producto tensor necesario para simular el circuito.

Circuito y simulación en el computador de IBM

Circuito



Resultado de la Simulación



Código que implementa el circuito en sus librerías

```
# Simulacion de circuito

q0 = [[[1,0]],[[0,0]]]
q00 = productoTensorMatrices(q0,q0)

c = [1/math.sqrt(2),0]
temp = [[[1,0],[1,0]],[[1,0],[-1,0]]]
H = multiEscalarMatrices(c,temp)

X = [[[0,0],[1,0]],[[1,0],[0,0]]]

M1 = productoTensorMatrices(H,H)

M2 = productoTensorMatrices(H,X)

temp2 = multiplicarMatrices(M2,q00)
rta = multiplicarMatrices(M1,temp2)
for i in rta:
    print(i)
main()
```

Resultado

```
C:\Users\USUARIO\Documents\2020\Universidad\semestre 9\CNYT\Laboratorio_1_CNYT>python Lab1.py
[[0.7071067811865474, 0.0]]
[[-0.7071067811865474, 0.0]]
[[0.0, 0.0]]
[[0.0, 0.0]]
```

Código que implementa el producto tensor

```
def completarMatrizResultado(temp,matriz,a,b):
    m = len(temp)
    n = len(temp[0])
    fila = 0
    col = 0
    for i in range((m*a),m*(a+1)):
        for j in range((n*b),n*(b+1)):
            matriz[i][j]=temp[fila][col]
            col+=1
        fila+=1
        col=0

def productoTensorMatrices(m1,m2):
    matriz = crearMatrizVacía(len(m1)*len(m2),len(m1[0]*len(m2[0])))
    for i in range(len(m1)):
        for j in range(len(m1[0])):
            temp = multiEscalarMatrices(m1[i][j],m2)
            completarMatrizResultado(temp, matriz, i, j)
    return matriz;
```