

COMPETENCIA DOBLE RENDIJA

INTEGRANTES:

- Thomas Feris Riaño
- Andrés Felipe Pineda
- Juan Sebastián Sánchez

EXPERIMENTO DOBLE RENDIJA

El experimento de la doble rendija fue realizado por primera vez por el científico Thomas Young en el año 1801, este experimento consistía en emitir luz a través de dos rendijas y observar un patrón de interferencias que es ocasionado luego de que la luz se difractara al pasar por las dos rendijas. Este experimento tiene una gran relevancia en el mundo de la mecánica cuántica ya que mediante él se pueden evidenciar conceptos como:

Superposición: El estado de un sistema puede contener múltiples historias

Interferencia: Las historias de los sistemas pueden interactuar entre si

MONTAJE EXPERIMENTO DOBLE RENDIJA

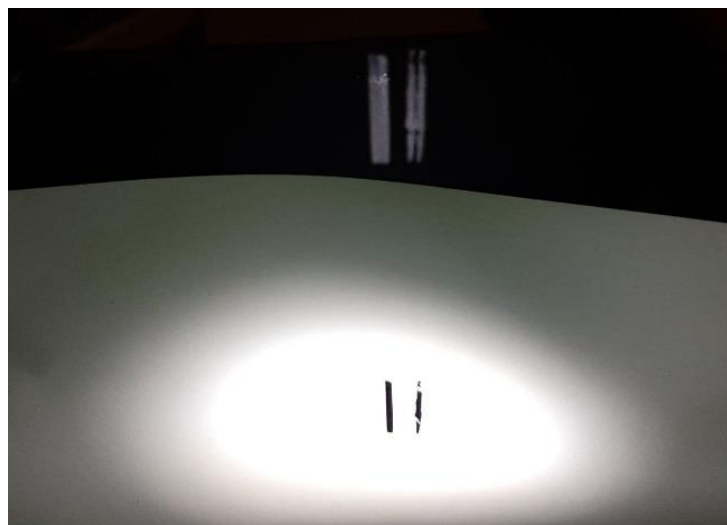
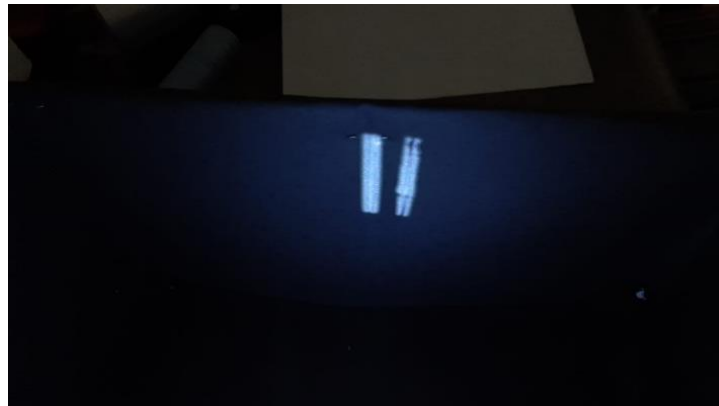
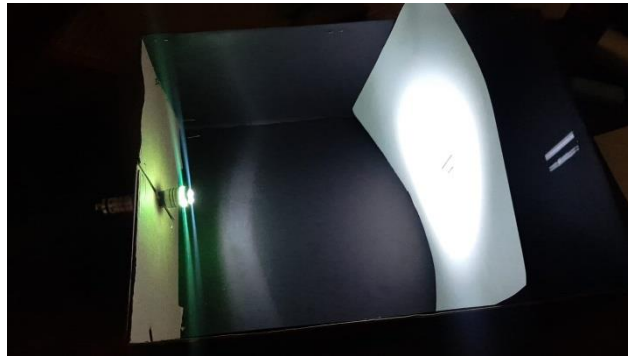
Para realizar el montaje del experimento de doble rendija se tomaron materiales bastantes fáciles de conseguir y cotidianos, para tal fenómeno que ocurre, para empezar realizamos dos cortes muy finos en la superficie, en este caso papel aluminio y después hacer un corte rectangular en otra superficie, de manera de que en la superficie resulte con una “ventana”, posteriormente se pegara con cinta la superficie en la que se hicieron las rendijas en la superficie en la cual se hizo la ventana, una vez hecho esto, se pondrá la superficie que se acaba de realizar delante de una pared u otra superficie que de preferencia tenga algún color oscuro, luego se apuntara con un emisor de luz (en este caso, un láser infrarrojo) a las rendijas que se encuentran en la superficie y se observara el patrón resultante en la superficie de color oscuro, en este caso cartulina de color negro.

MATERIALES:

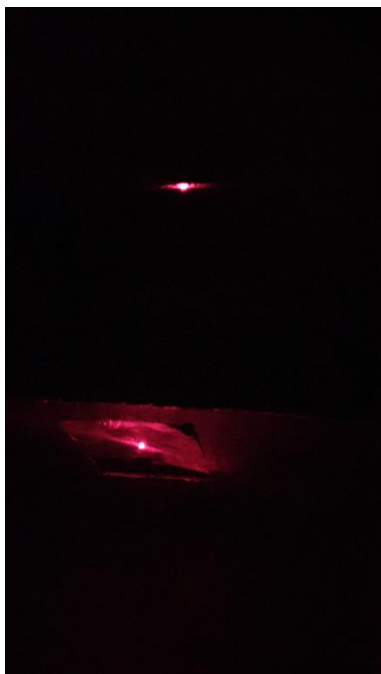
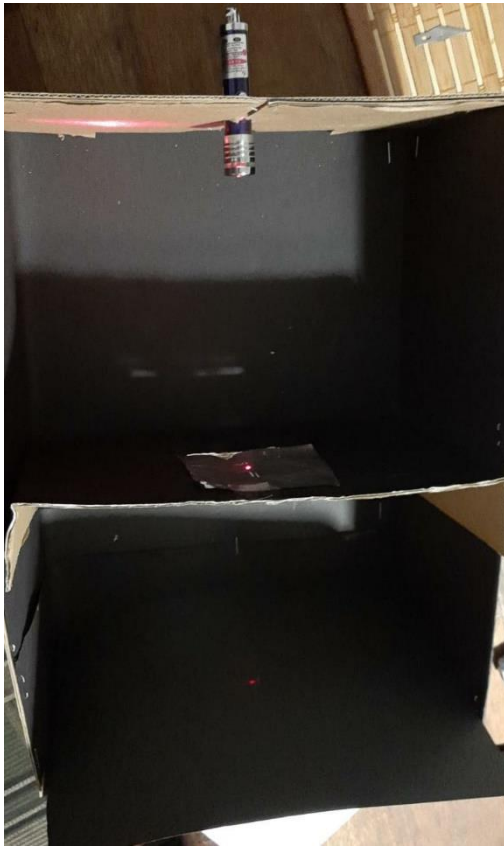
- 1 cartulina Blanca y Negra
- Laser infrarrojo
- Papel Aluminio
- Bisturí
- Cinta

FOTOS DEL EXPERIMENTO

LINTERNA



LASER



EXPLICACION

Muchas personas explican el patrón que se puede evidenciar en el experimento como un efecto causado por la interferencia entre las ondas que son generadas en el momento en el que la luz del infrarrojo pasa a través de las rendijas, sin embargo al momento de realizar el experimento lanzando electrones a la doble rendija en vez de los fotones que emite el infrarrojo, se vuelve a generar el mismo patrón, otra explicación a este fenómeno es la que plantea David Deutch en su libro "The fabric of reality", en la cual el autor plantea que lo que normalmente consideramos como el universo (El todo de la realidad física) realmente solo es una pequeña porción del multiverso, sumado a esto, los diferentes experimentos de interferencia con partículas son la muestra de que existe un multiverso que contiene varias contra partes de cada partícula en nuestro universo tangible, en otras palabras, el multiverso está particionado en varios universos paralelos y el fenómeno de la interferencia se debe considerar como algo que envuelve a más de una partícula, ya que son las múltiples historias de una partícula las que están interactuando entre sí.

SIMULACION

Implementamos una simulación que representa el fenómeno físico del experimento de la doble rendija.

Números Reales

```
def prettyprintingsVectores(vector):
    for i in range(len(vector)):
        for j in range(1):
            print(vector[i])
def prettyprintingsVectores2(vector):
    for i in range(len(vector)):
        for j in range(1):
            print(vector[i][j])
def matriz_sobre_Vector(m1,v1):
    cont = 1
    suma = 0
    matriz_Vector = [0 for i in range(len(m1))]
    for i in range(len(m1)):
        for j in range(len(m1)):
            for k in range(len(m1[0])):
                suma = suma + m1[i][k] * v1[k]
            if suma == 0:
                matriz_Vector[i] = False
            else:
                matriz_Vector[i] = True
            suma = 0
            break
    return matriz_Vector
def matriz_sobre_Vector2(m1,v1):
    cont = 1
```

```

matriz_Vector = [[[0,0] for j in range(1)] for i in range(len(m1))]
for i in range(len(m1)):
    for j in range(len(v1[0])):
        for k in range(len(m1[0])):
            matriz_Vector[i][j][0] = matriz_Vector[i][j][0] +
(m1[i][k][0] * v1[k][j] - m1[i][k][1]*v1[k][j+cont])
            matriz_Vector[i][j][1] = matriz_Vector[i][j][1] +
(m1[i][k][1] * v1[k][j] + m1[i][k][0]*v1[k][j+cont])
        break
    return(matriz_Vector)

```

Números Imaginarios

```

def cuantico_rendija(m,e,n):
    if len(m) == 0 or len(e) == 0:
        print("No se puede realizar la operación.")
    elif len(m[0]) != len(e):
        print("No se puede realizar la operación.")
    else:
        cont = 1
        while cont <= n:
            rendija = matriz_sobre_Vector2(m,e)
            cont += 1
        prettyprintingsVectores2(rendija)
    return(rendija)
def clasico_rendija(m,e,n):
    if len(m) == 0 or len(e) == 0:
        print("No se puede realizar la operación.")
    elif len(m[0]) != len(e):
        print("No se puede realizar la operación.")
    else:
        cont = 1
        while cont <= n:
            rendija = matriz_sobre_Vector2(m,e)
            cont += 1
        prettyprintingsVectores2(rendija)
    print(rendija)
    return(rendija)

```

Estos 2 programas están diseñados para, el primer de número reales y el segundo para incluir números imaginarios, su funcionamiento es del modo en que comprueba que se puede realizar la operación de la matriz por el estado, si es posible, dependiendo del número de clics, multiplica el estado por la matriz el número de veces que se pida y entrega el resultado deseado.

Al ejecutar estos programas obtenemos como resultado:

```

m2 = [[(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]
      ,[(1/(2**(1/2)),0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]
      ,[(1/(2**(1/2)),0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]
      ,[(0,0),(-1/(6**(1/2)),1/(6**(1/2))),(0,0),(1,0),(0,0),(0,0),(0,0),(0,0)]
      ,[(0,0),(-1/(6**(1/2)),-1/(6**(1/2))),(0,0),(0,0),(1,0),(0,0),(0,0),(0,0)]
      ,[(0,0),(1/(6**(1/2)),-1/(6**(1/2))),(0,0),(-1/(6**(1/2)),1/(6**(1/2))),(0,0),(0,0),(1,0),(0,0),(0,0)]
      ,[(0,0),(0,0),(-1/(6**(1/2)),-1/(6**(1/2))),(0,0),(0,0),(0,0),(1,0),(0,0)]
      ,[(0,0),(0,0),(1/(6**(1/2)),-1/(6**(1/2))),(0,0),(0,0),(0,0),(0,0),(1,0)]]""
""e2 = [(1,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]

```

Esto dará la respuesta a la simulación y el resultado que deseamos buscar.

```

0.0
0.0
0.0
0.16646399999999997
0.16646399999999997
0.0
0.16646399999999997
0.16646399999999997

```

SIMULACION PROBABILISTICA

