



## Examen

- **Autor:** Jonnathan Oswaldo Matute Curillo
- **Maestro:** Diego Quisi
- **Materia:** Simulación
- **Universidad:** Universidad Politécnica Salesiana
- **Carrera:** Ciencias de la computación

### Implementación del Generador de Cuadrados Medios

A continuación se realiza un ejemplo practico de como implementar y ejecutar el generador de numeros pseudoaleatorios utilizando el algoritmo de cuadros medios.

```
def cuadrados_medios(xo, digitos, iteraciones):
```

```
    lis = []
```

```

semilla = int(xo)
aux = gP(digitos)
for i in range(iteraciones):
    x= semilla**2
    l= len(str(x))
    u= str(x)[int(l/2)-aux[0]:int(l/2)+aux[1]]
    r= round(int(u)/10**digitos,2)
    lis.append(r)
    semilla=int(u)
return lis

```

## Implementación de Congruencia Lineal

Otro de los generadores de numeros aleatorios utilizados en la actualidad son los de congruencia lineal, a continuación se implementa en python su proceso para validación

```

def congruencia_Lineal(xo, a, b, m, iteraciones):
    aux = 0.00
    lis =[]
    for i in range(iteraciones):
        xo = (a * xo + b) % m
        aux = round(xo/m,2)
        lis.append(aux)
    return lis

```

## ▼ Lista

```

def lisDict(ngr, au, lis):
    gr = []
    aux=0.00
    for i in range(ngr+1):
        gr.append(round(aux,2))
        aux=aux+au
    a=0
    b=1
    ran={}
    for i in range(len(gr)-1):
        f=gr[a]
        s=gr[b]
        ran.update({str(f)+", "+str(s):[]})
        for i in lis:
            if i==0.05:
                if i >=f and i <=s:
                    ran[str(f)+", "+str(s)].append(i)
            else:
                if i >f and i <=s:
                    ran[str(f)+", "+str(s)].append(i)
        a=b
        b=a+1
    return ran

```

```

def gP(digitos):
    aux1 =0
    aux2 =0
    if digitos%2 !=0:
        aux1 = int(digitos/2)
        aux2 = int(digitos/2)+1
    else:
        aux1 = int(digitos/2)
        aux2 = aux1
    return aux1,aux2

from matplotlib import pyplot as plt
import math as mt

def chiCuadrado(lis, v):
    num = int(mt.sqrt(len(lis)))
    d = lisDict(num,1/num, lis)
    s = 0.00
    print(" Intervalo ", "      Ei ", "      Oi ", " (Oi-Ei)**2/Ei")
    for x, it in enumerate(d.items()):
        f = ((len(it[1])-num)**2)/num
        s+=f
        print(x, " ", str(num)+"("+it[0]+") ", len(it[1])," ", f)

    plt.hist(lis)
    plt.ylabel('Frecuencia')
    plt.xlabel('valores')
    plt.title('Chi cuadrado')
    plt.show()

    print("Suma : ",s)
    if s< v:
        return True
    else:
        return False

def main():
    valor = 16.9
    print("Comparacion 1 - cuadrados_medios")
    iteraciones = 100
    digitos = 7
    lista = cuadrados_medios(74731897457, digitos, iteraciones)
    res=chiCuadrado(lista,valor)
    print("Respuesta: ",res)

    print("")
    print("Comparacion 2 - congruencia_Lineal")

    lista2 = congruencia_Lineal(7,74731897457,37747318974,19,iteraciones)
    res2 = chiCuadrado(lista2,valor)
    print("Respuesta: ",res2)

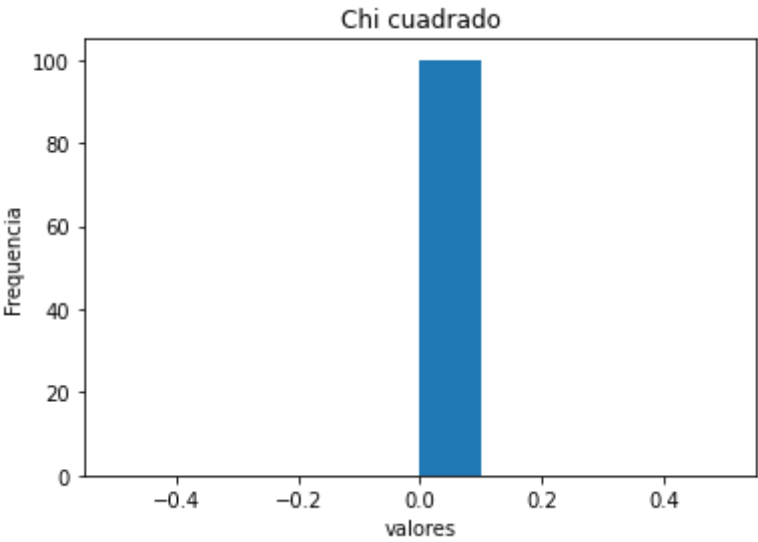
```

```
if __name__ == "__main__":  
    main()
```



Comparacion 1 - cuadrados\_medios

Intervalo	Ei	Oi	$(O_i - E_i)^2 / E_i$
0 10(0.0,0.1)	0	10.0	
1 10(0.1,0.2)	0	10.0	
2 10(0.2,0.3)	0	10.0	
3 10(0.3,0.4)	0	10.0	
4 10(0.4,0.5)	0	10.0	
5 10(0.5,0.6)	0	10.0	
6 10(0.6,0.7)	0	10.0	
7 10(0.7,0.8)	0	10.0	
8 10(0.8,0.9)	0	10.0	
9 10(0.9,1.0)	0	10.0	



Suma : 100.0  
Respuesta: False

✓ 0 s se ejecutó 14:57

