

Nombre: Jorge Arévalo

Docente: Ing. Diego Quisi

Materia: Simulación

Prueba Práctica

In [57]:

```
1 import numpy as np
2 import math as mt
3 import matplotlib.pyplot as plt
```

In [58]:

```
1 def metodoCongruenciaLineal(datox, a, c, mod, iteracion):
2     num = 0.00
3     arreglo = []
4     for i in range(iteracion):
5         datox = (a * datox + c) % mod
6         num = round(datox/mod,2)
7         arreglo.append(num)
8     return arreglo
```

In [59]:

```
1 def obtenerPos(digs):
2     dato1 = 0
3     dato2 = 0
4     if digs%2 != 0:
5         dato1 = int(digs/2)
6         dato2 = int(digs/2)+1
7     else:
8         dato1 = int(digs/2)
9         dato2 = dato1
10    return dato1,dato2
```

In [60]:

```
1 def metodoCuadradosMedios(iteracion, val, digs):
2     arreglo = []
3     x0_semilla = int(val)
4     aum = obtenerPos(digs)
5     for i in range(iteracion):
6         xn2= x0_semilla**2
7         lon = len(str(xn2))
8         ui = str(xn2)[int(lon/2)-aum[0]:int(lon/2)+aum[1]]
9         rn = round(int(ui)/10**digs,2)
10        #print(i, " ", x0_semilla," ",xn2, " ", Lon, " ",ui, " ", rn)
11        arreglo.append(rn)
12        x0_semilla=int(ui)
13    #print(" ")
14    return arreglo
15 iteracion = 100
16 digs = 7
17 arreglo = metodoCuadradosMedios(iteracion, 74731897457, digs)
```

In [61]:

```
1 def metodoLista(num_grupos, aumento, arreglo):
2     grupos = []
3     ini=0.00
4     for i in range(num_grupos+1):
5         grupos.append(round(ini,2))
6         ini=ini+aumento
7     a=0
8     b=1
9     rangos={}
10    for i in range(len(grupos)-1):
11        inf=grupos[a]
12        sup=grupos[b]
13        rangos.update({str(inf)+", "+str(sup):[]})
14        for i in arreglo:
15            if i==0.00:
16                if i >=inf and i <=sup:
17                    rangos[str(inf)+", "+str(sup)].append(i)
18            else:
19                if i >inf and i <=sup:
20                    rangos[str(inf)+", "+str(sup)].append(i)
21        a=b
22        b=a+1
23    return rangos
```

In [62]:

```

1 def metodoChi_Cuadrado(arreglo, valor):
2     n = int(mt.sqrt(len(arreglo)))
3     dic = metodoLista(n,1/n, arreglo)
4     suma = 0.00
5     print(" Intervalo ", "      Ei ", "      Oi ", " (Oi-Ei)**2/Ei")
6     for datox, it in enumerate(dic.items()):
7         f = ((len(it[1])-n)**2)/n
8         suma+=f
9         print(datox, "      ", str(n)+"("+it[0]+")      ", len(it[1]),"      ", f)
10    print("")
11    print("Valor Chi-Cuadrado Cuadrados Medios: ",suma)
12    if suma< valor:
13        return True
14    else:
15        return False
16 def metodoChi_Congruencia(arreglo, valor):
17     n = int(mt.sqrt(len(arreglo)))
18     dic = metodoLista(n,1/n, arreglo)
19     suma = 0.00
20     print(" Intervalo ", "      Ei ", "      Oi ", " (Oi-Ei)**2/Ei")
21     for datox, it in enumerate(dic.items()):
22         f = ((len(it[1])-n)**2)/n
23         suma+=f
24         print(datox, "      ", str(n)+"("+it[0]+")      ", len(it[1]),"      ", f)
25    print("")
26    print("Valor Chi-Cuadrado Congruencia Lineal: ",suma)
27    if suma< valor:
28        return True
29    else:
30        return False

```

METODO CUADRADOS MEDIOS

In [63]:

```

1 valor = 16.9
2 iteracion = 100
3 digs = 7
4 arreglo = metodoCuadradosMedios(iteracion, 74731897457, digs)
5 res=metodoChi_Cuadrado(arreglo,valor)
6 print(" ")
7 print("Frecuencia: ", arreglo)

```

Intervalo	Ei	Oi	(Oi-Ei)**2/Ei
0	10(0.0,0.1)	11	0.1
1	10(0.1,0.2)	18	6.4
2	10(0.2,0.3)	12	0.4
3	10(0.3,0.4)	7	0.9
4	10(0.4,0.5)	13	0.9
5	10(0.5,0.6)	9	0.1
6	10(0.6,0.7)	7	0.9
7	10(0.7,0.8)	5	2.5
8	10(0.8,0.9)	8	0.4
9	10(0.9,1.0)	10	0.0

Valor Chi-Cuadrado Cuadrados Medios: 12.600000000000001

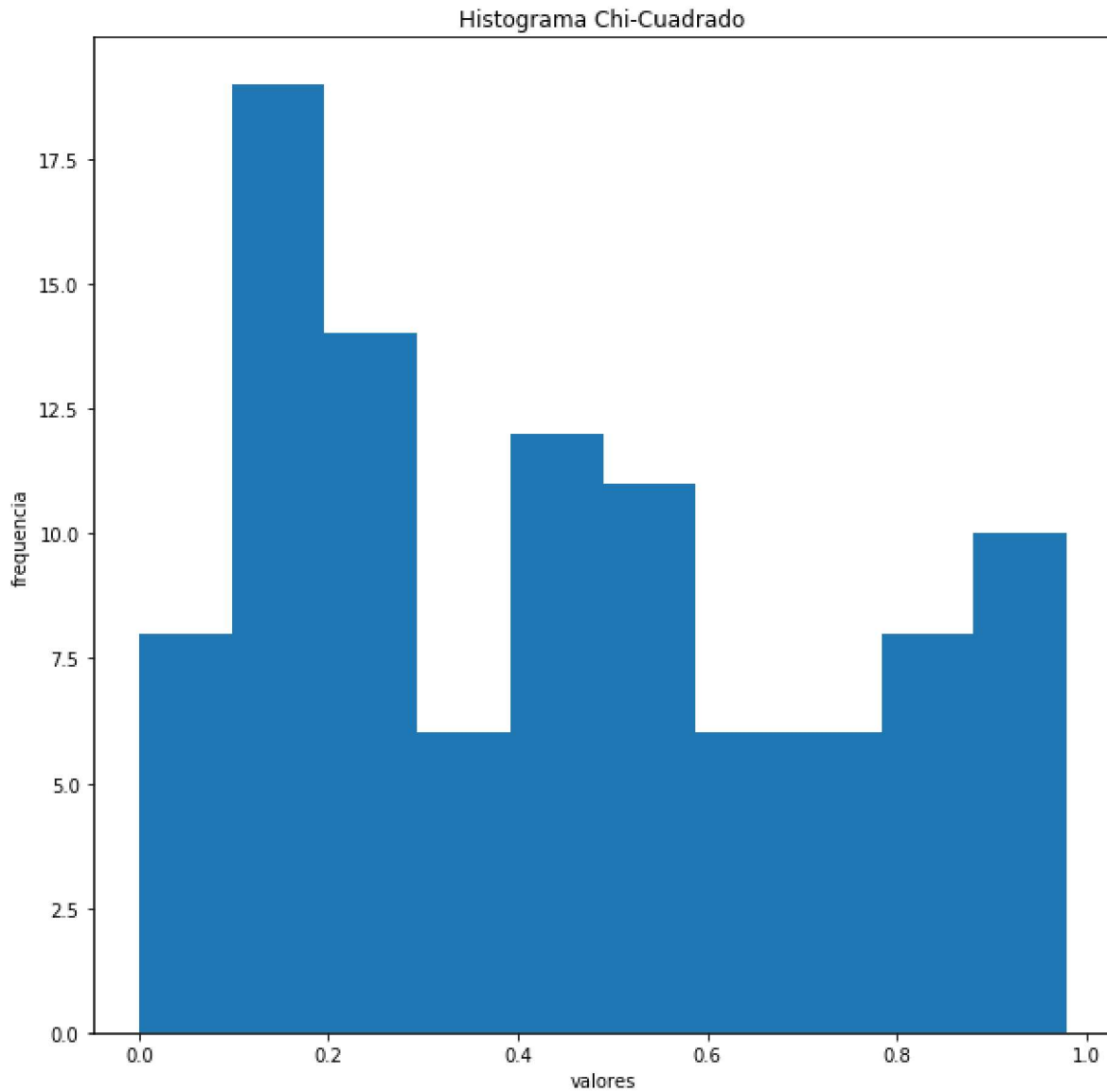
Frecuencia: [0.98, 0.84, 0.11, 0.22, 0.38, 0.17, 0.93, 0.41, 0.03, 0.68, 0.35, 0.35, 0.76, 0.41, 0.04, 0.11, 0.17, 0.23, 0.82, 0.7, 0.17, 0.48, 0.4, 0.24, 0.43, 0.65, 0.2, 0.71, 0.88, 0.93, 0.94, 0.13, 0.56, 0.55, 0.58, 0.18, 0.51, 0.52, 0.05, 0.97, 0.23, 0.23, 0.29, 0.5, 0.12, 0.97, 0.31, 0.16, 0.41, 0.76, 0.26, 0.63, 0.95, 0.1, 0.25, 0.88, 0.71, 0.16, 0.52, 0.38, 0.22, 0.0, 0.58, 0.55, 0.87, 0.0, 0.01, 0.2, 0.91, 0.44, 0.16, 0.5, 0.45, 0.81, 0.27, 0.19, 0.32, 0.14, 0.51, 0.1, 0.1, 0.48, 0.13, 0.24, 0.01, 0.19, 0.13, 0.84, 0.28, 0.65, 0.42, 0.72, 0.61, 0.92, 0.46, 0.96, 0.65, 0.0, 0.47, 0.87]

HISTOGRAMA CUADRADOS MEDIOS

In [64]:

```
1 #HISTOGRAMA CUADRADOS MEDIOS
2 print("HISTOGRAMA CUADRADOS MEDIOS")
3 plt.hist(arreglo)
4 plt.ylabel('frecuencia')
5 plt.xlabel('valores')
6 plt.title('Histograma Chi-Cuadrado')
7 plt.show()
```

HISTOGRAMA CUADRADOS MEDIOS



METODO DE CONGRUENCIA LINEAL

In [65]:

```

1 datox = 7
2 a = 74731897457
3 c = 37747318974
4 m = 19
5 arreglo2 = metodoCongruenciaLineal(datox,a,c,m,iteracion)
6 res2 = metodoChi_Congruencia(arreglo2,valor)

```

Intervalo	Ei	Oi	$(O_i - E_i)^2 / E_i$
0	10(0.0,0.1)	11	0.1
1	10(0.1,0.2)	11	0.1
2	10(0.2,0.3)	0	10.0
3	10(0.3,0.4)	22	14.4
4	10(0.4,0.5)	0	10.0
5	10(0.5,0.6)	0	10.0
6	10(0.6,0.7)	11	0.1
7	10(0.7,0.8)	11	0.1
8	10(0.8,0.9)	23	16.9
9	10(0.9,1.0)	11	0.1

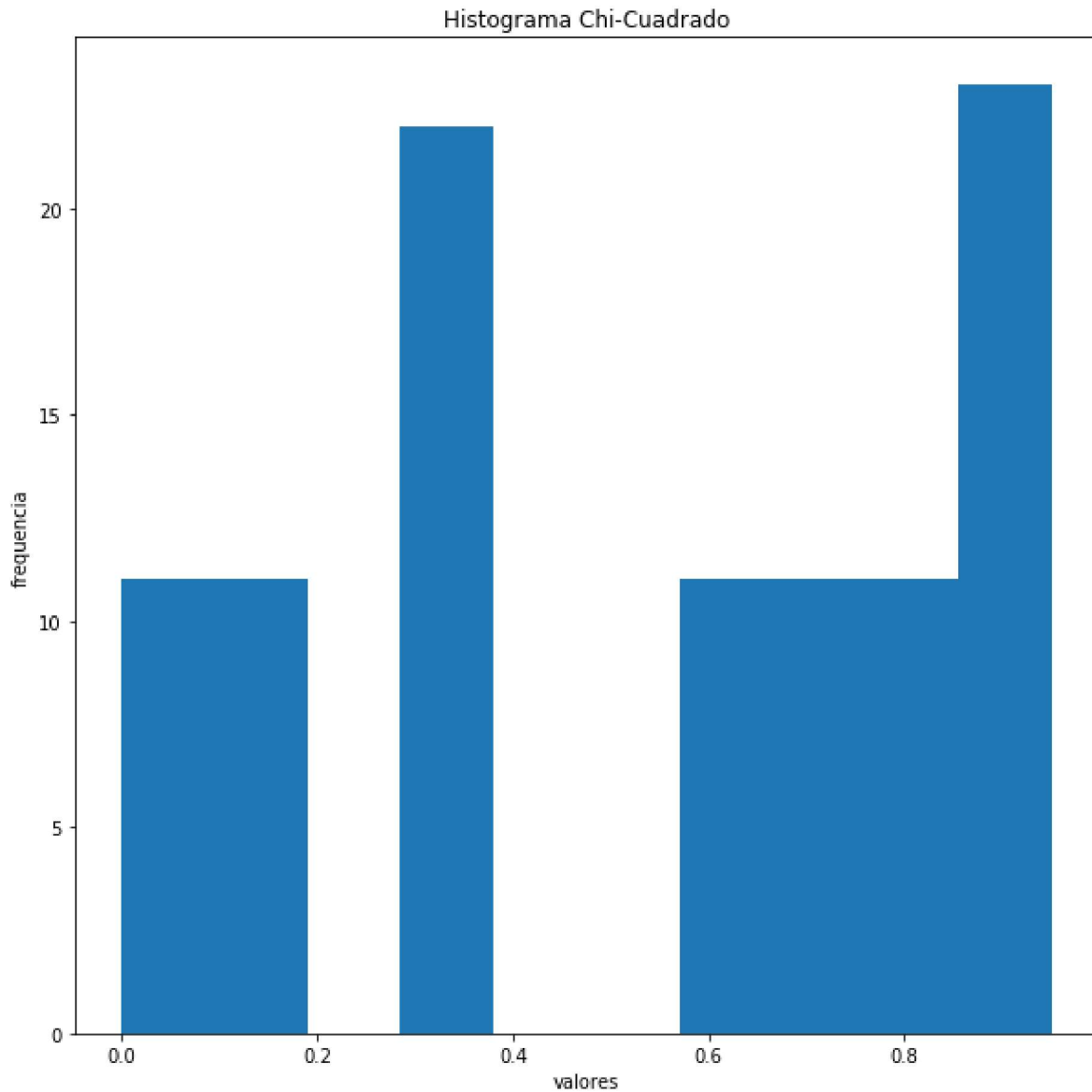
Valor Chi-Cuadrado Congruencia Lineal: 61.800000000000004

HISTOGRAMA CONGRUENCIA LINEAL

In [66]:

```
1 #HISTOGRAMA CONGRUENCIA LINEAL
2 print("HISTOGRAMA CONGRUENCIA LINEAL")
3 plt.hist(arreglo2)
4 plt.ylabel('frecuencia')
5 plt.xlabel('valores')
6 plt.title('Histograma Chi-Cuadrado')
7 plt.show()
```

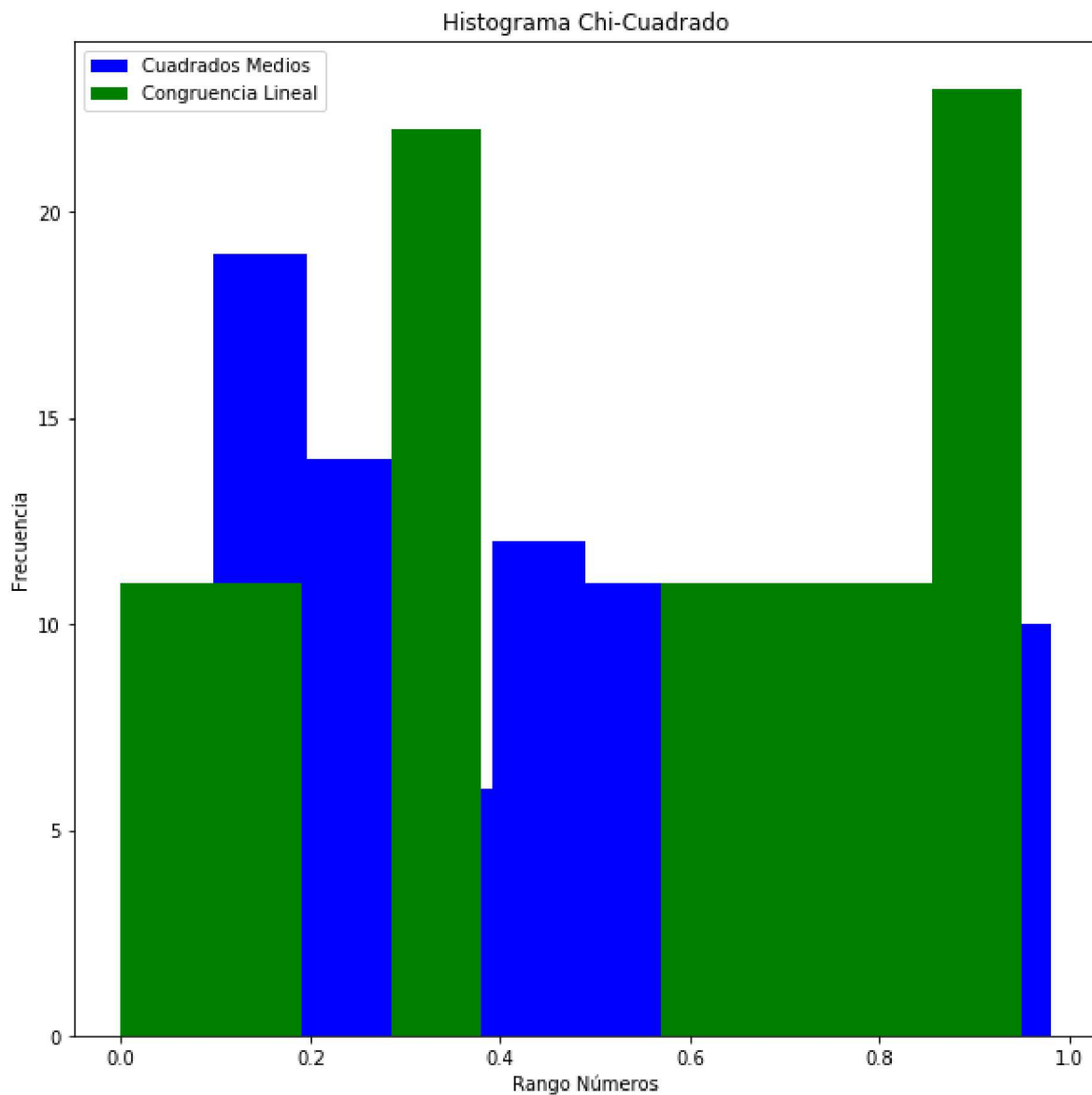
HISTOGRAMA CONGRUENCIA LINEAL



Comparación

In [67]:

```
1 plt.hist(arreglo,color='blue',label='Cuadrados Medios')
2 plt.hist(arreglo2,color='Green',label='Congruencia Lineal')
3 plt.xlabel('Rango Números')
4 plt.ylabel('Frecuencia')
5 plt.title('Histograma Chi-Cuadrado')
6 plt.legend(loc='upper left')
7 plt.show()
```



Conclusión

El mejor metodo que presenta es el de los Cuadrados Medios ya que genera numeros aleatorios adecuados para realizar la práctica.

Tenemos el valor de:

Chi-Cuadrado Cuadrados Medios: 12.6

Chi-Cuadrado Congruencia Lineal: 61.8

In []:

1	
---	--