

Congruencia Lineal

Materia:

SIMULACION

Docentes:

Ing. Diego Quisi

Estudiante:

Ricardo Jara

Generar 10 listas de 100 números aleatorios con 5 semillas con datos relacionados al sistema y 5 semillas personales.

```
In [1]: #Importamos Librerias
import matplotlib.pyplot as plt
import pandas as pd
import psutil
```

- Método que calculara la congruencia lineal y me retorna una lista de números en base a la a una semilla y la cantidad de números medios que deseo sacar.

```
In [15]: def congruencia(semilla,iteraciones,a,c,m, dig):
lista = []
for i in range(1, iteraciones):
    xn = (a*semilla + c) % m
    semilla = xn
    lista.append(xn)
return lista
```

- Semillas

```
In [16]: #aleatorias
semilla1 = 20202020
```

```

semilla2 = 20212021
semilla3 = 19970827
semilla4 = 20180805
semilla5 = 19961207
semilla6 = psutil.virtual_memory().total #memoria virtual
semilla7 = int(psutil.cpu_freq().current) #Frecuencias d HZ
semilla8 = int(psutil.disk_usage('/').total/1000000) #Capacidad disco Local C
semilla9 = psutil.disk_io_counters().write_count #Número de Lecturas del disco
semilla10 = psutil.net_io_counters().bytes_sent #Numero de paquetes recibidos
#Lista de semillas
semillas = [semilla1,semilla2,semilla3,semilla4,semilla5,semilla6,semilla7,semilla8,

```

```

In [17]: cantidad = 100
num_digi = 5
a=semillas[5]
c=semillas[8]
M=19
d = {}

#calculamos la lista de numeros randomicos
for l in semillas:
    lista = congruencia(l, cantidad,a,c,M,num_digi)
    #llenamos un dataframe con las listas
    d[str(l)] = lista
#Mostramos DF / lista de randomicos
df = pd.DataFrame(data=d)
with pd.option_context('display.max_rows', None, 'display.max_columns', None): # mo
    display(df)

```

	20202020	20212021	19970827	20180805	19961207	17027518464	2208	25493	22406	954
0	10	14	4	1	6	3	10	13	16	
1	8	13	10	11	3	4	8	7	6	
2	15	7	8	14	4	10	15	9	3	
3	0	9	15	13	10	8	0	2	4	
4	5	2	0	7	8	15	5	17	10	
5	16	17	5	9	15	0	16	12	8	
6	6	12	16	2	0	5	6	1	15	
7	3	1	6	17	5	16	3	11	0	
8	4	11	3	12	16	6	4	14	5	
9	10	14	4	1	6	3	10	13	16	
10	8	13	10	11	3	4	8	7	6	
11	15	7	8	14	4	10	15	9	3	
12	0	9	15	13	10	8	0	2	4	
13	5	2	0	7	8	15	5	17	10	
14	16	17	5	9	15	0	16	12	8	
15	6	12	16	2	0	5	6	1	15	
16	3	1	6	17	5	16	3	11	0	
17	4	11	3	12	16	6	4	14	5	
18	10	14	4	1	6	3	10	13	16	

	20202020	20212021	19970827	20180805	19961207	17027518464	2208	25493	22406	954
19	8	13	10	11	3	4	8	7	6	
20	15	7	8	14	4	10	15	9	3	
21	0	9	15	13	10	8	0	2	4	
22	5	2	0	7	8	15	5	17	10	
23	16	17	5	9	15	0	16	12	8	
24	6	12	16	2	0	5	6	1	15	
25	3	1	6	17	5	16	3	11	0	
26	4	11	3	12	16	6	4	14	5	
27	10	14	4	1	6	3	10	13	16	
28	8	13	10	11	3	4	8	7	6	
29	15	7	8	14	4	10	15	9	3	
30	0	9	15	13	10	8	0	2	4	
31	5	2	0	7	8	15	5	17	10	
32	16	17	5	9	15	0	16	12	8	
33	6	12	16	2	0	5	6	1	15	
34	3	1	6	17	5	16	3	11	0	
35	4	11	3	12	16	6	4	14	5	
36	10	14	4	1	6	3	10	13	16	
37	8	13	10	11	3	4	8	7	6	
38	15	7	8	14	4	10	15	9	3	
39	0	9	15	13	10	8	0	2	4	
40	5	2	0	7	8	15	5	17	10	
41	16	17	5	9	15	0	16	12	8	
42	6	12	16	2	0	5	6	1	15	
43	3	1	6	17	5	16	3	11	0	
44	4	11	3	12	16	6	4	14	5	
45	10	14	4	1	6	3	10	13	16	
46	8	13	10	11	3	4	8	7	6	
47	15	7	8	14	4	10	15	9	3	
48	0	9	15	13	10	8	0	2	4	
49	5	2	0	7	8	15	5	17	10	
50	16	17	5	9	15	0	16	12	8	
51	6	12	16	2	0	5	6	1	15	
52	3	1	6	17	5	16	3	11	0	
53	4	11	3	12	16	6	4	14	5	
54	10	14	4	1	6	3	10	13	16	

	20202020	20212021	19970827	20180805	19961207	17027518464	2208	25493	22406	954
55	8	13	10	11	3	4	8	7	6	
56	15	7	8	14	4	10	15	9	3	
57	0	9	15	13	10	8	0	2	4	
58	5	2	0	7	8	15	5	17	10	
59	16	17	5	9	15	0	16	12	8	
60	6	12	16	2	0	5	6	1	15	
61	3	1	6	17	5	16	3	11	0	
62	4	11	3	12	16	6	4	14	5	
63	10	14	4	1	6	3	10	13	16	
64	8	13	10	11	3	4	8	7	6	
65	15	7	8	14	4	10	15	9	3	
66	0	9	15	13	10	8	0	2	4	
67	5	2	0	7	8	15	5	17	10	
68	16	17	5	9	15	0	16	12	8	
69	6	12	16	2	0	5	6	1	15	
70	3	1	6	17	5	16	3	11	0	
71	4	11	3	12	16	6	4	14	5	
72	10	14	4	1	6	3	10	13	16	
73	8	13	10	11	3	4	8	7	6	
74	15	7	8	14	4	10	15	9	3	
75	0	9	15	13	10	8	0	2	4	
76	5	2	0	7	8	15	5	17	10	
77	16	17	5	9	15	0	16	12	8	
78	6	12	16	2	0	5	6	1	15	
79	3	1	6	17	5	16	3	11	0	
80	4	11	3	12	16	6	4	14	5	
81	10	14	4	1	6	3	10	13	16	
82	8	13	10	11	3	4	8	7	6	
83	15	7	8	14	4	10	15	9	3	
84	0	9	15	13	10	8	0	2	4	
85	5	2	0	7	8	15	5	17	10	
86	16	17	5	9	15	0	16	12	8	
87	6	12	16	2	0	5	6	1	15	
88	3	1	6	17	5	16	3	11	0	
89	4	11	3	12	16	6	4	14	5	
90	10	14	4	1	6	3	10	13	16	

	20202020	20212021	19970827	20180805	19961207	17027518464	2208	25493	22406	954
91	8	13	10	11	3	4	8	7	6	
92	15	7	8	14	4	10	15	9	3	
93	0	9	15	13	10	8	0	2	4	
94	5	2	0	7	8	15	5	17	10	
95	16	17	5	9	15	0	16	12	8	
96	6	12	16	2	0	5	6	1	15	
97	3	1	6	17	5	16	3	11	0	
98	4	11	3	12	16	6	4	14	5	