

Generación de números pseudo-aleatorios por congruencia lineal

AUTOR: CARLOS MOROCHO

In [5]:

```
#Importación de los paquetes
import math
import psutil
import resource
import matplotlib.pyplot as pp

%matplotlib inline
```

1. Definiendo la función de generación

In [6]:

```
def generar_numeros_aleatorios(x0, a, b, m, iteraciones):
    xn = x0

    for i in range(iteraciones):
        aux = (a * xn + b) % m

        if i > 0:
            ui = xn / m
            print(f'Paso: {i}, Xn: {xn}, Ui: {ui}')
        else:
            print(f'Paso: {i}, Xn: {xn}, Ui: ---')

        xn = aux
```

2. Usando información del Hardware como semilla inicial

In [10]:

```
#Usaremos información del disco
semilla1 = psutil.disk_io_counters()[0] #Contador de lecturas
semilla2 = psutil.disk_io_counters()[1] #Contador de escrituras
semilla3 = psutil.disk_io_counters()[2] #Total de bytes leídos
semilla4 = psutil.disk_io_counters()[3] #Total de bytes escritos
semilla5 = psutil.disk_io_counters()[4] #Tiempo total de lectura

print(f'Semilla 1: {semilla1}\n')
generar_numeros_aleatorios(semilla1, 5, 7, 11, 10)

print(f'\nSemilla 2: {semilla2}\n')
generar_numeros_aleatorios(semilla2, 7, 11, 3, 10)

print(f'\nSemilla 3: {semilla3}\n')
generar_numeros_aleatorios(semilla3, 17, 13, 11, 10)

print(f'\nSemilla 4: {semilla4}\n')
generar_numeros_aleatorios(semilla4, 3, 19, 23, 10)

print(f'\nSemilla 5: {semilla5}\n')
generar_numeros_aleatorios(semilla5, 31, 13, 19, 10)
```

Semilla 1: 99288

Paso: 0, Xn: 99288, Ui: ---

Paso: 1, Xn: 6, Ui: 0.5454545454545454

Paso: 2, Xn: 4, Ui: 0.36363636363636365

Paso: 3, Xn: 5, Ui: 0.45454545454545453

```
Paso: 3, Xn: 9, Ui: 0.4090909090909091
Paso: 4, Xn: 10, Ui: 0.9090909090909091
Paso: 5, Xn: 2, Ui: 0.18181818181818182
Paso: 6, Xn: 6, Ui: 0.5454545454545454
Paso: 7, Xn: 4, Ui: 0.36363636363636365
Paso: 8, Xn: 5, Ui: 0.45454545454545453
Paso: 9, Xn: 10, Ui: 0.9090909090909091
```

Semilla 2: 68054

```
Paso: 0, Xn: 68054, Ui: ---
Paso: 1, Xn: 1, Ui: 0.33333333333333333
Paso: 2, Xn: 0, Ui: 0.0
Paso: 3, Xn: 2, Ui: 0.66666666666666666
Paso: 4, Xn: 1, Ui: 0.33333333333333333
Paso: 5, Xn: 0, Ui: 0.0
Paso: 6, Xn: 2, Ui: 0.66666666666666666
Paso: 7, Xn: 1, Ui: 0.33333333333333333
Paso: 8, Xn: 0, Ui: 0.0
Paso: 9, Xn: 2, Ui: 0.66666666666666666
```

Semilla 3: 8166275072

```
Paso: 0, Xn: 8166275072, Ui: ---
Paso: 1, Xn: 7, Ui: 0.6363636363636364
Paso: 2, Xn: 0, Ui: 0.0
Paso: 3, Xn: 2, Ui: 0.18181818181818182
Paso: 4, Xn: 3, Ui: 0.2727272727272727
Paso: 5, Xn: 9, Ui: 0.8181818181818182
Paso: 6, Xn: 1, Ui: 0.09090909090909091
Paso: 7, Xn: 8, Ui: 0.7272727272727273
Paso: 8, Xn: 6, Ui: 0.5454545454545454
Paso: 9, Xn: 5, Ui: 0.45454545454545453
```

Semilla 4: 18382094336

```
Paso: 0, Xn: 18382094336, Ui: ---
Paso: 1, Xn: 10, Ui: 0.43478260869565216
Paso: 2, Xn: 3, Ui: 0.13043478260869565
Paso: 3, Xn: 5, Ui: 0.21739130434782608
Paso: 4, Xn: 11, Ui: 0.4782608695652174
Paso: 5, Xn: 6, Ui: 0.2608695652173913
Paso: 6, Xn: 14, Ui: 0.6086956521739131
Paso: 7, Xn: 15, Ui: 0.6521739130434783
Paso: 8, Xn: 18, Ui: 0.782608695652174
Paso: 9, Xn: 4, Ui: 0.17391304347826086
```

Semilla 5: 1018554

```
Paso: 0, Xn: 1018554, Ui: ---
Paso: 1, Xn: 18, Ui: 0.9473684210526315
Paso: 2, Xn: 1, Ui: 0.05263157894736842
Paso: 3, Xn: 6, Ui: 0.3157894736842105
Paso: 4, Xn: 9, Ui: 0.47368421052631576
Paso: 5, Xn: 7, Ui: 0.3684210526315789
Paso: 6, Xn: 2, Ui: 0.10526315789473684
Paso: 7, Xn: 18, Ui: 0.9473684210526315
Paso: 8, Xn: 1, Ui: 0.05263157894736842
Paso: 9, Xn: 6, Ui: 0.3157894736842105
```

3. Usando semillas definidas por el estudiante

In [11]:

```
#Usaremos las siguientes semillas
semilla1 = 23455
semilla2 = 66613
semilla3 = 29449645
semilla4 = 758293
semilla5 = 92747176

print(f'Semilla 1: {semilla1}\n')
generar_numeros_aleatorios(semilla1, 5, 7, 11, 10)

print(f'\nSemilla 2: {semilla2}\n')
```

```
generar_numeros_aleatorios(semilla2, 7, 11, 3, 10)

print(f'\nSemilla 3: {semilla3}\n')
generar_numeros_aleatorios(semilla3, 17, 13, 11, 10)

print(f'\nSemilla 4: {semilla4}\n')
generar_numeros_aleatorios(semilla4, 3, 19, 23, 10)

print(f'\nSemilla 5: {semilla5}\n')
generar_numeros_aleatorios(semilla5, 31, 13, 19, 10)
```

Semilla 1: 23455

Paso: 0, Xn: 23455, Ui: ---
Paso: 1, Xn: 0, Ui: 0.0
Paso: 2, Xn: 7, Ui: 0.6363636363636364
Paso: 3, Xn: 9, Ui: 0.8181818181818182
Paso: 4, Xn: 8, Ui: 0.7272727272727273
Paso: 5, Xn: 3, Ui: 0.2727272727272727
Paso: 6, Xn: 0, Ui: 0.0
Paso: 7, Xn: 7, Ui: 0.6363636363636364
Paso: 8, Xn: 9, Ui: 0.8181818181818182
Paso: 9, Xn: 8, Ui: 0.7272727272727273

Semilla 2: 66613

Paso: 0, Xn: 66613, Ui: ---
Paso: 1, Xn: 0, Ui: 0.0
Paso: 2, Xn: 2, Ui: 0.6666666666666666
Paso: 3, Xn: 1, Ui: 0.3333333333333333
Paso: 4, Xn: 0, Ui: 0.0
Paso: 5, Xn: 2, Ui: 0.6666666666666666
Paso: 6, Xn: 1, Ui: 0.3333333333333333
Paso: 7, Xn: 0, Ui: 0.0
Paso: 8, Xn: 2, Ui: 0.6666666666666666
Paso: 9, Xn: 1, Ui: 0.3333333333333333

Semilla 3: 29449645

Paso: 0, Xn: 29449645, Ui: ---
Paso: 1, Xn: 10, Ui: 0.9090909090909091
Paso: 2, Xn: 7, Ui: 0.6363636363636364
Paso: 3, Xn: 0, Ui: 0.0
Paso: 4, Xn: 2, Ui: 0.1818181818181818
Paso: 5, Xn: 3, Ui: 0.2727272727272727
Paso: 6, Xn: 9, Ui: 0.8181818181818182
Paso: 7, Xn: 1, Ui: 0.0909090909090909
Paso: 8, Xn: 8, Ui: 0.7272727272727273
Paso: 9, Xn: 6, Ui: 0.5454545454545454

Semilla 4: 758293

Paso: 0, Xn: 758293, Ui: ---
Paso: 1, Xn: 14, Ui: 0.6086956521739131
Paso: 2, Xn: 15, Ui: 0.6521739130434783
Paso: 3, Xn: 18, Ui: 0.782608695652174
Paso: 4, Xn: 4, Ui: 0.17391304347826086
Paso: 5, Xn: 8, Ui: 0.34782608695652173
Paso: 6, Xn: 20, Ui: 0.8695652173913043
Paso: 7, Xn: 10, Ui: 0.43478260869565216
Paso: 8, Xn: 3, Ui: 0.13043478260869565
Paso: 9, Xn: 5, Ui: 0.21739130434782608

Semilla 5: 92747176

Paso: 0, Xn: 92747176, Ui: ---
Paso: 1, Xn: 9, Ui: 0.47368421052631576
Paso: 2, Xn: 7, Ui: 0.3684210526315789
Paso: 3, Xn: 2, Ui: 0.10526315789473684
Paso: 4, Xn: 18, Ui: 0.9473684210526315
Paso: 5, Xn: 1, Ui: 0.05263157894736842
Paso: 6, Xn: 6, Ui: 0.3157894736842105
Paso: 7, Xn: 9, Ui: 0.47368421052631576
Paso: 8, Xn: 7, Ui: 0.3684210526315789
Paso: 9, Xn: 2, Ui: 0.10526315789473684

4. Conclusiones

Mediante el presente trabajo se ha implementado y usado el método por congruencia lineal para la generación de números pseudo-aleatorios. El método de congruencia lineal genera números aleatorios dentro de un periodo, los cuales se repiten cada periodo. La primera parte toma como semillas la información referente a los procesos de entrada y salida del disco duro, mientras que en la segunda se establecen de forma arbitraria por el usuario.