

Números pseudo-aleatorios por medios cuadrados

AUTOR: CARLOS MOROCHO

In [11]:

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

%matplotlib inline
```

1. Definiendo la función de generación

In [34]:

```
def generar_numeros_aleatorios(x0, iteraciones, total_digitos):
    xn = x0

    for i in range(iteraciones):
        xn_pow = str(int(math.pow(int(xn), 2)))
        longitud = len(xn_pow)
        centro = math.ceil(longitud / 2)

        if total_digitos % 2 == 0:
            porcion_izquierda = int(total_digitos / 2)
            porcion_derecha = int(total_digitos / 2)
        else:
            porcion_izquierda = math.floor(total_digitos / 2)
            porcion_derecha = int(total_digitos - porcion_izquierda)

        aux = xn_pow[centro - porcion_izquierda:centro] + xn_pow[centro:centro + porcion_derecha]
        ui = int(aux) / math.pow(10, total_digitos)

        print(f'I: {i + 1}, Xn: {xn}, Xn**2: {xn_pow}, Longitud: {longitud}, Semilla: {aux}, Ui: {ui}')

        xn = aux

    if aux == 0.0:
        break
```

2. Usando información del Hardware como semilla inicial

In [41]:

```
#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(str(semilla1), 10, 4)

print(f'\nSemilla 2: {semilla2}\n')
generar_numeros_aleatorios(str(semilla2), 10, 4)

print(f'\nSemilla 3: {semilla3}\n')
generar_numeros_aleatorios(str(semilla3), 10, 4)

print(f'\nSemilla 4: {semilla4}\n')
generar_numeros_aleatorios(str(semilla4), 10, 4)
```

```
print(f'\nSemilla 5: {semilla5}\n')
generar_numeros_aleatorios(str(semilla5), 10, 4)
```

Semilla 1: 27931

```
I: 1, Xn: 27931, Xn**2: 780140761, Longitud: 9, Semilla: 1407, Ui: 0.1407
I: 2, Xn: 1407, Xn**2: 1979649, Longitud: 7, Semilla: 7964, Ui: 0.7964
I: 3, Xn: 7964, Xn**2: 63425296, Longitud: 8, Semilla: 4252, Ui: 0.4252
I: 4, Xn: 4252, Xn**2: 18079504, Longitud: 8, Semilla: 0795, Ui: 0.0795
I: 5, Xn: 0795, Xn**2: 632025, Longitud: 6, Semilla: 3202, Ui: 0.3202
I: 6, Xn: 3202, Xn**2: 10252804, Longitud: 8, Semilla: 2528, Ui: 0.2528
I: 7, Xn: 2528, Xn**2: 6390784, Longitud: 7, Semilla: 9078, Ui: 0.9078
I: 8, Xn: 9078, Xn**2: 82410084, Longitud: 8, Semilla: 4100, Ui: 0.41
I: 9, Xn: 4100, Xn**2: 16810000, Longitud: 8, Semilla: 8100, Ui: 0.81
I: 10, Xn: 8100, Xn**2: 65610000, Longitud: 8, Semilla: 6100, Ui: 0.61
```

Semilla 2: 39100

```
I: 1, Xn: 39100, Xn**2: 1528810000, Longitud: 10, Semilla: 8810, Ui: 0.881
I: 2, Xn: 8810, Xn**2: 77616100, Longitud: 8, Semilla: 6161, Ui: 0.6161
I: 3, Xn: 6161, Xn**2: 37957921, Longitud: 8, Semilla: 9579, Ui: 0.9579
I: 4, Xn: 9579, Xn**2: 91757241, Longitud: 8, Semilla: 7572, Ui: 0.7572
I: 5, Xn: 7572, Xn**2: 57335184, Longitud: 8, Semilla: 3351, Ui: 0.3351
I: 6, Xn: 3351, Xn**2: 11229201, Longitud: 8, Semilla: 2292, Ui: 0.2292
I: 7, Xn: 2292, Xn**2: 5253264, Longitud: 7, Semilla: 5326, Ui: 0.5326
I: 8, Xn: 5326, Xn**2: 28366276, Longitud: 8, Semilla: 3662, Ui: 0.3662
I: 9, Xn: 3662, Xn**2: 13410244, Longitud: 8, Semilla: 4102, Ui: 0.4102
I: 10, Xn: 4102, Xn**2: 16826404, Longitud: 8, Semilla: 8264, Ui: 0.8264
```

Semilla 3: 1337814016

```
I: 1, Xn: 1337814016, Xn**2: 1789746341406048256, Longitud: 19, Semilla: 4140, Ui: 0.414
I: 2, Xn: 4140, Xn**2: 17139600, Longitud: 8, Semilla: 1396, Ui: 0.1396
I: 3, Xn: 1396, Xn**2: 1948816, Longitud: 7, Semilla: 4881, Ui: 0.4881
I: 4, Xn: 4881, Xn**2: 23824161, Longitud: 8, Semilla: 8241, Ui: 0.8241
I: 5, Xn: 8241, Xn**2: 67914081, Longitud: 8, Semilla: 9140, Ui: 0.914
I: 6, Xn: 9140, Xn**2: 83539600, Longitud: 8, Semilla: 5396, Ui: 0.5396
I: 7, Xn: 5396, Xn**2: 29116816, Longitud: 8, Semilla: 1168, Ui: 0.1168
I: 8, Xn: 1168, Xn**2: 1364224, Longitud: 7, Semilla: 6422, Ui: 0.6422
I: 9, Xn: 6422, Xn**2: 41242084, Longitud: 8, Semilla: 2420, Ui: 0.242
I: 10, Xn: 2420, Xn**2: 5856400, Longitud: 7, Semilla: 5640, Ui: 0.564
```

Semilla 4: 879198208

```
I: 1, Xn: 879198208, Xn**2: 772989488950411264, Longitud: 18, Semilla: 8895, Ui: 0.8895
I: 2, Xn: 8895, Xn**2: 79121025, Longitud: 8, Semilla: 1210, Ui: 0.121
I: 3, Xn: 1210, Xn**2: 1464100, Longitud: 7, Semilla: 6410, Ui: 0.641
I: 4, Xn: 6410, Xn**2: 41088100, Longitud: 8, Semilla: 0881, Ui: 0.0881
I: 5, Xn: 0881, Xn**2: 776161, Longitud: 6, Semilla: 7616, Ui: 0.7616
I: 6, Xn: 7616, Xn**2: 58003456, Longitud: 8, Semilla: 0034, Ui: 0.0034
I: 7, Xn: 0034, Xn**2: 1156, Longitud: 4, Semilla: 1156, Ui: 0.1156
I: 8, Xn: 1156, Xn**2: 1336336, Longitud: 7, Semilla: 3633, Ui: 0.3633
I: 9, Xn: 3633, Xn**2: 13198689, Longitud: 8, Semilla: 1986, Ui: 0.1986
I: 10, Xn: 1986, Xn**2: 3944196, Longitud: 7, Semilla: 4419, Ui: 0.4419
```

Semilla 5: 630582

```
I: 1, Xn: 630582, Xn**2: 397633658724, Longitud: 12, Semilla: 3365, Ui: 0.3365
I: 2, Xn: 3365, Xn**2: 11323225, Longitud: 8, Semilla: 3232, Ui: 0.3232
I: 3, Xn: 3232, Xn**2: 10445824, Longitud: 8, Semilla: 4458, Ui: 0.4458
I: 4, Xn: 4458, Xn**2: 19873764, Longitud: 8, Semilla: 8737, Ui: 0.8737
I: 5, Xn: 8737, Xn**2: 76335169, Longitud: 8, Semilla: 3351, Ui: 0.3351
I: 6, Xn: 3351, Xn**2: 11229201, Longitud: 8, Semilla: 2292, Ui: 0.2292
I: 7, Xn: 2292, Xn**2: 5253264, Longitud: 7, Semilla: 5326, Ui: 0.5326
I: 8, Xn: 5326, Xn**2: 28366276, Longitud: 8, Semilla: 3662, Ui: 0.3662
I: 9, Xn: 3662, Xn**2: 13410244, Longitud: 8, Semilla: 4102, Ui: 0.4102
I: 10, Xn: 4102, Xn**2: 16826404, Longitud: 8, Semilla: 8264, Ui: 0.8264
```

3. Usando semillas definidas por el estudiante

In [42]:

```
#Usaremos las siguientes semillas
semilla1 = 23455
```

```

semilla1 = 23455
semilla2 = 66613
semilla3 = 29449645
semilla4 = 758293
semilla5 = 92747176

print(f'Semilla 1: {semilla1}\n')
generar_numeros_aleatorios(str(semilla1), 10, 4)

print(f'\nSemilla 2: {semilla2}\n')
generar_numeros_aleatorios(str(semilla2), 10, 4)

print(f'\nSemilla 3: {semilla3}\n')
generar_numeros_aleatorios(str(semilla3), 10, 4)

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

print(f'\nSemilla 5: {semilla5}\n')
generar_numeros_aleatorios(str(semilla5), 10, 4)

```

Semilla 1: 23455

```

I: 1, Xn: 23455, Xn**2: 550137025, Longitud: 9, Semilla: 1370, Ui: 0.137
I: 2, Xn: 1370, Xn**2: 1876900, Longitud: 7, Semilla: 7690, Ui: 0.769
I: 3, Xn: 7690, Xn**2: 59136100, Longitud: 8, Semilla: 1361, Ui: 0.1361
I: 4, Xn: 1361, Xn**2: 1852321, Longitud: 7, Semilla: 5232, Ui: 0.5232
I: 5, Xn: 5232, Xn**2: 27373824, Longitud: 8, Semilla: 3738, Ui: 0.3738
I: 6, Xn: 3738, Xn**2: 13972644, Longitud: 8, Semilla: 9726, Ui: 0.9726
I: 7, Xn: 9726, Xn**2: 94595076, Longitud: 8, Semilla: 5950, Ui: 0.595
I: 8, Xn: 5950, Xn**2: 35402500, Longitud: 8, Semilla: 4025, Ui: 0.4025
I: 9, Xn: 4025, Xn**2: 16200625, Longitud: 8, Semilla: 2006, Ui: 0.2006
I: 10, Xn: 2006, Xn**2: 4024036, Longitud: 7, Semilla: 2403, Ui: 0.2403

```

Semilla 2: 66613

```

I: 1, Xn: 66613, Xn**2: 4437291769, Longitud: 10, Semilla: 7291, Ui: 0.7291
I: 2, Xn: 7291, Xn**2: 53158681, Longitud: 8, Semilla: 1586, Ui: 0.1586
I: 3, Xn: 1586, Xn**2: 2515396, Longitud: 7, Semilla: 1539, Ui: 0.1539
I: 4, Xn: 1539, Xn**2: 2368521, Longitud: 7, Semilla: 6852, Ui: 0.6852
I: 5, Xn: 6852, Xn**2: 46949904, Longitud: 8, Semilla: 9499, Ui: 0.9499
I: 6, Xn: 9499, Xn**2: 90231001, Longitud: 8, Semilla: 2310, Ui: 0.231
I: 7, Xn: 2310, Xn**2: 5336100, Longitud: 7, Semilla: 3610, Ui: 0.361
I: 8, Xn: 3610, Xn**2: 13032100, Longitud: 8, Semilla: 0321, Ui: 0.0321
I: 9, Xn: 0321, Xn**2: 103041, Longitud: 6, Semilla: 0304, Ui: 0.0304
I: 10, Xn: 0304, Xn**2: 92416, Longitud: 5, Semilla: 2416, Ui: 0.2416

```

Semilla 3: 29449645

```

I: 1, Xn: 29449645, Xn**2: 867281590626025, Longitud: 15, Semilla: 5906, Ui: 0.5906
I: 2, Xn: 5906, Xn**2: 34880836, Longitud: 8, Semilla: 8808, Ui: 0.8808
I: 3, Xn: 8808, Xn**2: 77580864, Longitud: 8, Semilla: 5808, Ui: 0.5808
I: 4, Xn: 5808, Xn**2: 33732864, Longitud: 8, Semilla: 7328, Ui: 0.7328
I: 5, Xn: 7328, Xn**2: 53699584, Longitud: 8, Semilla: 6995, Ui: 0.6995
I: 6, Xn: 6995, Xn**2: 48930025, Longitud: 8, Semilla: 9300, Ui: 0.93
I: 7, Xn: 9300, Xn**2: 86490000, Longitud: 8, Semilla: 4900, Ui: 0.49
I: 8, Xn: 4900, Xn**2: 24010000, Longitud: 8, Semilla: 0100, Ui: 0.01
I: 9, Xn: 0100, Xn**2: 10000, Longitud: 5, Semilla: 0000, Ui: 0.0
I: 10, Xn: 0000, Xn**2: 0, Longitud: 1, Semilla: 0, Ui: 0.0

```

Semilla 4: 758293

```

I: 1, Xn: 758293, Xn**2: 575008273849, Longitud: 12, Semilla: 0827, Ui: 0.0827
I: 2, Xn: 0827, Xn**2: 683929, Longitud: 6, Semilla: 8392, Ui: 0.8392
I: 3, Xn: 8392, Xn**2: 70425664, Longitud: 8, Semilla: 4256, Ui: 0.4256
I: 4, Xn: 4256, Xn**2: 18113536, Longitud: 8, Semilla: 1135, Ui: 0.1135
I: 5, Xn: 1135, Xn**2: 1288225, Longitud: 7, Semilla: 8822, Ui: 0.8822
I: 6, Xn: 8822, Xn**2: 77827684, Longitud: 8, Semilla: 8276, Ui: 0.8276
I: 7, Xn: 8276, Xn**2: 68492176, Longitud: 8, Semilla: 4921, Ui: 0.4921
I: 8, Xn: 4921, Xn**2: 24216241, Longitud: 8, Semilla: 2162, Ui: 0.2162
I: 9, Xn: 2162, Xn**2: 4674244, Longitud: 7, Semilla: 7424, Ui: 0.7424
I: 10, Xn: 7424, Xn**2: 55115776, Longitud: 8, Semilla: 1157, Ui: 0.1157

```

Semilla 5: 92747176

```

I: 1, Xn: 92747176, Xn**2: 8602038655974976, Longitud: 16, Semilla: 8655, Ui: 0.8655
I: 2, Xn: 8655, Xn**2: 74909025, Longitud: 8, Semilla: 9090, Ui: 0.9090

```

```
I: 2, Xn: 8090, Xn**2: 65448100, Longitud: 8, Semilla: 8090, Ui: 0.654481
I: 3, Xn: 9090, Xn**2: 82628100, Longitud: 8, Semilla: 6281, Ui: 0.6281
I: 4, Xn: 6281, Xn**2: 39450961, Longitud: 8, Semilla: 4509, Ui: 0.4509
I: 5, Xn: 4509, Xn**2: 20331081, Longitud: 8, Semilla: 3310, Ui: 0.331
I: 6, Xn: 3310, Xn**2: 10956100, Longitud: 8, Semilla: 9561, Ui: 0.9561
I: 7, Xn: 9561, Xn**2: 91412721, Longitud: 8, Semilla: 4127, Ui: 0.4127
I: 8, Xn: 4127, Xn**2: 17032129, Longitud: 8, Semilla: 0321, Ui: 0.0321
I: 9, Xn: 0321, Xn**2: 103041, Longitud: 6, Semilla: 0304, Ui: 0.0304
I: 10, Xn: 0304, Xn**2: 92416, Longitud: 5, Semilla: 2416, Ui: 0.2416
```

4. Conclusiones

El método por medios cuadrados para la generación de números pseudo-aleatorios. 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.

Gracias a los números pseudo-aleatorios se puede dar más realidad a los entornos de simulación, aunque cabe recordar que el método por medios cuadrados se encuentra mayormente en desuso.

In []: