

UNIVERSIDAD POLITÉCNICA SALESIANA

EI VECINO - CUENCA

Estudiante: Gustavo Gualpa

Profesor: Ing. Diego Quisi

Asignatura: Simulación

Fecha: 20/01/2020.

Tema: Generación de Números Pseudoaleatorios.

```
In [7]: #Importacion de librerias.  
import collections
```

```
In [8]: def valuesDiv(long,numeroOriginal,digitos):  
        numero=int((long/2)-(digitos/2)+1)  
        sNumero=str(numeroOriginal)  
        numUI=sNumero[numero-1:numero+3]  
        return int(numUI)
```

```
In [9]: def searchSeed(valor,lista):  
        #print('Xn',valor)  
        for x in range(0,len(lista)):  
            #print(lista[x])  
            if (lista[x]==valor):  
                return True
```

```

In [10]: def generarNumeroPseudoaleatorios(iteraciones,semilla):
    #Declaracion de la variables.
    iteraciones=iteraciones
    #Semilla
    Xo=semilla
    digitos=4
    #Valores para la tabla
    Xn=0
    XnPo=0
    longitud=0
    Ui=0
    Rn=0
    #Areglo de todas la semillas
    arregloSemillas=[]
    semillasRepetidas=[]

    #Verificar primero que Xo sea mayor a 3.
    if len(str(Xo))>3:
        print('PROCESO NORMAL')
        for i in range(iteraciones):
            #Cuando es cero se empieza con la semilla inicial
            if (i==0):
                Xn=Xo
                XnPo=Xn**2
                longitud=len(str(XnPo))
                Ui=valuesDiv(longitud,XnPo,digitos)
                #Agregamos al arreglo de semillas
                #print('Agregamos',Xn)
                arregloSemillas.append(Xn)
                #Valor
                Rn=Ui/10000
                print(Rn)
            #Cuando es diferente de cero, la semilla es el valor de Ui.
            else:
                Xn=Ui
                XnPo=Xn**2
                longitud=len(str(XnPo))
                Ui=valuesDiv(longitud,XnPo,digitos)
                #print('Agregamos',Xn)
                #or o in arregloSemillas:
                # print(o)
                Rn=Ui/10000
                print(Rn)
                #Buscamos el numero repetido.
                busquedaSemilla=searchSeed(Xn,arregloSemillas)
                if(busquedaSemilla==True):
                    #Agregamos la semilla repetida
                    semillasRepetidas.append(Xn)
                    print('LA SEMILLA SE VUELE A REPETIR EN LA ITERACION:',i,'VAL')
                    #break;
                #Agregamos la semilla
                arregloSemillas.append(Xn)

    else:
        print('LA SEMILLA ES MENOR A 3')

```

```
print('LA FRECUENCIA DE REPETICION ES DE:')  
#Vemos la frecuencia de repeticion  
counter=collections.Counter(semillasRepetidas)  
print(counter)
```

GENERACION DE NUMERO PSEUDOALEATORIOS, NUMERO DIFERENTES

NÚMERO 1

```
In [11]: iteraciones=70  
         semilla=8370  
         generarNumeroPseudoaleatorios(iteraciones,semilla)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474
0.0166

```
0.2755
0.59
0.81
0.61
0.21
0.41
0.81
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100
0.21
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 60 VALOR: 6100
0.41
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 61 VALOR: 2100
0.81
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 62 VALOR: 4100
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 63 VALOR: 8100
0.21
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 64 VALOR: 6100
0.41
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 65 VALOR: 2100
0.81
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 66 VALOR: 4100
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 67 VALOR: 8100
0.21
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 68 VALOR: 6100
0.41
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 69 VALOR: 2100
LA FRECUENCIA DE REPETICION ES DE:
Counter({8100: 3, 6100: 3, 2100: 3, 4100: 2})
```

NÚMERO 2

```
In [12]: iteraciones=50  
         semilla=8370  
         generarNumeroPseudoaleatorios(iteraciones,semilla)
```

PROCESO NORMAL

0.0569

0.2376

0.6453

0.6412

0.1137

0.2927

0.5673

0.1829

0.3452

0.9163

0.9605

0.256

0.5536

0.6472

0.8867

0.6236

0.8876

0.7833

0.3558

0.6593

0.4676

0.8649

0.8052

0.8347

0.6724

0.2121

0.4986

0.8601

0.9772

0.4919

0.1965

0.8612

0.1665

0.7722

0.6292

0.5892

0.7156

0.2083

0.3388

0.4785

0.8962

0.3174

0.0742

0.5056

0.5631

0.7081

0.1405

0.974

0.8676

0.2729

LA FRECUENCIA DE REPETICION ES DE:

Counter()

NÚMERO 3

```
In [13]: iteraciones=10  
semilla=8370  
generarNumeroPseudoaleatorios(iteraciones,semilla)
```

PROCESO NORMAL

0.0569

0.2376

0.6453

0.6412

0.1137

0.2927

0.5673

0.1829

0.3452

0.9163

LA FRECUENCIA DE REPETICION ES DE:

Counter()

NÚMERO 4

```
In [14]: iteraciones=5  
semilla=8370  
generarNumeroPseudoaleatorios(iteraciones,semilla)
```

PROCESO NORMAL

0.0569

0.2376

0.6453

0.6412

0.1137

LA FRECUENCIA DE REPETICION ES DE:

Counter()

NÚMERO 5

```
In [15]: iteraciones=35
          semilla=8370
          generarNumeroPseudoaleatorios(iteraciones,semilla)
```

PROCESO NORMAL

0.0569

0.2376

0.6453

0.6412

0.1137

0.2927

0.5673

0.1829

0.3452

0.9163

0.9605

0.256

0.5536

0.6472

0.8867

0.6236

0.8876

0.7833

0.3558

0.6593

0.4676

0.8649

0.8052

0.8347

0.6724

0.2121

0.4986

0.8601

0.9772

0.4919

0.1965

0.8612

0.1665

0.7722

0.6292

LA FRECUENCIA DE REPETICION ES DE:

Counter()

GENERACION DE NUMERO PSEUDOALEATORIOS, DIFERENTES TECNICAS.

```
In [16]: import psutil
          import os
          # Indicamos la ruta del disco.
          disk_usage = psutil.disk_usage("C:\\\\")
```



```
In [17]: def memory_usage_psutil():  
    # return the memory usage in MB  
    import psutil  
    import os  
    process = psutil.Process(os.getpid())  
    mem = process.memory_info().rss / float(2 ** 20)  
    return mem
```

```

In [18]: def generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo):
#Declaracion de la variables.
iteraciones=iteraciones
#Semilla
Xo=semilla
digitos=4
#Valores para la tabla
Xn=0
XnPo=0
longuitud=0
Ui=0
Rn=0
#Areglo de todas la semillas
arregloSemillas=[]
semillasRepetidas=[]

#Verificar primero que Xo sea mayor a 3.
if len(str(Xo))>3:
    print('PROCESO NORMAL')
    for i in range(iteraciones):
        #Cuando es cero se empieza con la semilla inicial
        if (i==0):
            Xn=Xo
            XnPo=Xn**2
            longuitud=len(str(XnPo))
            Ui=valuesDiv(longuitud,XnPo,digitos)
            #Agregamos al arreglo de semillas
            #print('Agregamos',Xn)
            arregloSemillas.append(Xn)
            #Valor
            Rn=Ui/10000
            print(Rn)
        #Cuando es diferente de cero, la semilla es el valor de Ui.
        else:
            Xn=Ui
            XnPo=Xn**2
            longuitud=len(str(XnPo))
            Ui=valuesDiv(longuitud,XnPo,digitos)
            #print('Agregamos',Xn)
            #or o in arregloSemillas:
            #    print(o)
            Rn=Ui/10000
            print(Rn)
            #Buscamos el numero repetido.
            busquedaSemilla=searchSeed(Xn,arregloSemillas)
            if(busquedaSemilla==True):
                #Agregamos la semilla repetida
                semillasRepetidas.append(Xn)
                print('LA SEMILLA SE VUELE A REPETIR EN LA ITERACION:',i,'VAL')
                if tipo=='total':
                    nuevo = disk_usage.total
                    Ui=nuevo
                    print('CAMBIO DE SEMILLA TOTAL,LA NUEVA ES:', Ui)
                if tipo=='libre':
                    nuevo = disk_usage.free

```

```
        Ui=nuevo
        print('CAMBIO DE SEMILLA LIBRE,LA NUEVA ES:', Ui)
    if tipo=='usado':
        nuevo = disk_usage.used
        Ui=nuevo
        print('CAMBIO DE SEMILLA USADO,LA NUEVA ES:', Ui)
    if tipo=='memory':
        numero = memory_usage_psutil()#memorya en MB
        nuevo = (numero*1024)*1024#Memory en bytes
        Ui=nuevo
        print('CAMBIO DE SEMILLA MEMORIA,LA NUEVA ES:', Ui)
    if tipo=='memory2':
        numero = memory_usage_psutil()#memorya en MB
        nuevo = (numero*1024)
        Ui=nuevo
        print('CAMBIO DE SEMILLA MEMORIA 2,LA NUEVA ES:', Ui)

    #Agregamos la semilla
    arregloSemillas.append(Xn)

else:
    print('LA SEMILLA ES MENOR A 3')

print('LA FRECUENCIA DE REPETICION ES DE:')
#Vemos la frecuencia de repeticion
counter=collections.Counter(semillasRepetidas)
print(counter)
```

CON EL ESPACIO TOTAL DE DISCO

```
In [19]: iteraciones=70  
         semilla=8370  
         tipo='total'  
         generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474

```
0.0166
0.2755
0.59
0.81
0.61
0.21
0.41
0.81
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100
CAMBIO DE SEMILLA TOTAL, LA NUEVA ES: 510771392512
0.0864
0.4649
0.6132
0.6014
0.1681
0.8257
0.178
0.1684
0.8358
0.8561
LA FRECUENCIA DE REPETICION ES DE:
Counter({8100: 1})
```

CON EL ESPACIO LIBRE DE DISCO

```
In [20]: iteraciones=70  
         semilla=8370  
         tipo='libre'  
         generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474

```
0.0166
0.2755
0.59
0.81
0.61
0.21
0.41
0.81
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100
CAMBIO DE SEMILLA LIBRE, LA NUEVA ES: 43682861056
0.0037
0.1369
0.8741
0.405
0.4025
0.2006
0.024
0.576
0.1776
0.1541
LA FRECUENCIA DE REPETICION ES DE:
Counter({8100: 1})
```

CON EL ESPACIO USADO DE DISCO

```
In [21]: iteraciones=70  
         semilla=8370  
         tipo='usado'  
         generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474

0.0166

0.2755

0.59

0.81

0.61

0.21

0.41

0.81

0.61

LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100

CAMBIO DE SEMILLA USADO, LA NUEVA ES: 467088531456

0.1772

0.1399

0.9572

0.6231

0.8253

0.112

0.2544

0.4719

0.2689

0.2307

LA FRECUENCIA DE REPETICION ES DE:

Counter({8100: 1})

TAMAÑO DE MEMORIA UTILIZADA EN bytes.

```
In [22]: iteraciones=70  
         semilla=8370  
         tipo='memory'  
         generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474

```
0.0166
0.2755
0.59
0.81
0.61
0.21
0.41
0.81
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100
CAMBIO DE SEMILLA MEMORIA, LA NUEVA ES: 54640640.0
0.5396
0.1168
0.3642
0.2641
0.9748
0.0235
0.5522
0.4924
0.2457
0.0368
LA FRECUENCIA DE REPETICION ES DE:
Counter({8100: 1})
```

TAMAÑO DE MEMORIA UTILIZADA 2.

```
In [24]: iteraciones=70  
         semilla=8370  
         tipo='memory2'  
         generarNumeroPseudoaleatoriosUso(iteraciones,semilla,tipo)
```

PROCESO NORMAL

0.0569
0.2376
0.6453
0.6412
0.1137
0.2927
0.5673
0.1829
0.3452
0.9163
0.9605
0.256
0.5536
0.6472
0.8867
0.6236
0.8876
0.7833
0.3558
0.6593
0.4676
0.8649
0.8052
0.8347
0.6724
0.2121
0.4986
0.8601
0.9772
0.4919
0.1965
0.8612
0.1665
0.7722
0.6292
0.5892
0.7156
0.2083
0.3388
0.4785
0.8962
0.3174
0.0742
0.5056
0.5631
0.7081
0.1405
0.974
0.8676
0.2729
0.4474

```
0.0166
0.2755
0.59
0.81
0.61
0.21
0.41
0.81
0.61
LA SEMILLA SE VUELE A REPETIR EN LA ITERACION: 59 VALOR: 8100
CAMBIO DE SEMILLA MEMORIA 2, LA NUEVA ES: 53348.0
0.0091
0.8281
0.5749
0.051
0.601
0.1201
0.4424
0.5717
0.684
0.7856
LA FRECUENCIA DE REPETICION ES DE:
Counter({8100: 1})
```

In []: