

DEBER GENERAR NUMEROS PSEUDO-ALEATORIOS CON COMPONENTES DEL COMPUTADOR

NOMBRE: Yupanki Malki

Se procede a gener numeros pseudoaleatorios tomando como semilla los valores de los componentes de la computadora como son la temperatura, cpu y la RAM.

Para poder obtener estos valores procedemos a utilizar dos librerias y definimos dos metodos para relizar los calculos necesarios para poder generar los numeros.

```
In [26]: 1 import wmi #
          2 import cpuinfo
          3

In [31]: 1 #Obtener temperatura
          2 w = wmi.WMI(namespace="root\\wmi")
          3 print(w.MSAcpi_ThermalZoneTemperature()[0])
          4 temperatura_pc = w.MSAcpi_ThermalZoneTemperature()[0].CurrentTemp
          5 #obtener numero de calculos flops
          6 OxC = 16 #operaciones de punto flotante por segundo del procesador
          7 procesadores = float(cpuinfo.get_cpu_info()['count'])
          8 velocidad = float(cpuinfo.get_cpu_info()['hz_advertised_friendly'])
          9 flops = procesadores * velocidad * OxC
         10 numero_flops=int(flops * 1024)
         11 print(numero_flops)
         12 #obtener uso de procesador y RAM
         13 uso_procesador=psutil.cpu_percent()
         14 ram=psutil.virtual_memory().total
         15 uso_ram=psutil.virtual_memory().used
         16 print(uso_procesador)
         17
         18 def cortes(num_digitos):
         19     digstos1 =0
         20     digitos2=0
         21     if num_digitos%2 !=0:
         22         digstos1 = int(num_digitos / 2)
         23         digitos2 = int(num_digitos / 2) + 1
         24     else:
         25         digstos1 = int(num_digitos / 2)
         26         digitos2 = int(num_digitos / 2)
         27     return digstos1,digitos2
         28
         29 def calculo_num(num_iteras, valor_inicio, num_digitos):
         30     semilla_xi = int(valor_inicio)
         31     aux = cortes(num_digitos)
         32     print("#-ITERACIÓN*", "*Xn*", "-Xn*Xn-", "-Longitud-", "-Ui-")
         33     for i in range(num_iteras):
         34         xn2= semilla_xi ** 2
         35         lon = len(str(xn2))
         36         ui = str(xn2)[int(lon/2)-aux[0]:int(lon/2)+aux[1]]
         37         rn = int(ui) / 10 ** num_digitos
         38         print(i, " ", semilla_xi, " ", xn2, " ", lon, " ", ui, " ")
         39         semilla_xi=int(ui)
```

```
40 print(" ")
```

```
instance of MSAcpi_ThermalZoneTemperature
{
    Active = TRUE;
    ActiveTripPoint = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    ActiveTripPointCount = 0;
    CriticalTripPoint = 3762;
    CurrentTemperature = 3242;
    InstanceName = "ACPI\\ThermalZone\\THRM_0";
    PassiveTripPoint = 3832;
    Reserved = 0;
    SamplingPeriod = 0;
    ThermalConstant1 = 0;
    ThermalConstant2 = 0;
    ThermalStamp = 40;
};

432537
24.7
```

```
In [32]: 1 valores_iniciales =[temperatura_pc, numero_flops, uso_procesador,
2 iteraciones = 16
3 digitos_iniciales = 4
4 for i in valores_iniciales:
5     print("i: ", i)
6     calculo_num(iteraciones, i, digitos_iniciales)
```

```
i: 3242
*#-ITERACIÓN* *Xn* -Xn*Xn- -Longitud- -Ui- -Rn-
0 3242 10510564 8 5105 0.5105
1 5105 26061025 8 0610 0.061
2 610 372100 6 7210 0.721
3 7210 51984100 8 9841 0.9841
4 9841 96845281 8 8452 0.8452
5 8452 71436304 8 4363 0.4363
6 4363 19035769 8 0357 0.0357
7 357 127449 6 2744 0.2744
8 2744 7529536 7 5295 0.5295
9 5295 28037025 8 0370 0.037
10 370 136900 6 3690 0.369
11 3690 13616100 8 6161 0.6161
12 6161 37957921 8 9579 0.9579
13 9579 91757241 8 7572 0.7572
14 7572 57335184 8 3351 0.3351
15 3351 11229201 8 2292 0.2292
```

```
In [ ]:
```