

Cuadrados Medios

Materia:

SIMULACION

Docentes:

Ing. Diego Quisi

Estudiante:

Ricardo Jara

Generar 10 listas de 100 numeros arelaotorios con 5 semillas con datos relacionados al sistema y 5 semillas personales.

```
In [11]: #Importamos Librerias
import matplotlib.pyplot as plt
import pandas as pd
import psutil
```

- Método para indica cuantas posiciones hacia adelante o atrás puedo correr en base a la mitad de un numero para sacar una cantidad de números

```
In [2]: def medios(digs):
        if digs % 2 != 0:
            return int(digs / 2), int(digs / 2) + 1
        else:
            return int(digs / 2), int(digs / 2)
```

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

```
In [9]: def cuadrados_medios(iters, semilla, dig):
        lista = []
        posI, posF = medios(dig)
```

```
num = int(semilla)
for i in range(iters):
    num = num ** 2
    tam = len(str(num))
    centro = str(num)[int(tam / 2) - posI: int(tam / 2) + posF]
    lista.append(int(centro))
    num = int(centro)
return lista
```

- Semillas

```
#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,
```

```
cantidad = 100
num_digi = 5
d = {}

#calculamos la Lista de numeros randomicos
for l in semillas:
    lista = cuadrados_medios(cantidad, semilla1, 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)
```

[illegible]

[illegible]

[illegible]

	20202020	20212021	19970827	20180805	19961207	17027518464	1505	25493	833836	3
85	81733	81733	81733	81733	81733	81733	81733	81733	81733	
86	2832	2832	2832	2832	2832	2832	2832	2832	2832	
87	2022	2022	2022	2022	2022	2022	2022	2022	2022	
88	8848	8848	8848	8848	8848	8848	8848	8848	8848	
89	28710	28710	28710	28710	28710	28710	28710	28710	28710	
90	42641	42641	42641	42641	42641	42641	42641	42641	42641	
91	82548	82548	82548	82548	82548	82548	82548	82548	82548	
92	41723	41723	41723	41723	41723	41723	41723	41723	41723	
93	8087	8087	8087	8087	8087	8087	8087	8087	8087	
94	39956	39956	39956	39956	39956	39956	39956	39956	39956	
95	64819	64819	64819	64819	64819	64819	64819	64819	64819	
96	15027	15027	15027	15027	15027	15027	15027	15027	15027	
97	58107	58107	58107	58107	58107	58107	58107	58107	58107	
98	64234	64234	64234	64234	64234	64234	64234	64234	64234	
99	60067	60067	60067	60067	60067	60067	60067	60067	60067	