Práctica 2

Sergio Guachalla

Método congruencial lineal

Ejercicio 1: Se pide generar números aleatorios

Datos:

$$c = 89; \quad x_0 = 5; \quad m = 10^2; \quad a = 81$$

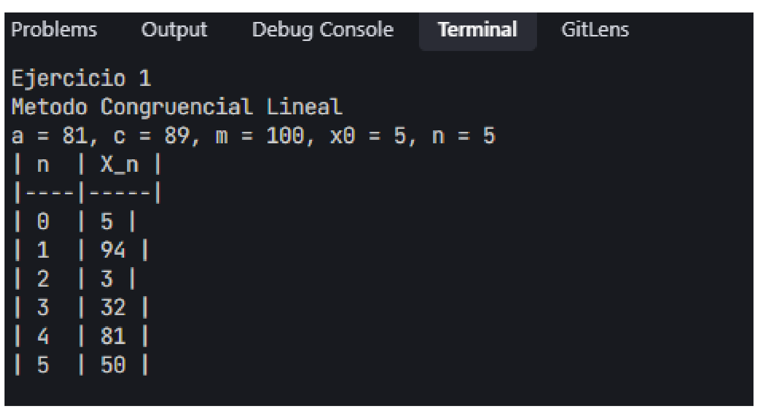
La fórmula para el método congruencial lineal es:

$$x_{i+1} = (ax_i + c) \mod m$$

Para n números aleatorios:

n	X_n
0	5
1	$(81 \times 5 + 89) \mod 100 = 94$
2	$(81 \times 94 + 89) \mod 100 = 3$
3	$(81 \times 3 + 89) \mod 100 = 32$
4	$(81 \times 32 + 89) \mod 100 = 81$
5	$(81 \times 81 + 89) \mod 100 = 50$

Resultado del script:


```
Problems   Output   Debug Console   Terminal   GitLens

Ejercicio 1
Metodo Congruencial Lineal
a = 81, c = 89, m = 100, x0 = 5, n = 5
| n  | X_n |
|----|-----|
| 0  | 5 |
| 1  | 94 |
| 2  | 3 |
| 3  | 32 |
| 4  | 81 |
| 5  | 50 |
```

Figure 1: *
Corresponde al script `congruencial_linear.py`

Método congruencial multiplicativo

Ejercicio 1

Datos:

$$c = 16; \quad x_0 = 13; \quad m = 10^8; \quad a = 211$$

$$X_{n+1} = (211 \cdot x_n \cdot 16) \mod 10^8; \quad x_0 = 13$$

n	X_n
0	13
1	$(211 \times 13 \times 16) \mod 10^8 = 13$
2	$(211 \times 13 \times 16) \mod 10^8 = 438$
3	$(211 \times 43 \times 16) \mod 10^8 = 48$
4	$(211 \times 48 \times 16) \mod 10^8 = 80$
5	$(211 \times 80 \times 16) \mod 10^8 = 35$

Resultado del script:

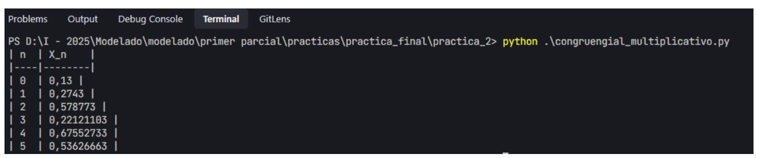


Figure 2: *
Corresponde al script `congruencial_multiplicativo.py`

Algoritmo de cuadrados medios

Datos iniciales:

- Semilla: $X_0 = 9803$
- Constante: $a = 6965$
- Dígitos a considerar: $D = 4$
- Cantidad de números a generar: 5

Cálculo de los números:

Paso 1:

$$Y_0 = 6965 \times 9803 = 68261895$$

Dígitos centrales: **2778**, $X_1 = 2778$, $R_1 = 0.2778$

Paso 2:

$$Y_1 = 6965 \times 2778 = 19345170$$

Dígitos centrales: **3487**, $X_2 = 3487$, $R_2 = 0.3487$

Paso 3:

$$Y_2 = 6965 \times 3487 = 24299155$$

Dígitos centrales: **2869**, $X_3 = 2869$, $R_3 = 0.2869$

Paso 4:

$$Y_3 = 6965 \times 2869 = 19979185$$

Dígitos centrales: **9825**, $X_4 = 9825$, $R_4 = 0.9825$

Paso 5:

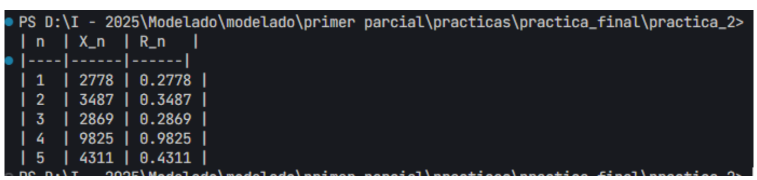
$$Y_4 = 6965 \times 9825 = 68438625$$

Dígitos centrales: **4311**, $X_5 = 4311$, $R_5 = 0.4311$

Resultado Final:

$$R_1 = 0.2778, \quad R_2 = 0.3487, \quad R_3 = 0.2869, \quad R_4 = 0.9825, \quad R_5 = 0.4311$$

Resultado del script:



```
PS D:\I - 2025\Modelado\modelado\primer parcial\practicas\practica_final\practica_2>
| n  | X_n  | R_n   |
|----|------|------|
| 1  | 2778 | 0.2778 |
| 2  | 3487 | 0.3487 |
| 3  | 2869 | 0.2869 |
| 4  | 9825 | 0.9825 |
| 5  | 4311 | 0.4311 |
```

Figure 3: *
Corresponde al script `cuadrados_medios.py`

Algoritmo de Productos Medios

Datos Iniciales:

- Semilla inicial: $X_0 = 5015$
- Segundo valor: $X_1 = 5734$
- Dígitos a considerar: $D = 4$

Cálculo paso a paso:

Paso 1: $Y_0 = 5015 \times 5734 = 28723710$, dígitos centrales: **7560**, $X_2 = 7560$, $R_2 = 0.7560$

Paso 2: $Y_1 = 5734 \times 7560 = 43305040$, dígitos centrales: **3489**, $X_3 = 3489$, $R_3 = 0.3490$

Paso 3: $Y_2 = 7560 \times 3489 = 26334840$, dígitos centrales: **3844**, $X_4 = 3844$, $R_4 = 0.3844$

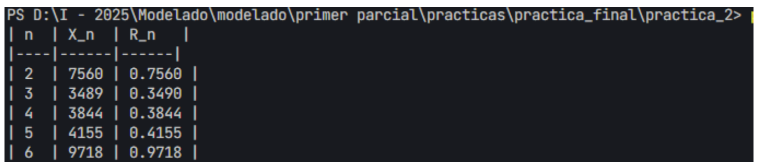
Paso 4: $Y_3 = 3489 \times 3844 = 13402056$, dígitos centrales: **4155**, $X_5 = 4155$, $R_5 = 0.4155$

Paso 5: $Y_4 = 3844 \times 4155 = 15992420$, dígitos centrales: **9718**, $X_6 = 9718$, $R_6 = 0.9718$

Resultado Final:

$$R_2 = 0.7560, \quad R_3 = 0.3490, \quad R_4 = 0.3844, \quad R_5 = 0.4155, \quad R_6 = 0.9718$$

Resultado del script:



```
PS D:\I - 2025\Modelado\modelado\primer parcial\practicas\practica_final\practica_2>
| n  | X_n  | R_n   |
|----|------|------|
| 2  | 7560 | 0.7560 |
| 3  | 3489 | 0.3490 |
| 4  | 3844 | 0.3844 |
| 5  | 4155 | 0.4155 |
| 6  | 9718 | 0.9718 |
```

Figure 4: *
Corresponde al script `productos_medios.py`