

Ejercicio 1 - Simulación de una moneda (promedio de caras)

Objetivo:

Simular el lanzamiento de una moneda con `[random 2]` (0 = cruz, 1 = cara), contar cuántas veces sale “cara” y graficar el **promedio acumulado** en un array.

Hints:

- Usar `[metro]` para simular lanzamientos automáticos.
- `[random 2]` genera 0 o 1.
- Usar `[select 0]` para detectar cruz, y `[moses 1]` para cara.
- Crear un acumulador para contar total de lanzamientos y otro para contar caras.
- Dividir caras / total y guardar en un array para graficar el promedio.

Objetos clave a usar:

`[metro]`, `[random 2]`, `[select]`, `[moses]`, `[float]`, `[+ 1]`, `[expr]`, `[tabwrite]`,
`[acumulador]`

Ejercicio 2 - Simulación de Galton Board (distribución normal)

Objetivo:

Simular la caída de una bolita que pasa por 4 niveles de bifurcaciones aleatorias (izquierda/derecha). El resultado final (posición) se grafica en un array de 16 posiciones, formando una campana.

Hints:

- Una bolita elige entre 0 o 1 cuatro veces → suma total será entre 0 y 4 (en este ejemplo, si se suman 4 decisiones binarias).
- Convertir suma en posición final.
- Contar cuántas bolitas caen en cada posición.

Ejercicio 3 - Simulación Monte Carlo para estimar π

Objetivo:

Simular puntos al azar dentro de un cuadrado unitario (x e y entre 0 y 1). Calcular cuántos caen dentro de un círculo inscrito ($x^2 + y^2 < 1$). Estimar π como:

$$\pi \approx 4 \times \text{puntos dentro del círculo} / \text{total de puntos}.$$

Graficar el valor de π a medida que van saliendo los puntos, deberá converger....cuál es el error en función de la cantidad de puntos (error porcentual entre el valor estimado y el real 3.14159...)

Hints:

- Usar `[random]` para generar x e y entre 0 y 1.
- Calcular $x^2 + y^2 < 1$ con `[expr]`.
- Usar `[moses 1]` para verificar si está dentro del círculo.
- Usar 2 acumuladores: uno para puntos totales, otro para los que caen dentro.
- Multiplicar el cociente por 4.

EJERCICIO 4 (Versión Monte Carlo) - Cálculo del área de una figura geométrica (ej. Triángulo definido por 3 puntos)

Objetivo:

Simular muchos puntos al azar dentro del rectángulo que contiene al triángulo definido por tres puntos dados, y determinar si esos puntos caen dentro del triángulo.

Hints - ¿Cómo saber si un punto está dentro de un triángulo?

Método 1 - Usar sub-áreas (clásico):

Dado un punto P y triángulo ABC si el área de $PAB + PBC + PCA = ABC$, el punto está dentro.
Esto requiere calcular varias áreas.

$$A = 0.5 * \text{abs}(x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))$$

INTERACCIÓN con el TECLADO- Aumentar/Disminuir valor con teclas (A y D)

Objetivo:

Usar las teclas **A** (izquierda) y **D** (derecha) para aumentar o disminuir el valor de una variable visualizada en un `[number]`.

Hints:

- Tecla **A** = 97, **D** = 100 (códigos ASCII).
- Usar `[select 97 100]`.
- Para **A**, restar 1; para **D**, sumar 1.
- Guardar valor con `[float]`.

Objetos clave:

`[key]`, `[select]`, `[+ 1]`, `[- 1]`, `[float]`, `[number]`