



Título: EL MÉTODO MONTE CARLO

Nivel educativo: 3º-4º ESO.

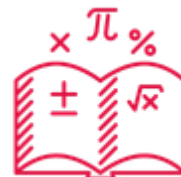
Materia/s: Matemáticas-Tecnología.

Temporalización: Tercer trimestre.

Descripción breve de la actividad

El método de Monte Carlo es una forma probabilística de aproximar el número π (pi) usando números aleatorios. El método consiste en inscribir un círculo en un cuadrado de manera que el lado del cuadrado sea igual al radio del círculo. Después lanzaremos puntos al azar. La proporción de puntos que caen dentro del círculo está relacionada con π .

Utilizaremos la placa Micro:bit conectada al navegador para lanzar los puntos y comprobar cómo se va acercando al resultado esperado según aumentamos el número de tiradas.



Objetivos

1. Comprender el concepto y la utilidad del método de Monte Carlo como técnica de estimación basada en simulaciones y generación de números aleatorios
2. Comprender la relación entre matemáticas y tecnología, viendo cómo conceptos abstractos pueden resolverse mediante programación y experimentación digital.

3. Promover el trabajo cooperativo, compartiendo código, discutiendo resultados y mejorando la simulación en equipo.
4. Comprender la relación entre matemáticas y tecnología, viendo cómo conceptos abstractos pueden resolverse mediante programación y experimentación digital.

Competencias clave a desarrollar: Competencia digital, competencia matemática, competencia en comunicación lingüística, competencia personal, social y de aprender a aprender, competencia ciudadana, competencia emprendedora.



Pasos a seguir

1. **Introducción: el método de Monte Carlo.** Explicación paso a paso de cómo la relación entre el número de puntos que caen dentro del círculo y el número de puntos lanzados es igual a $\pi/4$.
2. **Presentación del Proyecto:** se explica el proyecto que se va a realizar.
3. **Antes de empezar a programar:**
 - a. Para crear puntos aleatorios necesitamos dos variables: una para x y otra para y.
 - b. Los puntos que caen dentro del círculo son los que cumplen la siguiente condición: $x^2 + y^2 \leq R^2$.
 - c. No se pueden generar números aleatorios decimales, es mejor que los números estén entre 0 y 200, por ejemplo.
 - d. Explicar la comunicación puerto serie con micro:bit.
4. **Programación:** Se realiza un programa sencillo utilizando las nociones de programación y matemáticas aprendidas en los pasos anteriores.
5. **Prueba y mejora del programa.**

Sugerencias

Aunque hablemos de números aleatorios en realidad son pseudo aleatorios ya que son creados usando un algoritmo.

El navegador Firefox no permite la transmisión de datos con Micro:bit..

Es conveniente empezar el programa creando las variables necesarias: x, y, tiradas, dentro del círculo y pi.

Se puede ir cambiando el número de veces que se repite el bucle para acercarse al valor teórico más rápidamente.

Mostrar el número en la pantalla de la micro:bit suele ralentizar el programa.



Recursos

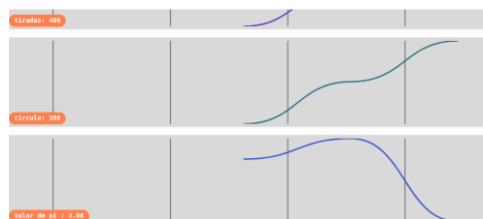
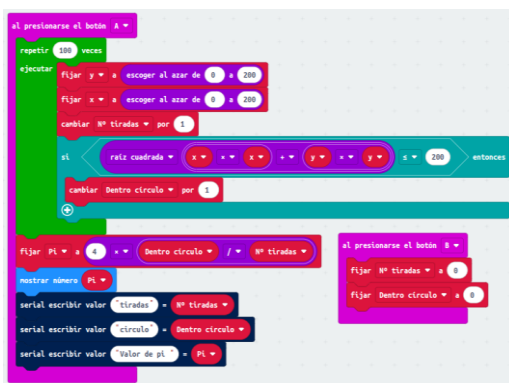
- **Personales:** Docentes y alumnado.
- **Materiales:** Micro:bit y ordenador con conexión a internet.



Espacios: Aula de informática/Aula de clase.

Tipo de actividad: Colaborativa.

[Programa](#)
[MakeCode](#)





Evaluación



Criterios de Evaluación	Criterios de Evaluación			
	4 Excelente	3 Satisfactorio	2 Mejorable	1 Insuficiente
1. Comprende el concepto y la utilidad del método de Monte Carlo	Explica el método con claridad y precisión, incluyendo la idea de simulación, aleatoriedad y su uso para estimar π .	Explica el método de forma correcta pero sin profundidad o con algún detalle incompleto.	Muestra comprensión parcial: menciona la simulación pero confunde aspectos clave o no relaciona bien con la estimación.	No logra explicar el método o muestra ideas incorrectas sobre su funcionamiento .
2. Justifica por qué el método permite aproximar π	Relaciona correctamente proporciones, áreas y recuento de puntos, mostrando comprensión del fundamento matemático.	Justifica parcialmente la relación entre puntos y π , aunque con simplificaciones .	Reconoce que se obtiene una aproximación pero no explica el razonamiento matemático.	No justifica la relación entre el método y el valor de π .



3. Interpreta los resultados obtenidos y analiza el error	Analiza los datos con precisión, compara con el valor real de π y explica el origen del error.	Interpreta los resultados de forma correcta pero sin análisis profundo del error.	Ofrece una interpretación superficial o incompleta; muestra dificultades para comparar con π .	No interpreta los resultados o presenta conclusiones incorrectas.
4. Implementa una solución tecnológica a un problema matemático	Programa la micro:bit con un código funcional, claro y bien estructurado que reproduce correctamente el método.	El código funciona con pequeños fallos o falta de optimización, pero cumple el objetivo.	El código presenta errores que afectan significativamente la simulación.	No consigue implementar el código o este no se relaciona con el método.
5. Participación activa en el trabajo cooperativo	Participa continuamente, aporta ideas relevantes y mejora activamente el trabajo del equipo.	Participa de manera constante pero con aportaciones más limitadas.	Participación irregular o poco significativa.	No participa o dificulta el trabajo del equipo.
6. Comparte su código y colabora en la depuración	Comparte su trabajo abiertamente, revisa y mejora el código de sus compañeros con actitud constructiva.	Comparte y revisa código ocasionalmente.	Comparte el código solo cuando se le solicita; contribuye poco a la depuración.	No comparte código o evita colaborar en mejoras.



Pensamiento computacional

A elegir las que consideréis oportuno:

Lógica (predicción y análisis): utilizar el razonamiento para hacer predicciones, resolver problemas y tomar decisiones basadas en la información disponible.

Algoritmos (pasos y reglas): seguir una serie de pasos o instrucciones bien definidas para resolver un problema o completar una tarea.

Descomposición (dividir en partes): dividir un problema grande en partes más pequeñas y manejables, que son más fáciles de entender y resolver.

Patrones (detectar y usar similitudes): identificar similitudes o patrones en problemas o datos, lo que facilita encontrar soluciones más rápidas y eficientes.

Abstracción (eliminar detalles innecesarios): Simplificar un problema eliminando detalles que no son importantes, para enfocarse en lo que es relevante y esencial.

Autoría

Esta actividad ha sido realizada por el **autor**, en el marco del **Programa Código Escuela 4.0 _Madrid**:

Jaime Temes Mendoza