

Mini EP 1

Alfredo Goldman e Vitor Terra

O objetivo deste mini EP é verificar a influência da escolha da linguagem de programação no tempo de execução de um determinado algoritmo. Neste caso, vocês devem implementar em duas linguagens diferentes à sua escolha o cálculo de pi por meio do método de Monte Carlo.

Neste método, gera-se um número muito grande  $N$  de pontos aleatórios  $(x, y)$  no quadrado  $[0.0, 1.0] \times [0.0, 1.0]$ . Para este Mini EP, adotaremos  $N = 10,000,000$ . Verifica-se então quantos desses pontos estão contidos no círculo unitário, ou seja, para quantos pontos  $(x, y)$  tem-se que  $x^2 + y^2 \leq 1$ . Como  $x$  e  $y$  estão no primeiro quadrante, tais pontos estão contidos no quarto de círculo de raio 1, cuja área é  $\pi/4$ .

Para um número muito grande de pontos, a razão entre o número de pontos dentro da figura e o número de pontos dentro do quadrado é uma estimativa da razão entre a área da figura e a área do quadrado. Como a figura (quarto de círculo) tem área  $\pi/4$  e o quadrado tem área 1, o valor de pi pode ser aproximado por:

$$4 * (\text{número de pontos tais que } x^2 + y^2 \leq 1) / N$$

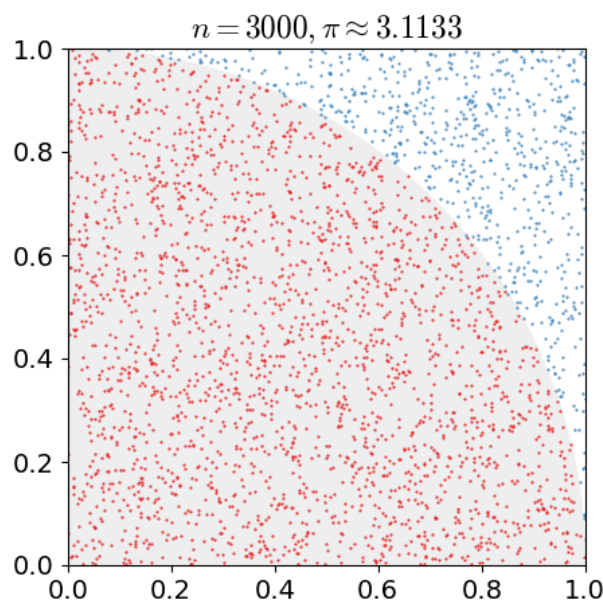


Figura 1: Ilustração do método de Monte Carlo para cálculo de pi ([versão original animada aqui](#))

O pseudo-código para o algoritmo descrito anteriormente é:

```
N <- 10^7  
acertos <- 0  
repita N vezes:  
    x <- rand()  (float entre 0 e 1)  
    y <- rand()  
    if x * x + y * y <= 1.0:  
        acertos <- acertos + 1  
print(4 * acertos / N)
```

Implemente o pseudo-código acima em duas linguagens de programação à sua escolha. Meça o tempo de execução de cada uma das implementações usando uma ferramenta externa ao código (por exemplo, no Linux, use a ferramenta `time`). Em programas com predominância de cálculos matemáticos em sua execução, tais como o deste Mini EP, espera-se que as variações no tempo transcorrido sejam pequenas para diferentes execuções. Logo, para este exercício, basta informar o tempo médio de algumas execuções (pelo menos 10).

Envie os resultados obtidos [neste formulário](#) para que possamos comparar com os demais da turma. Envie também pelo formulário o valor do `BogoMIPS` do processador nos quais foi realizada a execução. Trata-se, em resumo, de uma medida relacionada ao desempenho do processador.

No Linux, o `BogoMIPS` pode ser encontrado no arquivo `/proc/cpuinfo` e obtido pelo seguinte comando: `grep -i bogomips /proc/cpuinfo`. No Windows e demais SOs, procure na Internet o valor do `BogoMIPS` esperado para o modelo do seu processador.