

Atividade Prática

Programação Paralela e Distribuída

Prof. Dr. Lucas Venezian Pova

Objetivo da atividade

Criar um **simulador simplificado de atendimento em caixas de supermercado** que permita estimar, de forma experimental, o número de caixas necessários para reduzir o tempo médio de atendimento dos clientes. Nesta etapa inicial, trabalharemos apenas com a variabilidade dos *tempos de serviço*, sem ainda modelar explicitamente os intervalos de chegada.

Nota Didática

Conceito de estocástico. Um processo estocástico é aquele governado por variáveis aleatórias, isto é, cujo comportamento não é totalmente previsível, mas segue distribuições de probabilidade. No nosso simulador, o tempo de atendimento de cada cliente é *estocástico*: cada execução produz resultados diferentes, ainda que dentro de uma tendência média. Para aprofundamento: Ross, Sheldon M. *Introduction to Probability Models*. Academic Press.

Formulação matemática

Cada tempo de atendimento S_i é gerado a partir de:

$$S_i = \max(\varepsilon, \mu + \sigma \cdot Z_i), \quad Z_i \sim \mathcal{N}(0, 1),$$

onde:

- μ : média do tempo de atendimento (min);
- σ : desvio-padrão do tempo de atendimento (min);
- Z_i : amostra de uma distribuição normal padrão (média 0, variância 1);
- $\varepsilon > 0$: limite inferior para evitar tempos negativos.

A média dos N atendimentos simulados é:

$$\bar{S} = \frac{1}{N} \sum_{i=1}^N S_i.$$

Código-fonte (Java)

SimulacaoCaixaSupermercado

Objetivo da classe. Gerar amostras de tempos de atendimento a partir de uma distribuição normal e calcular a média desses tempos. A classe concentra os parâmetros da simulação e retorna o resultado de cada rodada.

```
1 public class SimulacaoCaixaSupermercado {
2
3     private int numeroCaixas;
4     private int mediaAtendimentos;
5     private double mediaTempoAtendimentoPorCliente;
6     private double desvioPadraoTempoAtendimentoPorCliente;
7
8     private static final double TEMPO_MINIMO_ATENDIMENTO = 0.1;
9
10    private final java.util.Random rng = new java.util.Random(42);
11
12    public void setNumeroCaixas(int n) { this.numeroCaixas = n; }
13    public void setMediaAtendimentos(int n) { this.mediaAtendimentos = n; }
14    public void setMediaTempoAtendimentoPorCliente(double mu) {
15        this.mediaTempoAtendimentoPorCliente = mu;
16    }
17    public void setDesvioPadraoTempoAtendimentoPorCliente(double sigma) {
18        this.desvioPadraoTempoAtendimentoPorCliente = sigma;
19    }
20
21    private double tempoAtendimentoNormalTruncado() {
22        double z = rng.nextGaussian();
23        double s = mediaTempoAtendimentoPorCliente +
24            desvioPadraoTempoAtendimentoPorCliente * z;
25        return (s < TEMPO_MINIMO_ATENDIMENTO) ? TEMPO_MINIMO_ATENDIMENTO : s;
26    }
27
28    public double simular() {
29        double soma = 0.0;
30        for (int i = 0; i < mediaAtendimentos; i++) {
31            soma += tempoAtendimentoNormalTruncado();
32        }
33        return soma / mediaAtendimentos;
34    }
35 }
```

Main

Objetivo da classe. Configurar parâmetros, executar múltiplas rodadas da simulação e calcular estatísticas resumidas (média das médias e desvio-padrão).

```
1 import java.util.ArrayList;
```

```

2 import java.util.List;
3
4 public class Main {
5
6     public static void main(String... args) {
7
8         final int NUMERO_SIMULACOES = 1000;
9         final List<Double> mediasAtendimento = new ArrayList<>();
10
11         SimulacaoCaixaSupermercado simulador = new SimulacaoCaixaSupermercado();
12
13         for (int i = 0; i < NUMERO_SIMULACOES; i++) {
14             simulador.setNumeroCaixas(1);
15             simulador.setMediaAtendimentos(100);
16             simulador.setMediaTempoAtendimentoPorCliente(5.00);
17             simulador.setDesvioPadraoTempoAtendimentoPorCliente(0.50);
18
19             double mediaAtendimento = simulador.simular();
20             mediasAtendimento.add(mediaAtendimento);
21         }
22
23         double media = media(mediasAtendimento);
24         double dp = desvioPadrao(mediasAtendimento, media);
25
26         System.out.printf("Mdia dos tempos de atendimento (%.0f simulaes): %.3f
27 min%n",
28                             (double) NUMERO_SIMULACOES, media);
29         System.out.printf("Desvio-padro das mdias: %.3f min%n", dp);
30     }
31
32     private static double media(List<Double> xs) {
33         double s = 0.0;
34         for (double x : xs) s += x;
35         return s / xs.size();
36     }
37
38     private static double desvioPadrao(List<Double> xs, double m) {
39         double s2 = 0.0;
40         for (double x : xs) {
41             double d = x - m;
42             s2 += d * d;
43         }
44         return Math.sqrt(s2 / (xs.size() - 1));
45     }
46 }

```

Atividades propostas

1. Execute a simulação para $\mu = 5,0$, $\sigma = 0,5$, $N = 100$ clientes e 1000 rodadas. Registre a média e o desvio-padrão obtidos.
2. Varie o número de caixas de 1 para 2 e 3. Compare os resultados obtidos e discuta qualitativamente como mais caixas podem reduzir o tempo médio de atendimento.
3. Varie σ (ex.: 0,25, 1,0, 2,0) e observe como a variabilidade impacta os resultados médios.
4. Escreva um parágrafo explicando por que este simulador é considerado *estocástico* e como isso representa situações reais.