

SIMULADOR DE COLETA DE LIXO: GESTÃO E SUSTENTABILIDADE

Instituto de Ensino Superior ICEV

Teresina - Piauí

Bruno Moraes de Sousa Ribeiro, Sávio José Sousa Macêdo

savio.macedo@somosicev.com | bruno.ribeiro@somosicev.com **Abstract**

Abstract: *This paper presents the development of a waste collection simulator applied to the urban context of Teresina, Brazil. The project was motivated by the need to better understand, in a practical way, the logistical challenges involved in solid waste management. Built in Java using custom data structures, the simulator models the full operation of urban waste collection, including waste generation, truck routing, unloading at transfer stations, and transportation to the landfill. The tool enables performance analysis under different scenarios by varying the number of trucks and their assigned routes. The results highlight how operational adjustments—such as increasing the number of trips or redistributing service zones—directly impact collection efficiency, helping to identify bottlenecks and support decision-making. In addition to its academic relevance, the simulator serves as a didactic tool for studying data structures and event-driven system modeling.*

Resumo: *Este trabalho descreve o desenvolvimento de um simulador de coleta de lixo aplicado ao contexto urbano da cidade de Teresina – PI. A proposta surgiu a partir da necessidade de compreender, de forma prática, os desafios logísticos envolvidos na gestão de resíduos sólidos. O simulador foi construído em Java, com o uso de estruturas de dados próprias, e permite visualizar o funcionamento de uma operação completa de coleta, incluindo geração de lixo, roteirização de caminhões, descarregamento em estações de transferência e transporte até o aterro. A ferramenta possibilita a análise do desempenho em diferentes cenários, considerando variações no número de caminhões e nas rotas definidas. Os resultados obtidos mostram como ajustes na operação – como o aumento de viagens ou redistribuição das zonas atendidas – impactam diretamente a eficiência da coleta, revelando pontos críticos e auxiliando na tomada de decisões. Além de sua utilidade acadêmica, o simulador se mostra aplicável como apoio didático para o estudo de estruturas de dados e modelagem de sistemas baseados em eventos.*

1. Introdução e Visão Geral do Projeto

O projeto foi desenvolvido no contexto da disciplina de Estruturas de Dados, com foco em Gestão de Resíduos Sólidos e Sustentabilidade Ambiental. O sistema realiza uma simulação

orientada a eventos discretos, modelando o processo de coleta de lixo na cidade de **Teresina, Piauí**.

O principal objetivo da aplicação é gerenciar a logística de coleta, levando em consideração fatores como tempos de trajeto, períodos de pico de tráfego e capacidades limitadas dos veículos. A simulação é executada em ciclos diários, englobando as etapas de geração de lixo, coleta, transferência e descarregamento.

2. Arquitetura do Sistema

O sistema possui uma arquitetura modular, organizada em múltiplos pacotes Java, descritos a seguir:

- `simulador.caminhoes`: Inclui as classes `CaminhaoPequeno` e `CaminhaoGrande`.
- `simulador.estacoes`: Contém a classe `EstacaoDeTransferencia`.
- `simulador.eventos`: Abriga as classes de eventos, como `EventoColeta`, `EventoChegadaEstacao`, entre outros, além da `AgendaEventos`.
- `simulador.zona`: Composto pelas classes `Zona`, `Zonas` e `GerenciadorZonas`.
- `simulador.configTempo`: Responsável pelo gerenciamento do tempo, com as classes `GerenciadorTempo` e `TempoDetalhado`.
- `simulador.configuracao`: Define os parâmetros globais da simulação.
- `estruturas`: Contém as estruturas de dados personalizadas como `Lista` e `Fila`.
- `simulador`: Agrega as classes centrais como `Main`, `Simulador`, `SimuladorGUI` e `EstatisticasDia`.

A classe principal, `Simulador`, é responsável por inicializar os elementos do sistema e processar os eventos conforme a `AgendaEventos`.

3. Estruturas de Dados Personalizadas

foram implementadas estruturas de dados próprias, sem o uso do Java Collections Framework:

Lista

- Lista duplamente encadeada genérica.
- Utilizada para armazenar zonas, caminhões pequenos e eventos.
- Suporta inserções ordenadas, remoções e acesso por índice.
- Apresenta complexidade **O(n)** para remoção de elementos específicos.

Fila

- Fila genérica do tipo FIFO.
- Utilizada na gestão dos caminhões pequenos nas estações.
- Operações básicas com complexidade **O(1)**.
- Não permite acesso aleatório ou ordenação.

4. Algoritmos Implementados e Fluxo Operacional

A simulação segue o seguinte fluxo operacional baseado em eventos:

1. **Inicialização:** Criação de zonas e estações.
2. **Geração de Lixo:** Quantidade aleatória por zona.
3. **Distribuição de Caminhões:** Com capacidade e número fixo de viagens diárias.
4. **Gestão de Eventos:** Eventos processados em ordem cronológica.
5. **Coleta de Lixo:** Caminhão coleta até sua capacidade ou esvaziar a zona.
6. **Cálculo de Tempos:** Considera horários de pico e congestionamentos.
7. **Transferência para Estação:** Caminhão pequeno desloca-se para descarregar.
8. **Gestão de Filas e Descarregamento:** Controle da fila e acionamento de caminhões grandes.
9. **Resumo Diário:** Exibição de dados e reinício da agenda para o próximo dia.

5. Classes e Componentes Chave

- **CaminhaoPequeno.java:** Executa a coleta nas zonas.
- **CaminhaoGrande.java:** Transporta resíduos das estações ao aterro.
- **Zona.java:** Responsável por gerar e armazenar lixo.
- **EstacaoDeTransferencia.java:** Recebe caminhões pequenos e aciona caminhões grandes.
- **GerenciadorTempo.java:** Calcula e ajusta tempos.
- **configuracao.java:** Centraliza os parâmetros da simulação.
- **Classes de Evento:** Controlam a lógica temporal da simulação.
- **SimuladorGUI.java:** Interface gráfica desenvolvida em JavaFX.

6. Resultados Esperados vs. Obtidos

Resultados Esperados:

Esperava-se que o sistema simulasse uma operação realista de coleta de resíduos, oferecendo subsídios para decisões logísticas e acadêmicas.

Resultados Obtidos:

- **Geração de Resíduos** distribuída entre as zonas.
- **Relatórios por Caminhão**, com dados detalhados de tempo, carga, zona e viagens.
- **Informações das Estações**, com horários, carga e próximos passos.
- **Despacho de Caminhões Grandes**, com controle automático do volume.

Desafios:

- **Ordem Cronológica dos Eventos:** Demandou alto controle lógico.
- **Cálculo de Tempos Realistas:** Ocorreram erros e complexidades.
- **Gestão de Fila:** Exigiu atenção especial para evitar bloqueios e sobrecargas.

7. Tecnologias Utilizadas

- **Java (JDK 24)**
- **JavaFX** (interface gráfica)
- **IntelliJ IDEA** (IDE)
- **Git** (controle de versão)
- **Programação Orientada a Objetos**
- **Estruturas de Dados Personalizadas**
- **Simulação Baseada em Eventos**

8. Gráfico de Resultados Simulados

O gráfico abaixo demonstra claramente como o aumento do número de caminhões e, principalmente, do número de viagens por caminhão (Cenário 3), impacta positivamente a eficiência da coleta de lixo, resultando em menos lixo remanescente nas zonas, especialmente naquelas que inicialmente apresentavam gargalos, como a Zona Sul e Norte. Um exemplo disso pode ser observado no Cenário 1, onde a Zona Norte ficou com lixo remanescente de 25 toneladas por não ter caminhões suficientes.

Lixo Gerado vs. Coletado por Zona e Cenário

