

Relatório Final do Projeto: Simulador de Coleta de Lixo para Teresina

Curso: Engenharia de Software **Disciplina:** Estrutura de Dados **Alunos:** Felipe Duan da Silva Sousa, Adryan Ryan Silva Guimarães **Professor:** Ricardo Sekeff **Data de Entrega:** 22 de maio de 2025

1. Introdução

O presente trabalho tem como objetivo o desenvolvimento de um simulador computacional que representa de forma realista o processo de coleta de lixo na cidade de Teresina, Piauí. Esta iniciativa está alinhada com os princípios de sustentabilidade ambiental e gestão eficiente de resíduos sólidos. Utilizando o paradigma da programação orientada a eventos e estruturas de dados implementadas do zero, o sistema permite a análise detalhada de operações logísticas, alocação de recursos e desempenho das zonas de coleta e estações de transferência.

O projeto surgiu da necessidade de aplicar conceitos teóricos de Estrutura de Dados em um contexto prático, passado pelo professor. A coleta de lixo é uma atividade essencial em centros urbanos, e a sua simulação permite avaliar cenários, planejar estratégias e compreender gargalos operacionais de forma segura e didática.

2. Tema do Projeto

Simulação orientada a eventos da coleta de lixo urbano na cidade de Teresina, com utilização de estruturas de dados desenvolvidas manualmente e visualização interativa através de JavaFX. O projeto combina conceitos de computação aplicada, sustentabilidade e eficiência logística.

3. Objetivos

Geral: Desenvolver um simulador para representar a dinâmica da coleta de lixo em Teresina, com gestão eficiente das rotas, dos recursos logísticos e dos eventos da simulação.

Específicos:

- Criar estruturas de dados (listas, filas, pilhas) do zero, sem uso de coleções prontas do Java.
 - Modelar os elementos do sistema, incluindo caminhões pequenos, caminhões grandes, zonas de coleta e estações de transferência.
 - Gerenciar eventos como geração de lixo, coleta, descarga, transferência e encerramento de turno.
 - Exibir logs e estatísticas em tempo real, oferecendo feedback sobre o funcionamento interno do sistema.
 - Permitir a configuração de parâmetros como capacidades dos caminhões, tempos de viagem e tolerâncias operacionais.
 - Visualizar a simulação com animações através de uma interface gráfica responsiva desenvolvida com JavaFX.
-

4. Requisitos Atendidos

Funcionais:

- Divisão da cidade em 5 zonas (Sul, Norte, Centro, Leste e Sudeste), cada uma com sua própria taxa de geração de lixo.
- Geração configurável de lixo por zona, com valores mínimo e máximo definidos em parâmetros.
- Caminhões pequenos (2, 4, 8 e 10 toneladas) com limite de viagens diárias e tempos de trajeto aleatórios que variam conforme horários de pico e fora de pico.
- Gerenciamento de filas em estações de transferência para descarregar o lixo em caminhões grandes de 20 toneladas.
- Caminhões grandes respeitam uma tolerância de espera, sendo redirecionados ao aterro sanitário ao atingir o tempo-limite.
- Inclusão dinâmica de novos caminhões grandes quando a demanda supera a capacidade atual.

Não funcionais:

- Sistema modular, com separação clara entre entidades, eventos e lógica da interface.
- Logs informativos, com detalhamento das operações realizadas.
- Visualização gráfica amigável e interativa, com animações sincronizadas com o tempo da simulação.
- Foco em eficiência e clareza na implementação das estruturas de dados.

5. Metodologia e Implementação

O projeto foi desenvolvido utilizando a linguagem Java e organizado em pacotes específicos:

`simulador.eventos`, `simulador.zona`, `simulador.util`, `estruturas.lista`, `estruturas.filas`, `simulador.caminhoes`, `simulador.estacoes`, entre outros. A estrutura modular do projeto permite manutenção e expansão facilitadas.

As estruturas de dados utilizadas foram implementadas do zero, respeitando os princípios da disciplina. As TAD's, como `Lista`, `Fila`, `ListaDuplamenteEncadeada`, e a `FilaEncadeada`, foram fundamentais para o gerenciamento das filas de caminhões nas estações e da agenda de eventos.

A classe `Simulador` é responsável por inicializar os parâmetros da simulação, distribuir eventos na agenda, processar cada etapa da coleta e atualizar os registros em tempo real. Eventos como `EventoGeracaoLixoZona`, `EventoColeta`, `EventoTransferenciaParaEstacao` e `EventoEstacaoTransferencia` foram criados para modelar todo o ciclo logístico.

O JavaFX foi utilizado para compor uma interface interativa, com painéis que representam visualmente cada zona da cidade, os caminhões e as estações. O usuário pode controlar a velocidade da simulação, visualizar os logs em tempo real e acompanhar as animações de deslocamento e coleta.

6. Resultados Obtidos

- Simulação precisa com coleta distribuída entre as zonas e comportamento realista.
- Caminhões pequenos e grandes gerenciados dinamicamente, respeitando tempo, limite de carga e zonas prioritárias.
- Sistema de redirecionamento implementado com base em zonas com maior acúmulo de lixo.
- Geração de logs detalhados com todas as operações e estados da simulação.
- Integração com JavaFX permitindo visualização intuitiva, com cronômetro, controle de velocidade e animação de caminhões.

7. Conclusão

Este projeto proporcionou uma aplicação prática dos conceitos de Estrutura de Dados, especialmente no uso de listas e filas encadeadas, além de reforçar boas práticas de programação orientada a objetos, modularização e separação de responsabilidades.

A simulação criada permite visualizar de forma didática o impacto da logística urbana na coleta de lixo, ajudando a compreender os desafios da gestão de resíduos em cenários reais. A implementação em JavaFX agregou uma camada visual essencial para a compreensão dos eventos, e os desafios enfrentados nos permitiram amadurecer nossas habilidades de projeto, análise de sistemas e implementação gráfica.

Trata-se de uma solução funcional, extensível e alinhada à proposta de sustentabilidade, com possibilidade de evoluir para aplicações reais no futuro, inclusive em contextos de planejamento urbano e educação ambiental.

Anexos

- Capturas de tela da interface JavaFX em execução.
- Exemplo de logs gerados durante uma simulação completa.
- Diagrama de pacotes e estrutura de classes.
- Estrutura de diretórios do projeto.
- Documentação JavaDoc com descrição das classes e métodos principais.