

ICEV

Eduardo Pedro Santos Libório
Iaggo Vinícius de Oliveira Silva
Leo Augusto Moraes Freitas

**Simulador de Coleta de Lixo para Teresina Gestão de Resíduos
Sólidos e Sustentabilidade Ambiental.**

Relatório Técnico

Teresina, Piauí
2025

Relatório Técnico

1. Pré-projeto

A implementação do sistema de simulação de gestão de coleta de lixo foi precedida pela construção de um diagrama de classes UML que serve como base conceitual inicial para o desenvolvimento das classes e suas interações. O diagrama descreve as principais entidades envolvidas no processo de coleta e transporte de resíduos, suas responsabilidades e relações de dependência.

Componentes Principais do Diagrama:

Simulador: Classe responsável por iniciar, pausar e gerar relatórios da simulação. Controla o fluxo principal da aplicação.

Zona: Representa uma região da cidade onde o lixo é gerado. Possui atributos como `nomeZona` e `taxaGeracaoLixo`, além de um método `gerarLixo()` que simula a produção de resíduos.

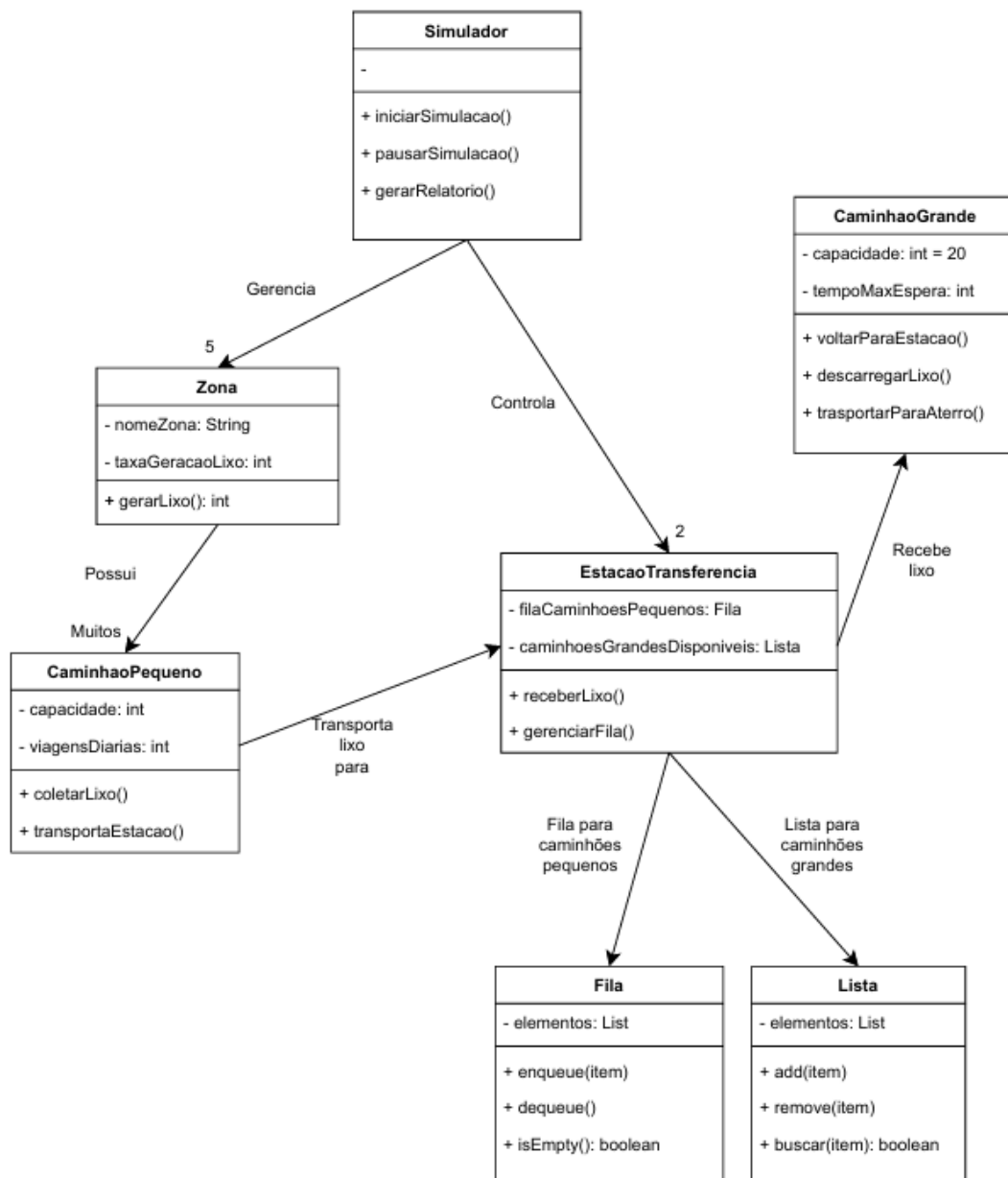
CaminhaoPequeno: Responsável por coletar o lixo nas zonas e transportá-lo até a estação de transferência. Armazena informações sobre capacidade e limite de viagens diárias.

EstacaoTransferencia: Local intermediário entre as zonas e o destino final do lixo. Gerencia uma fila de caminhões pequenos (Fila) e uma lista de caminhões grandes (Lista). Possui métodos para receber lixo e gerenciar o fluxo de caminhões.

CaminhaoGrande: Transporte de longa distância que leva o lixo da estação para o aterro sanitário. Define capacidade fixa, tempo máximo de espera e métodos para descarregar e transportar o lixo.

Fila: Estrutura de dados baseada em FIFO (First-In, First-Out), usada para armazenar a sequência de chegada dos caminhões pequenos à estação.

Lista: Estrutura usada para manter e gerenciar os caminhões grandes disponíveis na estação.



Para garantir uma melhor organização e modularização do código, as classes do sistema foram distribuídas em três pacotes distintos. O pacote estruturasDeDados agrupa as classes responsáveis pelas estruturas auxiliares, como Fila e Lista, que oferecem suporte à lógica de controle da simulação. O pacote cidade concentra os elementos do mundo simulado, incluindo entidades como CaminhaoPequeno, CaminhaoGrande, Zona e EstacaoTransferencia, representando os componentes físicos e lógicos da cidade. Já o pacote

application contém as classes de controle e execução da simulação, como a classe principal Simulador, além de módulos auxiliares para geração de estatísticas e relatórios. Essa separação em pacotes facilita a manutenção, reutilização e expansão futura do sistema.

2. Estruturas de dados

2.2 Fila

A classe Fila aparece como uma estrutura de dados auxiliar usada pela EstacaoTransferencia para controlar a ordem de chegada dos caminhões pequenos (CaminhaoPequeno). Ela é representada com um atributo elementos: List e os métodos básicos de manipulação de filas: enqueue(item), dequeue() e isEmpty().

No entanto, na implementação prática em Java, a estrutura Fila foi elaborada com um modelo de lista encadeada personalizada, utilizando nós internos ao invés de uma List genérica. Isso garante maior controle sobre os elementos, menor sobrecarga e facilita a customização para as necessidades da simulação.

A classe Fila é usada pela EstacaoTransferencia para armazenar os caminhões pequenos que chegam cheios de lixo. Ela garante que os caminhões sejam processados na ordem correta (FIFO), simulando uma fila real de espera, permitindo que o gerenciamento da estação decida quando um caminhão grande pode ser acionado para transporte, com base na quantidade de caminhões pequenos enfileirados.

2.2 Fila

A classe Lista aparece como uma estrutura auxiliar utilizada pela EstacaoTransferencia para armazenar os caminhões grandes (CaminhaoGrande) disponíveis para transporte de lixo ao aterro. Ela é representada como uma lista encadeada simples personalizada, com operações específicas de adição e remoção em posições arbitrárias. Essa abordagem oferece flexibilidade e controle sobre a manipulação dos caminhões grandes.

A Lista representa os caminhões grandes em espera na EstacaoTransferencia, prontos para transportar grandes volumes de lixo acumulado por vários caminhões pequenos. Seu uso garante que os caminhões grandes possam ser adicionados, removidos ou inspecionados conforme a lógica de controle da simulação.

2.3 Lista de Strings

A classe MinhaListaDeStrings foi implementada como uma estrutura auxiliar para armazenar logs textuais e informações descritivas geradas durante a execução da simulação. Seu principal objetivo é registrar dados importantes para análises futuras, como comparações entre diferentes execuções, rastreamento de eventos, coleta de métricas ou geração de relatórios históricos.

2. Cidade

3.1 Caminhão Grande

A classe `CaminhaoGrande` representa um veículo de grande porte utilizado para transportar grandes volumes de lixo da estação de transferência até o aterro sanitário. Ela possui uma capacidade máxima de 20.000 kg e mantém o controle da carga atual transportada.

O método `getCargaAtual()` permite acessar a quantidade de lixo atualmente armazenada no caminhão, sendo útil para monitoramento e controle do carregamento. Já o método `carregar(int quantidade)` é responsável por adicionar resíduos à carga do caminhão. Caso a quantidade informada ultrapasse o limite máximo permitido, o caminhão é carregado até sua capacidade total, evitando sobrecarga e mantendo a integridade da simulação.

Para determinar se o caminhão já pode ser enviado ao aterro, utiliza-se o método `prontoParaPartir()`, que retorna verdadeiro quando a carga atinge ou supera os 20.000 kg. Por fim, o método `descarregar(int quantidade)` simula o processo de despejo do lixo no aterro. Ele reduz a carga do caminhão em até quantidade quilogramas e imprime no console a quantidade descarregada, junto com a carga restante. Esse comportamento permite acompanhar a operação e registrar o progresso da simulação.

Em conjunto, esses métodos oferecem uma modelagem funcional completa do ciclo de transporte do lixo por caminhões grandes, desde o carregamento na estação até o descarregamento no destino final. `CaminhaoGrandePadrao` representando um tipo padrão de caminhão grande utilizado na simulação. Ela herda toda a estrutura e comportamento da classe pai. Sua única implementação adicional é o construtor, que inicializa a carga atual do caminhão com o valor zero. Essa ação garante que, ao ser instanciado, um caminhão grande padrão sempre começará vazio e abre espaço para futuras extensões com comportamentos personalizados, caso diferentes tipos de caminhões grandes venham a ser implementados.

3.2 Caminhão Pequeno

A classe `CaminhaoPequeno` é uma classe abstrata que representa o modelo base de todos os caminhões de pequeno porte utilizados na simulação de coleta de lixo. Ela é projetada para ser estendida por subclasses específicas, que definirão detalhes como capacidade e comportamento adicional, mas já oferece um conjunto robusto de funcionalidades essenciais para o funcionamento do sistema.

Cada caminhão pequeno possui um identificador único (`id`), gerado automaticamente com base em um contador estático, o que permite rastrear suas operações de maneira individual. A classe controla a capacidade máxima e a carga atual de lixo, garantindo que o caminhão não ultrapasse seus limites de armazenamento. Também mantém o número de viagens realizadas por dia e um limite diário, utilizado para controlar a disponibilidade operacional dos veículos.

A movimentação do caminhão é simulada através do controle de deslocamento entre zonas e a estação de transferência. Isso é feito com o uso de um tempo restante de

viagem (`tempoParaChegar`) que é decrementado a cada ciclo de simulação, até que o caminhão chegue ao destino. A lógica distingue se o caminhão está indo em direção à estação ou a uma zona, possibilitando simular rotas de ida e volta.

Um dos aspectos mais importantes da classe é o suporte a roteiros de coleta. A simulação pode definir um roteiro padrão compartilhado entre todos os caminhões, ou um roteiro individual, exclusivo para cada instância. O método `getProximaZona()` é responsável por recuperar a próxima zona do roteiro (circular), alternando automaticamente entre as zonas ao longo das viagens. Isso permite que o sistema organize a coleta de maneira cíclica, cobrindo diversas regiões da cidade.

Além disso, a classe fornece métodos para simular o processo de coleta, verificando se o caminhão pode coletar mais lixo e realizando o carregamento se houver espaço. Também é possível descarregar parte da carga na estação, registrar e resetar viagens diárias, bem como iniciar e monitorar deslocamentos.

Com todos esses mecanismos, `CaminhaoPequeno` oferece uma abstração completa para o controle logístico da frota de coleta, simulando aspectos como limite operacional diário, deslocamento, carga e roteirização. É uma peça central na dinâmica do sistema, servindo como elo entre as zonas da cidade e a estação de transferência, permitindo uma simulação realista e controlada do fluxo de resíduos.

A classe `CaminhaoPequenoPadrao`, `CaminhaoPequeno4t`, `CaminhaoPequeno8t`, `CaminhaoPequeno10t` são implementações concretas da classe abstrata `CaminhaoPequeno`, representando um modelo padrão de caminhão pequeno utilizado na simulação. Ela define os parâmetros específicos de operação desse tipo de veículo, como capacidade de carga, limite de viagens diárias e comportamento de coleta.

No construtor da classe, a capacidade do caminhão é definida como 2.000, 4.000, 8.000 ou 10.000 kg, o que indica o volume máximo de lixo que ele pode transportar por viagem. A carga inicial é sempre zero, e o limite de viagens por dia é configurado, respeitando a lógica da simulação de limitar a atuação dos caminhões para refletir restrições operacionais reais, como tempo e combustível.

O método `coletar(int quantidade)` é sobrescrito para manter a mesma lógica básica de carregamento, garantindo que o caminhão só receba lixo se houver espaço suficiente em sua carga atual. Se a operação for válida, a quantidade de lixo é adicionada e o método retorna `true`; caso contrário, a coleta é negada e retorna `false`.

3.3 Zonas

A classe `Zona` representa uma região da cidade dentro da simulação de coleta de lixo. Cada instância dessa classe simula o comportamento de uma zona urbana, incluindo a geração periódica de resíduos e o tempo necessário para que um caminhão chegue até ela a depender do horário do dia. É uma das peças fundamentais do modelo, pois define a demanda por coleta e interage diretamente com os caminhões pequenos.

Cada zona possui um nome identificador e um valor interno que representa a quantidade de lixo acumulado. A geração de lixo é feita de forma aleatória, controlada pelos limites mínimo e máximo que podem ser configurados dinamicamente. Essa geração é realizada pelo método `gerarLixo()`, que incrementa o total acumulado e retorna a quantidade gerada na iteração.

Além disso, a classe permite a coleta de lixo por meio do método `coletarLixo(int quantidade)`, que remove até a quantidade especificada do lixo acumulado, garantindo que não haja coleta superior ao disponível. O sistema exibe no console a quantidade coletada e o lixo restante, simulando uma resposta visual típica de um sistema logístico.

Alguns métodos não foram possíveis de ser implementados, mas fazem parte da classe para implementações futuras, como é o caso do método `calcularTempoViagem(int tempoAtual)` que utilizaria o horário fornecido (em minutos desde 0h) para determinar se o deslocamento até a zona ocorre em período de pico.

3.4 Estação de Transferência

A classe `EstacaoPadrao` representa uma estação de transferência de lixo que gerencia o armazenamento temporário de resíduos e a logística de transporte entre caminhões pequenos e grandes. Quando caminhões pequenos chegam, são enfileirados e descarregam gradualmente (até 1000 kg por minuto) no armazenamento da estação. Se o tempo de espera exceder um limite pré-definido (30 minutos por padrão), um caminhão grande adicional é acionado para evitar congestionamentos. O lixo acumulado é então transferido para caminhões grandes, que o transportam para o aterro em lotes de 1000 kg por minuto. Caso um caminhão grande atinja sua capacidade máxima, ele é enviado imediatamente ao aterro, garantindo um fluxo contínuo e eficiente.

O método `receberCaminhaoPequeno()` é responsável por gerenciar a chegada de caminhões pequenos à estação. Quando um caminhão pequeno chega, ele é imediatamente adicionado ao final da fila de espera (`filaCaminhoesPequenos`).

O `processarFila()` realiza duas operações principais de forma simultânea:

Primeiro, controla o tempo de espera dos caminhões. Se houver veículos na fila, incrementa o contador de tempo. Quando esse tempo excede o limite pré-definido (`tempoMaximoEspera`), o método aciona automaticamente um novo caminhão grande para auxiliar na operação, demonstrando um mecanismo de auto-regulação do sistema.

Segundo, gerencia o processo de transferência de lixo. A cada ciclo, o método tenta remover até 1000kg do caminhão que está na frente da fila. Essa transferência é feita de forma progressiva - se o caminhão ainda tiver lixo após a transferência, permanece na fila; caso contrário, é removido completamente da fila. Esse processo simula de maneira realista a operação de uma estação de transferência de resíduos.

O Método `descarregarParaCaminhaoGrande()` coordena a transferência do lixo acumulado na estação para os caminhões grandes. Ele primeiro verifica se há caminhões grandes disponíveis na lista. Se não houver, utiliza o caminhão passado como parâmetro.

O processo de transferência ocorre em lotes de até 1000kg por ciclo, mantendo um fluxo constante e controlado. Quando um caminhão grande atinge sua capacidade máxima (determinado pelo método `prontoParaPartir()`), o método automaticamente o prepara para envio ao aterro, descarregando todo o seu conteúdo e removendo-o da lista de caminhões ativos.

O método `enviarCaminhoesGrandesParaAterro()` é acionado quando há necessidade de enviar caminhões grandes ao aterro antes que estejam completamente cheios. Ele opera de forma similar ao descarregamento normal, mas com foco específico na liberação dos veículos.

O processo remove até 1000kg por ciclo do caminhão selecionado. Se o caminhão esvaziar completamente durante essa operação, é imediatamente removido da lista ativa.

4. Application

4.1 Simulacao

A classe `Simulacao` representa o núcleo do sistema de simulação de coleta de lixo urbano. Ela gerencia todo o fluxo de operações, desde a geração de lixo nas zonas urbanas até o transporte para a estação de transferência e posterior envio ao aterro. A classe implementa `Serializable` para permitir o salvamento e carregamento do estado da simulação.

A classe mantém vários atributos importantes para o funcionamento da simulação. O objeto `estatisticas` armazena todos os dados e métricas gerados durante a execução. O `timer` controla o avanço do tempo simulado, enquanto `tempoSimulado` registra quantos minutos já se passaram na simulação. A `estacao` representa a estação central de transferência de lixo, e a `filaCaminhoesPequenos` contém todos os caminhões de coleta disponíveis. Além disso, há cinco objetos `Zona` que representam diferentes regiões da cidade (norte, sul, leste, centro e sudeste).

O método `Iniciar()` é responsável por inicializar toda a simulação. Ele começa criando a estação central e configurando os parâmetros de geração de lixo para cada zona urbana. Em seguida, define dois roteiros de coleta distintos - um padrão e um inverso - e cria oito caminhões de diferentes capacidades (variando de 2 a 10 toneladas). Cada caminhão é posicionado em sua zona inicial e o timer é iniciado para avançar a simulação a cada segundo de tempo real.

A classe oferece três métodos fundamentais para controle de execução: `pausar()` interrompe temporariamente a simulação, `continuarSimulacao()` retoma a execução após uma pausa, e `encerrar()` finaliza completamente a simulação, cancelando o timer e liberando recursos.

O método `atualizarSimulacao()` coordena quatro processos principais: geração de lixo nas zonas urbanas, operação dos caminhões coletores, processamento na estação de transferência e operações periódicas. A geração de lixo segue parâmetros configurados para cada zona, enquanto os caminhões realizam coleta progressiva (200kg por vez) e retornam à estação quando cheios. Na estação, o lixo é transferido para caminhões maiores, que periodicamente são enviados ao aterro.

O sistema coleta extensivos dados operacionais através do objeto `estatisticas`, incluindo quantidades de lixo gerado e coletado, número de viagens realizadas e status dos veículos. Relatórios periódicos são gerados a cada 5 minutos simulados, proporcionando uma visão detalhada do andamento da simulação. A cada 100 minutos (representando um novo dia), os contadores de viagens dos caminhões são resetados.

5. Testes

5.1 Geração de lixo por zona

A cidade é dividida em cinco zonas (sul, norte, centro, leste e sudeste) com geração diária de lixo definida por intervalos configuráveis:

```
// Configuração das zonas
zonaNorte.configurarGeracaoLixo(2000, 5000);
zonaSul.configurarGeracaoLixo(3000, 6000);
zonaLeste.configurarGeracaoLixo(1500, 4500);
zonaCentro.configurarGeracaoLixo(2500, 5500);
zonaSudeste.configurarGeracaoLixo(1800, 4000);
```

```
--- GERACAO DE LIXO NAS ZONAS ---
Zona Norte: Gerou 2473kg de lixo. Total acumulado: 8622kg.
Zona Sul: Gerou 5328kg de lixo. Total acumulado: 13577kg.
Zona Leste: Gerou 2472kg de lixo. Total acumulado: 6401kg.
Zona Centro: Gerou 5047kg de lixo. Total acumulado: 8950kg.
Zona Sudeste: Gerou 3315kg de lixo. Total acumulado: 7204kg.
```

O sistema permite a configuração personalizada da produção diária de lixo para cada zona, estabelecendo intervalos variáveis onde o mínimo de geração é de 2.000 kg e o máximo atinge 6.000 kg por dia. A variação diária dentro desses intervalos é calculada aleatoriamente, introduzindo um elemento dinâmico que simula flutuações naturais na produção de resíduos urbanos.

5.2 Variedade de caminhões pequenos

Caminhões pequenos com capacidades de 2, 4, 8 e 10 toneladas percorrem as zonas, realizando viagens com tempos determinados aleatoriamente, conforme parâmetros para horários de pico e fora de pico:

```
// Criar caminhões padrão
filaCaminhoesPequenos.enqueue(new CaminhaoPequenoPadrao()); // ID 0001 (2t)
filaCaminhoesPequenos.enqueue(new CaminhaoPequeno4t()); // ID 0002 (4t)
filaCaminhoesPequenos.enqueue(new CaminhaoPequeno8t()); // ID 0003 (8t)
filaCaminhoesPequenos.enqueue(new CaminhaoPequeno10t()); // ID 0004 (10t)
```

```
// Definir roteiros
Zona[] roteiroPadrao = {zonaCentro, zonaNorte, zonaSul, zonaLeste, zonaSudeste};
Zona[] roteiroInverso = {zonaSudeste, zonaLeste, zonaSul, zonaNorte, zonaCentro};
```

```
Simulação iniciada...
Caminhão ID 0001 (2000kg, 0/10 viagens) iniciando na Zona Centro (Padrão)
Caminhão ID 0002 (4000kg, 0/6 viagens) iniciando na Zona Centro (Padrão)
Caminhão ID 0003 (8000kg, 0/4 viagens) iniciando na Zona Centro (Padrão)
Caminhão ID 0004 (10000kg, 0/2 viagens) iniciando na Zona Centro (Padrão)
Caminhão ID 0006 (Roteiro 2) (2000kg, 0/10 viagens) iniciando na Zona Sudeste (Inverso)
Caminhão ID 0007 (Roteiro 2) (4000kg, 0/6 viagens) iniciando na Zona Sudeste (Inverso)
Caminhão ID 0008 (Roteiro 2) (8000kg, 0/4 viagens) iniciando na Zona Sudeste (Inverso)
Caminhão ID 0009 (Roteiro 2) (10000kg, 0/2 viagens) iniciando na Zona Sudeste (Inverso)
```

A simulação utiliza caminhões pequenos com capacidades variadas (2, 4, 8 e 10 toneladas) para coletar lixo nas diferentes zonas da cidade. Cada caminhão segue um roteiro predefinido, transportando os resíduos até a estação central de transferência. Embora o modelo original previsse a implementação de tempos de viagem variáveis, considerando horários de pico e fora de pico, essa funcionalidade não foi possível de ser implementada na versão atual do sistema.

Ainda assim, os caminhões operam em ciclos contínuos, coletando lixo progressivamente (200 kg por vez) e retornando à estação quando atingem sua capacidade máxima, mantendo a lógica central da simulação funcional. Futuras atualizações poderão incorporar essa complexidade adicional para maior fidelidade ao cenário urbano real.

5.3 Viagens máximas

Cada caminhão pequeno possui um número máximo de viagens diárias e, ao atingir sua capacidade, dirige-se para uma das duas estações de transferência.

```
public CaminhaoPequeno10t() {
    super();
    this.capacidade = 10000;
    this.cargaAtual = 0;
    this.maxViagensPorDia = 2;
}
```

```

    */
    public boolean podeViajar() {
        return viagensRealizadasHoje < maxViagensPorDia;
    }

    * Registrar viagem.
    public void registrarViagem() {
        viagensRealizadasHoje++;
    }

    * Resetar viagens.
    public void resetarViagens() {
        viagensRealizadasHoje = 0;
    }

```

```

if (!caminhao.podeViajar()) {
    String msg = "Caminhão " + caminhao.getId() + " atingiu o limite de viagens.";
    estatisticas.adicionarLog(msg);
    noAtual = noAtual.getProx();
    continue;
}

```

Cada caminhão pequeno (com capacidades de 2, 4, 8 ou 10 toneladas) possui um limite máximo de viagens diárias pré-definido. Quando atingem sua capacidade máxima de carga durante a coleta em uma zona urbana, os caminhões automaticamente se direcionam para uma das duas estações de transferência disponíveis no sistema. Esse limite de viagens foi implementado para simular restrições operacionais reais, como jornadas de trabalho e manutenção periódica dos veículos.

Atualmente, quando um caminhão pequeno atinge seu número máximo de viagens diárias, ele é automaticamente recolhido e armazenado na fila, onde aguarda até o próximo ciclo de operação (simulando o início de um novo dia). Essa abordagem garante que os veículos respeitem seus limites operacionais e permite um gerenciamento simples e sequencial da frota. No entanto, isso pode atrapalhar na logística de gerar um novo caminhão grande ao atingir o limite de tempo de espera, por isso, futuramente estes caminhões serão armazenados em uma Lista de espera.

5.4 Filas de caminhões pequenos

Nas estações de transferência, os caminhões pequenos aguardam em fila para descarregar o lixo em caminhões grandes de 20 toneladas:

```
// Processar transferência progressiva (1000 kg por minuto)
if (filaCaminhoesPequenos.tamanho() > 0) {
    CaminhaoPequeno pequeno = filaCaminhoesPequenos.verProximoDaFila();
    if (pequeno != null && pequeno.getCargaAtual() > 0) {
        // Transferir até 1000 kg por minuto
        int quantidadeTransferida = Math.min(1000, pequeno.getCargaAtual());
        int descarregado = pequeno.descarregar(quantidadeTransferida); // Modificado para
        lixoArmazenado += descarregado;

        System.out.println("Caminhão pequeno ID " + pequeno.getId() +
            " transferiu " + descarregado + "kg para estação " + nome/* +
            " (restante: " + pequeno.getCargaAtual() + "kg)*/");

        // Se o caminhão esvaziou, remove da fila
        if (pequeno.getCargaAtual() == 0) {
            filaCaminhoesPequenos.dequeue();
        }
    }
}
}
```

```
--- PROCESSAMENTO DA ESTACAO ---
Caminhões pequenos esperando há 3 minutos (limite: 30 minutos)
Caminhão pequeno ID 0002 transferiu 1000kg para estação Estação Central
```

Quando os caminhões pequenos chegam às estações de transferência, eles são direcionados para uma fila de espera, onde aguardam sua vez para descarregar o lixo na estação de transferência.

```
public void descarregarParaCaminhaoGrande(CaminhaoGrande novoCaminhao) {
    CaminhaoGrande caminhaoAtual = listaCaminhoesGrandes.verProximoDaLista();

    if (caminhaoAtual == null) {
        caminhaoAtual = novoCaminhao;
        listaCaminhoesGrandes.add(caminhaoAtual, 0);
    }

    // Transferir até 1000 kg por minuto para o caminhão grande
    int quantidadeTransferida = Math.min(1000, lixoArmazenado);
    if (quantidadeTransferida > 0) {
        caminhaoAtual.carregar(quantidadeTransferida);
        lixoArmazenado -= quantidadeTransferida;
        System.out.println("Estação " + nome + " transferiu " + quantidadeTransferida +
            "kg para caminhão grande (total: " + caminhaoAtual.getCargaAtual() + "
    )

    if (caminhaoAtual.prontoParaPartir()) {
        System.out.println("Caminhão grande está cheio e será enviado ao aterro com " + caminhaoAtual.getCargaAtual());
        caminhaoAtual.descarregar(caminhaoAtual.getCargaAtual());
        listaCaminhoesGrandes.remove(0);
    }
}
}
```

5.5 Limite de espera de caminhões grandes

Na estação de transferência, os caminhões grandes de 20 toneladas recebem progressivamente a carga dos veículos menores através de um processo controlado (1000

kg por minuto simulado). Quando atingem sua capacidade total ou quando atingem um limite de espera de 30 minutos de simulação, estes caminhões maiores transportam a carga consolidada para o aterro sanitário final.

```
--- PROCESSAMENTO DA ESTACAO ---
Caminhões pequenos esperando há 11 minutos (limite: 30 minutos)
Caminhão pequeno ID 0001 transferiu 1000kg para estação Estação Central
Estação Estação Central transferiu 1000kg para caminhão grande (total: 3000kg)
Caminhão grande descarregou 1000kg no aterro. Restante: 2000kg.
Caminhão grande enviado para o aterro (tolerância atingida) com 1000kg (restant
```

5.6 Limite de espera de caminhões pequenos

Se o tempo máximo de espera dos caminhões pequenos nas estações for excedido, a simulação acionar a inclusão de um novo caminhão grande:

```
if (caminhoesEsperando) {
    tempoEsperaAtual++;

    System.out.println("Caminhões pequenos esperando há " + tempoEsperaAtual +
        " minutos (limite: " + tempoMaximoEspera + " minutos)");

    if (tempoEsperaAtual >= tempoMaximoEspera) {
        System.out.println("ALERTA: Tempo de espera excedido! Acionando caminhão gra
        CaminhaoGrande novoCaminhao = new CaminhaoGrandePadrao();
        listaCaminhoesGrandes.add(novoCaminhao, 0);
        tempoEsperaAtual = 0;
    }
} else {
    tempoEsperaAtual = 0;
}
```

Quando o tempo de espera dos caminhões pequenos na estação de transferência ultrapassa o limite máximo pré-definido (ex.: 30 minutos), o sistema aciona automaticamente um caminhão grande adicional como medida de contingência.

5.7 Relatórios e armazenamento de logs

A Simulação deverá fornecer logs e estatísticas em tempo real, evidenciando a dinâmica dos eventos.

A cada 5 minutos de simulação o sistema emite um relatório com informações de lixo gerado, lixo recolhido, porcentagem de coleta realizada, número de viagens realizadas, numero de caminhoes em transito e o numero de caminhoes em coleta:

```
=== RELATÓRIO DE ESTATÍSTICAS ===
Total de lixo gerado: 546.584 kg
Total de lixo coletado: 44.600 kg
Percentual de coleta: 8,2%
Total de viagens realizadas: 6
Caminhões em trânsito: 2
Caminhões em coleta: 6
=====
```

Além disso, todas as informações de terminal são armazenadas em logs, permitindo o acompanhamento histórico e a otimização contínua do processo.

```
320 =====
321
322 [23/05/2025 03:09:44] Zona Norte: Gerou 2363kg de lixo. Total acumulado: 71204kg.
323 [23/05/2025 03:09:44] Zona Sul: Gerou 4598kg de lixo. Total acumulado: 99401kg.
324 [23/05/2025 03:09:44] Zona Leste: Gerou 2166kg de lixo. Total acumulado: 62249kg.
325 [23/05/2025 03:09:44] Zona Centro: Gerou 5286kg de lixo. Total acumulado: 71513kg.
326 [23/05/2025 03:09:44] Zona Sudeste: Gerou 3207kg de lixo. Total acumulado: 51868kg.
327 [23/05/2025 03:09:44] Caminhão 0001 coletou 200kg na Zona Norte. Carga: 1800/2000
328 [23/05/2025 03:09:44] Caminhão 0002 coletou 200kg na Zona Norte. Carga: 3200/4000
329 [23/05/2025 03:09:44] Caminhão 0003 coletou 200kg na Zona Centro. Carga: 4400/8000
330 [23/05/2025 03:09:44] Caminhão 0004 coletou 200kg na Zona Centro. Carga: 4400/10000
331 [23/05/2025 03:09:44] Caminhão 0006 coletou 200kg na Zona Leste. Carga: 1200/2000
332 [23/05/2025 03:09:44] Caminhão 0007 na fila da estação central
333 [23/05/2025 03:09:44] Caminhão 0008 coletou 200kg na Zona Sudeste. Carga: 4400/8000
334 [23/05/2025 03:09:44] Caminhão 0009 coletou 200kg na Zona Sudeste. Carga: 4400/10000
335 [23/05/2025 03:09:45] Zona Norte: Gerou 2519kg de lixo. Total acumulado: 73323kg.
336 [23/05/2025 03:09:45] Zona Sul: Gerou 3364kg de lixo. Total acumulado: 102765kg.
337 [23/05/2025 03:09:45] Zona Leste: Gerou 3892kg de lixo. Total acumulado: 65941kg.
338 [23/05/2025 03:09:45] Zona Centro: Gerou 2725kg de lixo. Total acumulado: 73838kg.
```

6. Conclusão

Este relatório detalhou o funcionamento da simulação de coleta de resíduos urbanos, desde a geração de lixo nas zonas da cidade até o transporte final para o aterro sanitário. O sistema atual já oferece uma base sólida para análise operacional. No entanto, para responder à pergunta central: "Quantos caminhões de 20 toneladas são necessários para atender à demanda de Teresina?" são necessárias melhorias e expansões no código. Atualmente, o sistema não considera variações no tempo de deslocamento devido a tráfego, horários de pico ou condições climáticas.

Para otimizar a eficiência do sistema, planeja-se substituir as atuais filas por estruturas de dados mais sofisticadas. Listas prioritárias permitirão atender primeiro zonas críticas, como mercados públicos e hospitais, onde o acúmulo de lixo representa maiores riscos sanitários. Além disso, a implementação de tabelas hash (HashMap) viabilizará consultas instantâneas aos veículos por ID ou zona de atuação, agilizando decisões operacionais.

A estrutura do código passará por uma significativa reorganização para aumentar sua manutenibilidade e escalabilidade. A modularização separará claramente as diferentes lógicas do sistema – como geração de lixo, roteirização e coleta de estatísticas – em classes especializadas.

Para transformar dados brutos em insights acionáveis, o sistema será equipado com ferramentas avançadas de visualização. Gráficos de carga acumulada por zona ajudarão a identificar padrões de geração de resíduos, enquanto histogramas de viagens necessárias fornecerão uma visão clara da demanda operacional. Modelos de otimização linear, que considera capacidade das estações, tempo médio de viagem e margens de segurança, permitirão calcular com precisão o número mínimo necessário de caminhões grandes.