

Documentação do Código do Simulador de Elevador: Modelagem, Estruturas de Dados e Heurísticas

BRUNO IBIAPINA SILVA MARQUES

Instituto de Ensino Superior - ICEV

Curso de Engenharia de Software

Email: bruno.marques@somosicev.com

PAULO HENRIQUE FERNANDES DA CRUZ

Instituto de Ensino Superior - ICEV

Curso de Engenharia de Software

Email: paulo_henrique.cruz@somosicev.com

Abstract—Este documento descreve a modelagem e implementação de um sistema de simulação de elevadores inteligentes utilizando Java e estruturas de dados personalizadas. O foco é educacional, promovendo o uso de listas, filas e algoritmos de controle em tempo discreto.

I. INTRODUÇÃO

O Simulador de Elevador Inteligente é um projeto acadêmico que visa replicar o comportamento de sistemas de transporte vertical em edifícios. A aplicação é desenvolvida em Java, utilizando princípios da programação orientada a objetos, estruturas de dados manuais e simulação discreta. O objetivo principal é estudar o impacto de diferentes heurísticas no desempenho de um sistema multi-elevador.

II. MODELAGEM DO SISTEMA

A. Diagrama Geral de Classes (Resumo)

- **EntidadeSimulavel (abstract)**: base para atualização temporal.
- **Andar, Elevador, CentralDeControle, Predio, Simulador**: herdam de EntidadeSimulavel.
- **Pessoa, Painelelevador, Lista, Ponteiro, Fila, HeuristicaElevador**.

III. DESCRIÇÃO DAS PRINCIPAIS CLASSES

- **Simulador**: executa ciclos temporais, gera pessoas e coleta métricas.
- **Predio**: estrutura principal contendo os andares e a central.
- **CentralDeControle**: coordena a lógica de cada elevador.
- **Elevador**: executa movimentação, embarque/desembarque.
- **Andar**: mantém fila de espera e interações com o painel.
- **Pessoa**: possui dados como origem, destino e tempos.

IV. ESTRUTURAS DE DADOS UTILIZADAS

A. Lista Encadeada

Implementação manual com inserção/remoção em ambos os extremos.

- Usada em: passageiros, elevadores ativos, andares.

- Métodos principais: `inserirInicio`, `inserirFim`, `removerInicio`, `getTamanho`.

B. Fila (FIFO)

Especialização da Lista com inserção no final e remoção no início.

- Usada para: fila de pessoas aguardando elevador.

C. Ponteiro

Classe que representa um nó da Lista, contendo um objeto e um ponteiro para o próximo.

V. ALGORITMOS IMPLEMENTADOS

A. Atualização do Sistema

O método `atualizar(int minuto)` é invocado a cada ciclo para todos os objetos simuláveis.

B. Movimento do Elevador

- Se parado: verifica chamadas pendentes.
- Se em movimento: decide parar ou mudar de direção com base nos passageiros e na fila de espera.

C. Heurísticas

- **SEM_HEURISTICA**: FCFS (First Come, First Served). Elevadores atendem chamadas na ordem em que surgem, sem critério otimizado.
- **OTIMIZACAO_TEMPO_ESPERA**: Escolhe o elevador que pode atender a chamada mais rapidamente, minimizando o tempo de espera da pessoa..
- **OTIMIZACAO_ENERGIA**: Tenta reduzir o número de movimentos dos elevadores, diminuindo o consumo de energia.

D. Estatísticas

- Pessoas geradas e transportadas.
- Tempo médio de espera.
- Tempo médio de viagem.

VI. INTERFACE GRÁFICA

A interface gráfica foi desenvolvida em Java Swing e exibe:

- A posição e ocupação dos elevadores.
- As filas de espera por andar.
- Estatísticas atualizadas em tempo real.

VII. CONSIDERAÇÕES FINAIS

O projeto serve como ferramenta de aprendizado em estruturas de dados e simulação. Sua modularidade permite fácil expansão para incluir novas heurísticas, sensores simulados e algoritmos de aprendizado de máquina.

AGRADECIMENTOS

Agradecemos ao corpo docente do curso de Engenharia de Software do ICEV pelo suporte técnico e teórico ao longo do desenvolvimento do projeto.

REFERENCES

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.
- [2] R. Lafore, *Data Structures and Algorithms in Java*, 2nd ed. Indianapolis, IN: Sams Publishing, 2002.
- [3] M. T. Goodrich, R. Tamassia, and M. H. Goldwasser, *Data Structures and Algorithms in Java*, 6th ed. Hoboken, NJ: Wiley, 2014.
- [4] E. Horowitz, S. Sahni, and D. Mehta, *Fundamentals of Data Structures in C++*, 2nd ed. Hyderabad, India: Universities Press, 2008.
- [5] D. E. Knuth, *The Art of Computer Programming*, Vol. 1: Fundamental Algorithms, 3rd ed. Boston, MA: Addison-Wesley, 1997.