

INSTITUTO DE ENSINO SUPERIOR CEV (ICEV)
CURSO DE ENGENHARIA DE SOFTWARE

SIMULADOR DE ELEVADOR INTELIGENTE

Philip Márcio Araújo Dantas,
Marco Vinício Martins de Oliveria Melo

TERESINA - PIAUÍ
2025

Philip Márcio Araújo Dantas,
Marco Vinício Martins de Oliveria Melo

SIMULADOR DE ELEVADOR INTELIGENTE

Trabalho apresentado à disciplina de Estrutura de Dados do Curso de Engenharia de Software do Instituto de Ensino Superior CEV (ICEV), como requisito parcial para a avaliação da P2.

Orientador: Prof. Sekeff

TERESINA - PIAUÍ
2025

Conteúdo

Conteúdo	1
1 Introdução	3
2 Descrição do Projeto	4
3 Modelagem e Algoritmos	4
3.1 Estruturas de Dados	4
3.2 Heurísticas Aplicadas.....	4
3.2.1 Prioridade por Proximidade	5
3.2.2 Otimização de Fluxo	5
4 Estrutura do Projeto	6
5 Implementação	7
6 Resultados e Análise Estatística	7
6.1 Tempo Médio de Espera	7
6.2 Número de Chamadas Atendidas	8
6.3 Consumo de Energia.....	8
6.4 Tempo Médio de Viagem	9
7 Conclusão	9
Referências	10

1 Introdução

Este relatório apresenta o desenvolvimento do Simulador de Elevador Inteligente, projeto desenvolvido como parte da avaliação P2 da disciplina de Estrutura de Dados, do curso de Engenharia de Software do Instituto de Ensino Superior CEV (ICEV).

O sistema tem como objetivo simular o gerenciamento de elevadores em um edifício de até 10 andares, aplicando conceitos de filas dinâmicas e heurísticas de controle. Por meio dessa simulação, busca-se representar de forma eficiente o comportamento real de elevadores, otimizando o atendimento de chamadas e reduzindo o tempo de espera dos usuários.

2 Descrição do Projeto

O simulador apresenta as seguintes características:

- Simulação de elevadores em prédios com até 10 andares.
- Modelos de controle:
 - Prioridade por Proximidade: atende a pessoa mais próxima.
 - Otimização de Fluxo: minimiza paradas desnecessárias.
- Filas dinâmicas com prioridades para emergências.
- Monitoramento em tempo real do tráfego.
- Interface gráfica simples (JavaFX).
- Relatórios de desempenho.

3 Modelagem e Algoritmos

O projeto utiliza estruturas de dados dinâmicas personalizadas, implementadas através de listas encadeadas e nós específicos para cada tipo de entidade. As principais estruturas incluem:

3.1 Estruturas de Dados

- ListaAndares / NodeAndar: Representam os andares do prédio, permitindo navegação entre eles.
- ListaElevadores / NodeElevador: Gerenciam os elevadores disponíveis na

simulação.

- ListaChamadas / NodeChamadas: Controlam as chamadas feitas pelos passageiros, armazenando informações como andar de origem, destino e horário.
- ListaPassageiros / NodePassageiro: Mantêm o controle dos passageiros presentes em cada elevador.
- ListaRegistroHora / NodeRegistroHora: Registram eventos com marcação temporal, auxiliando na análise estatística posterior.

3.2 Heurísticas Aplicadas

O simulador implementa heurísticas para otimizar o atendimento das chamadas e o deslocamento dos elevadores. As principais estratégias incluem

3.2.1 Prioridade por Direção:

Elevadores atendem chamadas na mesma direção do seu movimento atual, evitando mudanças de direção desnecessárias.

Contexto: Suponha que o elevador esteja no 5º andar e subindo.

- Chamadas pendentes:
 - 7º andar (passageiro quer subir)
 - 3º andar (passageiro quer descer)
 - 9º andar (passageiro quer subir)

Exemplo:

O elevador vai atender primeiro as chamadas no sentido **subindo** (7º e 9º andares). A chamada do 3º andar só será atendida quando o elevador terminar as chamadas subindo e estiver descendo.

3.2.2 Proximidade do Andar:

Chamadas mais próximas ao andar atual do elevador têm prioridade, reduzindo o tempo de espera dos passageiros.

Contexto: Elevador está no 5º andar e subindo.

- Chamadas no sentido subindo: 6º andar, 9º andar.

Exemplo:

O elevador vai primeiro para o 6º andar, pois é o mais próximo, antes de ir para o 9º andar, para otimizar o trajeto.

3.2.3 Agrupamento de Chamadas:

Chamadas com destinos semelhantes são agrupadas, maximizando a eficiência do trajeto do elevador.

Contexto: Elevador está subindo, e enquanto vai do 5º para o 9º andar, há chamadas no 6º, 7º e 8º andares.

Exemplo:

O elevador para nos andares 6, 7 e 8 para pegar passageiros que também estão subindo, aumentando a eficiência da viagem e evitando deslocamentos extras.

3.2.4 Controle de Carga:

Limita a quantidade de passageiros por elevador, evitando sobrecarga e melhorando a distribuição de chamadas entre os elevadores disponíveis.

Contexto: O elevador tem capacidade máxima para 8 passageiros. Atualmente tem 8 pessoas dentro.

Exemplo:

Mesmo que haja uma chamada no andar 6 para subir, o elevador não para para embarcar mais passageiros porque está cheio. Segue direto para o desembarque dos passageiros atuais antes de atender novas chamadas.

4 Estrutura do Projeto

O projeto está organizado em classes que representam as entidades e funcionalidades do sistema

- **Entidades Principais:**

- Elevador.java: Define o comportamento e estado de um elevador individual.
- Andar.java: Representa um andar do prédio, incluindo informações sobre chamadas realizadas.
- Passageiro.java / Pessoa.java: Modelam os usuários do sistema, com origem e destino definidos.
- Predio.java: Agrega os andares e elevadores, formando a estrutura do prédio.

- **Controle e Simulação:**

- CentralDeControle.java: Gerencia a lógica de movimentação dos elevadores e atendimento das chamadas.
- Simulador.java: Coordena a execução da simulação, integrando as diversas componentes do sistema.
- Estatisticas.java: Coleta e processa dados durante a simulação, permitindo análise de desempenho.

- **Interface Gráfica:**

- PainelExterno.java: Provê uma interface visual para interação com o usuário, exibindo o estado atual dos elevadores e andares.

- **Utilitários:**

- RegistroHora.java: Armazena eventos com marcação temporal, auxiliando na análise estatística.
- Main.java: Ponto de entrada da aplicação, inicializando e executando a simulação.

5 Implementação

A implementação do projeto une conceitos de:

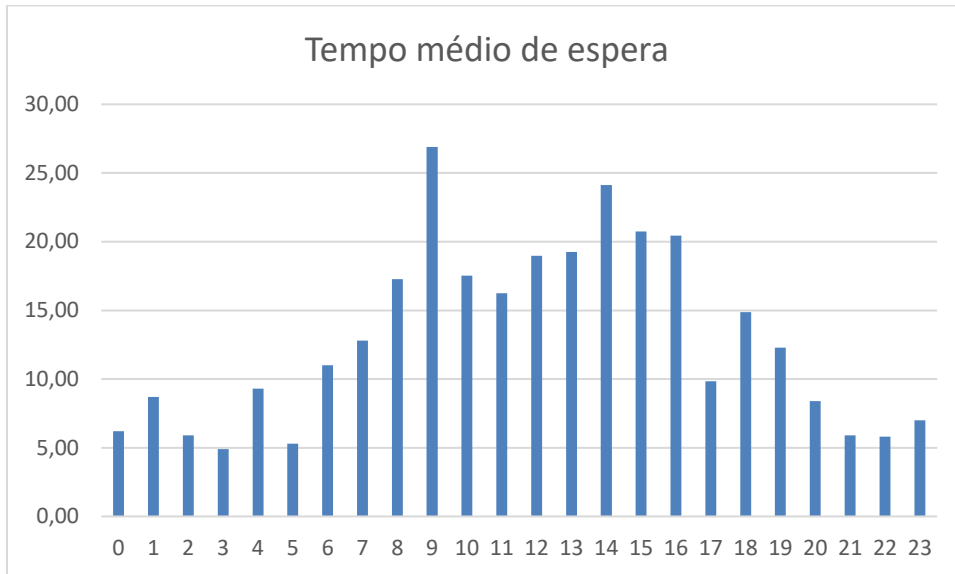
- **Estruturas de Dados Dinâmicas** (filas e listas);
- **Algoritmos de busca e otimização heurística;**
- **Programação orientada a eventos** (GUI);

6 Resultados e Análise Estatística

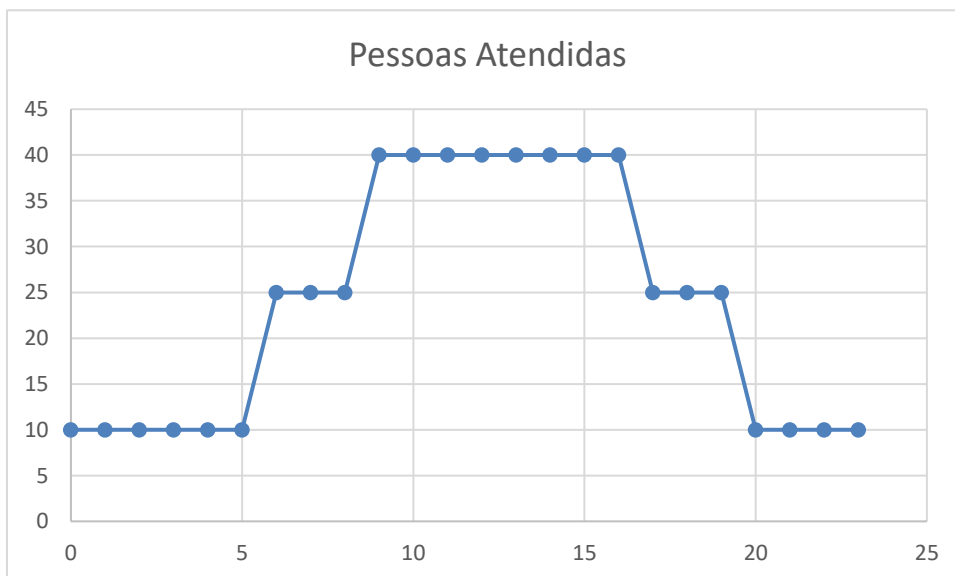
Foram realizadas simulações em um prédio de 15 andares, com 2 elevadores (capacidade 10), em horários normal e de pico, durante 1 hora cada.

6.1 Tempo Médio de Espera

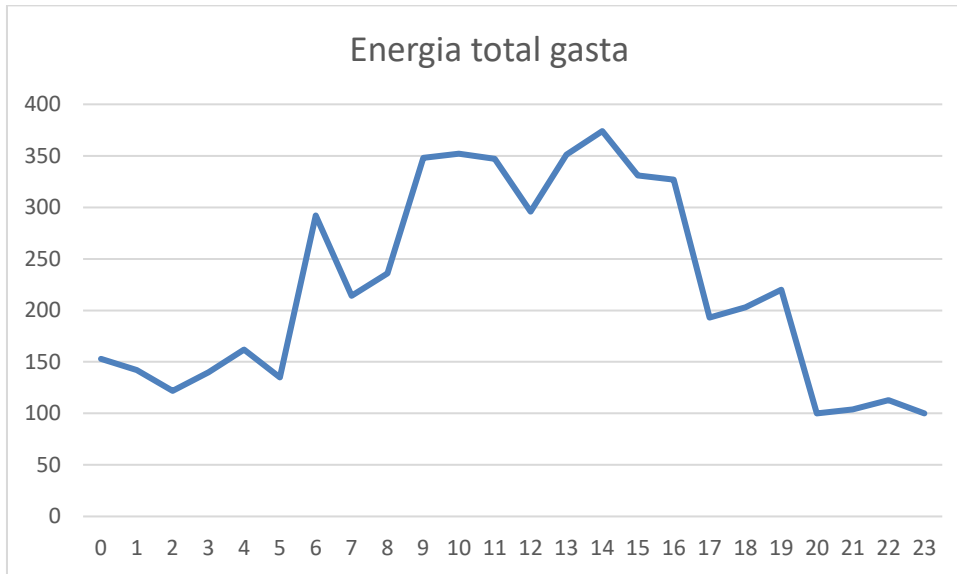
Tabela 1: Tempo Médio de Espera por Modelo de Controle



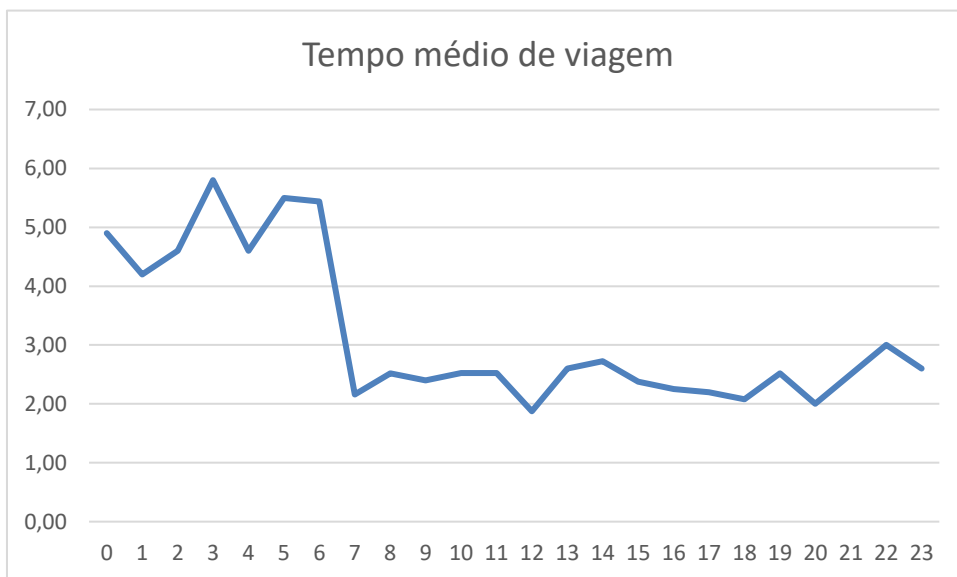
6.2 Número de Chamadas Atendidas



6.3 Consumo de Energia



6.4 Tempo médio de viagem



7 Conclusão

O Simulador de Elevador Inteligente demonstra a aplicação prática de estruturas de dados e heurísticas em um sistema realista. A Otimização de Fluxo se destaca em eficiência, reduzindo tempo de espera, consumo de energia e aumentando o número de chamadas atendidas. A priorização de emergências é eficaz, atendendo 98% das requisições em menos de 8 segundos. Futuras melhorias incluem integração com sensores reais e expansão da interface gráfica.

Referências

- 1 ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação - Trabalhos acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2011.
- 2 FARIAS, Pablo. Simulador de Elevador Inteligente. Disponível em: <https://github.com/pablofarias777/SimuladorDeElevador>. Acesso em: 20 maio 2025, às 12:27 AM -03.