

Documentação do Projeto Elevadores

Reinaldo Fernandes Coelho and Felipe da Silva Rocha Cardoso

Resumo—O sistema de simulação de elevadores modela o funcionamento de elevadores em um prédio, gerenciando o transporte de pessoas com base em diferentes heurísticas (ordem de chegada, otimização de tempo ou energia). O sistema inclui classes para representar andares, elevadores, painéis de chamada, pessoas e estatísticas, além de uma interface gráfica para configuração e visualização. Estruturas de dados como filas de prioridade, listas encadeadas e iteradores são usadas para gerenciar pessoas, destinos e logs. A simulação permite configurar parâmetros como número de andares, elevadores, capacidade e tipo de painel (único botão, dois botões ou painel numérico), exibindo estatísticas como tempo médio de espera e energia consumida. O sistema suporta serialização para salvar/carregar estados e testes para avaliar diferentes configurações.

Palavras-chave: simulação de elevadores, heurísticas, fila de prioridade, lista encadeada, interface gráfica, painel de chamada, estatísticas, serialização, tempo de espera, consumo de energia.

Abstract—The elevator simulation system models the operation of elevators in a building, managing passenger transport based on different heuristics (first-come-first-served, time optimization, or energy optimization). The system comprises classes to represent floors, elevators, call panels, people, and statistics, along with a graphical interface for configuration and visualization. Data structures such as priority queues, linked lists, and iterators are used to manage people, destinations, and logs. The simulation allows configuration of parameters like the number of floors, elevators, capacity, and panel type (single button, two buttons, or numeric panel), displaying statistics such as average waiting time and energy consumption. The system supports serialization for saving/loading states and testing for evaluating different configurations.

Keywords: elevator simulation, heuristics, priority queue, linked list, graphical interface, call panel, statistics, serialization, waiting time, energy consumption.

Introdução

Esta documentação descreve o sistema de simulação de elevadores, detalhando as funcionalidades e estruturas de dados de cada classe conforme especificado no documento fornecido. O sistema simula o comportamento de elevadores em um prédio, gerenciando pessoas, chamadas de elevadores, movimentação e estatísticas. A documentação segue o padrão IEEE, organizada em blocos de texto com destaque para os tópicos, separando claramente funcionalidades e estruturas de dados de cada classe.

Escopo

O sistema simula um prédio com múltiplos andares e elevadores, coordenando o transporte de pessoas com base em diferentes heurísticas e tipos de painéis de chamada. Inclui uma interface gráfica para configuração e visualização, além de funcionalidades para coleta de estatísticas e registro de logs. As classes são projetadas para serem modulares, com estruturas de dados específicas para gerenciar filas, listas e prioridades.

Organização do Documento

Cada seção descreve uma classe do sistema, dividida em:

- **Funcionalidades:** Descrição das principais responsabilidades e comportamentos da classe

- **Estruturas de Dados:** Detalhamento das estruturas utilizadas para armazenamento e manipulação de dados
- **Métodos Principais:** Lista dos métodos mais relevantes, com suas funcionalidades

1. Andar

Funcionalidades

Representa um andar do prédio na simulação de elevadores. Gerencia a fila de pessoas aguardando no andar. Controla o painel de chamada do elevador, que pode ser de três tipos: único botão, dois botões (subir/descer) ou painel numérico. Permite adicionar e remover pessoas da fila de espera. Registra chamadas no painel com base no tipo configurado. Exibe a lista de pessoas aguardando no andar.

Estruturas de Dados

- **FilaPrioridade** pessoasAguardando: Fila de prioridade que armazena objetos Pessoa, priorizando pessoas com necessidades especiais
- **PainelElevador** painel: Objeto que representa o painel de chamada do andar, configurado com um tipo específico (TipoPainel)

Métodos Principais

- **adicionarPessoa(Pessoa pessoa):** Adiciona uma pessoa à fila de espera e aciona o painel conforme o tipo (chamada geral, subir/descer ou andar específico)
- **removerPessoa(Pessoa pessoa):** Remove uma pessoa específica da fila de espera
- **listarPessoasAguardando():** Exibe as pessoas aguardando no andar
- **getNumero():** Retorna o número do andar
- **getPessoasAguardando():** Retorna a fila de pessoas aguardando
- **getPainel():** Retorna o objeto PainelElevador

2. CentralDeControle

Funcionalidades

Gerencia múltiplos elevadores no prédio, coordenando suas ações. Atualiza o estado de todos os elevadores a cada minuto simulado. Escolhe o próximo destino de um elevador com base em uma heurística selecionada (ordem de chegada, otimização de tempo ou energia). Evita conflitos, garantindo que um andar não seja atribuído a múltiplos elevadores simultaneamente.

Estruturas de Dados

- **Lista elevadores:** Lista encadeada que armazena objetos Elevador

Métodos Principais

- **atualizar(int minutoSimulado):** Atualiza o estado de todos os elevadores a cada minuto
- **escolherProximoDestino(Elevador elevador):** Determina o próximo destino com base na heurística:

- Modelo 1 (Ordem de Chegada): Escolhe o próximo andar com chamada na direção atual, ordenando andares de forma crescente (subindo) ou decrescente (descendo)
- Modelo 2 (Otimização de Tempo): Escolhe o andar com maior pontuação, considerando número de pessoas e chamadas, ponderado pela distância
- Modelo 3 (Otimização de Energia): Escolhe o andar mais próximo com chamadas, priorizando menor distância e maior número de pessoas
- getElevadores(): Retorna a lista de elevadores
- getSimulador(): Retorna o objeto Simulador

3. Elevador

Funcionalidades

Representa um elevador, gerenciando posição, direção, pessoas a bordo e destinos Realiza embarque e desembarque de pessoas Move-se entre andares conforme destinos atribuídos pela CentralDeControle Registra logs de ações, como embarque, desembarque e escolha de destino Consome energia com base em movimento e paradas, calculada pela classe Estatísticas

Estruturas de Dados

- Fila pessoasNoElevador: Fila que armazena pessoas dentro do elevador
- Lista destinos: Lista encadeada que armazena os andares de destino
- Lista logs: Lista de logs que registra as ações do elevador

Métodos Principais

- atualizar(int minutoSimulado): Atualiza o estado do elevador, movendo-o, desembarcando/embarcando pessoas e escolhendo novos destinos
- desembarcarPessoas(): Remove pessoas que chegaram ao destino
- embarcarPessoas(int andarAtual): Adiciona pessoas do andar atual, respeitando capacidade máxima e prioridade
- adicionarLog(int minutoSimulado, String decisao): Registra uma ação no log
- getAndar(int numeroAndar): Obtém o objeto Andar correspondente
- Getters e setters: Para atributos como id, andarAtual, pessoasNoElevador, destinos, subindo, heuristica, predio

4. EntidadeSimulavel

Funcionalidades

Classe abstrata que define a interface para entidades atualizáveis na simulação, como Elevador, CentralDeControle e Predio Fornece um método abstrato atualizar, implementado pelas subclasses

Métodos Principais

- atualizar(int minutoSimulado): Método abstrato que atualiza o estado da entidade a cada minuto simulado

5. Estatísticas

Funcionalidades

Coleta e armazena estatísticas da simulação, como tempo médio de espera, chamadas atendidas, energia consumida e

pessoas transportadas Permite zerar estatísticas para reiniciar a simulação

Estruturas de Dados

- ArrayList temposEspera: Lista que armazena os tempos de espera de cada pessoa

Métodos Principais

- registrarEspera(int tempoEspera): Registra o tempo de espera de uma pessoa
- registrarChamadaAtendida(): Incrementa o contador de chamadas atendidas
- registrarEnergia(double energia): Adiciona energia consumida (0.5 por parada, 1.0 por movimento)
- registrarPessoaTransportada(): Incrementa o contador de pessoas transportadas
- getTempoMedioEspera(): Calcula a média dos tempos de espera
- zerar(): Reseta todas as estatísticas

6. Fila

Funcionalidades

Implementa uma estrutura de dados de fila (FIFO) para gerenciar elementos genéricos Usada para armazenar pessoas aguardando no andar ou dentro do elevador Suporta operações de enfileirar, desenfileirar, verificar se está vazia e obter o tamanho

Estruturas de Dados

- No inicio, No fim: Nós que apontam para o início e o fim da fila
- int tamanho: Contador do número de elementos na fila
- Classe interna No: Representa um nó da fila, contendo um elemento e um ponteiro para o próximo nó

Métodos Principais

- enfileirar(Object elemento): Adiciona um elemento ao final da fila
- desenfileirar(): Remove e retorna o elemento do início da fila
- isVazia(): Verifica se a fila está vazia
- tamanho(): Retorna o número de elementos
- getPonteiroInicio(): Retorna um ponteiro para o início da fila, usado para iteração

7. FilaPrioridade

Funcionalidades

Estende a classe Fila para implementar uma fila de prioridade Pessoas prioritárias (ex.: cadeirantes, idosos) são inseridas no início da fila Mantém a ordem FIFO para pessoas não prioritárias

Estruturas de Dados

Herda as estruturas da classe Fila (No inicio, No fim, int tamanho)

Métodos Principais

- enfileirar(Object elemento): Insere pessoas prioritárias no início e não prioritárias no final da fila

8. GerenciadorSimulacao

Funcionalidades

Gera uma lista de pessoas com atributos aleatórios para a simulação (ID, andar de origem, destino, prioridade, minuto de chegada) Garante que o andar de origem e destino sejam diferentes

Estruturas de Dados

- List pessoas: Lista de objetos Pessoa gerada para a simulação

Métodos Principais

- gerarListaPessoas(int quantidadePessoas, int quantidadeAndares, Scanner scanner): Cria uma lista de pessoas com atributos aleatórios

9. InterfaceGrafica

Funcionalidades

Fornecer uma interface gráfica para configurar, executar e visualizar a simulação Permite configurar parâmetros como número de andares, elevadores, capacidade, velocidade, tempos de viagem, heurística e tipo de painel Exibe visualmente o prédio, elevadores, pessoas aguardando, painéis, estatísticas e logs Oferece controles para iniciar, pausar, continuar, reiniciar e voltar à configuração

Estruturas de Dados

- List pessoas: Lista de pessoas geradas para a simulação

Métodos Principais

- criarPainelConfiguracao(): Cria o painel de configuração com campos para parâmetros
- criarPainelControle(): Cria o painel de controle com botões e slider de velocidade
- renderizarPredio(Graphics2D g): Desenha o prédio, andares, elevadores e pessoas
- atualizar(): Atualiza a interface com informações de pessoas, estatísticas e logs
- iniciarSimulacao(): Inicializa a simulação com os parâmetros configurados
- reiniciarSimulacao(): Reinicia a simulação mantendo os parâmetros
- agendarAdicaoDePessoas(): Agenda a adição de pessoas com base no minuto de chegada

10. Lista

Funcionalidades

Implementa uma lista encadeada genérica para armazenar elementos como andares, elevadores, destinos ou logs Suporta inserção no início e no final, remoção, verificação de existência e limpeza da lista

Estruturas de Dados

- No inicio: Nó inicial da lista encadeada
- Classe interna No: Representa um nó da lista, contendo um elemento genérico e um ponteiro para o próximo nó

Métodos Principais

- inserirFim(T elemento), inserirInicio(T elemento): Insere elementos no final ou início da lista
- remover(T elemento): Remove um elemento específico

- contem(T elemento): Verifica se um elemento está na lista
- getInicio(): Retorna um ponteiro para o início da lista
- isVazia(), tamanho(): Verifica se está vazia e retorna o tamanho

11. LogElevador

Funcionalidades

Armazena informações sobre ações de um elevador em um momento específico (minuto simulado, andar atual, decisão tomada) Registra eventos como embarque, desembarque ou escolha de destino

Métodos Principais

- toString(): Formata o log para exibição
- Getters: Para minutoSimulado, andarAtual, decisao, idElevador

12. Main

Funcionalidades

Ponto de entrada do programa Inicializa a interface gráfica em uma thread do Swing

Métodos Principais

- main(String[] args): Inicia a interface gráfica

13. NoGenerico

Funcionalidades

Interface genérica que define métodos para nós de estruturas de dados como Fila e Lista Garante que os nós sejam serializáveis e suportem operações genéricas

Métodos Principais

- getElemento(): Retorna o elemento armazenado no nó
- getProximo(), setProximo(NoGenerico proximo): Gerencia o ponteiro para o próximo nó

14. PainelElevador

Funcionalidades

Representa o painel de chamada de um andar, com três tipos: UNICO_BOTAO, DOIS_BOTOES, PAINEL_NUMERICO Gerencia chamadas (subir, descer, geral ou andares específicos) Permite resetar o estado do painel

Estruturas de Dados

- Lista andaresDestino: Lista encadeada que armazena os andares selecionados no painel numérico

Métodos Principais

- pressionarChamadaGeral(), pressionarSubir(), pressionarDescer(), pressionarAndar(int andar): Acionam botões ou registram andares de destino
- resetar(): Reseta o estado do painel
- Getters: Para botaoSubirAtivado, botaoDescerAtivado, chamadaGeralAtivada, andaresDestino, tipoPainel

15. Pessoa

Funcionalidades

Representa uma pessoa na simulação, com informações sobre origem, destino, prioridade e estado Registra o minuto de chegada e o status de chegada ao destino

Métodos Principais

- entrarElevador(), sairElevador(): Atualiza o estado de dentro do elevador
- getPosicaoAtual(): Retorna a posição atual (andar de origem, destino ou -1 se dentro do elevador)
- Getters: Para id, andarOrigem, andarDestino, minutoChegada, prioritaria, chegouAoDestino

16. Ponteiro

Funcionalidades

Implementa um iterador para percorrer estruturas de dados baseadas em nós (Fila e Lista) Permite acessar o elemento atual e avançar para o próximo nó

Estruturas de Dados

- NoGenerico atual: Referência ao nó atual

Métodos Principais

- getElemento(): Retorna o elemento do nó atual
- getProximo(): Retorna um novo ponteiro para o próximo nó
- avancar(): Move o ponteiro para o próximo nó
- isValido(), temProximo(): Verifica a validade do ponteiro e a existência de um próximo nó

17. Predio

Funcionalidades

Representa o prédio, contendo andares e a central de controle dos elevadores Atualiza o estado do prédio a cada minuto simulado Permite resetar o estado de andares e elevadores Verifica se uma pessoa já está no sistema

Estruturas de Dados

- Lista andares: Lista encadeada de objetos Andar

Métodos Principais

- atualizar(int minutoSimulado): Atualiza a central de controle
- resetar(): Limpa filas de pessoas nos andares e elevadores e reseta painéis
- pessoaJaNoSistema(Pessoa pessoa): Verifica se uma pessoa já está no sistema
- getAndares(), getCentral(): Retorna a lista de andares e a central de controle

18. Simulador

Funcionalidades

Controla a execução da simulação, gerenciando tempo simulado, velocidade e interação com a interface gráfica Inicializa o prédio com andares e elevadores Gerencia o timer da simulação, atualizando o estado a cada minuto Verifica se todas as pessoas chegaram aos destinos para encerrar a simulação Suporta serialização para salvar e carregar o estado

Métodos Principais

- iniciar(), pausar(), continuar(), encerrar(): Controlam a execução da simulação
- atualizarInterface(): Atualiza a interface gráfica
- isHorarioPico(): Verifica se o minuto simulado está no horário de pico

- todasPessoasChegaram(): Verifica se todas as pessoas chegaram aos destinos
- gravar(String nomeArquivo), carregar(String nomeArquivo): Serializam e desserializam a simulação
- deveAdicionarPessoa(Pessoa pessoa): Verifica se uma pessoa deve ser adicionada com base no minuto de chegada

19. Teste

Funcionalidades

Executa simulações com diferentes combinações de heurísticas e tipos de painel Configura cenários com andares, elevadores e pessoas predefinidas Executa, pausa, retoma e exibe estatísticas da simulação Testa a serialização do simulador

Estruturas de Dados

- List pessoas: Lista de configurações de pessoas para a simulação

Métodos Principais

- main(String[] args): Executa testes para todas as combinações de heurísticas e painéis
- getAndarPorNumero(Predio predio, int numero): Obtém o objeto Andar correspondente a um número

20. TipoPainel

Funcionalidades

Enum que define os tipos de painel de chamada: UNICO_BOTAO, DOIS_BOTOES, PAINEL_NUMERICO Fornece um método para obter a constante pelo índice

Métodos Principais

- getByIndice(int indice): Retorna o tipo de painel com base no índice

Conclusão

Esta documentação cobre todas as classes do sistema de simulação de elevadores, detalhando suas funcionalidades, estruturas de dados e métodos principais O sistema é modular, com classes bem definidas para gerenciar andares, elevadores, pessoas, estatísticas e interface gráfica As estruturas de dados, como filas, listas e filas de prioridade, suportam as operações necessárias para a simulação eficiente A documentação segue o padrão IEEE, garantindo clareza e organização para facilitar a manutenção e o entendimento do sistema.