

## **Introdução**

O objetivo deste documento é descrever o raciocínio utilizado na construção do algoritmo para o controle de dois elevadores. O sistema desenvolvido tinha como objetivo melhorar a movimentação dos elevadores, diminuindo o tempo de espera dos moradores.

## **Objetivo**

Desenvolver um sistema eficiente que gerencie dois elevadores em um edifício de 10 andares, identificados como "Elevador A" e "Elevador B". O sistema deve atender às chamadas dos moradores, otimizando o tempo de espera e proporcionando uma movimentação eficiente dos elevadores.

## **Lógica do Algoritmo**

Para realizar a lógica do algoritmo foi utilizado as seguintes classes e funções:

Classe Elevador:

Armazena dados como andar atual, destino, e estado (parado, subindo, descendo).

Implementa um timer para simular o tempo de deslocamento entre andares.

Classe Chamada:

Representa uma chamada de elevador com informações de origem, destino e direção.

Classe Edifício:

É a classe principal do algoritmo, ela gerencia a interface gráfica e também as funções do elevador. Ela gerencia as chamadas pendentes, movimentação dos elevadores e exibição do estado atual.

Tenta fazer uso de uma lógica simples para atender chamadas pendentes, analisando a menor distância entre os elevadores e a direção do movimento deles.

Função chamarElevador:

Recebe os parâmetros do método construtor da Classe Chamada. Realiza a verificação da direção analisando a origem e o destino, e determina qual elevador atenderá a chamada em razão de alguns parâmetros colocados.

Função elevMenor:

Verifica a menor distância entre os elevadores e a origem da chamada não atendida.

Função movimentaElevador:

Gerencia o movimento dos elevadores entre os andares.

Define a direção atual do elevador e o embarque/desembarque conforme verificação dos andares.

Inclui um timer de 2 segundos para a troca de andar.

Função embarcar:

Cuida do embarque no elevador.

Limpa a chamada atendida.

Verifica se há alguém embarcado e a direção do elevador para respeitar o chamado.

Função desembarcando:

Realiza o desembarque.

Limpa a lista de pessoas embarcadas.

Função destino:

Gerencia o destino para o qual o elevador está se dirigindo.

Atualiza a direção quando ela é alterada no embarque.

### **Exibição do Estado Atual:**

Fornece uma interface gráfica para mostrar o estado atual de cada elevador, incluindo andar, direção e chamadas pendentes.

### **Simulação de Tempo:**

Simula o passar do tempo para passar a cada andar (não tendo sido implementado a pausa a cada parada), movimento dos elevadores e atualização do estado.

### **Estruturas de Dados:**

Utiliza classes para representar Elevador, Chamada e Edifício.

Listas são empregadas para armazenar chamadas pendentes e estado dos elevadores.

### **Tomada de Decisões:**

Prioriza atender chamadas pendentes em uma direção eficiente, minimizando o tempo total de deslocamento.

Implementa um tempo mínimo de espera após uma parada para otimizar a movimentação.

### **Conclusão:**

O algoritmo implementado demonstra uma abordagem para otimizar o controle de elevadores em um edifício residencial. A lógica aplicada tenta minimizar o tempo de espera dos moradores, proporcionando uma solução adaptável a diferentes cenários. Existem melhorias que poderiam ser implantados, como a pausa a cada parada, estudo e implementação de um algoritmo de decisão mais eficiente, para melhorar a eficiência dos elevadores