

---

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**DEPARTAMENTO ACADÊMICO DE ELETRÔNICA (DAELN)**

Curso: Engenharia Eletrônica

Disciplina: Fundamentos de Programação 1 (IF61F)

Professores: Ednilson José da Silva, Danillo Leal Belmonte e João Luiz Rebelatto

**Projeto**

**Desenvolvimento de Simulador para Controle de Elevador**

---

**Objetivo**

Desenvolver em Linguagem C um simulador para controle de elevador.

**Sobre o Simulador**

Planejar o controle de grupo de elevadores é provavelmente o problema matemático mais importante a se resolver em sistemas de elevadores. É um desafio e tanto uma vez que possui, mesmo em cenários mais simples, um número excessivamente grande de soluções possíveis. O objetivo de um algoritmo controlador de elevadores é encontrar a solução que otimiza algum certo parâmetro de interesse. Tal otimização pode envolver um ou mais dos seguintes parâmetros: maximizar a capacidade manipulada, minimizar o tempo médio de espera ou o tempo médio de viagem, por exemplo.

O controle de grupo de elevadores se aplica a sistemas com dois ou mais elevadores operando no mesmo grupo, ou seja, elevadores que estão sujeitos a um conjunto comum de chamadas de passageiros necessitando de elevador. A função do controlador de grupo é alocar as chamadas assim que requisitadas para um (e somente um) dos elevadores do grupo, com o intuito de minimizar um dos mais dos parâmetros citados acima. Geralmente, a alocação de uma chamada para determinado elevador é definitiva, no sentido de que o sistema de controle não pode desalocá-la deste elevador e alocá-la em seguida para um elevador diferente caso um cenário mais interessante se torne disponível.

O objetivo do projeto é desenvolver um simulador de controle de elevadores, que receba como entrada a quantidade de andares do edifício e a quantidade de elevadores disponíveis, e a seguir apresente como seria a movimentação dos elevadores frente a requisições de chamadas feitas por passageiros. O algoritmo de controle do grupo de elevadores deve ser implementado visando otimizar alguma métrica que deve ser definida pelo programador. Ao fim da simulação, um relatório deve ser apresentado ao usuário.

## Requisitos Funcionais do Simulador

Os requisitos funcionais **mínimos** do simulador são:

### Menus

---

- Na tela inicial, o simulador deve apresentar um menu com as seguintes opções para o usuário:
  - 1) Instruções
  - 2) Definição dos parâmetros
  - 3) Iniciar simulação
  - 4) Estatísticas da simulação
  - 5) Sair
- 1) Caso escolha a opção 1) Instruções, deve-se apresentar as instruções de utilização do simulador ao usuário.
- 2) Ao escolher a opção 2) Definição dos parâmetros, a quantidade de andares do edifício, quantidade de elevadores disponíveis e a capacidade de cada elevador (número máximo de passageiros que comporta, igual para todos os elevadores) deverão ser solicitadas ao usuário.
  - Considere que existem ao menos 2 elevadores e 5 andares (que devem ser utilizados como valores padrão, caso o usuário parta direto para a opção seguinte).
- 3) Ao escolher a opção 3) Iniciar simulação, a simulação se inicia entrando em um laço de repetição em que chamadas pelos elevadores são recebidas na entrada, e o controlador de grupo de elevadores deve escalonar adequadamente as prioridades de cada chamada e por consequência movimentar os elevadores. Mais detalhes são fornecidos na próxima subseção.
- 4) Ao escolher a opção 4) Estatísticas da simulação, as estatísticas da última simulação são apresentadas ao usuário;
- 5) Ao escolher a opção 5) Sair, o simulador deverá ser encerrado.

### Durante a Simulação

---

- 1) Considere que todos os elevadores operam nos seguintes tempos:
  - Tempo que um elevador demora para subir/descer um andar: 5 segundos
  - Tempo de abertura/fechamento da porta do elevador: 1 segundo

Tempos gastos com a aceleração dos elevadores a partir do repouso e com o embarque e desembarque dos passageiros podem ser desprezados.
- 2) O simulador deve permitir que as requisições de chamadas (compostas por um **tempo**, **andar de origem** e **andar de destino**) sejam realizadas a partir do teclado, de um arquivo-texto previamente preenchido ou geradas aleatoriamente. Isso deve ser definido pelo programador antes de iniciar o simulador. Porém, o código deve contemplar as três possibilidades.
- 3) Para facilitar, considere que há apenas um botão de chamada de elevador por andar (ao invés de um botão “desce” e outro “sobe”).
- 4) O algoritmo de controle implementado é de **escolha livre**, mas deve levar em consideração as exigências a seguir na hora de realizar a alocação:

- As chamadas de dentro do elevador devem sempre possuir prioridade sobre as chamadas de fora do elevador;
  - O elevador não deve mudar de direção enquanto estiver carregado de passageiros;
  - O elevador deve sempre parar no andar de destino de um passageiro (não pode passar direto por esse andar);
  - Passageiros com o intuito de se deslocar em uma direção não devem entrar em um elevador que se desloca em direção oposta;
  - Um elevador não deve parar em um andar onde não há passageiros para embarcar ou desembarcar.
- 5) Deve ser desenhado na tela um esquema (usando biblioteca gráfica ou mesmo caracteres de texto) representando o prédio e os elevadores se movimentando dentro dele, bem como as situações dos botões de dentro dos elevadores e de cada andar do prédio.
- 6) A simulação se encerra quando um tempo limite definido pelo programador for atingido, ou quando o usuário optar por encerrá-la, apertando um botão pré-estabelecido, por exemplo.
- 7) Ao final da simulação, deve-se arquivar dois grupos de informações:
- O log da simulação, que registra uma nova mensagem quando da ocorrência de cada novo evento, por exemplo:
 

```
t = 0:  p1 chamou elevador no andar 0 atendido pelo elevador e1
t = 1:  porta de e1 fechou, subindo (p1 -> 23)
t = 10: p2 chamou no andar 10 atendido por e1
t = 51:  e1 parou no andar 10
t = 52:  porta de e1 abriu
t = 53:  porta de e1 fechou, subindo (p1 -> 23, p2 -> 23)
t = 118: e1 parou no andar 23
t = 119: porta de e1 abriu
```
  - As estatísticas da simulação, contendo, dentre outras possibilidades: *i)* carga média dos elevadores (número médio de usuários por elevador por tempo); *ii)* tempo médio de espera; *iii)* tempo máximo de espera; *iv)* tempo médio de viagem.

## Requisitos do Software

- 1) Os logs e estatísticas devem ser armazenadas em um arquivo texto, passível de alteração.
- 2) Deve ser utilizada alocação dinâmica de memória.
- 3) Utilize estruturas (struct) para representar os objetos do simulador (elevador, passageiro e prédio, por exemplo).
- 4) Utilize funções sempre que possível e conveniente (por exemplo, para representar os menus, imprimir a movimentação dos elevadores na tela, detectar chamada dos elevadores, movimentar os elevadores, etc).
- 5) Separe o código em arquivos .h e .c.

Tabela 1: **Composição da Nota**

|              | Item                                       | Explicação  | Peso (%)   |
|--------------|--|---|------------|
| 1            | Funcionamento                              | Funcionamento do simulador, de acordo com os “Requisitos Funcionais do Simulador” exigidos.                               | 30         |
| 2            | Estruturas                                 | Utilização correta de estruturas ( <code>struct</code> ).   | 10         |
| 3            | Funções                                    | Utilização correta de funções, com passagem de parâmetros por valor e por referência.                                     | 20         |
| 4            | Alocação dinâmica memória                  | Correta alocação dinâmica de memória.   | 10         |
| 5            | Arquivo-texto                              | Utilização/manipulação adequada de arquivo-texto para armazenamento das estatísticas.                                     | 10         |
| 6            | Arquivos <code>.h</code> e <code>.c</code> | Separação do código em arquivo(s) de cabeçalho ( <code>.h</code> ) e seu(s) arquivo(s) <code>.c</code> correspondente(s). | 10         |
| 7            | Etapas                                     | Cumprir o proposto para cada etapa (ver Moodle), dentro do prazo estabelecido.  | 10         |
| <b>TOTAL</b> |  |   | <b>100</b> |

### Composição da Nota

A composição da nota está apresentada na Tabela 1.

- **(Observação 1)** As notas dos itens 2-6 estão condicionadas ao funcionamento do simulador!
- **(Observação 2)** Não entregar o relatório dentro do prazo ou no formato estabelecido: **decrécimo de 1,0 (um ponto) na nota.**

### Dicas e Observações

- Novas funcionalidades podem ser agregadas ao simulador, não precisando o mesmo estar restrito ao itens apresentados anteriormente;
- Apesar de ser uma atividade em dupla, **as notas serão individuais**, podendo variar entre integrantes da mesma equipe, de acordo com o empenho demonstrado em sala bem como da demonstração de conhecimento durante a apresentação do projeto;
- Data limite para entrega do relatório (através do Moodle)<sup>1</sup>: **28/06/2017.**
- Data de apresentação do projeto: **29/06/2017.**

<sup>1</sup>Todos os integrantes da equipe devem fazer o *upload* do relatório no Moodle!