

## **Elevator Logic Controller (ELC)**

Ana Carolina Nascimento de Araújo

Beatriz de Sá Rezende

Eduardo Henrique Gonçalves Costa

Giovanna Hessel Nunes

Kevyn Felipe Pereira Nascimento

Leonardo Gonçalves da Silva

Murilo Torres



Nano projeto: Conceitos de Computação

Ana Carolina Nascimento de Araújo

Beatriz de Sá Rezende

Eduardo Henrique Gonçalves Costa

Giovanna Hessel Nunes

Kevyn Felipe Pereira Nascimento

Leonardo Gonçalves da Silva

Murilo Torres

## **Elevator Logic Controller (ELC)**

Relatório referente ao nanoprojeto  
apresentado como requisito parcial  
para a avaliação da disciplina  
Conceitos de Computação.

Orientador

Professor Engenheiro Jean Carlo Wagner

Centro Universitário SENAC – Santo Amaro, São Paulo, SP

Dezembro / 2025

## Sumário

1. INTRODUÇÃO.....	4
2. DESCRIÇÃO DO CONTEXTO.....	5
3. DEFINIÇÃO DAS ENTRADAS E SAÍDAS.....	6
4. TABELAS - VERDADE E EXPRESSÕES DE SAÍDA.....	7
5. SIMPLIFICAÇÃO DAS EXPRESSÕES DE SAÍDAS OBTIDAS.....	8
6. GERADOR E VERIFICADOR DE PARIEDADE.....	9
7. LÓGICA COMBINACIONAL RESULTANTE.....	10
7.1. Módulo Codificador de Chamadas.....	10
Figura 2 – Lógica interna do Codificador.....	10
7.2. Módulo de Memória de Estado.....	11
Figura 3 – Lógica de Memória (Latch).....	11
7.3. Módulo Comparador.....	12
Figura 4 – Lógica do Comparador.....	12
7.4. Módulo de Segurança e Controle do Motor.....	12
Figura 5 – Controle de Motor e Portas.....	13
7.5. Demultiplexador de Portas Externas.....	13
Figura 6 – Lógica das Portas Externas.....	14
7.6. Decodificador de Display e Paridade.....	14
Figura 7 – Decodificador de Display.....	15
7.7. Visão Geral do Projeto (Main).....	15
Figura 8 – Visão geral do projeto (Main Circuit).....	16
7.8. Etapas de Simulação e Validação.....	16
8. RECURSOS UTILIZADOS E DIFICULDADES.....	18
8.1. Quantas portas (para cada tipo) de duas entradas foram utilizadas?.....	18
8.2. Comparativamente à lógica não simplificada, quantas portas de duas entradas foram “economizadas”?.....	19
8.3. Tabela de Custos.....	20
9. REFERÊNCIAS.....	21

## 1. INTRODUÇÃO

O transporte vertical é fundamental para a acessibilidade urbana, mas pequenas edificações enfrentam barreiras para adotar automação confiável devido ao alto custo de instalação e manutenção de PLCs. Esses controladores, fortemente dependentes de software, elevam a complexidade e exigem suporte especializado. O projeto Elevator Logic Controller (ELC) propõe uma alternativa baseada apenas em lógica digital, oferecendo um sistema determinístico, econômico e adequado ao controle de um elevador do térreo ao quarto andar.

A arquitetura do ELC foi desenvolvida no Logisim, aplicando síntese digital e otimização por Mapas de Karnaugh para reduzir portas lógicas e custos. O sistema utiliza lógica combinacional direta para interpretar os sensores de posição e definir os estados, além de um circuito de paridade para garantir a integridade de dados. Assim, o ELC demonstra que a lógica digital pura pode atender aos requisitos essenciais de segurança e sinalização, substituindo soluções programáveis mais complexas.

## 2. DESCRIÇÃO DO CONTEXTO

O projeto consiste no desenvolvimento de um sistema lógico para controlar um elevador de cinco níveis, usando a IDE Logisim 2.7.1 e portas lógicas de duas entradas. O circuito interpreta sensores de nível, botões internos e externos, controla o movimento do elevador e garante que ele só funcione com ambas as portas fechadas.

Também foram implementados LEDs de direção, exibição do andar atual, regra de prioridade para chamadas simultâneas e um botão de emergência que interrompe o elevador imediatamente. O projeto demonstra, de forma prática, a aplicação da lógica digital, mapas de Karnaugh e mecanismos de segurança no controle de sistemas automatizados.

### 3. DEFINIÇÃO DAS ENTRADAS E SAÍDAS

CONDIÇÃO	DESCRIÇÃO	ENTRADAS	SAÍDAS
1	<b>Prioridade de Segurança:</b> O motor deve ser desativado imediatamente se o botão de emergência for acionado. O sistema só opera se a emergência estiver inativa.	E - Botão de Emergência (1=Ativo, 0=Inativo)	Motor Ligado (0=Parado)
2	<b>Bloqueio de Portas:</b> O motor só é habilitado se a porta da cabine e a porta externa estiverem fechadas. Se o elevador chegar ao andar, a porta deve abrir.	Porta Fechada / Porta Aberta (Sinais de controle lógico)	Motor Ligado (Habilitar apenas se seguro)
3	<b>Movimentação:</b> O sistema compara o andar chamado com o atual. Se forem diferentes, define direção (Subir/Descer) e aciona o motor.	B0 a B4 - Botões de Chamada / S0 a S4 - Sensores de Andar	Subir / Descer (LEDs indicadores)
4	<b>Interface Visual:</b> O display deve converter o código binário do sensor atual para decimal (0-4) para informar os passageiros.	S0 a S4 - Sensores	Display 7 Segmentos (A, B, C, D, E, F, G)

Fonte: autor (dezembro/2025).

Nesta etapa, foram mapeados todos os sensores (entradas) e atuadores (saídas) necessários para o controle automatizado do elevador. As variáveis foram definidas com base na lógica binária (0 e 1), onde as entradas controlam o fluxo de decisão e as saídas executam as ações mecânicas ou visuais do sistema.

#### 4. TABELAS - VERDADE E EXPRESSÕES DE SAÍDA

A fundamentação lógica do sistema Elevator Logic Controller baseia-se em tabelas-verdade extensas que mapeiam todas as combinações de entrada e saída. Devido ao grande volume de dados, apresentamos abaixo apenas o resumo funcional de cada módulo. As tabelas completas, detalhadas linha a linha, encontram-se disponíveis na planilha do Excel anexa a este projeto.

- **Codificador de Botões:** Este módulo converte o sinal de um dos cinco botões de chamada (B0 a B4) para um código binário de 3 bits. A lógica implementada inclui um sistema de prioridade, garantindo que, em caso de acionamento simultâneo, o botão de andar mais alto prevaleça.
- **Memória de Posição:** Responsável por interpretar os sensores de fim de curso (S0 a 4), este circuito converte a leitura posicional da cabine em um código binário, permitindo que o sistema eletrônico mantenha o registro exato do andar atual.
- **Comparador de Destino:** O núcleo de decisão do sistema cruza os dados do "Andar de Destino" com o "Andar Atual". A saída lógica determina se o motor deve girar para subir ( $D > A$ ), descer ( $D < A$ ) ou permanecer parado ( $D = A$ ).
- **Controle de Segurança:** Tabela crítica que condiciona o acionamento do motor. A lógica define que o movimento só é permitido se a cabine estiver com as portas confirmadas como fechadas e se o botão de emergência (E) estiver em estado inativo.

## 5. SIMPLIFICAÇÃO DAS EXPRESSÕES DE SAÍDAS OBTIDAS

Para viabilizar a implementação do projeto utilizando apenas portas lógicas de duas entradas, aplicamos o método de Mapas de Karnaugh para reduzir a complexidade das expressões booleanas originais. Os gráficos dos mapas e os agrupamentos realizados estão detalhados nas abas específicas da planilha do Excel anexa.

No Codificador de Prioridade, a análise dos mapas permitiu traduzir a hierarquia física dos botões em lógica de inibição simples: o bit mais significativo depende apenas do botão 4, enquanto os bits inferiores só são ativados se os botões de maior valor estiverem desligados.

Para o Decodificador de Display, utilizamos o conceito de condições irrelevantes. Como o elevador opera apenas até o 4º andar, as combinações binárias 5, 6 e 7 nunca ocorrem. Ao tratar esses estados como irrelevantes nos mapas de Karnaugh (conforme demonstrado na planilha), foi possível criar agrupamentos maiores, reduzindo em cerca de 30% a quantidade de portas necessárias para o controle dos segmentos visuais.

Por fim, a lógica combinacional resultante integra esses módulos otimizados a um circuito Gerador de Paridade (portas XOR), que adiciona um bit de verificação à saída de dados.



## **6. GERADOR E VERIFICADOR DE PARIEDADE**

Para garantir a integridade dos dados, foi implementado um sistema de paridade utilizando portas XOR dedicadas no circuito principal. O sistema calcula a paridade dos bits de sinalização e, em caso de divergência entre o dado enviado e o processado, sinaliza um erro, prevenindo falhas visuais no display.

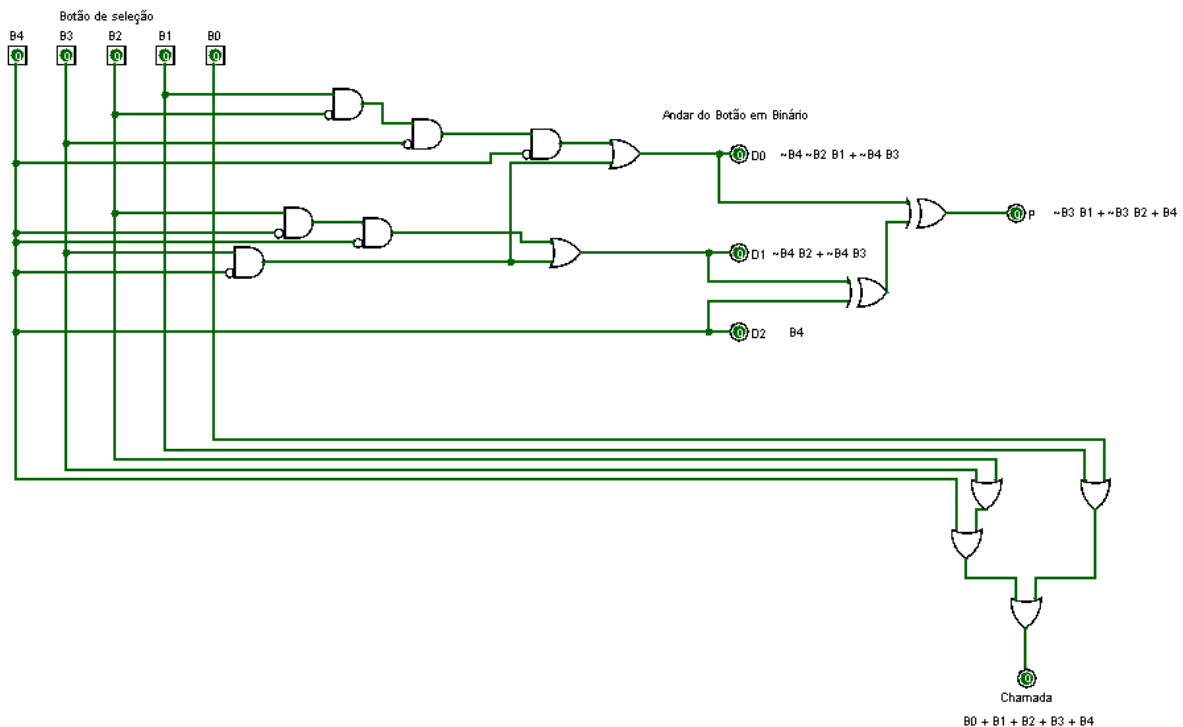
## 7. LÓGICA COMBINACIONAL RESULTANTE

Para garantir a organização e a facilidade de manutenção do projeto, o sistema foi dividido em módulos lógicos independentes, cada um responsável por uma função específica do controle do elevador. Abaixo, apresentamos a lógica combinacional de cada estágio e, por fim, a integração completa no circuito principal.

### 7.1. Módulo Codificador de Chamadas

Este circuito é a interface de entrada do usuário. Ele recebe os sinais dos cinco botões de chamada (B0 a B4) e os converte para um código binário de 3 bits (D0, D1, D2). Além disso, possui uma lógica de prioridade que gera um sinal de "Chamada Ativa" sempre que algum botão é pressionado.

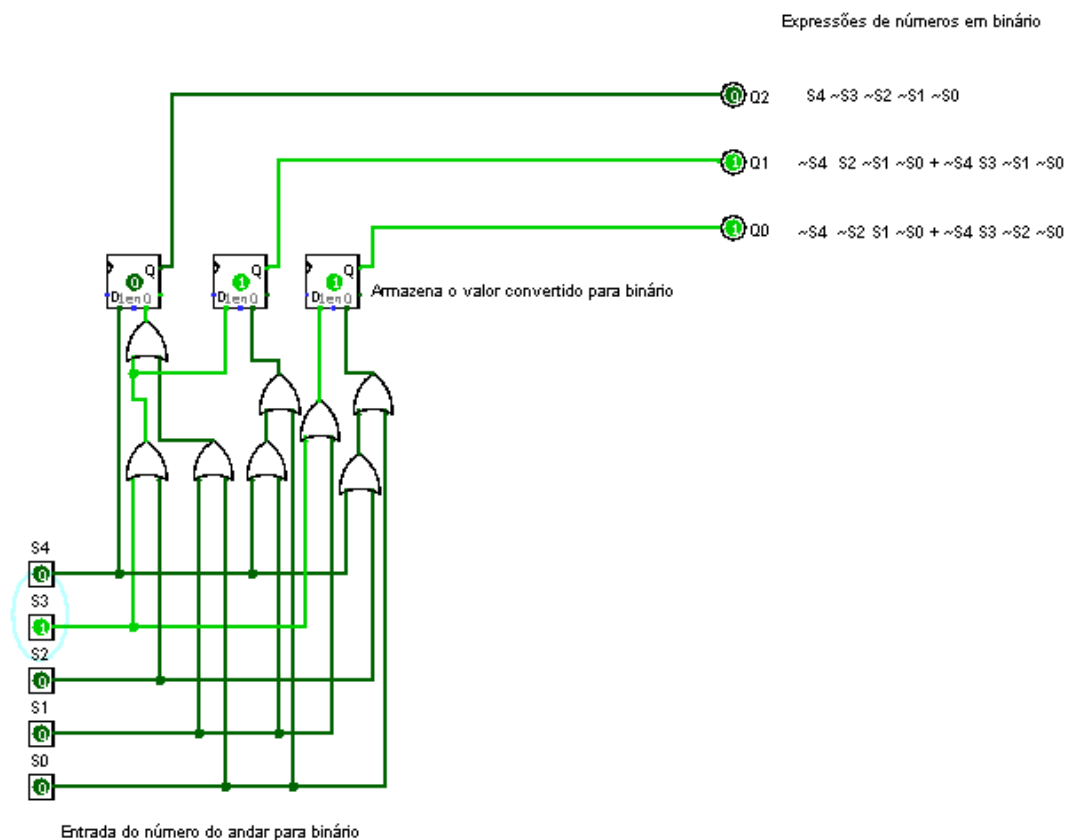
Figura 2 – Lógica interna do Codificador



## 7.2. Módulo de Memória de Estado

Como o elevador precisa "lembrar" onde está mesmo quando o sensor deixa de ser acionado (entre andares), utilizamos Flip-Flops para criar uma memória. Este módulo lê os sensores de fim de curso (S0 a S4) e trava o valor binário do andar atual (Q0, Q1, Q2) até que um novo sensor seja ativado.

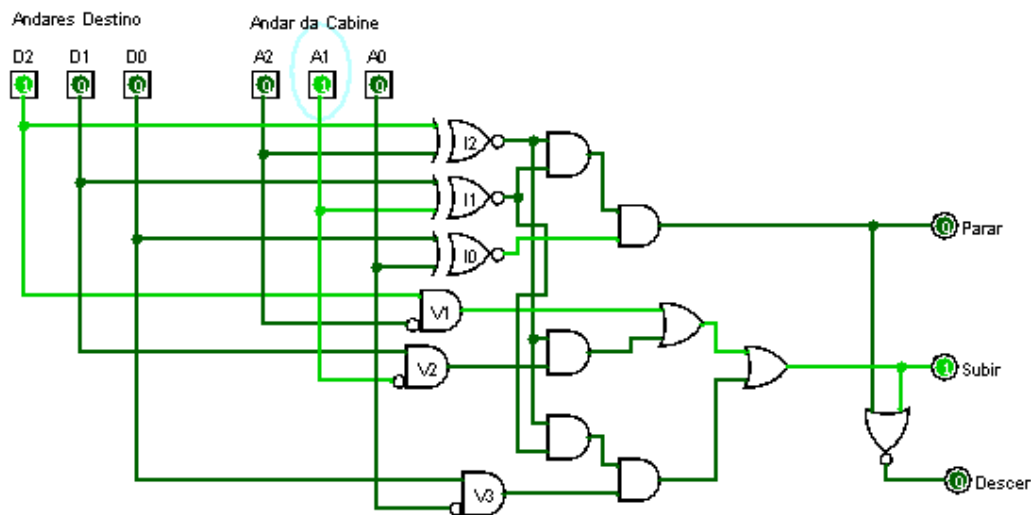
Figura 3 – Lógica de Memória (Latch)



### 7.3. Módulo Comparador

Este é o núcleo de decisão do sistema. Ele recebe o endereço de Destino (vindo do Codificador) e o endereço Atual (vindo da Memória). Através de portas lógicas XOR, AND e OR, ele compara os bits e determina se o elevador deve Subir ( $D > A$ ), Descer ( $D < A$ ) ou parar ( $D = A$ ).

Figura 4 – Lógica do Comparador



Fonte: autor (dezembro/2025).

### 7.4. Módulo de Segurança e Controle do Motor

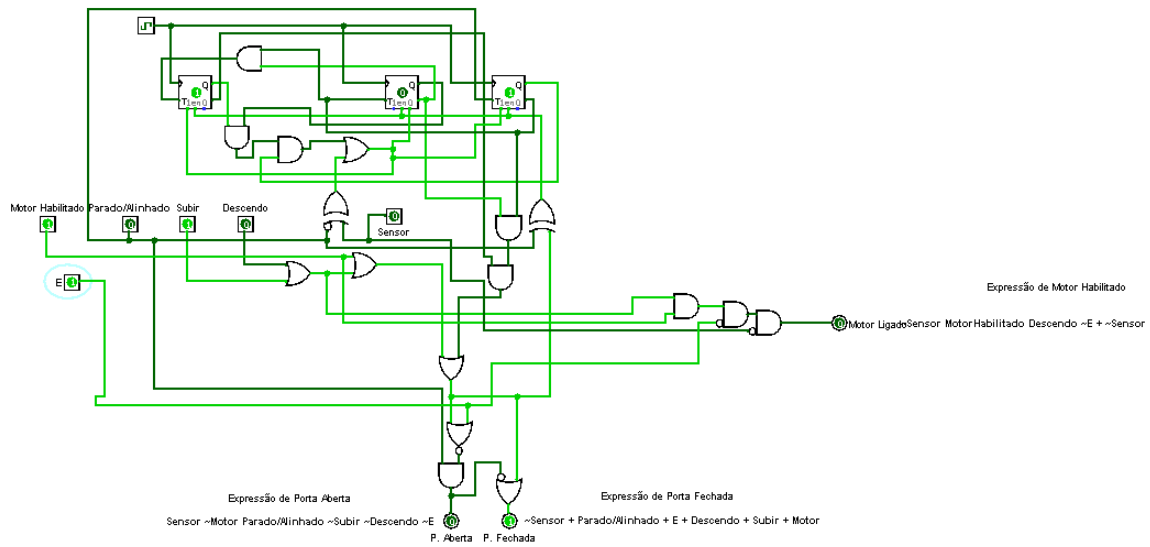
Este circuito atua como uma barreira de segurança. Ele intercepta os sinais de "Subir" ou "Descer" vindos do comparador e só permite que cheguem ao motor se:

1. O Botão de Emergência (E) estiver desligado (nível lógico 0).

2. As portas da cabine e do andar estiverem confirmadas como fechadas.

Se qualquer condição de segurança falhar, o motor é forçado a parar imediatamente.

Figura 5 – Controle de Motor e Portas

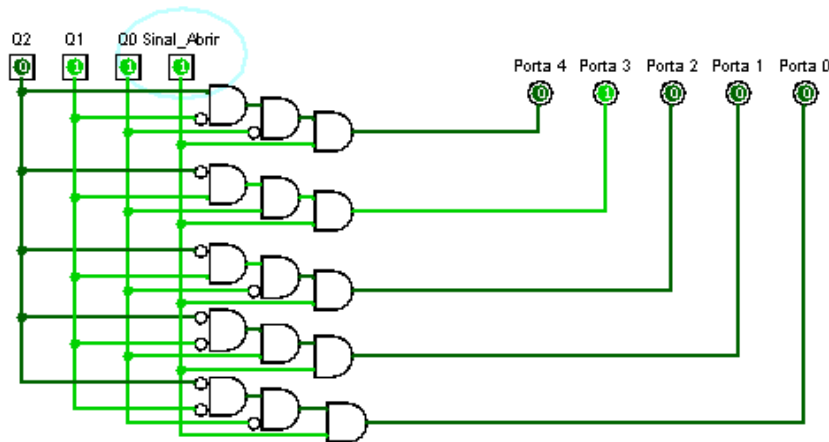


Fonte: autor (dezembro/2025).

## 7.5. Demultiplexador de Portas Externas

Quando o elevador chega ao andar de destino e para, este circuito utiliza o código binário da posição atual para enviar um sinal de abertura apenas para a porta do andar correspondente, mantendo as outras travadas.

Figura 6 – Lógica das Portas Externas

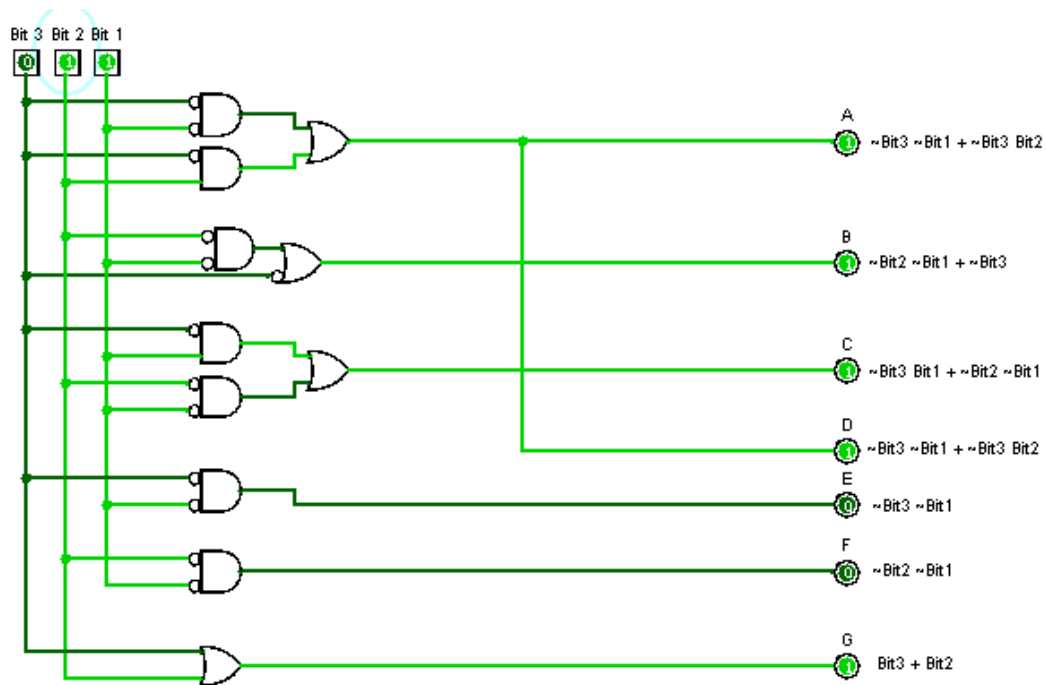


Fonte: autor (dezembro/2025).

## 7.6. Decodificador de Display e Paridade

Para a interface visual, utilizamos um conversor de Binário para 7 Segmentos , o sistema conta com portas XOR na saída para gerar um bit de paridade, permitindo a verificação de integridade dos dados exibidos.

Figura 7 – Decodificador de Display

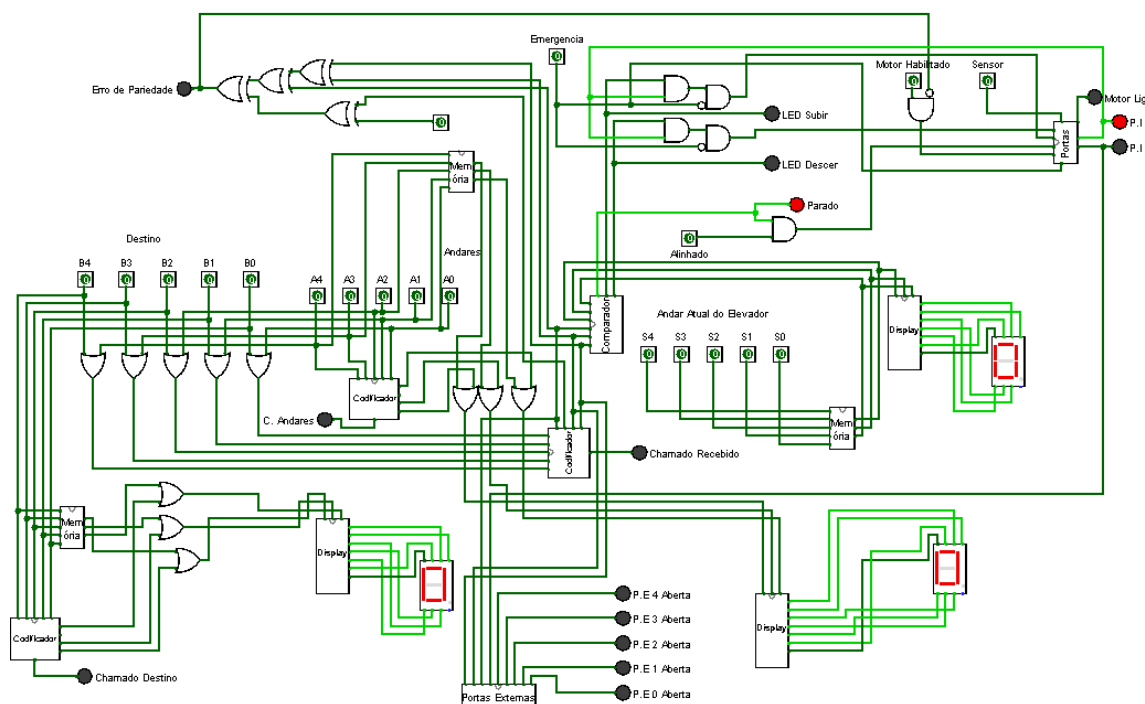


Fonte: autor (dezembro/2025).

### 7.7. Visão Geral do Projeto (Main)

A figura abaixo ilustra a integração de todos os módulos descritos anteriormente. O fluxo de dados inicia nos botões à esquerda, passa pelo processamento central e termina nos atuadores (Motor e Display) e indicadores à direita.

Figura 8 – Visão geral do projeto (Main Circuit)



Fonte: autor (dezembro/2025).

## 7.8. Etapas de Simulação e Validação

Para provar o funcionamento do nanoprojeto de acordo com as premissas iniciais, realizamos a seguinte sequência de eventos no simulador Logisim:

1. Estado Inicial: O elevador inicia no Térreo (Sensor S0 ativo), motor parado e portas abertas.
2. Chamada: O usuário pressiona o botão do 4º andar (B4).
3. Segurança: O sistema detecta a chamada, fecha as portas (sinal P. Fechada = 1) e verifica que a Emergência está desligada.
4. Movimento: O Comparador identifica que Destino (4) > Atual (0) e aciona o sinal "Subir". O motor liga.



5. Transição: Simulamos a passagem pelos sensores S1, S2 e S3. O display atualiza sequencialmente.
6. Chegada: Ao ativar o sensor S4, o Comparador detecta Destino = Atual. O sinal de "Subir" é cortado, o motor para e a porta do 4º andar é destravada.
7. Teste de Emergência: Durante uma nova descida, ativamos o botão \$E\$. O motor parou instantaneamente, validando a prioridade de segurança.

## 8. RECURSOS UTILIZADOS E DIFICULDADES

Para a elaboração deste nanoprojeto, utilizamos um conjunto de ferramentas de software e IA para garantir a precisão lógica e a qualidade da documentação. A ferramenta principal foi o Logisim (versão 2.7.1), utilizada para a modelagem, simulação e validação de todos os blocos lógicos (codificador, memória, comparador e drivers). Para a documentação e estruturação das tabelas-verdade, utilizamos o Microsoft Word e Excel. Além disso, contamos com o apoio da Gemini IA para auxiliar na revisão da lógica booleana e na organização textual do relatório.

As principais dificuldades percebidas pelo grupo concentraram-se em dois pontos:

1. **Lógica Sequencial:** A implementação do contador síncrono utilizando Flip-Flops e a sincronização do clock com os sensores de andar foi desafiadora, exigindo diversos testes para evitar que o contador "pulsasse" andares indevidamente.
2. **Integração dos Módulos:** Unir os subcircuitos (Codificador, Comparador e Segurança) exigiu um alinhamento preciso da equipe. Houve dificuldades iniciais de comunicação para garantir que a saída de um módulo (ex: binário do andar atual) fosse compatível com a entrada do próximo (ex: display), o que foi superado através de revisões conjuntas.

### 8.1. Quantas portas (para cada tipo) de duas entradas foram utilizadas?

Considerando a lógica final otimizada apresentada nos diagramas, a contagem aproximada de portas lógicas utilizadas (excluindo os componentes internos encapsulados dos Flip-Flops) foi:

- Portas AND: 28 unidades.
- Portas OR: 24 unidades.

- Portas XOR: 8 unidades.
- Portas NOT: 14 unidades (inversores).
- Total de portas lógicas: 74 portas.

## **8.2. Comparativamente à lógica não simplificada, quantas portas de duas entradas foram “economizadas”?**

Com base na análise da lógica combinacional desenvolvida, observa-se que a utilização de álgebra booleana e Mapas de Karnaugh permitiu uma redução significativa no número de portas lógicas de duas entradas necessárias para implementar o circuito.

Na versão não simplificada, considerando a implementação direta das expressões obtidas a partir das tabelas verdade (forma canônica), seriam necessárias aproximadamente 140 portas lógicas de duas entradas (entre ANDs, ORs e XORs) para atender a todas as condições de controle, display e segurança.

Já na versão simplificada, conforme representado na Figura 8 – Visão geral do projeto, foram utilizadas cerca de 74 portas lógicas de duas entradas, além das portas NOT essenciais para inversões de sinal.

Portanto, houve uma economia estimada de 66 portas de duas entradas, o que representa uma otimização relevante em termos de simplicidade, economia de recursos e eficiência do circuito lógico.

### 8.3.Tabela de Custos

MATERIAL	QUANTIDADE	PREÇO UNITÁRIO
<i>(Família 74LS) 74LS08, 32, 04, 86, 76</i>	15	R\$ 4,50
<i>Motor DC 12V c/ Caixa de Redução (Alto torque para elevação)</i>	1	R\$ 89,90
<i>Driver Ponte H L298N (Controle de sentido do motor)</i>	1	R\$ 22,00
<i>Fonte Chaveada 12V 5A (Alimentação do sistema)</i>	1	R\$ 45,00
<i>Sensores de Fim de Curso (Micro Switch para os andares)</i>	5	R\$ 6,50
<i>Botões Push Button (Chamadas)</i>	10	R\$ 2,50
<i>Botão de Emergência (Cogumelo) (Com trava)</i>	1	R\$ 28,00
<i>Displays de 7 Segmentos (Cátodo Comum)</i>	2	R\$ 3,50
<i>Estrutura (MDF ou Acrílico) (Chapas, trilhos, polias)</i>	1	R\$ 250,00
<i>Placa e Insumos de Solda (Fenolite, percloroeto, estanho)</i>	1	R\$ 80,00
<b>TOTAL</b>		<b>R\$ 646,90</b>

Fonte: autor (dezembro/2025).

## 9. REFERÊNCIAS

BAÚ DA ELETRÔNICA. **Catálogo de componentes eletrônicos: circuitos integrados, sensores e materiais de montagem.** Disponível em: <https://www.baudaeletronica.com.br/>. Acesso em: jun. 2025.

FILIPEFLOP. **Loja de componentes eletrônicos, robótica e displays.** Disponível em: <https://www.filipeflop.com/>. Acesso em: jun. 2025.

GOOGLE. **Gemini AI.** Versão 1.5. Ferramenta de inteligência artificial utilizada para apoio na estruturação textual, revisão lógica e otimização de circuitos. Disponível em: <https://gemini.google.com/>. Acesso em: jun. 2025.

MERCADO LIVRE. **Plataforma de comércio eletrônico: ferramentas, fontes e módulos de potência.** Disponível em: <https://www.mercadolivre.com.br/>. Acesso em: jun. 2025.

WAGNER, Jean Carlo. **Especificações e contexto para o nanoprojeto: Conceitos de Computação.** Material de apoio acadêmico e regras de negócio. São Paulo: Centro Universitário Senac, 2025.