

**EDUARDO BONI
HUGO YOSHINAGA**

Nota final

8.9
(correto e um)

22/01/04

**ESTUDO DA INTEGRAÇÃO ENTRE SISTEMA DE
ELEVADORES E SISTEMA DE CIRCUITO FECHADO
DE TELEVISÃO EM EDIFÍCIOS INTELIGENTES**

Trabalho de formatura apresentado à
Escola Politécnica da USP, para
obtenção do título de engenheiro.

**São Paulo
Dezembro 2003**

EDUARDO BONI 2937760
HUGO YOSHINAGA 2931152

***ESTUDO DA INTEGRAÇÃO ENTRE SISTEMA DE
ELEVADORES E SISTEMA DE CIRCUITO FECHADO
DE TELEVISÃO EM EDIFÍCIOS INTELIGENTES***

Trabalho de formatura apresentado à
Escola Politécnica da USP para
obtenção do grau de engenheiro.

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Paulo E. Miyagi
Co-orientadora:
Gladys D. B. Gustin
Coordenador:
Prof. Dr. Lucas Moscato

São Paulo
Dezembro 2003

1357016

TF-03
B 641e

Elevadores
Integração de Funções Elementares
Circuitos Integrados
Edifícios Inteligentes
Sistemas Híbridos

cad. 001357016

DEDALUS - Acervo - EPMN



31600010682

Aos nossos pais por nos terem dado
incentivo e suporte para vencermos
os desafios encontrados.

AGRADECIMENTOS

Nossos sinceros agradecimentos:

A Deus por nos iluminar os caminhos do conhecimento.

Ao nosso orientador Prof. Dr Paulo Eigi Miyagi, não só pela excelente orientação neste trabalho mas sim por todo o ensinamento transmitido durante a graduação.

À nossa supervisora Gladys Deifan Bastidas Gustin, pelas diretrizes e por estar sempre presente nos momentos necessários.

Ao amigo Bruno de Souza Carmo por ser um exemplo a seguir.

A todos que de alguma forma colaboraram para este trabalho.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO E JUSTIFICATIVA	1
1.2	OBJETIVOS	2
2	SISTEMAS DE ELEVADORES.....	3
2.1	INTRODUÇÃO	3
2.2	FUNCIONAMENTO	5
2.3	SISTEMA DE CONTROLE DOS ELEVADORES	6
3	LÓGICA FUZZY	9
3.1	INTRODUÇÃO À LÓGICA FUZZY	9
3.1.1	<i>Discussões Preliminares.....</i>	<i>9</i>
3.1.2	<i>Definição</i>	<i>9</i>
3.2	CONTROLE BASEADO EM LÓGICA FUZZY	10
3.3	APLICAÇÃO A ELEVADORES	13
4	ESTUDO E DESENVOLVIMENTO DE ESTRATÉGIAS DE CONTROLE PARA SISTEMAS DE ELEVADORES.....	15
4.1	ESPECIFICAÇÃO DO SISTEMA.....	15
4.2	MODELAGEM E DESENVOLVIMENTO DE ESTRATÉGIAS DE CONTROLE PARA O ESTUDO DE CASO	17
4.2.1	<i>Arquitetura de Controle Adotada.....</i>	<i>18</i>
4.2.2	<i>Estratégia de Controle Supervisório sem a Integração com o Sistema de Circuito Fechado de Televisão.....</i>	<i>19</i>
4.2.3	<i>Estratégia de controle com a integração com o sistema de circuito fechado de televisão 28</i>	
4.3	DESENVOLVIMENTO DO VISUALIZADOR	30
4.4	RESULTADOS OBTIDOS	31
5	CONCLUSÃO.....	45
6	REFERÊNCIAS BIBLIOGRÁFICAS	46

APÊNDICE I – Algoritmo de Simulação Considerando as Informações Adicionais do Circuito Fechado de Televisão.....	I.1
---	------------

APÊNDICE II – Visualizador de Estados do Edifício.....	II.1
---	-------------

LISTA DE FIGURAS

Figura 1 – Estrutura do sistemas de elevadores	6
Figura 3 – Diagrama conceitual básico do sistema de controle de SED	7
Figura 4 – Gráfico de Pertinência	11
Figura 5 - Esquema do edifício modelado	16
Figura 6 - Arquitetura de Controle de um sistema de controle de grupo implementado com lógica fuzzy	18
Figura 7 – Estrutura do processamento da lógica fuzzy	21
Figura 8 – Função de pertinência para o tempo de espera do usuário	22
Figura 9 – Função de pertinência para o tempo estimado de chegada do elevador	23
Figura 10 – Função de pertinência para a prioridade.....	24
Figura 11 – Regras de inferência utilizadas no sistema.....	25
Figura 12 – Gráfico de relacionamento do tempo de espera e do tempo estimado de chegada do elevador com a prioridade.	26
Figura 13 – Visualização do primeiro caso para teste	27
Figura 14 – Visualização do segundo caso para teste.....	28
Figura 16 - Interface do visualizador	32
Figura 17 - Situação inicial de simulação	33
Figura 18 - Chegada das 7 pessoas nos pavimentos 13 e 14	33
Figura 19 - Elevadores 1 e 2 rumo aos pavimentos 13 e 14 e chegada de mais 7 pessoas nos dois pavimentos	34
Figura 20 - Chegada dos elevadores 1 e 2 nos pavimentos 13 e 14, e entrada de 12 pessoas (capacidade máxima) em cada um	35
Figura 21 - Descida dos elevadores 1 e 2 com 12 pessoas cada, e subida dos elevadores 3 e 4 a fim de atender às 9 pessoas aguardando nos pavimentos 13 e 14	35
Figura 22 - Descarregamento dos elevadores 1 e 2 no pavimento térreo, e descida dos elevadores 3 e 4 com nove passageiros cada	36
Figura 23 - Chegada dos elevadores 3 e 4 no pavimento térreo para descarregamento	36
Figura 24 - Situação final da simulação sem considerar as informações das câmeras	37
Figura 25 - Situação inicial com câmera	38
Figura 26 - Chegada de 7 usuários aguardando em cada um dos pavimentos 13 e 14, e elevadores 1 e 2 subindo para atender as chamadas	38
Figura 27 - Chegada de mais 7 pessoas em cada um dos pavimentos 13 e 14.....	39
Figura 28 - Controlador envia os elevadores 3 e 4 para as chamadas dos pavimentos 13 e 14	40
Figura 29 - Elevadores 1 e 2 chegam aos pavimentos 14 e 13 respectivamente, enquanto que os elevadores 3 e 4 já estão subindo.....	40
Figura 30 - Elevadores 1 e 2 iniciam movimento de descida com 12 passageiros cada, enquanto elevadores 3 e 4 estão chegando aos andares 13 e 14.....	41
Figura 31 - Elevadores 3 e 4 chegam aos pavimentos 14 e 13 e atendem aos 9 usuários de cada pavimento, enquanto elevadores 1 e 2 estão descendo.	41

Figura 32 - Elevadores 1 e 2 atingem seu destino, enquanto elevadores 3 e 4 estão descendo.	42
Figura 33 - Elevadores 3 e 4 também chegam ao seu destino	42
Figura 34 - Situação final da simulação considerando as informações das câmeras .	43

LISTA DE TABELAS

Tabela 1 – Classificação dos dispositivos do sistema	7
Tabela 2 – Parâmetros adotados como entradas do software para o cálculo do número de elevadores.....	16
Tabela 3 - Especificações do modelo VF3.....	17
Tabela 4 - Filas de prioridades de chamadas	19
Tabela 5 - Regra para tráfego de descida.....	25
Tabela 6 - Métodos para as funções de inferência	26
Tabela 7 - Alocação das chamadas considerando o sistema de CFTV	30
Tabela 8 - Comparação dos tempos sem e com a integração com sistema de circuito fechado de televisão	44

RESUMO

Este trabalho visa otimizar o atendimento dos usuários de elevadores em edifícios inteligentes através da integração do sistema de elevadores com o sistema de circuito fechado de televisão. Assim, o controle dos elevadores deverá considerar informações adicionais como o número de passageiros aguardando em cada pavimento, bem como o número de usuários que estão dentro do elevador, de maneira a aprimorar a tomada de decisões, otimizar o atendimento, reduzir o tempo de espera e aumentar a eficiência do sistema.

Neste projeto, o foco está na proposta de estratégias de controle de elevadores e no desenvolvimento de um visualizador gráfico como ferramenta para analisar a dinâmica do sistema em função das estratégias propostas.

ABSTRACT

This work has the purpose of optimizing the service of the elevators in an intelligent building through the integration of the elevator system with the closed circuit of television. So, the control of the elevators should consider additional information such as number of passengers waiting in each floor, number of users that are inside the elevators, in a way that will improve the decision process, optimize the service, reduce the waiting time and increase the efficiency of the system.

In this project, the focus is the proposal of strategies to the elevator control and the development of a graphic viewer as a tool to analyze the dynamics of the system in function of the proposed strategies.

1 Introdução

1.1 *Motivação e justificativa*

O crescimento da demanda por sistemas prediais mais eficientes, que proporcionem um maior benefício aos usuários, tem contribuído para o desenvolvimento de novas técnicas para sua concepção e operação, evidenciando a necessidade de sua integração (Godoy, 1999).

No caso do sistema de elevadores, a importância da sua integração com outros sistemas prediais visa uma operação mais efetiva e otimizada destes, pois esta interação com outros sistemas do edifício permite que novas funcionalidades associadas à segurança, conforto dos ocupantes, confiabilidade e robustez do controle possam ser implementadas (Bastidas Gustin, 1999).

Deste modo o projeto e operação de elevadores necessitam de estudos e técnicas eficientes de modelagem e análise, que selecionem as melhores alternativas de modo a maximizar a produtividade dos ocupantes, otimizar a utilização de recursos e minimizar o custo.

Neste contexto, o presente projeto visa estudar e propor estratégias de controle para sistema de elevadores que considerem a integração deste com outros sistemas prediais, especificamente com o sistema de circuito fechado de televisão. Assim, considera-se que a informação do número de pessoas em espera em cada pavimento dado pelo sistema de circuito fechado de televisão deve melhorar a operação do sistema de elevadores.

Neste trabalho também é desenvolvido um visualizador gráfico que permite observar a operação do sistema de elevadores e analisar as estratégias de controle propostas.

1.2 Objetivos

Os objetivos principais deste trabalho são:

- Estudar e propor estratégias de controle para um sistema de elevadores que considerem a integração deste com o sistema de circuito fechado de televisão;
- Desenvolver um visualizador gráfico que permita observar e analisar a dinâmica do sistema em função das estratégias propostas.

2 Sistemas de Elevadores

A fim de realizar o estudo e desenvolvimento de estratégias para controle de elevadores, é necessário primeiramente identificar as principais características do sistema de elevadores, para depois focar nos quesitos onde se pode ter uma melhora do desempenho.

2.1 Introdução

Segundo Godoy (1999), desde que o homem ocupou mais de um pavimento numa edificação, existe a procura de alguma forma de movimento vertical. As formas mais antigas foram, é claro, as escadas, elevadores acionados por animais etc. Antigas ruínas romanas mostram sinais de poços onde algum tipo de elevador em plataforma móvel foi instalado.

No início do século XIX, elevadores acionados por máquinas a vapor fizeram sua aparição, principalmente para transporte vertical de materiais, mas, ocasionalmente também, para pessoas. Frequentemente os resultados eram desastrosos devido ao fato das cordas serem de fibras naturais e não haver nenhum meio de segurar a plataforma de transporte caso a corda arrebentasse.

Em 1857, o primeiro elevador de passageiros foi instalado na loja de E.V. Haughwout & Company, em Nova York, USA, operando entre cinco pavimentos a uma velocidade de 0,2m/s. O primeiro elevador elétrico surgiu em 1889, também em Nova York, no Edifício Demarest, e a primeira versão automática foi instalada em 1894, pela Otis Elevator Company.

A eletricidade permitiu o atendimento de padrões mínimos de operação, de modo que os parâmetros de tempo numa viagem puderam ser estabelecidos e especificados. A velocidade deixou de depender do volume da água ou da pressão de vapor, passando a ser estabelecida pelo projeto elétrico, conseguindo-se então, a suavidade de operação tão comum nos elevadores atuais. Atualmente emprega-se acionamento por motor CA ou CC, possibilitando a variação de velocidade e de torque pela variação da tensão aplicada a armadura. Tais sistemas ou suas variações

são usados nos elevadores de alta qualidade de hoje e são conhecidos como de tensão variável ou elevadores de multi-tensão.

Ainda segundo Godoy (1999), os sistemas de elevadores são instalados em edifícios para satisfazer as necessidades de transporte de seus ocupantes e visitantes. São necessários em virtude de conforto humano e conveniência ou por normas estatutárias. Em escritórios e outros prédios comerciais, elevadores são instalados a fim de aumentar a eficiência produtiva de seus ocupantes.

Um gerente de edifício, por exemplo, está interessado na operação dos elevadores e no atendimento ao tráfego pois o serviço de elevadores está em geral entre os primeiros lugares na lista de reclamações dos usuários de um prédio. Assim a melhora de sua operação e o aumento de sua confiabilidade são pontos fundamentais e que devem ser considerados.

O aumento do número de edifícios de altura média e elevada a partir da II Guerra Mundial tem sido um desafio aos sistemas de elevadores. As últimas três décadas têm visto a aceitação das cabines automáticas e a introdução de melhores sistemas de controle. Entretanto, não se nota uma evolução significativa nos métodos de análise de tráfego de usuários e de projeto.

A circulação no prédio, tanto horizontal quanto vertical, é um parâmetro funcional fundamental para qualquer edifício, e, portanto, se se deseja um edifício bem sucedido, é essencial que os responsáveis pela concepção do edifício interajam com um especialista nas soluções para sistemas de elevadores. Deste modo, poder-se-ia desenvolver um projeto considerando por exemplo um menor número de poços no edifício, ou um rearranjo dos pavimentos de serviço e dos pavimentos terminais principais. O efeito final esperado é um prédio apropriadamente estruturado para um “bom” acesso, com elevadores suficientes para servir à população proposta e atender “bem” à circulação prevista.

Os métodos convencionais de projetos de elevadores baseiam-se principalmente em técnicas analíticas aplicadas ao tráfego de pico de entrada. Os efeitos do período anterior aos momentos antes do pico de interesse, as peculiaridades do sistema de controle e o pouco tempo insuficiente para coletar dados estatísticos suficientes, contribuem para a imprecisão dos métodos de cálculo.

Para obter uma informação acurada é necessário rever a engenharia de projeto e recorrer às técnicas de simulação.

Outra preocupação com relação ao projeto é a existência de transporte/movimentação horizontal de alta densidade. Por exemplo, conforme ônibus ou trens chegam nas imediações do edifício trazendo usuários do sistema de elevadores, grupos de pessoas entram no edifício, a maioria delas considerando individualmente um certo tempo de viagem para chegar em um certo destino. Com meios de transporte que permitem descarregar centenas de pessoas em curto espaço de tempo, o sistema de transporte vertical do prédio fica submetido a severas demandas de pico de entrada. Outro exemplo seria o de um prédio suburbano de escritórios com um grande estacionamento. A demanda para os elevadores do prédio pode ser diretamente relacionada ao tempo requerido para as pessoas entrarem no estacionamento, estacionarem seus automóveis e caminharem em direção ao edifício.

2.2 Funcionamento

A operação do sistema de elevadores pode ser explicada utilizando como referência um usuário que vai de um pavimento A para um pavimento B.

No pavimento A, o passageiro registra sua chamada apertando o botão situado neste pavimento (especificando se será para subir ou descer). O sistema de controle de grupo registra a chamada sinalizando isso ao usuário através de um sinal luminoso e seleciona um elevador para servi-lo.

Quando da chegada de um elevador ao pavimento A, a sinalização se apaga e as portas do elevador são abertas. Ao entrar na cabine, o passageiro registra através de botões de comando da cabine seu destino (no caso, o pavimento B), sendo que o sinal correspondente é enviado ao sistema de controle da cabine. Este sistema registra o comando e sinaliza isso ao usuário através de um sinal luminoso. As portas então se fecham e o elevador se move para o pavimento B. Ao atingir o pavimento B o elevador pára, a sinalização se apaga, abrem-se as portas e o usuário desembarca. A figura 1 apresenta os elementos envolvidos nesta operação (Bastidas Gustin, 1999):

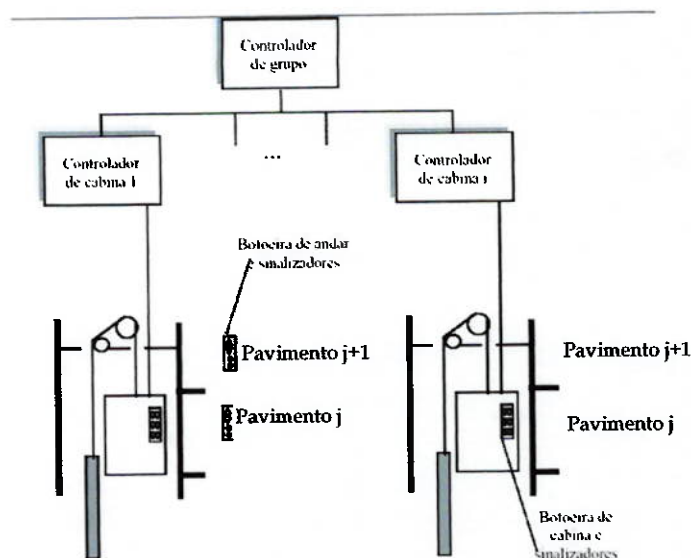


Figura 1 - Estrutura do sistemas de elevadores

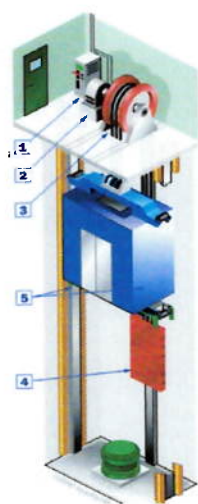


Figura 2 - Componentes do elevador

Os componentes de um elevador podem ser observados na figura 2. O número (1) indica o sistema de controle situado na casa de máquinas. O número (2) indica o motor elétrico. Os cabos que ligam a cabine ao contra-peso são envoltos em roldanas (3). O contra-peso é indicado pelo item (4). Já o elemento (5) representa os trilhos, que atuam de forma a evitar o balanço do contra-peso e da cabine, agindo também como item de segurança em caso de rompimento do cabo.

2.3 Sistema de Controle dos Elevadores

Considerando como um sistema a eventos discretos (SED), o sistema de elevadores pode ser decomposto para seu estudo conforme o esquema apresentado na figura 3 ([Miyagi, 1996]).

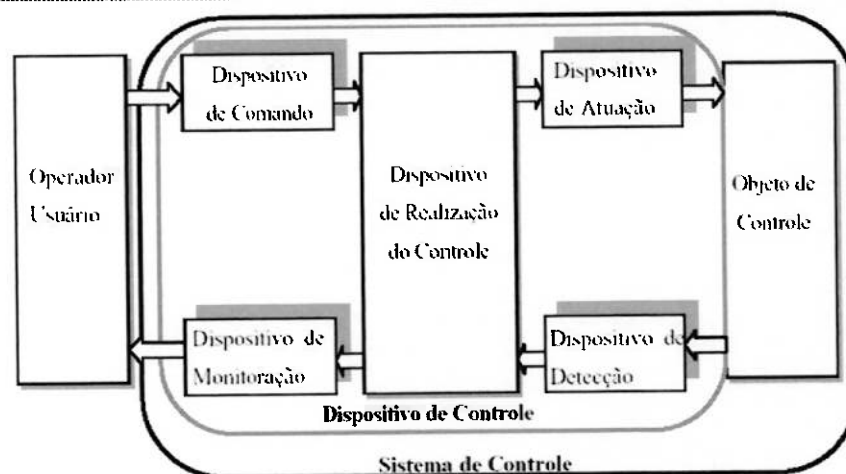


Figura 3 – Diagrama conceitual básico do sistema de controle de SED

Segundo este esquema os principais dispositivos utilizados no controle do sistema de elevadores podem ser classificados da seguinte forma (vide tabela 1):

Tabela 1 – Classificação dos dispositivos do sistema

Classificação	Dispositivos
Dispositivo de Comando	Botoeiras (cabines e pavimentos)
Dispositivo de Atuação	Motores das cabines, atuadores das portas
Dispositivo de Detecção	Fotocélulas nas portas, detetores de peso nas cabines, detetores de posição
Dispositivo de Realização	Controlador programável, computador
Dispositivo de Monitoração	Sinalizadores de direção e de posição, sinalizadores de registro de comando (da cabine e do pavimento), alarmes sonoros

Os objetivos do controle de sistemas de elevadores se baseiam nos requerimentos gerais para estes sistemas, os quais são (Thumm, 1995):

- Alta disponibilidade;
- Baixo consumo de energia;
- Menor espaço requerido dentro do edifício;

- Menores tempos de espera;
- Baixos custos de aquisição, instalação e manutenção
- Maior conforto;

A partir destes requerimentos, alguns objetivos do sistema de controle podem ser definidos (Sasaki et al., 1996)

- Maximizar a capacidade de trabalho para diferentes tipos de tráfego;
- Minimizar o tempo de espera médio para os passageiros;
- Reduzir o consumo de energia pela redução nos tempos de viagem e do número de partidas dos atuadores (motores elétricos por exemplo);
- Minimizar o tempo médio de serviço para todos os passageiros, isto é, o tempo total gasto pelos passageiros esperando por uma cabine e viajando até seu pavimento de destino;

Como alguns destes objetivos são conflitantes entre si, é impossível satisfazer todos os critérios simultaneamente. Assim, procura-se, para um controle eficiente do sistema, satisfazer cada critério em algum sentido ou atender um objetivo específico.

Tendo em vista esses critérios, nota-se o papel fundamental da estratégia de controle no rendimento da instalação. A finalidade do controle é estabelecer a prioridade no atendimento às chamadas, de acordo com as características do edifício. Para isso são concebidos e instalados controladores que atuam na partida, na parada, no sentido de movimento do carro, na seleção das chamadas e em outras funções correlatas (Industrias Villares, 1994).

Existem diversas abordagens existentes para implementação do controle de grupo de elevadores, entre as quais pode-se citar algoritmos genéticos, lógica fuzzy e redes neurais.

A lógica fuzzy foi a escolhida para utilização neste trabalho, pois é uma abordagem que permite a utilização de uma grande quantidade de variáveis que não são totalmente exatas (tais como o número de passageiros que será contado através das câmeras). O capítulo seguinte descreve com mais detalhe esta lógica.

3 Lógica Fuzzy

A seguir é apresentada uma breve introdução à lógica fuzzy que é a técnica considerada no presente trabalho para o desenvolvimento da estratégia de controle do sistema de elevadores.

3.1 Introdução à Lógica Fuzzy

3.1.1 Discussões Preliminares

Na vida prática pode-se considerar que 99% dos problemas são (Zadeh, 1965):

- Complexos demais para a aplicação de modelos convencionais ou “clássicos”;
- Definidos de maneira vaga, imprecisa, incerta;
- Definições “qualitativas” normalmente não impedem que se chegue a resultados “quantitativos”.

Segundo Zadeh (1965): Certas classes de objetos não admitem tratamento convencional. Existem problemas onde apesar de se chegar a números precisos a questão desta “precisão”:

1º - não é relevante para a solução do problema;

2º - pode vir a impedir que uma solução que satisfaça as condições especificadas seja encontrada.

Assim, foi desenvolvida a lógica fuzzy.

3.1.2 Definição

A lógica clássica é a ciência dos princípios formais e normativos do “raciocínio preciso” (Trillas et al, 1984). Já a lógica fuzzy pode ser definida como a ciência dos

princípios formais e normativos do raciocínio aproximado considerando-se o raciocínio preciso como caso limite (Trillas et al, 1984).

A lógica fuzzy fundamenta novas e poderosas soluções para uma enorme variedade de aplicações reais. Ela permite construir núcleos de decisão computacionais que obtêm conclusões concretas a partir de informações imprecisas, incertas ou vagas. (Menezes, 2003)

Esta lógica é capaz de extrair conclusões objetivas a partir de situações com uma certa ambigüidade, assemelhando-se ao comportamento humano na tomada de decisões. (Menezes, 2003)

3.2 Controle baseado em Lógica Fuzzy

Lógica Fuzzy é um modo de interfacear processos analógicos, que se baseiam em variáveis dentro de uma faixa contínua, com um computador digital, que as trata como valores numéricos bem definidos (valores discretos).

Por exemplo, considerando um sistema de frenagem dirigido por um microcontrolador; o microcontrolador toma as decisões baseado na temperatura do freio, velocidade, e outras variáveis do sistema. A variável "temperatura" neste sistema pode ser dividida dentro de uma faixa de estados: "frio", "fresco", "moderado", "morno", "quente", "muito quente". Contudo, a pertinência de um estado para o próximo é difícil de fixar e um limiar arbitrário deve ser fixado para dividir o "morno" do "quente", o que resultaria numa mudança descontínua quando o valor de entrada passasse pelo limiar e assim o microcontrolador não seria capaz de detectar isso. (Menezes, 2003)

A solução então é criar os estados Fuzzy que permitem associar a pertinência gradual de um estado para o outro. Assim pode-se definir a temperatura de entrada usando funções intermediárias.

Desta forma, o estado da variável de entrada não mais salta abruptamente de um estado para o próximo. Ao invés disso ele perde gradualmente o valor de pertinência de um estado enquanto vai ganhando maior valor de pertinência do próximo estado. Até que em algum momento, o "valor verdadeiro" da temperatura do

freio estará quase sempre em algum ponto entre as funções de pertinência: 0.6 nominal e 0.4 morno, ou 0.7 nominal e 0.3 fresco, e assim por diante (vide figura 4). (Menezes, 2003)

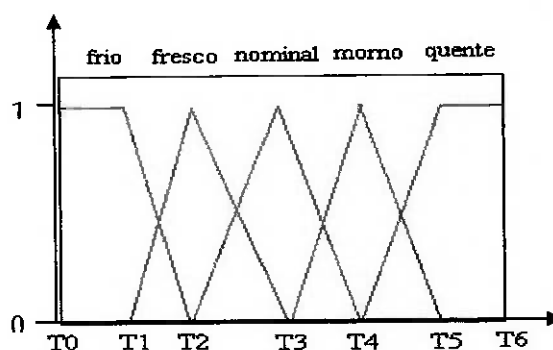


Figura 4 – Gráfico de Pertinência

As variáveis de entrada em um sistema de controle fuzzy são em geral mapeadas dentro de conjuntos de “funções de pertinência consecutivas”. O processo de conversão de um valor de entrada intermediário em um valor fuzzy é chamado de “fuzificação”. Note que um sistema de controle pode ter entradas chaveadas (on/off) junto com entradas analógicas, as entradas (on/off) terão sempre um valor verdadeiro igual a 0 ou 1 – entretanto tais entradas são de fato apenas um caso simplificado de uma variável fuzzy e o sistema pode tratá-las sem dificuldade.

Determinando o mapeamento das variáveis de entrada dentro das “funções de pertinência” e “valores verdadeiros”, o controlador então toma decisões para que as ações sejam efetuadas segundo as seguintes regras:

IF temperatura do freio = morna *AND* velocidade = não muito rápida
THEN pressão do freio = ligeiramente reduzida onde, neste caso, as duas variáveis de entrada são “temperatura do freio” e “velocidade”. A variável de saída, “pressão do freio”, é semelhantemente gerada a partir de um conjunto fuzzy que pode ter valores como “estático”, “ligeiramente reduzido”, “ligeiramente acrescido”, e assim por diante.

A decisão é baseada em um conjunto de regras: todas as regras aplicadas são invocadas, usando as “funções de pertinência” e “valores verdadeiros” obtidos das

entradas. Determina-se o resultado da regra, que é mapeada dentro da “função consecutiva” e do “valor verdadeiro” controlando a variável saída, e depois combina-se estes resultados para gerar uma resposta específica executando-se um procedimento conhecido como “defuzzificação”. A combinação de operações *fuzzy* e regras baseadas na “conclusão” compõem um “sistema *fuzzy expert*”. (Menezes, 2003)

Controladores *fuzzy* consistem assim de um estágio de entrada, um estágio de processamento, e um estágio de saída. O estágio de entrada mapeia dados de sensores ou outros tipos de entrada (tais como chaves, interruptores entre outros) de maneira apropriada através das “funções de pertinência” e “valores verdadeiros”; o estágio de processamento invoca a regra adequada e gera um resultado para cada uma destas regras e então combina os resultados obtidos; e finalmente o estágio de saída converte o resultado combinado do estágio anterior para um valor específico.

A forma mais comum de “funções de pertinência” é a triangular (como mostrado anteriormente na figura 4), embora curvas trapezoidais além de outras formas também serem usadas, a forma é geralmente menos importante do que o número de funções e o onde são colocadas. De 3 a 7 curvas são geralmente apropriadas para cobrir a faixa requerida de valores de entrada (“universo do discurso” na linguagem *fuzzy*).

O estágio de processamento é, como já discutido, baseado em uma coleção de regras lógicas na forma de declarações *IF-THEN*, onde o *IF* é chamado de “antecedente” e o *THEN* é chamado de “conseqüência”. Sistemas de controle *fuzzy* típicos têm dúzias de regras.

Considere uma regra para um termostato:

IF (temperatura é “quente”) *THEN* (aquecedor está “alto”)

Em alguns casos, as “funções de pertinência” podem ser mudadas pelos chamados “limitadores” que são equivalentes a adjetivos; limitadores comuns incluem “quase”, “próximo”, “perto de”, “aproximadamente”, “muito”, “ligeiramente”, “muito”, “extremamente” e “um pouco”. Estas operações podem ter definições precisas, apesar delas variarem consideravelmente.

Na prática, o conjunto de regras *fuzzy* tem vários antecedentes que são combinados usando operadores *fuzzy*, tais como *AND*, *OR* e *NOT* (apesar de novamente as definições variarem): *AND* (em uma definição popular) simplesmente usa o peso mínimo para todos os antecedentes, enquanto *OR* usa os valores máximos. *NOT* subtrai uma função de pertinência de 1 dando a função complementar.

Existem vários modos diferentes para definir o resultado de uma regra, mas um dos mais comuns e simples é o chamado método de conclusão "max-min", em que a saída da função de pertinência é dada pelo valor verdadeiro gerado pela premissa.

Regras podem ser resolvidas em hardware paralelo ou em software sequencial. O resultado de todas as regras é "defuzzificado" para um valor específico por diferentes métodos, cada qual com suas vantagens e desvantagens.

O método do centróide é muito popular. Neste, o "centro de massa" do resultado fornece o valor específico; outra aproximação é o método "*height*", que considera o maior valor. O método do centróide favorece a regra com a saída de maior área, enquanto o método "*height*" favorece a regra com o maior valor de saída.

O projeto de sistemas de controle *fuzzy* é baseado em métodos empíricos -- basicamente uma aproximação metódica para tentativa e erro. Existem poucas regras pré-definidas no presente momento uma vez que a tecnologia é ainda nova. O processo de projeto em geral segue os seguintes passos:

- 1º) Documenta-se as especificações operacionais do sistema e entradas e saídas;
- 2º) Documenta-se os conjuntos fuzzy para as entradas;
- 3º) Documenta-se o conjunto de regras;
- 4º) Determina-se o método de defuzzificação;
- 5º) Realiza-se testes para verificação do sistema, ajustando os parâmetros como requerido.

3.3 Aplicação a elevadores

A Lógica Fuzzy tem se mostrado adequada para tratar informações imprecisas e convertê-las para um formato numérico, de fácil manipulação pelos computadores.

A primeira aplicação de lógica fuzzy em sistemas de elevadores foi em um projeto da empresa Mitsubishi (Umeda et al, 1989), onde a regra apropriada de atendimento era selecionada após cada chamada de elevador.

A lógica fuzzy foi usada mais tarde para identificar padrões de tráfego baseados na intensidade e no fluxo de usuários para entrada, saída e movimentação entre os pavimentos do edifício (Siikonen, 1991).

Após estudo das teorias e ferramentas disponíveis para a abordagem do controle de elevadores, considerou-se a utilização, no presente trabalho, de lógica fuzzy como técnica para a modelagem e desenvolvimento das estratégias de controle. A escolha dessa técnica deve-se ao fato desta considerar incertezas associadas a um sistema e a processos que envolvem o homem, seus procedimentos e suas decisões. Além disso, foram identificados diversos trabalhos que utilizando esta técnica tem conseguido resultados bons, entre eles estão os de Gudwin (1998) sobre controle de grupo com adaptação linear utilizando lógica fuzzy e Siikonen (1997) sobre medição e previsão de tráfego de passageiros, com conseqüente ajuste da estratégia de controle utilizando lógica. Tem-se ainda equipamentos existentes no mercado como os controladores de sistema de elevadores de grupo “EJ-1000” da Toshiba e o “Alta” da Kone que já possuem lógica fuzzy.

4 Estudo e Desenvolvimento de Estratégias de Controle para Sistemas de Elevadores

4.1 Especificação do Sistema

A fim de se estudar as estratégias de controle do sistema de elevadores, é proposto um estudo de caso e são considerados critérios de desempenho a serem adotados a fim de facilitar a escolha e a comparação entre as soluções obtidas.

A primeira decisão a ser realizada refere-se ao tipo de edifício que será usado como o estudo de caso. Para o presente projeto escolheu-se um prédio de escritórios, uma vez que neste tipo de edifício ocorre um fluxo de pessoas elevado e um sistema de elevadores com um controle eficiente se faz necessário. Para este prédio deve-se definir o número de pavimentos e de elevadores a serem considerados.

Trabalhos realizados a respeito de estratégias de controle de elevadores indicam que desenvolver uma solução genérica independentemente do número de pavimentos ou uma solução para um número de pavimentos muito elevado é relativamente complexo, explicando o porquê de trabalhos como os de (Gudwin, 1998) realizarem simulações com um número fixo de andares e elevadores.

Desta forma, nesta fase o número de pavimentos e o número de elevadores será fixado a fim de simplificar o problema. Para a especificação do número de pavimentos do prédio que será objeto de estudo, utilizou-se como parâmetro o número médio de pavimentos de edifícios de escritórios na cidade de São Paulo, SP. Considerou-se assim um edifício com 17 pavimentos onde 2 estão no subsolo.

O segundo parâmetro a ser determinado para a especificação do sistema é o número de elevadores no edifício. Para isto, recorreu-se ao material da empresa Otis, que fornece o número de elevadores necessários dependendo do número de pavimentos e de pessoas por pavimento. Na tabela 2 tem-se os parâmetros que foram adotados como entrada no *software* da empresa Otis: (estes parâmetros foram

estabelecidos através de uma pequena amostragem a fim de determinar um padrão de edifício médio de escritórios encontrados em São Paulo).

Tabela 2 - Parâmetros adotados como entradas do software para o cálculo do número de elevadores

Número de pavimentos acima do térreo	14
Número de pessoas por pavimento	40
Número de subsolos	2
Número de vagas em cada pavimento do estacionamento	120

Utilizando este programa chega-se ao número de 4 elevadores.

A figura 5 ilustra um esquema realizado no software “*Simtower*” onde um edifício é modelado, com 15 pavimentos (os 2 subsolos não aparecem na figura) e 4 elevadores.



Figura 5 - Esquema do edifício modelado

Na tabela 3 são apresentados os dados obtidos do sistema VF3 da Otis que foi determinado, pelo software *on-line* da Otis, como o mais adequado para o modelo de edifício considerado como estudo de caso.

Tabela 3 - Especificações do modelo VF3

Capacidade da cabine (pessoas)	12
Capacidade da cabine (kg)	900
Velocidade de operação (m/s)	2,5

A fim de medir o desempenho global do sistema, utilizar-se-ão duas variáveis:

A primeira variável será o tempo necessário para que todos os usuários que desejam sair do edifício sejam transportados do seu piso para o térreo. Para tratar esta variável considera-se os usuários alocados em uma fila pré-definida que será dada como entrada do programa de controle de grupo de elevadores. A segunda variável é a somatória dos tempos que cada chamada levou para ser atendida por cada elevador. Isto é, se no andar tiver um número de pessoas maior que 12, ou seja, um só elevador não for capaz de atender esta chamada, este tempo continua a ser incrementado até que todas estas pessoas sejam alocadas em elevadores e a chamada possa ser dada como satisfeita.

4.2 Modelagem e desenvolvimento de estratégias de controle para o estudo de caso

A modelagem do sistema de elevadores será realizada tendo em conta as especificações definidas no item 4.1.

Em geral, o controle de sistemas complexos que envolvem múltiplas variáveis com alto grau de incerteza e que segue padrões e regras lingüísticas (por exemplo, tempo de espera “grande”, prioridade “alta”) pode ser tratado pela lógica fuzzy.

Assim, no presente trabalho escolheu-se pela lógica fuzzy para a modelagem das estratégias de controle do sistema e para sua implementação utilizou-se o *software* Matlab®. Uma grande vantagem é que este software permite o

desenvolvimento de um visualizador em tempo real, já integrado ao programa, otimizando a interface do controlador com o visualizador.

4.2.1 Arquitetura de Controle Adotada

A arquitetura de controle adotada é a de controle de grupo de elevadores, no qual há um controlador para cada cabine a fim de comandar operações de subida/descida e abertura/fechamento de portas. Além destes controladores, existe um controle supervisão de grupo, que comanda os controladores de cada cabine. A figura 6 apresenta a arquitetura de controle de um sistema de elevadores, baseado no trabalho de (Gudwin, 1998), cujo supervisão está implementado com lógica fuzzy.

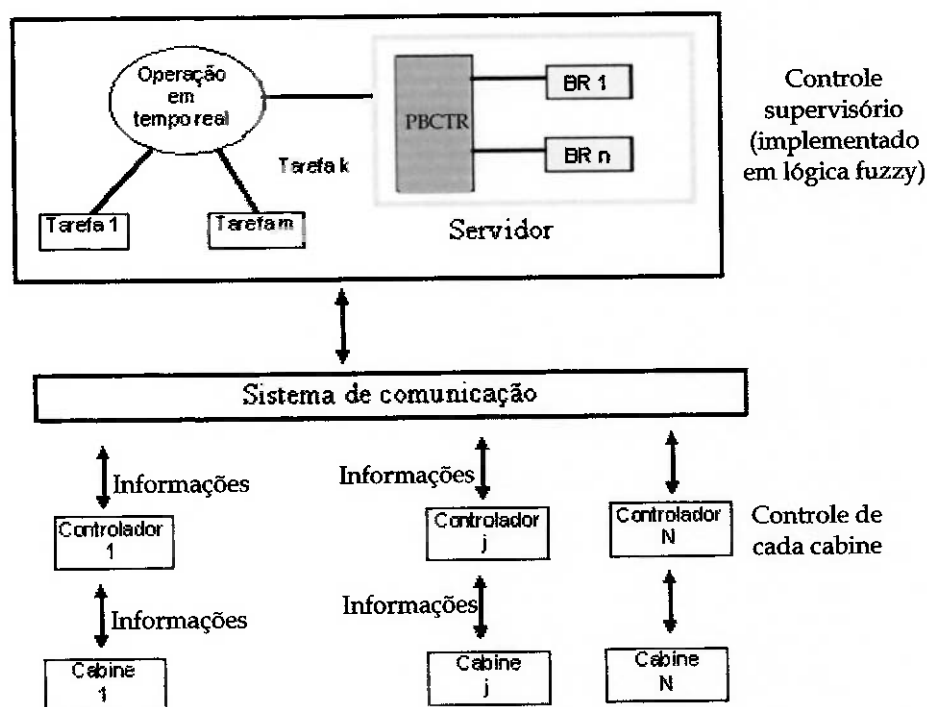


Figura 6 - Arquitetura de Controle de um sistema de controle de grupo implementado com lógica fuzzy

Na figura 6, o controle supervisão tem como dado de entrada o estado de cada elevador (posição, em movimento/parado, sentido: descida/subida, chamadas de cabine, entre outras informações.). Além disso, há uma base de regras fuzzy (BR1,

BR2,..., BRn) que são processadas pelo PBCTR (Processador de Base de Conhecimento em Tempo Real) e são utilizadas no processo de inferência a fim de determinar a estratégia mais adequada para o padrão de tráfego vigente. De acordo com a base de regra (BR) apropriada, será determinado o padrão de tráfego escolhido (tráfego normal, tráfego de subida, tráfego de descida). No estudo de caso será utilizado o padrão de tráfego de pico de descida por se tratar de um padrão de tráfego crítico.

4.2.2 Estratégia de Controle Supervisório sem a Integração com o Sistema de Circuito Fechado de Televisão

Para realizar o controle supervisório do sistema, utilizar-se-á como estratégia de alocação de chamadas de pavimento aos elevadores uma metodologia que considera prioridades (de 0 a 100) para cada chamada. Deste modo, para cada elevador é definida uma fila de acordo com a ordem de prioridades das chamadas. Para a determinação das prioridades é utilizada a lógica fuzzy considerando como entradas o tempo de espera do usuário (tempo que levou para atender a chamada) e o tempo estimado de chegada do elevador (calculado a partir de métodos probabilísticos).

Se duas ou mais cabines estiverem qualificadas para atender a mesma chamada, elas competem entre si baseadas no seu tempo estimado de chegada. Caso ainda haja empate, então é realizada uma escolha aleatória. A tabela 4 mostra um exemplo desta estratégia. As células em destaque indicam as chamadas que foram alocadas para cada cabine após os critérios de desempate terem sido aplicados.

Tabela 4 – Filas de prioridades de chamadas

Identificação do pavimento (identificação do sentido)				
Cabine 1	4 (descida)	7 (descida)	3 (descida)	7 (descida)
Cabine 2	13 (descida)	8 (descida)	14 (descida)	3 (descida)
Cabine 3	4 (descida)	10 (descida)	1 (descida)	11 (descida)
Cabine 4	7 (descida)	14 (descida)	8 (descida)	11 (descida)

Pode-se verificar no exemplo acima que tanto a cabine 1 quanto a cabine 3 estavam qualificadas para atender à chamada de descida do pavimento 4, porém apenas a cabine número 1 foi alocada após a utilização do critério de desempate.

A estrutura considerada para o controlador de grupo em lógica fuzzy é apresentada na figura 7.

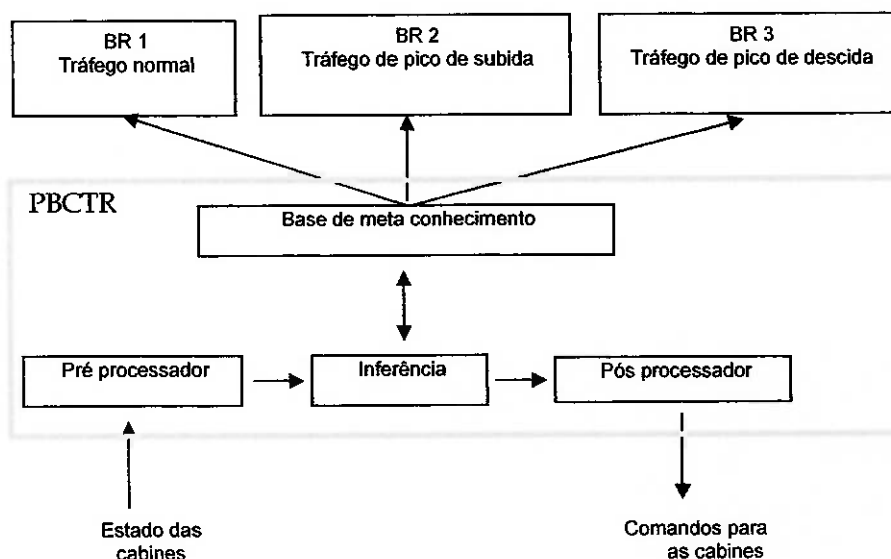


Figura 7 – Estrutura do processamento da lógica fuzzy

Os estados das cabines são as entradas para o controle supervisorio realizar a fuzzificação destas variáveis no pré-processamento. A partir dos estados que estão representados em forma lingüística, é realizada a transformação para a linguagem computacional utilizando funções de pertinência para cada uma das entradas. É realizada então a inferência dessas entradas utilizando a base de meta conhecimento que contém as regras de inferência (representada por regras do tipo *if-then*). Após esta etapa, é realizada a desfuzzificação, transformando a saída computacional em comandos para as cabines através do pós-processador que contém a função de pertinência da variável de saída.

A - Funções de Pertinência

Neste trabalho, optou-se por utilizar duas entradas para o controlador fuzzy:

- Tempo de espera;
- Tempo estimado de chegada do elevador ao pavimento.

Para cada uma das entradas, definiu-se uma função de pertinência, as quais foram elaboradas usando o *software* Matlab® (vide figuras 8 e 9).

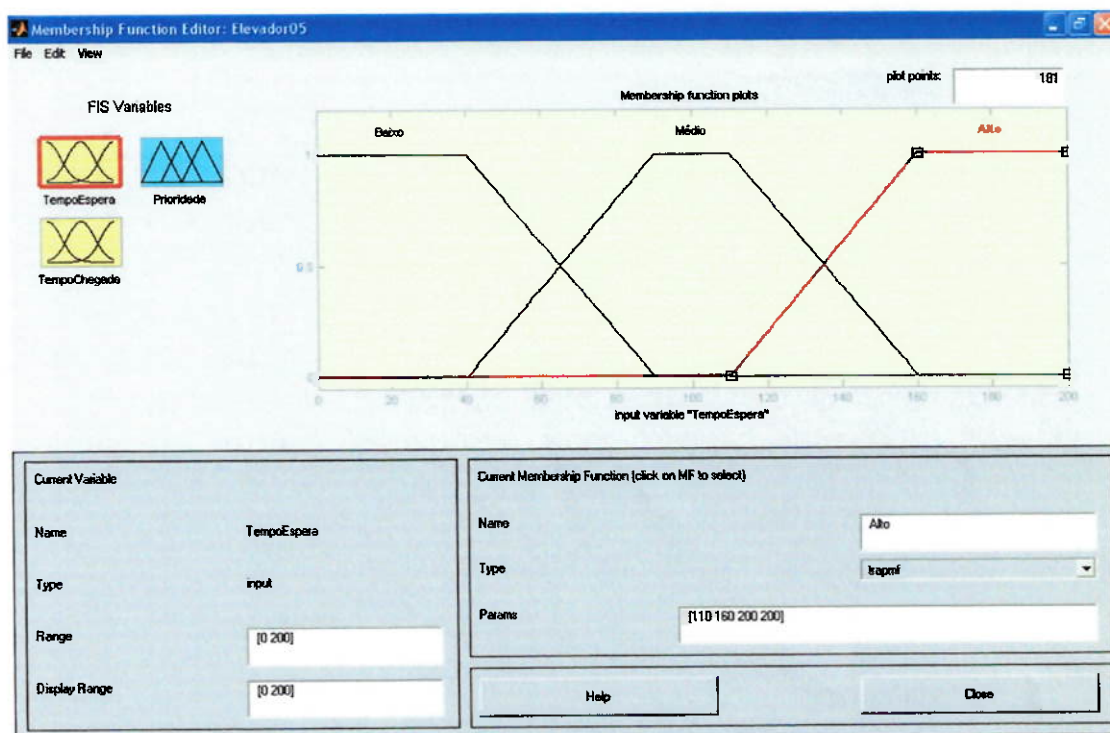


Figura 8 – Função de pertinência para o tempo de espera do usuário

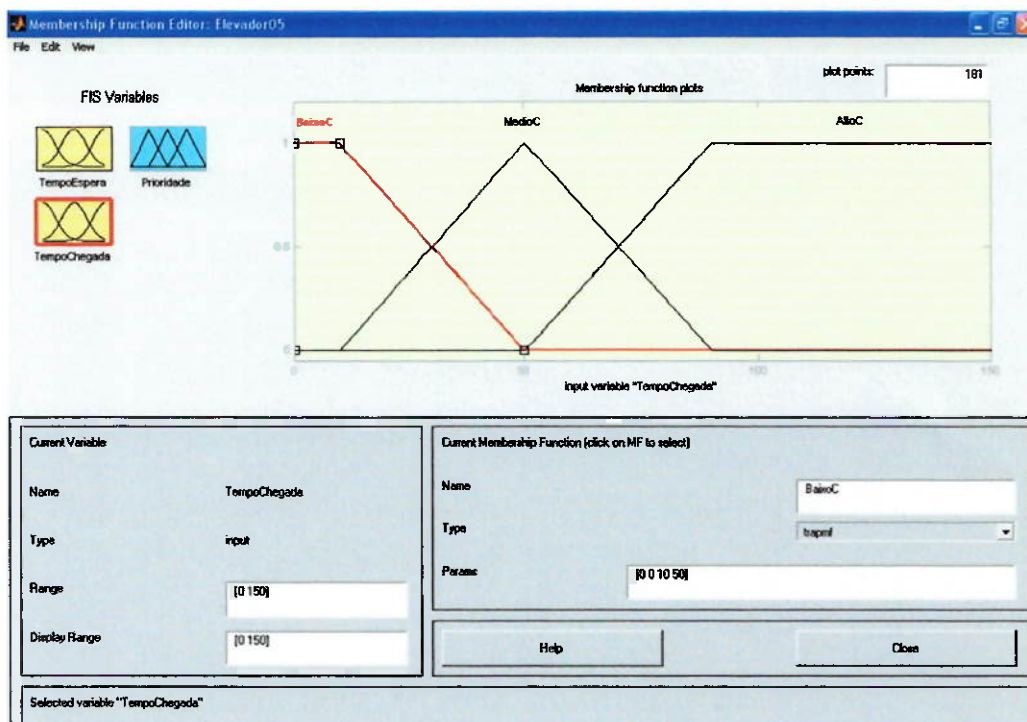


Figura 9 – Função de pertinência para o tempo estimado de chegada do elevador

Pode-se observar que a função de pertinência para o tempo de espera do usuário é uma função trapezoidal linear que possui três termos (BaixoE, MédioE e AltoE), correspondentes a um tempo de espera considerado baixo, médio e alto. Empiricamente, tem-se que para a maioria dos casos uma função linear é suficiente. (Menezes, 2003)

Para o tempo estimado de chegada do elevador, tem-se também três termos (BaixoC, MédioC e AltoC), correspondentes aos tempos estimados de chegada do elevador ao pavimento considerados baixo, médio e alto.

Foi elaborada também uma função de pertinência para a variável de saída do controle supervisorio, que no caso é a prioridade (0 a 100). Esta função de pertinência pode ser visualizada na figura 10.

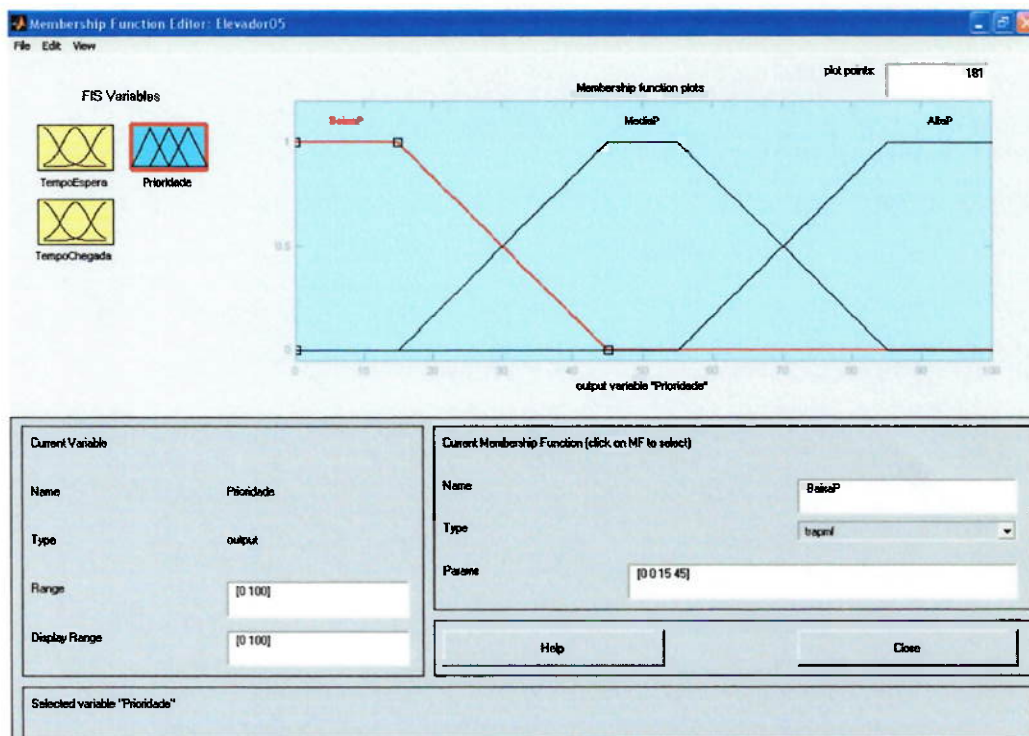


Figura 10 – Função de pertinência para a prioridade

Esta função de pertinência possui três termos (BaixaP, MédiaP, AltaP) correspondentes a uma baixa, média ou alta prioridade, dependendo das entradas do controle supervisorio.

Todas as funções de pertinência foram baseadas nas funções utilizadas no trabalho de Gudwin, (1998), pois as características do edifício deste trabalho são semelhantes às características do presente trabalho.

B - Regras de Inferência

A inferência é realizada baseada em regras definidas. Estas regras podem variar dependendo do tipo de tráfego presente em determinado momento (tráfego normal, tráfego de pico de subida, tráfego de pico de descida). Neste trabalho considerou-se apenas o tráfego de pico de descida por ser um dos mais críticos.

As regras obtidas basearam-se no trabalho de Gudwin, (1998), e foram inseridas no modelo do Matlab® como apresentado na figura 11.

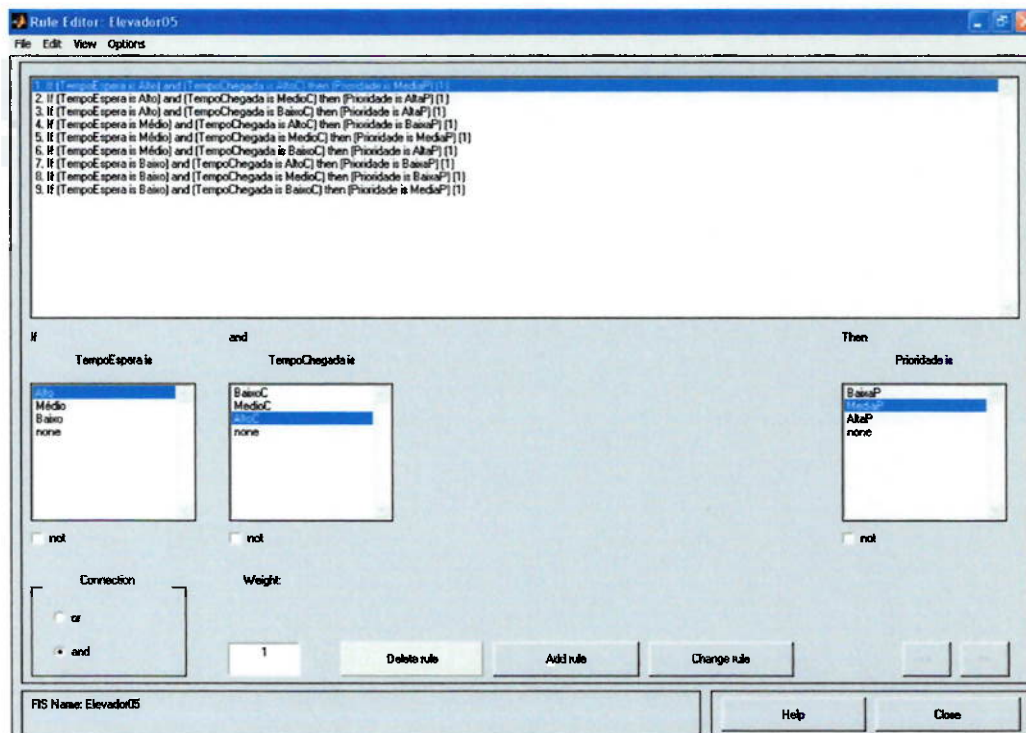


Figura 11 – Regras de inferência utilizadas no sistema

Estas regras podem ser melhor visualizadas na forma de matriz apresentada na tabela 5.

Tabela 5 – Regra para tráfego de descida

Tempo estimado de chegada Tempo de espera E	Alto E	Médio E	Baixo E
	Alto C	Médio C	Baixo C
Alto C	Médio	Baixo	Baixo
Médio C	Alto	Médio	Baixo
Baixo C	Alto	Alto	Médio

Esta tabela faz o papel das regras *if-then*, realizando um *and* entre o tempo de espera e o tempo estimado de chegada do elevador (j) para atender uma chamada no pavimento (i).

Os métodos das funções de inferência são “and”, “or”, “agregation”, “desfuzification”. Neste trabalho foram utilizados as seguintes regras para estas funções, conforme ilustra a tabela 6:

Tabela 6 - Métodos para as funções de inferência

Função de inferência	Regras
“And”	Mínimo
“Or”	Máximo
“Desfuzification”	Centróide

A partir das regras de inferência e das funções de pertinência, é possível desenvolver um gráfico da superfície relacionando as entradas (tempo de espera e tempo estimado de chegada do elevador) com a saída (prioridade). Este gráfico pode ser observado na figura 12.

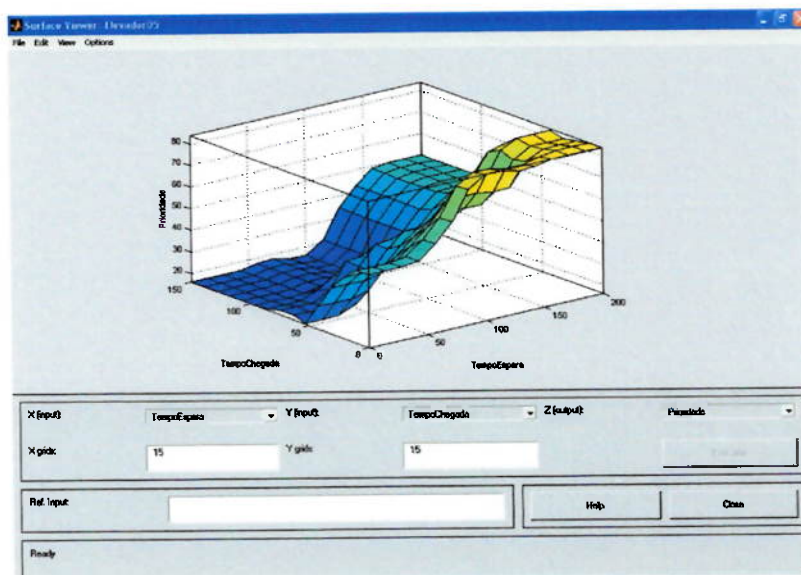


Figura 12 - Gráfico de relacionamento do tempo de espera e do tempo estimado de chegada do elevador com a prioridade.

Para cada conjunto “elevador - pavimento”, quando houver uma chamada, será necessária uma iteração do controlador fuzzy (pois para cada conjunto haverá um tempo de espera, um tempo de chegada e uma prioridade).

C - Teste do Controlador Fuzzy

A fim de testar o controlador, com suas funções de pertinência e regras de inferência, utilizou-se a própria interface Matlab®. Nas figuras 13 e 14 pode-se verificar dois exemplos de funcionamento do controlador.

Para o caso de tempo de espera igual a 60 segundos e tempo estimado de chegada do elevador no pavimento igual a 80 segundos, a prioridade será 31.6 (vide figura 13).

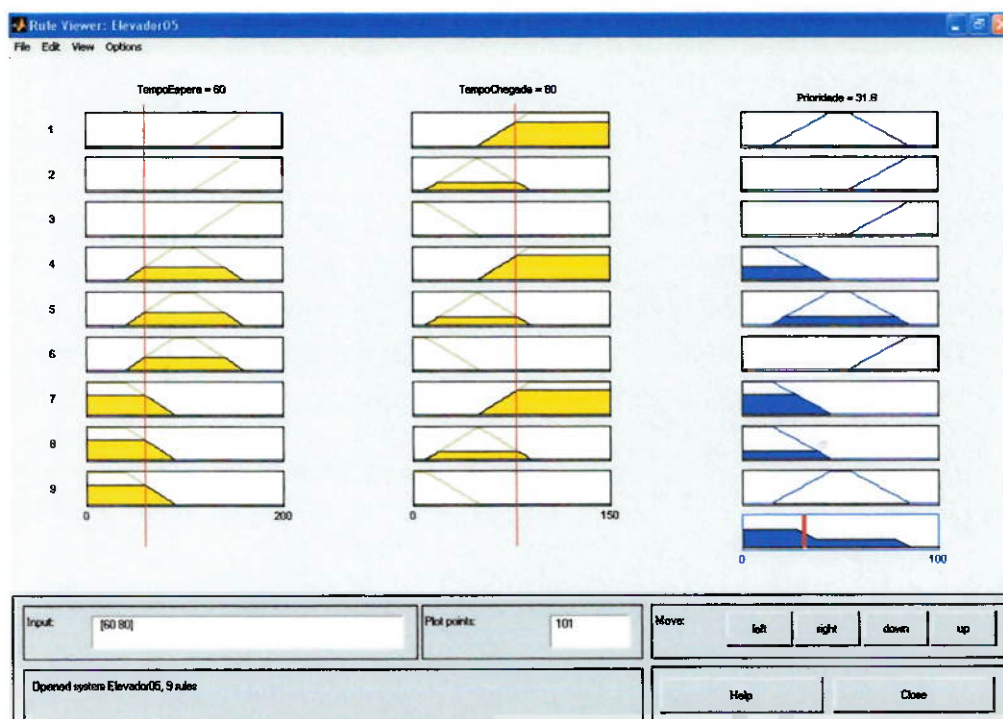


Figura 13 – Visualização do primeiro caso para teste

Mantendo o tempo de espera em 60 segundos, e reduzindo o tempo estimado de chegada do elevador no pavimento para 20 segundos, a prioridade deverá ser

maior, o que pode ser verificado pela figura 14, sendo a prioridade igual a 53,2 neste caso.



Figura 14 – Visualização do segundo caso para teste

Portanto, tem-se como resultado um controlador fuzzy contendo duas entradas (tempo de espera e tempo estimado de chegada do elevador) e do qual se obtém a prioridade de cada chamada para cada elevador.

4.2.3 Estrategia de controle com a integração com o sistema de circuito fechado de televisão

De posse do algoritmo de controle do sistema de elevadores desenvolvido no *software* Matlab®, realizou-se a integração deste sistema com o sistema de circuito fechado de televisão.

Conforme definido anteriormente, as informações adicionais consideradas para o controle do sistema de elevadores são provenientes do sistema de circuito fechado de televisão. Basicamente estas são: o número de usuários aguardando atendimento

em cada pavimento, bem como o número de usuários dentro do elevador, as quais são obtidas através de aquisição, tratamento e processamento das imagens captadas pelas câmeras do sistema de circuito fechado de televisão. Estes procedimentos estão fora do escopo deste projeto, que utilizará estes números como entrada do sistema, considerando que estas etapas foram já executadas.

Assim, tem-se as seguintes informações adicionais que serão ser utilizadas na alocação de chamadas aos elevadores:

- Número de usuários aguardando em cada pavimento;
- Número de usuários presentes em cada elevador.

Primeiramente, pensou-se em utilizar a informação dos usuários aguardando em cada pavimento como entrada adicional no controlador *fuzzy*. No caso de haver mais de 12 pessoas aguardando isto acarretaria um aumento da prioridade, sendo perfeitamente coerente. Porém, este aumento da prioridade ainda implicaria alocar apenas um elevador para esta chamada, o que faria com que o sistema não se aproveitasse completamente das novas informações obtidas com as câmeras.

Tendo sido descartado o uso destas novas informações no controlador *fuzzy*. Passou-se então para a idéia de utilizar estes novos dados na alocação efetiva das chamadas, baseada nas prioridades. Assim, o cálculo das prioridades continua a ser realizado da mesma maneira através do controlador *fuzzy*, porém a alocação das chamadas aos elevadores foi alterada para utilizar as novas informações.

Na alocação da chamada, o controle supervisor aloca o chamado que tiver maior prioridade para cada elevador, e nesse momento verifica se cada elevador será capaz de atender a todos os usuários que realizaram a chamada.

Para realizar esta verificação, o controlador se utiliza da informação do número de usuários aguardando em cada pavimento e o número de usuários presentes em cada elevador. Caso o número de usuários da chamada alocada a um elevador ultrapasse a capacidade deste, o controlador permite a alocação desta chamada a um outro elevador. Para exemplificar esta estratégia, será utilizada a mesma tabela 4, porém com a nova regra. Os resultados são apresentados na tabela 7.

Tabela 7 - Alocação das chamadas considerando o sistema de CFTV

Identificação do pavimento (identificação do sentido)				
Cabine 1	4 (descida)	7 (descida)	3 (descida)	7 (descida)
Cabine 2	13 (descida)	8 (descida)	14 (descida)	3 (descida)
Cabine 3	4 (descida)	10 (descida)	1 (descida)	11 (descida)
Cabine 4	7 (descida)	14 (descida)	8 (descida)	11 (descida)

Ao contrário do que ocorreu sem a integração com o circuito fechado de televisão, neste caso o controlador verificou que no pavimento 4, havia mais pessoas aguardando do que a capacidade da cabine 1. Assim, ao invés de impedir a alocação de outra cabine, permitiu que esta chamada fosse alocada para a cabine 3, reduzindo assim, o tempo de espera dos usuários que estavam aguardando neste pavimento.

4.3 Desenvolvimento do Visualizador

O visualizador foi desenvolvido diretamente no Matlab®, contendo os pavimentos, o número de pessoas aguardando em cada pavimento, bem como as posições dos quatro elevadores, com seus respectivos números de passageiros. O elevador é plotado no andar em que ele se encontra, através do número de passageiros. (vide figura 15)

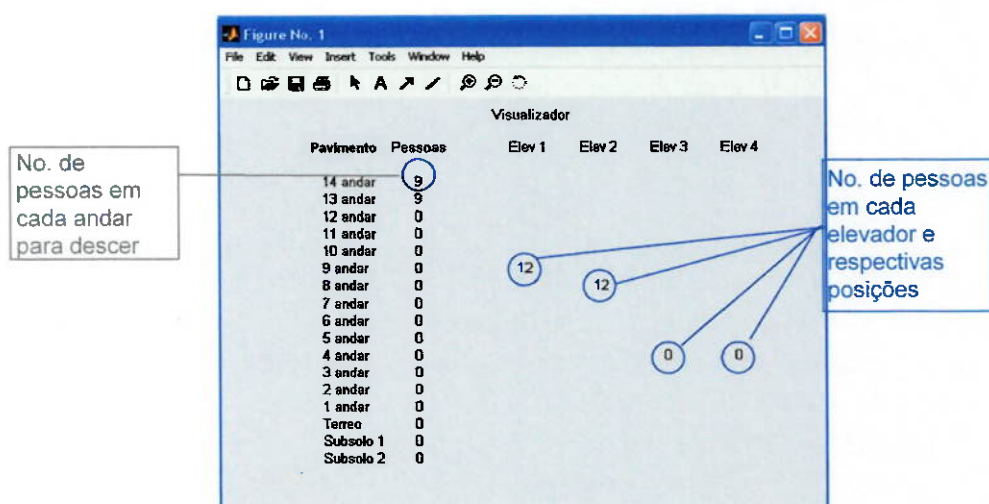


Figura 15 - Visualizador do Simulador

4.4 Resultados Obtidos

De posse do visualizador e do algoritmo desenvolvido no *software* Matlab® contendo tanto os controladores locais de cada elevador como o controle supervisorio com a estratégia de alocação de chamadas, foi possível realizar a simulação do funcionamento dos elevadores.

Para realizar a simulação e possibilitar a comparação dos resultados com e sem a integração do circuito fechado de televisão ao sistema de controle de grupo de elevadores, é necessário que as mesmas entradas sejam fornecidas nos dois casos.

Inicialmente, o algoritmo foi desenvolvido para que fosse possível definir a chegada de pessoas em todos os pavimentos do edifício. Porém, ao se tentar executar este programa no Matlab® (devido a este possuir mais de 20.000 linhas de código) notou-se uma lentidão exagerada na execução da simulação. Esta lentidão foi da ordem de dezenas de minutos para realizar apenas uma passagem do *loop* do programa, inviabilizando o programa cujo objetivo era de realizar simulações em tempo real. Assim, optou-se por retirar do algoritmo as linhas que tratavam dos passageiros chegando nos pavimentos, exceto dos pavimentos 13 e 14 do edifício.

Ao executar o programa, viu-se que o tempo de execução passou a ser razoável e que este estava sendo realizado em tempo real, viabilizando a contagem de tempo para fins de estudo.

A entrada utilizada, que é a mesma para os dois casos em estudo (com e sem a integração do sistema de câmeras), é composta por uma chegada de sete pessoas a cada quinze segundos em cada um dos dois pavimentos (13 e 14).

O visualizador desenvolvido apresenta a seguinte interface (vide figura 16).

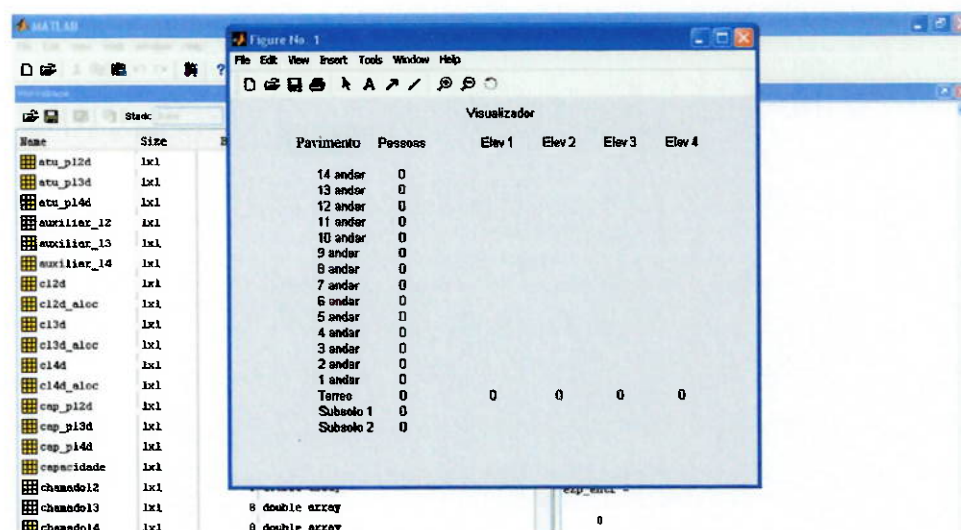


Figura 16 - Interface do visualizador

Apresentam-se nesta interface o pavimento térreo, além dos dois subsolos e dos quatorze andares acima do térreo. Em cada pavimento, tem-se o número de pessoas aguardando o elevador. Como o presente trabalho trata apenas do tráfego de pico de descida, o número de pessoas representa os usuários que solicitam descer.

Além disso, o visualizador apresenta a posição dos elevadores, bem como o número de passageiros presentes neles.

Primeiramente, foi desenvolvido o algoritmo para o caso em que o sistema de elevadores não estaria integrado com o sistema CFTV.

Tem-se o estado inicial, com zero pessoas aguardando o elevador em cada pavimento, com todos os elevadores vazios e aguardando no pavimento térreo (vide figura 17).

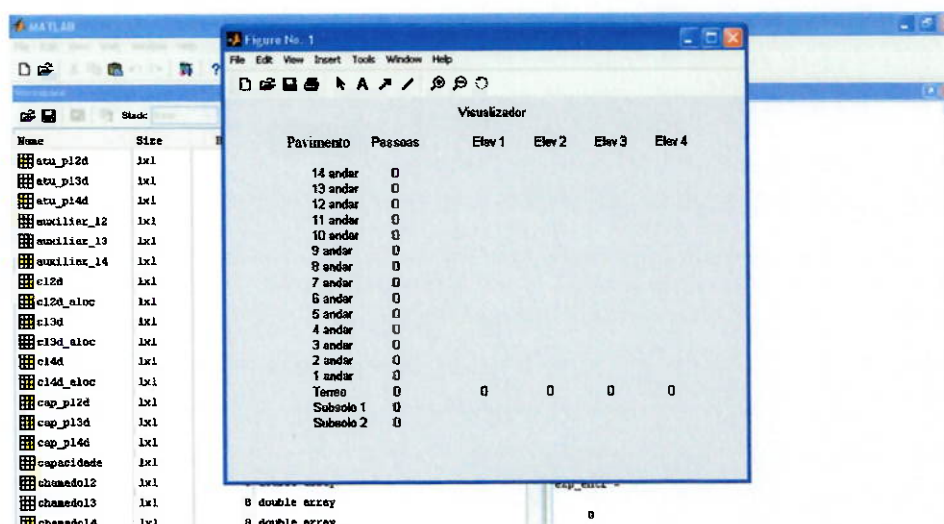


Figura 17 - Situação inicial de simulação

Posteriormente, tem-se a chegada de 7 passageiros em cada um dos pavimentos 13 e 14 (vide figura 18).

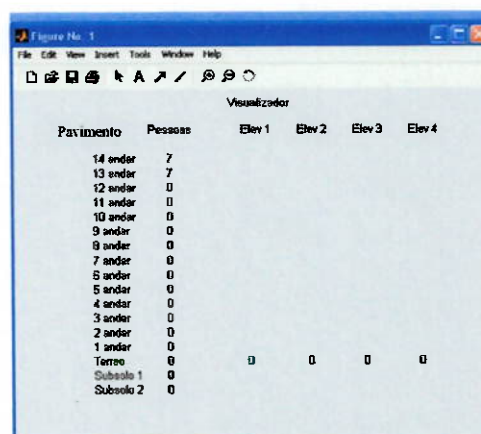


Figura 18 - Chegada das 7 pessoas nos pavimentos 13 e 14

A seguir, tem-se os elevadores 1 e 2 rumo aos pavimentos 13 e 14 que solicitaram chamada, e a chegada de mais 7 pessoas em cada um dos pavimentos 13 e 14 (vide figura 19).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	14				
13 andar	14				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0	0	0		
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0			0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 19 - Elevadores 1 e 2 rumo aos pavimentos 13 e 14 e chegada de mais 7 pessoas nos dois pavimentos

Nota-se que apesar de já haver 14 pessoas aguardando em cada pavimento, apenas dois elevadores foram alocados, pois como o sistema de câmeras não está integrado, o sistema não sabe que em cada pavimento há mais pessoas do que a capacidade dos elevadores.

Tem-se então a chegada dos elevadores 1 e 2 nos pavimentos 13 e 14 respectivamente, com a entrada de 12 pessoas (capacidade máxima) em cada elevador. Nesse instante tem-se a chegada de mais 7 pessoas nos pavimentos 13 e 14. Assim, 12 pessoas (capacidade do elevador) foram atendidas em cada pavimento, restam 9 pessoas no pavimento 13 e 9 pessoas no pavimento 14 (vide figura 20).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	9	12			
13 andar	9		12		
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0			0	0
Térreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 20 - Chegada dos elevadores 1 e 2 nos pavimentos 13 e 14, e entrada de 12 pessoas (capacidade máxima) em cada um

Nesse instante o controlador recebe a informação de que ainda há pessoas aguardando atendimento, e envia os elevadores 3 e 4, enquanto os elevadores 1 e 2 já estão descendo para levar os passageiros a seu andar de destino (vide figura 21).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	9				
13 andar	9				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0	12			
8 andar	0		12		
7 andar	0				
6 andar	0				
5 andar	0			0	0
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Térreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 21 - Descida dos elevadores 1 e 2 com 12 pessoas cada, e subida dos elevadores 3 e 4 a fim de atender às 9 pessoas aguardando nos pavimentos 13 e 14

Os elevadores 1 e 2 chegam ao térreo e as pessoas iniciam o processo de saída. Enquanto isso, os elevadores 3 e 4 descem com os 9 passageiros de cada um dos pavimentos 13 e 14 (vide figura 22).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				9
12 andar	0				
11 andar	0				
10 andar	0			9	
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Térreo	0	12	12		
Subsolo 1	0				
Subsolo 2	0				

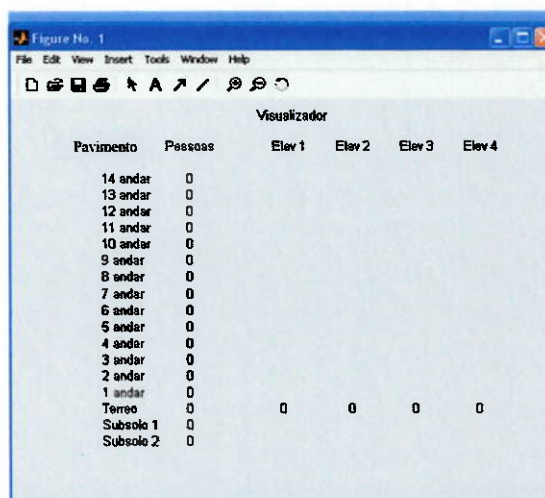
Figura 22 - Descarregamento dos elevadores 1 e 2 no pavimento térreo, e descida dos elevadores 3 e 4 com nove passageiros cada

Os elevadores 3 e 4 chegam ao pavimento térreo a fim de possibilitar a descida de seus usuários (vide figura 23).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Térreo	0	0	0	9	9
Subsolo 1	0				
Subsolo 2	0				

Figura 23 - Chegada dos elevadores 3 e 4 no pavimento térreo para descarregamento

Situação final do programa, com todos os elevadores vazios e nenhum usuário aguardando atendimento (vide figura 24).



Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Térreo	0	0	0	0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 24 - Situação final da simulação sem considerar as informações das câmeras

A seguir, foi desenvolvido o algoritmo que se utiliza das novas informações advindas da integração com o circuito fechado de televisão. Estas informações são utilizadas na alocação de chamadas, e ao alocar uma chamada para um elevador, o sistema verifica constantemente se apenas este elevador será capaz de atender todos os usuários aguardando no pavimento. Para isto, baseia-se no número de pessoas em cada pavimento e o número de pessoas no elevador dado pelo sistema de CFTV.

Assim, obteve-se a seguinte sequência na simulação:

Primeiramente, temos a situação inicial com todos elevadores vazios e parados no pavimento térreo, sem pessoas aguardando atendimento (vide figura 25).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Térreo	0	0	0	0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 25 - Situação inicial com câmera

A seguir, tem-se a chegada de 7 pessoas em cada um dos pavimentos 13 e 14, com o controlador alocando cada uma dessas chamadas aos elevadores 1 e 2, que começam a subir (vide figura 26).

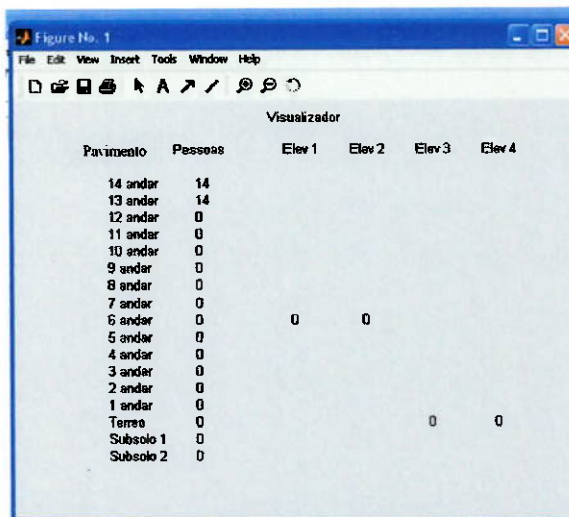
Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	7				
13 andar	7				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0	0	0		
Térreo	0			0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 26 - Chegada de 7 usuários aguardando em cada um dos pavimentos 13 e 14, e elevadores 1 e 2 subindo para atender as chamadas

Quando os elevadores 1 e 2 estão no sexto pavimento, há a chegada de mais 7 pessoas em cada um dos pavimentos 13 e 14. A câmera, juntamente com seu sistema de aquisição, tratamento e processamento das imagens envia estes dados ao sistema de elevadores. O sistema de controle de elevadores verifica que havendo 14 pessoas

em cada pavimento, os elevadores 1 e 2 não poderão atender a todos, já que a capacidade de cada elevador é de 12 pessoas.

Assim, o controle supervisão envia os elevadores 3 e 4 (já que os dois estavam ociosos, a fim de minimizar o tempo de espera) para atender as pessoas que restarão nos pavimentos 14 e 13 (vide figura 27).



Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	14				
13 andar	14				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0	0	0		
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0			0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 27 - Chegada de mais 7 pessoas em cada um dos pavimentos 13 e 14

Tem-se portanto os elevadores 1 e 3 em caminho para atender as pessoas que estão no pavimento 14, bem como os elevadores 2 e 4 para atender as pessoas que estão no pavimento 13 (vide figura 28).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	14				
13 andar	14				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0	0	0		
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0			0	0
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 28 - Controlador envia os elevadores 3 e 4 para as chamadas dos pavimentos 13 e 14

A seguir, os elevadores 1 e 2 chegam nos andares 14 e 13 respectivamente, e atendem 12 usuários. Neste instante, chegam mais 7 usuários em cada um dos pavimentos 13 e 14, restando portanto 9 usuários a serem atendidos nos pavimentos 13 e 14, sendo que os elevadores 3 e 4 já tinham sido alocados pelo controlador supervisorio que percebeu a necessidade através das novas informações advindas das câmeras (vide figura 29).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	9	12			
13 andar	9		12		
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0			0	0
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 29 - Elevadores 1 e 2 chegam aos pavimentos 14 e 13 respectivamente, enquanto que os elevadores 3 e 4 já estão subindo.

Os elevadores 1 e 2 descem com 12 pessoas cada um enquanto os elevadores 3 e 4 já estão chegando aos pavimentos 13 e 14 (vide figura 30).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	9				
13 andar	9				
12 andar	0	12			
11 andar	0				
10 andar	0		12	0	0
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 30 - Elevadores 1 e 2 iniciam movimento de descida com 12 passageiros cada, enquanto elevadores 3 e 4 estão chegando aos andares 13 e 14.

Os elevadores 3 e 4 chegam aos pavimentos 14 e 13, e atendem os 9 usuários restantes em cada pavimento, enquanto os elevadores 1 e 2 já estão chegando ao seu pavimento de destino (vide figura 31).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0			9	
13 andar	0				9
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0	12	12		
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0				
Subsolo 1	0				
Subsolo 2	0				

Figura 31 - Elevadores 3 e 4 chegam aos pavimentos 14 e 13 e atendem aos 9 usuários de cada pavimento, enquanto elevadores 1 e 2 estão descendo.

Os elevadores 1 e 2 atingem seu pavimento de destino e começam a descarregar os passageiros. Enquanto isso, os elevadores 3 e 4 estão descendo com 9 passageiros cada um (vide figura 32).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0			9	
5 andar	0				9
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0	12	12		
Subsolo 1	0				
Subsolo 2	0				

Figura 32 - Elevadores 1 e 2 atingem seu destino, enquanto elevadores 3 e 4 estão descendo.

Os elevadores 3 e 4 também atingem seu pavimento de destino, e começam a descarregar os passageiros (vide figura 33).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0	0	0	9	9
Subsolo 1	0				
Subsolo 2	0				

Figura 33 - Elevadores 3 e 4 também chegam ao seu destino

Finalmente todos os usuários atingiram seu pavimento de destino, os elevadores estão vazios e não há nenhum usuário aguardando atendimento em algum dos pavimentos (vide figura 34).

Pavimento	Pessoas	Elev 1	Elev 2	Elev 3	Elev 4
14 andar	0				
13 andar	0				
12 andar	0				
11 andar	0				
10 andar	0				
9 andar	0				
8 andar	0				
7 andar	0				
6 andar	0				
5 andar	0				
4 andar	0				
3 andar	0				
2 andar	0				
1 andar	0				
Terreo	0	0	0	0	0
Subsolo 1	0				
Subsolo 2	0				

Figura 34 - Situação final da simulação considerando as informações das câmeras

Além deste caso, foram realizadas outras simulações com diferentes tempos de chegada e diferentes números de pessoas esperando serviço. O caso visto acima e mais duas simulações com seus respectivos tempos foram os seguintes:

- **Caso 1:** Chegada de 7 passageiros a cada 15 segundos em cada um dos pavimentos 13 e 14 tendo como destino o pavimento térreo, sendo a primeira chegada 5 segundos após o início da simulação totalizando 21 passageiros em cada um dos pavimentos.
- **Caso 2:** Chegada de 3 passageiros a cada 5 segundos em cada um dos pavimentos 13 e 14 tendo como destino o pavimento térreo, sendo a primeira chegada 5 segundos após o início da simulação totalizando 18 passageiros em cada pavimento.
- **Caso 3:** No pavimento 14, chegada de 7 passageiros a cada 10 segundos, com a primeira chegada 5 segundos após o início da simulação, totalizando 21 passageiros. No pavimento 14, chegada de 4 passageiros a cada 7 segundos, com a primeira chegada 5 segundos após o início da simulação, sendo 20 o número total de pessoas.

A seguir (vide tabela 8) apresenta-se a comparação dos tempos obtidos sem e com a integração do sistema de elevadores com o sistema de circuito fechado de televisão.

Tabela 8 - Comparação dos tempos sem e com a integração com sistema de circuito fechado de televisão

	Variável	Sem Cameras (s)	Com Câmeras (s)	Redução(%)
Caso 1	Tempo que a chamada 13 ficou ativada	62,6	44,894	28,3%
	Tempo que a chamada 14 ficou ativada	64,853	46,026	29,0%
	Tempo total p/ saída dos usuários	111,7	91,942	17,7%
Caso 2	Tempo que a chamada 13 ficou ativada	59,235	59,153	0,1%
	Tempo que a chamada 14 ficou ativada	62,599	62,51	0,1%
	Tempo total p/ saída dos usuários	103,489	103,429	0,1%
Caso 3	Tempo que a chamada 13 ficou ativada	61,518	50,052	18,6%
	Tempo que a chamada 14 ficou ativada	63,772	40,258	36,9%
	Tempo total p/ saída dos usuários	109,217	99,583	8,8%

Pode-se perceber pela tabela de tempos dos casos em estudo que houve uma redução do tempo em que as chamadas ficaram ativadas, bem como do tempo total para que todos os usuários chegassem dos seus andares de origem para seus destinos.

Esta redução foi sensível nos casos 1 e 3, pois nestes exemplos a fila de pessoas, bem como a taxa de chegada, fizeram com que em algum momento houvesse mais de 12 pessoas aguardando atendimento, sobrepassando portanto a capacidade máxima do elevador.

Isto porque, através da informação vinda do sistema de CFTV, o controlador já tomou conhecimento da existência de um número de pessoas aguardando maior que a capacidade do elevador que estava alocado para a chamada. Assim, o controlador pôde então de posse destas novas informações se antecipar e enviar outro elevador para a chamada.

Percebe-se portanto que antes de utilizar esta nova funcionalidade, deve-se realizar um estudo no edifício a fim de verificar se nos momentos de tráfego de pico de subida e de descida, há uma ocorrência grande de vezes em que o número de pessoas aguardando atendimento supera a capacidade do elevador. Dentre os casos testados neste trabalho, percebe-se que nos casos 1 e 3 foi verificada esta situação e, portanto, houve uma redução de cerca de 28% e de 18% a 36% no tempo de atendimento aos chamados respectivamente.

5 Conclusão

Através deste trabalho foi possível analisar a importância da integração dos sistemas prediais em edifícios inteligentes, mais especificamente do sistema de elevadores, que é um sistema crítico em edifícios com grande número de pessoas.

Pôde-se verificar os benefícios que podem ser obtidos ao integrar sistemas que muitas vezes já estão presentes isoladamente em edifícios, tais como o sistema de elevadores e o sistema de circuito fechado de televisão.

O algoritmo foi desenvolvido com sucesso, tendo sido simplificado devido à limitação da capacidade de processamento do Matlab®. Além disso, o visualizador também pôde ser desenvolvido no mesmo *software*, facilitando assim a interface entre o sistema de controle e o visualizador.

De posse do algoritmo e do visualizador, foram realizadas as simulações de casos a fim de verificar o benefício trazido pela integração do sistema de elevadores com o sistema de circuito fechado de televisão.

Notou-se pelos resultados obtidos que nos casos em que havia mais pessoas aguardando o elevador do que a capacidade deste, a integração propicia uma redução considerável do tempo de espera do usuário, validando assim o investimento a ser realizado.

Conclui-se assim que no caso de edifícios em que há frequentemente um número de pessoas aguardando atendimento maior que a capacidade do elevador, pode-se sugerir a utilização da integração do sistema de elevadores com o sistema de circuito fechado de televisão devido à sensível redução dos tempos de espera dos usuários.

6 Referências Bibliográficas

- GODOY, J. **Controle vertical em edifícios: Uma análise de Fluxo de Tráfego e Consumo Energético**.1999. 186p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 1999.
- GUDWIN, R.; GOMIDE, F A **Fuzzy Elevator Group Controller with Linear Context Adaptation**. In: FUZZ-IEEE98, WCCI'98 - IEEE World Congress on Computational Intelligence, 1998.
- BASTIDAS GUSTIN, G. **Aplicação de Redes de Petri interpretadas na modelagem de sistemas de elevadores em edifícios inteligentes**.1999. 64p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 1999.
- INDÚSTRIAS VILLARES S. A. **Manual de transporte vertical em edifícios – 17**. Ed. – São Paulo: Pini, 1994.
- MENEZES, F.; YOUSSEF A.; JUNIOR M. **Lógica Fuzzy**. Disponível em (15/07/2003): http://bigbob.sites.uol.com.br/fuzzy_1.htm.
- MIYAGI, P.E.: **Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos**, Editora Edgard Blücher Ltda., São Paulo, Brasil, 1999.
- SASAKI, K.; MARKON, S.; MAKAGAWA, M. **Elevator Group Supervisory Control System Using Neural Networks**. Elevator World Magazine, Janeiro 1996.
- SIKONEN, M.L. Simulation - A Tool for Enhanced Elevator Bank Design. **Elevator World**. April, 1991.

SIKKONEN, M.L. **Elevator Group Control with Artificial Intelligence**. Helsinki, 1997.

THUMM, G. Elevators and Escalators Within Intelligent Buildings: New Methods and Technologies in Planning and Construction of Intelligent Building I. In *IB/IC Intelligent Buildings Congress*. Ed. A. Lustig. Tel-Aviv, 1995.

TRILLAS, E.; SKALA, H.J.; TERMINI S. **Aspects of Vagueness**. Dordrecht and Boston: D. Reidel, 1984.

UMEDA, Y.; UETANI, K.; UJIHARA, H.; Tsuji, S. **Fuzzy Theory and Intelligent Options**. Elevator World. Julho, 1989.

ZADEH, L. **Fuzzy Sets**, Information and Control, John Wiley & Sons, Inc, Vol. 8, 1965.

APÊNDICE I

**ALGORITMO DE SIMULAÇÃO
CONSIDERANDO AS INFORMACOES
ADICIONAIS DO CIRCUITO FECHADO DE
TELEVISAO**

```

% Carrega o modelo fuzzy na variável elev
elevfuzzy=readfis('elevador0');

saida=0;

% Chamadas de Pisos

% descida
c12d=0;
c13d=0;
c14d=0;

% Já tem alocado qtas pessoas no andar?

% descida
c12d_aloc=0;
c13d_aloc=0;
c14d_aloc=0;

% Tempos totais que as chamadas ficaram ativas

ttot_c12d=0;
ttot_c13d=0;
ttot_c14d=0;

% Pessoas em Cada Andar

% Para Descer, e Index cumulativo do Último a subir no elevador
% e Index cumulativo do atual aguardando para subir no elevador
% em cada andar, com uma fila (filap***d)

pm02d=0;
pm01d=0;
p00d=0;
p01d=0;
p02d=0;
p03d=0;
p04d=0;
p05d=0;
p06d=0;
p07d=0;
p08d=0;
p09d=0;
p10d=0;
p11d=0;
p12d=0;
ult_p12d=0;
atu_p12d=0;
cap_p12d=0;
p13d=0;
ult_p13d=0;
atu_p13d=0;
cap_p13d=0;
p14d=0;
ult_p14d=0;
atu_p14d=0;
cap_p14d=0;

% CAPACIDADE DO ELEVADOR
capacidade=12;

% ELEVADOR 01

% Situação (p_ok = parado e não ok,
% p_saindo = parado e saindo gente,
% p_saindo & = parado e já saíram,
% p_entrando = parado e entrando gente,
% p_ok = parado e ok para partir,
% a = andando)
el=998;

% Direção (s=subindo, d=descendo, n=nenhum)
eld=0;

% Pessoas no elevador
elp=0;

% Pessoas que vão entrar no elevador no andar
elp_entr=0;

% Posição
elam02=0;
elam01=0;
ela00=1;
ela01=0;
ela02=0;
ela03=0;
ela04=0;

```

```
ela05=0;
ela06=0;
ela07=0;
ela08=0;
ela09=0;
ela10=0;
ela11=0;
ela12=0;
ela13=0;
ela14=0;
```

```
%ultima posicao do elevador
```

```
elam02_ult=0;
elam01_ult=0;
ela00_ult=1;
ela01_ult=0;
ela02_ult=0;
ela03_ult=0;
ela04_ult=0;
ela05_ult=0;
ela06_ult=0;
ela07_ult=0;
ela08_ult=0;
ela09_ult=0;
ela10_ult=0;
ela11_ult=0;
ela12_ult=0;
ela13_ult=0;
ela14_ult=0;
```

```
%Botões Interiores
```

```
elbm02=0;
elbm01=0;
elb00=0;
elb01=0;
elb02=0;
elb03=0;
elb04=0;
elb05=0;
elb06=0;
elb07=0;
elb08=0;
elb09=0;
elb10=0;
elb11=0;
elb12=0;
elb13=0;
elb14=0;
```

```
%Passageiros no Elevador 01 querendo ir para cada um dos Andares
```

```
pelm02=0;
pelm01=0;
pel00=0;
```

```
%Chamadas elevadas ao Elevador 01 de cada andar pelo controle supervisório
```

```
%para subir e descer
```

```
elcm02s=0;
elcm01s=0;
elc00s=0;
elc01s=0;
elc02s=0;
elc03s=0;
elc04s=0;
elc05s=0;
elc06s=0;
elc07s=0;
elc08s=0;
elc09s=0;
elc10s=0;
elc11s=0;
elc12s=0;
elc13s=0;
elc14s=0;
```

```
elcm02d=0;
elcm01d=0;
elc00d=0;
elc01d=0;
elc02d=0;
elc03d=0;
elc04d=0;
elc05d=0;
elc06d=0;
elc07d=0;
elc08d=0;
elc09d=0;
elc10d=0;
elc11d=0;
elc12d=0;
elc13d=0;
elc14d=0;
```

```
%ELEVADOR 01
```

```

% Situação (p - nok = parado e não ok,
%      p_saindo = parado e saindo gente,
%      p_saindoek = parado e já saiu,
%      p_entrando = parado e entrando gente,
%      p_ok = parado e ok para partir,
%      a = andando)
e2=998;

% Direção (s=subindo, d=descendo, n=nenhum)
e2d='';

% Pessoas no elevador
e2p=0;

% Pessoas que vão entrar no elevador no andar
e2p_entr=0;

% Posicao
e2am02=0;
e2am01=0;
e2a00=1;
e2a01=0;
e2a02=0;
e2a03=0;
e2a04=0;
e2a05=0;
e2a06=0;
e2a07=0;
e2a08=0;
e2a09=0;
e2a10=0;
e2a11=0;
e2a12=0;
e2a13=0;
e2a14=0;

% Última posicao do elevador
e2am02_ult=0;
e2am01_ult=0;
e2a00_ult=1;
e2a01_ult=0;
e2a02_ult=0;
e2a03_ult=0;
e2a04_ult=0;
e2a05_ult=0;
e2a06_ult=0;
e2a07_ult=0;
e2a08_ult=0;
e2a09_ult=0;
e2a10_ult=0;
e2a11_ult=0;
e2a12_ult=0;
e2a13_ult=0;
e2a14_ult=0;

% Botões Interiores
e2bm02=0;
e2bm01=0;
e2b00=0;
e2b01=0;
e2b02=0;
e2b03=0;
e2b04=0;
e2b05=0;
e2b06=0;
e2b07=0;
e2b08=0;
e2b09=0;
e2b10=0;
e2b11=0;
e2b12=0;
e2b13=0;
e2b14=0;

% Passageiros no Elevador 01 querendo ir para cada um dos Andares
pe2m02=0;
pe2m01=0;
pe200=0;

% Chamadas alocadas ao Elevador 01 de cada andar pelo controle superveniente
% para subir e descer
e2cm02s=0;
e2cm01s=0;
e2c00s=0;
e2c01s=0;
e2c02s=0;
e2c03s=0;
e2c04s=0;
e2c05s=0;
e2c06s=0;
e2c07s=0;
e2c08s=0;
e2c09s=0;

```

```
e2c10s=0;
e2c11s=0;
e2c12s=0;
e2c13s=0;
e2c14s=0;
```

```
e2cm02d=0;
e2cm01d=0;
e2c00d=0;
e2c01d=0;
e2c02d=0;
e2c03d=0;
e2c04d=0;
e2c05d=0;
e2c06d=0;
e2c07d=0;
e2c08d=0;
e2c09d=0;
e2c10d=0;
e2c11d=0;
e2c12d=0;
e2c13d=0;
e2c14d=0;
```

```
%ELEVADOR 03
```

```
%Situacao (p, nck = parado e nao ok,
% p_sando = parado e saindo gente,
% p_saidack = parado e ja saiu,
% p_entrando = parado e entrando gente,
% p_ok = parado e ok para partir,
% a = andando)
```

```
e3=998;
```

```
%Direcao (s=subindo, d=descendo, n=nenhum)
```

```
e3d=;
```

```
%Pessoas no elevador
```

```
e3p=0;
```

```
%Pessoas que vao entrar no elevador no andar
```

```
e3p_entr=0;
```

```
%Posicao
```

```
e3am02=0;
e3am01=0;
e3a00=1;
e3a01=0;
e3a02=0;
e3a03=0;
e3a04=0;
e3a05=0;
e3a06=0;
e3a07=0;
e3a08=0;
e3a09=0;
e3a10=0;
e3a11=0;
e3a12=0;
e3a13=0;
e3a14=0;
```

```
%Ultima posicao do elevador
```

```
e3am02_ult=0;
e3am01_ult=0;
e3a00_ult=1;
e3a01_ult=0;
e3a02_ult=0;
e3a03_ult=0;
e3a04_ult=0;
e3a05_ult=0;
e3a06_ult=0;
e3a07_ult=0;
e3a08_ult=0;
e3a09_ult=0;
e3a10_ult=0;
e3a11_ult=0;
e3a12_ult=0;
e3a13_ult=0;
e3a14_ult=0;
```

```
e3bm02=0;
e3bm01=0;
e3b00=0;
e3b01=0;
e3b02=0;
e3b03=0;
e3b04=0;
e3b05=0;
e3b06=0;
e3b07=0;
e3b08=0;
e3b09=0;
e3b10=0;
e3b11=0;
```

```
e3b12=0;
e3b13=0;
e3b14=0;
```

```
%Passageiros no Elevador 01 querendo ir para cada um dos Andares
```

```
pe3m02=0;
pe3m01=0;
pe300=0;
```

```
%Chamadas alocadas ao Elevador 01 de cada andar pelo controle supervisorio
```

```
%-para subir e descer
```

```
e3cm02s=0;
e3cm01s=0;
e3c00s=0;
e3c01s=0;
e3c02s=0;
e3c03s=0;
e3c04s=0;
e3c05s=0;
e3c06s=0;
e3c07s=0;
e3c08s=0;
e3c09s=0;
e3c10s=0;
e3c11s=0;
e3c12s=0;
e3c13s=0;
e3c14s=0;
```

```
e3cm02d=0;
e3cm01d=0;
e3c00d=0;
e3c01d=0;
e3c02d=0;
e3c03d=0;
e3c04d=0;
e3c05d=0;
e3c06d=0;
e3c07d=0;
e3c08d=0;
e3c09d=0;
e3c10d=0;
e3c11d=0;
e3c12d=0;
e3c13d=0;
e3c14d=0;
```

```
%ELEVADOR 04
```

```
%Situacao (p, n, ck = parado e nao ok,
```

```
% p_saindo = parado e saindo gente,
```

```
% p_saindo = parado e ja sairam,
```

```
% p_entrando = parado e entrando gente,
```

```
% p_ck = parado e ck para partir,
```

```
% a = andando)
```

```
e4=998;
```

```
%Direcao (s=subindo, d=descendo, n=nenhum)
```

```
e4d=0;
```

```
%Pessoas no elevador
```

```
e4p=0;
```

```
%Pessoas que vao entrar no elevador no andar
```

```
e4p_entr=0;
```

```
%Posicao
```

```
e4am02=0;
```

```
e4am01=0;
```

```
e4a00=1;
```

```
e4a01=0;
```

```
e4a02=0;
```

```
e4a03=0;
```

```
e4a04=0;
```

```
e4a05=0;
```

```
e4a06=0;
```

```
e4a07=0;
```

```
e4a08=0;
```

```
e4a09=0;
```

```
e4a10=0;
```

```
e4a11=0;
```

```
e4a12=0;
```

```
e4a13=0;
```

```
e4a14=0;
```

```
%Ultima posicao do elevador
```

```
e4am02_ult=0;
```

```
e4am01_ult=0;
```

```
e4a00_ult=1;
```

```
e4a01_ult=0;
```

```
e4a02_ult=0;
```

```
e4a03_ult=0;
```

```
e4a04_ult=0;
e4a05_ult=0;
e4a06_ult=0;
e4a07_ult=0;
e4a08_ult=0;
e4a09_ult=0;
e4a10_ult=0;
e4a11_ult=0;
e4a12_ult=0;
e4a13_ult=0;
e4a14_ult=0;
```

```
%Botões Internos
```

```
e4bm02=0;
e4bm01=0;
e4b00=0;
e4b01=0;
e4b02=0;
e4b03=0;
e4b04=0;
e4b05=0;
e4b06=0;
e4b07=0;
e4b08=0;
e4b09=0;
e4b10=0;
e4b11=0;
e4b12=0;
e4b13=0;
e4b14=0;
```

```
%Passageiros no Elevador 01 querendo ir para cada um dos Andares
```

```
pe4m02=0;
pe4m01=0;
pe400=0;
```

```
%Chamadas alocadas ao Elevador 01 de cada andar pelo controle superviso
```

```
%para subir e descer
```

```
e4cm02s=0;
e4cm01s=0;
e4c00s=0;
e4c01s=0;
e4c02s=0;
e4c03s=0;
e4c04s=0;
e4c05s=0;
e4c06s=0;
e4c07s=0;
e4c08s=0;
e4c09s=0;
e4c10s=0;
e4c11s=0;
e4c12s=0;
e4c13s=0;
e4c14s=0;
```

```
e4cm02d=0;
e4cm01d=0;
e4c00d=0;
e4c01d=0;
e4c02d=0;
e4c03d=0;
e4c04d=0;
e4c05d=0;
e4c06d=0;
e4c07d=0;
e4c08d=0;
e4c09d=0;
e4c10d=0;
e4c11d=0;
e4c12d=0;
e4c13d=0;
e4c14d=0;
```

```
% CHEGADA DE PESSOAS (para descer)
```

```
%Andar 14
```

```
chegada14=timer('StartDelay',5,'Period',10,'ExecutionMode','fixedRate','TasksToExecute',3,'TimerFor','on p14d-cap p14d+1');
start(chegada14);
```

```
%Andar 13
```

```
chegada13=timer('StartDelay',5,'Period',7,'ExecutionMode','fixedRate','TasksToExecute',5,'TimerFor','on p13d-cap p13d+4');
start(chegada13);
```

```
tic;
```

```
tempodecorrido=0;
```

```
while saida==0
```

```
%Chegada de pessoas nos pavimentos (para descer)
```

```
u atu_p14d--cap_p14d && filap14d(atu_p14d+1)--1
```

```

while cap_p14d>atu_p14d
    if filap14d(atu_p14d+1)~=1
        atu_p14d=atu_p14d+1;
        p14d=p14d+1;
        c14d=0
        tc14d_ini=clock;
    end
    c14d=1;
end
end
end

if atu_p13d==cap_p13d && filap13d(atu_p13d+1)~=1
    while cap_p13d>atu_p13d
        if filap13d(atu_p13d+1)~=1
            atu_p13d=atu_p13d+1;
            p13d=p13d+1;
            c13d=0
            tc13d_ini=clock;
        end
        c13d=1;
    end
end
end

if atu_p12d==cap_p12d && filap12d(atu_p12d+1)~=1
    while cap_p12d>atu_p12d
        if filap12d(atu_p12d+1)~=1
            atu_p12d=atu_p12d+1;
            p12d=p12d+1;
            c12d=0
            tc12d_ini=clock;
        end
        c12d=1;
    end
end
end

%*****
%*****
% CONTROLE SUPERVISORIO
%*****
%*****

%
%Calculo do tempo estimado de chegada do elevador
%*****

if e1c14d==0 && e2c14d==0 && e3c14d==0 && e4c14d==0 && e1==997 && e2==997 && e3==997 && e4==997
    c14d_aloc=0;
end
if e1c13d==0 && e2c13d==0 && e3c13d==0 && e4c13d==0 && e1==997 && e2==997 && e3==997 && e4==997
    c13d_aloc=0;
end

%Se tiver chamada do pavimento
if c14d==1
    %Se ainda nao tem nenhum alocado
    if c14d_aloc==0
        %
        %elevador 1
        if e1a14==1 && e1d==0
            tchegada_e1a14=0;
        end
        if e1a13_ult==1
            if e1d==0 || e1d==1
                tohchegada_e1a14=2*1;
            else
                tchegada_e1a14=100;
            end
        end
        if e1a12_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a14=2*2;
            else
                tchegada_e1a14=100;
            end
        end
        if e1a11_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a14=2*3;
            else
                tchegada_e1a14=100;
            end
        end
        if e1a10_ult==1
            if e1d==0 || e1d==1
                tohchegada_e1a14=2*4;
            else
                tchegada_e1a14=100;
            end
        end
        if e1a09_ult==1

```

```

    if e1d==0 || e1d==1
        tchegada_e1a14=2*5;
    else
        tchegada_e1a14=100;
    end
end
if e1a08_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*6;
    else
        tchegada_e1a14=100;
    end
end
if e1a07_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*7;
    else
        tchegada_e1a14=100;
    end
end
if e1a06_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*8;
    else
        tchegada_e1a14=100;
    end
end
if e1a05_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*9;
    else
        tchegada_e1a14=100;
    end
end
if e1a04_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*10;
    else
        tchegada_e1a14=100;
    end
end
if e1a03_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*11;
    else
        tchegada_e1a14=100;
    end
end
if e1a02_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*12;
    else
        tchegada_e1a14=100;
    end
end
if e1a01_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*13;
    else
        tchegada_e1a14=100;
    end
end
if e1a00_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*14;
    else
        tchegada_e1a14=100;
    end
end
if e1a001_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*15;
    else
        tchegada_e1a14=100;
    end
end
if e1a002_ult==1
    if e1d==0 || e1d==1
        tchegada_e1a14=2*16;
    else
        tchegada_e1a14=100;
    end
end
end

%
%elevador 2
if e2a14==1 && e2d==0
    tchegada_e2a14=0;
end
if e2a13_ult==1
    if e2d==0 || e2d==1
        tchegada_e2a14=2*1;
    else
        tchegada_e2a14=100;
    end
end
end

```

```

if e2a12_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*2;
    else
        tchegada_e2a14=100;
    end
end
if e2a11_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*3;
    else
        tchegada_e2a14=100;
    end
end
if e2a10_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*4;
    else
        tchegada_e2a14=100;
    end
end
if e2a09_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*5;
    else
        tchegada_e2a14=100;
    end
end
if e2a08_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*6;
    else
        tchegada_e2a14=100;
    end
end
if e2a07_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*7;
    else
        tchegada_e2a14=100;
    end
end
if e2a06_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*8;
    else
        tchegada_e2a14=100;
    end
end
if e2a05_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*9;
    else
        tchegada_e2a14=100;
    end
end
if e2a04_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*10;
    else
        tchegada_e2a14=100;
    end
end
if e2a03_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*11;
    else
        tchegada_e2a14=100;
    end
end
if e2a02_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*12;
    else
        tchegada_e2a14=100;
    end
end
if e2a01_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*13;
    else
        tchegada_e2a14=100;
    end
end
if e2a00_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*14;
    else
        tchegada_e2a14=100;
    end
end
if e2am01_ult==1
    if e2d==0 || e2d==5
        tchegada_e2a14=2*15;
    else
        tchegada_e2a14=100;
    end
end

```

```

end
end
if e2a02_ult==1
    if e2d==0 || e2d==1
        tchegada_e2a14=2*16;
    else
        tchegada_e2a14=100;
    end
end

%elevador 3
if e3a14==1 && e3d==0
    tchegada_e3a14=0;
end
if e3a13_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*1;
    else
        tchegada_e3a14=100;
    end
end
if e3a12_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*2;
    else
        tchegada_e3a14=100;
    end
end
if e3a11_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*3;
    else
        tchegada_e3a14=100;
    end
end
if e3a10_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*4;
    else
        tchegada_e3a14=100;
    end
end
if e3a09_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*5;
    else
        tchegada_e3a14=100;
    end
end
if e3a08_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*6;
    else
        tchegada_e3a14=100;
    end
end
if e3a07_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*7;
    else
        tchegada_e3a14=100;
    end
end
if e3a06_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*8;
    else
        tchegada_e3a14=100;
    end
end
if e3a05_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*9;
    else
        tchegada_e3a14=100;
    end
end
if e3a04_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*10;
    else
        tchegada_e3a14=100;
    end
end
if e3a03_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*11;
    else
        tchegada_e3a14=100;
    end
end
if e3a02_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a14=2*12;
    end
end

```

```

else
    tchegada_e3a14=100;
end
end
if e3a01_ult==1
    if e3d==0 || e3d==
        tchegada_e3a14=2*13;
    else
        tchegada_e3a14=100;
    end
end
if e3a00_ult==1
    if e3d==0 || e3d==
        tchegada_e3a14=2*14;
    else
        tchegada_e3a14=100;
    end
end
if e3a01_ult==1
    if e3d==0 || e3d==
        tchegada_e3a14=2*15;
    else
        tchegada_e3a14=100;
    end
end
if e3a02_ult==1
    if e3d==0 || e3d==
        tchegada_e3a14=2*16;
    else
        tchegada_e3a14=100;
    end
end
end

%-----
%elevador 4
if e4a14==1 && e4d==
    tchegada_e4a14=0;
end
if e4a13_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*1;
    else
        tchegada_e4a14=100;
    end
end
if e4a12_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*2;
    else
        tchegada_e4a14=100;
    end
end
if e4a11_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*3;
    else
        tchegada_e4a14=100;
    end
end
if e4a10_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*4;
    else
        tchegada_e4a14=100;
    end
end
if e4a09_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*5;
    else
        tchegada_e4a14=100;
    end
end
if e4a08_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*6;
    else
        tchegada_e4a14=100;
    end
end
if e4a07_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*7;
    else
        tchegada_e4a14=100;
    end
end
if e4a06_ult==1
    if e4d==0 || e4d==
        tchegada_e4a14=2*8;
    else
        tchegada_e4a14=100;
    end
end
if e4a05_ult==1

```

```

    if e4d==0 || e4d==1
        tchegada_e4a14=2*9;
    else
        tchegada_e4a14=100;
    end
end
if e4a04_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*10;
    else
        tchegada_e4a14=100;
    end
end
if e4a03_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*11;
    else
        tchegada_e4a14=100;
    end
end
if e4a02_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*12;
    else
        tchegada_e4a14=100;
    end
end
if e4a01_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*13;
    else
        tchegada_e4a14=100;
    end
end
if e4a00_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*14;
    else
        tchegada_e4a14=100;
    end
end
if e4a01_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*15;
    else
        tchegada_e4a14=100;
    end
end
if e4a02_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a14=2*16;
    else
        tchegada_e4a14=100;
    end
end
end

end

if c13d==1
    %Se ainda nao tem ninguem alocado
    if c13d_aloc==0
        %
        %Seletor 1
        if e1a14_ult==1
            tchegada_e1a13=(1/4)*p14d+2*1;

        end
        if e1a13==1 && e1d==0
            tchegada_e1a13=0;
        end
        if e1a12_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a13=2*1;
            else
                tchegada_e1a13=100;
            end
        end
        if e1a11_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a13=2*2;
            else
                tchegada_e1a13=100;
            end
        end
        if e1a10_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a13=2*3;
            else
                tchegada_e1a13=100;
            end
        end
        if e1a09_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a13=2*4;
            end
        end
    end
end

```



```

    tchegada_e2a13=100;
end
end
if e2a11_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*2;
    else
        tchegada_e2a13=100;
    end
end
if e2a10_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*3;
    else
        tchegada_e2a13=100;
    end
end
if e2a09_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*4;
    else
        tchegada_e2a13=100;
    end
end
if e2a08_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*5;
    else
        tchegada_e2a13=100;
    end
end
if e2a07_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*6;
    else
        tchegada_e2a13=100;
    end
end
if e2a06_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*7;
    else
        tchegada_e2a13=100;
    end
end
if e2a05_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*8;
    else
        tchegada_e2a13=100;
    end
end
if e2a04_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*9;
    else
        tchegada_e2a13=100;
    end
end
if e2a03_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*10;
    else
        tchegada_e2a13=100;
    end
end
if e2a02_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*11;
    else
        tchegada_e2a13=100;
    end
end
if e2a01_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*12;
    else
        tchegada_e2a13=100;
    end
end
if e2a00_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*13;
    else
        tchegada_e2a13=100;
    end
end
if e2a001_ult==1
    if e2d==y || e2d==y
        tchegada_e2a13=2*14;
    else
        tchegada_e2a13=100;
    end
end
if e2a002_ult==1
    if e2d==y || e2d==y

```

```

    tchegada_e2a13=2*15;
else
    tchegada_e2a13=100;
end
end

%
%Subprograma 3
if e3a14_ult==1
    tchegada_e3a13=(1/4)*p14d+2*1;

end
if e3a13==1 && e3d==0
    tchegada_e3a13=0;
end
if e3a12_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*1;
    else
        tchegada_e3a13=100;
    end
end
if e3a11_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*2;
    else
        tchegada_e3a13=100;
    end
end
if e3a10_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*3;
    else
        tchegada_e3a13=100;
    end
end
if e3a09_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*4;
    else
        tchegada_e3a13=100;
    end
end
if e3a08_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*5;
    else
        tchegada_e3a13=100;
    end
end
if e3a07_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*6;
    else
        tchegada_e3a13=100;
    end
end
if e3a06_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*7;
    else
        tchegada_e3a13=100;
    end
end
if e3a05_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*8;
    else
        tchegada_e3a13=100;
    end
end
if e3a04_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*9;
    else
        tchegada_e3a13=100;
    end
end
if e3a03_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*10;
    else
        tchegada_e3a13=100;
    end
end
if e3a02_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*11;
    else
        tchegada_e3a13=100;
    end
end
if e3a01_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a13=2*12;
    end
end

```

```

else
    tchegada_e3a13=100;
end
end
if e3a00_ult==1
    if e3d==1 || e3d==2
        tchegada_e3a13=2*13;
    else
        tchegada_e3a13=100;
    end
end
if e3a01_ult==1
    if e3d==1 || e3d==2
        tchegada_e3a13=2*14;
    else
        tchegada_e3a13=100;
    end
end
if e3a02_ult==1
    if e3d==1 || e3d==2
        tchegada_e3a13=2*15;
    else
        tchegada_e3a13=100;
    end
end

%
% elevador 4
if e4a14_ult==1
    tchegada_e4a13=(1/4)*p14d+2*1;

end
if e4a13==1 && e4d==1
    tchegada_e4a13=0;
end
if e4a12_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*1;
    else
        tchegada_e4a13=100;
    end
end
if e4a11_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*2;
    else
        tchegada_e4a13=100;
    end
end
if e4a10_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*3;
    else
        tchegada_e4a13=100;
    end
end
if e4a09_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*4;
    else
        tchegada_e4a13=100;
    end
end
if e4a08_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*5;
    else
        tchegada_e4a13=100;
    end
end
if e4a07_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*6;
    else
        tchegada_e4a13=100;
    end
end
if e4a06_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*7;
    else
        tchegada_e4a13=100;
    end
end
if e4a05_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*8;
    else
        tchegada_e4a13=100;
    end
end
if e4a04_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a13=2*9;
    else

```

```

        tchegada_e4a13=100;
    end
end
if e4a03_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*10;
    else
        tchegada_e4a13=100;
    end
end
if e4a02_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*11;
    else
        tchegada_e4a13=100;
    end
end
if e4a01_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*12;
    else
        tchegada_e4a13=100;
    end
end
if e4a00_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*13;
    else
        tchegada_e4a13=100;
    end
end
if e4a00_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*14;
    else
        tchegada_e4a13=100;
    end
end
if e4a02_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a13=2*15;
    else
        tchegada_e4a13=100;
    end
end

end
end

if c12d==1
    %Sei ainda nao tem memoria alocada
    if c12d_aloc==0
        %Sei ainda nao tem memoria alocada
        %Sei ainda nao tem memoria alocada
        if e1a14_ult==1
            tchegada_e1a12=(1/4)*p14d+(1/4)*p13d+2*2;

        end
        if e1a13_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a12=(1/4)*p13d+2*1;
            else
                tchegada_e1a12=2*1+(1/4)*p14d+(1/4)*p13d+2*2;
            end
        end
        if e1a12==1 && e1d==0
            tchegada_e1a12=0;
        end
        if e1a11_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a12=2*1;
            else
                tchegada_e1a12=100;
            end
        end
        if e1a10_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a12=2*2;
            else
                tchegada_e1a12=100;
            end
        end
        if e1a09_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a12=2*3;
            else
                tchegada_e1a12=100;
            end
        end
        if e1a08_ult==1
            if e1d==0 || e1d==1
                tchegada_e1a12=2*4;
            end
        end
    end
end

```

```

    tchegada_e1a12=100;
end
end
if e1a07_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*5;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a06_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*6;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a05_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*7;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a04_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*8;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a03_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*9;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a02_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*10;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a01_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*11;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a00_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*12;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a00_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*13;
    else
        tchegada_e1a12=100;
    end
end
end
if e1a00_ult==1
    if e1d==1 || e1d==2
        tchegada_e1a12=2*14;
    else
        tchegada_e1a12=100;
    end
end
end

%-----
%servidor 2
if e2a14_ult==1
    tchegada_e2a12=(1/4)*p14d+(1/4)*p13d+2*2;
end
end
if e2a13_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=(1/4)*p13d+2*1;
    else
        tchegada_e2a12=2*1+(1/4)*p14d+(1/4)*p13d+2*2;
    end
end
end
if e2a12==1 && e2d==1
    tchegada_e2a12=0;
end
end
if e2a11_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*1;
    end
end
end

```

```

else
    tchegada_e2a12=100;
end
end
if e2a10_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*2;
    else
        tchegada_e2a12=100;
    end
end
if e2a09_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*3;
    else
        tchegada_e2a12=100;
    end
end
if e2a08_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*4;
    else
        tchegada_e2a12=100;
    end
end
if e2a07_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*5;
    else
        tchegada_e2a12=100;
    end
end
if e2a06_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*6;
    else
        tchegada_e2a12=100;
    end
end
if e2a05_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*7;
    else
        tchegada_e2a12=100;
    end
end
if e2a04_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*8;
    else
        tchegada_e2a12=100;
    end
end
if e2a03_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*9;
    else
        tchegada_e2a12=100;
    end
end
if e2a02_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*10;
    else
        tchegada_e2a12=100;
    end
end
if e2a01_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*11;
    else
        tchegada_e2a12=100;
    end
end
if e2a00_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*12;
    else
        tchegada_e2a12=100;
    end
end
if e2am01_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*13;
    else
        tchegada_e2a12=100;
    end
end
if e2am02_ult==1
    if e2d==1 || e2d==2
        tchegada_e2a12=2*14;
    else
        tchegada_e2a12=100;
    end
end
end

```

```

%-----
%Subroutines
if e3a14_ult==1
    tchegada_e3a12=(1/4)*p14d+(1/4)*p13d+2*2;

end

if e3a13_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=(1/4)*p13d+2*1;
    else
        tchegada_e3a12=2*1+(1/4)*p14d+(1/4)*p13d+2*2;
    end
end

if e3a12==1 && e3d==0
    tchegada_e3a12=0;
end

if e3a11_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*1;
    else
        tchegada_e3a12=100;
    end
end

if e3a10_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*2;
    else
        tchegada_e3a12=100;
    end
end

if e3a09_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*3;
    else
        tchegada_e3a12=100;
    end
end

if e3a08_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*4;
    else
        tchegada_e3a12=100;
    end
end

if e3a07_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*5;
    else
        tchegada_e3a12=100;
    end
end

if e3a06_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*6;
    else
        tchegada_e3a12=100;
    end
end

if e3a05_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*7;
    else
        tchegada_e3a12=100;
    end
end

if e3a04_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*8;
    else
        tchegada_e3a12=100;
    end
end

if e3a03_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*9;
    else
        tchegada_e3a12=100;
    end
end

if e3a02_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*10;
    else
        tchegada_e3a12=100;
    end
end

if e3a01_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*11;
    else
        tchegada_e3a12=100;
    end
end
end

```

```

if e3a00_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*12;
    else
        tchegada_e3a12=100;
    end
end
if e3a01_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*13;
    else
        tchegada_e3a12=100;
    end
end
if e3a02_ult==1
    if e3d==0 || e3d==1
        tchegada_e3a12=2*14;
    else
        tchegada_e3a12=100;
    end
end

%
% Elevador 4
if e4a14_ult==1
    tchegada_e4a12=(1/4)*p14d+(1/4)*p13d+2*2;
end
if e4a13_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=(1/4)*p13d+2*1;
    else
        tchegada_e4a12=2*1+(1/4)*p14d+(1/4)*p13d+2*2;
    end
end

if e4a12==1 && e4d==0
    tchegada_e4a12=0;
end

if e4a11_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*1;
    else
        tchegada_e4a12=100;
    end
end
if e4a10_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*2;
    else
        tchegada_e4a12=100;
    end
end
if e4a09_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*3;
    else
        tchegada_e4a12=100;
    end
end
if e4a08_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*4;
    else
        tchegada_e4a12=100;
    end
end
if e4a07_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*5;
    else
        tchegada_e4a12=100;
    end
end
if e4a06_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*6;
    else
        tchegada_e4a12=100;
    end
end
if e4a05_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*7;
    else
        tchegada_e4a12=100;
    end
end
if e4a04_ult==1
    if e4d==0 || e4d==1
        tchegada_e4a12=2*8;
    else
        tchegada_e4a12=100;
    end
end

```

```

end
if e4a03_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*9;
    else
        tchegada_e4a12=100;
    end
end
if e4a02_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*10;
    else
        tchegada_e4a12=100;
    end
end
if e4a01_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*11;
    else
        tchegada_e4a12=100;
    end
end
if e4a00_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*12;
    else
        tchegada_e4a12=100;
    end
end
if e4a01_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*13;
    else
        tchegada_e4a12=100;
    end
end
if e4a02_ult==1
    if e4d==1 || e4d==2
        tchegada_e4a12=2*14;
    else
        tchegada_e4a12=100;
    end
end
end
end

```

```

%*Calculo do tempo de espera de cada chamado
%*****

```

```

if c14d==1 && c14d_aloc==0
    tespera_a14d=etime(clock,tc14d_ini);
end
if c13d==1 && c13d_aloc==0
    tespera_a13d=etime(clock,tc13d_ini);
end
if c12d==1 && c12d_aloc==0
    tespera_a12d=etime(clock,tc12d_ini);
end
end

```

```

%*FUZZIFICACAO, INFERENCIA E DESFUZZIFICACAO
%*****

```

```

if c14d==1 && c14d_aloc==0
    prioridade_a14d=evalfis([tchegada_e1a14 tespera_a14d;tchegada_e2a14 tespera_a14d;tchegada_e3a14 tespera_a14d;tchegada_e4a14 tespera_a14d],elevfuzzy);
end
if c13d==1 && c13d_aloc==0
    prioridade_a13d=evalfis([tchegada_e1a13 tespera_a13d;tchegada_e2a13 tespera_a13d;tchegada_e3a13 tespera_a13d;tchegada_e4a13 tespera_a13d],elevfuzzy);
end
if c12d==1 && c12d_aloc==0
    prioridade_a12d=evalfis([tchegada_e1a12 tespera_a12d;tchegada_e2a12 tespera_a12d;tchegada_e3a12 tespera_a12d;tchegada_e4a12 tespera_a12d],elevfuzzy);
end
end

```

```

%*Alocacao dos Elevadores aos Andares
%*****

```

```

auxiliar_14=300;
auxiliar_13=300;
auxiliar_12=300;

mdex_14=0;
mdex_13=0;
index_12=0;
index_11=0;

```

```

chamados=0;
chamado14=0;
chamado13=0;
chamado12=0;

```

1.24

```

if c14d==1 && (c14d_aloc==0 || p14d>12)
    chamados=chamados+1;
    chamado14=1;
end
if c13d==1 && (c13d_aloc==0 || p13d>12)
    chamados=chamados+1;
    chamado13=1;
end
if c12d==1 && (c12d_aloc==0 || p12d>12)
    chamados=chamados+1;
    chamado12=1;
end

if chamados==1
    if chamado14==1
        prioridades_e1=14;
        prioridades_e2=14;
        prioridades_e3=14;
        prioridades_e4=14;
        if e1c14d==0 && e1c13d==0 && e1c12d==0
            auxiliar_14=tchegada_e1a14;
            index_14=1;
        end
        if tchegada_e2a14 < auxiliar_14
            if e2c14d==0 && e2c13d==0 && e2c12d==0
                index_14=2;
                auxiliar_14=tchegada_e2a14;
            end
        end
        if tchegada_e3a14 < auxiliar_14
            if e3c14d==0 && e3c13d==0 && e3c12d==0
                index_14=3;
                auxiliar_14=tchegada_e3a14;
            end
        end
        if tchegada_e4a14 < auxiliar_14
            if e4c14d==0 && e4c13d==0 && e4c12d==0
                index_14=4;
            end
        end
        if index_14==1
            e1c14d=1;
            c14d_aloc=c14d_aloc+capacidade-e1p;
        end
        if index_14==2 && e2p==0
            e2c14d=1;
            c14d_aloc=c14d_aloc+capacidade-e2p;
        end
        if index_14==3
            e3c14d=1;
            c14d_aloc=c14d_aloc+capacidade-e3p;
        end
        if index_14==4 && e3c14d==0
            e4c14d=1;
            c14d_aloc=c14d_aloc+capacidade-e4p;
        end
    end
    if chamado13==1
        prioridades_e1=13;
        prioridades_e2=13;
        prioridades_e3=13;
        prioridades_e4=13;
        if e1c14d==0 && e1c13d==0 && e1c12d==0
            auxiliar_13=tchegada_e1a13;
            index_13=1;
        end
        if tchegada_e2a13 < auxiliar_13
            if e2c14d==0 && e2c13d==0 && e2c12d==0
                index_13=2;
                auxiliar_13=tchegada_e2a13;
            end
        end
        if tchegada_e3a13 < auxiliar_13
            if e3c14d==0 && e3c13d==0 && e3c12d==0
                index_13=3;
                auxiliar_13=tchegada_e3a13;
            end
        end
        if tchegada_e4a13 < auxiliar_13
            if e4c14d==0 && e4c13d==0 && e4c12d==0
                index_13=4;
            end
        end
        if index_13==1
            e1c13d=1;
            c13d_aloc=c13d_aloc+capacidade-e1p;

```

```

end
if index_13==2
    e2c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e2p;
end
if index_13==3
    e3c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e3p;
end
if index_13==4 && e3c13d==0
    e4c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e4p;
end

end

if chamado12==1
    prioridades_e1=12;
    prioridades_e2=12;
    prioridades_e3=12;
    prioridades_e4=12;
    if e1c14d==0 && e1c13d==0 && e1c12d==0
        auxiliar_12=tchegada_e1a12;
        index_12=1;
    end
    if tchegada_e2a12 < auxiliar_12
        if e2c14d==0 && e2c13d==0 && e2c12d==0
            index_12=2;
            auxiliar_12=tchegada_e2a12;
        end
        if tchegada_e3a12 < auxiliar_12
            if e3c14d==0 && e3c13d==0 && e3c12d==0
                index_12=3;
                auxiliar_12=tchegada_e3a12;
            end
            if tchegada_e4a12 < auxiliar_12
                if e4c14d==0 && e4c13d==0 && e4c12d==0
                    index_12=4;
                end
            end
        end

        if index_12==1
            e1c12d=1;
            c12d_aloc=c12d_aloc+capacidade-e1p;
        end
        if index_12==2
            e2c12d=1;
            c12d_aloc=c12d_aloc+capacidade-e2p;
        end
        if index_12==3
            e3c12d=1;
            c12d_aloc=c12d_aloc+capacidade-e3p;
        end
        if index_12==4
            e4c12d=1;
            c12d_aloc=c12d_aloc+capacidade-e4p;
        end
    end

end

if chamados==2
    if chamado14==1 && chamado13==1
        if prioridade_a14d(1) >= prioridade_a13d(1)
            prioridades_e1=[14 13];
        else
            prioridades_e1=[13 14];
        end
        if prioridade_a14d(2) >= prioridade_a13d(2)
            prioridades_e2=[14 13];
        else
            prioridades_e2=[13 14];
        end
        if prioridade_a14d(3) >= prioridade_a13d(3)
            prioridades_e3=[14 13];
        else
            prioridades_e3=[13 14];
        end
        if prioridade_a14d(4) >= prioridade_a13d(4)
            prioridades_e4=[14 13];
        else
            prioridades_e4=[13 14];
        end
    end
end

if chamado14==1 && chamado12==1
    if prioridade_a14d(1) >= prioridade_a12d(1)
        prioridades_e1=[14 12];
    else
        prioridades_e1=[12 14];
    end
    if prioridade_a14d(2) >= prioridade_a12d(2)
        prioridades_e2=[14 12];
    else

```

```

    prioridades_e2=[12 14];
end
if prioridade_a14d(3) >= prioridade_a12d(3)
    prioridades_e3=[14 12];
else
    prioridades_e3=[12 14];
end
if prioridade_a14d(4) >= prioridade_a12d(4)
    prioridades_e4=[14 12];
else
    prioridades_e4=[12 14];
end
end

if chamado13==1 && chamado12==1
    if prioridade_a13d(1) >= prioridade_a12d(1)
        prioridades_e1=[13 12];
    else
        prioridades_e1=[12 13];
    end
    if prioridade_a13d(2) >= prioridade_a12d(2)
        prioridades_e2=[13 12];
    else
        prioridades_e2=[12 13];
    end
    if prioridade_a13d(3) >= prioridade_a12d(3)
        prioridades_e3=[13 12];
    else
        prioridades_e3=[12 13];
    end
    if prioridade_a13d(4) >= prioridade_a12d(4)
        prioridades_e4=[13 12];
    else
        prioridades_e4=[12 13];
    end
end
%verifica se houve empate e aloca um melhor
%variavel 14
if chamado14==1
    if prioridades_e1(1)==14
        if e1c12d==0 && e1c13d==0 && e1c14d==0
            auxiliar_14=tchegada_e1a14;
            index_14=1;
        end
    end
    if prioridades_e2(1)==14
        if tchegada_e2a14 < auxiliar_14
            if e2c12d==0 && e2c13d==0 && e2c14d==0
                auxiliar_14=tchegada_e2a14;
                index_14=2;
            end
        end
    end
    if prioridades_e3(1)==14
        if tchegada_e3a14 < auxiliar_14
            if e3c12d==0 && e3c13d==0 && e3c14d==0
                auxiliar_14=tchegada_e3a14;
                index_14=3;
            end
        end
    end
    if prioridades_e4(1)==14
        if tchegada_e4a14 < auxiliar_14
            if e4c12d==0 && e4c13d==0 && e4c14d==0
                auxiliar_14=tchegada_e4a14;
                index_14=4;
            end
        end
    end
end

if index_14==1
    e1c14d=1;
    c14d_aloc=c14d_aloc+capacidade-e1p;
end
if index_14==2 && e2p==0
    e2c14d=1;
    c14d_aloc=c14d_aloc+capacidade-e2p;
end
if index_14==3
    e3c14d=1;
    c14d_aloc=c14d_aloc+capacidade-e3p;
end
if index_14==4 && e3c14d==0
    e4c14d=1;
    c14d_aloc=c14d_aloc+capacidade-e4p;
end

end

if chamado13==1
    if (prioridades_e1(1)==13 || prioridades_e1(2)==13)
        if e1c14d==0 && e1c13d==0 && e1c12d==0
            auxiliar_13=tchegada_e1a13;
            index_13=1;
        end
    end

```

```

end
if (prioridades_e2(1)==13 || prioridades_e2(2)==13)
    if e2c14d==0 && e2c13d==0 && e2c12d==0
        if tchegada_e2a13 < auxiliar_13
            auxiliar_13=tchegada_e2a13;
            index_13=2;
        end
    end
end
if (prioridades_e3(1)==13 || prioridades_e3(2)==13)
    if e3c14d==0 && e3c13d==0 && e3c12d==0
        if tchegada_e3a13 < auxiliar_13
            auxiliar_13=tchegada_e3a13;
            index_13=3;
        end
    end
end
if (prioridades_e4(1)==13 || prioridades_e4(2)==13)
    if e4c14d==0 && e4c13d==0 && e4c12d==0
        if tchegada_e4a13 < auxiliar_13
            auxiliar_13=tchegada_e4a13;
            index_13=4;
        end
    end
end

if index_13==1
    e1c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e1p;
end
if index_13==2
    e2c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e2p;
end
if index_13==3
    e3c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e3p;
end
if index_13==4
    e4c13d=1;
    c13d_aloc=c13d_aloc+capacidade-e4p;
end

chamado12=1
if (prioridades_e1(1)==12 || prioridades_e1(2)==12 || prioridades_e1(3)==12)
    if (e1c14d==0 && e1c13d==0)
        auxiliar_12=tchegada_e1a12;
        index_12=1;
    end
end
if (prioridades_e2(1)==12 || prioridades_e2(2)==12 || prioridades_e2(3)==12)
    if (e2c14d==0 && e2c13d==0)
        if tchegada_e2a12 < auxiliar_12
            auxiliar_12=tchegada_e2a12;
            index_12=2;
        end
    end
end
if (prioridades_e3(1)==12 || prioridades_e3(2)==12 || prioridades_e3(3)==12)
    if (e3c14d==0 && e3c13d==0)
        if tchegada_e3a12 < auxiliar_12
            auxiliar_12=tchegada_e3a12;
            index_12=3;
        end
    end
end
if (prioridades_e4(1)==12 || prioridades_e4(2)==12 || prioridades_e4(3)==12)
    if (e4c14d==0 && e4c13d==0)
        if tchegada_e4a12 < auxiliar_12
            auxiliar_12=tchegada_e4a12;
            index_12=4;
        end
    end
end

if index_12==1
    e1c12d=1;
    c12d_aloc=c12d_aloc+capacidade-e1p;
end
if index_12==2
    e2c12d=1;
    c12d_aloc=c12d_aloc+capacidade-e2p;
end
if index_12==3
    e3c12d=1;
    c12d_aloc=c12d_aloc+capacidade-e3p;
end
if index_12==4
    e4c12d=1;
    c12d_aloc=c12d_aloc+capacidade-e4p;
end
end

```

```

end

%*****
%*****
%CONTROLADOR LOCAL - ELEVADOR 01

%Entrada e Saida de Pessoas

%Chegando no Andar Menos02
if e1am02==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e1bm02==1
        %reseta o botao
        e1bm02=0;
        %coloca o elevador em estado parado- saindo gente
        e1=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        tels=timer('startDelay',e1p,'TimerFcn','elp=0,pe1m02=0,e1=996');
        start(tels);
    else
        %se o botao do referido andar nao estiver pressionado, entao nao
        %vai sair gente e pode colocar ok a saida de pessoas
        e1=996;
    end
    %se ja tiverem saido as pessoas do elevador que queiram descer no
    %andar
    if e1==996
        %como trataremos apenas pico de descida
        e1=998;
    end

    %DEFINICAO DA DIRECAO DO ELEVADOR
    %A priori, deixar como 'Neutra' a direcao e depois verifica se deve
    %ser de subida ou descida
    eld=0;
    if e1bm01==1 || e1b00==1 || e1b01==1 || e1b02==1 || e1b03==1 || e1b04==1 || e1b05==1 || e1b06==1
        eld=1;
    end
    if e1b07==1 || e1b08==1 || e1b09==1 || e1b10==1 || e1b11==1 || e1b12==1 || e1b13==1 || e1b14==1
        eld=-1;
    end
    if e1cm01s==1 || e1cm01d==1 || e1c00s==1 || e1c00d==1 || e1c01s==1 || e1c01d==1 || e1c02s==1 || e1c02d==1
        eld=1;
    end
    if e1c03s==1 || e1c03d==1 || e1c04s==1 || e1c04d==1 || e1c05s==1 || e1c05d==1 || e1c06s==1 || e1c06d==1
        eld=-1;
    end
    if e1c07s==1 || e1c07d==1 || e1c08s==1 || e1c08d==1 || e1c09s==1 || e1c09d==1 || e1c10s==1 || e1c10d==1
        eld=-1;
    end
    if e1c11s==1 || e1c11d==1 || e1c12s==1 || e1c12d==1 || e1c13s==1 || e1c13d==1 || e1c14s==1 || e1c14d==1
        eld=-1;
    end
end

end

%Chegando no Andar Menos01
if e1am01==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e1bm01==1
        %reseta o botao
        e1bm01=0;
        %coloca o elevador em estado parado- saindo gente
        e1=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        tels=timer('startDelay',pe1m01,'TimerFcn','elp=elp-pe1m01,pe1m01=0,e1=996');
        start(tels);
    else
        %se o botao do referido andar nao estiver pressionado, entao nao
        %vai sair gente e pode colocar ok a saida de pessoas
        e1=996;
    end
    %se ja tiverem saido as pessoas do elevador que queiram descer no
    %andar
    if e1==996
        %Se houver uma chamada do andar alocada para o elevador 1
        %dependendo se o elevador estiver de subida ou de descida
        if eld==1
            %caso esteja para subir, verifica se ha alocao de chamada
            %para subida
            e1=998;
        else
            %caso esteja para descer, verifica se ha alocao de chamada
            %para descida

```

```

    el=998;
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
eld=0;
if elb00==1 || elb01==1 || elb02==1 || elb03==1 || elb04==1 || elb05==1 || elb06==1
    eld=0;
end
if elb07==1 || elb08==1 || elb09==1 || elb10==1 || elb11==1 || elb12==1 || elb13==1 || elb14==1
    eld=0;
end
if elc00==1 || elc00d==1 || elc01s==1 || elc01d==1 || elc02s==1 || elc02d==1
    eld=0;
end
if elc03s==1 || elc03d==1 || elc04s==1 || elc04d==1 || elc05s==1 || elc05d==1 || elc06s==1 || elc06d==1
    eld=0;
end
if elc07s==1 || elc07d==1 || elc08s==1 || elc08d==1 || elc09s==1 || elc09d==1 || elc10s==1 || elc10d==1
    eld=0;
end
if elc11s==1 || elc11d==1 || elc12s==1 || elc12d==1 || elc13s==1 || elc13d==1 || elc14s==1 || elc14d==1
    eld=0;
end
if elbm02==1
    eld=0;
end
if elcm02s==1
    eld=0;
end
end

%Chegando no Andar 00 (terreo)
if ela00==1
    %Se o botao deste andar do elevador estiver pressionado
    if elb00==1
        %Retorna o botao
        elb00=0;
        %Coloca o elevador em estado parado, saindo gente
        el=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        tcls=timer('startDelete',pe100,TimerFcn,'(t-pe100,pe100-0,el=996);');
        start(tcls);
    else
        if el==995
            %Se o botao do referido andar nao estiver pressionado, entao nao
            %vao sair gente e pode colocar vk a saida de pessoas
            el=996;
        end
    end
    %Se ja tiverem saído as pessoas do elevador que queiram descer no
    %andar
    if el==996
        %Se houver uma chamada do andar alocado para o elevador 1,
        %determinando se o elevador estiver de subida ou de descida
        if eld==0
            el=998;
        end
        if eld==0 || eld==1
            el=998;
        end
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
eld=0;
if elb01==1 || elb02==1 || elb03==1 || elb04==1 || elb05==1 || elb06==1
    eld=0;
end
if elb07==1 || elb08==1 || elb09==1 || elb10==1 || elb11==1 || elb12==1 || elb13==1 || elb14==1
    eld=0;
end
if elc01s==1 || elc01d==1 || elc02s==1 || elc02d==1
    eld=0;
end
if elc03s==1 || elc03d==1 || elc04s==1 || elc04d==1 || elc05s==1 || elc05d==1 || elc06s==1 || elc06d==1
    eld=0;
end
if elc07s==1 || elc07d==1 || elc08s==1 || elc08d==1 || elc09s==1 || elc09d==1 || elc10s==1 || elc10d==1
    eld=0;
end
if elc11s==1 || elc11d==1 || elc12s==1 || elc12d==1 || elc13s==1 || elc13d==1 || elc14s==1 || elc14d==1
    eld=0;
end
if elbm01==1 || elbm02==1
    eld=0;
end
end
if elcm01s==1 || elcm01d==1 || elcm02s==1

```

```
    eld=0;  
end  
end
```

```
%Chegando no Andar 01
```

```
if ela01==1  
    el=998;  
end
```

```
%Chegando no Andar 02
```

```
if ela02==1  
    el=998;  
end
```

```
%Chegando no Andar 03
```

```
if ela03==1  
    el=998;  
end
```

```
%Chegando no Andar 04
```

```
if ela04==1  
    el=998;  
end
```

```
%Chegando no Andar 05
```

```
if ela05==1  
    el=998;  
end
```

```
%Chegando no Andar 06
```

```
if ela06==1  
    el=998;  
end
```

```
%Chegando no Andar 07
```

```
if ela07==1  
    el=998;  
end
```

```
%Chegando no Andar 08
```

```
if ela08==1  
    el=998;  
end
```

```
%Chegando no Andar 09
```

```
if ela09==1  
    el=998;  
end
```

```
%Chegando no Andar 10
```

```
if ela10==1  
    el=998;  
end
```

```
%Chegando no Andar 11
```

```
if ela11==1  
    el=998;  
end
```

```

end

%Chegando no Andar 12

if e1a12==1
    e1=996;

    if e1==996

        %Se houver uma chamada do andar alocada para o elevador 1,
        %dependendo se o elevador estiver de subida ou de descida
        if e1d==
            e1=998;

        end

        if e1d==1 || e1d==if
            %Ao estar para descer, verifica se ha alocação de chamada
            %para descida
            if e1c12d==1
                %reseta a alocação de chamada
                e1c12d=0;
                %coloca o elevador em estado de parado e entrando gente
                e1=997;
                %calcula o numero de pessoas que entraro no elevador, que
                %no caso eh o minimo entre a capacidade do elevador menos
                %as pessoas que estao dentro dele, e o numero de pessoas
                %aguardando para entrar
                e1p_entr=min(capacidade-e1p,p12d);
                %calcula um delta T para a entrada das pessoas, sendo 1s por
                %pessoa e depois desse tempo faz update do numero de
                %pessoas no elevador e no andar, e coloca o elevador em
                %situacao parado e ok para subir ou descer
                tele=timer('start12d',e1p_entr,'TimerFcn',@to=e1p+e1p_entr,p12d-p12d-e1p_entr-998);
                start(tele);
                %Se nao houver outro elevador alocado pra esse andar,
                %coloca como zero o numero de pessoas aguardando jah
                %alocadas para algum elevador

                if e2c12d==0 && e3c12d==0 && e4c12d==0
                    c12d_aloc=0;
                end

                %Enquanto houver pessoas que vao entrar no elevador,
                %verifica pra qual andar eles vao atraves de seu numero
                %ex. 0 para andar menos 02, 1 para menos 01, 2 para
                %terreo, 3 para 1o andar, etc), apertando o botao e
                %aumentando a variavel de pessoas querendo ir para esse
                %andar
                while e1p_entr>0
                    ult_p12d=ult_p12d+1;
                    if filap12d(ult_p12d)==0
                        e1bm02=1;
                        pe1m02=pe1m02+1;
                    end
                    if filap12d(ult_p12d)==1
                        e1bm01=1;
                        pe1m01=pe1m01+1;
                    end
                    if filap12d(ult_p12d)==2
                        e1bm00=1;
                        pe1m00=pe1m00+1;
                    end
                end

                %reduz em 1 o numero de pessoas que vao entrar no
                %elevador, apos este ter sido incluido no elevador
                e1p_entr=e1p_entr-1;
            end

            %Se nao tiver mais ninguem querendo subir no elevador na
            %direcao especificada, entao zera o botao de chamado do
            %apartamento e contabiliza o tempo que esta ficou ativado,
            %sendo um tempo cumulativo

            if p12d==0
                c12d=0;
                tc12d_fim=clock;
                ttot_c12d=ttot_c12d+etime(tc12d_fim,tc12d_ini);
            end
        else
            e1=998;
        end
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e1d= ;

if e1b13==1 || e1b14==1
    e1d=;
end

```

```

if e1c13s==1 || e1c13d==1 || e1c14s==1 || e1c14d==1
    eld=0;
end
if e1b06==1 || e1b05==1 || e1b04==1 || e1b03==1 || e1b02==1 || e1b01==1 || e1b00==1 || e1bm01==1 || e1bm02==1
    eld=0;
end
if e1b11==1 || e1b10==1 || e1b09==1 || e1b08==1 || e1b07==1
    eld=0;
end
if e1c00s==1 || e1c00d==1 || e1c01s==1 || e1c01d==1 || e1c02s==1 || e1c02d==1
    eld=0;
end
if e1c03s==1 || e1c03d==1 || e1c04s==1 || e1c04d==1 || e1cm01s==1 || e1cm01d==1 || e1cm02s==1
    eld=0;
end
if e1c05s==1 || e1c05d==1 || e1c06s==1 || e1c06d==1 || e1c07s==1 || e1c07d==1 || e1c08s==1 || e1c08d==1
    eld=0;
end
if e1c09s==1 || e1c09d==1 || e1c10s==1 || e1c10d==1 || e1c11s==1 || e1c11d==1
    eld=0;
end
end
end

```

**Chegando no Andar 13*

```

if e1a13==1
    %Se o botão deste andar do elevador 1 estiver pressionado...
    if e1b13==1
        %Reseta o botão
        e1b13=0;
        %Coloca o elevador em estado parado, saindo gente
        e1=995;
        %Coloca um delay T para a saída das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do número de pessoas
        %no elevador, e coloca elevador em situação parado e com saída
        %de pessoas finalizada
        tel=timer('StartDelay',pe113,'TimerFcn','tpe=lp-pe113,pe113-0,e1=996');
        start(tel);
    else
        if e1==997
            %Se o botão do referido andar não estiver pressionado, então não
            %vai sair gente e pode colocar ok a saída de pessoas
            e1=996;
        end
    end
end
%Se já tiverem saído as pessoas do elevador que querem descer no
%andar
if e1==996
    %Se houver uma chamada do andar alocada para o elevador 1,
    %dependendo se o elevador estiver de subida ou de descida
    if eld==0
        e1=998;
    end
end

if eld==0 || eld==1
    %Caso esteja para descer, verifica se há alocação de chamada
    %para descida
    if e1c13d==1
        %Reseta a alocação de chamada
        e1c13d=0;
        %Coloca o elevador em estado de parado e entrando gente
        e1=997;
        %Calcula o número de pessoas que entrarão no elevador, que
        %no caso é o mínimo entre a capacidade do elevador menos
        %as pessoas que estão dentro dele, e o número de pessoas
        %aguardando para entrar
        e1p_entr=min(capacidade-e1p,p13d);
        %Coloca um delay T para a entrada das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do número de
        %pessoas no elevador e no andar, e coloca o elevador em
        %situação parado e ok para subir ou descer
        tele=timer('StartDelay',12,'TimerFcn','t1=998');
        start(tele);
        e1p=e1p+e1p_entr,p13d=p13d-e1p_entr;
        %Se não houver outro elevador alocado pra esse andar,
        %coloca como zero o número de pessoas aguardando já
        %alocadas para alguns elevadores

        %if e2c13d==0 && e3c13d==0 && e4c13d==0
        %    c13d_aloc=0;
        %end

        %Enquanto houver pessoas que vão entrar no elevador,
        %verifica pra qual andar eles vão através de seu número
        %ex: 0 para andar menos 02, 1 para menos 01, 2 para
        %terreo, 3 para 1o andar, etc), apertando o botão e
        %aumentando a variável de pessoas querendo ir para esse
        %andar
        while e1p_entr>0
            ult_p13d=ult_p13d+1;
            if filap13d(ult_p13d)==0
                e1bm02=1;
                pe1m02=pe1m02+1;
            end
        end
    end
end

```

```

end
if filap13d(ult_p13d)==1
    e1bm01=1;
    pe1m01=pe1m01+1;
end
if filap13d(ult_p13d)==2
    e1b00=1;
    pe100=pe100+1;
end

%senda em 1 o numero de pessoas que vao entrar no
%elevador, apos este ter sido incluido no elevador
elp_ent=elp_ent+1;
end

%Se nao tiver mais ninguem querendo subir no elevador na
%direcao especificada, entao zera o botao de chamado do
%apartamento e contabiliza o tempo que este ficou ativado,
%sendo um tempo cumulativo

if p13d==0
    c13d=0;
    tc13d_fim=clock;
    ttot_c13d=ttot_c13d+etime(tc13d_fim,tc13d_in);
end
else
    if e1==997
        e1=998;
    end
end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como ?Nenhuma direcao e depois verifica se deve
%subir ou descer
eld='';

if e1b14==1
    eld='';
end
if e1c14s==1 || e1c14d==1
    eld='';
end
if e1b06==1 || e1b05==1 || e1b04==1 || e1b03==1 || e1b02==1 || e1b01==1 || e1b00==1 || e1bm01==1 || e1bm02==1
    eld='';
end
if e1b12==1 || e1b11==1 || e1b10==1 || e1b09==1 || e1b08==1 || e1b07==1
    eld='';
end
if e1c00s==1 || e1c00d==1 || e1c01s==1 || e1c01d==1 || e1c02s==1 || e1c02d==1
    eld='';
end
if e1c03s==1 || e1c03d==1 || e1c04s==1 || e1c04d==1 || e1cm01s==1 || e1cm01d==1 || e1cm02s==1
    eld='';
end
if e1c05s==1 || e1c05d==1 || e1c06s==1 || e1c06d==1 || e1c07s==1 || e1c07d==1 || e1c08s==1 || e1c08d==1
    eld='';
end
if e1c09s==1 || e1c09d==1 || e1c10s==1 || e1c10d==1 || e1c11s==1 || e1c11d==1 || e1c12s==1 || e1c12d==1
    eld='';
end
end

end

%Chegando no Andar 14
if e1a14==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e1b14==1
        %reseta o botao
        e1b14=0;
        %coloca o elevador em estado parado - sendo gente
        e1=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizada
        telt=timer('StartDelay',elp,'TimerFcn','t=0,pe114=0,e1=996');
        start(telt);
    else
        if e1==997
            %Se o botao do referido andar nao estiver pressionado, entao nao
            %vai sair gente e pode colocar ok a saida de pessoas
            e1=996;
        end
    end
end

%se ja tiverem saido as pessoas do elevador que queiram descer no
%andar
if e1==996
    %Se houver uma chamada do andar alocada para o elevador 1
    if e1c14d==1
        %reseta a alocao de chamada
        e1c14d=0;
    end
end

```

```

%coloca o elevador em estado de parado e entrando gente
el=997;
%calcula o numero de pessoas que entraro no elevador, que
%no caso eh o minimo entre a capacidade do elevador menos
%as pessoas que estao dentro dele, e o numero de pessoas
%aguardando para entrar
elp_entr=min(capacidade-el.p14d);
%coloca um delay T para a entrada das pessoas, sendo Ts por
%pessoa, e depois deve tempo faz update do numero de
%pessoas no elevador e no andar e coloca o elevador em
%estado de parado e ts para subir ou descer
tele=timer('startDelay',elp_entr,'timerFcn',@el=998);
start(tele);
elp=elp+elp_entr; p14d=p14d-elp_entr;
%se nao houver outro elevador alocado pra esse andar,
%coloca como zero o numero de pessoas aguardando sub
%alocadas para algum elevador

%if el14d==0 && el314d==0 && el514d==0
%   el14d_aloc=0;
%end

%Enquanto houver pessoas que vao entrar no elevador,
%verifica pra qual andar eles vao atraves de seu numero
%ex: 0 para andar menos 02, 1 para menos 01, 2 para
%aterreo, 3 para 1o andar etc) apertando o botao e
%aumentando a variavel de pessoas querendo ir para esse
%andar
while elp_entr>0
    ult_p14d=ult_p14d+1;
    if filap14d(ult_p14d)==1
        elbm01=1;
        pelm01=pelm01+1;
    end
    if filap14d(ult_p14d)==2
        elb00=1;
        pel00=pel00+1;
    end
    if filap14d(ult_p14d)==0
        elbm02=1;
        pelm02=pelm02+1;
    end
    %adiaz em 1 o numero de pessoas que vao entrar no
    %elevador, apos este ter sido incluido no elevador
    elp_entr=elp_entr-1;
end

%se nao tiver mais ninguem querendo subir no elevador na
%direcao especificada, criaa zero o botao de chamado do
%apartamento e contabiliza o tempo que este ficou ativado,
%sendo um tempo cumulativo

if p14d==0
    c14d=0;
    tc14d_fim=clock;
    ttot_c14d=ttot_c14d+etime(tc14d_fim,tc14d_in);
end
else
    if el==997
        el=998;
    end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ir de subida ou descida
eld=0;
if elbm01==1 || elb00==1 || elb01==1 || elb02==1 || elb03==1 || elb04==1 || elb05==1 || elb06==1
    eld=1;
end
if elb07==1 || elb08==1 || elb09==1 || elb10==1 || elb11==1 || elb12==1 || elb13==1 || elbm02==1
    eld=-1;
end
if elcm01s==1 || elcm01d==1 || elc00s==1 || elc00d==1 || elc01s==1 || elc01d==1 || elc02s==1 || elc02d==1
    eld=0;
end
if elc03s==1 || elc03d==1 || elc04s==1 || elc04d==1 || elc05s==1 || elc05d==1 || elc06s==1 || elc06d==1
    eld=0;
end
if elc07s==1 || elc07d==1 || elc08s==1 || elc08d==1 || elc09s==1 || elc09d==1 || elc10s==1 || elc10d==1
    eld=0;
end
if elc11s==1 || elc11d==1 || elc12s==1 || elc12d==1 || elc13s==1 || elc13d==1 || elcm02s==1 || elcm02d==1
    eld=0;
end

end

%
%
%Como nao haja comando para subida, verifica se deve subir o elevador
if eld==0
    if elcm02==1 && el==998
        elcm02=0;
    end
end

```

```

*tempo de 2s para chegar a outro andar
telm02m01=timer(StartDelay,2,TimerFcn',elam01=1,elam02 ult=0,elam01 ult=1);
start(tel02m01);
end
if elam01==1 && el==998
    elam01=0;
    *tempo de 2s para chegar a outro andar
    telm0100=timer(StartDelay,2,TimerFcn',ela00=1,elam01 ult=0,ela00 ult=1);
    start(tel0100);
end
if ela00==1 && el==998
    ela00=0;
    *tempo de 2s para chegar a outro andar
    tel0001=timer(StartDelay,2,TimerFcn',ela01=1,ela00 ult=0,ela01 ult=1);
    start(tel0001);
end
if ela01==1 && el==998
    ela01=0;
    *tempo de 2s para chegar a outro andar
    tel0102=timer(StartDelay,2,TimerFcn',la02=1,ela01 ult=0,ela02 ult=1);
    start(tel0102);
end
if ela02==1 && el==998
    ela02=0;
    *tempo de 2s para chegar a outro andar
    tel0203=timer(StartDelay,2,TimerFcn',la03=1,ela02 ult=0,ela03 ult=1);
    start(tel0203);
end
if ela03==1 && el==998
    ela03=0;
    *tempo de 2s para chegar a outro andar
    tel0304=timer(StartDelay,2,TimerFcn',ela04=1,ela03 ult=0,ela04 ult=1);
    start(tel0304);
end
if ela04==1 && el==998
    ela04=0;
    *tempo de 2s para chegar a outro andar
    tel0405=timer(StartDelay,2,TimerFcn',ela05=1,ela04 ult=0,ela05 ult=1);
    start(tel0405);
end
if ela05==1 && el==998
    ela05=0;
    *tempo de 2s para chegar a outro andar
    tel0506=timer(StartDelay,2,TimerFcn',ela06=1,ela05 ult=0,ela06 ult=1);
    start(tel0506);
end
if ela06==1 && el==998
    ela06=0;
    *tempo de 2s para chegar a outro andar
    tel0607=timer(StartDelay,2,TimerFcn',ela07=1,ela06 ult=0,ela07 ult=1);
    start(tel0607);
end
if ela07==1 && el==998
    ela07=0;
    *tempo de 2s para chegar a outro andar
    tel0708=timer(StartDelay,2,TimerFcn',ela08=1,ela07 ult=0,ela08 ult=1);
    start(tel0708);
end
if ela08==1 && el==998
    ela08=0;
    *tempo de 2s para chegar a outro andar
    tel0809=timer(StartDelay,2,TimerFcn',ela09=1,ela08 ult=0,ela09 ult=1);
    start(tel0809);
end
if ela09==1 && el==998
    ela09=0;
    *tempo de 2s para chegar a outro andar
    tel0910=timer(StartDelay,2,TimerFcn',ela10=1,ela09 ult=0,ela10 ult=1);
    start(tel0910);
end
if ela10==1 && el==998
    ela10=0;
    *tempo de 2s para chegar a outro andar
    tel1011=timer(StartDelay,2,TimerFcn',ela11=1,ela10 ult=0,ela11 ult=1);
    start(tel1011);
end
if ela11==1 && el==998
    ela11=0;
    *tempo de 2s para chegar a outro andar
    tel1112=timer(StartDelay,2,TimerFcn',ela12=1,ela11 ult=0,ela12 ult=1);
    start(tel1112);
end
if ela12==1 && el==998
    ela12=0;
    *tempo de 2s para chegar a outro andar
    tel1213=timer(StartDelay,2,TimerFcn',ela13=1,ela12 ult=0,ela13 ult=1);
    start(tel1213);
end
if ela13==1 && el==998
    ela13=0;
    *tempo de 2s para chegar a outro andar
    tel1314=timer(StartDelay,2,TimerFcn',ela14=1,ela13 ult=0,ela14 ult=1);
    start(tel1314);
end
end
end

```

%Como háo comando para Jascida, verifica se deve descer o elevador

```

if eld==1
if elal4==1 && el==998
    elal4=0;
    %tempo de 2s para chegar a outro andar
    tel1413=timer( 'startDelay',2,'TimerFcn',elal3=1,elal4=0,elal3=0,elal4=1 );
    start(tel1413);
end
if elal3==1 && el==998
    elal3=0;
    %tempo de 2s para chegar a outro andar
    tel1312=timer( 'startDelay',2,'TimerFcn',elal2=1,elal3=0,elal2=0,elal3=1 );
    start(tel1312);
end
if elal2==1 && el==998
    elal2=0;
    %tempo de 2s para chegar a outro andar
    tel1211=timer( 'startDelay',2,'TimerFcn',elal1=1,elal2=0,elal1=0,elal2=1 );
    start(tel1211);
end
if elal1==1 && el==998
    elal1=0;
    %tempo de 2s para chegar a outro andar
    tel1110=timer( 'startDelay',2,'TimerFcn',elal0=1,elal1=0,elal0=0,elal1=1 );
    start(tel1110);
end
if elal0==1 && el==998
    elal0=0;
    %tempo de 2s para chegar a outro andar
    tel1009=timer( 'startDelay',2,'TimerFcn',ela09=1,elal0=0,ela09=0,elal0=1 );
    start(tel1009);
end
if ela09==1 && el==998
    ela09=0;
    %tempo de 2s para chegar a outro andar
    tel0908=timer( 'startDelay',2,'TimerFcn',ela08=1,ela09=0,ela08=0,ela09=1 );
    start(tel0908);
end
if ela08==1 && el==998
    ela08=0;
    %tempo de 2s para chegar a outro andar
    tel0807=timer( 'startDelay',2,'TimerFcn',ela07=1,ela08=0,ela07=0,ela08=1 );
    start(tel0807);
end
if ela07==1 && el==998
    ela07=0;
    %tempo de 2s para chegar a outro andar
    tel0706=timer( 'startDelay',2,'TimerFcn',ela06=1,ela07=0,ela06=0,ela07=1 );
    start(tel0706);
end
if ela06==1 && el==998
    ela06=0;
    %tempo de 2s para chegar a outro andar
    tel0605=timer( 'startDelay',2,'TimerFcn',ela05=1,ela06=0,ela05=0,ela06=1 );
    start(tel0605);
end
if ela05==1 && el==998
    ela05=0;
    %tempo de 2s para chegar a outro andar
    tel0504=timer( 'startDelay',2,'TimerFcn',ela04=1,ela05=0,ela04=0,ela05=1 );
    start(tel0504);
end
if ela04==1 && el==998
    ela04=0;
    %tempo de 2s para chegar a outro andar
    tel0403=timer( 'startDelay',2,'TimerFcn',ela03=1,ela04=0,ela03=0,ela04=1 );
    start(tel0403);
end
if ela03==1 && el==998
    ela03=0;
    %tempo de 2s para chegar a outro andar
    tel0302=timer( 'startDelay',2,'TimerFcn',ela02=1,ela03=0,ela02=0,ela03=1 );
    start(tel0302);
end
if ela02==1 && el==998
    ela02=0;
    %tempo de 2s para chegar a outro andar
    tel0201=timer( 'startDelay',2,'TimerFcn',ela01=1,ela02=0,ela01=0,ela02=1 );
    start(tel0201);
end
if ela01==1 && el==998
    ela01=0;
    %tempo de 2s para chegar a outro andar
    tel0100=timer( 'startDelay',2,'TimerFcn',ela00=1,ela01=0,ela00=0,ela01=1 );
    start(tel0100);
end
if ela00==1 && el==998
    ela00=0;
    %tempo de 2s para chegar a outro andar
    tel00m01=timer( 'startDelay',2,'TimerFcn',elam01=1,ela00=0,elam01=0,ela00=1 );
    start(tel00m01);
end
if elam01==1 && el==998
    elam01=0;
    %tempo de 2s para chegar a outro andar

```

```

te1m01m02=timer(StartDelay,2,TimerFcn,'e1am02=1; e1am01 ult=0; e1am02 ult=1;');
start(te1m01m02);
end
end

%*****
%*****
%CONTROLADOR LOCAL - ELEVADOR 02

%Entrada e Saida de Pessoas

%Chegando no Andar Menos02
if e2am02==1
    %Se o botao deste andar do elevador 1 estiver pressionado...
    if e2bm02==1
        %reseta o botao
        e2bm02=0;
        %coloca o elevador em estado parado, saindo gente
        e2=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te2s=timer(StartDelay,e2p,TimerFcn,'e2o=0;pe2m02=0;e2=996');
        start(te2s);
    else
        %se o botao do referido andar nao estiver pressionado, entao nao
        %vai sair gente e pode colocar ok a saida de pessoas
        e2=996;
    end
    %se ja tiverem saido as pessoas do elevador que queriam descer no
    %andar
    if e2==996
        %Como trataremos apenas puxo de descida
        e2=998;
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e2d=0;
if e2bm01==1 || e2b00==1 || e2b01==1 || e2b02==1 || e2b03==1 || e2b04==1 || e2b05==1 || e2b06==1
    e2d=1;
end
if e2b07==1 || e2b08==1 || e2b09==1 || e2b10==1 || e2b11==1 || e2b12==1 || e2b13==1 || e2b14==1
    e2d=-1;
end
if e2cm01s==1 || e2cm01d==1 || e2c00s==1 || e2c00d==1 || e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
    e2d=0;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1
    e2d=0;
end
if e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1 || e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1
    e2d=0;
end
if e2c11s==1 || e2c11d==1 || e2c12s==1 || e2c12d==1 || e2c13s==1 || e2c13d==1 || e2c14s==1 || e2c14d==1
    e2d=0;
end
end

end

%Chegando no Andar Menos01
if e2am01==1
    %Se o botao deste andar do elevador 1 estiver pressionado...
    if e2bm01==1
        %reseta o botao
        e2bm01=0;
        %coloca o elevador em estado parado, saindo gente
        e2=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te2s=timer(StartDelay,pe2m01,TimerFcn,'e2p=e2p-pe2m01;pe2m01=0;e2=996');
        start(te2s);
    else
        if e1==995
            %se o botao do referido andar nao estiver pressionado, entao nao
            %vai sair gente e pode colocar ok a saida de pessoas
            e2=996;
        end
    end
    %se ja tiverem saido as pessoas do elevador que queriam descer no
    %andar
    if e2==996
        %Se houver uma chamada do andar alocada para o elevador 1,
        %dependendo se o elevador estiver de subida ou de descida
        if e2d==1
            %caso esteja para subir, verifica se ha alocao de chamada
            %para subida
            e2=998;
        end
        if e2d==0 || e2d=-1

```

```

%senso encia para descer, verifica se ha alocação de chamada
%para descida

e2=998;
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e2d=0;
if e2b00==1 || e2b01==1 || e2b02==1 || e2b03==1 || e2b04==1 || e2b05==1 || e2b06==1
    e2d=1;
end
if e2b07==1 || e2b08==1 || e2b09==1 || e2b10==1 || e2b11==1 || e2b12==1 || e2b13==1 || e2b14==1
    e2d=-1;
end
if e2c00s==1 || e2c00d==1 || e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
    e2d=0;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1
    e2d=0;
end
if e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1 || e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1
    e2d=0;
end
if e2c11s==1 || e2c11d==1 || e2c12s==1 || e2c12d==1 || e2c13s==1 || e2c13d==1 || e2c14s==1 || e2c14d==1
    e2d=0;
end
if e2bm02==1
    e2d=0;
end
if e2cm02s==1
    e2d=0;
end
end

%Chegando no Andar 00 (terreo)
if e2a00==1
    %Se o botao deste andar do elevador estiver pressionado
    if e2b00==1
        %pressa o botao
        e2b00=0;
        %coloca o elevador em estado parado, sendo gente
        e2=995;
        %coloca um delta T para a saída das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saída
        %de pessoas ja finalizado
        te2s=timer('StartDelay',pe200,'TimerFcn','2p=e2p-pe200,pe200=0,e2=996');
        start(te2s);
    else
        if e2=995
            %Se o botao do referido andar nao estiver pressionado, entao nao
            %vai sair gente e pode colocar ok a saída de pessoas
            e2=996;
        end
    end
end
%se ja tiverem saído as pessoas do elevador que queriam descer no
%semalar
if e2=996
    %Se houver uma chamada do andar alocada para o elevador 1,
    %dependendo se o elevador estiver de subida ou de descida
    if e2d==1
        e2=998;
    end

    if e2d==0 || e2d=-1
        e2=998;
    end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e2d=0;
if e2b01==1 || e2b02==1 || e2b03==1 || e2b04==1 || e2b05==1 || e2b06==1
    e2d=1;
end
if e2b07==1 || e2b08==1 || e2b09==1 || e2b10==1 || e2b11==1 || e2b12==1 || e2b13==1 || e2b14==1
    e2d=-1;
end
if e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
    e2d=0;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1
    e2d=0;
end
if e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1 || e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1
    e2d=0;
end
if e2c11s==1 || e2c11d==1 || e2c12s==1 || e2c12d==1 || e2c13s==1 || e2c13d==1 || e2c14s==1 || e2c14d==1
    e2d=0;
end
if e2bm01==1 || e2bm02==1
    e2d=0;
end

```

```
end
if e2cm01s==1 || e2cm01d==1 || e2cm02s==1
    e2d=1;
end
end

%Chegando no Andar 01
if e2a01==1
    e2=998;
end

%Chegando no Andar 02
if e2a02==1
    e2=998;
end

%Chegando no Andar 03
if e2a03==1
    e2=998;
end

%Chegando no Andar 04
if e2a04==1
    e2=998;
end

%Chegando no Andar 05
if e2a05==1
    e2=998;
end

%Chegando no Andar 06
if e2a06==1
    e2=998;
end

%Chegando no Andar 07
if e2a07==1
    e2=998;
end

%Chegando no Andar 08
if e2a08==1
    e2=998;
end

%Chegando no Andar 09
if e2a09==1
    e2=998;
end

%Chegando no Andar 10
if e2a10==1
    e2=998;
end

%Chegando no Andar 11
if e2a11==1
```

```

e2=998;

end

%Chegando no Andar 12

if e2a12==1
    e2=996;

    if e2==996

        %Se houver uma chamada do andar alocada para o elevador 1,
        %dependendo se o elevador estiver de subida ou de descida
        if e2d==1
            e2=998;

        end

        if e2d==1 || e2d==0
            %caso esteja para descer, verifica se ha alocao de chamada
            %para descida
            if e2c12d==1
                %reseta a alocao de chamada
                e2c12d=0;
                %coloca o elevador em estado de parado e entrando gente
                e2=997;
                %calcula o numero de pessoas que entraro no elevador, que
                %no caso eh o maximo entre a capacidade do elevador menos
                %as pessoas que estao dentro dele, e o numero de pessoas
                %aguardando para entrar
                e2p_entr=min(capacidade-e2p,p12d)
                %calcula um delta T para a entrada das pessoas, sendo 1s por
                %pessoa, e depois desse tempo faz update do numero de
                %pessoas no elevador e no andar, e coloca o elevador em
                %situacao parado e ok para subir ou descer
                te2=timer('start','clav,e2p_entr','TimerFcn',e2o=e2o+e2o entr,p12d=p12d-e2o entr,e2=998);
                start(te2);

                %Se nao houver outro elevador alocado pra esse andar,
                %coloca como zero o numero de pessoas aguardando ja
                %alocadas para algum elevador

                if e2c12d==0 && e3c12d==0 && e4c12d==0
                    c12d_aloc=0;

                end

                %Enquanto houver pessoas que vao entrar no elevador,
                %verifica pra qual andar elas vao atraves de seu numero
                %ex: 0 para andar menos 02, 1 para menos 01, 2 para
                %terreo, 3 para 1o andar, etc), apertando o botao e
                %acrescentando a variavel de pessoas querendo ir para esse
                %andar
                while e2p_entr>0
                    ult_p12d=ult_p12d+1;
                    if filap12d(ult_p12d)==0
                        e2bm02=1;
                        pe2m02=pe2m02+1;
                    end
                    if filap12d(ult_p12d)==1
                        e2bm01=1;
                        pe2m01=pe2m01+1;
                    end
                    if filap12d(ult_p12d)==2
                        e2b00=1;
                        pe200=pe200+1;
                    end

                    %reduz em 1 o numero de pessoas que vao entrar no
                    %elevador, apos este ter sido incluido no elevador
                    e2p_entr=e2p_entr-1;
                end

                %Se nao tiver mais ninguem querendo subir no elevador na
                %direcao especificada, entao zera o botao de chamado do
                %apartamento e contabiliza o tempo que este ficou ativado,
                %sendo um tempo cumulativo

                if p12d==0
                    c12d=0;
                    tc12d_fim=clock;
                    ttot_c12d=ttot_c12d+etime(tc12d_fim,tc12d_ini);
                end
            else
                e2=997
                e2=998;

            end
        end
    end
end
end

```

%DEFINICAO DA DIRECAO DO ELEVADOR

%A priori, deixar como Nextra a direcao e depois verifica se deve
%ser de subida ou descida

```

e2d=0;

if e2b13==1 || e2b14==1
    e2d=1;
end
if e2c13s==1 || e2c13d==1 || e2c14s==1 || e2c14d==1
    e2d=1;
end
if e2b06==1 || e2b05==1 || e2b04==1 || e2b03==1 || e2b02==1 || e2b01==1 || e2b00==1 || e2bm01==1 || e2bm02==1
    e2d=1;
end
if e2b11==1 || e2b10==1 || e2b09==1 || e2b08==1 || e2b07==1
    e2d=1;
end
if e2c00s==1 || e2c00d==1 || e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
    e2d=1;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2cm01s==1 || e2cm01d==1 || e2cm02s==1
    e2d=1;
end
if e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1 || e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1
    e2d=1;
end
if e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1 || e2c11s==1 || e2c11d==1
    e2d=1;
end
end

```

```

end

```

```

%Chegando no Andar 13

```

```

if e2a13==1
    %Se o botão deste andar do elevador 1 estiver pressionado
    if e2b13==1
        %Reseta o botão
        e2b13=0;
        %coloca o elevador em estado parado - saindo gente
        e2=995;
        %coloca um delta T para a saída das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situação parado e com saída
        %de pessoas ja finalizado
        te2s=timer('StartDelta',pe213,'TimerFcn',20-e2b-ne213-ne213+0,e2=996.);
        start(te2s);
    else
        if e2==997
            %Se o botão do referido andar não estiver pressionado, então não
            %vem sair gente e pode colocar ok a saída de pessoas
            e2=996;
        end
    end
end
%Se já tiverem saído as pessoas do elevador que queriam descer no
%andar
if e2==996
    %Se houver uma chamada do andar aloca-la para o elevador 1.
    %dependendo se o elevador estiver de subida ou de descida
    if e2d==1
        e2=998;
    end

    if e2d==0 || e2d==1
        %caso esteja para descer, verifica se há alocação de chamada
        %para descida
        if e2c13d==1
            %Reseta a alocação de chamada
            e2c13d=0;
            %coloca o elevador em estado de parado e entrando gente
            e2=997;
            %calcula o numero de pessoas que entraram no elevador, que
            %se caso ch o numero entre a capacidade do elevador menos
            %as pessoas que estão dentro dele, e o numero de pessoas
            %aguardando para entrar
            e2p_entr=min(capacidade-e2p,p13d)
            %coloca um delta T para a entrada das pessoas, sendo 1s por
            %pessoa, e depois desse tempo faz update do numero de
            %pessoas no elevador e no andar, e coloca o elevador em
            %situação parado e ok para subir ou descer
            te2e=timer('StartDelta',e2p_entr,'TimerFcn',2=998.);
            start(te2e);
            e2p=e2p+e2p_entr;p13d=p13d-e2p_entr;
            %Se não houver outro elevador alocado pra esse andar,
            %coloca como zero o numero de pessoas aguardando jah
            %alocadas para algum elevador

            %if e2c13d==0 && e2c13d==0 && e2c13d==0
            %e_c13d_aloc=0;
            %end

            %Enquanto houver pessoas que vão entrar no elevador,
            %verifica pra qual andar eles vão através de seu numero
            %ex: 0 para andar menos 02, 1 para menos 01, 2 para
            %atraso, 3 para 1o andar etc), apertando o botão e
            %aumentando a variavel de pessoas querendo ir para esse
            %andar

```

```

while e2p_entr>0
    ult_p13d=ult_p13d+1
    if filap13d(ult_p13d)==0
        e2bm02=1;
        pe2m02=pe2m02+1;
    end
    if filap13d(ult_p13d)==1
        e2bm01=1;
        pe2m01=pe2m01+1;
    end
    if filap13d(ult_p13d)==2
        e2b00=1;
        pe200=pe200+1;
    end

    %reduz em 1 o numero de pessoas que vao entrar no
    %elevador, apos este ter sido incluido no elevador
    e2p_entr=e2p_entr-1
end

%Se nao tiver mais ninguem querendo subir no elevador na
%direcao especifica da, entao zera o botao de chamado do
% pavimento e contabiliza o tempo que este ficou ativado,
%sendo um tempo cumulativo

if p13d==0
    c13d=0;
    tc13d_fim=clock;
    ttot_c13d=ttot_c13d+etime(tc13d_fim,tc13d_ini);
end

else
    if e2==997
        e2=998;
    end
end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%Aprioi, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e2d=0;

if e2b14==1
    e2d=1;
end
if e2c14s==1 || e2c14d==1
    e2d=1;
end
if e2b06==1 || e2b05==1 || e2b04==1 || e2b03==1 || e2b02==1 || e2b01==1 || e2b00==1 || e2bm01==1 || e2bm02==1
    e2d=1;
end
if e2b12==1 || e2b11==1 || e2b10==1 || e2b09==1 || e2b08==1 || e2b07==1
    e2d=1;
end
if e2c00s==1 || e2c00d==1 || e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
    e2d=1;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2cm01s==1 || e2cm01d==1 || e2cm02s==1
    e2d=1;
end
if e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1 || e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1
    e2d=1;
end
if e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1 || e2c11s==1 || e2c11d==1 || e2c12s==1 || e2c12d==1
    e2d=1;
end

end

%Chegando no Andar 14
if e2a14==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e2b14==1
        %seta o botao
        e2b14=0;
        %coloca o elevador em estado parado, saindo gente
        e2=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te2s=timer('startDelta',e2p,'TimerFcn','e2p=0;pe214=0;e2=996;');
        start(te2s);
    else
        if e2==997
            %Se o botao do referido andar nao estiver pressionado, entao nao
            %vai sair gente e pode colocar ok a saida de pessoas
            e2=996;
        end
    end
end

%Se ja tiverem saído as pessoas do elevador que queriam descer no

```

```

%andar
if e2==996
%Se houver uma chamada do andar alocada para o elevador 1
if e2c14d==1
%Envia a alocacao de chamada
e2c14d=0;
%coloca o elevador em estado de parado e entrando gente
e2=997;
%calcula o numero de pessoas que entraro no elevador, que
%no caso eh o numero entre a capacidade do elevador menos
%as pessoas que estao dentro dele, e o numero de pessoas
%aguardando para entrar
e2p_entr=min(capacidade-e2p,p14d)
%coloca um delta T para a entrada das pessoas, sendo Ts por
%pessoa, e depois desse tempo faz update do numero de
%pessoas no elevador e no andar, e coloca o elevador em
%estado parado e ok para subir ou descer
te2e=timer('StartDelay',e2p_entr,'TimerFcn','e2-998');
start(te2e);
e2p=e2p+e2p_entr;p14d=p14d-e2p_entr;
%Se nao houver outro elevador alocado pra esse andar,
%coloca como zero o numero de pessoas aguardando sub
%alocadas para algum elevador

%if e2c14d==0 && e2c14d==0 && e2c14d==0
% e2c14d_aloc=0;
%end

%Enquanto houver pessoas que vao entrar no elevador,
%verifica pra qual andar eles vao atraves de seu numero
%ex. 0 para andar menos 02, 1 para menos 01, 2 para
%terreo, 3 para 1o andar, etc), apertando o botao e
%aumentando a variavel de pessoas querendo ir para esse
%andar
while e2p_entr>0
ult_p14d=ult_p14d+1
if filap14d(ult_p14d)==1
e2bm01=1;
pe2m01=pe2m01+1;
end
if filap14d(ult_p14d)==2
e2b00=1;
pe200=pe200+1;
end
if filap14d(ult_p14d)==0
e2bm02=1;
pe2m02=pe2m02+1;
end
%reduz em 1 o numero de pessoas que vao entrar no
%elevador, apos este ter sido incluído no elevador
e2p_entr=e2p_entr-1;
end

%Se nao tiver mais ninguem querendo subir no elevador na
%direcao especificada, entao zera o botao de chamado do
%apartamento e contabiliza o tempo que esta ficou ativado,
%sendo um tempo cumulativo

if p14d==0
c14d=0;
tc14d_fim=clock;
ttot_c14d=ttot_c14d+etime(tc14d_fim,tc14d_ini);
end
else
if e2==997
e2=998;
end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e2d=;
if e2bm01==1 || e2b00==1 || e2b01==1 || e2b02==1 || e2b03==1 || e2b04==1 || e2b05==1 || e2b06==1
e2d=;
end
if e2b07==1 || e2b08==1 || e2b09==1 || e2b10==1 || e2b11==1 || e2b12==1 || e2b13==1 || e2bm02==1
e2d=;
end
if e2cm01==1 || e2cm01d==1 || e2c00s==1 || e2c00d==1 || e2c01s==1 || e2c01d==1 || e2c02s==1 || e2c02d==1
e2d=;
end
if e2c03s==1 || e2c03d==1 || e2c04s==1 || e2c04d==1 || e2c05s==1 || e2c05d==1 || e2c06s==1 || e2c06d==1
e2d=;
end
if e2c07s==1 || e2c07d==1 || e2c08s==1 || e2c08d==1 || e2c09s==1 || e2c09d==1 || e2c10s==1 || e2c10d==1
e2d=;
end
if e2c11s==1 || e2c11d==1 || e2c12s==1 || e2c12d==1 || e2c13s==1 || e2c13d==1 || e2cm02s==1 || e2cm02d==1
e2d=;
end
end
end

```

```

% Caso haja comando para subida, verifica se deve subir o elevador
if e2d==1
    if e2am02==1 && e2==998
        e2am02=0;
        % tempo de 2s para chegar a outro andar
        te2m02m01=timer(StartDelay,2,TimerFcn,e2am01=1,e2am02 ult=0,e2am01 ult=1);
        start(te2m02m01);
    end
    if e2am01==1 && e2==998
        e2am01=0;
        % tempo de 2s para chegar a outro andar
        te2m0100=timer(StartDelay,2,TimerFcn,e2a00=1,e2am01 ult=0,e2a00 ult=1);
        start(te2m0100);
    end
    if e2a00==1 && e2==998
        e2a00=0;
        % tempo de 2s para chegar a outro andar
        te20001=timer(StartDelay,2,TimerFcn,e2a01=1,e2a00 ult=0,e2a01 ult=1);
        start(te20001);
    end
    if e2a01==1 && e2==998
        e2a01=0;
        % tempo de 2s para chegar a outro andar
        te20102=timer(StartDelay,2,TimerFcn,e2a02=1,e2a01 ult=0,e2a02 ult=1);
        start(te20102);
    end
    if e2a02==1 && e2==998
        e2a02=0;
        % tempo de 2s para chegar a outro andar
        te20203=timer(StartDelay,2,TimerFcn,e2a03=1,e2a02 ult=0,e2a03 ult=1);
        start(te20203);
    end
    if e2a03==1 && e2==998
        e2a03=0;
        % tempo de 2s para chegar a outro andar
        te20304=timer(StartDelay,2,TimerFcn,e2a04=1,e2a03 ult=0,e2a04 ult=1);
        start(te20304);
    end
    if e2a04==1 && e2==998
        e2a04=0;
        % tempo de 2s para chegar a outro andar
        te20405=timer(StartDelay,2,TimerFcn,e2a05=1,e2a04 ult=0,e2a05 ult=1);
        start(te20405);
    end
    if e2a05==1 && e2==998
        e2a05=0;
        % tempo de 2s para chegar a outro andar
        te20506=timer(StartDelay,2,TimerFcn,e2a06=1,e2a05 ult=0,e2a06 ult=1);
        start(te20506);
    end
    if e2a06==1 && e2==998
        e2a06=0;
        % tempo de 2s para chegar a outro andar
        te20607=timer(StartDelay,2,TimerFcn,e2a07=1,e2a06 ult=0,e2a07 ult=1);
        start(te20607);
    end
    if e2a07==1 && e2==998
        e2a07=0;
        % tempo de 2s para chegar a outro andar
        te20708=timer(StartDelay,2,TimerFcn,e2a08=1,e2a07 ult=0,e2a08 ult=1);
        start(te20708);
    end
    if e2a08==1 && e2==998
        e2a08=0;
        % tempo de 2s para chegar a outro andar
        te20809=timer(StartDelay,2,TimerFcn,e2a09=1,e2a08 ult=0,e2a09 ult=1);
        start(te20809);
    end
    if e2a09==1 && e2==998
        e2a09=0;
        % tempo de 2s para chegar a outro andar
        te20910=timer(StartDelay,2,TimerFcn,e2a10=1,e2a09 ult=0,e2a10 ult=1);
        start(te20910);
    end
    if e2a10==1 && e2==998
        e2a10=0;
        % tempo de 2s para chegar a outro andar
        te21011=timer(StartDelay,2,TimerFcn,e2a11=1,e2a10 ult=0,e2a11 ult=1);
        start(te21011);
    end
    if e2a11==1 && e2==998
        e2a11=0;
        % tempo de 2s para chegar a outro andar
        te21112=timer(StartDelay,2,TimerFcn,e2a12=1,e2a11 ult=0,e2a12 ult=1);
        start(te21112);
    end
    if e2a12==1 && e2==998
        e2a12=0;
        % tempo de 2s para chegar a outro andar
        te21213=timer(StartDelay,2,TimerFcn,e2a13=1,e2a12 ult=0,e2a13 ult=1);
        start(te21213);
    end
    if e2a13==1 && e2==998
        e2a13=0;
    end
end

```

```

%tempo de 2s para chegar a outro andar
te21314=timer('StartDelay',2,'TimerFcn','e2a14=1; e2a13_ult=0; e2a14_ult=1;');
start(te21314);
end

end

%Caso haja comando para descer, verifica se deve descer o elevador
if e2d==1
    if e2a14==1 && e2==998
        e2a14=0;
        %tempo de 2s para chegar a outro andar
        te21413=timer('StartDelay',2,'TimerFcn','e2a13=1; e2a14_ult=0; e2a13_ult=1;');
        start(te21413);
    end
    if e2a13==1 && e2==998
        e2a13=0;
        %tempo de 2s para chegar a outro andar
        te21312=timer('StartDelay',2,'TimerFcn','e2a12=1; e2a13_ult=0; e2a12_ult=1;');
        start(te21312);
    end
    if e2a12==1 && e2==998
        e2a12=0;
        %tempo de 2s para chegar a outro andar
        te21211=timer('StartDelay',2,'TimerFcn','e2a11=1; e2a12_ult=0; e2a11_ult=1;');
        start(te21211);
    end
    if e2a11==1 && e2==998
        e2a11=0;
        %tempo de 2s para chegar a outro andar
        te21110=timer('StartDelay',2,'TimerFcn','e2a10=1; e2a11_ult=0; e2a10_ult=1;');
        start(te21110);
    end
    if e2a10==1 && e2==998
        e2a10=0;
        %tempo de 2s para chegar a outro andar
        te21009=timer('StartDelay',2,'TimerFcn','e2a09=1; e2a10_ult=0; e2a09_ult=1;');
        start(te21009);
    end
    if e2a09==1 && e2==998
        e2a09=0;
        %tempo de 2s para chegar a outro andar
        te20908=timer('StartDelay',2,'TimerFcn','e2a08=1; e2a09_ult=0; e2a08_ult=1;');
        start(te20908);
    end
    if e2a08==1 && e2==998
        e2a08=0;
        %tempo de 2s para chegar a outro andar
        te20807=timer('StartDelay',2,'TimerFcn','e2a07=1; e2a08_ult=0; e2a07_ult=1;');
        start(te20807);
    end
    if e2a07==1 && e2==998
        e2a07=0;
        %tempo de 2s para chegar a outro andar
        te20706=timer('StartDelay',2,'TimerFcn','e2a06=1; e2a07_ult=0; e2a06_ult=1;');
        start(te20706);
    end
    if e2a06==1 && e2==998
        e2a06=0;
        %tempo de 2s para chegar a outro andar
        te20605=timer('StartDelay',2,'TimerFcn','e2a05=1; e2a06_ult=0; e2a05_ult=1;');
        start(te20605);
    end
    if e2a05==1 && e2==998
        e2a05=0;
        %tempo de 2s para chegar a outro andar
        te20504=timer('StartDelay',2,'TimerFcn','e2a04=1; e2a05_ult=0; e2a04_ult=1;');
        start(te20504);
    end
    if e2a04==1 && e2==998
        e2a04=0;
        %tempo de 2s para chegar a outro andar
        te20403=timer('StartDelay',2,'TimerFcn','e2a03=1; e2a04_ult=0; e2a03_ult=1;');
        start(te20403);
    end
    if e2a03==1 && e2==998
        e2a03=0;
        %tempo de 2s para chegar a outro andar
        te20302=timer('StartDelay',2,'TimerFcn','e2a02=1; e2a03_ult=0; e2a02_ult=1;');
        start(te20302);
    end
    if e2a02==1 && e2==998
        e2a02=0;
        %tempo de 2s para chegar a outro andar
        te20201=timer('StartDelay',2,'TimerFcn','e2a01=1; e2a02_ult=0; e2a01_ult=1;');
        start(te20201);
    end
    if e2a01==1 && e2==998
        e2a01=0;
        %tempo de 2s para chegar a outro andar
        te20100=timer('StartDelay',2,'TimerFcn','e2a00=1; e2a01_ult=0; e2a00_ult=1;');
        start(te20100);
    end
    if e2a00==1 && e2==998
        e2a00=0;
        %tempo de 2s para chegar a outro andar

```

```

te200m01=timer(StartDelay,2,TimerFor,e2am01-1,e2a00_uh-0,e2am01_uh-1);
start(te200m01);
end
if e2am01==1 && e2==998
    e2am01=0;
    %tempo de 2s para chegar a outro andar
    te2m01m02=timer(StartDelay,2,TimerFor,e2am02-1,e2am01_uh-0,e2am02_uh-1.);
    start(te2m01m02);
end
end

%*****
%*****
%CONTROLADOR LOCAL - ELEVADOR 03

%Entrada e Saida de Pessoas

%Chegando no Andar Menos02
ife3am02==1
    %Se o botao deste andar do elevador 1 estiver pressionado...
    if e3bm02==1
        %pressa o botao
        e3bm02=0;
        %coloca o elevador em estado parado, saindo gente
        e3=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te3s=timer(StartDelay,e3p,TimerFor,e3p-0,pe3m02-0,e3-996);
        start(te3s);
    else
        %se o botao do referido andar nao estiver pressionado, entao nao
        %vai sair gente e pode colocar ok a saida de pessoas
        e3=996;
    end
    %se ja tiverem saido as pessoas do elevador que queiram descer no
    %andar
    if e3==996
        %Como trataremos apenas pico de descida
        e3=998;
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%subir de subida ou descida
e3d=0;
ife3bm01==1 || e3b00==1 || e3b01==1 || e3b02==1 || e3b03==1 || e3b04==1 || e3b05==1 || e3b06==1
    e3d=1;
end
ife3b07==1 || e3b08==1 || e3b09==1 || e3b10==1 || e3b11==1 || e3b12==1 || e3b13==1 || e3b14==1
    e3d=-1;
end
ife3cm01s==1 || e3cm01d==1 || e3c00s==1 || e3c00d==1 || e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d=0;
end
ife3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1
    e3d=0;
end
ife3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1 || e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1
    e3d=0;
end
ife3c11s==1 || e3c11d==1 || e3c12s==1 || e3c12d==1 || e3c13s==1 || e3c13d==1 || e3c14s==1 || e3c14d==1
    e3d=0;
end
end

end

%Chegando no Andar Menos01
ife3am01==1
    %Se o botao deste andar do elevador 1 estiver pressionado...
    if e3bm01==1
        %pressa o botao
        e3bm01=0;
        %coloca o elevador em estado parado, saindo gente
        e3=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te3s=timer(StartDelay,pe3m01,TimerFor,e3p-e3p-pe3m01,pe3m01-0,e3-996);
        start(te3s);
    else
        %se o botao do referido andar nao estiver pressionado, entao nao
        %vai sair gente e pode colocar ok a saida de pessoas
        e3=996;
    end
    %se ja tiverem saido as pessoas do elevador que queiram descer no
    %andar
    if e3==996
        %Se houver uma chamada do andar acima para o elevador 1,
        %dependendo se o elevador estiver de subida ou de descida
        if e3d==1
            %caso esteja para subir, verifica se ha alocao de chamada
            %para subida

```

```

e3=998;
end

if e3d==1 || e3d==2
    %caso esteja para descer, verifica se ha alocação de chamada
    %para descida

e3=998;
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e3d=0;
if e3b00==1 || e3b01==1 || e3b02==1 || e3b03==1 || e3b04==1 || e3b05==1 || e3b06==1
    e3d=1;
end
if e3b07==1 || e3b08==1 || e3b09==1 || e3b10==1 || e3b11==1 || e3b12==1 || e3b13==1 || e3b14==1
    e3d=-1;
end
if e3c00s==1 || e3c00d==1 || e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d=0;
end
if e3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1
    e3d=0;
end
if e3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1 || e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1
    e3d=0;
end
if e3c11s==1 || e3c11d==1 || e3c12s==1 || e3c12d==1 || e3c13s==1 || e3c13d==1 || e3c14s==1 || e3c14d==1
    e3d=0;
end
if e3bm02==1
    e3d=0;
end
if e3cm02s==1
    e3d=0;
end
end

%Chegando no Andar 00 (terreo)
if e3a00==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e3b00==1
        %reseta o botao
        e3b00=0;
        %coloca o elevador em estado parado, sendo gente
        e3=995;
        %coloca um delta T para a saída das pessoas, sendo 1s por
        %pessoa, e depois deve tempo faz update do numero de pessoas
        %no elevador, e coloca o elevador em situacao parado e com saída
        %de pessoas ja finalizado
        te3s=timer(StartDelay,pe300,TimerFor3p=e3p-pe300,pe300-0,e3-996,);
        start(te3s);
    else
        if e3==995
            %se o botao do referido andar nao estiver pressionado, entao nao
            %vai sair gente e pode colocar ok a saída de pessoas
            e3=996;
        end
    end
end
%se ja tiverem saído as pessoas do elevador que querem descer no
%andar
if e3==996
    %Se houver uma chamada do andar alocada para o elevador 1.
    %dependendo se o elevador estiver de subida ou de descida
    if e3d==1
        e3=998;
    end

    if e3d==1 || e3d==2
        e3=998;
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e3d=0;
if e3b01==1 || e3b02==1 || e3b03==1 || e3b04==1 || e3b05==1 || e3b06==1
    e3d=1;
end
if e3b07==1 || e3b08==1 || e3b09==1 || e3b10==1 || e3b11==1 || e3b12==1 || e3b13==1 || e3b14==1
    e3d=-1;
end
if e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d=0;
end
if e3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1
    e3d=0;
end
if e3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1 || e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1
    e3d=0;
end
if e3c11s==1 || e3c11d==1 || e3c12s==1 || e3c12d==1 || e3c13s==1 || e3c13d==1 || e3c14s==1 || e3c14d==1
    e3d=0;
end

```

```

        e3d= ;
    end
    if e3bm01==1 || e3bm02==1
        e3d= ;
    end
    if e3cm01s==1 || e3cm01d==1 || e3cm02s==1
        e3d= ;
    end
end

%Chegando no Andar 01
if e3a01==1

    e3=998;

end

%Chegando no Andar 02
if e3a02==1

    e3=998;

end

%Chegando no Andar 03
if e3a03==1

    e3=998;

end

%Chegando no Andar 04
if e3a04==1

    e3=998;

end

%Chegando no Andar 05
if e3a05==1

    e3=998;

end

%Chegando no Andar 06
if e3a06==1

    e3=998;

end

%Chegando no Andar 07
if e3a07==1

    e3=998;

end

%Chegando no Andar 08
if e3a08==1

    e3=998;

end

%Chegando no Andar 09
if e3a09==1

    e3=998;

end

%Chegando no Andar 10
if e3a10==1

    e3=998;

end

```

```

%Chegando no Andar 11
if e3a11==1
    e3=998;

end

%Chegando no Andar 12
if e3a12==1
    e3=996;

    if e3==996

        %Se houver uma chamada do andar alcada para o elevador 1.
        %dependendo se o elevador estiver de subida ou de descida
        if e3d==1
            e3=998;

        end

        if e3d==1 || e3d==0
            %caso esteja para descer, verifica se ha alocao de chamada
            %para descida
            if e3c12d==1
                %seta a alocao de chamada
                e3c12d=0;
                %coloca o elevador em estado de parado e entrando gente
                e3=997;
                %calcula o numero de pessoas que entram no elevador, que
                %no caso eh o minimo entre a capacidade do elevador menos
                %as pessoas que estao dentro dele, e o numero de pessoas
                %aguardando para entrar
                e3p_entr=min(capacidade-e3p,p12d)
                %coloca um delta t para a entrada das pessoas, sendo 1s por
                %pessoa, e depois desse tempo faz update do numero de
                %pessoas no elevador e no andar, e coloca o elevador em
                %status ao parado e ele para a subir ou descer
                te3e=timer('startDelay',e3p_entr,'TimeFcn',@e3=998);
                start(te3e);
                e3p=e3p+e3p_entr;p12d=p12d-e3p_entr;
                %Se nao houver outro elevador alocado pra esse andar,
                %coloca como zero o numero de pessoas aguardando ja
                %alocadas para algum elevador

                if e3c12d==0 && e3c12d==0 && e4c12d==0
                    c12d_aloc=0;
                end

                %Enquanto houver pessoas que vao entrar no elevador,
                %verifica pra qual andar eles vao atraves de seu numero
                %ex. 0 para andar menos 02, 1 para menos 01, 2 para
                %terceiro, 3 para 1o andar, etc), apertando o botao e
                %aumentando a variavel de pessoas querendo ir para esse
                %andar
                while e3p_entr>0
                    ult_p12d=ult_p12d+1;
                    if filap12d(ult_p12d)==0
                        e3bm02=1;
                        pe3m02=pe3m02+1;
                    end
                    if filap12d(ult_p12d)==1
                        e3bm01=1;
                        pe3m01=pe3m01+1;
                    end
                    if filap12d(ult_p12d)==2
                        e3bm00=1;
                        pe3m00=pe3m00+1;
                    end

                    %reduz em 1 o numero de pessoas que vao entrar no
                    %elevador, apos este ter sido incluído no elevador
                    e3p_entr=e3p_entr-1;
                end

                %Se nao tiver mais ninguem querendo subir no elevador na
                %direcao especificada, entao zera o botao de chamado do
                %apartamento e contabiliza o tempo que este ficou ativado,
                %sendo um tempo cumulativo

                if p12d==0
                    c12d=0;
                    tc12d_fim=clock;
                    ttot_c12d=ttot_c12d+etime(tc12d_fim,tc12d_ini);
                end
            else
                e3=998;
            end
        end
    end
end
end
end

```

%DEFINICAO DA DIRECAO DO ELEVADOR

```

%Se a priori, deixar como Neutra a direcao e depois verifica se deve
%seir de subida ou descida
e3d=0;

if e3b13==1 || e3b14==1
    e3d=1;
end
if e3c13s==1 || e3c13d==1 || e3c14s==1 || e3c14d==1
    e3d=1;
end
if e3b06==1 || e3b05==1 || e3b04==1 || e3b03==1 || e3b02==1 || e3b01==1 || e3b00==1 || e3bm01==1 || e3bm02==1
    e3d=1;
end
if e3b11==1 || e3b10==1 || e3b09==1 || e3b08==1 || e3b07==1
    e3d=1;
end
if e3c00s==1 || e3c00d==1 || e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d=1;
end
if e3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3cm01s==1 || e3cm01d==1 || e3cm02s==1
    e3d=1;
end
if e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1 || e3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1
    e3d=1;
end
if e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1 || e3c11s==1 || e3c11d==1
    e3d=1;
end
end

%Chegando no Andar 13
if e3a13==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e3b13==1
        %Reseta o botao
        e3b13=0;
        %coloca o elevador em estado parado, saindo gente
        e3=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizado
        te3e=timer('StartDelay',pe313,'TimerFcn','e3p=e3p-pe313;pe313=0,e3=996;');
        start(te3e);
    else
        if e3==997
            %se o botao do referido andar nao estiver pressionado, entao nao
            %vao sair gente e pode colocar ok a saida de pessoas
            e3=996;
        end
    end
end
%se ja tiverem sendo as pessoas do elevador que queriam descer no
%andar
if e3==996
    %se houver uma chamada do andar alocada para o elevador 1,
    %dependendo se o elevador estiver de subida ou de descida
    if e3d==1
        e3=998;
    end

    if e3d==0 || e3d==1
        %caso esteja para descer, verifica se ha alocao de chamada
        %para descida
        if e3c13d==1
            %reseta a alocao de chamada
            e3c13d=0;
            %coloca o elevador em estado de parado e entrando gente
            e3=997;
            %calcula o numero de pessoas que entraro no elevador, que
            %no caso eh o maximo entre a capacidade do elevador menos
            %as pessoas que estao dentro dele, e o numero de pessoas
            %aguardando para entrar
            e3p_entr=min(capacidade-e3p,p13d);
            %coloca um delta T para a entrada das pessoas, sendo 1s por
            %pessoa, e depois desse tempo faz update do numero de
            %pessoas no elevador e no andar, e coloca o elevador em
            %situacao parado e ok para subir ou descer
            te3e=timer('StartDelay',e3p_entr,'TimerFcn','3=998;');
            start(te3e);
            e3p=e3p+e3p_entr;p13d=p13d-e3p_entr;
            %se nao houver outro elevador alocado pra esse andar,
            %coloca como zero o numero de pessoas aguardando jah
            %alocadas para algum elevador

            %se e3c13d==0 && e3c13d==0 && e3c13d==0
            %e3c13d_aloc=0;
            %end

            %Enquanto houver pessoas que vao entrar no elevador,
            %verifica pra qual andar elas vao atraves de seu numero
            %ex. 0 para andar menos 02, 1 para menos 01, 2 para
            %menos 0, 3 para 01, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

```

```

%aumentando o variavel de pessoas querendo ir para esse
%andar
while e3p_entr>0
    ult_p13d=ult_p13d+1;
    if filap13d(ult_p13d)==0
        e3bm02=1;
        pe3m02=pe3m02+1;
    end
    if filap13d(ult_p13d)==1
        e3bm01=1;
        pe3m01=pe3m01+1;
    end
    if filap13d(ult_p13d)==2
        e3b00=1;
        pe300=pe300+1;
    end

    %reduz em 1 o numero de pessoas que vao entrar no
    %elevador, apos este ter sido incluido no elevador
    e3p_entr=e3p_entr-1;
end

%Se nao tiver mais ninguem querendo subir no elevador na
%direcao especificada, entao zera o botao de chamado do
%apartamento e calcula/da o tempo que este ficou ativado,
%sendo um tempo cumulativo

if p13d==0
    c13d=0;
    tc13d_fim=clock;
    ttot_c13d=ttot_c13d+etime(tc13d_fim,tc13d_ini);
end

else
    if e3==997
        e3=998;
    end
end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%a priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e3d='';

if e3b14==1
    e3d='';
end
if e3c14s==1 || e3c14d==1
    e3d='';
end
if e3b06==1 || e3b05==1 || e3b04==1 || e3b03==1 || e3b02==1 || e3b01==1 || e3b00==1 || e3bm01==1 || e3bm02==1
    e3d='';
end
if e3b12==1 || e3b11==1 || e3b10==1 || e3b09==1 || e3b08==1 || e3b07==1
    e3d='';
end
if e3c00s==1 || e3c00d==1 || e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d='';
end
if e3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3cm01s==1 || e3cm01d==1 || e3cm02s==1
    e3d='';
end
if e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1 || e3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1
    e3d='';
end
if e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1 || e3c11s==1 || e3c11d==1 || e3c12s==1 || e3c12d==1
    e3d='';
end
end

end

%Chegando no Andar 14
if e3a14==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e3b14==1
        %reseta o botao
        e3b14=0;
        %coloca o elevador em estado parado, sendo gente
        e3=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saida
        %de pessoas ja finalizada
        te3s=timer( StartDelay,e3p,TimerFcn,-3p-0,pe314-0,e3-996 );
        start(te3s);
    else
        if e3==997
            %se o botao do referido andar nao estiver pressionado, entao nao
            %vao sair gente e pode colocar ok a saida de pessoas
            e3=996;
        end
    end
end

```

```

end
%Se ja tiverem saído as pessoas do elevador que queiram descer no
%andar
if e3==996
    %Se houver uma chamada do andar alocado para o elevador
    if e3c14d==1
        %Verifica a alocação de chamada
        e3c14d=0;
        %Coloca o elevador em estado de parado e entrando gente
        e3=997;
        %Calcula o numero de pessoas que entraram no elevador, que
        %no caso eh o minimo entre a capacidade do elevador menos
        %as pessoas que estão dentro dele, e o numero de pessoas
        %aguardando para entrar
        e3p_entr=min(capacidade-e3p,p14d);
        %Coloca um delay T para a entrada das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de
        %pessoas no elevador e no andar, e coloca o elevador em
        %situacao parado e ok para subir ou descer
        te3=timer( 'StartDelay',e3p_entr,'TimerFcn',e3=998.);
        start(te3);
        e3p=e3p+e3p_entr;p14d=p14d-e3p_entr;
        %Se não houver outro elevador alocado pra esse andar,
        %coloca como zero o numero de pessoas aguardando sub
        %alocadas para algum elevador

        %if e3c14d==0 && e3c14d==0 && e3c14d==0
        %    c14d_aloc=0;
        %end

        %Enquanto houver pessoas que vão entrar no elevador,
        %verifica pra qual andar eles vão através de seu numero
        %ex: 0 para andar menos 02, 1 para menos 01, 2 para
        %aterco, 3 para 1o andar, etc), apertando o botão e
        %aumentando a variavel de pessoas querendo ir para esse
        %andar
        while e3p_entr>0
            ult_p14d=ult_p14d+1;
            if filap14d(ult_p14d)==1
                e3bm01=1;
                pe3m01=pe3m01+1;
            end
            if filap14d(ult_p14d)==2
                e3b00=1;
                pe300=pe300+1;
            end
            if filap14d(ult_p14d)==0
                e3bm02=1;
                pe3m02=pe3m02+1;
            end
            %atualiz em 1 o numero de pessoas que vão entrar no
            %elevador, apos este ter sido incluído no elevador
            e3p_entr=e3p_entr-1;
        end

        %Se não tiver mais ninguém querendo subir no elevador na
        %direção especificada, então zera o botão de chamada do
        %apartamento e contabiliza o tempo que este ficou ativado,
        %sendo um tempo cumulativo

        if p14d==0
            c14d=0;
            tc14d_fim=clock;
            ttot_c14d=ttot_c14d+etime(tc14d_fim,tc14d_ini);
        end
    else
        if e3==997
            e3=998;
        end
    end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e3d=0;
if e3bm01==1 || e3b00==1 || e3b01==1 || e3b02==1 || e3b03==1 || e3b04==1 || e3b05==1 || e3b06==1
    e3d=1;
end
if e3b07==1 || e3b08==1 || e3b09==1 || e3b10==1 || e3b11==1 || e3b12==1 || e3b13==1 || e3bm02==1
    e3d=-1;
end
if e3cm01==1 || e3cm01d==1 || e3c00s==1 || e3c00d==1 || e3c01s==1 || e3c01d==1 || e3c02s==1 || e3c02d==1
    e3d=-1;
end
if e3c03s==1 || e3c03d==1 || e3c04s==1 || e3c04d==1 || e3c05s==1 || e3c05d==1 || e3c06s==1 || e3c06d==1
    e3d=-1;
end
if e3c07s==1 || e3c07d==1 || e3c08s==1 || e3c08d==1 || e3c09s==1 || e3c09d==1 || e3c10s==1 || e3c10d==1
    e3d=-1;
end
if e3c11s==1 || e3c11d==1 || e3c12s==1 || e3c12d==1 || e3c13s==1 || e3c13d==1 || e3cm02s==1 || e3cm02d==1
    e3d=-1;
end
end

```

```

%Se caso haja comando para subida, verifica se deve subir o elevador
if e3d==1
    e3am02==1 && e3==998
    e3am02=0;
    %tempo de 2s para chegar a outro andar
    te3m02m01=timer(StartDelay,2,TimerFcn,e3am01=1,e3am02_ult=0,e3am01_ult=1);
    start(te3m02m01);
end
if e3am01==1 && e3==998
    e3am01=0;
    %tempo de 2s para chegar a outro andar
    te3m0100=timer(StartDelay,2,TimerFcn,e3a00=1,e3am01_ult=0,e3a00_ult=1);
    start(te3m0100);
end
if e3a00==1 && e3==998
    e3a00=0;
    %tempo de 2s para chegar a outro andar
    te30001=timer(StartDelay,2,TimerFcn,e3a01=1,e3a00_ult=0,e3a01_ult=1);
    start(te30001);
end
if e3a01==1 && e3==998
    e3a01=0;
    %tempo de 2s para chegar a outro andar
    te30102=timer(StartDelay,2,TimerFcn,e3a02=1,e3a01_ult=0,e3a02_ult=1);
    start(te30102);
end
if e3a02==1 && e3==998
    e3a02=0;
    %tempo de 2s para chegar a outro andar
    te30203=timer(StartDelay,2,TimerFcn,e3a03=1,e3a02_ult=0,e3a03_ult=1);
    start(te30203);
end
if e3a03==1 && e3==998
    e3a03=0;
    %tempo de 2s para chegar a outro andar
    te30304=timer(StartDelay,2,TimerFcn,e3a04=1,e3a03_ult=0,e3a04_ult=1);
    start(te30304);
end
if e3a04==1 && e3==998
    e3a04=0;
    %tempo de 2s para chegar a outro andar
    te30405=timer(StartDelay,2,TimerFcn,e3a05=1,e3a04_ult=0,e3a05_ult=1);
    start(te30405);
end
if e3a05==1 && e3==998
    e3a05=0;
    %tempo de 2s para chegar a outro andar
    te30506=timer(StartDelay,2,TimerFcn,e3a06=1,e3a05_ult=0,e3a06_ult=1);
    start(te30506);
end
if e3a06==1 && e3==998
    e3a06=0;
    %tempo de 2s para chegar a outro andar
    te30607=timer(StartDelay,2,TimerFcn,e3a07=1,e3a06_ult=0,e3a07_ult=1);
    start(te30607);
end
if e3a07==1 && e3==998
    e3a07=0;
    %tempo de 2s para chegar a outro andar
    te30708=timer(StartDelay,2,TimerFcn,e3a08=1,e3a07_ult=0,e3a08_ult=1);
    start(te30708);
end
if e3a08==1 && e3==998
    e3a08=0;
    %tempo de 2s para chegar a outro andar
    te30809=timer(StartDelay,2,TimerFcn,e3a09=1,e3a08_ult=0,e3a09_ult=1);
    start(te30809);
end
if e3a09==1 && e3==998
    e3a09=0;
    %tempo de 2s para chegar a outro andar
    te30910=timer(StartDelay,2,TimerFcn,e3a10=1,e3a09_ult=0,e3a10_ult=1);
    start(te30910);
end
if e3a10==1 && e3==998
    e3a10=0;
    %tempo de 2s para chegar a outro andar
    te31011=timer(StartDelay,2,TimerFcn,e3a11=1,e3a10_ult=0,e3a11_ult=1);
    start(te31011);
end
if e3a11==1 && e3==998
    e3a11=0;
    %tempo de 2s para chegar a outro andar
    te31112=timer(StartDelay,2,TimerFcn,e3a12=1,e3a11_ult=0,e3a12_ult=1);
    start(te31112);
end
if e3a12==1 && e3==998
    e3a12=0;
    %tempo de 2s para chegar a outro andar
    te31213=timer(StartDelay,2,TimerFcn,e3a13=1,e3a12_ult=0,e3a13_ult=1);
    start(te31213);
end

```

```

if e3a13==1 && e3==998
    e3a13=0;
    %tempo de 2s para chegar a outro andar
    te31314=timer('StartDelay',2,'TimerFcn','e3a14=1; e3a13_ult=0; e3a14_ult=1;');
    start(te31314);
end

end

% Caso haja comando para descer, verif. a se deve descer o elevador
if e3d==0
    if e3a14==1 && e3==998
        e3a14=0;
        %tempo de 2s para chegar a outro andar
        te31413=timer('StartDelay',2,'TimerFcn','e3a13=1; e3a14_ult=0; e3a13_ult=1;');
        start(te31413);
    end
    if e3a13==1 && e3==998
        e3a13=0;
        %tempo de 2s para chegar a outro andar
        te31312=timer('StartDelay',2,'TimerFcn','e3a12=1; e3a13_ult=0; e3a12_ult=1;');
        start(te31312);
    end
    if e3a12==1 && e3==998
        e3a12=0;
        %tempo de 2s para chegar a outro andar
        te31211=timer('StartDelay',2,'TimerFcn','e3a11=1; e3a12_ult=0; e3a11_ult=1;');
        start(te31211);
    end
    if e3a11==1 && e3==998
        e3a11=0;
        %tempo de 2s para chegar a outro andar
        te31110=timer('StartDelay',2,'TimerFcn','e3a10=1; e3a11_ult=0; e3a10_ult=1;');
        start(te31110);
    end
    if e3a10==1 && e3==998
        e3a10=0;
        %tempo de 2s para chegar a outro andar
        te31009=timer('StartDelay',2,'TimerFcn','e3a09=1; e3a10_ult=0; e3a09_ult=1;');
        start(te31009);
    end
    if e3a09==1 && e3==998
        e3a09=0;
        %tempo de 2s para chegar a outro andar
        te30908=timer('StartDelay',2,'TimerFcn','e3a08=1; e3a09_ult=0; e3a08_ult=1;');
        start(te30908);
    end
    if e3a08==1 && e3==998
        e3a08=0;
        %tempo de 2s para chegar a outro andar
        te30807=timer('StartDelay',2,'TimerFcn','e3a07=1; e3a08_ult=0; e3a07_ult=1;');
        start(te30807);
    end
    if e3a07==1 && e3==998
        e3a07=0;
        %tempo de 2s para chegar a outro andar
        te30706=timer('StartDelay',2,'TimerFcn','e3a06=1; e3a07_ult=0; e3a06_ult=1;');
        start(te30706);
    end
    if e3a06==1 && e3==998
        e3a06=0;
        %tempo de 2s para chegar a outro andar
        te30605=timer('StartDelay',2,'TimerFcn','e3a05=1; e3a06_ult=0; e3a05_ult=1;');
        start(te30605);
    end
    if e3a05==1 && e3==998
        e3a05=0;
        %tempo de 2s para chegar a outro andar
        te30504=timer('StartDelay',2,'TimerFcn','e3a04=1; e3a05_ult=0; e3a04_ult=1;');
        start(te30504);
    end
    if e3a04==1 && e3==998
        e3a04=0;
        %tempo de 2s para chegar a outro andar
        te30403=timer('StartDelay',2,'TimerFcn','e3a03=1; e3a04_ult=0; e3a03_ult=1;');
        start(te30403);
    end
    if e3a03==1 && e3==998
        e3a03=0;
        %tempo de 2s para chegar a outro andar
        te30302=timer('StartDelay',2,'TimerFcn','e3a02=1; e3a03_ult=0; e3a02_ult=1;');
        start(te30302);
    end
    if e3a02==1 && e3==998
        e3a02=0;
        %tempo de 2s para chegar a outro andar
        te30201=timer('StartDelay',2,'TimerFcn','e3a01=1; e3a02_ult=0; e3a01_ult=1;');
        start(te30201);
    end
    if e3a01==1 && e3==998
        e3a01=0;
        %tempo de 2s para chegar a outro andar
        te30100=timer('StartDelay',2,'TimerFcn','e3a00=1; e3a01_ult=0; e3a00_ult=1;');
        start(te30100);
    end
    if e3a00==1 && e3==998

```

```

e3a00=0;
%tempo de 2s para chegar a outro andar
te300m01=timer( 'startDelay',2,'TimerFcn','3am01=1; e3a00_ult=0; e3am01_ult=1;');
start(te300m01);
end
if e3am01==1 && e3==998
e3am01=0;
%tempo de 2s para chegar a outro andar
te3m01m02=timer( 'startDelay',2,'TimerFcn','3am02=1; e3am01_ult=0; e3am02_ult=1;');
start(te3m01m02);
end
end

%*****
%*****
%CONTROLADOR LOCAL - ELEVADOR 04

%Entrada e Saida de Pessoas

%Chegando no Andar Menos02
if e4am02==1
%Se o botao deste andar do elevador 1 estiver pressionado
if e4bm02==1
%reseta o botao
e4bm02=0;
%coloca o elevador em estado parado, saindo gente
e4=995;
%coloca um delta T para a saida das pessoas, sendo 1s por
%pessoa, e depois desse tempo faz update do numero de pessoas
%no elevador, e coloca elevador em situacao parado e com saida
%de pessoas ja finalizado
te4s=timer( 'startDelay',e4p,'TimerFcn','4p=0;pe4m02=0;e4=996;');
start(te4s);
else
%se o botao do referido andar nao estiver pressionado, entao nao
%vai sair gente e pode colocar ok a saida de pessoas
e4=996;
end
%se ja tiverem saido as pessoas do elevador que queriam descer no
%andar
if e4==996
%como tratamos apenas pico de descida
e4=998;
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e4d='';
if e4bm01==1 || e4b00==1 || e4b01==1 || e4b02==1 || e4b03==1 || e4b04==1 || e4b05==1 || e4b06==1
e4d='';
end
if e4b07==1 || e4b08==1 || e4b09==1 || e4b10==1 || e4b11==1 || e4b12==1 || e4b13==1 || e4b14==1
e4d='';
end
if e4cm01s==1 || e4cm01d==1 || e4c00s==1 || e4c00d==1 || e4c01s==1 || e4c01d==1 || e4c02s==1 || e4c02d==1
e4d='';
end
if e4c03s==1 || e4c03d==1 || e4c04s==1 || e4c04d==1 || e4c05s==1 || e4c05d==1 || e4c06s==1 || e4c06d==1
e4d='';
end
if e4c07s==1 || e4c07d==1 || e4c08s==1 || e4c08d==1 || e4c09s==1 || e4c09d==1 || e4c10s==1 || e4c10d==1
e4d='';
end
if e4c11s==1 || e4c11d==1 || e4c12s==1 || e4c12d==1 || e4c13s==1 || e4c13d==1 || e4c14s==1 || e4c14d==1
e4d='';
end
end

end

%Chegando no Andar Menos01
if e4am01==1
%Se o botao deste andar do elevador 1 estiver pressionado
if e4bm01==1
%reseta o botao
e4bm01=0;
%coloca o elevador em estado parado, saindo gente
e4=995;
%coloca um delta T para a saida das pessoas, sendo 1s por
%pessoa, e depois desse tempo faz update do numero de pessoas
%no elevador, e coloca elevador em situacao parado e com saida
%de pessoas ja finalizado
te4s=timer( 'startDelay',pe4m01,'TimerFcn','4p=e4p-pe4m01;pe4m01=0;e4=996;');
start(te4s);
else
%se o botao do referido andar nao estiver pressionado, entao nao
%vai sair gente e pode colocar ok a saida de pessoas
e4=996;
end
%se ja tiverem saido as pessoas do elevador que queriam descer no
%andar
if e4==996
%Se houver uma chamada do andar alocada para o elevador 1,
%dependendo se o elevador estiver de subida ou de descida

```

```

if e4d==x
    %caso esteja para subir, verifica se ha alocação de chamada
    %para subida
    e4=998;
end

if e4d==d || e4d==x
    %caso esteja para descer, verifica se ha alocação de chamada
    %para descida

    e4=998;
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e4d= n;
if e4b00==1 || e4b01==1 || e4b02==1 || e4b03==1 || e4b04==1 || e4b05==1 || e4b06==1
    e4d= ;
end
if e4b07==1 || e4b08==1 || e4b09==1 || e4b10==1 || e4b11==1 || e4b12==1 || e4b13==1 || e4b14==1
    e4d= ;
end
if e4c00s==1 || e4c00d==1 || e4c01s==1 || e4c01d==1 || e4c02s==1 || e4c02d==1
    e4d= ;
end
if e4c03s==1 || e4c03d==1 || e4c04s==1 || e4c04d==1 || e4c05s==1 || e4c05d==1 || e4c06s==1 || e4c06d==1
    e4d= ;
end
if e4c07s==1 || e4c07d==1 || e4c08s==1 || e4c08d==1 || e4c09s==1 || e4c09d==1 || e4c10s==1 || e4c10d==1
    e4d= ;
end
if e4c11s==1 || e4c11d==1 || e4c12s==1 || e4c12d==1 || e4c13s==1 || e4c13d==1 || e4c14s==1 || e4c14d==1
    e4d= ;
end
if e4bm02==1
    e4d= ;
end
if e4cm02s==1
    e4d= ;
end
end

%Chegando no Andar 00 (terreo)
if e4a00==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e4b00==1
        %pressa o botao
        e4b00=0;
        %coloca o elevador em estado parado, saindo gente
        e4=995;
        %coloca um delay T para a saída das pessoas, sendo Ts por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com saída
        %de pessoas ja finalizado
        te4s=timer('startDelay',pe400,'TimerFcn','e4p=e4p-pe400/pe400-0,e4=996');
        start(te4s);
    else
        if e4=995
            %se o botao do referido andar nao estiver pressionado, entao nao
            %vao sair gente e pode colocar ok a saída de pessoas
            e4=996;
        end
    end
end

%se ja tiverem saído as pessoas do elevador que queriam descer no
%andar
if e4=996
    %Se houver uma chamada do andar abaixo para o elevador 1,
    %dependendo se o elevador estiver de subida ou de descida
    if e4d==x
        e4=998;
    end

    if e4d==d || e4d==x
        e4=998;
    end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e4d= ;
if e4b01==1 || e4b02==1 || e4b03==1 || e4b04==1 || e4b05==1 || e4b06==1
    e4d= ;
end
if e4b07==1 || e4b08==1 || e4b09==1 || e4b10==1 || e4b11==1 || e4b12==1 || e4b13==1 || e4b14==1
    e4d= ;
end
if e4c01s==1 || e4c01d==1 || e4c02s==1 || e4c02d==1
    e4d= ;
end
if e4c03s==1 || e4c03d==1 || e4c04s==1 || e4c04d==1 || e4c05s==1 || e4c05d==1 || e4c06s==1 || e4c06d==1
    e4d= ;
end
if e4c07s==1 || e4c07d==1 || e4c08s==1 || e4c08d==1 || e4c09s==1 || e4c09d==1 || e4c10s==1 || e4c10d==1
    e4d= ;
end

```

```

    e4d= ;
end
if e4c11s==1 || e4c11d==1 || e4c12s==1 || e4c12d==1 || e4c13s==1 || e4c13d==1 || e4c14s==1 || e4c14d==1
    e4d= ;
end
if e4bm01==1 || e4bm02==1
    e4d= ;
end
if e4cm01s==1 || e4cm01d==1 || e4cm02s==1
    e4d= ;
end
end

%Chegando no Andar 01
if e4a01==1
    e4=998;
end

%Chegando no Andar 02
if e4a02==1
    e4=998;
end

%Chegando no Andar 03
if e4a03==1
    e4=998;
end

%Chegando no Andar 04
if e4a04==1
    e4=998;
end

%Chegando no Andar 05
if e4a05==1
    e4=998;
end

%Chegando no Andar 06
if e4a06==1
    e4=998;
end

%Chegando no Andar 07
if e4a07==1
    e4=998;
end

%Chegando no Andar 08
if e4a08==1
    e4=998;
end

%Chegando no Andar 09
if e4a09==1
    e4=998;
end

%Chegando no Andar 10
if e4a10==1

```

```

e4=998;

end

%Chegando no Andar 11
if e4a11==1

    e4=998;

end

%Chegando no Andar 12
if e4a12==1
    e4=996;

    if e4==996

        %Se houver uma chamada do andar alocada para o elevador i,
        %dependendo se o elevador estiver de subida ou de descida
        if e4d==s
            e4=998;

        end

        if e4d==d || e4d==s
            %caso esteja para descer, verifica se ha alocação de chamada
            %para descida
            if e4c12d==1
                %reseta a alocação de chamada
                e4c12d=0;
                %coloca o elevador em estado de parado e entrando gente
                e4=997;
                %calcula o numero de pessoas que entraro no elevador, que
                %no caso eh o minimo entre a capacidade do elevador menos
                %as pessoas que estao dentro dele, e o numero de pessoas
                %aguardando para entrar
                e4p_entr=min(capacidade-e4p,p12d)
                %coloca um delta T para a entrada das pessoas, sendo 1s por
                %pessoa, e depois desse tempo faz update do numero de
                %pessoas no elevador e no andar, e coloca o elevador em
                %situacao parado e ok para subir ou descer
                tc4=timer('StartDelay',e4p_entr,'TimerFcn',e4=998 );
                start(tc4);
                e4p=e4p+e4p_entr;p12d=p12d-e4p_entr;
                %se nao houver outro elevador alocado pra esse andar,
                %coloca como zero o numero de pessoas aguardando jah
                %alocadas para algum elevador

                if e4c12d==0 && e4c12d==0 && e4c12d==0
                    c12d_aloc=0;
                end

                %Enquanto houver pessoas que vao entrar no elevador,
                %verifica pra qual andar eles vao atraves de seu numero
                %el ex. 0 para andas menos 02, 1 para menos 01, 2 para
                %terreo, 3 para 1o andar, etc), apertando o botao e
                %aumentando a variavel de pessoas querendo ir para esse
                %andar
                while e4p_entr>0
                    ult_p12d=ult_p12d+1;
                    if filap12d(ult_p12d)~=0
                        e4bm02=1;
                        pe4m02=pe4m02+1;
                    end
                    if filap12d(ult_p12d)~=1
                        e4bm01=1;
                        pe4m01=pe4m01+1;
                    end
                    if filap12d(ult_p12d)~=2
                        e4b00=1;
                        pe400=pe400+1;
                    end

                    %reduz em 1 o numero de pessoas que vao entrar no
                    %elevador, apos este ter sido incluido no elevador
                    e4p_entr=e4p_entr-1;
                end

                %se nao tiver mais ninguem querendo subir no elevador na
                %direcao especificada, entao zera o botao de chamado do
                %elevamento e contabiliza o tempo que este ficou ativado,
                %sendo um tempo cumulativo

                p12d=0;
                c12d=0;
                tc12d_fim=clock;
                ttot_c12d=ttot_c12d+etime(tc12d_fim,tc12d_ini);
            end

        else
            e4=998;
        end
    end
end
end

```

```

%DEFINICAO DA DIRECAO DO ELEVADOR
%Se a priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e4d=0;

if e4b13==1 || e4b14==1
    e4d=1;
end
if e4c13s==1 || e4c13d==1 || e4c14s==1 || e4c14d==1
    e4d=1;
end
if e4b06==1 || e4b05==1 || e4b04==1 || e4b03==1 || e4b02==1 || e4b01==1 || e4b00==1 || e4bm01==1 || e4bm02==1
    e4d=1;
end
if e4b11==1 || e4b10==1 || e4b09==1 || e4b08==1 || e4b07==1
    e4d=1;
end
if e4c00s==1 || e4c00d==1 || e4c01s==1 || e4c01d==1 || e4c02s==1 || e4c02d==1
    e4d=1;
end
if e4c03s==1 || e4c03d==1 || e4c04s==1 || e4c04d==1 || e4cm01s==1 || e4cm01d==1 || e4cm02s==1 || e4cm02d==1
    e4d=1;
end
if e4c05s==1 || e4c05d==1 || e4c06s==1 || e4c06d==1 || e4c07s==1 || e4c07d==1 || e4c08s==1 || e4c08d==1
    e4d=1;
end
if e4c09s==1 || e4c09d==1 || e4c10s==1 || e4c10d==1 || e4c11s==1 || e4c11d==1
    e4d=1;
end
end
end

```

```

%Chegando no Andar 13

```

```

if e4a13==1
    %Se o botao deste andar do elevador 1 estiver pressionado
    if e4b13==1
        %seta o botao
        e4b13=0;
        %coloca o elevador em estado parado -saindo gente
        e4=995;
        %coloca um delta T para a saida das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de pessoas
        %no elevador, e coloca elevador em situacao parado e com anda
        %de pessoas ja finalizado
        te4s=timer('StartDelay',pe413,'TimerFcn','e4p=e4p-pe413;e4=996');
        start(te4s);
    else
        if e4==997
            %Se o botao do referido andar nao estiver pressionado, entao nao
            %vao sair gente e pode colocar ok a saida de pessoas
            e4=996;
        end
    end
end
%Se ja tiverem saido as pessoas do elevador que queriam descer no
%andar
if e4==996
    %Se houver uma chamada do andar alocado para o elevador 1,
    %dependendo se o elevador estiver de subida ou de descida
    if e4d==1
        e4=998;
    end
end
end

```

```

if e4d==0 || e4d==1
    %Se esta para descer, verifica se ha alocao de chamada
    %para descida
    if e4c13d==1
        %seta a alocao de chamada
        e4c13d=0;
        %coloca o elevador em estado de parado e entrando gente
        e4=997;
        %calcula o numero de pessoas que entraram no elevador, que
        %no caso eh o numero entre a capacidade do elevador menos
        %as pessoas que estao dentro dele, e o numero de pessoas
        %aguardando para entrar
        e4p_entr=min(capacidade-e4p,p13d);
        %coloca um delta T para a entrada das pessoas, sendo 1s por
        %pessoa, e depois desse tempo faz update do numero de
        %pessoas no elevador e no andar, e coloca o elevador em
        %situacao parado e ok para subir ou descer
        te4e=timer('StartDelay',e4p_entr,'TimerFcn','e4=998');
        start(te4e);
        e4p=e4p+e4p_entr;p13d=p13d-e4p_entr;
        %Se nao houver outro elevador alocado pra esse andar,
        %coloca como zero o numero de pessoas aguardando jah
        %alocadas para algum elevador

        %if e4c13d==0 && e4c13d==0 && e4c13d==0
        %    c13d_aloc=0;
        %end
    end
end

```

```

%Enquanto houver pessoas que vão entrar no elevador,
%verifica pra qual andar elas vão através de seu número
%ex. 0 para andar menos 02, 1 para menos 01, 2 para
%terreo, 3 para 1o andar, etc), apertando o botão e
%aumentando a variável de pessoas querendo ir para esse
%andar
while e4p_entr>0
    ult_p13d=ult_p13d+1;
    if filap13d(ult_p13d)==0
        e4bm02=1;
        pe4m02=pe4m02+1;
    end
    if filap13d(ult_p13d)==1
        e4bm01=1;
        pe4m01=pe4m01+1;
    end
    if filap13d(ult_p13d)==2
        e4b00=1;
        pe400=pe400+1;
    end

    %reduz em 1 o número de pessoas que vão entrar no
    %elevador, após este ter sido incluído no elevador
    e4p_entr=e4p_entr-1;
end

%Se não tiver mais ninguém querendo subir no elevador na
%direção especificada, então zera o botão de chamado do
%apartamento e contabiliza o tempo que este ficou ativado,
%usando um tempo cumulativo

if p13d==0
    c13d=0;
    tc13d_fim=clock;
    ttot_c13d=ttot_c13d+etime(tc13d_fim,tc13d_ini);
end

else
    if e4==997
        e4=998;
    end
end
end
end

%DEFINIÇÃO DA DIREÇÃO DO ELEVADOR
%A priori, deixar como Neutra a direção e depois verifica se deve
%ser de subida ou descida
e4d=0;

if e4b14==1
    e4d=-1;
end
if e4c14s==1 || e4c14d==1
    e4d=-1;
end
if e4b06==1 || e4b05==1 || e4b04==1 || e4b03==1 || e4b02==1 || e4b01==1 || e4b00==1 || e4bm01==1 || e4bm02==1
    e4d=-1;
end
if e4b12==1 || e4b11==1 || e4b10==1 || e4b09==1 || e4b08==1 || e4b07==1
    e4d=-1;
end
if e4c00s==1 || e4c00d==1 || e4c01s==1 || e4c01d==1 || e4c02s==1 || e4c02d==1
    e4d=-1;
end
if e4c03s==1 || e4c03d==1 || e4c04s==1 || e4c04d==1 || e4cm01s==1 || e4cm01d==1 || e4cm02s==1
    e4d=-1;
end
if e4c05s==1 || e4c05d==1 || e4c06s==1 || e4c06d==1 || e4c07s==1 || e4c07d==1 || e4c08s==1 || e4c08d==1
    e4d=-1;
end
if e4c09s==1 || e4c09d==1 || e4c10s==1 || e4c10d==1 || e4c11s==1 || e4c11d==1 || e4c12s==1 || e4c12d==1
    e4d=-1;
end
end

%Chegando no Andar 14
if e4a14==1
    %Se o botão deste andar do elevador 1 estiver pressionado...
    if e4b14==1
        %zerar o botão
        e4b14=0;
        %coloca o elevador em estado parado - saindo gente
        e4=995;
        %coloca um delay T para a saída das pessoas, sendo Ts por
        %pessoa, e depois desse tempo faz update do número de pessoas
        %no elevador, e coloca elevador em situação parado e com saída
        %de pessoas já finalizada
        tc4s=timer('StartDelay',e4p,'TimerFcn','e4p=0;pe414=0;e4=996;');
        start(tc4s);
    else
        e4=997
    end
end

```

```

%se o botao do referido andar nao estiver pressionado, entao nao
%vai sair gente e pode colocar ok a saida de pessoas
e4=996;
end
end
%se ja tiverem saido as pessoas do elevador que queiram descer no
%andar
if e4==996
%se houver uma chamada do andar alocada para o elevador 1
if e4c14d==1
%atenta a alocacao de chamada
e4c14d=0;
%coloca o elevador em estado de para-lá e entrando gente
e4=997;
%calcula o numero de pessoas que entraro no elevador, que
%no caso eh o minimo entre a capacidade do elevador maximo
%as pessoas que estao dentro dele, e o numero de pessoas
%aguardando para entrar
e4p_entr=min(capacidade-e4p,p14d)
%coloca um delay 1 para a entrada das pessoas, sendo 1s por
%pessoa, e depois desse tempo faz update do numero de
%pessoas no elevador e no andar, e coloca o elevador em
%situacao parado e ok para subir ou descer
te4=timer('StartDelay',e4p_entr,'TimerFcn','e4=998;');
start(te4);
e4p=e4p+e4p_entr;p14d=p14d-e4p_entr;
%se nao houver outro elevador alocado pra esse andar,
%coloca como zero o numero de pessoas aguardando ja
%alocadas para algum elevador

%if e4c14d==0 && e4c14d==0 && e4c14d==0
%e4c14d=0;
%end

%Enquanto houver pessoas que vao entrar no elevador,
%verifica pra qual andar elas vao atraves de seu numero
%ex: 0 para andar menos 02, 1 para menos 01, 2 para
%terreo, 3 para 1o andar, etc), apertando o botao e
%aumentando a variavel de pessoas querendo ir para esse
%andar
while e4p_entr>0
ult_p14d=ult_p14d+1;
if filap14d(ult_p14d)==1
e4bm01=1;
pe4m01=pe4m01+1;
end
if filap14d(ult_p14d)==2
e4bm00=1;
pe4m00=pe4m00+1;
end
if filap14d(ult_p14d)==0
e4bm02=1;
pe4m02=pe4m02+1;
end
%reduz em 1 o numero de pessoas que vao entrar no
%elevador, apos este ter sido incluído no elevador
e4p_entr=e4p_entr-1;
end

%se nao tiver mais ninguém querendo subir no elevador no
%direcao especificada, entao zera o botao de chamado do
%apartamento e contabiliza o tempo que este ficou ativado,
%sendo um tempo cumulativo

if p14d==0
c14d=0;
tc14d_fim=clock;
ttot_c14d=ttot_c14d+etime(tc14d_fim,tc14d_ini);
end
else
if e4==997
e4=998;
end
end
end

%DEFINICAO DA DIRECAO DO ELEVADOR
%A priori, deixar como Neutra a direcao e depois verifica se deve
%ser de subida ou descida
e4d=;
if e4bm01==1 || e4bm00==1 || e4bm01==1 || e4bm02==1 || e4bm03==1 || e4bm04==1 || e4bm05==1 || e4bm06==1
e4d=;
end
if e4bm07==1 || e4bm08==1 || e4bm09==1 || e4bm10==1 || e4bm11==1 || e4bm12==1 || e4bm13==1 || e4bm02==1
e4d=;
end
if e4cm01s==1 || e4cm01d==1 || e4cm00s==1 || e4cm00d==1 || e4cm01s==1 || e4cm01d==1 || e4cm02s==1 || e4cm02d==1
e4d=;
end
if e4cm03s==1 || e4cm03d==1 || e4cm04s==1 || e4cm04d==1 || e4cm05s==1 || e4cm05d==1 || e4cm06s==1 || e4cm06d==1
e4d=;
end
if e4cm07s==1 || e4cm07d==1 || e4cm08s==1 || e4cm08d==1 || e4cm09s==1 || e4cm09d==1 || e4cm10s==1 || e4cm10d==1
e4d=;
end
if e4cm11s==1 || e4cm11d==1 || e4cm12s==1 || e4cm12d==1 || e4cm13s==1 || e4cm13d==1 || e4cm02s==1 || e4cm02d==1

```

```

e4d==1;
end

end

%
%
% caso haja comando para subida, verifica se deve subir o elevador
if e4d==1
if e4am02==1 && e4==998
e4am02=0;
%tempo de 2s para chegar a outro andar
te4m02m01=timer(StartDelay,2,TimerFcn','e4m01=1; e4am02_ult=0; e4am01_ult=1);
start(te4m02m01);
end
if e4am01==1 && e4==998
e4am01=0;
%tempo de 2s para chegar a outro andar
te4m0100=timer(StartDelay,2,TimerFcn','e4a00=1; e4am01_ult=0; e4a00_ult=1);
start(te4m0100);
end
if e4a00==1 && e4==998
e4a00=0;
%tempo de 2s para chegar a outro andar
te40001=timer(StartDelay,2,TimerFcn','e4a01=1; e4a00_ult=0; e4a01_ult=1);
start(te40001);
end
if e4a01==1 && e4==998
e4a01=0;
%tempo de 2s para chegar a outro andar
te40102=timer(StartDelay,2,TimerFcn','e4a02=1; e4a01_ult=0; e4a02_ult=1);
start(te40102);
end
if e4a02==1 && e4==998
e4a02=0;
%tempo de 2s para chegar a outro andar
te40203=timer(StartDelay,2,TimerFcn','e4a03=1; e4a02_ult=0; e4a03_ult=1);
start(te40203);
end
if e4a03==1 && e4==998
e4a03=0;
%tempo de 2s para chegar a outro andar
te40304=timer(StartDelay,2,TimerFcn','e4a04=1; e4a03_ult=0; e4a04_ult=1);
start(te40304);
end
if e4a04==1 && e4==998
e4a04=0;
%tempo de 2s para chegar a outro andar
te40405=timer(StartDelay,2,TimerFcn','e4a05=1; e4a04_ult=0; e4a05_ult=1);
start(te40405);
end
if e4a05==1 && e4==998
e4a05=0;
%tempo de 2s para chegar a outro andar
te40506=timer(StartDelay,2,TimerFcn','e4a06=1; e4a05_ult=0; e4a06_ult=1);
start(te40506);
end
if e4a06==1 && e4==998
e4a06=0;
%tempo de 2s para chegar a outro andar
te40607=timer(StartDelay,2,TimerFcn','e4a07=1; e4a06_ult=0; e4a07_ult=1);
start(te40607);
end
if e4a07==1 && e4==998
e4a07=0;
%tempo de 2s para chegar a outro andar
te40708=timer(StartDelay,2,TimerFcn','e4a08=1; e4a07_ult=0; e4a08_ult=1);
start(te40708);
end
if e4a08==1 && e4==998
e4a08=0;
%tempo de 2s para chegar a outro andar
te40809=timer(StartDelay,2,TimerFcn','e4a09=1; e4a08_ult=0; e4a09_ult=1);
start(te40809);
end
if e4a09==1 && e4==998
e4a09=0;
%tempo de 2s para chegar a outro andar
te40910=timer(StartDelay,2,TimerFcn','e4a10=1; e4a09_ult=0; e4a10_ult=1);
start(te40910);
end
if e4a10==1 && e4==998
e4a10=0;
%tempo de 2s para chegar a outro andar
te41011=timer(StartDelay,2,TimerFcn','e4a11=1; e4a10_ult=0; e4a11_ult=1);
start(te41011);
end
if e4a11==1 && e4==998
e4a11=0;
%tempo de 2s para chegar a outro andar
te41112=timer(StartDelay,2,TimerFcn','e4a12=1; e4a11_ult=0; e4a12_ult=1);
start(te41112);
end
if e4a12==1 && e4==998
e4a12=0;

```

```

%tempo de 2s para chegar a outro andar
te41213=timer('StartDelay',2,'TimerFcn',@e4a13_ult=0,e4a13_ult=1.);
start(te41213);
end

if e4a13==1 && e4==998
e4a13=0;
%tempo de 2s para chegar a outro andar
te41314=timer('StartDelay',2,'TimerFcn',@e4a14=1,e4a13_ult=0,e4a14_ult=1.);
start(te41314);
end

end

%Caso haja comando para descer, verifica se deve descer o elevador
if e4d==#
if e4a14==1 && e4==998
e4a14=0;
%tempo de 2s para chegar a outro andar
te41413=timer('StartDelay',2,'TimerFcn',@e4a13=1,e4a14_ult=0,e4a13_ult=1.);
start(te41413);
end
if e4a13==1 && e4==998
e4a13=0;
%tempo de 2s para chegar a outro andar
te41312=timer('StartDelay',2,'TimerFcn',@e4a12=1,e4a13_ult=0,e4a12_ult=1.);
start(te41312);
end
if e4a12==1 && e4==998
e4a12=0;
%tempo de 2s para chegar a outro andar
te41211=timer('StartDelay',2,'TimerFcn',@e4a11=1,e4a12_ult=0,e4a11_ult=1.);
start(te41211);
end
if e4a11==1 && e4==998
e4a11=0;
%tempo de 2s para chegar a outro andar
te41110=timer('StartDelay',2,'TimerFcn',@e4a10=1,e4a11_ult=0,e4a10_ult=1.);
start(te41110);
end
if e4a10==1 && e4==998
e4a10=0;
%tempo de 2s para chegar a outro andar
te41009=timer('StartDelay',2,'TimerFcn',@e4a09=1,e4a10_ult=0,e4a09_ult=1.);
start(te41009);
end
if e4a09==1 && e4==998
e4a09=0;
%tempo de 2s para chegar a outro andar
te40908=timer('StartDelay',2,'TimerFcn',@e4a08=1,e4a09_ult=0,e4a08_ult=1.);
start(te40908);
end
if e4a08==1 && e4==998
e4a08=0;
%tempo de 2s para chegar a outro andar
te40807=timer('StartDelay',2,'TimerFcn',@e4a07=1,e4a08_ult=0,e4a07_ult=1.);
start(te40807);
end
if e4a07==1 && e4==998
e4a07=0;
%tempo de 2s para chegar a outro andar
te40706=timer('StartDelay',2,'TimerFcn',@e4a06=1,e4a07_ult=0,e4a06_ult=1.);
start(te40706);
end
if e4a06==1 && e4==998
e4a06=0;
%tempo de 2s para chegar a outro andar
te40605=timer('StartDelay',2,'TimerFcn',@e4a05=1,e4a06_ult=0,e4a05_ult=1.);
start(te40605);
end
if e4a05==1 && e4==998
e4a05=0;
%tempo de 2s para chegar a outro andar
te40504=timer('StartDelay',2,'TimerFcn',@e4a04=1,e4a05_ult=0,e4a04_ult=1.);
start(te40504);
end
if e4a04==1 && e4==998
e4a04=0;
%tempo de 2s para chegar a outro andar
te40403=timer('StartDelay',2,'TimerFcn',@e4a03=1,e4a04_ult=0,e4a03_ult=1.);
start(te40403);
end
if e4a03==1 && e4==998
e4a03=0;
%tempo de 2s para chegar a outro andar
te40302=timer('StartDelay',2,'TimerFcn',@e4a02=1,e4a03_ult=0,e4a02_ult=1.);
start(te40302);
end
if e4a02==1 && e4==998
e4a02=0;
%tempo de 2s para chegar a outro andar
te40201=timer('StartDelay',2,'TimerFcn',@e4a01=1,e4a02_ult=0,e4a01_ult=1.);
start(te40201);
end
if e4a01==1 && e4==998
e4a01=0;
%tempo de 2s para chegar a outro andar

```

```

te40100=timer('StartDelay',2,'TimerFcn','>4a00-1, e4a01_ult=0, e4a00_ult=1');
start(te40100);
end
if e4a00==1 && e4==998
    e4a00=0;
    %tempo de 2s para chegar a outro andar
    te400m01=timer('StartDelay',2,'TimerFcn','>4am01-1, e4a00_ult=0, e4am01_ult=1');
    start(te400m01);
end
if e4am01==1 && e4==998
    e4am01=0;
    %tempo de 2s para chegar a outro andar
    te4m01m02=timer('StartDelay',2,'TimerFcn','>4am02-1, e4am01_ult=0, e4am02_ult=1');
    start(te4m01m02);
end
end

visu3
pause(1)

if filap13d(atu_p13d+1)~=1 && filap14d(atu_p14d+1)~=1 && p14d==0 && p13d==0 && e1p==0 && e2p==0 && e3p==0 && e4p==0
    saida=1;
end

end
visu3
tempodecorrido=toc;

```

1.64

APÊNDICE II

VISUALIZADOR DE ESTADOS DO EDIFÍCIO

% ANEXO II - Visualizador de Estados do Edifício

```
%variáveis usadas
% p14d p13d p12d ... p01d p00d pm01d
% elai4 ult elai3 ult ... elam02 ult
% elp
% se2ai4 ult ...
% se2p
% se3ai4 ult ...
% se3p
% se4ai4 ult ...
% se4p

hold off
plot(1,1)
axis off
title('Visualizador')
maxx=1000;
maxy=100;
axis([0 maxx 0 maxy])
title('Visualizador')
text(50,(maxy/20)*19,'Andar')
text(200,(maxy/20)*19,'Pessoas')
text(450,(maxy/20)*19,'Elev 1')
text(600,(maxy/20)*19,'Elev 2')
text(750,(maxy/20)*19,'Elev 3')
text(900,(maxy/20)*19,'Elev 4')
text(50,(maxy/20), 'Subsco 2')
text(50,(maxy/20)*2, 'Subsco 1')
text(50,(maxy/20)*3, 'Terreo')
text(50,(maxy/20)*4, '1 andar')
text(50,(maxy/20)*5, '2 andar')
text(50,(maxy/20)*6, '3 andar')
text(50,(maxy/20)*7, '4 andar')
text(50,(maxy/20)*8, '5 andar')
text(50,(maxy/20)*9, '6 andar')
text(50,(maxy/20)*10, '7 andar')
text(50,(maxy/20)*11, '8 andar')
text(50,(maxy/20)*12, '9 andar')
text(50,(maxy/20)*13, '10 andar')
text(50,(maxy/20)*14, '11 andar')
text(50,(maxy/20)*15, '12 andar')
text(50,(maxy/20)*16, '13 andar')
text(50,(maxy/20)*17, '14 andar')

%posição das pessoas no eixo x do gráfico
posicaopessoas=250;

%Número de pessoas esperando pra descer no andar 14
if(p14d==0)
text(posicaopessoas,(maxy/20)*17,0)
end
if(p14d==1)
text(posicaopessoas,(maxy/20)*17,1)
end
if(p14d==2)
text(posicaopessoas,(maxy/20)*17,2)
end
if(p14d==3)
text(posicaopessoas,(maxy/20)*17,3)
end
if(p14d==4)
text(posicaopessoas,(maxy/20)*17,4)
end
if(p14d==5)
text(posicaopessoas,(maxy/20)*17,5)
end
if(p14d==6)
text(posicaopessoas,(maxy/20)*17,6)
end
if(p14d==7)
text(posicaopessoas,(maxy/20)*17,7)
end
if(p14d==8)
text(posicaopessoas,(maxy/20)*17,8)
end
if(p14d==9)
text(posicaopessoas,(maxy/20)*17,9)
end
if(p14d==10)
text(posicaopessoas,(maxy/20)*17,10)
end
if(p14d==11)
text(posicaopessoas,(maxy/20)*17,11)
end
if(p14d==12)
text(posicaopessoas,(maxy/20)*17,12)
end
if(p14d==13)
text(posicaopessoas,(maxy/20)*17,13)
end
if(p14d==14)
text(posicaopessoas,(maxy/20)*17,14)
end
if(p14d==15)
```

```

text(posicaoopessoas,(maxy/20)*17,15)
end
if(p14d==16)
text(posicaoopessoas,(maxy/20)*17,16)
end
if(p14d==17)
text(posicaoopessoas,(maxy/20)*17,17)
end
if(p14d==18)
text(posicaoopessoas,(maxy/20)*17,18)
end
if(p14d==19)
text(posicaoopessoas,(maxy/20)*17,19)
end
if(p14d==20)
text(posicaoopessoas,(maxy/20)*17,20)
end
if(p14d==21)
text(posicaoopessoas,(maxy/20)*17,21)
end
if(p14d==22)
text(posicaoopessoas,(maxy/20)*17,22)
end
if(p14d==23)
text(posicaoopessoas,(maxy/20)*17,23)
end
if(p14d==24)
text(posicaoopessoas,(maxy/20)*17,24)
end
if(p14d==25)
text(posicaoopessoas,(maxy/20)*17,25)
end

```

```

%Numero de pessoas esperando pra descer no andar 13

```

```

if(p13d==0)
text(posicaoopessoas,(maxy/20)*16,0)
end
if(p13d==1)
text(posicaoopessoas,(maxy/20)*16,1)
end
if(p13d==2)
text(posicaoopessoas,(maxy/20)*16,2)
end
if(p13d==3)
text(posicaoopessoas,(maxy/20)*16,3)
end
if(p13d==4)
text(posicaoopessoas,(maxy/20)*16,4)
end
if(p13d==5)
text(posicaoopessoas,(maxy/20)*16,5)
end
if(p13d==6)
text(posicaoopessoas,(maxy/20)*16,6)
end
if(p13d==7)
text(posicaoopessoas,(maxy/20)*16,7)
end
if(p13d==8)
text(posicaoopessoas,(maxy/20)*16,8)
end
if(p13d==9)
text(posicaoopessoas,(maxy/20)*16,9)
end
if(p13d==10)
text(posicaoopessoas,(maxy/20)*16,10)
end
if(p13d==11)
text(posicaoopessoas,(maxy/20)*16,11)
end
if(p13d==12)
text(posicaoopessoas,(maxy/20)*16,12)
end
if(p13d==13)
text(posicaoopessoas,(maxy/20)*16,13)
end
if(p13d==14)
text(posicaoopessoas,(maxy/20)*16,14)
end
if(p13d==15)
text(posicaoopessoas,(maxy/20)*16,15)
end
if(p13d==16)
text(posicaoopessoas,(maxy/20)*16,16)
end
if(p13d==17)
text(posicaoopessoas,(maxy/20)*16,17)
end
if(p13d==18)
text(posicaoopessoas,(maxy/20)*16,18)
end
if(p13d==19)
text(posicaoopessoas,(maxy/20)*16,19)
end
if(p13d==20)
text(posicaoopessoas,(maxy/20)*16,20)
end

```

```

end
!(p13d==21)
text(posicaoopessoas,(maxy/20)*16, 21 )
end
!(p13d==22)
text(posicaoopessoas,(maxy/20)*16, 22 )
end
!(p13d==23)
text(posicaoopessoas,(maxy/20)*16, 23 )
end
!(p13d==24)
text(posicaoopessoas,(maxy/20)*16, 24 )
end
!(p13d==25)
text(posicaoopessoas,(maxy/20)*16, 25 )
end

```

%Numero de pessoas esperando pra descer no andar 12

```

!(p12d==0)
text(posicaoopessoas,(maxy/20)*15, 0 )
end
!(p12d==1)
text(posicaoopessoas,(maxy/20)*15, 1 )
end
!(p12d==2)
text(posicaoopessoas,(maxy/20)*15, 2 )
end
!(p12d==3)
text(posicaoopessoas,(maxy/20)*15, 3 )
end
!(p12d==4)
text(posicaoopessoas,(maxy/20)*15, 4 )
end
!(p12d==5)
text(posicaoopessoas,(maxy/20)*15, 5 )
end
!(p12d==6)
text(posicaoopessoas,(maxy/20)*15, 6 )
end
!(p12d==7)
text(posicaoopessoas,(maxy/20)*15, 7 )
end
!(p12d==8)
text(posicaoopessoas,(maxy/20)*15, 8 )
end
!(p12d==9)
text(posicaoopessoas,(maxy/20)*15, 9 )
end
!(p12d==10)
text(posicaoopessoas,(maxy/20)*15, 10 )
end
!(p12d==11)
text(posicaoopessoas,(maxy/20)*15, 11 )
end
!(p12d==12)
text(posicaoopessoas,(maxy/20)*15, 12 )
end
!(p12d==13)
text(posicaoopessoas,(maxy/20)*15, 13 )
end
!(p12d==14)
text(posicaoopessoas,(maxy/20)*15, 14 )
end
!(p12d==15)
text(posicaoopessoas,(maxy/20)*15, 15 )
end
!(p12d==16)
text(posicaoopessoas,(maxy/20)*15, 16 )
end
!(p12d==17)
text(posicaoopessoas,(maxy/20)*15, 17 )
end
!(p12d==18)
text(posicaoopessoas,(maxy/20)*15, 18 )
end
!(p12d==19)
text(posicaoopessoas,(maxy/20)*15, 19 )
end
!(p12d==20)
text(posicaoopessoas,(maxy/20)*15, 20 )
end
!(p12d==21)
text(posicaoopessoas,(maxy/20)*15, 21 )
end
!(p12d==22)
text(posicaoopessoas,(maxy/20)*15, 22 )
end
!(p12d==23)
text(posicaoopessoas,(maxy/20)*15, 23 )
end
!(p12d==24)
text(posicaoopessoas,(maxy/20)*15, 24 )
end
!(p12d==25)
text(posicaoopessoas,(maxy/20)*15, 25 )
end

```

```
%Numero de pessoas esperando pra descer no andar 11
```

```
(p11d=0)
text(posicaoopessoas,(maxy/20)*14,0)
end
(p11d=1)
text(posicaoopessoas,(maxy/20)*14,1)
end
(p11d=2)
text(posicaoopessoas,(maxy/20)*14,2)
end
(p11d=3)
text(posicaoopessoas,(maxy/20)*14,3)
end
(p11d=4)
text(posicaoopessoas,(maxy/20)*14,4)
end
(p11d=5)
text(posicaoopessoas,(maxy/20)*14,5)
end
(p11d=6)
text(posicaoopessoas,(maxy/20)*14,6)
end
(p11d=7)
text(posicaoopessoas,(maxy/20)*14,7)
end
(p11d=8)
text(posicaoopessoas,(maxy/20)*14,8)
end
(p11d=9)
text(posicaoopessoas,(maxy/20)*14,9)
end
(p11d=10)
text(posicaoopessoas,(maxy/20)*14,10)
end
(p11d=11)
text(posicaoopessoas,(maxy/20)*14,11)
end
(p11d=12)
text(posicaoopessoas,(maxy/20)*14,12)
end
(p11d=13)
text(posicaoopessoas,(maxy/20)*14,13)
end
(p11d=14)
text(posicaoopessoas,(maxy/20)*14,14)
end
(p11d=15)
text(posicaoopessoas,(maxy/20)*14,15)
end
(p11d=16)
text(posicaoopessoas,(maxy/20)*14,16)
end
(p11d=17)
text(posicaoopessoas,(maxy/20)*14,17)
end
(p11d=18)
text(posicaoopessoas,(maxy/20)*14,18)
end
(p11d=19)
text(posicaoopessoas,(maxy/20)*14,19)
end
(p11d=20)
text(posicaoopessoas,(maxy/20)*14,20)
end
(p11d=21)
text(posicaoopessoas,(maxy/20)*14,21)
end
(p11d=22)
text(posicaoopessoas,(maxy/20)*14,22)
end
(p11d=23)
text(posicaoopessoas,(maxy/20)*14,23)
end
(p11d=24)
text(posicaoopessoas,(maxy/20)*14,24)
end
(p11d=25)
text(posicaoopessoas,(maxy/20)*14,25)
end
```

```
%Numero de pessoas esperando pra descer no andar 10
```

```
(p10d=0)
text(posicaoopessoas,(maxy/20)*13,0)
end
(p10d=1)
text(posicaoopessoas,(maxy/20)*13,1)
end
(p10d=2)
text(posicaoopessoas,(maxy/20)*13,2)
end
(p10d=3)
text(posicaoopessoas,(maxy/20)*13,3)
end
(p10d=4)
```

```

text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==5)
text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==6)
text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==7)
text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==8)
text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==9)
text(posicao pessoas,(maxy/20)*13, )
end
if(p10d==10)
text(posicao pessoas,(maxy/20)*13, 10)
end
if(p10d==11)
text(posicao pessoas,(maxy/20)*13, 11)
end
if(p10d==12)
text(posicao pessoas,(maxy/20)*13, 12)
end
if(p10d==13)
text(posicao pessoas,(maxy/20)*13, 13)
end
if(p10d==14)
text(posicao pessoas,(maxy/20)*13, 14)
end
if(p10d==15)
text(posicao pessoas,(maxy/20)*13, 15)
end
if(p10d==16)
text(posicao pessoas,(maxy/20)*13, 16)
end
if(p10d==17)
text(posicao pessoas,(maxy/20)*13, 17)
end
if(p10d==18)
text(posicao pessoas,(maxy/20)*13, 18)
end
if(p10d==19)
text(posicao pessoas,(maxy/20)*13, 19)
end
if(p10d==20)
text(posicao pessoas,(maxy/20)*13, 20)
end
if(p10d==21)
text(posicao pessoas,(maxy/20)*13, 21)
end
if(p10d==22)
text(posicao pessoas,(maxy/20)*13, 22)
end
if(p10d==23)
text(posicao pessoas,(maxy/20)*13, 23)
end
if(p10d==24)
text(posicao pessoas,(maxy/20)*13, 24)
end
if(p10d==25)
text(posicao pessoas,(maxy/20)*13, 25)
end

```

%Numero de pessoas esperando pra descer no andar 09

```

if(p09d==0)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==1)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==2)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==3)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==4)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==5)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==6)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==7)
text(posicao pessoas,(maxy/20)*12, )
end
if(p09d==8)
text(posicao pessoas,(maxy/20)*12, )
end

```

```

(p09d==9)
text(posicaoopessoas,(maxy/20)*12,9)
end
i(p09d==10)
text(posicaoopessoas,(maxy/20)*12,10)
end
i(p09d==11)
text(posicaoopessoas,(maxy/20)*12,11)
end
i(p09d==12)
text(posicaoopessoas,(maxy/20)*12,12)
end
i(p09d==13)
text(posicaoopessoas,(maxy/20)*12,13)
end
i(p09d==14)
text(posicaoopessoas,(maxy/20)*12,14)
end
i(p09d==15)
text(posicaoopessoas,(maxy/20)*12,15)
end
i(p09d==16)
text(posicaoopessoas,(maxy/20)*12,16)
end
i(p09d==17)
text(posicaoopessoas,(maxy/20)*12,17)
end
i(p09d==18)
text(posicaoopessoas,(maxy/20)*12,18)
end
i(p09d==19)
text(posicaoopessoas,(maxy/20)*12,19)
end
i(p09d==20)
text(posicaoopessoas,(maxy/20)*12,20)
end
i(p09d==21)
text(posicaoopessoas,(maxy/20)*12,21)
end
i(p09d==22)
text(posicaoopessoas,(maxy/20)*12,22)
end
i(p09d==23)
text(posicaoopessoas,(maxy/20)*12,23)
end
i(p09d==24)
text(posicaoopessoas,(maxy/20)*12,24)
end
i(p09d==25)
text(posicaoopessoas,(maxy/20)*12,25)
end
end

```

%Numero de pessoas esperando em cada andar 08

```

i(p08d==0)
text(posicaoopessoas,(maxy/20)*11,0)
end
i(p08d==1)
text(posicaoopessoas,(maxy/20)*11,1)
end
i(p08d==2)
text(posicaoopessoas,(maxy/20)*11,2)
end
i(p08d==3)
text(posicaoopessoas,(maxy/20)*11,3)
end
i(p08d==4)
text(posicaoopessoas,(maxy/20)*11,4)
end
i(p08d==5)
text(posicaoopessoas,(maxy/20)*11,5)
end
i(p08d==6)
text(posicaoopessoas,(maxy/20)*11,6)
end
i(p08d==7)
text(posicaoopessoas,(maxy/20)*11,7)
end
i(p08d==8)
text(posicaoopessoas,(maxy/20)*11,8)
end
i(p08d==9)
text(posicaoopessoas,(maxy/20)*11,9)
end
i(p08d==10)
text(posicaoopessoas,(maxy/20)*11,10)
end
i(p08d==11)
text(posicaoopessoas,(maxy/20)*11,11)
end
i(p08d==12)
text(posicaoopessoas,(maxy/20)*11,12)
end
i(p08d==13)
text(posicaoopessoas,(maxy/20)*11,13)
end
end

```

```

(p08d==14)
text(posicaoopessoas,(maxy/20)*11,14)
end
(p08d==15)
text(posicaoopessoas,(maxy/20)*11,15)
end
(p08d==16)
text(posicaoopessoas,(maxy/20)*11,16)
end
(p08d==17)
text(posicaoopessoas,(maxy/20)*11,17)
end
(p08d==18)
text(posicaoopessoas,(maxy/20)*11,18)
end
(p08d==19)
text(posicaoopessoas,(maxy/20)*11,19)
end
(p08d==20)
text(posicaoopessoas,(maxy/20)*11,20)
end
(p08d==21)
text(posicaoopessoas,(maxy/20)*11,21)
end
(p08d==22)
text(posicaoopessoas,(maxy/20)*11,22)
end
(p08d==23)
text(posicaoopessoas,(maxy/20)*11,23)
end
(p08d==24)
text(posicaoopessoas,(maxy/20)*11,24)
end
(p08d==25)
text(posicaoopessoas,(maxy/20)*11,25)
end

```

*%Número de pessoas esperando pra descer no andar 07

```

(p07d==0)
text(posicaoopessoas,(maxy/20)*10,0)
end
(p07d==1)
text(posicaoopessoas,(maxy/20)*10,1)
end
(p07d==2)
text(posicaoopessoas,(maxy/20)*10,2)
end
(p07d==3)
text(posicaoopessoas,(maxy/20)*10,3)
end
(p07d==4)
text(posicaoopessoas,(maxy/20)*10,4)
end
(p07d==5)
text(posicaoopessoas,(maxy/20)*10,5)
end
(p07d==6)
text(posicaoopessoas,(maxy/20)*10,6)
end
(p07d==7)
text(posicaoopessoas,(maxy/20)*10,7)
end
(p07d==8)
text(posicaoopessoas,(maxy/20)*10,8)
end
(p07d==9)
text(posicaoopessoas,(maxy/20)*10,9)
end
(p07d==10)
text(posicaoopessoas,(maxy/20)*10,10)
end
(p07d==11)
text(posicaoopessoas,(maxy/20)*10,11)
end
(p07d==12)
text(posicaoopessoas,(maxy/20)*10,12)
end
(p07d==13)
text(posicaoopessoas,(maxy/20)*10,13)
end
(p07d==14)
text(posicaoopessoas,(maxy/20)*10,14)
end
(p07d==15)
text(posicaoopessoas,(maxy/20)*10,15)
end
(p07d==16)
text(posicaoopessoas,(maxy/20)*10,16)
end
(p07d==17)
text(posicaoopessoas,(maxy/20)*10,17)
end
(p07d==18)
text(posicaoopessoas,(maxy/20)*10,18)
end
(p07d==19)

```

```

text(posicaoopessoas,(maxy/20)*10,10)
end
(p07d==20)
text(posicaoopessoas,(maxy/20)*10,20)
end
(p07d==21)
text(posicaoopessoas,(maxy/20)*10,21)
end
(p07d==22)
text(posicaoopessoas,(maxy/20)*10,22)
end
(p07d==23)
text(posicaoopessoas,(maxy/20)*10,23)
end
(p07d==24)
text(posicaoopessoas,(maxy/20)*10,24)
end
(p07d==25)
text(posicaoopessoas,(maxy/20)*10,25)
end

```

%Numero de pessoas esperando pra descer no andar 06

```

(p06d==0)
text(posicaoopessoas,(maxy/20)*9,0)
end
(p06d==1)
text(posicaoopessoas,(maxy/20)*9,1)
end
(p06d==2)
text(posicaoopessoas,(maxy/20)*9,2)
end
(p06d==3)
text(posicaoopessoas,(maxy/20)*9,3)
end
(p06d==4)
text(posicaoopessoas,(maxy/20)*9,4)
end
(p06d==5)
text(posicaoopessoas,(maxy/20)*9,5)
end
(p06d==6)
text(posicaoopessoas,(maxy/20)*9,6)
end
(p06d==7)
text(posicaoopessoas,(maxy/20)*9,7)
end
(p06d==8)
text(posicaoopessoas,(maxy/20)*9,8)
end
(p06d==9)
text(posicaoopessoas,(maxy/20)*9,9)
end
(p06d==10)
text(posicaoopessoas,(maxy/20)*9,10)
end
(p06d==11)
text(posicaoopessoas,(maxy/20)*9,11)
end
(p06d==12)
text(posicaoopessoas,(maxy/20)*9,12)
end
(p06d==13)
text(posicaoopessoas,(maxy/20)*9,13)
end
(p06d==14)
text(posicaoopessoas,(maxy/20)*9,14)
end
(p06d==15)
text(posicaoopessoas,(maxy/20)*9,15)
end
(p06d==16)
text(posicaoopessoas,(maxy/20)*9,16)
end
(p06d==17)
text(posicaoopessoas,(maxy/20)*9,17)
end
(p06d==18)
text(posicaoopessoas,(maxy/20)*9,18)
end
(p06d==19)
text(posicaoopessoas,(maxy/20)*9,19)
end
(p06d==20)
text(posicaoopessoas,(maxy/20)*9,20)
end
(p06d==21)
text(posicaoopessoas,(maxy/20)*9,21)
end
(p06d==22)
text(posicaoopessoas,(maxy/20)*9,22)
end
(p06d==23)
text(posicaoopessoas,(maxy/20)*9,23)
end
(p06d==24)
text(posicaoopessoas,(maxy/20)*9,24)
end

```

```

end
i(p06d==25)
text(posicaoopessoas,(maxy/20)*9,25)
end

%Numero de pessoas esperando pra descer no andar 5
i(p05d==0)
text(posicaoopessoas,(maxy/20)*8,0)
end
i(p05d==1)
text(posicaoopessoas,(maxy/20)*8,1)
end
i(p05d==2)
text(posicaoopessoas,(maxy/20)*8,2)
end
i(p05d==3)
text(posicaoopessoas,(maxy/20)*8,3)
end
i(p05d==4)
text(posicaoopessoas,(maxy/20)*8,4)
end
i(p05d==5)
text(posicaoopessoas,(maxy/20)*8,5)
end
i(p05d==6)
text(posicaoopessoas,(maxy/20)*8,6)
end
i(p05d==7)
text(posicaoopessoas,(maxy/20)*8,7)
end
i(p05d==8)
text(posicaoopessoas,(maxy/20)*8,8)
end
i(p05d==9)
text(posicaoopessoas,(maxy/20)*8,9)
end
i(p05d==10)
text(posicaoopessoas,(maxy/20)*8,10)
end
i(p05d==11)
text(posicaoopessoas,(maxy/20)*8,11)
end
i(p05d==12)
text(posicaoopessoas,(maxy/20)*8,12)
end
i(p05d==13)
text(posicaoopessoas,(maxy/20)*8,13)
end
i(p05d==14)
text(posicaoopessoas,(maxy/20)*8,14)
end
i(p05d==15)
text(posicaoopessoas,(maxy/20)*8,15)
end
i(p05d==16)
text(posicaoopessoas,(maxy/20)*8,16)
end
i(p05d==17)
text(posicaoopessoas,(maxy/20)*8,17)
end
i(p05d==18)
text(posicaoopessoas,(maxy/20)*8,18)
end
i(p05d==19)
text(posicaoopessoas,(maxy/20)*8,19)
end
i(p05d==20)
text(posicaoopessoas,(maxy/20)*8,20)
end
i(p05d==21)
text(posicaoopessoas,(maxy/20)*8,21)
end
i(p05d==22)
text(posicaoopessoas,(maxy/20)*8,22)
end
i(p05d==23)
text(posicaoopessoas,(maxy/20)*8,23)
end
i(p05d==24)
text(posicaoopessoas,(maxy/20)*8,24)
end
i(p05d==25)
text(posicaoopessoas,(maxy/20)*8,25)
end

%Numero de pessoas esperando pra descer no andar 4
i(p04d==0)
text(posicaoopessoas,(maxy/20)*7,0)
end
i(p04d==1)
text(posicaoopessoas,(maxy/20)*7,1)
end
i(p04d==2)
text(posicaoopessoas,(maxy/20)*7,2)
end
i(p04d==3)

```

```

text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==4)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==5)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==6)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==7)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==8)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==9)
text(posicao pessoas, (maxy/20)*7, )
end
!(p04d==10)
text(posicao pessoas, (maxy/20)*7, 10)
end
!(p04d==11)
text(posicao pessoas, (maxy/20)*7, 11)
end
!(p04d==12)
text(posicao pessoas, (maxy/20)*7, 12)
end
!(p04d==13)
text(posicao pessoas, (maxy/20)*7, 13)
end
!(p04d==14)
text(posicao pessoas, (maxy/20)*7, 14)
end
!(p04d==15)
text(posicao pessoas, (maxy/20)*7, 15)
end
!(p04d==16)
text(posicao pessoas, (maxy/20)*7, 16)
end
!(p04d==17)
text(posicao pessoas, (maxy/20)*7, 17)
end
!(p04d==18)
text(posicao pessoas, (maxy/20)*7, 18)
end
!(p04d==19)
text(posicao pessoas, (maxy/20)*7, 19)
end
!(p04d==20)
text(posicao pessoas, (maxy/20)*7, 20)
end
!(p04d==21)
text(posicao pessoas, (maxy/20)*7, 21)
end
!(p04d==22)
text(posicao pessoas, (maxy/20)*7, 22)
end
!(p04d==23)
text(posicao pessoas, (maxy/20)*7, 23)
end
!(p04d==24)
text(posicao pessoas, (maxy/20)*7, 24)
end
!(p04d==25)
text(posicao pessoas, (maxy/20)*7, 25)
end
end

```

*Número de pessoas esperando pra descer no andar 3

```

!(p03d==0)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==1)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==2)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==3)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==4)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==5)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==6)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==7)
text(posicao pessoas, (maxy/20)*6, )
end
!(p03d==8)
text(posicao pessoas, (maxy/20)*6, )
end

```

```

end
!(p03d==9)
textf(posicaoopessoas,(maxy/20)*6, )
end
!(p03d==10)
textf(posicaoopessoas,(maxy/20)*6, 10)
end
!(p03d==11)
textf(posicaoopessoas,(maxy/20)*6, 11)
end
!(p03d==12)
textf(posicaoopessoas,(maxy/20)*6, 12)
end
!(p03d==13)
textf(posicaoopessoas,(maxy/20)*6, 13)
end
!(p03d==14)
textf(posicaoopessoas,(maxy/20)*6, 14)
end
!(p03d==15)
textf(posicaoopessoas,(maxy/20)*6, 15)
end
!(p03d==16)
textf(posicaoopessoas,(maxy/20)*6, 16)
end
!(p03d==17)
textf(posicaoopessoas,(maxy/20)*6, 17)
end
!(p03d==18)
textf(posicaoopessoas,(maxy/20)*6, 18)
end
!(p03d==19)
textf(posicaoopessoas,(maxy/20)*6, 19)
end
!(p03d==20)
textf(posicaoopessoas,(maxy/20)*6, 20)
end
!(p03d==21)
textf(posicaoopessoas,(maxy/20)*6, 21)
end
!(p03d==22)
textf(posicaoopessoas,(maxy/20)*6, 22)
end
!(p03d==23)
textf(posicaoopessoas,(maxy/20)*6, 23)
end
!(p03d==24)
textf(posicaoopessoas,(maxy/20)*6, 24)
end
!(p03d==25)
textf(posicaoopessoas,(maxy/20)*6, 25)
end

```

*Número de pessoas esperando pra descer no andar 2

```

!(p02d==0)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==1)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==2)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==3)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==4)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==5)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==6)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==7)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==8)
textf(posicaoopessoas,(maxy/20)*5, 8)
end
!(p02d==9)
textf(posicaoopessoas,(maxy/20)*5, )
end
!(p02d==10)
textf(posicaoopessoas,(maxy/20)*5, 10)
end
!(p02d==11)
textf(posicaoopessoas,(maxy/20)*5, 11)
end
!(p02d==12)
textf(posicaoopessoas,(maxy/20)*5, 12)
end
!(p02d==13)
textf(posicaoopessoas,(maxy/20)*5, 13)
end

```

```

i:(p02d==14)
text(posicaoopessoas,(maxy/20)*5,14)
end
i:(p02d==15)
text(posicaoopessoas,(maxy/20)*5,15)
end
i:(p02d==16)
text(posicaoopessoas,(maxy/20)*5,16)
end
i:(p02d==17)
text(posicaoopessoas,(maxy/20)*5,17)
end
i:(p02d==18)
text(posicaoopessoas,(maxy/20)*5,18)
end
i:(p02d==19)
text(posicaoopessoas,(maxy/20)*5,19)
end
i:(p02d==20)
text(posicaoopessoas,(maxy/20)*5,20)
end
i:(p02d==21)
text(posicaoopessoas,(maxy/20)*5,21)
end
i:(p02d==22)
text(posicaoopessoas,(maxy/20)*5,22)
end
i:(p02d==23)
text(posicaoopessoas,(maxy/20)*5,23)
end
i:(p02d==24)
text(posicaoopessoas,(maxy/20)*5,24)
end
i:(p02d==25)
text(posicaoopessoas,(maxy/20)*5,25)
end

%%Numero de pessoas esperando pra descer no andar i
i:(p01d==0)
text(posicaoopessoas,(maxy/20)*4,0)
end
i:(p01d==1)
text(posicaoopessoas,(maxy/20)*4,1)
end
i:(p01d==2)
text(posicaoopessoas,(maxy/20)*4,2)
end
i:(p01d==3)
text(posicaoopessoas,(maxy/20)*4,3)
end
i:(p01d==4)
text(posicaoopessoas,(maxy/20)*4,4)
end
i:(p01d==5)
text(posicaoopessoas,(maxy/20)*4,5)
end
i:(p01d==6)
text(posicaoopessoas,(maxy/20)*4,6)
end
i:(p01d==7)
text(posicaoopessoas,(maxy/20)*4,7)
end
i:(p01d==8)
text(posicaoopessoas,(maxy/20)*4,8)
end
i:(p01d==9)
text(posicaoopessoas,(maxy/20)*4,9)
end
i:(p01d==10)
text(posicaoopessoas,(maxy/20)*4,10)
end
i:(p01d==11)
text(posicaoopessoas,(maxy/20)*4,11)
end
i:(p01d==12)
text(posicaoopessoas,(maxy/20)*4,12)
end
i:(p01d==13)
text(posicaoopessoas,(maxy/20)*4,13)
end
i:(p01d==14)
text(posicaoopessoas,(maxy/20)*4,14)
end
i:(p01d==15)
text(posicaoopessoas,(maxy/20)*4,15)
end
i:(p01d==16)
text(posicaoopessoas,(maxy/20)*4,16)
end
i:(p01d==17)
text(posicaoopessoas,(maxy/20)*4,17)
end
i:(p01d==18)
text(posicaoopessoas,(maxy/20)*4,18)
end
i:(p01d==19)

```

```

text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==20)
text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==21)
text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==22)
text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==23)
text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==24)
text(posicaoopessoas,(maxy/20)*4,1)
end
if(p01d==25)
text(posicaoopessoas,(maxy/20)*4,1)
end

%Numero de pessoas esperando pra descer no terraço
if(p00d==0)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==1)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==2)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==3)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==4)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==5)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==6)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==7)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==8)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==9)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==10)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==11)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==12)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==13)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==14)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==15)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==16)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==17)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==18)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==19)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==20)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==21)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==22)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==23)
text(posicaoopessoas,(maxy/20)*3,1)
end
if(p00d==24)
text(posicaoopessoas,(maxy/20)*3,1)
end

```

```

end
i(pm0d==25)
text(posicaopessoas,(maxy/20)*3,0)
end

%Numero de pessoas esperando pra descer no subsolo 1
i(pm01d==0)
text(posicaopessoas,(maxy/20)*2,0)
end
i(pm01d==1)
text(posicaopessoas,(maxy/20)*2,1)
end
i(pm01d==2)
text(posicaopessoas,(maxy/20)*2,2)
end
i(pm01d==3)
text(posicaopessoas,(maxy/20)*2,3)
end
i(pm01d==4)
text(posicaopessoas,(maxy/20)*2,4)
end
i(pm01d==5)
text(posicaopessoas,(maxy/20)*2,5)
end
i(pm01d==6)
text(posicaopessoas,(maxy/20)*2,6)
end
i(pm01d==7)
text(posicaopessoas,(maxy/20)*2,7)
end
i(pm01d==8)
text(posicaopessoas,(maxy/20)*2,8)
end
i(pm01d==9)
text(posicaopessoas,(maxy/20)*2,9)
end
i(pm01d==10)
text(posicaopessoas,(maxy/20)*2,10)
end
i(pm01d==11)
text(posicaopessoas,(maxy/20)*2,11)
end
i(pm01d==12)
text(posicaopessoas,(maxy/20)*2,12)
end
i(pm01d==13)
text(posicaopessoas,(maxy/20)*2,13)
end
i(pm01d==14)
text(posicaopessoas,(maxy/20)*2,14)
end
i(pm01d==15)
text(posicaopessoas,(maxy/20)*2,15)
end
i(pm01d==16)
text(posicaopessoas,(maxy/20)*2,16)
end
i(pm01d==17)
text(posicaopessoas,(maxy/20)*2,17)
end
i(pm01d==18)
text(posicaopessoas,(maxy/20)*2,18)
end
i(pm01d==19)
text(posicaopessoas,(maxy/20)*2,19)
end
i(pm01d==20)
text(posicaopessoas,(maxy/20)*2,20)
end
i(pm01d==21)
text(posicaopessoas,(maxy/20)*2,21)
end
i(pm01d==22)
text(posicaopessoas,(maxy/20)*2,22)
end
i(pm01d==23)
text(posicaopessoas,(maxy/20)*2,23)
end
i(pm01d==24)
text(posicaopessoas,(maxy/20)*2,24)
end
i(pm01d==25)
text(posicaopessoas,(maxy/20)*2,25)
end

%Numero de pessoas esperando pra descer no subsolo 2
text(posicaopessoas,(maxy/20)*1,0)

%Posicao do Elevador 1 no eixo x
posicaoelevador1=470;

%andar que se encontra o elevador 1
i(e1a14_ult==1)
i(e1p==12)

```

```

    text(posicaoelevador1,(maxy/20)*17,12)
end
if(elp==11)
    text(posicaoelevador1,(maxy/20)*17,11)
end
if(elp==10)
    text(posicaoelevador1,(maxy/20)*17,10)
end
if(elp==9)
    text(posicaoelevador1,(maxy/20)*17,9)
end
if(elp==8)
    text(posicaoelevador1,(maxy/20)*17,8)
end
if(elp==7)
    text(posicaoelevador1,(maxy/20)*17,7)
end
if(elp==6)
    text(posicaoelevador1,(maxy/20)*17,6)
end
if(elp==5)
    text(posicaoelevador1,(maxy/20)*17,5)
end
if(elp==4)
    text(posicaoelevador1,(maxy/20)*17,4)
end
if(elp==3)
    text(posicaoelevador1,(maxy/20)*17,3)
end
if(elp==2)
    text(posicaoelevador1,(maxy/20)*17,2)
end
if(elp==1)
    text(posicaoelevador1,(maxy/20)*17,1)
end
if(elp==0)
    text(posicaoelevador1,(maxy/20)*17,0)
end
end
end

if(ela13_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*16,12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*16,11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*16,10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*16,9)
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*16,8)
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*16,7)
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*16,6)
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*16,5)
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*16,4)
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*16,3)
    end
    if(elp==2)
        text(posicaoelevador1,(maxy/20)*16,2)
    end
    if(elp==1)
        text(posicaoelevador1,(maxy/20)*16,1)
    end
    if(elp==0)
        text(posicaoelevador1,(maxy/20)*16,0)
    end
end
end

if(ela12_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*15,12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*15,11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*15,10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*15,9)
    end
    if(elp==8)

```

```

    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==7)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==6)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==5)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==4)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==3)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==2)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==1)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
  if (el p==0)
    text(posicaoelevador1,(maxy/20)*15,0)
  end
end

if (el a11_ult==1)
  if (el p==12)
    text(posicaoelevador1,(maxy/20)*14,12)
  end
  if (el p==11)
    text(posicaoelevador1,(maxy/20)*14,11)
  end
  if (el p==10)
    text(posicaoelevador1,(maxy/20)*14,10)
  end
  if (el p==9)
    text(posicaoelevador1,(maxy/20)*14,9)
  end
  if (el p==8)
    text(posicaoelevador1,(maxy/20)*14,8)
  end
  if (el p==7)
    text(posicaoelevador1,(maxy/20)*14,7)
  end
  if (el p==6)
    text(posicaoelevador1,(maxy/20)*14,6)
  end
  if (el p==5)
    text(posicaoelevador1,(maxy/20)*14,5)
  end
  if (el p==4)
    text(posicaoelevador1,(maxy/20)*14,4)
  end
  if (el p==3)
    text(posicaoelevador1,(maxy/20)*14,3)
  end
  if (el p==2)
    text(posicaoelevador1,(maxy/20)*14,2)
  end
  if (el p==1)
    text(posicaoelevador1,(maxy/20)*14,1)
  end
  if (el p==0)
    text(posicaoelevador1,(maxy/20)*14,0)
  end
end

if (el a10_ult==1)
  if (el p==12)
    text(posicaoelevador1,(maxy/20)*13,12)
  end
  if (el p==11)
    text(posicaoelevador1,(maxy/20)*13,11)
  end
  if (el p==10)
    text(posicaoelevador1,(maxy/20)*13,10)
  end
  if (el p==9)
    text(posicaoelevador1,(maxy/20)*13,9)
  end
  if (el p==8)
    text(posicaoelevador1,(maxy/20)*13,8)
  end
  if (el p==7)
    text(posicaoelevador1,(maxy/20)*13,7)
  end
  if (el p==6)
    text(posicaoelevador1,(maxy/20)*13,6)
  end
  if (el p==5)
    text(posicaoelevador1,(maxy/20)*13,5)
  end
  if (el p==4)
    text(posicaoelevador1,(maxy/20)*13,4)
  end
end

```

```

text(posicaoelevador1,(maxy/20)*13,0)
end
!(elp==3)
text(posicaoelevador1,(maxy/20)*13,0)
end
!(elp==2)
text(posicaoelevador1,(maxy/20)*13,0)
end
!(elp==1)
text(posicaoelevador1,(maxy/20)*13,1)
end
!(elp==0)
text(posicaoelevador1,(maxy/20)*13,0)
end
end

!(ela09_ult==1)
!(elp==12)
text(posicaoelevador1,(maxy/20)*12,12)
end
!(elp==11)
text(posicaoelevador1,(maxy/20)*12,11)
end
!(elp==10)
text(posicaoelevador1,(maxy/20)*12,10)
end
!(elp==9)
text(posicaoelevador1,(maxy/20)*12,9)
end
!(elp==8)
text(posicaoelevador1,(maxy/20)*12,8)
end
!(elp==7)
text(posicaoelevador1,(maxy/20)*12,7)
end
!(elp==6)
text(posicaoelevador1,(maxy/20)*12,6)
end
!(elp==5)
text(posicaoelevador1,(maxy/20)*12,5)
end
!(elp==4)
text(posicaoelevador1,(maxy/20)*12,4)
end
!(elp==3)
text(posicaoelevador1,(maxy/20)*12,3)
end
!(elp==2)
text(posicaoelevador1,(maxy/20)*12,2)
end
!(elp==1)
text(posicaoelevador1,(maxy/20)*12,1)
end
!(elp==0)
text(posicaoelevador1,(maxy/20)*12,0)
end
end

!(ela08_ult==1)
!(elp==12)
text(posicaoelevador1,(maxy/20)*11,12)
end
!(elp==11)
text(posicaoelevador1,(maxy/20)*11,11)
end
!(elp==10)
text(posicaoelevador1,(maxy/20)*11,10)
end
!(elp==9)
text(posicaoelevador1,(maxy/20)*11,9)
end
!(elp==8)
text(posicaoelevador1,(maxy/20)*11,8)
end
!(elp==7)
text(posicaoelevador1,(maxy/20)*11,7)
end
!(elp==6)
text(posicaoelevador1,(maxy/20)*11,6)
end
!(elp==5)
text(posicaoelevador1,(maxy/20)*11,5)
end
!(elp==4)
text(posicaoelevador1,(maxy/20)*11,4)
end
!(elp==3)
text(posicaoelevador1,(maxy/20)*11,3)
end
!(elp==2)
text(posicaoelevador1,(maxy/20)*11,2)
end
!(elp==1)
text(posicaoelevador1,(maxy/20)*11,1)
end
!(elp==0)

```

```

text(posicaoelevador1,(maxy/20)*11, 0)
end
end

if (ela07_ult==1)
  if (elp==12)
    text(posicaoelevador1,(maxy/20)*10, 12)
  end
  if (elp==11)
    text(posicaoelevador1,(maxy/20)*10, 11)
  end
  if (elp==10)
    text(posicaoelevador1,(maxy/20)*10, 10)
  end
  if (elp==9)
    text(posicaoelevador1,(maxy/20)*10, 9)
  end
  if (elp==8)
    text(posicaoelevador1,(maxy/20)*10, 8)
  end
  if (elp==7)
    text(posicaoelevador1,(maxy/20)*10, 7)
  end
  if (elp==6)
    text(posicaoelevador1,(maxy/20)*10, 6)
  end
  if (elp==5)
    text(posicaoelevador1,(maxy/20)*10, 5)
  end
  if (elp==4)
    text(posicaoelevador1,(maxy/20)*10, 4)
  end
  if (elp==3)
    text(posicaoelevador1,(maxy/20)*10, 3)
  end
  if (elp==2)
    text(posicaoelevador1,(maxy/20)*10, 2)
  end
  if (elp==1)
    text(posicaoelevador1,(maxy/20)*10, 1)
  end
  if (elp==0)
    text(posicaoelevador1,(maxy/20)*10, 0)
  end
end

if (ela06_ult==1)
  if (elp==12)
    text(posicaoelevador1,(maxy/20)*9, 12)
  end
  if (elp==11)
    text(posicaoelevador1,(maxy/20)*9, 11)
  end
  if (elp==10)
    text(posicaoelevador1,(maxy/20)*9, 10)
  end
  if (elp==9)
    text(posicaoelevador1,(maxy/20)*9, 9)
  end
  if (elp==8)
    text(posicaoelevador1,(maxy/20)*9, 8)
  end
  if (elp==7)
    text(posicaoelevador1,(maxy/20)*9, 7)
  end
  if (elp==6)
    text(posicaoelevador1,(maxy/20)*9, 6)
  end
  if (elp==5)
    text(posicaoelevador1,(maxy/20)*9, 5)
  end
  if (elp==4)
    text(posicaoelevador1,(maxy/20)*9, 4)
  end
  if (elp==3)
    text(posicaoelevador1,(maxy/20)*9, 3)
  end
  if (elp==2)
    text(posicaoelevador1,(maxy/20)*9, 2)
  end
  if (elp==1)
    text(posicaoelevador1,(maxy/20)*9, 1)
  end
  if (elp==0)
    text(posicaoelevador1,(maxy/20)*9, 0)
  end
end

if (ela05_ult==1)
  if (elp==12)
    text(posicaoelevador1,(maxy/20)*8, 12)
  end
  if (elp==11)
    text(posicaoelevador1,(maxy/20)*8, 11)
  end
  if (elp==10)
    text(posicaoelevador1,(maxy/20)*8, 10)
  end
end

```

```

    text(posicaoelevador1,(maxy/20)*8,10)
end
if(elp==9)
    text(posicaoelevador1,(maxy/20)*8,9)
end
if(elp==8)
    text(posicaoelevador1,(maxy/20)*8,8)
end
if(elp==7)
    text(posicaoelevador1,(maxy/20)*8,7)
end
if(elp==6)
    text(posicaoelevador1,(maxy/20)*8,6)
end
if(elp==5)
    text(posicaoelevador1,(maxy/20)*8,5)
end
if(elp==4)
    text(posicaoelevador1,(maxy/20)*8,4)
end
if(elp==3)
    text(posicaoelevador1,(maxy/20)*8,3)
end
if(elp==2)
    text(posicaoelevador1,(maxy/20)*8,2)
end
if(elp==1)
    text(posicaoelevador1,(maxy/20)*8,1)
end
if(elp==0)
    text(posicaoelevador1,(maxy/20)*8,0)
end
end

if(ela04_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*7,12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*7,11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*7,10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*7,9)
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*7,8)
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*7,7)
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*7,6)
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*7,5)
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*7,4)
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*7,3)
    end
    if(elp==2)
        text(posicaoelevador1,(maxy/20)*7,2)
    end
    if(elp==1)
        text(posicaoelevador1,(maxy/20)*7,1)
    end
    if(elp==0)
        text(posicaoelevador1,(maxy/20)*7,0)
    end
end

if(ela03_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*6,12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*6,11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*6,10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*6,9)
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*6,8)
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*6,7)
    end
    if(elp==6)

```

```

    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==5)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==4)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==3)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==2)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==1)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
if(elp==0)
    text(posicaoelevador1,(maxy/20)*6, ' ')
end
end

if(ela02_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*5, 12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*5, 11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*5, 10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==2)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==1)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
    if(elp==0)
        text(posicaoelevador1,(maxy/20)*5, ' ')
    end
end

if(ela01_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*4, 12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*4, 11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*4, 10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*4, ' ')
    end
    if(elp==2)

```

```

    text(posicaoelevador1,(maxy/20)*4, )
end
if(elp==1)
    text(posicaoelevador1,(maxy/20)*4, )
end
if(elp==0)
    text(posicaoelevador1,(maxy/20)*4, )
end
end

if(ela00_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*3, 12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*3, 11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*3, 10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*3, 9)
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*3, 8)
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*3, 7)
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*3, 6)
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*3, 5)
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*3, 4)
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*3, 3)
    end
    if(elp==2)
        text(posicaoelevador1,(maxy/20)*3, 2)
    end
    if(elp==1)
        text(posicaoelevador1,(maxy/20)*3, 1)
    end
    if(elp==0)
        text(posicaoelevador1,(maxy/20)*3, 0)
    end
end

if(ela01_ult==1)
    if(elp==12)
        text(posicaoelevador1,(maxy/20)*2, 12)
    end
    if(elp==11)
        text(posicaoelevador1,(maxy/20)*2, 11)
    end
    if(elp==10)
        text(posicaoelevador1,(maxy/20)*2, 10)
    end
    if(elp==9)
        text(posicaoelevador1,(maxy/20)*2, 9)
    end
    if(elp==8)
        text(posicaoelevador1,(maxy/20)*2, 8)
    end
    if(elp==7)
        text(posicaoelevador1,(maxy/20)*2, 7)
    end
    if(elp==6)
        text(posicaoelevador1,(maxy/20)*2, 6)
    end
    if(elp==5)
        text(posicaoelevador1,(maxy/20)*2, 5)
    end
    if(elp==4)
        text(posicaoelevador1,(maxy/20)*2, 4)
    end
    if(elp==3)
        text(posicaoelevador1,(maxy/20)*2, 3)
    end
    if(elp==2)
        text(posicaoelevador1,(maxy/20)*2, 2)
    end
    if(elp==1)
        text(posicaoelevador1,(maxy/20)*2, 1)
    end
    if(elp==0)
        text(posicaoelevador1,(maxy/20)*2, 0)
    end
end

if(ela02_ult==1)
    if(elp==12)

```

```

    text(posicaoelevador1,(maxy/20)*1, 12)
end
if(e1p==11)
    text(posicaoelevador1,(maxy/20)*1, 11)
end
if(e1p==10)
    text(posicaoelevador1,(maxy/20)*1, 10)
end
if(e1p==9)
    text(posicaoelevador1,(maxy/20)*1, 9)
end
if(e1p==8)
    text(posicaoelevador1,(maxy/20)*1, 8)
end
if(e1p==7)
    text(posicaoelevador1,(maxy/20)*1, 7)
end
if(e1p==6)
    text(posicaoelevador1,(maxy/20)*1, 6)
end
if(e1p==5)
    text(posicaoelevador1,(maxy/20)*1, 5)
end
if(e1p==4)
    text(posicaoelevador1,(maxy/20)*1, 4)
end
if(e1p==3)
    text(posicaoelevador1,(maxy/20)*1, 3)
end
if(e1p==2)
    text(posicaoelevador1,(maxy/20)*1, 2)
end
if(e1p==1)
    text(posicaoelevador1,(maxy/20)*1, 1)
end
if(e1p==0)
    text(posicaoelevador1,(maxy/20)*1, 0)
end
end
end

```

*Posicao do Elevador 2 no eixo x

posicaoelevador2=630;

*Verificar que se encontra o elevador 2

```

if(e2a14_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*17, 12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*17, 11)
    end
    if(e2p==10)
        text(posicaoelevador2,(maxy/20)*17, 10)
    end
    if(e2p==9)
        text(posicaoelevador2,(maxy/20)*17, 9)
    end
    if(e2p==8)
        text(posicaoelevador2,(maxy/20)*17, 8)
    end
    if(e2p==7)
        text(posicaoelevador2,(maxy/20)*17, 7)
    end
    if(e2p==6)
        text(posicaoelevador2,(maxy/20)*17, 6)
    end
    if(e2p==5)
        text(posicaoelevador2,(maxy/20)*17, 5)
    end
    if(e2p==4)
        text(posicaoelevador2,(maxy/20)*17, 4)
    end
    if(e2p==3)
        text(posicaoelevador2,(maxy/20)*17, 3)
    end
    if(e2p==2)
        text(posicaoelevador2,(maxy/20)*17, 2)
    end
    if(e2p==1)
        text(posicaoelevador2,(maxy/20)*17, 1)
    end
    if(e2p==0)
        text(posicaoelevador2,(maxy/20)*17, 0)
    end
end
end

if(e2a13_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*16, 12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*16, 11)
    end
end

```

```

end
if (e2p==10)
    text(posicaoelevador2,(maxy/20)*16,10)
end
if (e2p==9)
    text(posicaoelevador2,(maxy/20)*16,9)
end
if (e2p==8)
    text(posicaoelevador2,(maxy/20)*16,8)
end
if (e2p==7)
    text(posicaoelevador2,(maxy/20)*16,7)
end
if (e2p==6)
    text(posicaoelevador2,(maxy/20)*16,6)
end
if (e2p==5)
    text(posicaoelevador2,(maxy/20)*16,5)
end
if (e2p==4)
    text(posicaoelevador2,(maxy/20)*16,4)
end
if (e2p==3)
    text(posicaoelevador2,(maxy/20)*16,3)
end
if (e2p==2)
    text(posicaoelevador2,(maxy/20)*16,2)
end
if (e2p==1)
    text(posicaoelevador2,(maxy/20)*16,1)
end
if (e2p==0)
    text(posicaoelevador2,(maxy/20)*16,0)
end
end

if (e2a12_ult==1)
    if (e2p==12)
        text(posicaoelevador2,(maxy/20)*15,12)
    end
    if (e2p==11)
        text(posicaoelevador2,(maxy/20)*15,11)
    end
    if (e2p==10)
        text(posicaoelevador2,(maxy/20)*15,10)
    end
    if (e2p==9)
        text(posicaoelevador2,(maxy/20)*15,9)
    end
    if (e2p==8)
        text(posicaoelevador2,(maxy/20)*15,8)
    end
    if (e2p==7)
        text(posicaoelevador2,(maxy/20)*15,7)
    end
    if (e2p==6)
        text(posicaoelevador2,(maxy/20)*15,6)
    end
    if (e2p==5)
        text(posicaoelevador2,(maxy/20)*15,5)
    end
    if (e2p==4)
        text(posicaoelevador2,(maxy/20)*15,4)
    end
    if (e2p==3)
        text(posicaoelevador2,(maxy/20)*15,3)
    end
    if (e2p==2)
        text(posicaoelevador2,(maxy/20)*15,2)
    end
    if (e2p==1)
        text(posicaoelevador2,(maxy/20)*15,1)
    end
    if (e2p==0)
        text(posicaoelevador2,(maxy/20)*15,0)
    end
end

if (e2a11_ult==1)
    if (e2p==12)
        text(posicaoelevador2,(maxy/20)*14,12)
    end
    if (e2p==11)
        text(posicaoelevador2,(maxy/20)*14,11)
    end
    if (e2p==10)
        text(posicaoelevador2,(maxy/20)*14,10)
    end
    if (e2p==9)
        text(posicaoelevador2,(maxy/20)*14,9)
    end
    if (e2p==8)
        text(posicaoelevador2,(maxy/20)*14,8)
    end
    if (e2p==7)
        text(posicaoelevador2,(maxy/20)*14,7)
    end

```

```

end
if (e2p==6)
    text(posicaoelevador2,(maxy/20)*14, 0)
end
if (e2p==5)
    text(posicaoelevador2,(maxy/20)*14, 1)
end
if (e2p==4)
    text(posicaoelevador2,(maxy/20)*14, 2)
end
if (e2p==3)
    text(posicaoelevador2,(maxy/20)*14, 3)
end
if (e2p==2)
    text(posicaoelevador2,(maxy/20)*14, 4)
end
if (e2p==1)
    text(posicaoelevador2,(maxy/20)*14, 5)
end
if (e2p==0)
    text(posicaoelevador2,(maxy/20)*14, 6)
end
end

if (e2a10_ult==1)
    if (e2p==12)
        text(posicaoelevador2,(maxy/20)*13, 12)
    end
    if (e2p==11)
        text(posicaoelevador2,(maxy/20)*13, 11)
    end
    if (e2p==10)
        text(posicaoelevador2,(maxy/20)*13, 10)
    end
    if (e2p==9)
        text(posicaoelevador2,(maxy/20)*13, 9)
    end
    if (e2p==8)
        text(posicaoelevador2,(maxy/20)*13, 8)
    end
    if (e2p==7)
        text(posicaoelevador2,(maxy/20)*13, 7)
    end
    if (e2p==6)
        text(posicaoelevador2,(maxy/20)*13, 6)
    end
    if (e2p==5)
        text(posicaoelevador2,(maxy/20)*13, 5)
    end
    if (e2p==4)
        text(posicaoelevador2,(maxy/20)*13, 4)
    end
    if (e2p==3)
        text(posicaoelevador2,(maxy/20)*13, 3)
    end
    if (e2p==2)
        text(posicaoelevador2,(maxy/20)*13, 2)
    end
    if (e2p==1)
        text(posicaoelevador2,(maxy/20)*13, 1)
    end
    if (e2p==0)
        text(posicaoelevador2,(maxy/20)*13, 0)
    end
end
end

if (e2a09_ult==1)
    if (e2p==12)
        text(posicaoelevador2,(maxy/20)*12, 12)
    end
    if (e2p==11)
        text(posicaoelevador2,(maxy/20)*12, 11)
    end
    if (e2p==10)
        text(posicaoelevador2,(maxy/20)*12, 10)
    end
    if (e2p==9)
        text(posicaoelevador2,(maxy/20)*12, 9)
    end
    if (e2p==8)
        text(posicaoelevador2,(maxy/20)*12, 8)
    end
    if (e2p==7)
        text(posicaoelevador2,(maxy/20)*12, 7)
    end
    if (e2p==6)
        text(posicaoelevador2,(maxy/20)*12, 6)
    end
    if (e2p==5)
        text(posicaoelevador2,(maxy/20)*12, 5)
    end
    if (e2p==4)
        text(posicaoelevador2,(maxy/20)*12, 4)
    end
    if (e2p==3)
        text(posicaoelevador2,(maxy/20)*12, 3)
    end

```

```

end
if(e2p==2)
    text(posicaoelevador2,(maxy/20)*12,*)
end
if(e2p==1)
    text(posicaoelevador2,(maxy/20)*12,*)
end
if(e2p==0)
    text(posicaoelevador2,(maxy/20)*12,*)
end
end

if(e2a08_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*11,12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*11,11)
    end
    if(e2p==10)
        text(posicaoelevador2,(maxy/20)*11,10)
    end
    if(e2p==9)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==8)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==7)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==6)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==5)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==4)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==3)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==2)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==1)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
    if(e2p==0)
        text(posicaoelevador2,(maxy/20)*11,*)
    end
end

if(e2a07_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*10,12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*10,11)
    end
    if(e2p==10)
        text(posicaoelevador2,(maxy/20)*10,10)
    end
    if(e2p==9)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==8)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==7)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==6)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==5)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==4)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==3)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==2)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==1)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
    if(e2p==0)
        text(posicaoelevador2,(maxy/20)*10,*)
    end
end
end

```

```

(e2a06_ult==1)
  if(e2p==12)
    text(posicaoelevador2,(maxy/20)*9,12)
  end
  if(e2p==11)
    text(posicaoelevador2,(maxy/20)*9,11)
  end
  if(e2p==10)
    text(posicaoelevador2,(maxy/20)*9,10)
  end
  if(e2p==9)
    text(posicaoelevador2,(maxy/20)*9,9)
  end
  if(e2p==8)
    text(posicaoelevador2,(maxy/20)*9,8)
  end
  if(e2p==7)
    text(posicaoelevador2,(maxy/20)*9,7)
  end
  if(e2p==6)
    text(posicaoelevador2,(maxy/20)*9,6)
  end
  if(e2p==5)
    text(posicaoelevador2,(maxy/20)*9,5)
  end
  if(e2p==4)
    text(posicaoelevador2,(maxy/20)*9,4)
  end
  if(e2p==3)
    text(posicaoelevador2,(maxy/20)*9,3)
  end
  if(e2p==2)
    text(posicaoelevador2,(maxy/20)*9,2)
  end
  if(e2p==1)
    text(posicaoelevador2,(maxy/20)*9,1)
  end
  if(e2p==0)
    text(posicaoelevador2,(maxy/20)*9,0)
  end
end

if(e2a05_ult==1)
  if(e2p==12)
    text(posicaoelevador2,(maxy/20)*8,12)
  end
  if(e2p==11)
    text(posicaoelevador2,(maxy/20)*8,11)
  end
  if(e2p==10)
    text(posicaoelevador2,(maxy/20)*8,10)
  end
  if(e2p==9)
    text(posicaoelevador2,(maxy/20)*8,9)
  end
  if(e2p==8)
    text(posicaoelevador2,(maxy/20)*8,8)
  end
  if(e2p==7)
    text(posicaoelevador2,(maxy/20)*8,7)
  end
  if(e2p==6)
    text(posicaoelevador2,(maxy/20)*8,6)
  end
  if(e2p==5)
    text(posicaoelevador2,(maxy/20)*8,5)
  end
  if(e2p==4)
    text(posicaoelevador2,(maxy/20)*8,4)
  end
  if(e2p==3)
    text(posicaoelevador2,(maxy/20)*8,3)
  end
  if(e2p==2)
    text(posicaoelevador2,(maxy/20)*8,2)
  end
  if(e2p==1)
    text(posicaoelevador2,(maxy/20)*8,1)
  end
  if(e2p==0)
    text(posicaoelevador2,(maxy/20)*8,0)
  end
end

if(e2a04_ult==1)
  if(e2p==12)
    text(posicaoelevador2,(maxy/20)*7,12)
  end
  if(e2p==11)
    text(posicaoelevador2,(maxy/20)*7,11)
  end
  if(e2p==10)
    text(posicaoelevador2,(maxy/20)*7,10)
  end
  if(e2p==9)
    text(posicaoelevador2,(maxy/20)*7,9)
  end

```

```

end
!(e2p==8)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==7)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==6)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==5)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==4)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==3)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==2)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==1)
text(posicaoelevador2,(maxy/20)*7, )
end
!(e2p==0)
text(posicaoelevador2,(maxy/20)*7, )
end
end

!(e2a03_ult==1)
!(e2p==12)
text(posicaoelevador2,(maxy/20)*6, 12)
end
!(e2p==11)
text(posicaoelevador2,(maxy/20)*6, 11)
end
!(e2p==10)
text(posicaoelevador2,(maxy/20)*6, 10)
end
!(e2p==9)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==8)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==7)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==6)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==5)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==4)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==3)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==2)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==1)
text(posicaoelevador2,(maxy/20)*6, )
end
!(e2p==0)
text(posicaoelevador2,(maxy/20)*6, )
end
end

!(e2a02_ult==1)
!(e2p==12)
text(posicaoelevador2,(maxy/20)*5, 12)
end
!(e2p==11)
text(posicaoelevador2,(maxy/20)*5, 11)
end
!(e2p==10)
text(posicaoelevador2,(maxy/20)*5, 10)
end
!(e2p==9)
text(posicaoelevador2,(maxy/20)*5, )
end
!(e2p==8)
text(posicaoelevador2,(maxy/20)*5, )
end
!(e2p==7)
text(posicaoelevador2,(maxy/20)*5, )
end
!(e2p==6)
text(posicaoelevador2,(maxy/20)*5, )
end
!(e2p==5)
text(posicaoelevador2,(maxy/20)*5, )

```

```

end
if(e2p==4)
    text(posicaoelevador2,(maxy/20)*5, )
end
if(e2p==3)
    text(posicaoelevador2,(maxy/20)*5, )
end
if(e2p==2)
    text(posicaoelevador2,(maxy/20)*5, )
end
if(e2p==1)
    text(posicaoelevador2,(maxy/20)*5, )
end
if(e2p==0)
    text(posicaoelevador2,(maxy/20)*5, )
end
end

if(e2a01_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*4,12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*4,11)
    end
    if(e2p==10)
        text(posicaoelevador2,(maxy/20)*4,10)
    end
    if(e2p==9)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==8)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==7)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==6)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==5)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==4)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==3)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==2)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==1)
        text(posicaoelevador2,(maxy/20)*4, )
    end
    if(e2p==0)
        text(posicaoelevador2,(maxy/20)*4, )
    end
end

if(e2a00_ult==1)
    if(e2p==12)
        text(posicaoelevador2,(maxy/20)*3,12)
    end
    if(e2p==11)
        text(posicaoelevador2,(maxy/20)*3,11)
    end
    if(e2p==10)
        text(posicaoelevador2,(maxy/20)*3,10)
    end
    if(e2p==9)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==8)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==7)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==6)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==5)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==4)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==3)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==2)
        text(posicaoelevador2,(maxy/20)*3, )
    end
    if(e2p==1)
        text(posicaoelevador2,(maxy/20)*3, )
    end

```

```

end
!(e2p==0)
text(posicaoelevador2,(maxy/20)*3, )
end
end

!(e2am01_ult==1)
!(e2p==12)
text(posicaoelevador2,(maxy/20)*2, 12)
end
!(e2p==11)
text(posicaoelevador2,(maxy/20)*2, 11)
end
!(e2p==10)
text(posicaoelevador2,(maxy/20)*2, 10)
end
!(e2p==9)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==8)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==7)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==6)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==5)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==4)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==3)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==2)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==1)
text(posicaoelevador2,(maxy/20)*2, )
end
!(e2p==0)
text(posicaoelevador2,(maxy/20)*2, )
end
end

!(e2am02_ult==1)
!(e2p==12)
text(posicaoelevador2,(maxy/20)*1, 12)
end
!(e2p==11)
text(posicaoelevador2,(maxy/20)*1, 11)
end
!(e2p==10)
text(posicaoelevador2,(maxy/20)*1, 10)
end
!(e2p==9)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==8)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==7)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==6)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==5)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==4)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==3)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==2)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==1)
text(posicaoelevador2,(maxy/20)*1, )
end
!(e2p==0)
text(posicaoelevador2,(maxy/20)*1, )
end
end

```

*A Posicao do Elevador 3 no eixo x
posicaoelevador3=780;

```

*andar que se encontra o elevador 3
i(e3a14_ult==1)
  i(e3p==12)
    text(posicaoelevador3,(maxy/20)*17,12)
  end
  i(e3p==11)
    text(posicaoelevador3,(maxy/20)*17,11)
  end
  i(e3p==10)
    text(posicaoelevador3,(maxy/20)*17,10)
  end
  i(e3p==9)
    text(posicaoelevador3,(maxy/20)*17,9)
  end
  i(e3p==8)
    text(posicaoelevador3,(maxy/20)*17,8)
  end
  i(e3p==7)
    text(posicaoelevador3,(maxy/20)*17,7)
  end
  i(e3p==6)
    text(posicaoelevador3,(maxy/20)*17,6)
  end
  i(e3p==5)
    text(posicaoelevador3,(maxy/20)*17,5)
  end
  i(e3p==4)
    text(posicaoelevador3,(maxy/20)*17,4)
  end
  i(e3p==3)
    text(posicaoelevador3,(maxy/20)*17,3)
  end
  i(e3p==2)
    text(posicaoelevador3,(maxy/20)*17,2)
  end
  i(e3p==1)
    text(posicaoelevador3,(maxy/20)*17,1)
  end
  i(e3p==0)
    text(posicaoelevador3,(maxy/20)*17,0)
  end
end

i(e3a13_ult==1)
  i(e3p==12)
    text(posicaoelevador3,(maxy/20)*16,12)
  end
  i(e3p==11)
    text(posicaoelevador3,(maxy/20)*16,11)
  end
  i(e3p==10)
    text(posicaoelevador3,(maxy/20)*16,10)
  end
  i(e3p==9)
    text(posicaoelevador3,(maxy/20)*16,9)
  end
  i(e3p==8)
    text(posicaoelevador3,(maxy/20)*16,8)
  end
  i(e3p==7)
    text(posicaoelevador3,(maxy/20)*16,7)
  end
  i(e3p==6)
    text(posicaoelevador3,(maxy/20)*16,6)
  end
  i(e3p==5)
    text(posicaoelevador3,(maxy/20)*16,5)
  end
  i(e3p==4)
    text(posicaoelevador3,(maxy/20)*16,4)
  end
  i(e3p==3)
    text(posicaoelevador3,(maxy/20)*16,3)
  end
  i(e3p==2)
    text(posicaoelevador3,(maxy/20)*16,2)
  end
  i(e3p==1)
    text(posicaoelevador3,(maxy/20)*16,1)
  end
  i(e3p==0)
    text(posicaoelevador3,(maxy/20)*16,0)
  end
end

i(e3a12_ult==1)
  i(e3p==12)
    text(posicaoelevador3,(maxy/20)*15,12)
  end
  i(e3p==11)
    text(posicaoelevador3,(maxy/20)*15,11)
  end
  i(e3p==10)
    text(posicaoelevador3,(maxy/20)*15,10)
  end
  i(e3p==9)

```

```

    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==8)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==7)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==6)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==5)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==4)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==3)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==2)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==1)
    text(posicaoelevador3,(maxy/20)*15, )
end
i'(e3p==0)
    text(posicaoelevador3,(maxy/20)*15, )
end
end

i'(e3a11_ult==1)
i'(e3p==12)
    text(posicaoelevador3,(maxy/20)*14, 12)
end
i'(e3p==11)
    text(posicaoelevador3,(maxy/20)*14, 11)
end
i'(e3p==10)
    text(posicaoelevador3,(maxy/20)*14, 10)
end
i'(e3p==9)
    text(posicaoelevador3,(maxy/20)*14, 9)
end
i'(e3p==8)
    text(posicaoelevador3,(maxy/20)*14, 8)
end
i'(e3p==7)
    text(posicaoelevador3,(maxy/20)*14, 7)
end
i'(e3p==6)
    text(posicaoelevador3,(maxy/20)*14, 6)
end
i'(e3p==5)
    text(posicaoelevador3,(maxy/20)*14, 5)
end
i'(e3p==4)
    text(posicaoelevador3,(maxy/20)*14, 4)
end
i'(e3p==3)
    text(posicaoelevador3,(maxy/20)*14, 3)
end
i'(e3p==2)
    text(posicaoelevador3,(maxy/20)*14, 2)
end
i'(e3p==1)
    text(posicaoelevador3,(maxy/20)*14, 1)
end
i'(e3p==0)
    text(posicaoelevador3,(maxy/20)*14, 0)
end
end

i'(e3a10_ult==1)
i'(e3p==12)
    text(posicaoelevador3,(maxy/20)*13, 12)
end
i'(e3p==11)
    text(posicaoelevador3,(maxy/20)*13, 11)
end
i'(e3p==10)
    text(posicaoelevador3,(maxy/20)*13, 10)
end
i'(e3p==9)
    text(posicaoelevador3,(maxy/20)*13, 9)
end
i'(e3p==8)
    text(posicaoelevador3,(maxy/20)*13, 8)
end
i'(e3p==7)
    text(posicaoelevador3,(maxy/20)*13, 7)
end
i'(e3p==6)
    text(posicaoelevador3,(maxy/20)*13, 6)
end
i'(e3p==5)

```

```

      text(posicaoelevador3,(maxy/20)*13,3)
    end
    if(e3p==4)
      text(posicaoelevador3,(maxy/20)*13,4)
    end
    if(e3p==3)
      text(posicaoelevador3,(maxy/20)*13,5)
    end
    if(e3p==2)
      text(posicaoelevador3,(maxy/20)*13,6)
    end
    if(e3p==1)
      text(posicaoelevador3,(maxy/20)*13,7)
    end
    if(e3p==0)
      text(posicaoelevador3,(maxy/20)*13,8)
    end
  end

if(e3a09_ult==1)
  if(e3p==12)
    text(posicaoelevador3,(maxy/20)*12,12)
  end
  if(e3p==11)
    text(posicaoelevador3,(maxy/20)*12,11)
  end
  if(e3p==10)
    text(posicaoelevador3,(maxy/20)*12,10)
  end
  if(e3p==9)
    text(posicaoelevador3,(maxy/20)*12,9)
  end
  if(e3p==8)
    text(posicaoelevador3,(maxy/20)*12,8)
  end
  if(e3p==7)
    text(posicaoelevador3,(maxy/20)*12,7)
  end
  if(e3p==6)
    text(posicaoelevador3,(maxy/20)*12,6)
  end
  if(e3p==5)
    text(posicaoelevador3,(maxy/20)*12,5)
  end
  if(e3p==4)
    text(posicaoelevador3,(maxy/20)*12,4)
  end
  if(e3p==3)
    text(posicaoelevador3,(maxy/20)*12,3)
  end
  if(e3p==2)
    text(posicaoelevador3,(maxy/20)*12,2)
  end
  if(e3p==1)
    text(posicaoelevador3,(maxy/20)*12,1)
  end
  if(e3p==0)
    text(posicaoelevador3,(maxy/20)*12,0)
  end
end

if(e3a08_ult==1)
  if(e3p==12)
    text(posicaoelevador3,(maxy/20)*11,12)
  end
  if(e3p==11)
    text(posicaoelevador3,(maxy/20)*11,11)
  end
  if(e3p==10)
    text(posicaoelevador3,(maxy/20)*11,10)
  end
  if(e3p==9)
    text(posicaoelevador3,(maxy/20)*11,9)
  end
  if(e3p==8)
    text(posicaoelevador3,(maxy/20)*11,8)
  end
  if(e3p==7)
    text(posicaoelevador3,(maxy/20)*11,7)
  end
  if(e3p==6)
    text(posicaoelevador3,(maxy/20)*11,6)
  end
  if(e3p==5)
    text(posicaoelevador3,(maxy/20)*11,5)
  end
  if(e3p==4)
    text(posicaoelevador3,(maxy/20)*11,4)
  end
  if(e3p==3)
    text(posicaoelevador3,(maxy/20)*11,3)
  end
  if(e3p==2)
    text(posicaoelevador3,(maxy/20)*11,2)
  end
  if(e3p==1)
    text(posicaoelevador3,(maxy/20)*11,1)
  end
end

```

```

    text(posicaoelevador3,(maxy/20)*11, )
end
!(e3p==0)
    text(posicaoelevador3,(maxy/20)*11, )
end
end
!(e3a07_ult==1)
!(e3p==12)
    text(posicaoelevador3,(maxy/20)*10, 12)
end
!(e3p==11)
    text(posicaoelevador3,(maxy/20)*10, 11)
end
!(e3p==10)
    text(posicaoelevador3,(maxy/20)*10, 10)
end
!(e3p==9)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==8)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==7)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==6)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==5)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==4)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==3)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==2)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==1)
    text(posicaoelevador3,(maxy/20)*10, )
end
!(e3p==0)
    text(posicaoelevador3,(maxy/20)*10, )
end
end
!(e3a06_ult==1)
!(e3p==12)
    text(posicaoelevador3,(maxy/20)*9, 12)
end
!(e3p==11)
    text(posicaoelevador3,(maxy/20)*9, 11)
end
!(e3p==10)
    text(posicaoelevador3,(maxy/20)*9, 10)
end
!(e3p==9)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==8)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==7)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==6)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==5)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==4)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==3)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==2)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==1)
    text(posicaoelevador3,(maxy/20)*9, )
end
!(e3p==0)
    text(posicaoelevador3,(maxy/20)*9, )
end
end
!(e3a05_ult==1)
!(e3p==12)
    text(posicaoelevador3,(maxy/20)*8, 12)
end
!(e3p==11)

```

```

    text(posicaoelevador3,(maxy/20)*8, 1)
end
if(e3p==10)
    text(posicaoelevador3,(maxy/20)*8, 10)
end
if(e3p==9)
    text(posicaoelevador3,(maxy/20)*8, 9)
end
if(e3p==8)
    text(posicaoelevador3,(maxy/20)*8, 8)
end
if(e3p==7)
    text(posicaoelevador3,(maxy/20)*8, 7)
end
if(e3p==6)
    text(posicaoelevador3,(maxy/20)*8, 6)
end
if(e3p==5)
    text(posicaoelevador3,(maxy/20)*8, 5)
end
if(e3p==4)
    text(posicaoelevador3,(maxy/20)*8, 4)
end
if(e3p==3)
    text(posicaoelevador3,(maxy/20)*8, 3)
end
if(e3p==2)
    text(posicaoelevador3,(maxy/20)*8, 2)
end
if(e3p==1)
    text(posicaoelevador3,(maxy/20)*8, 1)
end
if(e3p==0)
    text(posicaoelevador3,(maxy/20)*8, 0)
end
end

if(e3a04_ult==1)
    if(e3p==12)
        text(posicaoelevador3,(maxy/20)*7, 12)
    end
    if(e3p==11)
        text(posicaoelevador3,(maxy/20)*7, 11)
    end
    if(e3p==10)
        text(posicaoelevador3,(maxy/20)*7, 10)
    end
    if(e3p==9)
        text(posicaoelevador3,(maxy/20)*7, 9)
    end
    if(e3p==8)
        text(posicaoelevador3,(maxy/20)*7, 8)
    end
    if(e3p==7)
        text(posicaoelevador3,(maxy/20)*7, 7)
    end
    if(e3p==6)
        text(posicaoelevador3,(maxy/20)*7, 6)
    end
    if(e3p==5)
        text(posicaoelevador3,(maxy/20)*7, 5)
    end
    if(e3p==4)
        text(posicaoelevador3,(maxy/20)*7, 4)
    end
    if(e3p==3)
        text(posicaoelevador3,(maxy/20)*7, 3)
    end
    if(e3p==2)
        text(posicaoelevador3,(maxy/20)*7, 2)
    end
    if(e3p==1)
        text(posicaoelevador3,(maxy/20)*7, 1)
    end
    if(e3p==0)
        text(posicaoelevador3,(maxy/20)*7, 0)
    end
end

if(e3a03_ult==1)
    if(e3p==12)
        text(posicaoelevador3,(maxy/20)*6, 12)
    end
    if(e3p==11)
        text(posicaoelevador3,(maxy/20)*6, 11)
    end
    if(e3p==10)
        text(posicaoelevador3,(maxy/20)*6, 10)
    end
    if(e3p==9)
        text(posicaoelevador3,(maxy/20)*6, 9)
    end
    if(e3p==8)
        text(posicaoelevador3,(maxy/20)*6, 8)
    end
    if(e3p==7)

```

```

    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==6)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==5)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==4)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==3)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==2)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==1)
    text(posicaoelevador3,(maxy/20)*6, )
end
i:(e3p==0)
    text(posicaoelevador3,(maxy/20)*6, )
end
end

i:(e3a02_ult==1)
i:(e3p==12)
    text(posicaoelevador3,(maxy/20)*5, 12)
end
i:(e3p==11)
    text(posicaoelevador3,(maxy/20)*5, 11)
end
i:(e3p==10)
    text(posicaoelevador3,(maxy/20)*5, 10)
end
i:(e3p==9)
    text(posicaoelevador3,(maxy/20)*5, 9)
end
i:(e3p==8)
    text(posicaoelevador3,(maxy/20)*5, 8)
end
i:(e3p==7)
    text(posicaoelevador3,(maxy/20)*5, 7)
end
i:(e3p==6)
    text(posicaoelevador3,(maxy/20)*5, 6)
end
i:(e3p==5)
    text(posicaoelevador3,(maxy/20)*5, 5)
end
i:(e3p==4)
    text(posicaoelevador3,(maxy/20)*5, 4)
end
i:(e3p==3)
    text(posicaoelevador3,(maxy/20)*5, 3)
end
i:(e3p==2)
    text(posicaoelevador3,(maxy/20)*5, 2)
end
i:(e3p==1)
    text(posicaoelevador3,(maxy/20)*5, 1)
end
i:(e3p==0)
    text(posicaoelevador3,(maxy/20)*5, 0)
end
end

i:(e3a01_ult==1)
i:(e3p==12)
    text(posicaoelevador3,(maxy/20)*4, 12)
end
i:(e3p==11)
    text(posicaoelevador3,(maxy/20)*4, 11)
end
i:(e3p==10)
    text(posicaoelevador3,(maxy/20)*4, 10)
end
i:(e3p==9)
    text(posicaoelevador3,(maxy/20)*4, 9)
end
i:(e3p==8)
    text(posicaoelevador3,(maxy/20)*4, 8)
end
i:(e3p==7)
    text(posicaoelevador3,(maxy/20)*4, 7)
end
i:(e3p==6)
    text(posicaoelevador3,(maxy/20)*4, 6)
end
i:(e3p==5)
    text(posicaoelevador3,(maxy/20)*4, 5)
end
i:(e3p==4)
    text(posicaoelevador3,(maxy/20)*4, 4)
end
i:(e3p==3)
    text(posicaoelevador3,(maxy/20)*4, 3)
end

```

```

    text(posicaoelevador3,(maxy/20)*4, )
  end
  (e3p==2)
  text(posicaoelevador3,(maxy/20)*4, )
end
  (e3p==1)
  text(posicaoelevador3,(maxy/20)*4, )
end
  (e3p==0)
  text(posicaoelevador3,(maxy/20)*4, )
end
end

if(e3a00_ult==1)
  if(e3p==12)
    text(posicaoelevador3,(maxy/20)*3, 12)
  end
  (e3p==11)
  text(posicaoelevador3,(maxy/20)*3, 11)
end
  (e3p==10)
  text(posicaoelevador3,(maxy/20)*3, 10)
end
  (e3p==9)
  text(posicaoelevador3,(maxy/20)*3, 9)
end
  (e3p==8)
  text(posicaoelevador3,(maxy/20)*3, 8)
end
  (e3p==7)
  text(posicaoelevador3,(maxy/20)*3, 7)
end
  (e3p==6)
  text(posicaoelevador3,(maxy/20)*3, 6)
end
  (e3p==5)
  text(posicaoelevador3,(maxy/20)*3, 5)
end
  (e3p==4)
  text(posicaoelevador3,(maxy/20)*3, 4)
end
  (e3p==3)
  text(posicaoelevador3,(maxy/20)*3, 3)
end
  (e3p==2)
  text(posicaoelevador3,(maxy/20)*3, 2)
end
  (e3p==1)
  text(posicaoelevador3,(maxy/20)*3, 1)
end
  (e3p==0)
  text(posicaoelevador3,(maxy/20)*3, 0)
end
end

if(e3am01_ult==1)
  if(e3p==12)
    text(posicaoelevador3,(maxy/20)*2, 12)
  end
  (e3p==11)
  text(posicaoelevador3,(maxy/20)*2, 11)
end
  (e3p==10)
  text(posicaoelevador3,(maxy/20)*2, 10)
end
  (e3p==9)
  text(posicaoelevador3,(maxy/20)*2, 9)
end
  (e3p==8)
  text(posicaoelevador3,(maxy/20)*2, 8)
end
  (e3p==7)
  text(posicaoelevador3,(maxy/20)*2, 7)
end
  (e3p==6)
  text(posicaoelevador3,(maxy/20)*2, 6)
end
  (e3p==5)
  text(posicaoelevador3,(maxy/20)*2, 5)
end
  (e3p==4)
  text(posicaoelevador3,(maxy/20)*2, 4)
end
  (e3p==3)
  text(posicaoelevador3,(maxy/20)*2, 3)
end
  (e3p==2)
  text(posicaoelevador3,(maxy/20)*2, 2)
end
  (e3p==1)
  text(posicaoelevador3,(maxy/20)*2, 1)
end
  (e3p==0)
  text(posicaoelevador3,(maxy/20)*2, 0)
end
end

```

```

i(e3a102_ult==1)
i(e3p==12)
text(posicaoelevador3,(maxy/20)*1,12)
end
i(e3p==11)
text(posicaoelevador3,(maxy/20)*1,11)
end
i(e3p==10)
text(posicaoelevador3,(maxy/20)*1,10)
end
i(e3p==9)
text(posicaoelevador3,(maxy/20)*1,9)
end
i(e3p==8)
text(posicaoelevador3,(maxy/20)*1,8)
end
i(e3p==7)
text(posicaoelevador3,(maxy/20)*1,7)
end
i(e3p==6)
text(posicaoelevador3,(maxy/20)*1,6)
end
i(e3p==5)
text(posicaoelevador3,(maxy/20)*1,5)
end
i(e3p==4)
text(posicaoelevador3,(maxy/20)*1,4)
end
i(e3p==3)
text(posicaoelevador3,(maxy/20)*1,3)
end
i(e3p==2)
text(posicaoelevador3,(maxy/20)*1,2)
end
i(e3p==1)
text(posicaoelevador3,(maxy/20)*1,1)
end
i(e3p==0)
text(posicaoelevador3,(maxy/20)*1,0)
end
end

```

*Posicao do Elevador 4 no eixo x
posicaoelevador4=930;

*andar que se encontra o elevador 4

```

i(e4a14_ult==1)
i(e4p==12)
text(posicaoelevador4,(maxy/20)*17,12)
end
i(e4p==11)
text(posicaoelevador4,(maxy/20)*17,11)
end
i(e4p==10)
text(posicaoelevador4,(maxy/20)*17,10)
end
i(e4p==9)
text(posicaoelevador4,(maxy/20)*17,9)
end
i(e4p==8)
text(posicaoelevador4,(maxy/20)*17,8)
end
i(e4p==7)
text(posicaoelevador4,(maxy/20)*17,7)
end
i(e4p==6)
text(posicaoelevador4,(maxy/20)*17,6)
end
i(e4p==5)
text(posicaoelevador4,(maxy/20)*17,5)
end
i(e4p==4)
text(posicaoelevador4,(maxy/20)*17,4)
end
i(e4p==3)
text(posicaoelevador4,(maxy/20)*17,3)
end
i(e4p==2)
text(posicaoelevador4,(maxy/20)*17,2)
end
i(e4p==1)
text(posicaoelevador4,(maxy/20)*17,1)
end
i(e4p==0)
text(posicaoelevador4,(maxy/20)*17,0)
end
end
end

i(e4a13_ult==1)
i(e4p==12)
text(posicaoelevador4,(maxy/20)*16,12)
end
i(e4p==11)

```

```

    text(posicaoelevador4,(maxy/20)*16,11)
end
if(e4p==10)
    text(posicaoelevador4,(maxy/20)*16,10)
end
if(e4p==9)
    text(posicaoelevador4,(maxy/20)*16,9)
end
if(e4p==8)
    text(posicaoelevador4,(maxy/20)*16,8)
end
if(e4p==7)
    text(posicaoelevador4,(maxy/20)*16,7)
end
if(e4p==6)
    text(posicaoelevador4,(maxy/20)*16,6)
end
if(e4p==5)
    text(posicaoelevador4,(maxy/20)*16,5)
end
if(e4p==4)
    text(posicaoelevador4,(maxy/20)*16,4)
end
if(e4p==3)
    text(posicaoelevador4,(maxy/20)*16,3)
end
if(e4p==2)
    text(posicaoelevador4,(maxy/20)*16,2)
end
if(e4p==1)
    text(posicaoelevador4,(maxy/20)*16,1)
end
if(e4p==0)
    text(posicaoelevador4,(maxy/20)*16,0)
end
end

if(e4a12_ult==1)
    if(e4p==12)
        text(posicaoelevador4,(maxy/20)*15,12)
    end
    if(e4p==11)
        text(posicaoelevador4,(maxy/20)*15,11)
    end
    if(e4p==10)
        text(posicaoelevador4,(maxy/20)*15,10)
    end
    if(e4p==9)
        text(posicaoelevador4,(maxy/20)*15,9)
    end
    if(e4p==8)
        text(posicaoelevador4,(maxy/20)*15,8)
    end
    if(e4p==7)
        text(posicaoelevador4,(maxy/20)*15,7)
    end
    if(e4p==6)
        text(posicaoelevador4,(maxy/20)*15,6)
    end
    if(e4p==5)
        text(posicaoelevador4,(maxy/20)*15,5)
    end
    if(e4p==4)
        text(posicaoelevador4,(maxy/20)*15,4)
    end
    if(e4p==3)
        text(posicaoelevador4,(maxy/20)*15,3)
    end
    if(e4p==2)
        text(posicaoelevador4,(maxy/20)*15,2)
    end
    if(e4p==1)
        text(posicaoelevador4,(maxy/20)*15,1)
    end
    if(e4p==0)
        text(posicaoelevador4,(maxy/20)*15,0)
    end
end
end

if(e4a11_ult==1)
    if(e4p==12)
        text(posicaoelevador4,(maxy/20)*14,12)
    end
    if(e4p==11)
        text(posicaoelevador4,(maxy/20)*14,11)
    end
    if(e4p==10)
        text(posicaoelevador4,(maxy/20)*14,10)
    end
    if(e4p==9)
        text(posicaoelevador4,(maxy/20)*14,9)
    end
    if(e4p==8)
        text(posicaoelevador4,(maxy/20)*14,8)
    end
    if(e4p==7)

```

```

    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==6)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==5)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==4)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==3)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==2)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==1)
    text(posicaoelevador4,(maxy/20)*14, )
end
!(e4p==0)
    text(posicaoelevador4,(maxy/20)*14, )
end
end

!(e4a10_ult==1)
!(e4p==12)
    text(posicaoelevador4,(maxy/20)*13, 12)
end
!(e4p==11)
    text(posicaoelevador4,(maxy/20)*13, 11)
end
!(e4p==10)
    text(posicaoelevador4,(maxy/20)*13, 10)
end
!(e4p==9)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==8)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==7)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==6)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==5)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==4)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==3)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==2)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==1)
    text(posicaoelevador4,(maxy/20)*13, )
end
!(e4p==0)
    text(posicaoelevador4,(maxy/20)*13, )
end
end

!(e4a09_ult==1)
!(e4p==12)
    text(posicaoelevador4,(maxy/20)*12, 12)
end
!(e4p==11)
    text(posicaoelevador4,(maxy/20)*12, 11)
end
!(e4p==10)
    text(posicaoelevador4,(maxy/20)*12, 10)
end
!(e4p==9)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==8)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==7)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==6)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==5)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==4)
    text(posicaoelevador4,(maxy/20)*12, )
end
!(e4p==3)

```

```

      text(posicaoelevador4,(maxy/20)*12, )
    end
  i(e4p==2)
    text(posicaoelevador4,(maxy/20)*12, )
  end
  i(e4p==1)
    text(posicaoelevador4,(maxy/20)*12, )
  end
  i(e4p==0)
    text(posicaoelevador4,(maxy/20)*12, )
  end
end

i(e4a08_ult==1)
  i(e4p==12)
    text(posicaoelevador4,(maxy/20)*11, 12)
  end
  i(e4p==11)
    text(posicaoelevador4,(maxy/20)*11, 11)
  end
  i(e4p==10)
    text(posicaoelevador4,(maxy/20)*11, 10)
  end
  i(e4p==9)
    text(posicaoelevador4,(maxy/20)*11, 9)
  end
  i(e4p==8)
    text(posicaoelevador4,(maxy/20)*11, 8)
  end
  i(e4p==7)
    text(posicaoelevador4,(maxy/20)*11, 7)
  end
  i(e4p==6)
    text(posicaoelevador4,(maxy/20)*11, 6)
  end
  i(e4p==5)
    text(posicaoelevador4,(maxy/20)*11, 5)
  end
  i(e4p==4)
    text(posicaoelevador4,(maxy/20)*11, 4)
  end
  i(e4p==3)
    text(posicaoelevador4,(maxy/20)*11, 3)
  end
  i(e4p==2)
    text(posicaoelevador4,(maxy/20)*11, 2)
  end
  i(e4p==1)
    text(posicaoelevador4,(maxy/20)*11, 1)
  end
  i(e4p==0)
    text(posicaoelevador4,(maxy/20)*11, 0)
  end
end

i(e4a07_ult==1)
  i(e4p==12)
    text(posicaoelevador4,(maxy/20)*10, 12)
  end
  i(e4p==11)
    text(posicaoelevador4,(maxy/20)*10, 11)
  end
  i(e4p==10)
    text(posicaoelevador4,(maxy/20)*10, 10)
  end
  i(e4p==9)
    text(posicaoelevador4,(maxy/20)*10, 9)
  end
  i(e4p==8)
    text(posicaoelevador4,(maxy/20)*10, 8)
  end
  i(e4p==7)
    text(posicaoelevador4,(maxy/20)*10, 7)
  end
  i(e4p==6)
    text(posicaoelevador4,(maxy/20)*10, 6)
  end
  i(e4p==5)
    text(posicaoelevador4,(maxy/20)*10, 5)
  end
  i(e4p==4)
    text(posicaoelevador4,(maxy/20)*10, 4)
  end
  i(e4p==3)
    text(posicaoelevador4,(maxy/20)*10, 3)
  end
  i(e4p==2)
    text(posicaoelevador4,(maxy/20)*10, 2)
  end
  i(e4p==1)
    text(posicaoelevador4,(maxy/20)*10, 1)
  end
  i(e4p==0)
    text(posicaoelevador4,(maxy/20)*10, 0)
  end
end
end

```

```

    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==8)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==7)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==6)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==5)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==4)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==3)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==2)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==1)
    text(posicaoelevador4,(maxy/20)*7, )
end
if(e4p==0)
    text(posicaoelevador4,(maxy/20)*7, )
end

```

```

i(e4a06_ult==1)
  i(e4p==12)
    text(posicaoelevador4,(maxy/20)*9,12)
  end
  i(e4p==11)
    text(posicaoelevador4,(maxy/20)*9,11)
  end
  i(e4p==10)
    text(posicaoelevador4,(maxy/20)*9,10)
  end
  i(e4p==9)
    text(posicaoelevador4,(maxy/20)*9,9)
  end
  i(e4p==8)
    text(posicaoelevador4,(maxy/20)*9,8)
  end
  i(e4p==7)
    text(posicaoelevador4,(maxy/20)*9,7)
  end
  i(e4p==6)
    text(posicaoelevador4,(maxy/20)*9,6)
  end
  i(e4p==5)
    text(posicaoelevador4,(maxy/20)*9,5)
  end
  i(e4p==4)
    text(posicaoelevador4,(maxy/20)*9,4)
  end
  i(e4p==3)
    text(posicaoelevador4,(maxy/20)*9,3)
  end
  i(e4p==2)
    text(posicaoelevador4,(maxy/20)*9,2)
  end
  i(e4p==1)
    text(posicaoelevador4,(maxy/20)*9,1)
  end
  i(e4p==0)
    text(posicaoelevador4,(maxy/20)*9,0)
  end
end

i(e4a05_ult==1)
  i(e4p==12)
    text(posicaoelevador4,(maxy/20)*8,12)
  end
  i(e4p==11)
    text(posicaoelevador4,(maxy/20)*8,11)
  end
  i(e4p==10)
    text(posicaoelevador4,(maxy/20)*8,10)
  end
  i(e4p==9)
    text(posicaoelevador4,(maxy/20)*8,9)
  end
  i(e4p==8)
    text(posicaoelevador4,(maxy/20)*8,8)
  end
  i(e4p==7)
    text(posicaoelevador4,(maxy/20)*8,7)
  end
  i(e4p==6)
    text(posicaoelevador4,(maxy/20)*8,6)
  end
  i(e4p==5)
    text(posicaoelevador4,(maxy/20)*8,5)
  end
  i(e4p==4)
    text(posicaoelevador4,(maxy/20)*8,4)
  end
  i(e4p==3)
    text(posicaoelevador4,(maxy/20)*8,3)
  end
  i(e4p==2)
    text(posicaoelevador4,(maxy/20)*8,2)
  end
  i(e4p==1)
    text(posicaoelevador4,(maxy/20)*8,1)
  end
  i(e4p==0)
    text(posicaoelevador4,(maxy/20)*8,0)
  end
end

i(e4a04_ult==1)
  i(e4p==12)
    text(posicaoelevador4,(maxy/20)*7,12)
  end
  i(e4p==11)
    text(posicaoelevador4,(maxy/20)*7,11)
  end
  i(e4p==10)
    text(posicaoelevador4,(maxy/20)*7,10)
  end
  i(e4p==9)

```

```

(e4a03_ult:=1)
  if(e4p==12)
    text(posicaoelevador4,(maxy/20)*6,12)
  end
  if(e4p==11)
    text(posicaoelevador4,(maxy/20)*6,11)
  end
  if(e4p==10)
    text(posicaoelevador4,(maxy/20)*6,10)
  end
  if(e4p==9)
    text(posicaoelevador4,(maxy/20)*6,9)
  end
  if(e4p==8)
    text(posicaoelevador4,(maxy/20)*6,8)
  end
  if(e4p==7)
    text(posicaoelevador4,(maxy/20)*6,7)
  end
  if(e4p==6)
    text(posicaoelevador4,(maxy/20)*6,6)
  end
  if(e4p==5)
    text(posicaoelevador4,(maxy/20)*6,5)
  end
  if(e4p==4)
    text(posicaoelevador4,(maxy/20)*6,4)
  end
  if(e4p==3)
    text(posicaoelevador4,(maxy/20)*6,3)
  end
  if(e4p==2)
    text(posicaoelevador4,(maxy/20)*6,2)
  end
  if(e4p==1)
    text(posicaoelevador4,(maxy/20)*6,1)
  end
  if(e4p==0)
    text(posicaoelevador4,(maxy/20)*6,0)
  end
end

(e4a02_ult:=1)
  if(e4p==12)
    text(posicaoelevador4,(maxy/20)*5,12)
  end
  if(e4p==11)
    text(posicaoelevador4,(maxy/20)*5,11)
  end
  if(e4p==10)
    text(posicaoelevador4,(maxy/20)*5,10)
  end
  if(e4p==9)
    text(posicaoelevador4,(maxy/20)*5,9)
  end
  if(e4p==8)
    text(posicaoelevador4,(maxy/20)*5,8)
  end
  if(e4p==7)
    text(posicaoelevador4,(maxy/20)*5,7)
  end
  if(e4p==6)
    text(posicaoelevador4,(maxy/20)*5,6)
  end
  if(e4p==5)
    text(posicaoelevador4,(maxy/20)*5,5)
  end
  if(e4p==4)
    text(posicaoelevador4,(maxy/20)*5,4)
  end
  if(e4p==3)
    text(posicaoelevador4,(maxy/20)*5,3)
  end
  if(e4p==2)
    text(posicaoelevador4,(maxy/20)*5,2)
  end
  if(e4p==1)
    text(posicaoelevador4,(maxy/20)*5,1)
  end
  if(e4p==0)
    text(posicaoelevador4,(maxy/20)*5,0)
  end
end

```

```

    text(posicaoelevador4,(maxy/20)*5, )
end
i'(e4p==4)
    text(posicaoelevador4,(maxy/20)*5, )
end
i'(e4p==3)
    text(posicaoelevador4,(maxy/20)*5, )
end
i'(e4p==2)
    text(posicaoelevador4,(maxy/20)*5, )
end
i'(e4p==1)
    text(posicaoelevador4,(maxy/20)*5, )
end
i'(e4p==0)
    text(posicaoelevador4,(maxy/20)*5, )
end
end

i'(e4a01_ult==1)
i'(e4p==12)
    text(posicaoelevador4,(maxy/20)*4, 12)
end
i'(e4p==11)
    text(posicaoelevador4,(maxy/20)*4, 11)
end
i'(e4p==10)
    text(posicaoelevador4,(maxy/20)*4, 10)
end
i'(e4p==9)
    text(posicaoelevador4,(maxy/20)*4, 9)
end
i'(e4p==8)
    text(posicaoelevador4,(maxy/20)*4, 8)
end
i'(e4p==7)
    text(posicaoelevador4,(maxy/20)*4, 7)
end
i'(e4p==6)
    text(posicaoelevador4,(maxy/20)*4, 6)
end
i'(e4p==5)
    text(posicaoelevador4,(maxy/20)*4, 5)
end
i'(e4p==4)
    text(posicaoelevador4,(maxy/20)*4, 4)
end
i'(e4p==3)
    text(posicaoelevador4,(maxy/20)*4, 3)
end
i'(e4p==2)
    text(posicaoelevador4,(maxy/20)*4, 2)
end
i'(e4p==1)
    text(posicaoelevador4,(maxy/20)*4, 1)
end
i'(e4p==0)
    text(posicaoelevador4,(maxy/20)*4, 0)
end
end

i'(e4a00_ult==1)
i'(e4p==12)
    text(posicaoelevador4,(maxy/20)*3, 12)
end
i'(e4p==11)
    text(posicaoelevador4,(maxy/20)*3, 11)
end
i'(e4p==10)
    text(posicaoelevador4,(maxy/20)*3, 10)
end
i'(e4p==9)
    text(posicaoelevador4,(maxy/20)*3, 9)
end
i'(e4p==8)
    text(posicaoelevador4,(maxy/20)*3, 8)
end
i'(e4p==7)
    text(posicaoelevador4,(maxy/20)*3, 7)
end
i'(e4p==6)
    text(posicaoelevador4,(maxy/20)*3, 6)
end
i'(e4p==5)
    text(posicaoelevador4,(maxy/20)*3, 5)
end
i'(e4p==4)
    text(posicaoelevador4,(maxy/20)*3, 4)
end
i'(e4p==3)
    text(posicaoelevador4,(maxy/20)*3, 3)
end
i'(e4p==2)
    text(posicaoelevador4,(maxy/20)*3, 2)
end
i'(e4p==1)

```

```

    text(posicaoelevador4,(maxy/20)*3, )
end
i'(e4p==0)
    text(posicaoelevador4,(maxy/20)*3, )
end
end

i'(e4am01_ult==1)
i'(e4p==12)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==11)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==10)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==9)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==8)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==7)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==6)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==5)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==4)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==3)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==2)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==1)
    text(posicaoelevador4,(maxy/20)*2, )
end
i'(e4p==0)
    text(posicaoelevador4,(maxy/20)*2, )
end
end

i'(e4am02_ult==1)
i'(e4p==12)
    text(posicaoelevador4,(maxy/20)*1, 12)
end
i'(e4p==11)
    text(posicaoelevador4,(maxy/20)*1, 11)
end
i'(e4p==10)
    text(posicaoelevador4,(maxy/20)*1, 10)
end
i'(e4p==9)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==8)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==7)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==6)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==5)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==4)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==3)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==2)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==1)
    text(posicaoelevador4,(maxy/20)*1, )
end
i'(e4p==0)
    text(posicaoelevador4,(maxy/20)*1, )
end
end
end

```