UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

MÁRIO LINS ESTEVAM DE BARROS
RICHARD SASSOON
THOMAS CARLYLE MINETTO VILARINHO

Sistema de Controle para Elevadores Inteligentes

v. 1

São Paulo
2005

MÁRIO LINS ESTEVAM DE BARROS
RICHARD SASSOON
THOMAS CARLYLE MINETTO VILARINHO

Sistema de Controle para Elevadores Inteligentes

Projeto de Formatura apresentado à disciplina PCS 2502 – Laboratório de Projeto de Formatura II – Escola Politécnica da Universidade de São Paulo.

Orientadora: Profa. Dra. Anna Helena Reali Costa

v. 1

São Paulo
2005

DEDICATÓRIA

Às nossas famílias, namoradas, amigos, aos professores da Escola Politécnica pelo apoio, paciência e encorajamento dedicados.

A Mario Estevam de Barros (em memória).

AGRADECIMENTOS

À profa. Dra. Anna Helena Reali Costa, pela atenção e apoio durante o processo de definição e orientação.

À Escola Politécnica, pela infra-estrutura fornecida durantes os cinco anos de curso.

Aos funcionários hospitalares pela atenção e prontidão durante nossas pesquisas.

RESUMO

DE BARROS, M. L. E., SASSOON, R., VILARINHO, T. C. M. **Sistema de controle para elevadores inteligentes**. 2005. 114 f. Projeto de Formatura apresentado à disciplina PCS 2502 – Laboratório de Projeto de Formatura II – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

Os sistemas de controle de elevadores são importantes por visarem de otimizar diferentes variáveis referentes aos níveis de serviço e de desempenho, muitas vezes conflitantes entre si, e por isso são sistemas de alta complexidade. Este trabalho propõe uma nova abordagem para os sistemas de controle de forma a serem obtidos melhores resultados em comparação a alguns métodos existentes. A abordagem sustenta-se nas vantagens que podem ser obtidas por meio da lógica nebulosa. As principais variáveis envolvidas são identificadas e postas em testes de desempenho e melhoria, de acordo com diferentes modelagens do sistema proposto, cujo núcleo é comum, mas apresenta um módulo que pode ser adaptado para diferentes ambientes. É apresentado no trabalho um simulador de edifícios que foi desenvolvido com o objetivo de auxiliar no desenvolvimento do sistema.

Palavras-chave: Lógica Nebulosa, Sistemas de Controle de Elevadores, Simulação a Eventos Discretos.

ABSTRACT

DE BARROS, M. L. E., SASSOON, R., VILARINHO, T. C. M. **Control System for Intelligent Elevators**. 2005. 114 f. Graduation Project presented to the discipline PCS 2502 – Graduation Project Laboratory II – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

Elevator control systems are important since they must optimize different variables related to conflicting parameters such as quality of service and performance. This work proposes a new approach to elevator control systems which produces better results comparatively to some existing methods. The approach is based on advantages offered by the use of fuzzy logic. The main variables are identified and tested regarding performance and improvement, for different models of the proposed system, whose core is common, but that has a module that can be adapted to different environments. A building elevator simulator that has been developed for supporting the system implementation is also presented.

Keywords: Fuzzy Logic, Elevator Control Systems, Discrete Event Simulation.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

ACA	*Adaptive Call Allocation*;
API	*Application Program Interface*;
AWT	*Avarage Waiting Time*;
DP	*Down-peak*;
DISP	disponibilidade da cabine;
FATU	fator de utilização;
IDE	*Integrated Development Environment*;
IFC	*Interconnected Full Collective*;
L	*Large*;
NL	*Negative Large*;
NM	*Megative Medium*;
OC	Ocupação da cabine;
PC	*Power Consumption*;
PL	*Positive Large*;
PM	*Positive Medium*;
PF	Processador *Fuzzy*;
S	*Small*;
TEA	tempo de espera por atendimento;
UML	*Unified Modeling Language*;
UP	*Up-peak*;
VL	*Very Large*;
VS	*Very Small*;
XML	*Extensible Markup Language*;
Z	*Zero*.

SUMÁRIO

1 Introdução:

À medida que os edifícios tornam-se cada vez mais sofisticados, expandindo-se vertical e horizontalmente, a busca por fatores de qualidade de serviço e por maior produtividade na utilização do conjunto de elevadores torna-se cada vez mais importante. Estes objetivos, por sua vez, podem ser conflitantes, uma vez que, a agilidade do atendimento, tempos de viagem curtos e o conforto do usuário muitas vezes não são compatíveis com a procura por redução do consumo energético dos edifícios e de gastos com manutenção.

Estes conflitos refletem a complexidade envolvida na solução de um sistema de controle para elevadores cuja quantidade de variáveis é extensa e a formulação de uma solução-ótima ainda não existe. Este fato denota a importância dada ao assunto, justificada pela gama de pesquisas publicadas referentes a otimizações de sistemas de controle de elevadores, baseadas nos mais diversos métodos e tecnologias (1)(2)(3)(4). Este trabalho visa o estudo destas técnicas e também de tráfegos para certas categorias de edifícios.

1.1 Objetivo

O objetivo principal deste trabalho é a implementação de um Sistema de Controle de Elevadores Inteligentes.

Os objetivos secundários são: formação de recursos-humanos na área de estudo, consolidação do aprendizado de técnicas nebulosas, implementação de um simulador guiado por eventos, e estudo de um sistema de controle de elevadores, bem como o comportamento dos

elevadores de acordo com o objetivo do sistema de controle (diminuição do tempo de espera dos passageiros, economia de energia, etc.).

1.2 Organização do Trabalho

O trabalho está estruturado de forma que no capítulo 2 são descritos os principais sistemas, modelos e técnicas de controle de elevadores, de modo a ambientar o leitor com o que vem sendo pesquisado nos últimos tempos, ou já existe no mercado. É então apresentada, no capítulo 3, uma descrição panorâmica do sistema proposto para o controle de elevadores, além de alguns conceitos inerentes à solução apresentada.

O capítulo 4 aprofunda a especificação do sistema de controle, apresentando sua arquitetura, a modelagem do sistema nebuloso utilizado, as demais técnicas utilizadas no projeto, além da descrição da metodologia de testes do sistema de controle. No capítulo 5, descreve-se o simulador de elevadores, desenvolvido com o intuito de avaliar a eficiência do sistema de controle, e os testes realizados a fim de validar o simulador.

Em seguida, no capítulo 6, é apresentada a metodologia de desenvolvimento usada ao longo do projeto; são justificadas as escolhas tecnológicas envolvidas no mesmo, bem como os recursos e infra-estrutura necessários para o seu desenvolvimento. Ao longo do capítulo 7 são descritos os testes realizados com o sistema de controle construído, apresentando desde a modelagem dos dados de entrada do sistema até a especificação das referências utilizadas para a comparação e verificação da eficiência do sistema implementado. Por fim, no capítulo 8, são apresentadas as conclusões finais do sistema desenvolvido.

2 Panorama da Área

A problemática dos controles de elevadores está em estudo há um longo período de tempo devido à sua difícil solução e ao grande número de aspectos, ora conflitantes, que podem ser melhorados no uso dos elevadores.

Os primeiros métodos de controle de alocações em elevadores baseavam-se no algoritmo chamado *Single Call Control* (9), que hoje em dia, possui alguma utilidade apenas para pequenos edifícios com pouco tráfego.

Há duas formas de implementá-lo, com um botão ou com dois botões de chamada por andar. No caso onde há um botão por andar, se o elevador estiver vazio, ele atende uma eventual *Hall Call*, chamada externa (ou seja, chamada realizada em um dos painéis que se encontram nos andares, fora das cabines). E caso contrário, ignora esta chamada. Enquanto, na implementação de dois botões por andar (um para subir e outro para descer), se o elevador estiver vazio, ele atende uma eventual *Hall Call*. Se ele não estiver, atende apenas quando a *Hall Call* sinalizar a mesma direção que o elevador está seguindo no momento. Tal controle é bastante primitivo e a sua única vantagem é evitar que as pessoas viagem no sentido oposto ao seu destino (antes de chegarem em seus andares alvo) e diminuir o número de chamadas refeitas devido a uma lotação das cabines, uma vez que a menos que o elevador esteja vazio, o atendimento se dá no mesmo sentido de viajem do elevador (fato que diminui a probabilidade de existirem chamadas atentidas por cabines lotadas).

O método mais comum de controle de grupo de elevadores corresponde ao sistema *Interconnected Full Collective (IFC)* (1). Este foi o primeiro método significativo de alocação de

cabines de elevadores. Neste sistema, os elevadores atendem as chamadas mais próximas de si, levando em conta o sentido de viagem. Ou seja, se uma chamada está no sentido oposto ao do elevador, ele toma como distância, a distância até o último ponto a ser atendido somada com a distância deste último ponto ao ponto que realizou a chamada. As chamadas feitas internamente (por meio do painel localizado dentro da cabine) ao elevador são sempre atendidas em ordem seqüencial, e, após servir todas as chamadas no sentido de viagem, a cabine vai até o andar mais distante no sentido oposto.

Outro algoritmo bastante utilizado é o *Selective Collective*, que é muito semelhante ao IFC, com a diferença de ignorar chamadas na direção oposta, ao invés de calcular sua distância. Ambos atuam apenas de modo a diminuir o tempo de espera de atendimento, e ainda assim, de uma maneira um tanto superficial, não pesando o tempo de trânsito, o tempo gasto com as chamadas internas, a chance de uma chamada encontrar a cabine lotada, entre outros fatores que afetam o tempo de espera; o que leva a algumas escolhas equivocadas.

Os avanços das pesquisas na área de Inteligência Artificial promoveram o aparecimento de métodos de controle de elevadores, baseados em redes neurais e em lógica nebulosa (2), que aprendem o padrão de tráfego dos edifícios e adaptam-se a estes (3).

Os métodos adaptativos oferecem condições para que o sistema seja mais flexível e robusto, visto que ele estaria se adaptando conforme os padrões de tráfego observado. Destes métodos, destaca-se a aprendizagem por reforço com múltiplos agentes (4), onde cada agente usa uma rede neural para guardar suas estimativas de ações, sendo que esta rede pode, ou não, ser compartilhada. O grupo de agentes recebe reforços globais que são ruidosos na perspectiva do agente. Desta forma, tal abordagem combina vantagens da aprendizagem por reforço (em que há o aprendizado com base apenas nas próprias experiências) com a vantagem da multiplicidade de

agentes, situação onde o recebimento de um reforço de aprendizado torna-se global e guia todo o grupo para o objetivo de maneira mais eficiente.

Tal questão é bastante complexa para técnicas de aprendizado por reforço, visto que esses sistemas operam em um ambiente de muitos estados possíveis e com eventos discretos dinâmicos, sendo que seus estados não são plenamente observáveis e estáticos, uma vez que a taxa de chegada de passageiros muda muito rapidamente. Logo, as melhores técnicas acabam por exigir um elevado nível de processamento computacional. Uma outra dificuldade neste caso é que as ações tomadas pelos agentes provocam conseqüências de longo prazo.

Os sistemas implementados com lógica nebulosa, por sua vez, são dotados de alta flexibilidade, são menos susceptíveis a ruídos e exceções, podem ser modelados mais facilmente levando em conta variáveis distintas e sua implementação não é exclusiva, pode ser implementada aliada com outras formas de controle. Os sistemas que utilizam lógica nebulosa estabelecem patamares nebulosos e regras de inferência que, com base em valores nebulosos de entrada, produzem uma saída nebulosa. Esta, por sua vez, é *defuzzyficada* e então interpretada como a ação a ser tomada no controlador de elevadores.

Também existem sistemas baseados em conhecimento e busca exaustiva de estados (5), que decide os movimentos dos elevadores, ao invés da alocação das chamadas para uma determinada cabine. O algoritmo desse sistema utiliza busca no espaço de estados para achar a melhor solução, valendo-se de uma versão do algoritmo A*, onde o valor de mérito é baseado no desempenho de fato obtido e em um desempenho previsto (calculado). Além disso, o sistema envolve uma câmera externa em cada andar para monitorar e adquirir dados sobre os futuros passageiros.

Cada estado da busca é inferido pelo conjunto total de passageiros e elevadores que constam no sistema em um determinado instante, bem como os valores de seus atributos. A decisão de qual movimento cada elevador efetuará e, conseqüentemente, para qual estado ele transitará, é baseada em um fator de mérito Q, que deve ser olhado tanto do ponto de vista dos passageiros nos andares, como do ponto de vista dos passageiros nos elevadores.

Este algoritmo, de acordo com experimentos realizados pelos próprios idealizadores, apresentou, nos horários de pico, desempenho semelhante ao algoritmo *Selective Collective*, e melhor desempenho para outros horários, onde a distribuição das chamadas é mais aleatória. Logo, não representa uma solução muito boa, haja vista que envolveria um elevado número de câmeras (ao menos uma por andar) para um resultado não muito melhor que o algoritmo *Selective Collective*, que possui implementação mais simples e não necessita de processamento de imagens.

A maior parte dos sistemas tem enfoque no *Hall Call Time*, que corresponde ao tempo que a chamada externa leva para ser atendida por um elevador. No entanto, alguns sistemas não esperam que o valor médio do histórico desta variável se torne grande o bastante para, então, alocar as chamadas, visando diminuir seu valor médio. Tais sistemas tentam evitar que estes tempos cresçam demasiadamente, utilizando-se de técnicas que estimam os *Hall Call Times*. Assim, quando uma destas estimativas torna-se maior que um certo patamar, o controle intercede, ou alterando a rota das cabines a fim de que o *Hall Call Time* crítico, isto é, o *Hall Call Time* superior ao patamar, não seja atingido, ou, simplesmente, usando a estimativa destes valores para fundamentar uma determinada lógica de alocação de chamadas.

A técnica apresentada é utilizada em muitos sistemas de controle, inclusive o sistema desenvolvido neste projeto, que nela se baseia para o cálculo de uma de suas variáveis. No

entanto, há uma grande dificuldade em estimar tais valores visto que é impossível precisar o caminho a ser realizado por uma *Hall Call* ainda não atendida, mas já alocada à cabine. Por isso, o que a maioria dos sistemas, assim como o deste trabalho, faz é simplificar a estimativa usando apenas dados concretos e, conseqüentemente, o tempo mínimo de atendimento, que parte do pressuposto que todas as chamadas externas (realizadas por meio do painel localizado nos andares) alocadas gerarão chamadas internas (realizadas dentro da cabine, após o passageiro entrar na mesma) cujo destino já será atendido pela fila de chamadas corrente.

Há técnicas mais específicas que atuam para impedir certos comportamentos que acarretam em altos tempos de espera, ou outros valores de variáveis que sejam considerados ruins no desempenho do controle. Um desses comportamentos, ao qual muitos sistemas tentam combater, é o *bunching*, fenômeno no qual as cabines trafegam paralelamente, disputando chamadas dos mesmos andares (6). Tais técnicas encarregam-se de enviar elevadores para o térreo esporadicamente. Desta forma, aumentam a taxa de chamadas atendidas com sucesso, mas geram maior consumo de energia, muitas vezes desnecessário, e, portanto, não foram implementadas pelo sistema projetado.

Uma técnica de controle de elevadores que não foca em um algoritmo específico para a alocação das chamadas é o *Zoning*, onde cada cabine é assinalada para determinadas zonas do prédio, atendendo as chamadas de tais regiões e voltando para as mesmas quando ociosos. O objetivo deste controle é manter as cabines separadas e, conseqüentemente, garantir um baixo valor para o *Interval*, isto é, o tempo médio entre as partidas de elevadores do térreo. Há implementações de sistemas que utilizam *Zoning* baseando-se em zonas estáticas, definidas uma única vez ou agendadas para certos períodos, e zonas dinâmicas (7) (8), definidas com o andamento do sistema, calculadas por funções probabilísticas, algoritmos genéticos (9), ou por

lógica nebulosa (10). Para o controle implementado neste trabalho, optou-se por não realizar o *Zoning*, pois se trata de uma técnica cuja utilização é vantajosa apenas para prédios muito altos e com muitas cabines.

A alocação das chamadas nas filas dos elevadores é mais um fator desafiador na implementação do sistema de controle, uma vez que uma alocação, quando não realizada na última posição da fila, pode interferir nos resultados da lógica usada para alocar as chamadas anteriores. Além disso, como muitas chamadas na fila são ainda chamadas externas, torna-se ainda mais complicado estimar as melhores posições para a alocação das novas chamadas, pois aquelas possuem chamadas internas ainda latentes.

Há técnicas de alocação que analisam todas as possíveis rotas geradas a partir de cada uma das possíveis alocações e seus impactos sobre todas as chamadas já alocadas. Tal análise, a menos que bastante simplificada, é demasiadamente complexa e demanda muito processamento computacional. Desta forma, muitos trabalhos que envolvem tais técnicas apresentam alguns métodos para diminuir o espaço amostral, como: limitar que as chamadas internas sejam atendidas seqüencialmente no sentido de viagem do elevador, sem permitir desvios; ou que um elevador não troque de sentido até que todos os passageiros dentro da cabine tenham sido atendidos, ou ainda impedindo que os elevadores aceitem *Hall Calls* no sentido oposto ao seu atual.

Neste trabalho, com o intuito de otimizar o processo de alocação, foi criada uma lógica que leva a cabine a desempenhar um percurso com um mínimo de *zig-zags* (trocas de sentidos) e, conseqüentemente, favoreça um menor consumo de energia e melhor tempo de atendimento.

No algoritmo ACA (Adaptive Call Allocation) (11), apenas as novas chamadas são alocadas aos melhores vagões (segundo a variável cujo desempenho deseja-se melhorar), de

forma que os agendamentos anteriores permanecem alocados para as mesmas cabines. Com esse método, o tempo de cálculo reduz-se drasticamente até um nível prático para todos os sistemas de elevadores. O problema é que as alocações antigas podem perder suas otimizações quando o agendamento de uma chamada é realizado em um estágio muito prematuro. Assim como no ACA, optou-se por alocar as chamadas estaticamente para as melhores cabines, de forma que as alocações anteriores permaneçam com as mesmas cabines, uma vez que a troca de chamadas dinamicamente entre cabines aumenta bastante a complexidade do problema e é evitada na grande maioria dos sistemas de alocação.

O método de alocação dinâmica, em contrapartida ao ACA, faz suposições quanto à chegada de passageiros e calcula o custo de atender tais suposições, escolhendo a decisão de menor custo para cada *Hall Call* que ocorra, decidindo, então, para cada cabine, se esta continuará seu movimento, começará novo movimento, parará ou mudará de sentido. Funciona de forma similar a uma partida de xadrez onde, a cada evento, são estudadas as possíveis jogadas futuras e a estratégia é então redefinida.

Algoritmos genéticos (12) e otimizações dinâmicas de alocação com aprendizado por reforço (4) têm sido estudados para otimizar as rotas das chamadas existentes dos elevadores. No entanto, ainda não se alcançou nenhum resultado que não necessite de elevado poder computacional ou tempo de treinamento na ordem de anos para os algoritmos.

3 Descrição da Proposta

O sistema aqui proposto deverá ser capaz de exercer o controle de um grupo de elevadores, fornecendo a indicação de qual, dentre os elevadores do grupo, deverá atender uma nova chamada externa, de acordo com as condições atuais do sistema, bem como alocar esta chamada de maneira eficiente na fila de chamadas da cabine selecionada. O sistema deverá atender satisfatoriamente aos requisitos de tráfego identificados no estudo de cada tipo de edifício para o qual for modelado, admitindo ajustes.

Devido às restrições de espaço e recursos para a implementação do projeto, o sistema foi testado em um programa simulador de edifícios, também desenvolvido ao longo do projeto. O simulador oferece facilidades e flexibilidade para o fornecimento de entradas para teste e validação do sistema de controle, assim como o registro de informações sobre os atendimentos prestados pelo sistema.

A escolha da melhor cabine a atender uma determinada requisição será feita por um sistema de lógica nebulosa. Optou-se, neste caso, pelo sistema nebuloso graças às suas vantagens de tolerância de pequenas imprecisões, flexibilidade de adicionar novas funcionalidades, fácil modelagem de funções complexas e a possibilidade de ser integrado com outras técnicas de controle. A lógica nebulosa já é utilizada com sucesso em diversas áreas como classificação de dados, apoio à decisão, visão computacional, controle de processos, entre outros.

Considere a descrição sucinta do procedimento de um sistema nebuloso, dada a seguir. A lógica nebulosa apresenta regras de produção do tipo:

Se <antecedente> *então* <conseqüente>

Onde <antecedente> corresponde ao conjunto de condições (ligadas por conectivos lógicos "E" ou "OU") e <conseqüente>, às ações (correspondentes às saidas).

Para a obtenção do valor que ajude a escolher a ação a ser tomada, alguns passos são necessários:

1. Definição das variáveis lingüísticas, seus conjuntos nebulosos e correspondentes funções de pertinência.

   1.1. Uma variável lingüística representa um conceito ou variável de um problema, como por exemplo, a estatura de uma pessoa. Seus termos primários podem ser representados por conjuntos nebulosos. Termos primários especificam os valores lingüísticos da variável e podem ser, no caso da variável estatura, caracterizados como baixa, mediana, alta.

   1.2. Um conjunto nebuloso A definido no universo de discurso U é caracterizado por uma função de pertinência μA, a qual associa a cada elemento $a \in A$, $A \subset U$ um valor do intervalo real [0 , 1], que representa o grau de pertinência do elemento a no conjunto A. Seguindo o exemplo dado anteriormente, um conjunto nebuloso pode ser aquele que indica estaturas medianas, sendo que um elemento de valor 1.67m poderia possuir um grau de pertinência de 0.6 neste conjunto e de 0.2 no conjunto de estaturas altas.

2. Definição das regras nebulosas – Base de conhecimento;

3. Deve-se então escolher o modelo de inferência, que neste caso foi o modelo de inferência de Mamdani (13), que será explicado a seguir;

4. *Fuzzificação:* Como explicado anteriormente, os valores das variáveis lingüísticas de entrada são *fuzzyficados* por uma função de pertinência. Os valores obtidos, ou graus de pertinência, são então utilizados para a realização da inferência;

5. Inferência: O modelo de Mamdani agrega as regras nebulosas com conectivos lógicos do tipo "OU", sendo que as saídas de cada regra são geradas pela intersecção dos antecedentes por meio de conectivos lógicos do tipo "E". Isto significa que a saída de uma regra é obtida pelo menor grau de pertinência encontrado em seus antecedentes, e que a agregação de todas as regras faça com que os graus de pertinência dos termos primários da variável de saída sejam os maiores possíveis. Tais graus de pertinência geram uma área que deverá ser *deffuzyficada*. A Figura 1, retirada de (13) exemplifica o resultado final da união das saídas das regras, considerando que [6,18] representa o universo de discurso e que "A, M e B" representam os termos primários da variável de saída "Con";

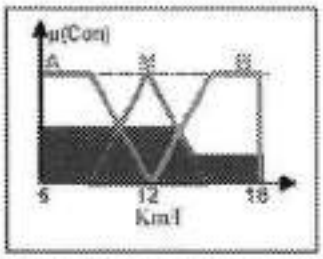

Figura 1 – Área obtida a partir dos graus de pertinência dos termos primários (A, M e B) de uma variável lingüística de saída.

6. *Defuzzyficação:* Com a área obtida na etapa 5, o mais comum é utilizar o método do centro de massa para descobrir um valor real para a saída.

Por fim, são implementados pré-controles para realizar as alocações das chamadas dentro das filas de atendimento de cada uma das cabines. Esses pré-controles atuam selecionando as melhores possibilidades de alocação de chamadas na cabine para que, então, as cabines sejam

avaliadas pelo sistema nebuloso. A implementação desses tipos de pré-controle é usual em sistemas de controle de elevadores mais complexos a fim de reduzir-se a necessidade de processamento computacional, mesmo podendo trazer influências externas com risco de comprometimento parcial dos métodos de controle complementares.

Cada tipo de controle desenvolvido está associado a um determinado pré-controle. As características de cada um são detalhadas no item 4.4.

4 Descrição do Sistema de Controle

Este capítulo tem como objetivo detalhar o sistema de controle e as três implementações realizadas no projeto.

4.1 Arquitetura do Sistema de Controle

A arquitetura do sistema é composta basicamente de três módulos, Árbitro, Processador *Fuzzy*, e Pré-Controle como mostra a Figura 2, sendo que os dois primeiros módulos formam o controle básico, ou Controle.

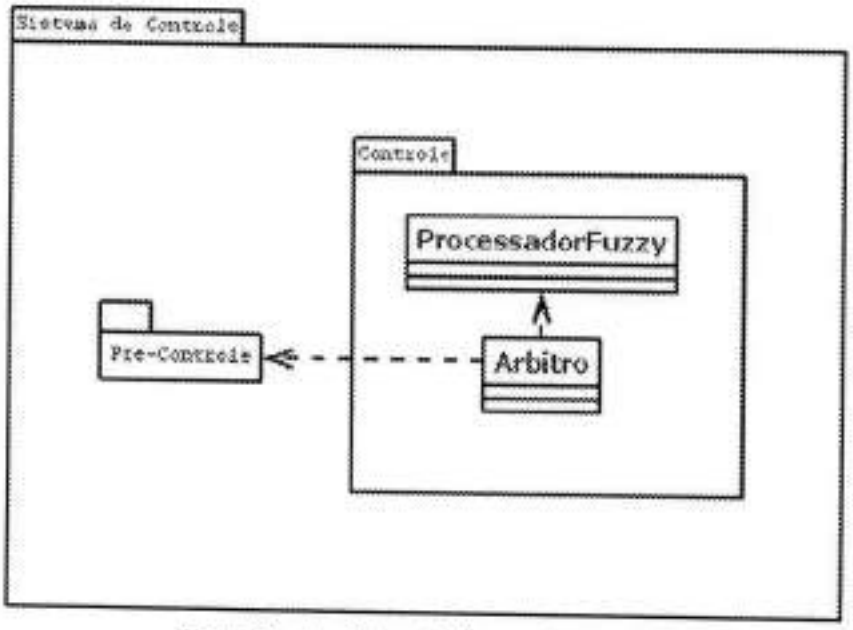

Figura 2 – Arquitetura do sistema de controle.

A seguir, uma descrição de cada módulo:

1. Processador *Fuzzy* (PF): é o responsável por receber as entradas das variáveis do sistema, disparar as regras de controle e gerar uma saída (a disponibilidade da cabine conforme as regras carregadas no processador). As variáveis de entrada utilizadas são denominadas TEA, FATU e OC são detalhadas no item 4.2, assim como a variável de saída, denominada DISP;
2. Árbitro: utiliza-se do PF para disparar as regras para cada uma das cabines. O PF retorna uma saída DISP, ou disponibilidade, para cada cabine. O Árbitro verifica qual delas apresentou o maior DISP, de forma a escolher a cabine mais apta para atender a nova chamada. No caso de uma chamada interna, o Árbitro escolhe a cabine de onde a chamada foi originada;
3. Pré-Controle: é peça fundamental no sistema, uma vez que se responsabiliza pela alocação das chamadas e, conseqüentemente, pelo cálculo da variável TEA. O Pré-Controle é chamado pelo Árbitro sempre que este precisa alocar uma chamada. Tal processo é definido como pseudo-alocação da chamada na fila de atendimento de cada cabine, pois neste instante ainda não foi decidido qual cabine irá atender a chamada. Após a decisão, a fila de atendimento da cabine é efetivamente atualizada pela fila gerada pelo pré-controle.

4.2 Variáveis utilizadas pelo módulo Controle

Foram analisadas algumas variáveis de controle, de acordo com outras implementações de sistemas de controle descritas na literatura, e chegou-se a três variáveis mais relevantes para

serem utilizadas como entradas no sistema proposto, que serão responsáveis por gerar um valor para a variável de saída DISP. São elas TEA, OC e FATU, descritas a seguir:

Tempo esperado para atendimento (TEA): Consiste na estimativa de quanto tempo a cabine em questão vai levar para atender uma chamada. Pode ser calculada somando-se os tempos necessários para atingir o andar da chamada, sendo estes tempos correspondentes ao tempo de trânsito e das paradas intermediárias da cabine. Sua relevância está no fato de que quanto menor for o seu valor, maior será a chance de atender uma chamada rapidamente. Esta é a variável que a maioria dos sistemas utiliza, de forma a reduzir o tempo médio de espera global.

Cálculo do TEA: Para o cálculo da variável TEA, é preciso que uma fila de atendimento para cada cabine tenha sido gerada com a nova chamada sendo pseudo-alocada, conforme descrito no item anterior, de forma que os tempos envolvidos nesta variável possam ser calculados levando a nova chamada em consideração. Esta pseudo-alocação é uma etapa muito importante do processo de controle, pois afeta diretamente o tempo de espera do passageiro.

O cálculo do TEA é obtido pela de acordo com a fórmula abaixo:

$$TEA = |Andar\_Cabine - Andar\ alvo\ da\ primeira\ posição\ da\ fila\ pseudo\text{-}alocada| * t1 + \left(\sum_{\substack{i = 2º\ elemento\ da\ fila\ pseudo\text{-}alocada \\ i = chamada\ pseudo\text{-}alocada}} |Andar_i - Andar_{i-1}| * t1 \right) + No\_paradas * t2$$

t1 = tempo de trânsito entre andares
t2 = tempo de parada por andar
No_paradas = Número de paradas intermediárias

Como exemplo de cálculo, considere a seguinte situação: A fila gerada pela pseudo-alocação possui os andares 3, 5 e 8 para a cabine Y. A cabine se encontra no andar 2 e a chamada pseudo-alocada foi gerada no andar 5. Se o tempo de trânsito entre andares é de 3 unidades de tempo(ut), e o tempo de parada por andar é de 2 ut, o TEA é obtido da seguinte maneira:

TEA = |2-3|*3 + (|5-3|)*3 + 1*2 = 11 ut

Ocupação(OC): Identifica quão ocupada está a cabine. Caso não se leve em conta a ocupação das cabines, corre-se o risco de alocar uma cabine lotada para atender uma requisição e acabar por não atendê-la, aumentando o tempo de espera das chamadas relacionadas com tal cabine e com aqueles que fizeram a requisição. Além do mais, uma cabine menos cheia garante maior conforto dos passageiros.

Cálculo da Ocupação: Cada cabine guarda o número de ocupantes dentro dela e fornece o valor quando solicitado. Como melhoria futura, poder-se-ia implementar um sistema que avaliasse a ocupação do elevador, levando em conta as pessoas e outros objetos que possam reduzir o espaço disponivel na cabine.

Fator de Utilização(FATU): Variável que relaciona o quanto cada cabine já foi utilizada. À primeira vista tal variável parece de pouca relevância, no entanto, um fator de utilização muito alto de uma cabine pode acarretar em uma alta demanda por manutenção, e conseqüentemente, aumento de gastos e prejuízo no serviço de atendimento das cabines restantes. Com tal variável, pretende-se aumentar a vida útil dos elevadores e diminuir o número de pedidos de manutenção para os mesmos.

Cálculo do FATU: O FATU de cada cabine é uma porcentagem do número de andares por ela percorridos em relação ao total de andares que todos as cabines do sistema percorreram.

Variável de saída DISP: Indica a disponibilidade de atendimento por uma determinada cabine. Quanto maior seu valor, mais interessante é escolher tal cabine para atender a uma nova

chamada, de acordo com a precedência estabelecida pelas regras e variáveis do processador *Fuzzy*. A cabine que apresentar o maior valor para DISP, será a escolhida.

Justificativa da escolha das variáveis: As variáveis TEA, OC e FATU foram as escolhidas por se mostrarem boas alternativas na otimização de um sistema de controle de elevadores. Elas englobam fatores importantes, já explicados, na melhoria do serviço e na melhor utilização dos recursos disponíveis.

Outras variáveis de entrada analisadas:

Tempo médio de espera pelo elevador: Corresponde à média do tempo de espera de todas as chamadas que foram alocadas para a cabine atender. É calculada somando todos os tempos de espera e dividindo pelo número de chamadas. Reflete na espera média de todos os usuários, de modo que um baixo valor desta variável indica um atendimento rápido aos usuários, onde a inserção de uma chamada a mais pesaria menos no atendimento das chamadas da cabine.

Distância da chamada: Corresponde à distância, em andares, da chamada à posição da cabine, levando-se em conta o sentido de viagem da mesma. É relacionada com o tempo de atendimento da chamada. Porém, é uma estimativa menos acurada em comparação ao TEA, visto que não leva em conta o tempo gasto com as chamadas intermediárias. Ao desconsiderar tal tempo, dá-se mais prioridade para o atendimento de cabines fisicamente mais próximas da chamada e, conseqüentemente, há um menor consumo de energia. Na Tabela 1 estão ilustradas as diferenças entre os cálculos em função da distância e em função do TEA; nota-se que, pelo cálculo da distância, a cabine 2 seria a mais apta para atender a nova chamada, enquanto que pelo cálculo do TEA, deve-se levar em conta a modelagem realizada com relação aos tempos de deslocamento entre andares (t1) e de paradas intermediárias (t2) para se saber qual seria a cabine mais apta.

Id cabine	fila de atendimento da cabine	andar da chamada pseudo-alocada	anda atual da cabine	TEA	Distância
1	3, 6, 8	6	2	|2-3|*t1 + (|6-3|)*t1 + 1*t2 = 4*t1 + t2	(6-2) = 4
2	4, 5, 6	6	3	|3-4|*t1 + (|5-4|+|6-5|)*t1 + 2*t2 = 3*t1 + 2*t2	(6-3) = 3

Tabela 1 – Comparação entre os cálculos de distância e de TEA.

Porcentagem de esperas longas: Determina a proporção de chamadas já alocadas à cabine em questão, cujo tempo de espera está demasiadamente longo (maior que um certo limiar) e pode ser afetado pela nova chamada. Para calcular esta variável é necessário manter registro de quanto cada uma das chamadas já alocadas à cabine estão esperando e avaliar se a nova chamada vai interferir no tempo das já alocadas. Esta variável indica o quão atarefada está a cabine, e é bastante interessante, visto que atribuir mais chamadas para uma cabine com muitas chamadas com esperas longas poderia tornar ainda maior o tempo de espera das chamadas já alocadas e resultar no atendimento da nova chamada apenas após um longo período de tempo. É uma variável usada para melhorar a qualidade de serviço para o passageiro, garantindo poucos atrasos fora de um certo limiar de tolerância.

4.3 Regras nebulosas para o Controle

Pelo fato do sistema proposto utilizar lógica nebulosa, a modelagem das regras nebulosas é um fator crítico para o bom funcionamento do sistema. Deve-se pensar em pesos coerentes para cada uma das variáveis, de modo a gerar saídas satisfatórias.

Como forma de simplificar a geração de regras, ou seja, ter um número não muito grande delas, adotou-se, para cada variável (TEA, OC, FATU), um conjunto com 3 termos primários:

baixo(a), médio(a) e alto(a), o que fez com que o número de regras utilizadas fosse $27(3^3)$. Apenas a variável de saída (DISP) possui 5 termos: muito baixa, baixa, média, alta e muito alta. As regras foram modeladas de modo que um maior valor de DISP indique a melhor cabine para atender uma chamada.

Deste modo, foram gerados arquivos de configuração das regras para cada caso específico de utilização, considerando-se a relevância das variáveis para cada tipo de tráfego e edifício onde seria implantado o sistema de controle. Foram utilizados como base os estudos apresentados no Anexo C para o desenvolvimento dos arquivos de configuração para os estabelecimentos hospitalares e edifícios residenciais. Tais arquivos foram modelados em condições especiais de tráfego, como, por exemplo, horário do almoço ou fim de tarde. As regras utilizadas podem ser encontradas no Anexo A, já o formato dos arquivos de configuração das mesmas podem ser observados no Anexo B.

4.4 Implementações do módulo Pré-Controle

Como visto no item 4.1, um elemento de diferenciação do comportamento do sistema de controle implementado reside na forma como eles realizam uma alocação prévia da nova chamada, ou pseudo-alocação, com o intuito de calcular a variável TEA. O elemento responsável por este processo é o Pré-Controle. Neste item, serão descritos os diferentes tipos de Pré-Controle utilizados pelo sistema de controle desenvolvido.

4.4.1 Pré-Controle Básico

O pré-controle básico foi implementado com o intuito de testar o funcionamento do simulador. Ele utiliza apenas uma cabine para atendimento e aloca todas as novas chamadas seqüencialmente, independentemente da origem ou do tipo. Portanto, é um controle bastante ineficiente.

4.4.2 Pré-Controle por Exaustão

O princípio básico utilizado por este pré-controle é a formação exaustiva de possibilidades de alocação. Cada cabine possui sua fila de atendimento de chamadas, composta por aqueles que lhe foram alocadas até então. Assim, dada uma nova chamada, o pré-controle exaustivo produz todas as possibilidades de alocação, ou pseudo-alocações, alterando a posição de inclusão da nova chamada na fila de atendimento de chamadas das cabines. A Figura 3 contém um exemplo que elucida esta questão. Considere que uma nova chamada é realizada no andar 6. O Pré-Controle recebe a fila de atendimento de cada uma das cabines. Esta fila é representada na figura como "fila de atendimento atual".

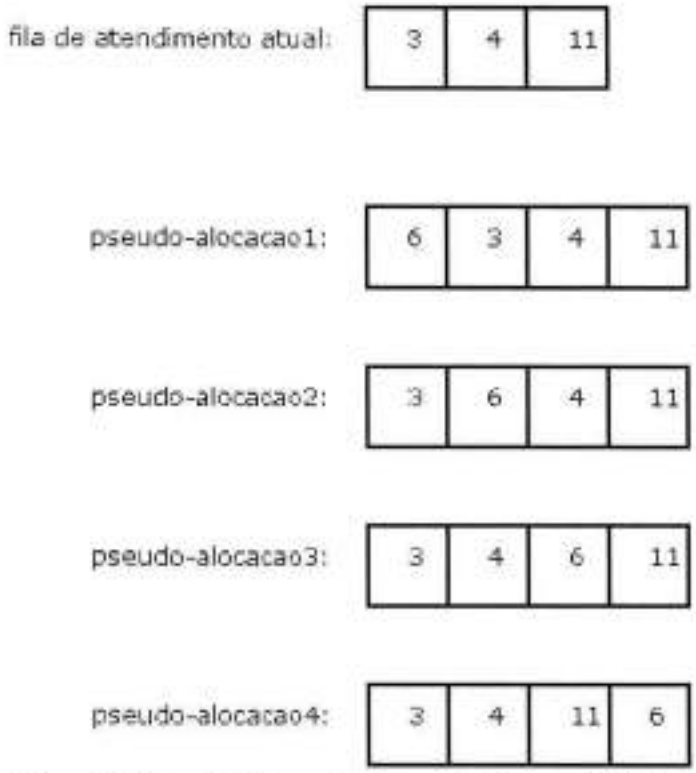

Figura 3 – Exemplo de alocação por exaustão de possibilidades.

Para cada cabine, o pré-controle efetua todas as possibilidades de pseudo-alocação, calculando seus TEA's.

Ao fim deste processo, para cada uma das cabines, é escolhida a possibilidade de pseudo-alocação com melhor TEA. O valor de cada TEA é passado, juntamente com os valores da OC e do FATU de cada cabine, para o processador *fuzzy*, que calculará, de acordo com suas regras, qual é a melhor cabine para atender esta nova chamada. No caso de uma chamada do tipo *interna*, a única diferença é que a cabine a ser escolhida após o processador *fuzzy* deve ser a mesma cabine na qual esta chamada foi gerada.

4.4.3 Pré-Controle levando em conta o Sentido da Chamada

Uma das soluções propostas para solucionar o problema do cálculo do TEA é a realização de uma pseudo-alocação através da elaboração de diversas regras que seriam responsáveis pelo melhor posicionamento da nova chamada na fila de atendimento da cabine, levando em conta, principalmente, o sentido das chamadas. Em suma, este posicionamento deve priorizar o atendimento em um único sentido, ou seja, chamadas contrárias ao sentido atual da cabine tendem a serem alocadas no fim da fila, de modo que a cabine postergue sua mudança de sentido para atender tais chamadas.

Este Pré-Controle atua de forma a respeitar a premissa de que um passageiro, ao entrar na cabine solicitada, faça uma viagem apenas no sentido escolhido no momento da realização da chamada, e desta forma traça uma fila de atendimento com a menor incidência possível de inversões de sentido do movimento do elevador, e, conseqüentemente, menor consumo de energia.

Este não é um problema trivial. Devido às muitas variáveis envolvidas, diversas situações podem ocorrer, devendo cada uma delas ser prevista. Este processo exige considerável quantidade de esforço computacional para a exaustão de todas as possibilidades.

O processo de pseudo-alocação é assim definido: para uma determinada cabine, é realizada a comparação do alvo, ou andar a ser atendido, e sentido da nova chamada com o andar no qual a cabine se encontra e com os alvos das chamadas presentes na fila de atendimento da cabine. Esta fila é percorrida enquanto não houver uma definição da posição na fila onde a chamada deve ser colocada. Para cada alvo em análise na fila, uma regra é escolhida de acordo com a comparação realizada previamente, indicando se a chamada deve ser colocada antes ou

depois do elemento em análise. Se a indicação for antes, a iteração na fila termina e a posição da chamada é definida. Através deste procedimento de pseudo-alocação, o TEA da cabine é calculado e enviado ao Controle.

As tabelas com todas as possibilidades de regras para o posicionamento da chamada na fila de uma cabine, estão disposta no Apêndice B. Na elaboração das tabelas foram considerados:

1. Se o alvo da nova chamada é maior, igual ou menor que o alvo em análise na fila.
2. Se o alvo da nova chamada é maior, igual ou menor que o alvo anterior ao alvo em análise na fila.
3. Sentidos da nova chamada e da cabine, com base na fila de atendimentos.
4. Se o alvo em análise na fila corresponde a um chamada interna.
5. O resultado da avaliação, que pode ser antes ou depois do alvo em análise na fila.

4.5 Testes do sistema de controle

Nesta seção, são apresentadas as metodologias dos testes que foram realizados para verificar o correto funcionamento do sistema de controle, antes de integrá-lo ao simulador.

Os testes aplicados para o sistema de controle não integrado ao simulador se dividem em testes do processador *fuzzy*, testes dos pré-controles, que já foram descritos no item anterior, e finalmente testes que compreendem a integração destes ao árbitro, o que resulta no sistema de

controle. Pelo fato do cálculo de TEA ter um peso grande no sistema, também foram realizados testes para validar seus cálculos.

Estes testes foram realizados com o objetivo de validar cada módulo individualmente para que houvesse um menor trabalho na integração com o simulador.

4.5.1 Testes do cálculo de TEA

Estes testes foram realizados de forma a validar os cálculos de TEA para diferentes filas de chamadas fornecidas como entrada, ou seja, cada fila apresentava uma seqüência diferente de chamadas. Foram testes simples e ocorreram conforme o esperado. De acordo com a fórmula do cálculo de TEA apresentada no item 4.2, verificou-se que a implementação estava de acordo com o que se desejava calcular. Na Tabela 2 estão alguns dos testes realizados.

Id cabine	fila de atendimento da cabine	andar da chamada pseudo-alocada	anda atual da cabine	TEA	Esperado
1	3, 6, 8	6	2	|2-3|*t1 + (|6-3|)*t1 + 1*t2 = 4*t1 + t2	4*t1 + t2
2	4, 5, 6	6	3	|3-4|*t1 + (|5-4|+|6-5|)*t1 + 2*t2 = 3*t1 + 2*t2	3*t1 + 2*t2

Tabela 2 – Testes do cálculo de TEA.

4.5.2 Testes do processador *fuzzy*.

A validação do processador *fuzzy*, nesta fase, consistiu apenas na verificação de que a implementação em Java, com o apoio da API FuzzyJ (14), correspondia ao modelado pelas regras, utilizando-se o método max-min de inferência de Mamdani (5). Desta forma, utilizou-se

do software Matlab provido do *Toolbox* de Lógica Nebulosa para confirmar o funcionamento da lógica implementada.

Foram testadas algumas combinações de entradas para que não houvesse dúvidas sobre os valores de saida obtidos pela API Java. Tais entradas foram geradas a partir de diversos valores contidos dentro do universo de discurso das variáveis lingüísticas de entrada.

A interface amigável do Matlab foi bastante útil para a comprovação dos testes, com o único inconveniente dos resultados não apresentarem a mesma precisão que a API Java; o Matlab só oferece precisão até a segunda casa decimal, enquanto a API Java oferece uma precisão com mais de dez casas decimais; o que pode ser decisivo em situações nas quais os valores obtidos na variável de saida estejam muito próximos. Comparações entre as facilidades oferecidas pelo Matlab com aquelas oferecidas pela API Java podem ser encontradas no Apêndice A.

Exemplo de teste:

No Java:

Nome Variável Entrada: FATU (fator de utilização da cabine)
Valor : 21.12676056338028 %
Nome Variável Entrada :: OC (ocupação)
Valor : 2.0 pessoas
Nome Variável Entrada :: TEA
Valor : 9.0

Nome Variável Saida :: DISP
Valor : 8.637133057151766

No Matlab:

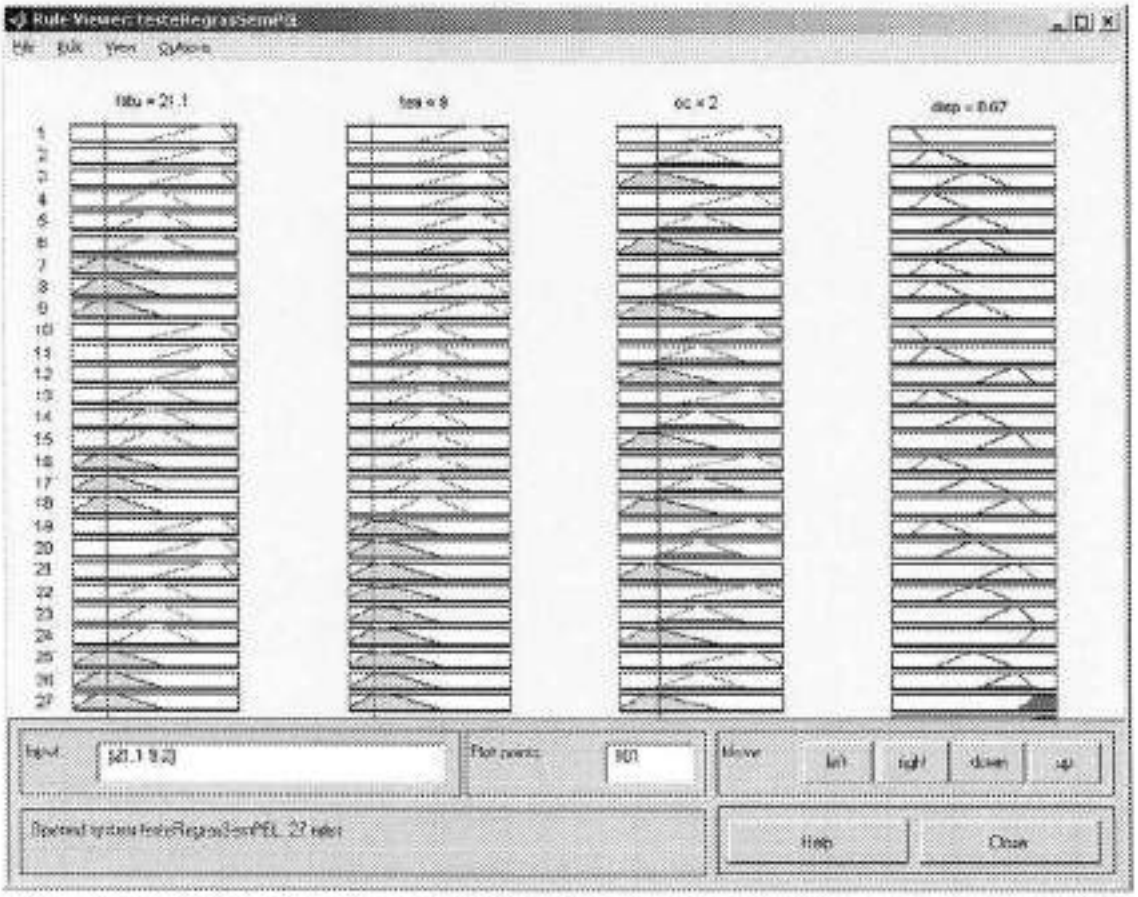

Figura 4 - Disparo de regras no Matlab.

Ao se observar a saida (variável DISP) obtida pela API Java com a obtida pelo Matlab, nota-se que a diferença de uma para a outra ocorre apenas na segunda casa decimal o que implica na aceitação dos resultados obtidos pela API Java.

Java	Matlab
8.637133	8.67

Tabela 3 - Resultado obtidos pelo FuzzyJ e pelo Matlab.

4.5.3 Testes do pré-controle

Como foi dito anteriormente, foram desenvolvidos dois pré-controles, além do pré-controle básico, para o sistema de controle. Aqui estão descritos os testes realizados para verificar o funcionamento de cada um deles.

I) Pré-controle básico

Os testes deste pré-controle foram simples, pois este pré-controle só foi necessário para a verificação do funcionamento do simulador. Como apenas uma cabine é utilizada, e a alocação de chamadas é feita seqüencialmente, foi possível observar que a cabine atendia de fato as chamadas na ordem em que eram geradas.

II) Pré-controle exaustivo

Este pré-controle aloca a nova chamada na posição da fila onde a somatória dos TEAs de todas as chamadas da fila (incluindo a nova chamada) tenha o menor valor. Portanto, tal pré-controle deve calcular a somatória para cada possível alocação da nova chamada e então escolher a de menor valor como a pseudo-alocação da cabine, para que as cabines possam, então, ser comparadas pelo processador *Fuzzy*.

Uma vez que o cálculo do TEA já havia sido testado e estava funcionando corretamente, bastava-se realizar o cálculo do TEA para todas as posições da fila e somar estes valores. Desta forma, nos testes de funcionamento, foram observados os seguintes aspectos:

- se todas as posições da fila estavam sendo levadas em conta na somatória do TEA da fila;

- se todas as possíveis alocações da nova chamada na fila são avaliadas pelo pré-controle;
- se o pré-controle estava, de fato, selecionando a alternativa de menor TEA;
- e, por fim, se a pré-alocação era realizada de acordo com a possibilidade de alocação que gerava o menor TEA;

Tais testes foram realizados diretamente na interface de desenvolvimento, uma vez que seria mais fácil observar os valores das variáveis e a iteração do processo pelas saídas impressas na tela e pelo *debugger* da IDE.

Foi observada a alocação correta na fila e a decisão acertada por aquela que apresentava a nova chamada em uma posição que gerava o menor TEA, e uma vez que o cálculo do TEA já havia sido testado, foi possivel garantir que o pré-controle exaustivo estava funcionando de acordo com o especificado.

III) Pré-controle com sentido

Como explicado anteriormente, este método é baseado em uma tabela de decisão, sendo bastante complexo devido às diversas situações possíveis de alocação. Deste modo, diversas configurações, que abrangessem todos os casos possíveis, foram testadas para comprovar a validade de todas as regras e corrigir eventuais inconsistências. Por se tratarem de testes um pouco mais complexos sua apresentação é descrita no Apêndice B.

4.5.4 Testes de integração do processador *fuzzy* e do pré-controle ao árbitro

Após o processador *fuzzy* e o pré-controle terem sido devidamente testados, eles foram integrados ao árbitro para o teste final antes da integração com o simulador. Foi gerada uma quantidade de entradas correspondente ao número de elevadores sendo considerados nos testes. Cada entrada era suficientemente diferenciada de modo a não poder ocorrer igualdades nas saídas do processador *fuzzy*, desse modo o árbitro deveria indicar corretamente qual elevador era o mais apto a atender uma nova chamada.

Os testes ocorreram normalmente, o árbitro apresentou o comportamento esperado ao indicar, para o atendimento de uma nova chamada, a cabine com o maior valor de DISP. Na Tabela 4 está um modelo de teste realizado.

Cabine	TEA	FATU	OC	DISP
1	10	25	6	5,2466
2	30	70	2	6,3854
3	20	10	0	7,6373
4	15	30	4	6,7401

Escolha do árbitro	Esperado
Cabine 3	Cabine 3

Tabela 4 – Teste de integração do processador *fuzzy* e do pré-controle ao árbitro.

5 Simulador

Devido à necessidade de uma avaliação menos sujeita a fatores estatísticos e à dificuldade de realizarem-se testes em um sistema de elevadores real, ou mesmo em protótipo de menor escala, optou-se por utilizar um simulador que pudesse gerar as entradas para o sistema de controle (sensores) e responder às suas saídas (atuadores) de forma a caracterizar o comportamento de um sistema de elevadores em um edifício de forma verossímil. A forma como o simulador se comunica com o sistema de controle pode ser vista na Figura 5.

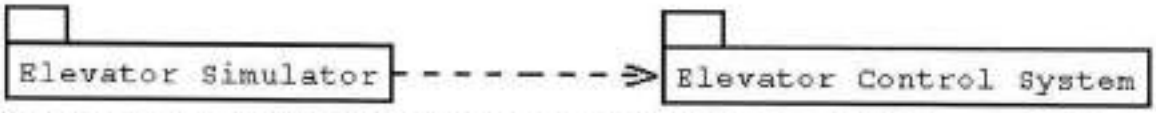

Figura 5 – Estrutura de pacotes em notação UML (*Unified Modeling Language*), denota como o simulador se comunica com o sistema de controle.

Inicialmente, foi feito um estudo da disponibilidade de simuladores de código-fonte aberto. Foram encontrados alguns simuladores bastante sofisticados com diversas simulações de tráfego de pessoas, porém, tanto pela existência de panes (*bugs*) desconhecidas, como pela falta de documentação de desenvolvimento, ou ainda pela utilização de parâmetros pouco relevantes para este estudo, optou-se pela elaboração de um simulador totalmente novo que se adaptasse às necessidades particulares deste projeto. A descrição dos pacotes utilizados, descritos pelo diagrama de classes do simulador desenvolvido, está detalhada no Anexo E. A descrição dos Casos de Uso do Simulador pode ser encontrada no Anexo F.

Para uma melhor visualização de como ocorre esta simulação, a Figura 6 mostra um protótipo de interface construída para o programa.

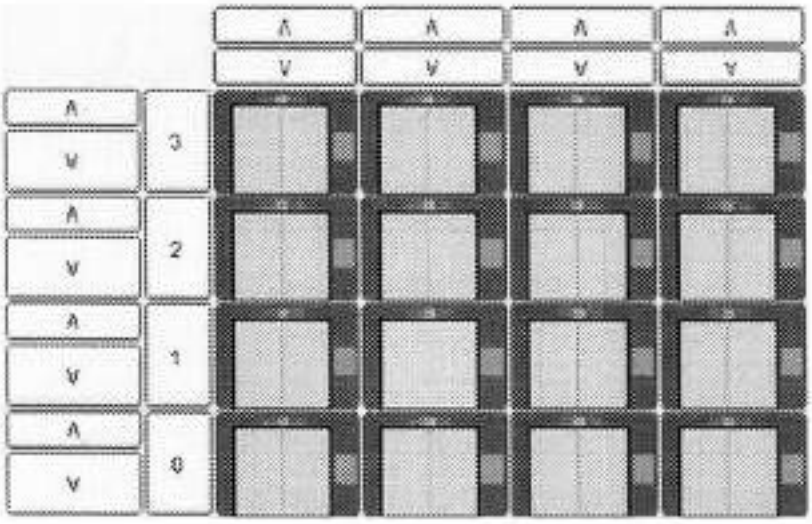

Figura 6 – Protótipo da Interface do Simulador.

As flechas à esquerda são botões usados pelo usuário para indicar o sentido que uma Pessoa vai querer deslocar-se dentro do Prédio. Ao selecionar um sentido, o usuário deve informar qual é o Andar destino, clicando no respectivo botão do painel interno, visualizado na Figura 7.

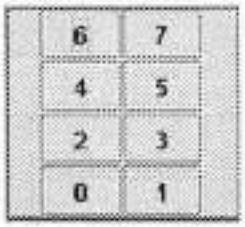

Figura 7 – Painel Interno, onde é escolhido o andar destino.

Desta forma, o Simulador criará uma nova Pessoa, cujo andar origem será o andar no qual foi selecionado o sentido; e o andar destino será o andar selecionado no painel interno. As flechas superiores indicam o sentido de deslocamento da Cabine. Se ambas estiverem apagadas, a Cabine está parada, caso contrário, a flecha acesa indica o sentido.

Quando a porta de uma cabine é aberta (para carga ou descarga de alguma pessoa), o simulador destaca a cabine, conforme ilustra a Figura 8.

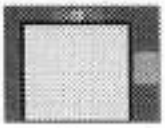

Figura 8 – Destaque de porta aberta da cabine.

Enquanto a cabine se desloca, o simulador enfatiza seu movimento, como pode ser notado na Figura 9.

Figura 9 – Ilustração de cabine em passagem por um determinado andar.

Uma descrição mais detalhada da interface com o usuário é fornecida no Anexo G.

5.1 Simulação

O simulador desenvolvido é um simulador de eventos discretos, do tipo orientado a eventos e com incrementos fixos de tempo. A vantagem de se utilizar um método como este é que ele permite a ocorrência de procedimentos paralelos controlados através de seus estados individuais (15).

A lógica desenvolvida para a simulação foi modelada em Redes de Petri (16) de modo a identificar as transições e processos envolvidos, como pode ser visto na Figura 10.

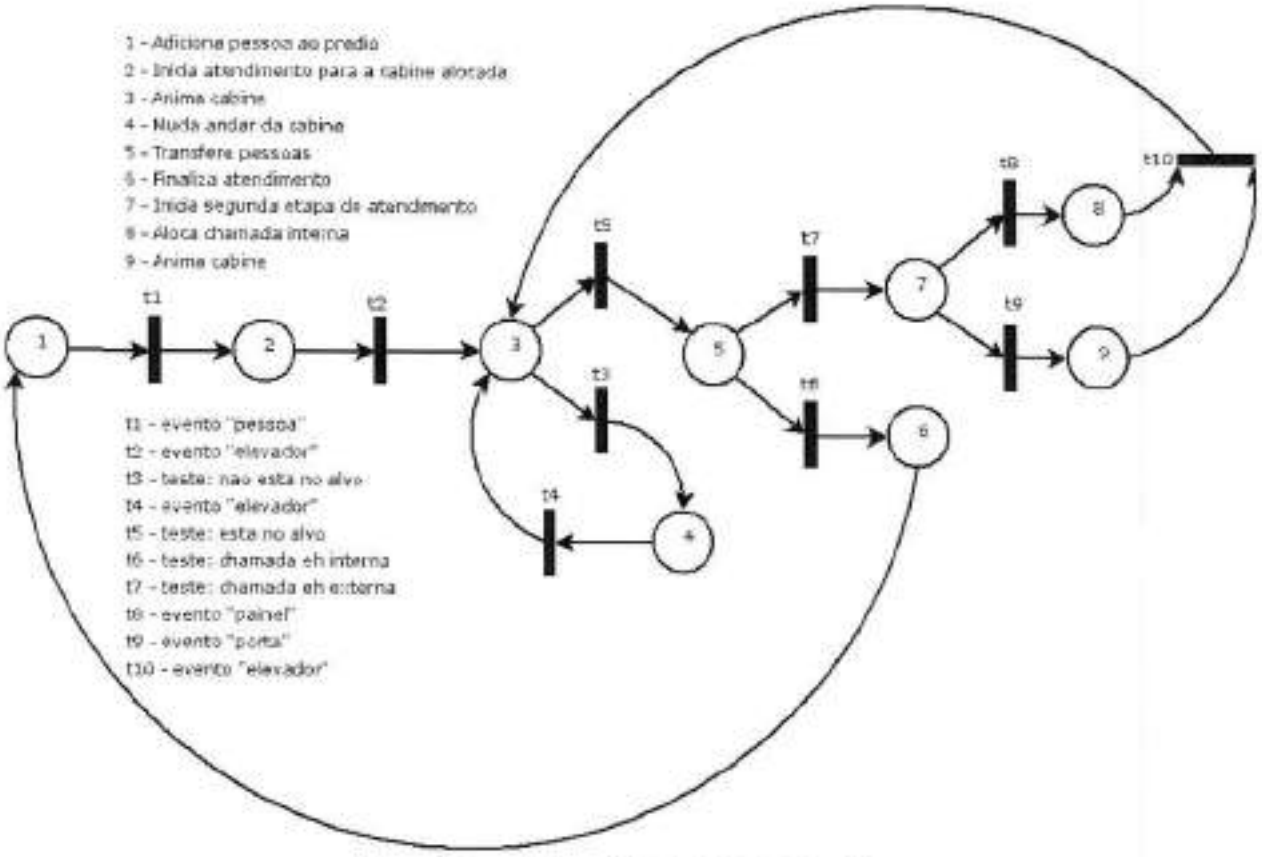

Figura 10 – modelo em Rede de Petri do Simulador.

O estado 1 indica que uma pessoa foi adicionada ao prédio, ou seja, há uma chamada externa a ser atendida no andar e no sentido solicitado por esta pessoa. A transição t1 representa a chegada do Evento "Pessoa". Quando este evento é disparado, a chamada criada pela pessoa é alocada para uma das cabines. No estado 2, o simulador inicia o atendimento a esta chamada. Um Evento "Elevador" é programado. A transição t2 equivale ao disparo deste evento. O estado 3 faz a animação da cabine e verifica se a cabine já chegou ao seu alvo. A transição t3 indica um Evento "Elevador" intermediário, ou seja, que não alcançou o alvo, enquanto a transição t5 indica um Evento "Elevador" final, isto é, que alcançou seu alvo. O estado 4 muda o andar da cabine e aguarda o disparo da transição t4, um evento "elevador".

O estado 5 pode ser resumido através da transferência das pessoas entre a cabine e o andar alvo. A transição t7 representa o resultado do teste de verificação de uma chamada do tipo externa. Isto significa que existe ainda uma segunda etapa de atendimento a ser realizada, que é o atendimento da chamada interna desta pessoa. A transição t6 indica que a chamada é interna e, portanto, o atendimento deve ser finalizado no estado 6.

No estado 7, a pessoa acabou de ter sua chamada externa atendida, entra na cabine e deve iniciar a segunda etapa do seu atendimento. Desta forma, um evento "painel", t8, transiciona a rede para o estado 8, onde sua chamada interna é alocada. A transição t9 é um evento "porta", que leva ao estado 9, onde é realizada a animação do fechamento da porta da cabine. A transição t10 nada mais é do que um Evento "Elevador", assim como a transição t2. No Anexo H, encontra-se a descrição de cada evento na forma de pseudocódigo. A forma de operar o Simulador é descrita no Anexo I.

5.2 Testes representativos do simulador

A comunicação entre simulador e o sistema de controle dá-se nos momentos em que as pessoas requisitam chamadas externas ou internas. Estas chamadas são transferidas do simulador para o sistema de controle que define qual cabine se encarregará de atender tal chamada e então aloca a chamada na fila da cabine escolhida. Uma vez que isto seja feito, o simulador volta ao seu funcionamento independente, analisando as filas de cada cabine e estabelecendo suas rotas, posições, entrada e saída de pessoas, registro de dados de utilização do sistema e atualizações gráficas na interface.

Desta forma, a fim de testar o funcionamento do simulador, foi necessário criar um pequeno módulo que se encarregasse de receber as chamadas realizadas pelas pessoas e as alocasse nas filas de atendimento dos elevadores. Tal controle encarregou-se de alocá-las seqüencialmente para o primeiro elevador do grupo, de modo que para a realização dos testes de interesse, bastava gerar lotes (*batches*) de entrada de pessoas com as características desejadas. Para alguns casos de teste, foi inserido um pequeno atuador neste controle básico de forma a colocar uma nova chamada no topo da fila, simulando o caso em que o elevador atende uma chamada requisitada enquanto ele estava em movimento e atendida antes de ele alcançar o seu destino inicial.

As características de chamadas testadas e seus respectivos resultados são analisados na Tabela 5.

Além disto, foi testada a entrada manual de chamadas e um grande arquivo de entradas de chamadas geradas arbitrariamente. Por fim, para conferir que todas as cabines estavam funcionando, colocou-se um mecanismo que selecionava aleatoriamente a cabine que atenderia uma chamada.

O Simulador apenas foi considerado validado quando todos estes testes apresentaram o comportamento esperado. E uma vez que os sistemas de controle foram finalizados, os mesmos testes foram executados a fim de confirmar a validação do simulador implementado.

Chamada	Comportamento Esperado	Funcionamento
Chamada realizada enquanto o elevador está em movimento.	Chamada alocada normalmente (de acordo com a política do módulo de controle básico implementado) para o elevador.	OK
Chamada realizada em andar Y enquanto o elevador está parado (e com a fila vazia) no andar X. (com X diferente de Y)	A chamada é alocada para o elevador que se movimenta até o andar que a realizou, recolhe a pessoa e atende à chamada interna.	OK
O mesmo exemplo anterior com X = Y	A chamada é alocada para o elevador que simplesmente abre as portas recolhe as pessoas e atende à chamada interna.	OK
Chamada do andar X solicitando viagem até o mesmo andar da chamada (X). Com elevador parado no andar X.	Chamada é alocada para o elevador que abre as portas, recebe a pessoa que realiza a chamada interna para o andar corrente. O elevador então atende a chamada (voltando a abrir suas portas e liberar a pessoa que acabou de entrar).	OK
Chamada do andar X solicitando viagem até o mesmo andar da chamada (X). Com elevador em movimento.	Chamada é alocada para o elevador que, após atender as primeiras chamadas da fila, atende a chamada a ser testada e desempenha o mesmo comportamento do item anterior.	OK
Chamada inserida na primeira posição da fila do elevador em uma posição dentro do caminho do elevador (enquanto este vai atender outra chamada)	A cabine atende a chamada que está na primeira posição da fila, gera sua chamada interna (que é alocada pelo controle), recebe a pessoa e atende a próxima chamada da fila.	OK
A mesma situação anterior, porém uma fração de tempo antes do elevador chegar no andar inserido na primeira posição da fila.	Mesmo comportamento do item anterior.	OK
A mesma situação anterior, porém realizada no momento em que o elevador passa pelo andar que realizou a chamada.	Elevador sobe ou desce para o próximo andar (de acordo com o que estava no topo da fila antes) e então volta para atender a nova chamada.	OK

Tabela 5 – Chamada – comportamento.

5.2.1 Teste do Sistema de Controle integrado ao Simulador

Após a consolidação dos testes anteriores, foi feita a integração do sistema de controle com o simulador. Devido aos testes terem sido bem conduzidos, e todos os módulos estarem bem encapsulados, a integração foi completada em pouco tempo. A interface do sistema de controle com o simulador é feita por um componente que se comunica com o árbitro, este indica o elevador escolhido e o simulador é responsável por mostrar na interface tal escolha, ao movimentar tal elevador em direção das chamadas alocadas a ele.

Com as mesmas entradas da Tabela 4 do item 4.5.4, acrescidas da indicação da chamada a ser atendida, observou-se o comportamento exato do simulador, ao ser realizada a animação da cabine escolhida pelo árbitro para o atendimento da chamada, no caso, a cabine 3.

6 Metodologia de projeto/ Implementação

Neste capítulo, será abordado como o projeto foi dividido para que pudesse ser realizado e como foram definidos as ferramentas e recursos necessários ao longo do tempo.

6.1 Descrição da metodologia do projeto

A partir da definição geral do projeto, em seu inicio, foram definidas responsabilidades para cada membro da equipe, de modo que a carga de trabalho ficasse bem distribuida entre eles e, assim, pudessem ser mais bem gerenciados o controle de prazos e os esforços por tarefa.

Cada componente da equipe se dedicou a estudos especificos relacionados à área de atuação do projeto. Abaixo podem ser observados a divisão de tarefas e o cronograma de todas as atividades envolvidas. Nota-se que um periodo (julho) foi menos produtivo e não gerou resultados significativos.

	Abril		Maio					Junho				Julho			
	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25
Estudo fuzzy															
Estudo controle															
Estudo simulador															
Definição de variáveis															
Estudo Matlab															
Estudo da API FuzzyJ															
Implementação controle															
Interface simulador															
Implementação simulador															
Relatório de especificação															
Relatório de junho															
Relatório de outubro															
Estudos de Caso															
Seleção de variáveis															
Integração Simulador/Controle															
Testes e correção de erros															
Documentação final															
Apresentação Orientadora															
Apresentação Final															

LEGENDA — Richard; Mario; Thomas; Todos; Richard e Thomas

Tabela 6 – Cronograma das Atividades abr/jul.

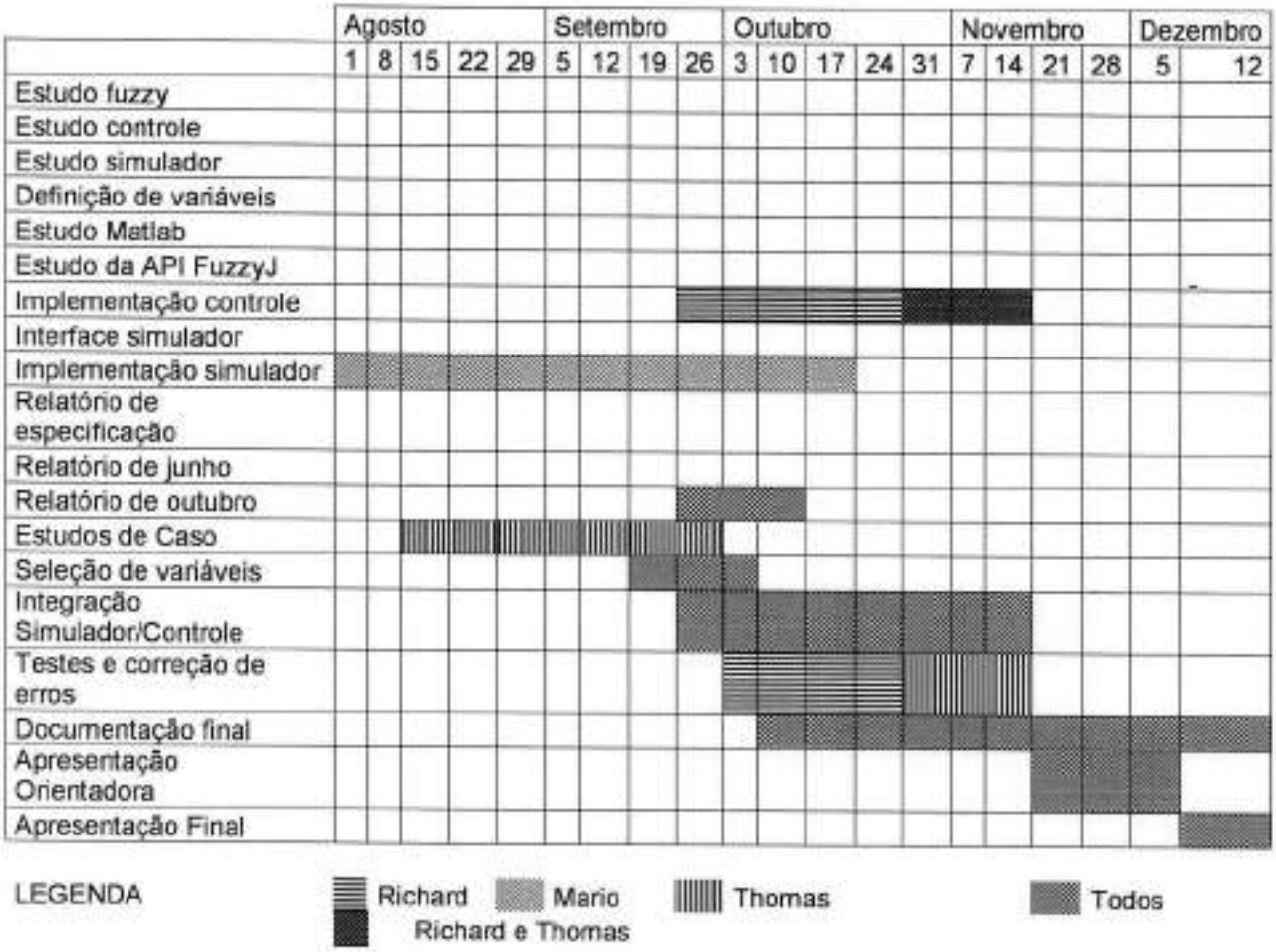

Tabela 7 – Cronograma das Atividades jul/dez.

Estudo *fuzzy*: estudo das técnicas e algoritmos baseados em lógica nebulosa.

Estudo controle: estudo acerca dos métodos existentes para controle de grupos de elevadores, e das técnicas aos quais estes se baseiam.

Estudo simulador: estudo do funcionamento de um simulador de grupo de elevadores.

Definição de variáveis: definição das variáveis a serem otimizadas e consideradas no projeto do controle dos elevadores

Estudo Matlab: estudo da ambiente de lógica nebulosa do Matlab, de forma a validar a implementação em Java.

Estudo API FuzzyJ: estudo da FuzzyJ como ferramenta de implementação de lógica nebulosa.

Implementação controle: implementação da lógica de controle do grupo de elevadores (incluindo as funções nebulosas).

Interface Simulador: desenvolvimento da interface do simulador de elevadores a ser utilizado.

Relatório de especificação: elaboração do relatório correspondente às especificações do projeto.

Relatório Doc. Detalhe (junho): relatório de documentação e detalhamento do projeto.

Estudos de Caso: estudos de sistemas reais de elevadores, fluxos de pessoas em edifícios onde o sistema possa ser implementado e normas relacionadas aos sistemas de elevadores.

Seleção de Variáveis: seleção das variáveis e diferenciação de seus pesos em cada tipo de utilização do sistema.

Integração dos módulos: integração do controle de elevadores e simulador.

Testes e correção de erros: testes e resolução de erros dos programas e algoritmos utilizados.

Documentação final: elaboração da documentação final do projeto.

Apresentação Orientadora: desenvolvimento do conteúdo a ser apresentado para a banca de avaliação do projeto, com assistência da orientadora.

Apresentação Final: apresentação do projeto perante a banca avaliadora.

Com relação à gerência do projeto, foram utilizadas, principalmente, as conversas do dia-a-dia e trocas de e-mail entre os integrantes, informando o andamento das atividades. Periodicamente foram realizadas reuniões presenciais entre os componentes do grupo para resolução de dúvidas em determinadas atividades, além de reuniões com a orientadora para discutir o andamento do projeto e assuntos técnicos relacionados.

6.2 Descrição da escolha das tecnologias

A equipe decidiu pela utilização de ferramentas Java para o desenvolvimento do projeto devido à grande familiaridade com a linguagem por parte dos integrantes, pela disponibilidade de APIs (*Application Program Interface*) que permitissem a implementação da arquitetura do projeto, de forma a restringir incompatibilidades durante a integração entre os módulos, e pela numerosa quantidade de ferramentas gratuitas disponíveis para utilização.

Para o desenvolvimento na linguagem, foi escolhida a IDE (*Integrated Development Environment*) NetBeans que oferece um ambiente eficiente para a criação de interfaces, o que foi de grande relevância para o desenvolvimento do simulador utilizado no projeto.

Os arquivos de configuração utilizados para a definição do sistema de controle, foram criados na linguagem XML (*Extensible Markup Language*), o que permitiu um melhor gerenciamento de todas as variáveis e regras envolvidas, além de facilitar o processo de *parsing* (análise sintática).

6.3 Descrição dos recursos e infra-estruturas necessários

O projeto foi baseado em um ambiente de simulação, não necessitando de recursos especiais para seu desenvolvimento. Um computador de 1.6Ghz e 256MB de memória RAM são suficientes para realizar os testes com o sistema.

Para a realização do projeto, foram necessários apenas uma IDE, já explicado anteriormente, a máquina virtual Java e algumas API's de apoio, para *parsing* de XML e para a implementação do sistema de controle (FuzzyJ).

As ferramentas utilizadas para gerar a documentação do projeto e os arquivos de testes foram um editor de texto e um editor de planilhas.

7 Experimentos e resultados

Para que fosse possível ser realizada uma avaliação confiável dos sistemas propostos neste trabalho, foi necessária a criação de entradas compatíveis, através de modelos de tráfegos residenciais e hospitalares, além do estabelecimento de referências de comparação. As referências usadas para tanto foram: variâncias das regras de lógica nebulosa implementadas, um sistema de lógica nebuloso estudado inicialmente (17) e (18) e distintas implementações do sistema de Pré-Controle, discutido no item 4.4.

7.1 Modelagem dos Tráfegos

As modelagens dos tráfegos dos elevadores foram baseadas em um estudo realizado acerca das normas regulamentares dos serviços de elevadores e de casos reais correspondentes aos edifícios cujos tráfegos devem ser atendidos pelo sistema a ser implementado, tal estudo encontra-se no Anexo C deste documento.

Com base neste estudo e nas normas ABNT (19), foram desenvolvidos modelos de tráfego condizentes com os padrões de tráfego real de cada um dos edifícios simulados.

Os modelos de tráfego são representados por arquivos de extensão XML escolhidos no início da simulação no modo automático. Tais arquivos delimitam o instante de chegada das pessoas, bem como seus andares de origem e destino. Segue abaixo um trecho do arquivo ilustrando a modelagem do mesmo:

```
<EVENTO>
    <PESSOA chegada="5" origem="7" destino="0"/>
    <PESSOA chegada="10" origem="7" destino="0"/>
    <PESSOA chegada="10" origem="6" destino="1"/>
    <PESSOA chegada="13" origem="3" destino="1"/>
    ...
</EVENTO>
```

7.1.1 Geração do tráfego residencial

A população dos prédios residenciais depende bastante do bairro e cidade onde o mesmo está localizado, do preço do seu condomínio e da área total do apartamento, de modo que pode ser que haja uma maior quantidade de moradores com um determinado perfil.

A fim de se criar o modelo de tráfego a ser usado como teste do sistema de elevadores, foram determinados alguns perfis de comportamento, e logo em seguida alguns perfis de ocupação.

Perfis de comportamento (em dias úteis):

- Aposentado: sai uma vez no período da manhã (fora do horário de pico, das 7:00 às 9:00) e volta depois de 30 a 60 minutos, e o mesmo ocorrendo durante a tarde, além de receber uma visita ao final da tarde, que deixa o prédio de noite.
- Adultos: de modo geral saem entre 7 e 8 horas e retornam entre 19 e 20 horas
- Adultos com filhos: de modo geral os pais levam as crianças para a escola ou creche quando saem para o trabalho, podendo ou não buscá-las e retornar para almoçarem juntos. Existe ainda a possibilidade de se contar com uma babá ou empregada doméstica que sai com as crianças durante à tarde ou manhã.

- Filhos jovens: saem de casa para a escola (de manhã ou de tarde) voltando no horário de almoço ou jantar, e saem para alguma atividade extra-classe durante o período sem aulas, retornando cerca de 3 ou 4 horas após terem saído.
- Nesses perfis básicos, foram inseridas algumas variantes, visto que alguns adultos saem mais cedo ou mais tarde em virtude de um rodízio de circulação de automóveis ou jornada de trabalho flexível, nem todos saem de carro, alguns saem de noite e etc.

Os perfis de habitação considerados foram de casais adultos com 0 a 3 filhos, casais de aposentados, além de pais solteiros. Sendo que cerca de 1,4 pessoas por apartamento se dirige à garagem em vez do térreo, para deixar o prédio.

Os tráfegos mais intensos em prédios residenciais correspondem ao período matinal que vai das 7:00 às 8:00, ao período das 12:00 às 14:00 e, por fim, ao período das 19:00 às 20:00. Logo, foram testados apenas estes períodos, até porque uma vez que o serviço atinja a qualidade esperada em tais horários, os outros períodos também estarão servidos com qualidade igual ou superior.

7.1.2 Geração do tráfego de prédios hospitalares

O tráfego de prédios hospitalares é bastante variável no decorrer do dia, havendo tanto o trânsito de funcionários (que estão em maior número durante os períodos matutinos e vespertinos do que nos plantões) quanto de pacientes e visitantes.

A maior demanda de serviço por parte do sistema de elevadores se dá nos horários de visitas e, portanto, este foi o primeiro caso a ser modelado. De acordo com estimativas conseguidas de funcionários de instituições hospitalares, tal tráfego se divide em

aproximadamente 40% de chegada de visitantes no primeiro quartil do período de visitas, 50% no segundo quartil e 10% no terceiro quartil, enquanto que a saída dos visitantes se dá como 30% no terceiro quarto do período de visitas e 70% no último quartil.

A outra demanda de maior impacto para o sistema consiste no horário de almoço e troca de turnos, onde há a transição entre *down-peak* e *up-peak* no caso do almoço (e o contrário na troca de turnos) durante o período.

A média de visitantes é de um visitante por leito, e há uma média de 25 funcionários por andar durante o dia e 8 funcionários nos plantões noturnos.

Assim sendo, foi gerada uma entrada aleatória seguindo tais premissas a fim de servir como entrada para o sistema.

7.2 Implementação de um sistema de controle para estudo comparativo

Como visto no item 4, todo Sistema de Controle dentro da arquitetura proposta necessita de um Controle e um Pré-Controle. No caso desta implementação, o Pré-Controle utilizado é o Exaustivo, visto no item 4.4.2, já o Controle foi adaptado a partir do Controle que já havia sido desenvolvido para os outros Pré-Controles.

O Controle aqui utilizado, baseado em (17) e (18), difere, em relação àquele proposto neste trabalho, quanto à função de avaliação e às regras utilizadas na escolha da melhor cabine para atender uma nova chamada. Este estudo comparativo se fez necessário para defrontar os resultados aqui obtidos com os de outros trabalhos já publicados e aceitos internacionalmente.

O novo sistema foi inserido como um módulo à parte de modo a poder ser selecionado antes de se iniciar a simulação. Ele foi testado nas mesmas condições que os sistemas anteriormente implementados, de modo que a comparação entre eles fosse válida.

O sistema utilizado para comparação tinha, como foco principal, a diminuição do tempo médio de espera para o atendimento de uma chamada. Para tanto, foi utilizada a seguinte função de avaliação:

$$\Phi(k) = T_{MED}(k) - \alpha * T\alpha(k)$$

onde,

$$T_{MED}(k) = \Sigma\, T_{parado}(k) + \Sigma\, T_{andando}(k)$$

Quando uma chamada ocorre, essa função é calculada para cada cabine (k = 0, 1, ..., n). Quanto menor o resultado da função, maior a probabilidade de alocar a cabine k para atender esta nova chamada.

$T_{MED}(k)$ é o tempo estimado para que a cabine chegue ao andar da nova chamada. É composto pelas componentes $\Sigma\, T_{parado}(k)$, que considera as chamadas (externas e internas) que serão atendidas pela cabine e o tempo correspondente em que ficará parado; e $\Sigma\, T_{andando}(k)$, que considera o tempo em que a cabine estará se movimentando até este andar.

A função $T\alpha(k)$ indica para uma cabine k o quão facilmente ele poderá atender a uma nova chamada considerando que o próximo andar a ser atendido seja um andar n. A partir deste andar n, considera-se uma mesma quantidade de andares para cima e para baixo. Se a nova chamada ocorre no próprio andar n, a função $T\alpha(k)$ terá valor 1, caso ocorra em andares dentro da faixa definida anteriormente (n[-,+]1, n[-,+]2, ..., n[-,+]f), terá seu valor ajustado para baixo, de acordo com a implementação da função.

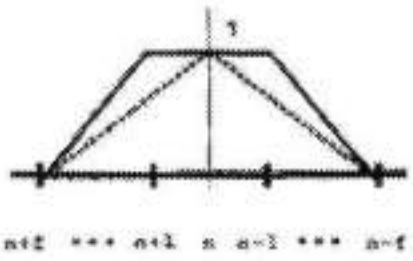

Figura 11 - Função Tα(k)

Na Figura 11, é possível perceber como esta função poderia ser implementada e, que conforme a chamada esteja mais distante, menor será a chance de escolha de tal cabine pelo sistema de controle.

De acordo com as referências, o fator α é introduzido na fórmula devido à complexidade inerente ao sistema de controle de elevadores. Ele é utilizado de forma a considerar as outras variáveis do sistema, como o *up-peak* (*UP*), *down-peak* (*DN*), tempo médio de espera *(AWT - average waiting time)* e consumo de energia (*PC – Power Consumption*). Pode-se observar que, se α for grande para uma determinada cabine, a probabilidade dele atender a chamada aumenta; o contrário também é verdadeiro.

Para o cálculo de α, foram considerados dois grupos de variáveis. O primeiro recebe como entrada UP e DN e gera como saída α', o segundo recebe AWT e PC e gera k. Então, α pode ser obtido da seguinte maneira:

$$\alpha = \alpha' + k$$

As regras que foram utilizadas se encontram nas tabelas a seguir:

UP \ DN	VS	S	M	L	VL
VS	VS	S	S	VS	VS
S	S	S	S	M	S
M	S	S	M	L	L
L	VS	M	L	L	L
VL	VS	S	L	L	VL

Tabela 8 - (UP, DN, α').

PC \ AWT	VS	S	M	L	VL
VS	ZE	ZE	PS	PM	PL
S	ZE	ZE	ZE	PS	PM
M	ZE	ZE	ZE	ZE	PS
L	NL	NL	NM	NS	NS
VL	NL	NL	NM	NS	NS

Tabela 9 - (AWT, PC, k).

Seguindo a lógica da tabela:

- *Se* UP é M e DN é L, *então* α' é L.
- *Se* UP é VL e DN é S, *então* α' é S.
- *Se* UP é S e DN é L, *então* α' é M.
- *Se* AWT é M e PC é L, *então* k é ZE.
- *Se* AWT é VL e PC é S, *então* k é NL.
- *Se* AWT é S e PC é L, *então* k é PS.

Como observação, deve-se ressaltar que UP e DN são variáveis que indicam a quantidade de pessoas atuando no padrão *up-peak* e *down-peak*, respectivamente. Os níveis nebulosos que as variáveis (UP, DN, AWT, α e PC) podem obter são VS, S, M, L e VL, respectivamente muito baixo, baixo, médio, alto e muito alto. Enquanto os níveis da variável k (NL, NM, ZE, PM e PL) são definidos como alto negativo, médio negativo, zero, médio positivo e alto positivo.

7.3 Comparações entre modelos de controle

Os desempenhos dos métodos de controle foram comparados, para prédios residenciais, utilizando-se três tipos de tráfegos: tráfego matinal (correspondente ao *down-peak*), tráfego do horário do almoço (correspondente a *up-peak* convergindo para *down-peak*) e tráfego do final da tarde (correspondente ao *up-peak*). Para hospitais, a comparação foi feita com base no tráfego do horário de visitas (correspondente a *up-peak* convergindo para *down-peak*).

A modelagem de todos os tráfegos foi realizada conforme o especificado no item 7.1, com a diferença de que, para o tráfego nos prédios residenciais, foi utilizado um prédio de seis andares (além do térreo e garagem) dotado de duas cabines de elevadores com capacidade de seis passageiros cada uma. Enquanto que, para o hospital, foi simulada a existência de 10 andares (além do térreo) e quatro cabines com a capacidade de oito pessoas cada uma. Desta forma, as particularidades de cada tipo de edifício foi obedecida.

Para os prédios residenciais foram observados os tempos de trânsito médio (quanto tempo em média o passageiro viaja no elevador), consumo de energia, tempo de espera médio global (quanto tempo uma pessoa espera pela chegada do elevador) e o fator de utilização das cabines.

O fator de utilização foi bem equilibrado entre todos os métodos de controle, onde todos permaneceram entre 47% a 53%, uma faixa de valores bastante satisfatória. Quanto ao consumo de energia, o método exaustivo e o implementado pelo artigo usado como base tiveram desempenho bastante similar, enquanto que a implementação com pré-controle, que leva em conta o sentido, obteve um consumo de energia em média 89% menor para os três tráfegos simulados (horário de almoço, início da manhã e fim da tarde).

O tempo de trânsito médio global permaneceu bastante similar entre o controle de referência e o exaustivo no período da manhã, mas nos outros períodos, o tempo de trânsito para o controle de referência foi cerca de 4% maior. Notou-se também que o controle com sentido teve um tempo 6% menor que o controle exaustivo (exceto no período da manhã, onde seu tempo também foi praticamente igual ao exaustivo).

Quanto ao tempo de espera de atendimento médio global, durante o horário do almoço o controle do artigo se sobressaiu, sendo praticamente 15% menor que os outros controles (cujos tempos foram bem similares). Nos tráfegos da manhã e da tarde, o controle exaustivo apresentou melhores resultados (22% menor que os outros controles durante a manhã e, no período do fim da tarde, 12% menor que o com sentido que foi 12% menor que o tempo do controle do artigo). Estes resultados são apresentados na Tabela 10.

		Parâmetros de Análise			
Situação	*Controlador*	*TIMG*	*TEMG*	*CE*	*FATU*
Manhã	*Referência*	10.90361446	14.68674699	193	4.6%
	Exaustão	10.95783133	11.34337349	181	2.7%
	Com Sentido	10.80722892	14.6626506	169	0.6%
Almoço	*Referência*	11.32432432	6.77027027	186.5	6.1%
	Exaustão	11.44594595	7.898648649	186.5	0.2%
	Com Sentido	10.57432432	7.959459459	168	3.6%
Fim de Tarde	*Referência*	11.73275862	10.9137931	161.5	2.1%
	Exaustão	11.27586207	8.413793103	166	4.2%
	Com Sentido	10.6637931	9.629310345	143	4.2%

Tabela 10 - Resultados Comparativos para o tráfego residencial.

A seguir são descritas as variáveis de controle do tráfego residencial:

TTMG (Tempo de Trânsito Médio Global): é a média entre os tempos médios que cada cabine leva para atender uma chamada interna, ou seja, o tempo que se passa desde que um passageiro entra na cabine até o mesmo sair dela.

TEMG (Tempo de Espera Médio Global): média do tempo médio de espera de cada elevador, onde o tempo de espera corresponde ao tempo que uma pessoa espera desde o momento em que faz uma chamada externa a um dos elevadores, até que a cabine chegue no andar em que a chamada externa foi realizada.

CE (Consumo de Energia): consumos de energia de todas as cabines somados. Este parâmetro é adquirido a partir do número de andares que cada cabine percorreu.

FATU (Fator de Utilização): corresponde ao quanto uma cabine andou a mais que o fator de utilização ótimo (igual utilização de todas as cabines).

Em relação ao tráfego hospitalar, observou-se um tempo de espera médio global muito alto do controle com sentido (quase 40% maior que os tempos dos outros controles) e um consumo de energia um pouco melhor que os outros controles (cerca de 5% menor). O tempo de trânsito do controle com sentido foi um pouco menor que os demais (cerca de 3% menor), mas a sua porcentagem de esperas longas foi 50% maior que a média dos outros controles. O controle exaustivo apresentou um número menor de esperas longas que o de referência, mas um tempo médio de espera global um pouco maior.

A Tabela 11 apresenta os resultados alcançados nos testes comparativos entre os diferentes controladores, para o caso de tráfego hospitalar.

	Parâmetros de Análise			
Controlador	*TTMG*	*TEMG*	*CE*	*PEL*
Referência	14.11	10.31071429	3780	3.917306053
Exaustão	14.08214286	10.96714286	3817	3.192439062
Com Sentido	13.88214286	14.86142857	3641	5.317077513

Tabela 11 - Resultados Comparativos para o tráfego hospitalar.

A variável de controle PEL é descrita a seguir:

PEL (Porcentagem de Esperas Longas): número de chamadas cujo tempo de espera superou um certo limiar (neste estudo, 40 segundos) em relação ao total de chamadas atendidas.

8 Conclusão

Com o estudo dos sistemas de controle e suas diversas implementações, fica claro que a complexidade de se projetar um sistema que seja satisfatório em todos os aspectos é muito grande, as dificuldades inerentes a este tipo de sistema obrigam o projetista a definir os parâmetros mais importantes que devem ser observados e otimizados, de modo que a implementação não seja complexa e nem conflitante, pelo fato de alguns parâmetros o serem.

Por todos os esforços decorrentes deste trabalho, ao realizar uma implementação diferenciada das já existentes, é evidente o quanto o assunto é amplo e pode ser explorado. O sistema apresentado neste trabalho está longe de ser o ideal e sabe-se muito bem que ele pode ser melhorado em muitos aspectos. Esta questão é abordada adiante, na análise crítica dos resultados e nas conclusões acerca do sistema em si.

A modelagem de um algoritmo de escolha de cabines usando lógica nebulosa aliada a um pré-controle (responsável por realizar as alocações das chamadas) mostrou-se ao longo do projeto uma solução bastante acertada, visto que convergiu em uma alternativa de controle bastante flexível e de resultados eficazes.

A pré-alocação levando o sentido em consideração apresentou o melhor consumo de energia e melhor tempo médio de viagem em praticamente todo os casos, além de ter tido marcas tão boas quanto os outros controles para os tráfegos de todos horários residenciais, revelando que este controle aparenta ser o mais adequado para prédios onde a população por andar não é tão elevada e onde os tráfegos intensos não duram muito (com um número restrito de elevadores). Ainda deve ser mencionado o aspecto de que, ao se alocar as cabines conforme seus sentidos de

movimentação, o passageiro percorre uma trajetória de sentido único até o seu destino, o que gera um nível maior de qualidade de serviço para o usuário final.

Em contrapartida, a pré-alocação levando em conta o sentido dos elevadores não atuou muito bem no tráfego hospitalar, uma vez que tal alocação de chamadas, ao priorizar o atendimento em um sentido único, acaba por otimizar também o número de passageiros por cabine, o que contribui para que algumas chamadas tenham de ser refeitas (quando o elevador que vai atender uma chamada externa já se encontra lotado). Tais chamadas refeitas contribuem enormemente para piorar a qualidade do serviço (tempo de espera de alguns dobra, e o número de esperas longas aumenta). Tal característica poderia ser evitada com a inserção de uma cabine de atendimento a mais (os testes com três cabines realmente mostraram um bom desempenho frente aos outros controles, com também três cabines).

O pré-controle exaustivo mostrou-se melhor para o tráfego hospitalar, pois define a posição da nova chamada levando em conta apenas critérios de velocidade de atendimento, ignorando o consumo de energia, tempo de viagem e o caminho que o elevador vai tomar. Desta forma, as chamadas são atendidas rapidamente, mas algumas viagens são mais longas e percorrem "zig-zags", porém no balanço final a média do serviço é muito boa, e a variável mais importante neste caso, porcentagem de esperas longas, teve o seu melhor resultado.

Mesmo com estes resultados, possíveis modificações na modelagem das bases nebulosas, como redefinição de patamares e ajuste de algumas regras, podem convergir para uma eficiência ainda maior do controle nebuloso para cada caso. No entanto, seria ainda necessária a realização de diversos testes, e conseqüentemente de mais tempo, para então se chegar à melhor configuração possível das regras para cada tráfego.

Uma vez que os controles apresentados já demonstraram eficiência em determinados tipos de tráfegos, basta chavear entre os tipos de controle conforme seja a expectativa de tráfego,

formando um controle híbrido que muda, de acordo com o tipo de tráfego, para o correspondente controlador mais eficiente para a tarefa.

Além da questão da eficiência demonstrada pelos controles desenvolvidos, o ambiente de testes (simulador) criado, a modularidade para se inserir um pré-controle independente e a flexibilidade de se alterar a base do controlador nebuloso, possibilita o teste de diversos tipos de controle para diversos tipos de edifícios. De modo que o resultado final converge para uma ferramenta de grande auxilio para se desenvolver e validar controles de elevadores customizados.

GLOSSÁRIO

Batches	arquivos em lote com com o objetivo de automatizar tarefas
Bugs	panes ou defeitos encontrados no software;
Bunching	fenômeno onde os elevadores começam a andar lado a lado e competir pelas chamadas mais próximas;
Cabine	vagão ou elevador, é meio de transporte vertical do qual os passageiros se utilizam;
Chamada	requisição por atendimento para o sistema de controle;
Chamada Externa	chamada ocorrida no Hall, fora da cabine;
Chamada Interna	chamada ocorrida dentro da cabine;
Fila de atendimento	fila composta por chamadas a serem atendidas por uma cabine;
Fuzzy	termo técnico referente à Lógica Nebulosa;
FuzzyJ	API Java para implementação de conceitos *Fuzzy*;
Hall Call	chamada externa;
Hall Call Time	tempo esperado por uma chamada externa;
Interval	tempo médio entre as partidas de elevadores do térreo;
Matlab	ferramenta de simulação com ambiente de desenvolvimento de sistemas baseados em lógica nebulosa;
Parsing	análise sintática;
Selective Collective	controle similar ao IFC;
Single Call Control	sistema primitivo de controle para elevadores;
zig-zags	ato de percorrer pequenos caminhos trechos em sentidos opostos;
Zoning	técnica de controle por zoneamento.

ANEXO A – Regras utilizadas por modelagem

1. Modelagem residencial considerando que o fator de utilização (FATU) deveria ser equilibrado entre as cabines:

TEA	FATU	OC	DISP
ALTO	ALTO	ALTA	MUITO_BAIXA
ALTO	ALTO	MEDIA	BAIXA
ALTO	ALTO	BAIXA	MEDIA
ALTO	MEDIO	ALTA	BAIXA
ALTO	MEDIO	MEDIA	MEDIA
ALTO	MEDIO	BAIXA	MEDIA
ALTO	BAIXO	ALTA	BAIXA
ALTO	BAIXO	MEDIA	BAIXA
ALTO	BAIXO	BAIXA	MEDIA
MEDIO	ALTO	ALTA	MUITO_BAIXA
MEDIO	ALTO	MEDIA	MEDIA
MEDIO	ALTO	BAIXA	ALTA
MEDIO	MEDIO	ALTA	BAIXA
MEDIO	MEDIO	MEDIA	MEDIA
MEDIO	MEDIO	BAIXA	ALTA
MEDIO	BAIXO	ALTA	BAIXA
MEDIO	BAIXO	MEDIA	MEDIA
MEDIO	BAIXO	BAIXA	ALTA
BAIXO	ALTO	ALTA	BAIXA
BAIXO	ALTO	MEDIA	MEDIA
BAIXO	ALTO	BAIXA	ALTA
BAIXO	MEDIO	ALTA	MEDIA
BAIXO	MEDIO	MEDIA	ALTA
BAIXO	MEDIO	BAIXA	MUITO_ALTA
BAIXO	BAIXO	ALTA	MEDIA
BAIXO	BAIXO	MEDIA	ALTA
BAIXO	BAIXO	BAIXA	MUITO_ALTA

Tabela 12 – Tabela de regras nebulosas FATU equilibrado.

2. Modelagem residencial considerando que o fator de utilização (FATU) não deveria ser equilibrado entre as cabines:

TEA	FATU	OC	DISP
ALTO	ALTO	ALTA	BAIXA
ALTO	ALTO	MEDIO	BAIXA
ALTO	ALTO	BAIXA	MEDIA
ALTO	MEDIO	ALTA	BAIXA
ALTO	MEDIO	MEDIO	MEDIA
ALTO	MEDIO	BAIXA	MEDIA
ALTO	BAIXO	ALTA	MUITO_BAIXA
ALTO	BAIXO	MEDIO	BAIXA
ALTO	BAIXO	BAIXA	MEDIA
MEDIO	ALTO	ALTA	BAIXA
MEDIO	ALTO	MEDIO	MEDIA
MEDIO	ALTO	BAIXA	ALTA
MEDIO	MEDIO	ALTA	BAIXA
MEDIO	MEDIO	MEDIO	MEDIA
MEDIO	MEDIO	BAIXA	ALTA
MEDIO	BAIXO	ALTA	MUITO_BAIXA
MEDIO	BAIXO	MEDIO	MEDIA
MEDIO	BAIXO	BAIXA	ALTA
BAIXO	ALTO	ALTA	MEDIA
BAIXO	ALTO	MEDIO	ALTA
BAIXO	ALTO	BAIXA	MUITO_ALTA
BAIXO	MEDIO	ALTA	MEDIA
BAIXO	MEDIO	MEDIO	ALTA
BAIXO	MEDIO	BAIXA	MUITO_ALTA
BAIXO	BAIXO	ALTA	BAIXA
BAIXO	BAIXO	MEDIO	MEDIA
BAIXO	BAIXO	BAIXA	ALTA

Tabela 13 - Tabela de regras nebulosas FATU não-equilibrado.

ANEXO B – Formato dos arquivos de configuração das regras nebulosas

A seguir é apresentado o formato de um arquivo de configuração de regras para o sistema de controle nebuloso.

- Definição de variáveis:

```
<VARIAVEL id="TEA" limInf="0" limSup="60" medida="ut">
    <TERMO id="baixo" vertices="0,10,15,35"/>
    <TERMO id="medio" vertices="15,27.5,32.5,45"/>
    <TERMO id="alto" vertices="25,45,50,60"/>
</VARIAVEL>
```

Sobre os campos utilizados, pode-se dizer que o "id" é o nome associado à variável ou ao termo, "limInf" e "limSup" são os limites inferior e superior do universo de discurso e "vertices" especifica os vértices da função de pertinência relacionada ao termo. Neste último caso, se forem 3 vértices, a função será do tipo triangular, e se forem 4 vértices, a função será trapezoidal. Podem ser definidos quantos termos forem necessários para cada variável.

- Definição de regras:

```
<REGRA id="tea_alto_fatu_baixo_oc_media" num="2">
        <ANTECEDENTE id="FATU_baixo">
                <VARIAVEL id="FATU"></VARIAVEL>
                <TERMO id="baixo"></TERMO>
        </ANTECEDENTE>
        <ANTECEDENTE id="TEA_alto">
                <VARIAVEL id="TEA"></VARIAVEL>
                <TERMO id="alto"></TERMO>
        </ANTECEDENTE>
        <ANTECEDENTE id="OC_media">
                <VARIAVEL id="OC"></VARIAVEL>
                <TERMO id="media"></TERMO>
        </ANTECEDENTE>
        <CONCLUSAO id="DISP_baixa">
                <VARIAVEL id="DISP"></VARIAVEL>
                <TERMO id="baixa"></TERMO>
        </CONCLUSAO>
</REGRA>
```

Neste caso, os antecedentes são as entradas da regra e a conclusão é a saída. O "id" de um antecedente ou conclusão é composto pelo nome da variável e o valor de seu termo associado.

ANEXO C - Estudo de tráfego de elevadores

A fim de definir melhor as necessidades de atendimento do serviço de elevadores, para cada tipo de edifício e, conseqüentemente, delinear as variáveis a serem avaliadas no controle, além de suas importâncias para cada caso, realizou-se um estudo das normas que regem padrões mínimos de atendimento e os tráfegos usuais para cada caso.

Os sistemas de elevadores podem ser implementados nos mais diversos estabelecimentos, desde prédios residenciais, *shopping centers*, hotéis, hospitais, edifícios comerciais, entre outros. Para a realização do estudo de tráfego em questão, foram levados em conta, apenas prédios comerciais, residenciais e hospitalares, havendo maior ênfase neste último caso, visto que os demais são mais simples e suas principais características são de conhecimento geral.

Normas para elevadores comerciais e residenciais:

No caso dos elevadores comerciais e residenciais, estes devem seguir a norma ABNT NBR NM 207, que estabelece que a capacidade de atendimento das chamadas deve ser tal que uma porcentagem da população (10% para residenciais e 12% para comerciais) seja atendida dentro de 5 minutos pelo sistema de elevadores (em uma situação de *up-peak*) (19). O tráfego usado para tal modelagem deve ser o de *up-peak*, visto que se trata do padrão que mais demanda a capacidade de transporte dos elevadores, e assim sendo, uma vez que o sistema atenda satisfatoriamente o *up-peak*, poderá atender também outros padrões de tráfegos.

Além de especificar a capacidade mínima de transporte em 5 minutos, as normas ABNT especificam um tempo máximo de espera de 80, 60, 50 e 40 segundos para respectivamente 1, 2, 3 e 4 cabines/ elevadores, que não se aplica a prédios residenciais e deve a todo custo ser evitado.

A fim de estimar tais valores para o sistema e conjunto de elevadores, a norma fornece alguns valores-padrão para serem usados, como o do número de paradas prováveis de elevadores, capacidade das cabines baseada em suas áreas, métodos para cálculo de variáveis como Capacidade Nominal de Tráfego, Intervalo de Tráfego e outras, além de definir alguns critérios para a relação de pessoas por andar, que corresponde a:

- Uma pessoa por 7 metros quadrados de sala para escritórios;
- Duas pessoas para apartamentos de um dormitório, quatro para dois dormitórios e cinco para três dormitórios;
- 2,5 pessoas por leito de hospital;
- 1,4 pessoa por vaga de garagem;

Comparando-se com os casos reais estudados, a modelagem de relação de pessoas por andar prevista pelas normas ABNT é um tanto quanto pessimista, haja visto que a mesma estimula valores muito maiores dos que os observados em campo.

Estudo do tráfego de prédios residenciais

Em prédios residenciais, é comum haver dois tipos de elevadores, os elevadores de serviços e os elevadores sociais. Ambos costumam serem utilizados normalmente pelos moradores, porém os elevadores de serviços também são usados pelos funcionários para retirar lixo, entregar correspondências, jornais e etc., dependendo da política de uso de cada edifício.

Em finais de semana, o tráfego de prédios residenciais é muito pouco padronizado. No entanto, nos dias úteis, o tráfego costuma ser principalmente de *down-peak* no período das 6:30 às 8:30, *up-peak* das 18:00 às 20:00 e ambos os tráfegos, porém com menor intensidade, entre 12:00 e 14:00 para os elevadores sociais, e de forma semelhante para os elevadores de serviço.

Além disso, os elevadores de serviço costumam ser utilizados em um período do dia para entrega e retirada de objetos, faxina ou realização de outros serviços pelos funcionários. Tal uso ocorre, normalmente, de forma bastante sistemática e em um horário de baixa utilização dos elevadores, onde o funcionário percorre todos os andares seqüencialmente realizando seu serviço em cada andar.

Os elevadores residenciais costumam ser os modelos de até 8 passageiros, o correspondente à carga máxima de 600kg, e que devem possuir como pré-requisitos: largura mínima da cabina de 1,1m, profundidade mínima de 1,4m e abertura lateral mínima da porta de 0,8m.

Para prédios residenciais, nota-se que, de modo geral, existem dois elevadores, um de serviço e outro social, para cada prédio de quatro quartos por andar até um total de 6 andares, e podendo haver um elevador social adicional para mais de 8 andares, apesar de terem sido constadas exceções. O número de elevadores está efetivamente mais relacionado com o "luxo" do apartamento, medido pelo preço do condomínio e metro quadrado, do que com as restrições das regras ABNT. A maioria dos sistemas de elevadores residenciais ainda não possui sistema de controle de elevadores, seus elevadores funcionam com atendimento dedicado com um painel de chamadas externas para cada cabine.

Também se deve levar em conta que os tráfegos de *down-peak* e *up-peak* tem como destino e origem, respectivamente, tanto o andar térreo quanto a garagem, sendo que o mais comum é que haja apenas um andar para a garagem. No entanto, há prédios que não possuem

garagem, pois os carros dos moradores encontram-se no andar térreo, e outros que possuem dois andares de garagem.

Estudo do tráfego de prédios comerciais

Os prédios comerciais seguem um padrão muito parecido com os prédios residenciais, possuindo elevadores de serviço e sociais. Seus tráfegos costumam ser do tipo *up-peak* entre as 07:00 e às 9:00, *down-peak* das 18:00 às 20:00 e com um tráfego híbrido começando como *down-peak* e tendo seu fluxo alterado gradualmente para *up-peak* no período de 12:00 às 14:00. Durante o resto do dia, o tráfego é bastante irregular, mas quase sempre partindo do andar térreo ou andar correspondente à garagem (sendo relativamente comum haver mais de um andar para a garagem) para os outros andares e o correspondente ao fluxo oposto. O fluxo entre andares de escritórios é praticamente inexistente.

Ao contrário dos prédios residenciais, os elevadores (normalmente apenas um) de serviços são exclusivos para transporte de cargas, lixo e outras operações não relacionadas com o público alvo do sistema e raramente são utilizados fora deste propósito, situando-se inclusive fora do sistema de controle dos elevadores.

Os prédios comerciais costumam possuir elevadores com capacidade de 10 a 14 passageiros, e carga máxima correspondente entre 800 a 1000 kg, com portas de largura maior que 1.100 metros, permitindo a passagem simultânea de duas pessoas, para andares de 500 a 600 metros quadrados e que comportam cerca de 50 a 80 pessoas por andar.

Normas para elevadores hospitalares:

No caso dos elevadores de prédios hospitalares, ou Estabelecimentos Assistenciais de saúde (nome com que estes são qualificados e diferenciados dos demais estabelecimentos), deve-se seguir a resolução RDC 50 de 21/02/2002 da Agência Nacional de Vigilância Sanitária. E fora tal resolução, a politica de elevadores dos hospitais deve seguir as normas ABNT NBR 14712, para elevadores elétricos de carga, monta-carga e elevadores de maca, e ABNT NBR NM 207, para elevadores elétricos de passageiros, esta última cujas determinações já foram abordadas no tópico das normas para elevadores residenciais.

O hospital deve ser capaz de transportar em cinco minutos 8% da população onde houver elevador monta-carga (elevadores próprios de menor volume e excluídos para o transporte de cargas) para o serviço de alimentação e material e 12% da população quando não houver (20).

Para os elevadores de passageiros, além de terem de atender os já citados 8% da população em 5 minutos, ao menos um deles deve seguir a norma ABNT NBR 13994 que compete aos elevadores para transporte de pessoas com deficiências.

Tráfego de prédios hospitalares

Partindo para uma análise mais prática, entrou-se em contato com um grande hospital público do pais e foi feita uma breve análise sobre a disposição dos seus serviços, estrutura do edifício e fluxo de pessoas.

Foram analisados dois prédios de tal hospital, um deles, correspondente à área de internação, possuindo onze andares servidos por elevadores e o outro, área de emergência,

possuindo quatro andares. Em ambos os prédios, todos os elevadores de funcionários e pacientes são grandes o bastante para que caiba uma maca.

No primeiro prédio, o fluxo de pessoas é bastante homogêneo entre os onze andares e nenhum dos andares precisa de atenção especial dos elevadores (todos devem ter a mesma prioridade de atendimento), com exceção do andar do sub-solo, que é atendido apenas pelos elevadores sujo e limpo.

As áreas compreendidas por este prédio e seus respectivos andares são:

- subsolo acesso para emergência e ambulatório
- térreo: área administrativa e acesso para prédio da emergência e ambulatório
- sobre-loja: área administrativa e acesso para prédio da emergência
- 2 andar: cardiologia
- 3 andar: neurologia e cardiovascular
- 4 andar: cirurgia geral
- 5 andar: policlínica
- 6 andar: cirurgia pediátrica e berçário
- 7 andar: pediatria
- 8 andar: clínica médica
- 9 andar: clínica médica
- 10 andar: ortopedia
- 11 andar: urologia e hemodiálise

Em tal prédio, há seis elevadores. Dois deles são sociais (destinados aos funcionários, visitantes e pacientes andando), dois elevadores para itens sujos e dois elevadores para itens

limpos e pacientes com macas. Apenas os elevadores sujo e limpo atendem o subsolo, visto que a comida, roupas e lixo são tratados no subsolo.

Os elevadores sociais têm alto tráfego de subida e descida às 7, às 13 e às 19 horas devido à troca de turnos dos funcionários, sendo que o tráfego começa como apenas de subida e inverte para o fluxo contrário até cerca de 30 minutos depois do início das trocas de turnos, e grande fluxo tanto de subida quanto de descida entre 13:00 e 14:00 (por causa do horário de almoço dos funcionários) e entre 15:00 (tráfego de subida) e 16:00 (tráfego de descida), por se tratar do horário de visita aos pacientes.

Os elevadores que carregam itens considerados sujos (roupas utilizadas, lixo, resíduo hospitalar e etc) não possuem um horário definido de fluxo, no entanto, como o elevador é alocado exclusivamente para a retirada destes itens, não há concorrência para uso. Normalmente o funcionário responsável sobe até o último andar e desce recolhendo os itens de andar em andar, de forma análoga a muitos serviços de limpezas em hotéis.

Para os elevadores limpos (que carregam as roupas limpas e esterilizadas e a comida a ser servida aos pacientes), a roupa limpa é entregue uma vez por dia as 6 horas da manhã e a comida é levada às 7:30, 11:00, 14:00 17:00 e 20:00 (sendo que para ambas, o carrinho sobe e vai atendendo todos os andares um por um seqüencialmente). O fluxo de pacientes de maca, assim como de outros pacientes para os outros elevadores, é bastante irregular, não sendo possível fazer uma padronização fiel do mesmo. Devido ao fato dos elevadores quebrarem com certa freqüência, elevadores limpos são também utilizados pelos funcionários na ausência dos elevadores sociais.

A área de emergência lida com os serviços mais críticos e é dividida da seguinte maneira, sendo que o andar do sub-solo também é atendido apenas pelos elevadores sujo e limpo:

- sub-solo acesso para emergência e ambulatório, transporte, material esterilizado, farmácia

- térreo: acesso a área administrativa e pronto socorro
- 2 andar: centro cirúrgico (maior fluxo de pacientes as 7:30 h e 13:30 h)
- 3 andar: neurocirurgia e transplante (maior fluxo de visitantes)
- 4 andar: UTI

Como se pode perceber, o pronto-socorro, área de maior criticidade de atendimento, se localiza no térreo e sua prioridade de demanda é um tanto quanto independente dos elevadores. No entanto, há ainda algumas situações como o trânsito de passageiros do pronto-socorro para o centro cirúrgico, onde há a necessidade de rápido atendimento.

Neste prédio, há seis elevadores, dois quais dois são sociais, há um elevador sujo e um exclusivamente para itens limpos, além de dois elevadores para pacientes em maca.

Os elevadores sociais têm alto tráfego entre as 7:00 às 8:00, 13:00 às 14:00, 16:00 às 17:00 e 19:00 às 20:00, onde tais tráfegos (exceto pelo tráfego das 16:00 que se refere ao periodo de visitação) correspondem à troca de turno dos funcionários e são caracterizados inicialmente por uma alta demanda de pessoas que saem do térreo para os outros andares e que com o andar do tempo se converte para o fluxo inverso (andares para o térreo).

Os elevadores sujo e limpo seguem o fluxo semelhante ao do prédio de internação. Os dois elevadores para paciente em maca não possuem nenhum horário de pico de atendimento (visto que não foi observado pelo hospital um padrão na chegada e saída de pacientes), mas costumam atender um maior fluxo do Pronto Socorro para o Centro Cirúrgico, e do Centro Cirúrgico para a UTI.

Estes elevadores são dotados de um botão que permite que o passageiro faça a sua chamada interna ao elevador ser atendida antes que a chamada atual e as outras que estavam na fila. A idéia é que houvesse um ascensorista controlando este teclado, no entanto, não há nenhum

funcionário dedicado para tal controle desde o início de sua operação. Portanto, tal controle acaba sendo feito pelos funcionários e percebe-se que muitos fazem uso dessa tecla mesmo quando não lidando com uma emergência e, conseqüentemente, atrapalhando o fluxo ideal de atendimento.

Priorização de variáveis e considerações sobre cada caso

Para os hospitais, devido à aleatoriedade das chegadas de emergência, o mais interessante é tentar diminuir ao máximo o número de esperas longas e assim contribuir para que uma chamada crítica não seja afetada pelo tempo de atendimento dos elevadores. Conforme foi visto, o teclado emergencial não é utilizado corretamente, e portanto não constará na modelagem estudada, de forma que, ao priorizar uma variável que contribua ara que haja um índice baixíssimo de esperas longas, o problema será resolvido. O tempo médio de espera de atendimento deverá possuir um peso alto para que as chamadas esperem menos tempo e garantir o cumprimento das normas ABNT ao qual o sistema de elevadores hospitalar deve obedecer.

Na modelagem dos elevadores hospitalares, não serão levados em conta os elevadores dedicados tais como os elevadores sujos e limpos, uma vez que seus tráfegos são restritos e agendados.

No caso dos elevadores comercias, as variáveis mais importantes são a porção de esperas longas e principalmente o tempo médio de atendimento, visto que nestes casos a percepção do serviço para o usuário é extremamente importante, além das necessidades de seguir as normas ABNT. Os elevadores de serviços não constarão na modelagem, visto que não são atendidos pelos sistemas de controle de elevadores comercias e se encarregam de atender uma demanda muito específica que não deve ser compartilhada com os outros elevadores.

O sistema de elevadores quando implementado para edifícios residenciais, deve levar mais em consideração o consumo de energia (a importância de tal variável pode ser notada no fato de que após o fim do racionamento de energia na cidade de São Paulo, muitos desses edifícios continuaram racionalizando o uso dos seus elevadores) e fator de utilização do que as variáveis relacionadas com tempo de atendimento (até porque neste caso as pessoas são menos sensíveis aos atrasos dos elevadores e as próprias normas ABNT não são tão rígidas), visto que o resultado destas afeta diretamente no valor do condomínio pago pelos moradores.

ANEXO D - Diagrama de Classes

Pode-se destacar que o pacote "Elevator Simulator" da Figura 5 constitui-se, no caso aqui exposto, das classes Prédio, Andar, Cabine, Pessoa, Simulador, Evento e Clock, como pode ser visto na Figura 12. Prédio é composto por Andar e Cabine, e depende de Simulador. Clock e Evento são agregados de Simulador. Simulador depende de Sistema_Controle e Pessoa. Já, SysControl, que agrega CabineCtl, faz parte do pacote "Elevator Control System", composto também por Arbitro, CabineCtl, Chamada, PreControle e ProcessadorFuzzy. Este pacote refere-se ao sistema de controle dos elevadores propriamente dito. O SysControl depende de Arbitro que, por sua vez, depende de ProcessadorFuzzy. O ProcessadorFuzzy usa Cabine e CabineCtl para adquirir os valores das variáveis de controle. CabineCtl usa PreControle para estimar uma dessas variáveis.

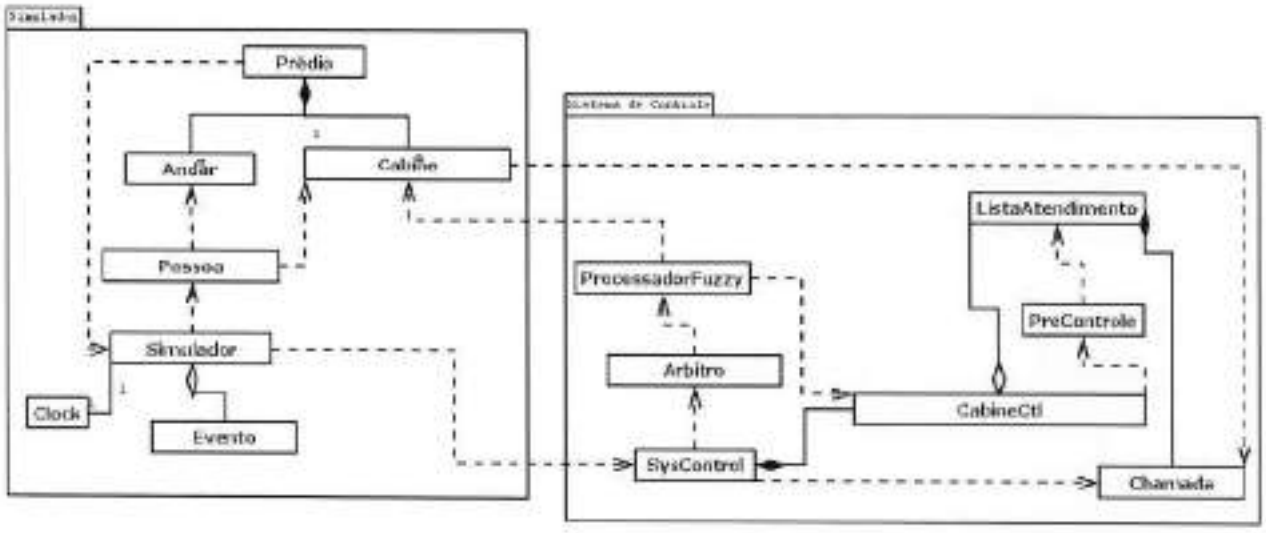

Figura 12 – Diagrama de Classes do Simulador em UML.

ANEXO E – Diagrama de Casos de Uso

Com o objetivo de facilitar a identificação dos requisitos de software, bem como auxiliar na modelagem do sistema, foi elaborado um diagrama de casos de uso, utilizando-se da notação UML (21), como pode ser visto na Figura 13.

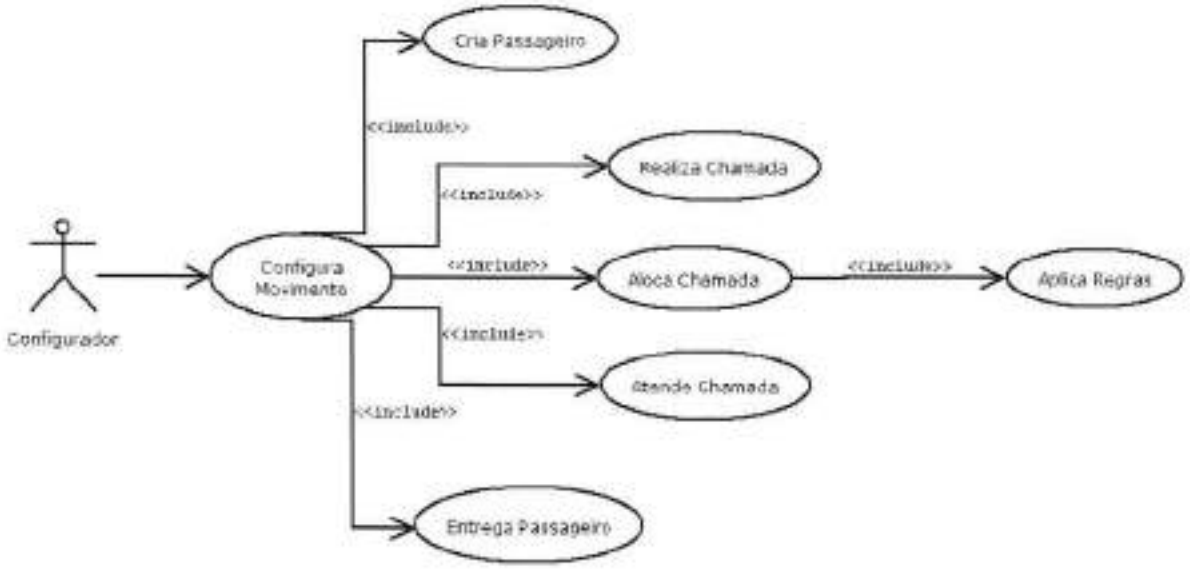

Figura 13 – Diagrama de Casos de Uso.

A seguir é descrito cada um dos casos de uso identificados acima:

- **Configura Movimento:** O configurador desencadeia o funcionamento do simulador de duas formas. Uma é através de um arquivo de configuração que é carregado pelo programa e executado em seguida. Outra é o configurador seguir os passos descritos no Anexo G para entrada manual de atuações no sistema.

- **Cria Passageiro:** Através da entrada realizada pelo configurador, o sistema deve criar o passageiro adicionando-o ao andar origem do passageiro.
- **Realiza Chamada:** O passageiro adicionado ao andar terá uma chamada externa relacionada a ele. As chamadas internas são criadas no momento em que a sua chamada externa associada é atendida, neste caso, a pessoa é adicionada à cabine atendente, e a chamada interna é criada. Esta chamada deve ser atendida por uma das cabines do sistema.
- **Aloca Chamada:** A chamada criada, para ser atendida, deve ser alocada à fila de atendimento de uma das cabines. Este caso de uso se encarrega de realizá-lo, ao utilizar-se do caso de uso "Aplica Regras".
- **Aplica Regras:** Este caso de uso é responsável por aplicar regras de controle às variáveis do sistema, identificando a cabine que melhor atende a chamada em questão.
- **Atende Chamada:** Uma chamada, ao ser alocada, deve ser atendida. Para isso, este caso de uso dispara um processo no qual a cabine se movimenta em direção ao andar alvo da chamada, de modo a poder atendê-la. No momento em que a cabine atinge o andar alvo, o passageiro é transferido do andar para a cabine.
- **Entrega Passageiro:** O passageiro, ao chegar ao seu andar destino, deve ser transferido da cabine para o andar.

ANEXO F – Descrição da Interface do Simulador

A interface que representará a dinâmica de um prédio real está ilustrada na Figura 14. O usuário deverá carregar um arquivo de configuração contendo a lista completa de chamadas a serem executadas durante a simulação, ou então, ele poderá criar estas chamadas interagindo com o programa através desta interface.

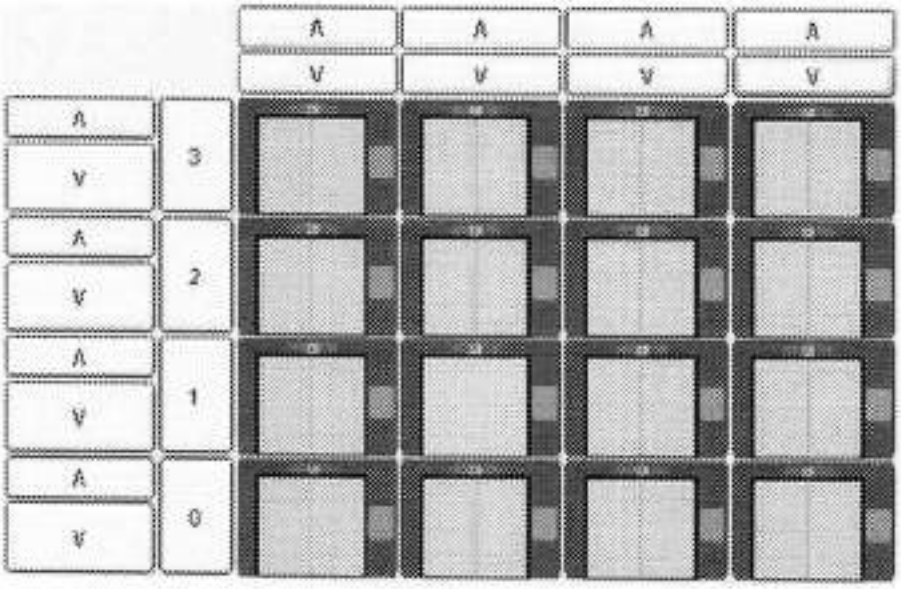

Figura 14 – Representação do prédio.

Na interface, pode-se identificar os andares e seus respectivos *halls* de elevadores. Um andar é representado por uma coluna horizontal, como visto na Figura 15, enquanto uma cabine é representada por uma coluna vertical, como pode ser visto na Figura 16.

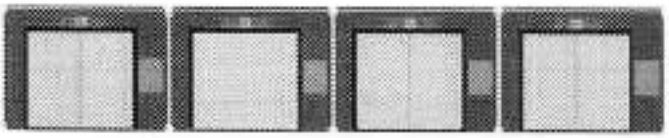

Figura 15 – Representação de um andar.

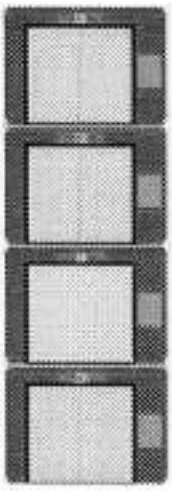

Figura 16 – Representação de uma cabine.

Em um dado andar, o passageiro pode encontrar as portas dos elevadores abertas, fechadas, ou em "passagem". Uma porta aberta é representada como na Figura 17, enquanto que uma porta fechada é representada como na Figura 18. Uma porta aberta indica que a cabine se encontra parada naquele andar.

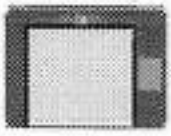

Figura 17 – Representação de uma porta aberta.

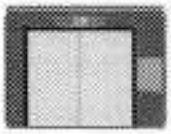

Figura 18 – Representação de uma porta fechada.

Uma porta está em "passagem" quando a cabine se encontra naquele andar apenas transitoriamente, ou seja, está em movimento para outro andar. A Figura 19 mostra como a interface representa uma porta em "passagem".

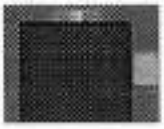

Figura 19 – Representação de uma porta em "passagem".

O painel à direita da representação do prédio contém botões que permitem que o usuário possa interagir com o programa. Este painel é destacado na Figura 20. A interação citada consiste na criação de novos passageiros, através da seleção do andar de origem, representado pelos botões numerados do painel de interação, e do sentido de locomoção desejado para o passageiro em questão, representado pelos botões com as setas "/\" e "\/".

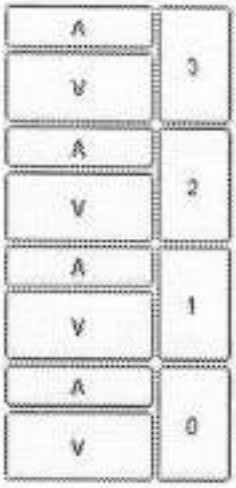

Figura 20 – Painel de interação para a criação de novos passageiros.

Após a seleção realizada na etapa acima descrita, surgirá um novo painel, representativo do painel interno da cabine, no qual o usuário deverá informar o destino do passageiro em questão. Este painel está ilustrado na Figura 21.

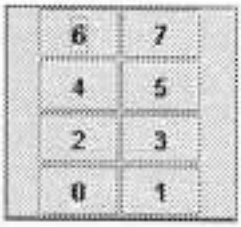

Figura 21 – Representação do painel interno da cabine.

Através destes dois painéis, o usuário pode selecionar a origem, o destino e o sentido do passageiro que acabou de ser criado, simulando a chegada de um passageiro no andar de origem. O programa, a partir deste momento, iniciará o processo de atendimento a este passageiro. Uma das cabines será designada a atender esta chamada, portanto, ele se deslocará até o andar da origem da chamada, e em seguida se deslocará até o andar do destino da chamada. No caso em

que houver mais de uma chamada, esta movimentação será regida por uma política de alocação de chamadas, a qual indicará o trajeto de cada cabine.

O painel superior à representação do prédio indica o sentido atual de cada cabine. Caso a seta "/\" esteja realçada, o sentido atual é para cima, caso a seta "\/" esteja realçada, o sentido atual é para baixo, caso nenhuma das setas estejam realçadas, então o elevador está parado.

ANEXO G - Pseudocódigo dos eventos do elevador

A seguir segue uma descrição mais detalhada dos principais eventos e processos mencionados anteriormente.

1 – Evento Pessoa

```
for(instante=0; instante< fim_simulacao; instante++){
  if(existe_evento == true){
     if(evento.tipo == pessoa){

        //adiciona pessoa ao andar de chegada
        prédio.addPessoaAndar(pessoa);

        controle.addChamada(tipo = externa);

     }
  }
}
```

2 - Inicia atendimentos para cada cabine:

```
for(instante=0; instante< fim_simulacao; instante++){
  for(cada cabine){

     //nova_chamada_externa => alvo>=0
     //identifica andar a ser atendido;
     alvo = cabine.getAlvo();

     if(fila_atendimento == vazia && nova_chamada_externa == true){

        //infere o sentido da cabine;
        cabine.atualizaSentido();

        addEvento(tipo = elevador, andar = andar mais proximo no sentido inferido);

        //fila_atendimento = !vazia;
        setPrecisaAnime(false);

        //andar_atual = andar;
        atualizaAndarDaCabine();

     }
  }
}
```

3 – Evento Elevador

```
for(instante=0; instante< fim_simulacao; instante++){
  if(existe_evento == true){
    if(evento.tipo == elevador){

      //atualiza distancia percorrida;
      cabine.incDistancia();

      if(andar_atual == andar_alvo){
        //faz animacao;
        simView.abrePorta();

        //prédio.transfereAndarElevador();
        transfere pessoas do elevador pro andar;
        cabine.atualizaSentidoFim();

        //verifica quantas pessoas podem entrar na cabine

        transfere pessoas do andar pro elevador;
        if(pessoa entra no elevador){
            addEvento(tipo = painel);
        }

        atualiza indicadores do simulador;

        addEvento(tipo = porta);

        //retira as chamadas atendidas neste andar
        cabine.atualizaFila{
          for(chamada = fila_atendimento.getNextChamada(); nao atingiu fim da fila; pega proxima chamada){
            if(andar da chamada == andar_atual && chamada.tipo == interna){
              retira chamada da fila;
            }
          }

          //infere sentido da cabine; (podia usar atualizaSentido)

          for(chamada = fila_atendimento.getNextChamada(); nao atingiu fim da fila; pega proxima chamada){
            if(andar da chamada == andar_atual && chamada.tipo == externa){ (e se o sentido for diferente?)
              retira chamada da fila;
            }
          }

          //infere sentido da cabine
        atualizaSentidoMeio();

        }
      }else{ // andar_atual != andar_alvo
          addEvento(tipo = elevador, andar = andar mais proximo no sentido inferido);

        //andar_atual = andar;
        atualizaAndarDaCabine();
      }
    }
  }
}
```

4 – Evento Painel

```
for(instante=0; instante< fim_simulacao; instante++){
  if(existe_evento == true){
```

```
        if(evento.tipo == painel){

            addChamada(tipo = interna);

        }
    }
}
```

5 – Evento Porta

```
for(instante=0; instante< fim_simulacao; instante++){
    if(existe_evento == true){
        if(evento.tipo == porta){

            faz animacao;

            if(chamadas a serem atendidas == true){
                addEvento(tipo = elevador);

                //atualiza sentido da cabine;
                atualizaSentidoMeio();

                //atualiza andar da cabine;
                atualizaAndarDaCabine();

            }else{
                cabine.setSentido(parada);
                fila_atendimento = vazia;
            }

        }
    }
}
```

ANEXO H – Manual do Usuário - Simulador

Início

Para iniciar o programa, deve-se escolher o Sistema de Controle a ser utilizado durante a simulação e, em seguida, o tipo de entrada a ser fornecida ao simulador: manual ou por arquivo. A janela onde estas configurações podem ser realizadas é representada na Figura 22.

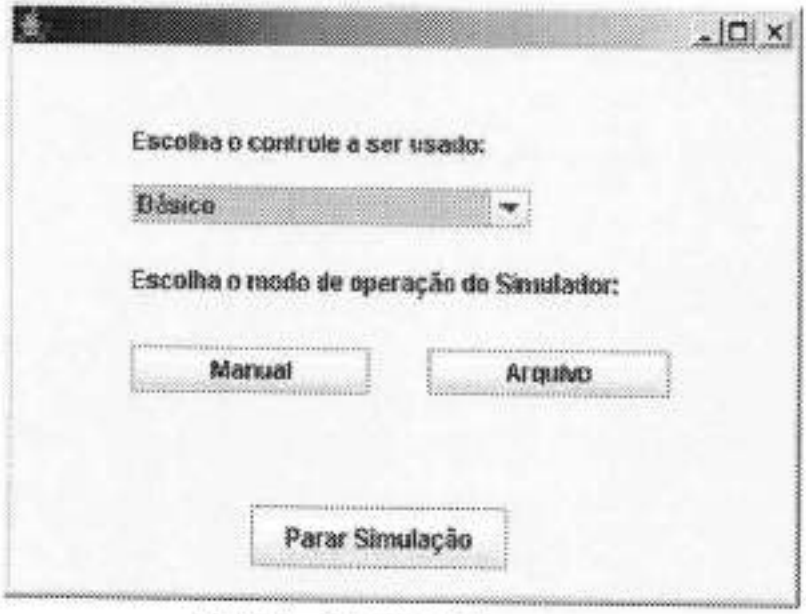

Figura 22 - Menu inicial do programa.

Pode ser notado também que existe um botão que permite a parada da simulação.

Criação Automática de Pessoas

Na escolha pelo processo automático, o arquivo indicado será carregado, e as informações sobre as entradas do programa serão utilizadas nos instantes também indicados no próprio arquivo.

Criação Manual de Pessoas

Para criar manualmente um novo passageiro, deve-se proceder da seguinte forma:

1. Clicar sobre o botão correspondente ao sentido desejado, no andar que será o andar origem desta nova pessoa. No caso da Figura 23, o círculo em destaque indica uma seleção possível. Basta clicar nesta seta "/\", relativa ao andar "1", para iniciar o processo de criação de uma nova pessoa. Note que não é preciso clicar sobre o botão relativo ao número do andar, pois as setas já estão ligadas ao andar correspondente.

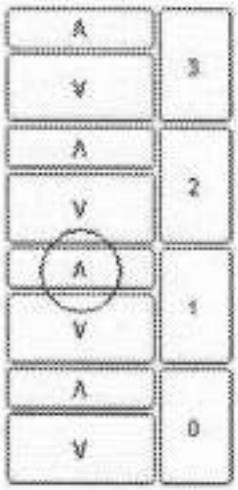

Figura 23 – Painel de criação com os botões de sentido e andar de origem.

2. Após serem selecionados o andar origem e o sentido, uma nova janela com o painel correspondente ao painel interno de uma cabine será gerada, onde pode ser selecionado o andar destino. O painel interno é ilustrado na Figura 24. Para selecionar o andar destino, basta clicar sobre o botão indicado com o número do andar correspondente.

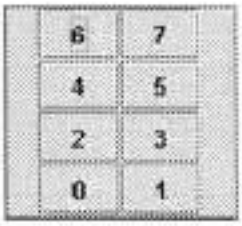

Figura 24 – Painel interno.

3. Neste ponto, o simulador instancia uma nova pessoa, passando os dados andar de origem, andar de destino e sentido escolhido. Associada a esta pessoa será gerada uma chamada a ser passada para o sistema de controle de elevadores. Esta chamada será responsável pelo atendimento da pessoa criada. Através da interface ilustrada na Figura 25, será possível acompanhar a movimentação das cabines e os atendimentos às pessoas criadas.

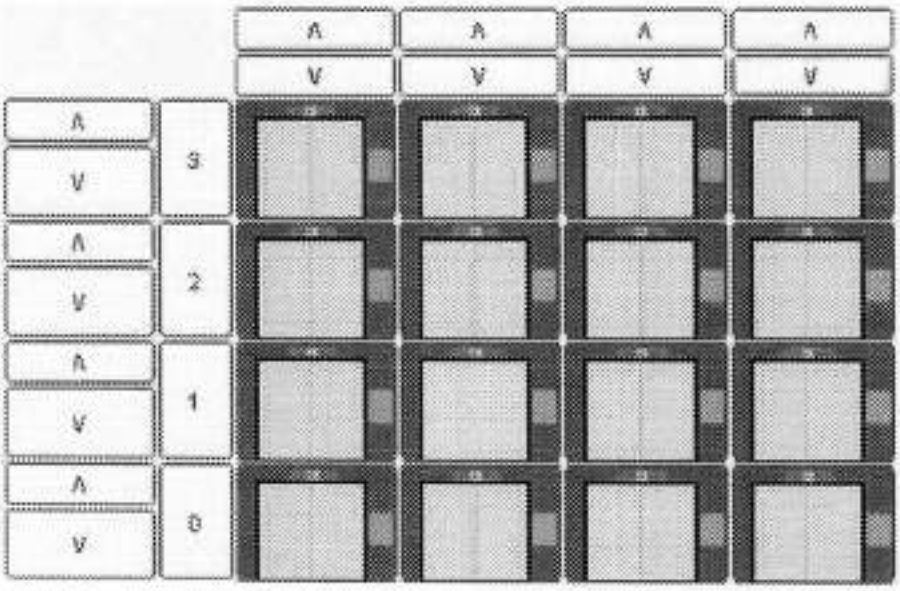

Figura 25 – Interface representante do prédio junto com os botões de interação.

REFERÊNCIAS

(1) SORSA, J., SIIKONEN, M. L., EHTAMO, H. Optimal control of double-deck elevator group using genetic algorithm. **International Transactions in Operational Research**, Vol. 10, N. 2, p. 103-114, 2003.

(2) KIM, C. B.; SEONG, K. A.; LEE-KWANG, H.; KIM, J. O. Design and Implementation of a Fuzzy Elevator Group Control System. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and humans**, Vol. 28, N. 3, p. 277-287, mai. 1998.

(3) AOKI, H., SASAKI, K. Group Supervisory Control System Assisted by Artificial Intelligence. **Elevator World**, Vol. 2, p. 70-80, fev, 1990.

(4) BARTO, A. G., CRITES, R. H. Elevator Group Control Using Multiple Reinforcement Learning Agents. **Machine Learning**, Vol. 33 N. 2-3 p 235-262, jan. 1998.

(5) ALANI, A. F., STONHAM, J., MEHTA, P., PROWSE, R. Performance Optimization of Knowledge-Based Elevator Group Supervisory Control System. **Elevator Technology**, Vol. 6, p. 114-121, 1998.

(6) AL-SHARIF, L. Bunching in Lifts . . . Why Does Bunching In Lifts Increase Waiting Time?. **Elevator World**, Vol. 44, N. 11, p75-77 nov. 1996.

(7) CHAN, W.L., SO, A. T. P., LAM, K.C. Dynamic Zoning in Elevator Traffic Control. **Elevator World**, p. 136-139, mar. 1997.

(8) SO, A. T. P., YU, J. K. L., CHAN, W. L. Dynamic Zoning Based Supervisory Control for Elevators. **Proc. of the 1999 IEEE - International Conference on Control Applications**, Vol. 2, p. 1591 – 159, ago 1999.

(9) SO, A. T. P., CHAN, W. L. Comprehensive Dynamic Zoning Algorithms. **Elevator World**, Vol. 45, N. 8, p. 99-109 set. 1997.

(10) HIKITA, S. New elevator group-control method for up-peak periods. **International Journal of Systems Science**, Vol. 34, N.5, p. 309-317, abr. 2003.

(11) SIIKONEN, M. L. **Planning and Control Models for Elevators in High-Rise Buildings**. 1997. PhD thesis, Helsinki Unverstity of Technology, 39p, 1997.

(12) CORTES, P., ONIEVA, J. L. L. A Genetic Algorithm for Controlling Elevator Group Systems. **Lecture Notes in Computer Science**, Vol. 2687, p.313-320, 2003.

(13) COSTA A. H. R. **Lógica Nebulosa**. Notas de aula do curso PCS2428 – Inteligência Artificial. São Paulo: PCS– EPUSP, 2004.

(14) FuzzyJ ToolKit for the Java(tm) Platform & FuzzyJess. National Research Council Canadá, 2005. Disponivel em: <http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html>. Acesso em: 15 nov. 2005.

(15) LAW, A. M., KELTON, W. D. **Simulation Modeling and Analysis**. 3 ed. McGraw-Hill Companies Inc., 2000, 760p.

(16) MARSAN, M. A. et al. **Modeling with Generalized Stochastic Petri Nets**. John Wiley & Sons, 1995, 301p.

(17) KIM, C. et al. A Fuzzy Approach to Elevator Group Control System. **IEEE Transactions on SMC**, v. 25, n. 6, p. 985-990, jun. 1995.

(18) WU, J. and WANG, Y. H. A Fuzzy Expert System for Elevator Group Control. **Proceedings of the Third International Workshop on Intelligent Control and Systems**, Atlantic City, v. 1, p. 946-949, fev. 2000.

(19) ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR NM-207**: Elevadores elétricos de passageiros: Requisitos de segurança para construção e instalação. Rio de Janeiro, 1999.

(20) NORMAS PARA PROJETO DE ESTABELCIMENTOS ASSISTENCIAIS DE SAÚDE. Resolução RDC ANVISA 50/2002, 144p.

(21) RUMBAUGH, J., JACOBSON, I., BOOCH, G. **The Unified Modeling Language Reference Manual**. Boston: Addison Wesley Professional., 2004, 752 p.

APÊNDICE A – Estudo da API FuzzyJ e comparação com Matlab

A API FuzzyJ é uma API java para representação e manipulação de informações nebulosas. Seu estudo se mostrou válido, pois o uso desta facilita o processo de integração com o simulador, que está sendo desenvolvido na mesma linguagem, Java. O Matlab, ao contrário, exigiria um maior esforço tanto para integração quanto para a realização de testes.

Esta API fornece todos os mecanismos necessários para se realizar o processo de definição de variáveis nebulosas e seus conceitos, para a definição das regras de inferência, para a *fuzzyficação* das variáveis de entrada, para o disparo do mecanismo de inferência e para a defuzzyficação das variáveis de saída. O método padrão de inferência é o de Mamdani e o método padrão de *defuzzyficação* é pelo centro de massa.

A seguir são apresentados os conceitos (classes) principais desta API:

1. ***FuzzyVariable***
    - Esta classe é utilizada para instanciar uma variável lingüística, incluindo o nome, a unidade de medida e o universo de discurso dessa variável. Pode-se ainda definir os conceitos nebulosos utilizados pela variável.
2. ***FuzzySet***
    - Esta classe é utilizada para a criação de funções de mapeamento para os conceitos definidos em uma variável lingüística. Mapeiam o valor de uma entrada para o seu correspondente valor verdade.

3. ***FuzzyRule***

   - Esta classe é utilizada para a definição das regras de inferência, sendo necessário especificar os antecedentes e os conseqüentes.

4. ***FuzzyValue***

   - Esta classe é utilizada para a definição dos antecedentes e dos conseqüentes de uma regra nebulosa. Também é utilizada para gerar as entradas correspondentes na hora do disparo das regras e para armazenar o valor das saídas.

Comparação da API FuzzyJ e do Matlab

Para se ter certeza de que os resultados obtidos por ambas as ferramentas não eram muito discrepantes, foi realizado um teste baseado no exemplo da referência (13). A seguir são mostradas as principais diferenças das duas ferramentas:

- Definição das variáveis lingüísticas:

  - **Java**: definição feita diretamente no código fonte, como exemplificado abaixo:

```
"velocidade = new FuzzyVariable("velocidade",0.0,120.0,"km/h");"
"velocidade.addTerm("baixa", new TrapezoidFuzzySet(0,0,30,50));"
"velocidade.addTerm("media",new TrapezoidFuzzySet(30,50,70,90)),"
 "velocidade.addTerm("alta", new TrapezoidFuzzySet(70,90,120,120));"
```

-**Matlab**: definição feita graficamente como na Figura 26:

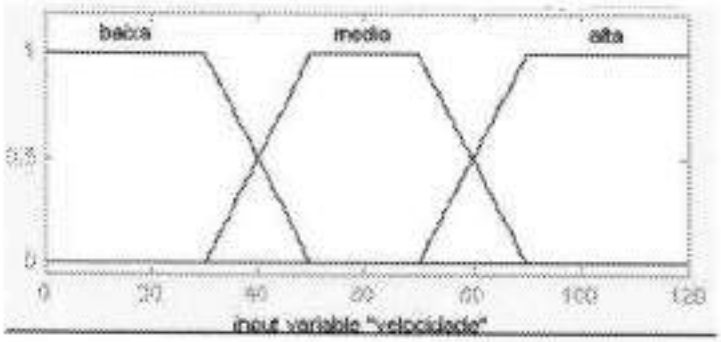

Figura 26 –Representação de uma variável no Matlab.

- Definição das regras de inferência:

-**Java**[1]: definição feita diretamente no código fonte, como exemplificado abaixo:

"velBaixaPneuVelho = new FuzzyRule();"
"velBaixaPneuVelho.addAntecedent(new FuzzyValue(velocidade,"baixa"));"
"velBaixaPneuVelho.addAntecedent(new FuzzyValue(pneu,"velho"));"
"velBaixaPneuVelho.addConclusion(new FuzzyValue(consumo,"alto"));"

-**Matlab:** definição feita graficamente como na Figura 27.

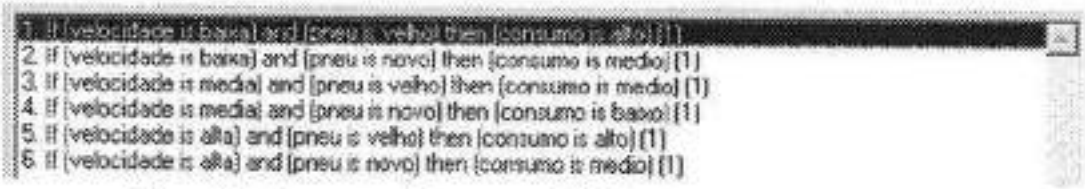

Figura 27 – Representação das regras de inferência no Matlab.

[1] Apenas uma das seis regras está representada.

- Disparo das regras e resultado da *defuzzyficação*:

  -**Java**[2]**:** feita diretamente no código fonte, como exemplificado abaixo:

  "velocidadeFv = new FuzzyValue(velocidade,new SingletonFuzzySet(35.0));"
  "pneuFv = new FuzzyValue(pneu,new SingletonFuzzySet(1.0));"
  "velBaixaPneuVelho.addInput(velocidadeFv);"
  "velBaixaPneuVelho.addInput(pneuFv);"
  "FuzzyValueVector fvv = velBaixaPneuVelho.execute();"
  "FuzzyValue globalOutput = fvv.fuzzyValueAt(0);"
  "double result = globalOutput.momentDefuzzify();"
  "consumo (result) é: 11.183962264150944"
  -**Matlab:** saída mostrada visualmente, como exemplificado na Figura 28.

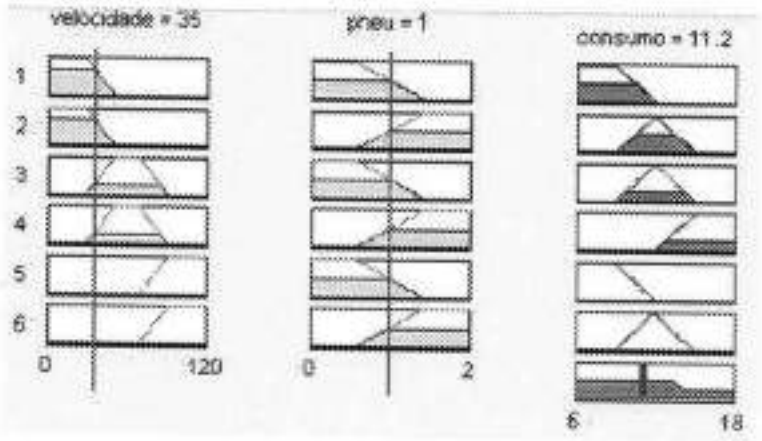

Figura 28 – Representação do disparo das regras e da defuzzyficação no Matlab.

Pode-se notar a correspondência entre as duas ferramentas e a facilidade de se trabalhar com o Matlab, principalmente por se tratar de uma ferramenta em que é possível visualizar todo o sistema nebuloso de modo gráfico. A API FuzzyJ pode não oferecer tais recursos gráficos mas tais recursos não são realmente necessários para a aplicação da lógica nebulosa, e a integração de ferramentas distintas é uma atividade que pode consumir bastante esforço fora do escopo do

[2] A lógica da união dos resultados de cada regra não está representada

projeto. Portanto, a utilização da FuzzyJ é a melhor opção para a implementação do sistema de lógica nebulosa neste caso.

APÊNDICE B – Tabelas e testes da Pseudo-Alocação com sentido

Neste apêndice são encontradas as tabelas de regras para a pseudo-alocação com sentido e alguns dos testes que foram realizados para a validação do método.

Para melhor entendimento da tabelas são necessárias algumas explicações da terminologia utilizada.

Na Tabela 14, "pos" significa a chamada em análise na fila de atendimento da cabine (ou seja, a chamada presente em uma dada posição na fila onde deseja testar-se a inserção da nova chamada, que deve ser alocada) e "chamada" corresponde à nova chamada a ser alocada.

Foi considerado que uma chamada engloba outra quando a primeira apresenta um sentido, para cima ou para baixo, que vai em direção da segunda. Algumas tabelas utilizam a denominação "anterior (= pos--)", que se refere ao alvo anterior ao alvo em análise na fila (ou seja, a posição da fila antecedente à posição com qual estou comparando a chamada a ser alocada no presente momento). A denominação "pos++ = lookahead (prox pos)", na Tabela 16, se refere ao alvo posterior ao alvo em análise na fila.

As Tabelas 15,16,17,18,19 e 20 são tabelas auxiliares na determinação do resultado do posicionamento de uma chamada na fila de atendimento

Regra id		Andar do Alvo Pos	Andar do último alvo analisado	Pos engloba chamada	Pos é interna	resultado
2 e 3	Alvo Chamada	<	>	X	X	*tab3
2 e 3	Alvo Chamada	<	>	X	X	*tab3
4	Alvo Chamada	<	<	sim	não	depois
5	Alvo Chamada	<	<	não	não	*tab4
5	Alvo Chamada	<	<	não	não	*tab4
5	Alvo Chamada	<	<	não	não	*tab4
5	Alvo Chamada	<	<	não	não	*tab4
6	Alvo Chamada	<	<	X	sim	depois
6	Alvo Chamada	<	<	X	sim	depois
7	Alvo Chamada	<	=	X	x	*tab3
8	Alvo Chamada	=	>	X	não	*tab1
9	Alvo Chamada	=	<	X	não	*tab1
10	Alvo Chamada	=	<	X	sim	*tab2
11	Alvo Chamada	=	>	X	sim	*tab2
13	Alvo Chamada	>	>	sim	não	depois
14	Alvo Chamada	>	>	não	não	*tab5
14	Alvo Chamada	>	>	não	não	*tab5
15	Alvo Chamada	>	>	X	sim	depois
16 e 17	Alvo Chamada	>	<	X	não	*tab6
16 e 17	Alvo Chamada	>	<	X	não	*tab6
16 e 17	Alvo Chamada	>	<	X	não	*tab6
18	Alvo Chamada	>	=	X	x	*tab6

Tabela 14 – Condições e regras de pseudo-alocação.

	sentido pos	resultado
sentido da chamada	=	ignora chamada
sentido da chamada	!=	depois

Tabela 15 – Correspondente ao tab 1 da Tabela 14.

pos++ = lookahead (prox pos)	
sentido da chamada engloba pos++	resultado
sim	ignora
não	depois

Tabela 16 - Correspondente ao tab 2 da Tabela 14.

se chamada é interna		chamada	resultado
antes(ou ignora)		↓	depois
		↑	antes ou ignora

Tabela 17 - Correspondente ao tab 3 da Tabela 14.

Se anterior não vazio -> sentido anterior (=pos --)	resultado
↑	depois
↓	antes

anterior (= pos --) é interna ou é vazia	chamada engloba pos ou é interna	resultado
sim	sim	antes
sim	não	depois

Tabela 18 - Correspondente ao tab 4 da Tabela 14.

Se anterior não vazio -> sentido anterior (=pos --)	resultado
↓	depois
↑	antes

anterior (= pos --) é interna ou é vazio	chamada engloba pos ou é interna	resultado
sim	sim	antes
sim	não	depois

Tabela 19 - Correspondente ao tab 5 da Tabela 14.

se chamada é interna		chamada	resultado
antes(ou ignora)		↓	antes
		↑	depois

Tabela 20 - Correspondente ao tab 6 da Tabela 14.

Para cada regra, foi gerada uma entrada para o pré-controle, para garantir que a saída esperada era, de fato, obtida. Nesta entrada, estavam contidas as informações das colunas "Alvo e sentido da Chamada", "Andar Atual da cabine" e "Fila Atual". A fila toda é percorrida para a realização da pré-alocação até que haja a definição da posição onde a nova chamada deve ser alocada, como explicado no item 4.4.3.

Na Tabela 21 pode-se ver o roteiro de alguns dos testes realizados para o pré-controle com sentido:

Regra Testada	Alvo e sentido da Chamada	Alvo em análise na fila	Andar Atual da cabine	Andar do último alvo analisado	Fila Atual	Fila Depois	Fila esperada
2 e 3	4↓	5↑	0	2	1↑, 2↑, 5↑	1↑, 2↑, 5↑, 4↓	1↑, 2↑, 5↑, 4↓
2 e 3	6↑	7	2	4	3↑,4, 7,2↓	3↑,4, 6↑,7,2↓	3↑,4, 6↑,7,2↓
5	2↓	4↑	0	3	3↑, 4↑, 6↑	3↑, 4↑, 6↑, 2↓	3↑, 4↑, 6↑, 2↓
11	5↓	5	0	2	1↑, 2↑, 5, 7↑	1↑, 2↑, 5, 7↑, 5↓	1↑, 2↑, 5, 7↑, 5↓
18	2↓	1↑	0	2	2, 1↑	2, 2↓, 1↑	2, 2↓, 1↑

Tabela 21 – Roteiro de testes para o pré-controle com sentido.

É válido explicar alguns dos testes para que se entenda o processo utilizado. Para o caso indicado na primeira linha da Tabela 21, vê-se que a nova chamada ocorreu no andar 4, abaixo do alvo em análise na fila que indica o andar 5, e acima do andar correspondente ao último alvo analisado, que é o 2. Na configuração indicada, as regras que devem ser observadas na Tabela 14 são a 2 e 3. Ao se olhar para a linha correspondente, percebe-se que o resultado do posicionamento da nova chamada será dado pela Tabela 17 e, como a chamada possui sentido para baixo, o resultado obtido é "depois", ou seja, a nova chamada deve ser colocada depois do alvo em análise na fila. Como o alvo em análise na fila era o último elemento da fila, a posição da

nova chamada é definida, resultando na configuração apresentada na coluna "Fila Depois" da Tabela 21.

Analogamente à primeira linha da Tabela 21, a segunda apresenta as mesmas características, com exceção do sentido da nova chamada, o que faz com que o resultado obtido pela Tabela 17 seja "antes", ou seja, a nova chamada deve ser posicionada antes do alvo em análise na fila. Conforme explicado no item 4.4.3, quando ocorre este resultado, o posicionamento da chamada é definido e a iteração na fila acaba, o que resulta na configuração apresentada na coluna "Fila Depois" da Tabela 21.

Pode-se perceber que diferentes configurações resultam no disparo de diferentes regras para se gerar a pseudo-alocação.

APÊNDICE C – Descrição do CD

O cd anexo ao trabalho contém 5 diretórios. Abaixo segue a descrição do conteúdo de cada um:

- **Código-fonte:** Contém todos os códigos-fonte do projeto, e os arquivos de configuração.
- **API_S:** Contém as API's utilizadas, de *parsing* de xml e de lógica nebulosa, para esta última, a documentação também está presente.
- **Executável:** Contém o arquivo "sistema.jar " que pode ser executado pelo arquivo "executaSistema.bat", iniciando o simulador de edifícios.
- **Tabelas:** Contém as tabelas utilizadas no projeto, para a elaboração das regras e do pré-controle.
- **Documentação Final:** Contém uma cópia da documentação final.