



SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM
COMPUTACIONAL E TECNOLOGIA INDUSTRIAL
Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de mestrado

**Um Modelo Baseado em Agentes para Simulação de
Indivíduos em Residências Unipessoais**

Apresentada por: Fábio Britto de Carvalho Almeida
Orientador: Dr. Valter de Senna

Dezembro de 2013

Fábio Britto de Carvalho Almeida

Um Modelo Baseado em Agentes para Simulação de Indivíduos em Residências Unipessoais

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Dr. Valter de Senna
SENAI CIMATEC

Salvador
SENAI CIMATEC
2013

Nota sobre o estilo do PPGMCTI

Esta dissertação de mestrado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

SENAI CIMATEC

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Mestrado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação [com distinção] da Dissertação de mestrado, intitulada “ Um Modelo Baseado em Agentes para Simulação de Indivíduos em Residências Unipessoais ”, apresentada no dia (dia) de (mês) de (ano), como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

Prof. Dr. Valter de Senna
SENAI CIMATEC

Membro interno da Banca:

Prof. Dr. Hernane Borges de Barros Pereira
SENAI CIMATEC

Membro externo da Banca:

Prof. Dr. José Garcia Vivas Miranda
UFBA

Resumo

Em virtude do envelhecimento populacional, do aumento de domicílios unipessoais e da grande quantidade de acidentes nesses domicílios, há uma clara necessidade de um acompanhamento mais próximo desses moradores, o que representa um grande desafio para as famílias, pois nem sempre é possível de ser realizado. Trata-se da oferta de atenção e cuidados especiais, que vão além daqueles normalmente oferecidos pela Previdência, pela Assistência Social ou pelas políticas no campo da Saúde Pública. A redução do tamanho da família, a entrada da mulher no mercado de trabalho, alterando sua função dentro da família e o surgimento de novos arranjos familiares, decorrentes de novas formas de união conjugal, tendem a comprometer as condições de cuidado e atendimento diretos às pessoas idosas na família. Diante desse problema, o objetivo geral dessa pesquisa é o desenvolvimento de uma simulação que seja capaz de gerar informações sobre o comportamento diário de pessoas que residem sozinhas. Com essas informações, será possível a aplicação e análise de algoritmos para reconhecimento de padrões e, com isso, perceber antecipadamente anormalidades na rotina desses indivíduos. Partindo-se da necessidade de construção de uma ferramenta capaz de gerar informações referentes ao dia a dia das pessoas em suas residências, primeiramente foi realizado um estudo para escolha da tecnologia, após isso, os requisitos para construção do modelo foram especificados e, em seguida, o modelo foi desenvolvido e implementado. Através dos resultados obtidos a partir do trabalho experimental realizado nesse estudo, pôde-se provar que o modelo proposto foi capaz de simular o comportamento diário dos usuários entrevistados.

Abstract

In view of the aging population, the increase of single person households and the large number of accidents in these households, there is an evident demand for mechanisms for monitoring these residents, which represents a major challenge for families, since it is not always possible to be performed in traditional ways. It aims to allow an special attention and care that go beyond those normally offered by Social Security, the Social Assistance or policies in the field of Public Health. The reduction of family size, caused especially by the entrance of women into the labor market, changed the weight of their roles in their families and the emergence of new family arrangements, arising from new forms of marital union, tends to undermine the conditions of care and direct care for the elderly in the families. Faced with this problem , the overall goal of this research is to develop a simulation that is capable of generating information about the daily behavior of people living alone . With this information , you can the application and analysis of algorithms for pattern recognition and , therefore , advance notice abnormalities in the routine of these individuals . Starting from the need to build a tool that can provide information regarding the daily lives of people in their homes, primarily a study to technology choice was made, and after that , the requirements for constructing the model were specified and then the model was developed and implemented . The results obtained from the experimental work carried out in this study , we could demonstrate that the proposed model was able to simulate the daily behavior of users interviewed.

Sumário

1	Introdução	1
1.1	Definição do problema	3
1.2	Objetivo	4
1.3	Limites e limitações	5
1.4	Aspectos metodológicos	5
1.5	Organização da Dissertação de mestrado	6
2	Agentes	7
2.1	Histórico	7
2.2	Conceitos	8
2.3	Propriedades	9
2.4	Características	10
2.5	Sistemas Multiagente	11
2.6	Simulação	14
3	O Modelo	17
3.1	Introdução	17
3.2	Análise de Requisitos	17
	3.2.1 Requisitos funcionais	18
	3.2.2 Requisitos não funcionais	19
3.3	Diagrama de Casos de Uso	19
3.4	Diagrama de Componentes	21
3.5	Modelo Conceitual	21
3.6	Protocolo ODD	23
	3.6.1 Propósito	24
	3.6.2 Variáveis de estado e escalas	24
	3.6.3 Visão de processo e escalonamento	27
	3.6.3.1 Agente Humano	27
	3.6.3.2 Agentes Sensores	28
	3.6.4 Conceitos de Projeto	29
	3.6.5 Inicialização	29
	3.6.6 Entrada	30
	3.6.7 Submodelos	30
4	Implementação do Modelo	33
4.1	O Questionário	33
4.2	A Simulação Anjo da Guarda	34
4.3	Resultados	42
5	Considerações finais	53
5.1	Conclusões	53
5.2	Contribuições	54
5.3	Atividades Futuras de Pesquisa	54
A	Questionário	55

B NetLogo	61
B.1 Histórico	61
B.2 O Ambiente Netlogo	62
B.3 A Linguagem NetLogo	64
B.4 Biblioteca de Modelos	64
B.5 Extensões	65
C Respostas do Questionário	66
Referências	67

Lista de Tabelas

2.1	Comparação entre Ferramentas de Simulação Baseadas em Sistemas Multiagente. Fonte: Adaptada de SAKURADA (2009)	16
3.1	Parâmetros, descrições e tipos utilizados no modelo Anjo da Guarda - Fonte: O autor	26
4.1	Matriz de Adjacência Fonte: O Autor.	40
4.2	Perguntas e Respostas do Questionário Fonte: O Autor.	43
C.1	Respostas do Questionário Aplicado. Fonte: O autor	66

Lista de Figuras

2.1	Interação entre agente e ambiente. Fonte: Russell e Norvig (2009).	9
3.1	Diagrama de Casos de Uso. Fonte: O Autor.	20
3.2	Diagrama de Componentes. Fonte: O Autor.	21
3.3	Arquitetura Conceitual do Modelo Proposto. Fonte: O Autor.	23
3.4	Os sete elementos do protocolo ODD. Fonte: Grimm et al. (2006).	24
3.5	Fluxograma do processo Agente Humano. Fonte: O autor.	28
3.6	Acionamento do agente sensor pelo agente humano. Fonte: O autor.	29
4.1	Software Questionário . Fonte: O Autor.	34
4.2	Fluxograma de utilização da Simulação Anjo da Guarda . Fonte: O Autor.	35
4.3	Script de Criação da Base de Dados . Fonte: O Autor.	35
4.4	Código de Conexão com o Banco de Dados . Fonte: O Autor.	36
4.5	Tela inicial da Simulação Anjo da Guarda . Fonte: O Autor.	37
4.6	Carregar Modelos. Fonte: O Autor.	38
4.7	Posicionamento dos Sensores. Fonte: O Autor.	38
4.8	Posicionamento das Portas. Fonte: O Autor.	39
4.9	(a) Planta com Portas Posicionadas (b) Grafo Representativo das Portas Fonte: O Autor.	39
4.10	Componentes de Inserção da Matriz de Adjacência. Fonte: O Autor.	40
4.11	Adicionar Móveis no Imóvel. Fonte: O Autor.	41
4.12	Apartamento com sensores posicionados. Fonte: O Autor.	44
4.13	Sensor Cama Idoso A DC. Fonte: O Autor.	46
4.14	Sensor Cama Idoso B DJ. Fonte: O Autor.	46
4.15	Sensor Cama Idoso C DV. Fonte: O Autor.	47
4.16	Sensor Mesa Idoso A DC. Fonte: O Autor.	47
4.17	Sensor Mesa Idoso B DJ. Fonte: O Autor.	48
4.18	Sensor Mesa Idoso C DV. Fonte: O Autor.	48
4.19	Sensor Sofá Idoso A DC. Fonte: O Autor.	49
4.20	Sensor Sofá Idoso B DJ. Fonte: O Autor.	49
4.21	Sensor Sofá Idoso C DV. Fonte: O Autor.	49
4.22	Sensor Chuveiro Idoso A DC. Fonte: O Autor.	50
4.23	Sensor Chuveiro Idoso B DJ. Fonte: O Autor.	50
4.24	Sensor Chuveiro Idoso C DV. Fonte: O Autor.	51
4.25	Sensor Porta Idoso A DC. Fonte: O Autor.	51
4.26	Sensor Porta Idoso B DJ. Fonte: O Autor.	52
4.27	Sensor Porta Idoso C DV. Fonte: O Autor.	52
B.1	Centro de Comandos do NetLogo. Fonte: O autor.	62
B.2	Interface de Controle do NetLogo. Fonte: O autor.	63

Lista de Algoritmos

Lista de Siglas

PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
IA	Inteligência Artificial
AI	Agentes Inteligentes
SMA	Sistemas Multiagentes
ONU	Organização das Nações Unidas
IBGE	Instituto Brasileiro de Geografia e Estatística
CCL	Center for Connected Learning
SMS	Short Message Service
SQL	Structured Query Language
JDBC	Java DataBase Connectivity
SGBD	Sistema de Gerenciamento de Banco de Dados
DML	Data Manipulation Language
DDL	Data Definition Language
ODD	Overview, Design Concepts and Details

Introdução

O envelhecimento é um fenômeno inevitável e progressivo, caracterizado por alterações morfológicas, funcionais, bioquímicas e psicológicas no indivíduo. Essas modificações trazem a perda ou diminuição da capacidade de adaptação ao meio ambiente, redução de equilíbrio, da acuidade visual e auditiva, entre outros, causando maior fragilidade, necessidade de assistência e maior incidência de processos patológicos, que podem levar o ser humano a incapacidade funcional, depressão, dependência e à morte.

No Brasil e no mundo, tem-se observado uma importante transição demográfica, caracterizada pelo crescente aumento da população idosa (representada por pessoas com idade igual ou superior a 60 anos) e redução da parcela de jovens, consequência da redução nas taxas de natalidade e mortalidade, associada a fenômenos como a universalização da educação, a intensificação da participação feminina no mercado de trabalho e a difusão do planejamento familiar. Segundo estudiosos, o fator de maior relevância na composição deste quadro foi a queda da fecundidade vivenciada pela população da maioria dos países. (BATISTA, 2008)

Dados obtidos na Previdência Social mostram o envelhecimento populacional por regiões do mundo, onde em 1950, pessoas com 60 anos ou mais na população correspondia na Europa a 12,1%, na América do Norte 12,4% e na América Latina e Caribe 5,9%. Já em 2000, o percentual de pessoas com essa idade cresceu para 20,3%, 16,2% e 8,0% na Europa, América do Norte e América Latina / Caribe, respectivamente. Acredita-se que em 2050 esses percentuais cheguem a 36,6% na Europa, 27,2% na América do Norte e 22,5% na América Latina e Caribe (BATISTA, 2008).

Em relação às pessoas com 80 anos ou mais, em 1950, na Europa e América do Norte, essa parcela correspondia a 9,1% da população total e na América Latina e Caribe, 6,5%. Em 2000, o percentual dessas pessoas cresceu para 14,6%, 19,8% e 11,1%, na Europa, América do Norte e América Latina / Caribe, respectivamente. Acredita-se que em 2050, haja também um maior crescimento, chegando até a 27,1% na Europa, 28,3% na América do Norte e 18,1% na América Latina e Caribe (BATISTA, 2008).

As grandes transformações que acarretaram em instabilidade do padrão demográfico brasileiro começaram a partir dos anos 40, com um consistente declínio dos níveis gerais de mortalidade, associado ao declínio da natalidade, intensificado nos anos 1980, segundo dados do IBGE. Esse fato provoca uma inversão na pirâmide etária populacional e tem como uma de suas principais consequências o aumento da proporção de pessoas idosas

na população. Desta forma, idosos e idosas se tornaram visíveis e suas necessidades e problemas passaram a pesar em toda a sociedade (CARVALHO, 2010).

No que diz respeito à família, esta vem passando por uma série de transformações ao longo do último século, seguindo e influenciando os acontecimentos históricos, econômicos, sociais e demográficos, que geram alteração em sua composição. Pode-se destacar como os principais acontecimentos que geram mudanças do padrão familiar, antes caracterizado por ser grande e com elevado número de filhos, a idade do casamento, mortalidade adulta, independência financeira feminina, o risco de divórcio e recasamento, o crescimento das famílias sem filhos, a expansão dos interesses individuais e a tendência e capacidade das pessoas, inclusive das idosas, de viverem sozinhas (CARVALHO, 2010).

Existe atualmente, diante da nova composição da estrutura familiar, da ciência, da tecnologia e ainda das facilidades e modernidades que cercam o indivíduo, uma grande tendência, seja entre indivíduos jovens ou idosos, de viverem de forma mais independente. Espera-se assim, que ao longo dos anos, haja um aumento gradativo do número de domicílios unipessoais, que são aqueles onde o indivíduo mora sozinho em sua residência. Diante das condições de saúde da população idosa e do aumento da expectativa de vida, espera-se também que esta tendência influencie a população idosa brasileira e mundial.(CAMARGOS; RODRIGUES; MACHADO, 2011)

A Organização das Nações Unidas (ONU) realizou pesquisa analisando os arranjos domiciliares dos idosos e apresentaram uma comparação dos dados obtidos em 130 países. Essa pesquisa teve como principais conclusões: aproximadamente um a cada sete idosos moram sozinhos; existe uma tendência a favor de vida independente (sozinho ou somente com o cônjuge), essa realidade é mais comumente encontrada em países desenvolvidos; há mais homens idosos casados (80%) do que mulheres idosas (45%) (CAMARGOS; RODRIGUES; MACHADO, 2010).

Segundo informações do IBGE, entre 1991 e 2000 houve um aumento de 1,9% da quantidade de idosos que vivem sozinhos no Brasil. Das regiões brasileiras, a que possui menor índice de idosos morando sozinhos, é a região Norte, enquanto que as regiões Sul, Sudeste e Centro-Oeste, no ano 2000, apresentaram praticamente a mesma porcentagem (12,8%). Ainda segundo o IBGE, a quantidade de idosos residindo de forma solitária vem crescendo, chegando em 2006 a 13,2%. Nesse mesmo ano, a população de 60 anos ou mais, foi responsável por 40,3% dos domicílios unipessoais brasileiros (CAMARGOS; RODRIGUES; MACHADO, 2011).

Pode-se destacar entre os motivos que levam o idoso a decisão de morar sozinho o aumento da esperança de vida, a morte do cônjuge, aspectos culturais, como a adoção do individualismo ao invés de assumir as obrigações familiares, inexistência de parentesco,

aumento da renda real, permitindo arcar com os custos envolvidos, que favorecem uma diminuição da dependência dos filhos e permitem uma vida mais independente e com mais privacidade (CAMARGOS; RODRIGUES; MACHADO, 2011).

Devido às perdas progressivas, sejam elas motoras ou cognitivas, que decorrem do envelhecimento do corpo humano, os idosos que moram sozinhos estão mais susceptíveis a sofrerem acidentes. As quedas são um grande problema para esses idosos, e estão proporcionalmente associadas à necessidade de hospitalização, cirurgia e incapacidades futuras. A estimativa da incidência de quedas por faixa etária é de 28% a 35% nos idosos com mais de 65 anos e 32% a 42% naqueles com mais de 75 anos.

Em um estudo realizado em 2002, cerca de 31% dos idosos disseram ter caído no ano anterior à pesquisa e 11% afirmaram ter sofrido duas ou mais quedas. Alguns estudos prospectivos indicam que 30% a 60% da população da comunidade com mais de 65 anos cai anualmente e metade apresenta quedas múltiplas. Aproximadamente 40% a 60% destes episódios levam a algum tipo de trauma, sendo 30% a 50% de menor gravidade, 5% a 6% lesões mais graves (não incluindo fraturas) e 5% de fraturas. Destas, as mais comuns são as vertebrais, em fêmur, úmero, rádio distal e costelas. Cerca de 1% das quedas leva à fratura do fêmur, que está fortemente ligada a prejuízos funcionais severos e muitas vezes irreversíveis (BUKSMAN, 2008).

Sendo assim, é necessária maior atenção às pessoas que por necessidade ou escolha residem sozinhas. Ações que objetivam apoiar essas pessoas são extremamente importantes tendo em vista que muitos não recebem nenhum tipo de ajuda. Diante deste problema, surge o projeto Anjo da Guarda, que é um projeto de doutorado do Programa de Pós-Graduação Stricto Sensu em Modelagem Computacional e Tecnologia Industrial (PPGMCTI) do SENAI CIMATEC e que tem o propósito de buscar uma solução computacional que aprende o comportamento diário de pessoas, principalmente idosos, que residem sozinhas e percebe antecipadamente comportamentos que são considerados fora dos padrões habituais. Assim, ele tem a função de contatar pessoas, informando-as de potenciais anormalidade, acelerando no processo de socorro e evitando que eventuais problemas se agravem. Com isso, essa pesquisa subsidia parcialmente o projeto de doutorado supracitado criando um modelo capaz de simular indivíduos que residem sós, realizando atividades domésticas e de gerar dados que representem o cotidiano desses indivíduos. Esses dados serão fundamentais para escolha de um algoritmo eficaz para reconhecimento de padrão.

1.1 Definição do problema

Em virtude do envelhecimento populacional, do aumento de domicílios unipessoais e da grande quantidade de acidentes nesses domicílios, há uma notória necessidade de um

acompanhamento mais próximo desses moradores, o que representa um grande desafio para as famílias, pois nem sempre é possível de ser realizado. Trata-se da oferta de atenção e cuidados especiais, que vão além daqueles normalmente oferecidos pela Previdência, pela Assistência Social ou pelas políticas no campo da Saúde Pública. A redução do tamanho da família, a entrada da mulher no mercado de trabalho, alterando o peso da sua função dentro da família e o surgimento de novos arranjos familiares, decorrentes de novas formas de união conjugal, tende a comprometer as condições de cuidado e atendimento diretos à pessoa idosa na família. (CAMARGOS; RODRIGUES; MACHADO, 2011) Diante da carência de dados sobre o comportamento cotidiano de indivíduos em suas residências, surge o seguinte problema de pesquisa: é possível desenvolver uma simulação computacional capaz de gerar padrões de comportamento de indivíduos que residem sozinhos?

1.2 Objetivo

Diante do problema exposto, o objetivo geral dessa pesquisa é desenvolver um modelo computacional que seja capaz de representar, a partir da aplicação de um questionário, indivíduos em residências unipessoais realizando atividades domésticas rotineiras e de gerar informações sobre o comportamento diário desses indivíduos, para que assim consiga-se aplicar algoritmos de reconhecimento de padrões e com isso, tornar possível a percepção antecipada de anormalidades na rotina desses indivíduos.

Os objetivos específicos estão definidos a seguir:

1. Estudar simulações e simuladores baseados em agentes;
2. Desenvolver um modelo que seja capaz de simular um indivíduo sozinho em seu imóvel;
3. Implementar um questionário capaz de identificar as atividades domésticas e as faixas de horários de realização de cada uma dessas atividades. Essas informações servirão como base de entrada para a montagem do comportamento do ser humano na simulação;
4. Implementar uma ferramenta que possibilite a criação e/ou carregamento de diferentes plantas de imóveis, seleção de móveis para adicioná-los nas plantas geradas, colocação de sensores ao longo do imóvel, que servirão para coletar informações do morador, e movimentação do morador;
5. Validar o modelo criado realizando entrevistas com pessoas que se enquadrem no perfil da pesquisa e confrontar os dados de entrada com os resultados obtidos pela ferramenta.

1.3 Limites e limitações

A ferramenta NETLOGO (TISUE; WILENSKY, 2004), escolhida para o desenvolvimento da simulação, possui algumas limitações no que diz respeito à estrutura de dados. Dessa forma, os algoritmos desenvolvidos tiveram que passar por grandes modificações para que estivessem de acordo com os recursos oferecidos pela plataforma.

O modelo desenvolvido nessa pesquisa de mestrado não visou contemplar na simulação todas as possíveis atividades rotineiras realizadas pelos seres humanos em suas residências, mas sim as mais relevantes, identificadas através de pesquisas, que fossem capazes de representar os padrões de comportamento de cada morador.

1.4 Aspectos metodológicos

Partindo-se da necessidade de construção de uma ferramenta que fosse capaz de gerar informações referentes ao dia a dia das pessoas em suas residências, foi necessária a realização de um estudo aprofundado de simulações baseadas em agentes. A partir desse estudo foi possível notar a existência de uma grande quantidade de ferramentas que auxiliam no desenvolvimento dessas simulações. Por ser uma ferramenta gratuita, com documentação acessível e por ser amplamente utilizada no meio acadêmico a ferramenta escolhida para implementação do modelo foi a NetLogo.

Após definida ferramenta, os requisitos foram analisados e especificados para o desenvolvimento do modelo. Já o processo de modelagem e desenvolvimento do sistema foi realizado seguindo as etapas descritas a seguir:

1. Brain Storm (Chuva de Ideias): Nessa etapa foram descritas em um papel todas as ideias para a elaboração de um modelo que fosse capaz de gerar plantas dinâmicas de imóveis, mobiliá-lo, posicionar sensores e simular o cotidiano de uma determinada pessoa dentro do domicílio criado;
2. Construção do Modelo: Foi feita uma seleção das melhores ideias geradas no item anterior e, a partir delas, definido quais seriam os agentes, como seria o ambiente e quais as regras de relacionamento entre ambos, ou seja, foi elaborado um modelo conceitual. Após isso, esse modelo foi detalhado utilizando o protocolo ODD;
3. Codificação e Testes: O modelo foi implementado de forma iterativa e incremental, onde no final de cada interação, foi realizado uma série de testes;
4. Validação: Foram realizadas entrevistas com três pessoas que moram sozinhas, foi dado carga na ferramenta com o questionário dessas entrevistas e os resultados

gerados foram confrontados com os dados iniciais.

1.5 Organização da Dissertação de mestrado

Este documento apresenta 5 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida.;
- **Capítulo 2 - Agentes :** Nesse capítulo serão apresentados os principais conceitos sobre agentes inteligentes, sistemas multiagentes e simulação computacional;
- **Capítulo 3 - Modelo Proposto:** É apresentado o modelo computacional para simulação de indivíduos que moram sozinhos juntamente com as etapas da sua elaboração;
- **Capítulo 4 - Implementação do Modelo:** As ferramentas geradas pela implementação do modelo são descritas e os resultados obtidos na pesquisa apresentados;
- **Capítulo 5 - Considerações Finais:** Considerações sobre o que foi produzido e melhorias futuras;

Agentes

Os conceitos de agentes (A) e de inteligência artificial (IA) estão associados. Isso pode ser comprovado através de Wooldridge e Jennings (1995) que definem IA como um campo de estudos das ciências da computação que visa à concepção de agentes que apresentam aspectos do comportamento inteligente.

Um dos grande objetivos em IA é a recriação de aspectos do comportamento humano. Em virtude disso, é provável que grandes esforços aplicados a pesquisas com esse propósito, tenham sido as principais responsáveis para o surgimento da tecnologia baseada em agentes.

Projetar agentes está relacionado com conceitos e técnicas presentes na IA e, embora exista uma grande variedade de contribuições acerca do tema na literatura acadêmica, Wooldridge e Jennings (1995) afirmam que não existe consenso absoluto sobre esta questão. Alguns pesquisadores consideram que AI pode ser um software autônomo capaz de realizar tarefas, com habilidade para interagir com o ambiente. Um agente pode ser um software planejado ou implementado através de conceitos característicos dos seres humanos.

A tecnologia baseada em sistemas multiagentes pode ser estudada sob o ponto de vista de apenas um agente ou sob o ponto de vista de diversos agentes em um mesmo ambiente, os sistemas multiagentes. Sendo assim, este capítulo tem como objetivo mostrar os principais conceitos que envolvem agentes e sistemas multiagentes, pois serviram como base para o desenvolvimento do modelo deste trabalho de mestrado.

2.1 *Histórico*

Desde os anos 70, há uma crescente necessidade da utilização de softwares com o objetivo de realizar atividades para ajudar os usuários. Esta necessidade incentivou a pesquisa nesse assunto e trouxe para a área da computação diversos conceitos essenciais à inteligência humana como raciocínio, aprendizagem e representação do conhecimento. (COSER, 1999)

Durante a década de 80, a comunidade de inteligência artificial começou a estudar novas áreas que tornassem possível a dinamização dos domínios de sistemas de IA. Isto é, os resultados simulados foram esquecidos e passou-se a explorar experimentações de possíveis

interações complexas com o mundo físico através de um novo conceito chamado de agentes. Segundo CHORAFAS (1998), “ *a principal diferença entre os agentes e as ferramentas atualmente disponíveis é que o agente observa e se comunica com o usuário, enquanto que uma ferramenta tradicional, como um objeto passivo, é algo que o usuário enxerga e manipula* ” .

Somente nos anos 90, a ideia de construção, através da emoção, de sistemas inteligentes capazes de interagir com o “mundo real” ganhou maior expressão. Isso foi possível após a publicação de dois trabalhos: as investigações sobre os mecanismos do medo realizadas por LeDoux (1996) e, em especial, as investigações sobre a relação entre emoção e racionalidade, realizadas por Damásio (1994). (MORGADO, 2005)

Entre os fatores que estimulam e influenciam na busca dos desenvolvedores a trabalhar com agentes, segundo COSER (1999) , podem-se destacar:

- Os agentes introduzem inteligência nas interfaces, possibilitando que os usuários tirem máximo proveito das aplicações por mais inexperientes que sejam;
- Podem ser utilizados para personalizar aplicações e serviços que vão de acordo com as preferências, metas e desejos dos usuários e ainda gerenciar a recuperação, disseminação e a pesquisa de informações através das redes de computadores, especialmente à internet;
- São aplicáveis de várias maneiras em sistemas de comércio eletrônico e auxiliam no funcionamento mais eficiente de organizações.

2.2 Conceitos

É interessante que antes da apresentação do conceito de agentes, seja realizada uma definição da palavra *agente*. Segundo o dicionário online MICHAELIS (WEISZFLOG, 2013), agente é algo que age, que exerce alguma ação. Correlacionando essa definição com o contexto da ciência da computação, pode-se dizer que um agente é um software que age, ou que exerce alguma ação que lhe foi encarregada por um usuário ou por outro software.

Não há na literatura consenso sobre o conceito de agentes, permitindo assim que cada pesquisador defina agentes de uma forma pessoal. Para o presente estudo, seguem definições de agentes utilizadas nessa pesquisa para concepção do modelo.

Segundo Wooldridge (2008), “ *um agente é um sistema de computador que está situado em algum ambiente e que é capaz de realizar ações autônomas neste ambiente, a fim*

de cumprir os seus objetivos ” . Complementando essa definição, SIGNORETTI (2012), afirma que “ *agentes são entidades reais ou virtuais, que possuem objetivos e, para atingi-los, devem ser capazes de comunicar-se com outros agentes e de interagir com o ambiente no qual estão inseridos* ” .

Para Russell e Norvig (2009), o que contribui para verificar se um sistema poderá ser visto como agente é o fato dele ser capaz de perceber o seu ambiente através de sensores e agir sobre eles por meio de atuadores, como pode-se observar ilustrado na Figura 2.1, que mostra a interação entre os agentes e o ambiente.

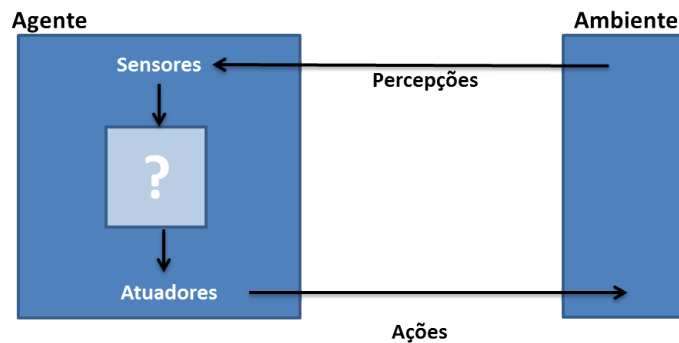


Figura 2.1: Interação entre agente e ambiente. Fonte: Russell e Norvig (2009).

Portanto conclui-se que a diferença entre agentes e os demais sistemas de computadores é que eles são capazes, de forma autônoma, de escolher a melhor ação possível diante de uma determinada situação, com o intuito de atender a um usuário ou a outro agente.

2.3 Propriedades

Diversos autores buscam propriedades que sejam capazes de diferenciar um agente dos demais softwares. Sendo assim, as principais propriedades levantadas, que são praticamente um consenso na literatura, foram definidas por Franklin e Graesser (1997). São elas:

- **Reativo:** recebe sinais de entrada dos seus sensores vindos do ambiente e pode executar ações contextualizadas que modifiquem o ambiente;
- **Autonomia:** deve possuir a possibilidade de agir sem a intervenção de usuários, além de poder controlar totalmente suas ações e seu estado;
- **Pró-atividade:** não devem apenas agir em resposta ao seu ambiente, mas devem agir por iniciativa própria de acordo com objetivos;

- Sociabilidade: devem poder interagir com outras entidades do ambiente de forma a resolver seus problemas e atividades;
- Adaptabilidade: devem poder mudar o seu comportamento devido à experiência;
- Receptividade: devem poder perceber o seu ambiente e responder adequadamente a mudanças;
- Temporalmente Contínuo: todo processo é executado de forma contínua;
- Mobilidade: podem estar aptos a se mover de uma máquina para outra;
- Flexibilidade: as ações realizadas pelos agentes não deverão seguir um roteiro;
- Caráter: os agentes devem possuir personalidades confiáveis além de estados emocionais.

Segundo Murta (2002) e Russell e Norvig (2009), o ambiente no qual os agentes estão inseridos também possuem propriedades:

- Acessibilidade: Indica se o ambiente pode ser completamente acessível através dos sensores dos agentes.
- Ser determinístico: É quando pode-se determinar a mudança do estado do ambiente a partir do seu estado atual ou de ações escolhidas pelo agente.
- Ser episódico: É quando o ambiente pode ser dividido em episódios, onde cada novo episódio não contém relação entre às ações que ocorreram nos estados anteriores.
- Estaticidade: Indica que mesmo com o passar do tempo o estado do ambiente permanece o mesmo;
- Ser discreto: É quando a quantidade de elementos contidos no ambiente são finitas.

2.4 Características

As características dos agentes se confundem com os aspectos dos sistemas complexos e agentes adaptativos. Segundo Surana et al. (2005), algumas dessas características são:

- Comportamento emergente e autônomo: Com as entidades individuais obedecendo a processos, surgem ações através de processos naturais e espontâneos;
- Adaptação e evolução: Reage ao ambiente e cria seu próprio ambiente.

Para HOLLAND (2003) os agentes adaptativos têm como característica principal a mudança de comportamento de acordo com seu aprendizado no tempo. Existem três características básicas dos agentes adaptativos:

- Sistemas de desempenho: comportamento do sistema quando as rotinas de aprendizagem param de funcionar;
- Sistema de atribuição de crédito: das regras que compõem um sistema de desempenho, algumas funcionam bem e outras não;
- Descoberta de novas regras: se existem regras que não funcionam bem é melhor substituir por outras que funcionam melhor.

Segundo Morgado (2005), Wooldridge (2008) e Franklin e Graesser (1997), a forma de classificação mais enfatizada na literatura é a representação interna dos agentes, caracterizando-os como cognitivos, reativos ou híbridos (cognitivo e reativo).

Ainda segundo Morgado (2005), os agentes cognitivos armazenam internamente uma representação do ambiente e dos outros agentes baseados em estados mentais. Além disso, eles são capazes de manter históricos de ações e interações realizadas para que, a partir dessas experiências, possam ser escolhidas suas próximas ações. Agentes desse tipo possuem organização social, controle deliberativo e sua forma de comunicação é direta. Já os agentes reativos, possuem um modelo de funcionamento baseado em ação-reação e na hipótese de que os comportamentos inteligentes são baseados na interação dos agentes com o ambiente. Eles possuem uma comunicação indireta, uma organização etológica e um controle não deliberativo.

2.5 *Sistemas Multiagente*

Sistemas Multiagentes (SMA) são sistemas computacionais formados pela reunião de vários agentes homogêneos ou heterogêneos. Apesar de possuírem uma função global em comum, cada um deles pode apresentar objetivos distintos, até por que cada agente é um elemento capaz de resolver problemas de forma independente e assíncrona. Desta forma, eles possuem características peculiares. São elas:

- Capacidade de interagir uns com os outros, trabalhando por uma meta em comum, o objetivo do sistema. Para isso, é necessário uma infraestrutura que possibilite a comunicação e/ou integração dos componentes do sistema, utilizando protocolos de interação baseados em seres humanos que pode incluir uma ou mais funcionalidades como: coordenação, cooperação, competição e negociação;

- Podem agir de forma autônoma, tomando decisões e atingindo os seus próprios objetivos.

Prasad e Lesser (1999) sugere que a investigação científica e a implementação prática desses sistemas estão focalizadas na criação de modelos e princípios que permitam a formulação de sociedades semiautônomas. Essas sociedades interagem de acordo com a conveniência e que estejam sob a supervisão de uma coordenação. Dentre as metodologias da coordenação, destacam-se as metodologias aplicáveis em domínios contendo agentes competitivos, onde a negociação é muito importante, e em domínios contendo agentes cooperativos, onde é relevante a organização estrutural da sociedade de agentes. De uma forma geral, segundo Jennings (2000), dentro de um sistema irá existir diferentes capacidades de percepção e ação, onde cada agente influenciará em partes do ambiente, podendo ou não haver coincidências nessas relações.

Os agentes formadores dos SMA, de acordo com GARCIA e SICHMAN (2003), podem ser classificados segundo a:

- perspectiva: representa o objetivo do sistema. Sistemas classificados dessa forma podem ser divididos como simulação, quando o objetivo são interações sociais, ou como resolução, quando o objetivo é resolver o problema de forma cooperativa e distribuída;
- abertura: classifica o sistema como aberto, quando agentes possuem objetivos variados, ou como fechado, quando o objetivo dos agentes é específico;
- granularidade: divide um sistema em baixa granularidade, quando ele é formado por poucos agentes, ou em alta granularidade, quando possui muitos agentes;
- composição: indica se o sistema é formado por agentes de um mesmo tipo (homogêneo) ou por agentes de diferentes tipos (heterogêneos);
- interação: define os diversos tipos de relacionamentos entre os agentes que compõem o sistema. Pode-se citar como formas de interação a comunicação, a cooperação, a coordenação e a negociação.

Estando presente a linguagem de comunicação e havendo a disponibilidade da construção de agentes, é necessário definir qual será a forma de interação, podendo essa ser entre agentes ou entre o agente e o ambiente. Essa interação pode ser realizada de forma direta, isto é, de forma explícita, ou indireta, onde ocorre emissão de sinais através do ambiente. São tipos de interação: a comunicação, a coordenação, a cooperação e a negociação.

A comunicação é um tipo de interação considerada como base. Esta interação permite uma expansão da capacidade de percepção, traz benefícios pela troca de informações e pode ainda influenciar na execução de ações ou alterar estados internos. Além disso, ela é baseada na teoria dos atos da fala que a trata como uma ação, evitando assim equívocos e tornando a conversação mais organizada e eficaz. As duas linguagens utilizadas são a interna e a externa, que permitem a troca de mensagens pelos agentes.

A coordenação tem como objetivo organizar a ordem das ações, definindo prioridades, o papel de cada agente, a forma de comunicação, a troca de informações e ainda a forma de acesso à recursos escassos. Morgado (2005), Wooldridge (2008) e Franklin e Graesser (1997) citam como exemplo de técnicas de coordenação:

- estruturas organizacionais, que baseiam-se em relações hierárquicas;
- normas e leis sociais, determinadas de maneira autoritária;
- planejamento multiagente, que define um plano para alcançar uma meta;
- intenções conjuntas, que possuem engajamentos conjuntos e individuais;
- modelagem mútua, que está de acordo com o que cada agente acredita sobre os objetivos e intenções dos demais, sem haver uma comunicação direta;
- planejamento global parcial, onde não existe um plano global e sim uma troca de informações para expandir planos parciais e assim alcançar o objetivo do sistema.

O tipo de coordenação dependerá do tipo do sistema onde um ou mais tipos podem ser utilizados.

Ainda se tratando dos tipos de interação, pode-se citar a cooperação que é realizada pela partilha de tarefas onde os agentes precisam da ajuda um do outro, em decorrência da incapacidade de executarem sozinhos determinada tarefa. Outra maneira de cooperação é a por partilha de resultados, que é quando os agentes ajudam-se mutuamente, através de trocas de informações, e como consequência são gerados acréscimos positivos para o sistema.

Já a negociação ocorre quando existem divergências em relação ao problema e um acordo passa a ser fundamental, ou ainda quando os agentes, apesar de poderem alcançar sua meta de maneira independente, entendem que negociando a execução podem trazer benefícios aos agentes envolvidos. Wooldridge (2008) define que o processo de negociação ocorre em uma série de rodadas com propostas baseadas em estratégias individuais de cada agente.

2.6 Simulação

Simulação é a imitação de algo real, ou de algo que será real, ao longo do tempo. De acordo com Morgado (2005), o “algo real” é representado por um sistema, responsável por englobar tudo que influencia no que será estudado durante a simulação. Esse sistema pode ser classificado como aberto, quando não é possível identificar ou quando não se conhece todas as variáveis que influenciam na simulação, ou como fechado que é quando se tem um conhecimento total de todas as variáveis influentes no sistema.

Os testes realizados na simulação não são executados direto no sistema real e sim em um modelo que simule, de forma adequada, o sistema real. Sendo assim, é necessária muita atenção para verificar se um modelo criado serve realmente para reproduzir um sistema real. As técnicas de simulação são comumente baseadas em modelos matemáticos ou estocásticos.

As simulações matemáticas e estocásticas normalmente restringem-se a levantar relações de causa-efeito entre as variáveis de entrada e de saída. Em decorrência disso, a simulação baseada em agentes surge com uma nova dinâmica de representação de um mundo artificial repleto de indivíduos, biológicos ou sociológicos, que sejam capazes de interagir entre si (ANDRADE, 2010). Cada indivíduo é representado por um agente que é descrito de forma descentralizada e é capaz de reproduzir ações locais em resposta a estímulos externos e a interações com outros agentes. Nesse tipo de simulação, não é possível determinar o objetivo geral observando partes do sistema, isso porque cada agente possui um objetivo específico e a união desses objetivos é que formará o objetivo geral.

As simulações multiagentes são normalmente utilizadas para modelagem de sistemas complexos e também em diversas áreas do conhecimento. O surgimento de diversos softwares que dão apoio e que facilitam o desenvolvimento de modelos desse tipo de simulação foi um fator que contribuiu para a crescente utilização dessa técnica. Dentre os softwares, podem-se destacar o AnyLogic (BORSHCHEV, 2005), NetLogo (TISUE; WILENSKY, 2004), Swarm (MINAR et al., 1996) e Repast (COLLIER, 2000).

A ferramenta AnyLogic foi desenvolvida pela empresa The AnyLogic Company, na linguagem de programação Java e possibilita o desenvolvimento de modelos e simulações em três abordagens distintas: dinâmica de sistemas, eventos discretos e sistemas multiagentes. Além disso, ela permite também a criação de modelos híbridos, ou seja, é possível utilizar em um mesmo modelo mais de um tipo de abordagem. Para utilização dessa ferramenta é necessário licença de uso.

NetLogo é uma linguagem de programação baseada em sistemas multiagentes e um ambiente favorável à modelagem de fenômenos naturais e sistemas complexos. Ela se adapta

muito bem a modelagem de sistemas complexos que evoluem ao longo do tempo, sendo projetada para ser utilizada tanto na educação quanto na pesquisa. Ela possibilita o controle de milhares de agentes de forma independente, todos operando simultaneamente. Ela traz também o princípio de “baixo limiar”, ou seja, até mesmo pessoas que não possuem experiência em programação conseguem desenvolver um modelo.

O sistema de simulação Swarm foi desenvolvida por Chris Langton com o objetivo inicial de modelar vidas artificiais. Atualmente ela é mantida pela Swarm Development Group, insituição sem fins lucrativos e é bastante utilizada para simulação de interação social e biológicos.

Já a ferramenta Repast foi desenvolvido por um grupo de pesquisa da Universidade de Chicago e possui como principais características:

- orientada a objetos;
- possui ferramentas para modelagem de redes sociais ;
- permite a modificação de agentes em tempo de execução;
- suporte à diversas linguagens de programação.

A tabela 2.1 faz uma comparação entre as ferramentas supracitadas.

Tabela 2.1: Comparação entre Ferramentas de Simulação Baseadas em Sistemas Multiagente.
 Fonte: Adaptada de SAKURADA (2009)

	AnyLogic	Swarm	Respast	Netlogo
Origem (País)	Rússia	EUA	EUA	EUA
Licença de uso	Software comercial	Software livre (código aberto)	Software livre (código aberto)	Software livre (código aberto)
Linguagens de Programação	Java	Java	Java, Python, .NET	Java
Sistema Operacional	Windows, Linux, MAC OSX, UNIX	Windows, Linux, MAC OSX, UNIX	Windows, Linux, MAC OSX, UNIX	Windows, Linux, MAC OSX, UNIX
Experiência em programação requerida?	Sim.Moderada	Sim.Elevada	Sim.Elavada	Não
Gráficos, diagramas e análise estatística integrados?	Sim	Sim	Sim	Sim
Disponibilidade de modelos de demonstração?	Sim	Sim	Sim	Sim
Disponibilidade de códigos dos modelos?	Não	Sim	Sim	Sim
Disponibilidade de tutoriais e guias?	Sim	Sim	Sim	Sim
Aplicações típicas	Planejamento, análise de processos em logística, manufatura, saúde, militar, ciências sociais	Sistemas biológicos, ecologia, sistemas de informação geográfica (GIS), ciências Sociais	Sistemas biológicos, ecologia, sistemas de informação geográfica (GIS), ciências Sociais	Sistemas biológicos, ecologia, sistemas de informação geográfica (GIS), ciências Sociais

Em decorrência da grande escala de utilização no meio acadêmico, da facilidade de obtenção do software, de sua documentação, da baixa complexidade de utilização e da sua gratuidade, o NetLogo foi a ferramenta escolhida para o desenvolvimento desse trabalho. No Apêndice B foram descritos os principais conceitos sobre essa ferramenta escolhida.

O Modelo

3.1 *Introdução*

Este capítulo objetiva mostrar o desenvolvimento de um modelo computacional para uma simulação capaz de representar um ser humano, que reside só, realizando as suas atividades diárias em seu lar. Esse modelo contempla a possibilidade de criação dinâmica de diferentes plantas de imóveis, a colocação de móveis e por fim, o posicionamento de sensores ao longo desta residência. Esses sensores são acionados no momento em que o morador passar próximo a eles e, estes mesmos sensores, armazenam automaticamente em um banco de dados a data e o horário de tal evento além do número de identificação do sensor.

O agente que representa o humano possui como principais propriedades a receptividade, onde o agente é capaz de perceber o ambiente e responder adequadamente às mudanças e temporalmente contínuo, onde todo processo de simulação é executado de forma contínua.

A estratégia adotada para o desenvolvimento do modelo seguiu as seguintes atividades: análise de requisitos, *brain storm*, projeto do modelo computacional e detalhamento do modelo utilizando o protocolo ODD (Overview, Design Concepts and Details) (GRIMM et al., 2006). Essas atividades serão descritas detalhadamente no decorrer deste capítulo.

3.2 *Análise de Requisitos*

Seguindo os preceitos da engenharia de software, o ciclo tradicional de desenvolvimento de sistemas é composto pelas fases de análise de requisitos, projeto, implementação e validação.

Segundo Sommerville (2003), requisitos de um sistema são descrições de todos os serviços que serão oferecidos pelo sistema juntamente com as suas restrições operacionais. Esses requisitos refletem as necessidades de clientes que precisam de uma solução computacional para resolução de um determinado problema. A análise de requisitos, de acordo com Pressman (2006), possibilita uma representação da informação e da função do sistema que pode ser traduzida em projeto procedimental, arquitetônico e de dados. Além disso, a especificação de requisitos proporciona tanto ao desenvolvedor quanto ao cliente critérios para avaliação da qualidade logo que o software for desenvolvido.

3.2.1 *Requisitos funcionais*

Requisitos funcionais de um sistema, de acordo com Sommerville (2003), definem detalhadamente as restrições operacionais, funcionalidades e os serviços do sistema. Sendo assim, diante do problema identificado, dos estudos realizados e do objetivo proposto, seguem abaixo os requisitos funcionais definidos:

- RF 01 Desenvolver questionário;

Descrição : Criar um questionário que seja capaz de obter os comportamentos de uma pessoa que reside sozinha.

- RF 02 Gerar arquivo txt com respostas do questionário;

Descrição : Possibilitar que o questionário respondido no RF01 seja salvo em um arquivo no formato txt.

- RF 03 Ler arquivo txt gerado pelo questionário;

Descrição : Possibilitar que arquivo salvo no RF02 seja lido e interpretado pelo sistema.

- RF 04 Permitir o desenho dinâmico de plantas de imóveis;

Descrição : É necessário que o usuário seja capaz de desenhar qualquer tipo de planta de imóvel.

- RF 05 Permitir a seleção e adição de móveis no imóvel;

Descrição : O sistema deverá conter um conjunto de móveis e possibilitar que esses móveis sejam adicionados à planta criada no RF 04.

- RF 06 - Permitir o posicionamento de sensores no imóvel;

Descrição : O sistema deverá permitir o posicionamento de sensores de monitoramento na planta criada na RF 04.

- RF 07 Carregar plantas pré-configuradas pelo sistema;

Descrição : Permitir a escolha de plantas já preconfiguradas com seus móveis, morador e formato.

- RF 08 Posicionar o morador no imóvel;

Descrição : Permitir o posicionamento inicial do morador em qualquer cômodo do imóvel criado no RF04.

- RF 09 Não permitir que o morador ultrapasse as paredes do imóvel;

Descrição : O morador não deverá ser capaz de ultrapassar os domínios criados pelo RF04.

- RF 10 Salvar em um banco de dados uma string contendo informações como data e hora na qual o morador passou por um sensor, bem como o número de identificação do sensor;

Descrição : O sistema deverá armazenar em uma base de dados uma string contendo informações como data e hora toda vez que um sensor for acionado pela passagem do morador.

- RF 11 Gerar rotina de atividades do morador com base no arquivo *txt* obtido pelo questionário;

Descrição : O sistema deverá conter um calendário e diariamente deverá gerar uma rotina para o morador, seguindo uma distribuição pré-fixada, com base no arquivo *txt* de resposta do questionário.

3.2.2 Requisitos não funcionais

Os requisitos não funcionais de um sistema não estão diretamente ligados às funcionalidades do sistema. Eles podem definir restrições sobre o processo de desenvolvimento e são frequentemente aplicados ao sistema como um todo (SOMMERVILLE, 2003). Seguindo esse conceito os seguintes requisitos não funcionais foram levantados:

- RNF 01 É necessário banco de dados MySQL;
- RNF 02 É necessário instalação da JDK mais recente;
- RNF 03 É necessário que ele seja executado pelos sistemas operacionais Windows, MAC OS e Linux;

3.3 Diagrama de Casos de Uso

Diante dos requisitos supracitados, o diagrama de casos de uso pôde ser desenhado, como mostra a Figura 3.1. Analisando o diagrama apresentado na Figura 3.1 observa-se a participação de apenas um ator, o Usuário, e algumas atividades *include*. Atividades *include* são aquelas fundamentais para o funcionamento das atividades a qual está associada.

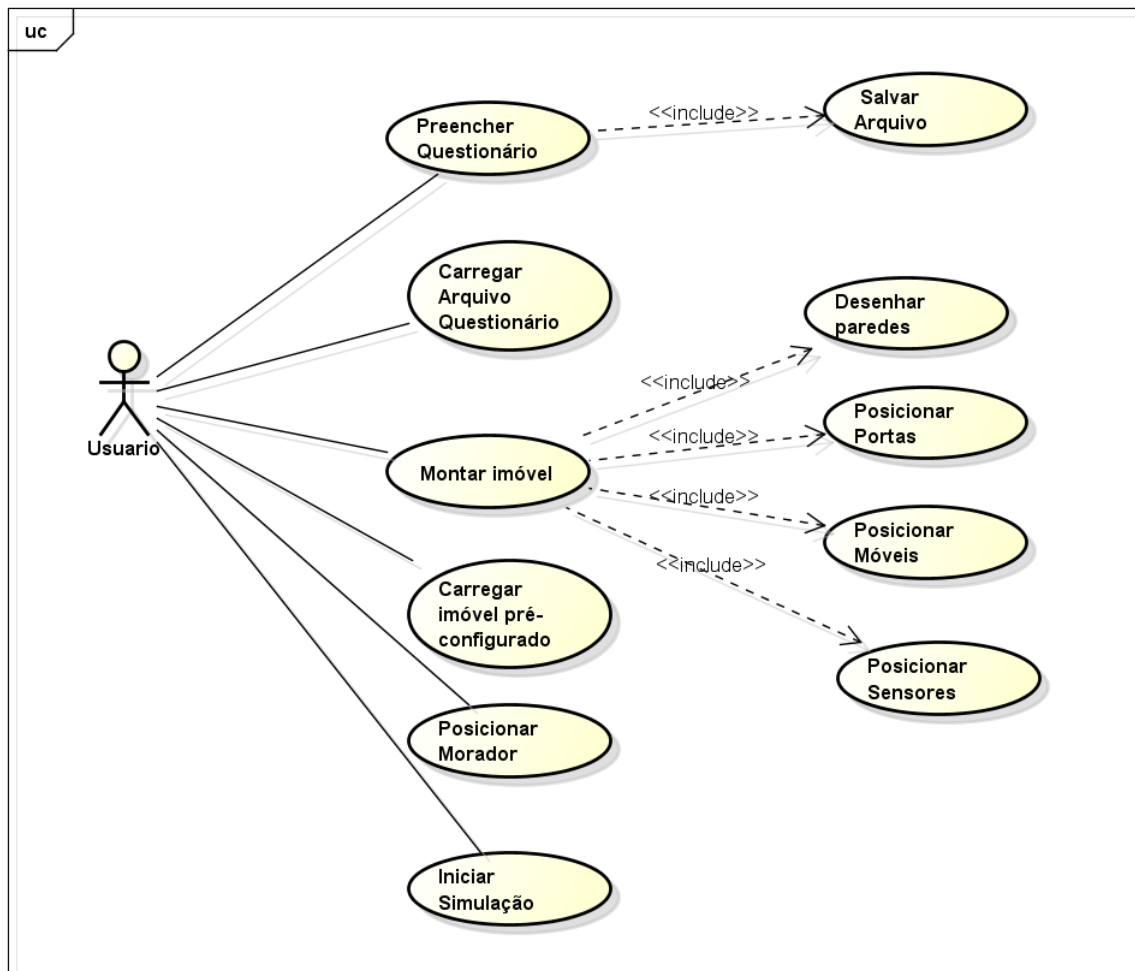


Figura 3.1: Diagrama de Casos de Uso. Fonte: O Autor.

Sendo assim, as atividades básicas que poderão ser executadas pelo usuário são:

- (a) Preencher questionário: O usuário deverá entrevistar pessoas que se encaixem no perfil da pesquisa e preencher o questionário. Depois de preenchido, um arquivo txt deverá ser criado contendo o conteúdo da entrevista;
- (b) Carregar Questionário: Será necessário que as respostas do questionário sejam abertas e interpretadas;
- (c) Montar Imóvel: Será possível a construção de imóveis;
- (d) Carregar imóvel pré-configurado: O sistema deverá possuir imóveis prontos e com móveis posicionados, sendo possível a adição de sensores e a edição das plantas;
- (e) Posicionar Morador: O usuário poderá escolher o local onde será adicionado o morador;
- (f) Iniciar a Simulação: É quando se iniciam as atividades rotineiras do morador em seu lar;

3.4 Diagrama de Componentes

Com os requisitos especificados foi possível a montagem do diagrama de componentes. Esse diagrama serve para ilustrar a arquitetura da solução computacional proposta para atender aos requisitos especificados. Na Figura 3.2 é possível verificar que a arquitetura proposta nesse projeto de mestrado é formada por três macro componentes: *Software Questionário*, *Simulação Anjo da Guarda* e *Base de Dados*.

O componente *Software Questionário* é responsável por identificar o cotidiano de pessoas que residem sozinhas e salvar essas informações em um arquivo *txt*. Esse arquivo servirá de entrada para o outro componente do sistema o *Simulação Ajno da Guarda*.

O componente *Simulação Ajno da Guarda* permite a simulação dos indivíduos entrevistados a partir do arquivo carregado, a montagem dinâmica de imóveis e o armazenamento de informações no componente *Banco de Dados*. Para ser possível o envio de informações para o *Banco de Dados*, o componente *Simulação Ajno da Guarda* é formado pelo módulo interno *NetLogo-SQL*, responsável por estabelecer a conexão entre os componentes.

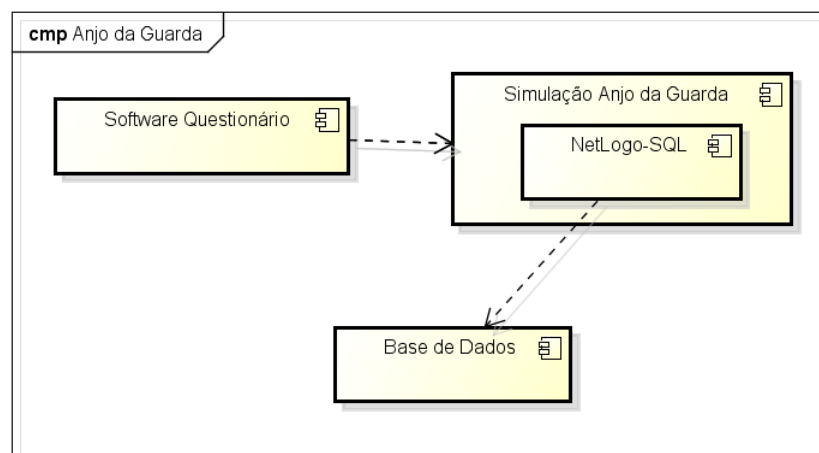


Figura 3.2: Diagrama de Componentes. Fonte: O Autor.

3.5 Modelo Conceitual

Depois de realizada toda a análise dos requisitos e criado uma arquitetura de componentes, foi iniciado o processo de desenvolvimento do modelo conceitual do projeto. Primeiramente foi realizado um *brain storm*. Essa atividade é definida pela colocação em um papel de todas as ideias que surgirem para o desenvolvimento de uma solução computacional no contexto do problema levantado. Em seguida, as ideias anotadas no *brain storm* foram refinadas e foi projetado um modelo com o objetivo de atender a todos os

requisitos especificados. De uma forma ampla, o modelo proposto visa simular, com base em um questionário, comportamentos diários de moradores solitários. Com esse intuito, sete entidades principais foram identificadas. São elas:

- **Humano:** É a entidade responsável por representar o morador. Esta entidade possui uma estrutura interna responsável por armazenar o posicionamento das portas e dos móveis existentes no ambiente. Além disso, possui uma lista contendo um conjunto fixo de atividades domésticas juntamente com os horários de execução de cada uma dessas atividades;
- **Ambiente:** Esta entidade representa o cenário onde ocorrerá a simulação;
- **Imóvel:** Esta entidade é responsável por delimitar o ambiente e criar uma residência;
- **Móvel:** Esta entidade representa os móveis do imóvel. Essa entidade é utilizada pela entidade Humano para simular a realização das atividades diárias e possui uma variável para representar a sua forma geométrica;
- **Porta:** Esta entidade representa as portas do imóvel. Esta entidade é extremamente importante, pois como o Humano não poderá passar entre as paredes do imóvel, as portas serão os pontos de acesso entre os cômodos da residência. Ela possui variáveis que são responsáveis por representar a sua posição no ambiente;
- **Sensor:** Esta entidade representa os objetos sensores que armazenarão, em um banco de dados, informações sobre o cotidiano do morador. Esta entidade possui um evento que é capaz de captar a movimentação da entidade Humano. Toda vez que ele capta essa movimentação, utiliza os valores das variáveis contidas na entidade Calendário para enviá-los a um banco de dados;
- **Calendário:** Entidade responsável por representar os dias e horas da simulação. Ela possui um contador que representa os segundos, o qual é incrementado, em tempo de execução, durante a simulação. A partir dos segundos as suas variáveis internas que representam os minutos, horas, dias e datas são modificadas;

A Figura 3.3 ilustra a arquitetura conceitual do modelo proposto. Conforme pode ser observado, são mostrados os relacionamentos entre as entidades juntamente com as suas cardinalidades. É importante salientar que apesar do software de simulação escolhido utilizar uma linguagem de programação estruturada, essa modelagem é fundamental para a implementação da solução.

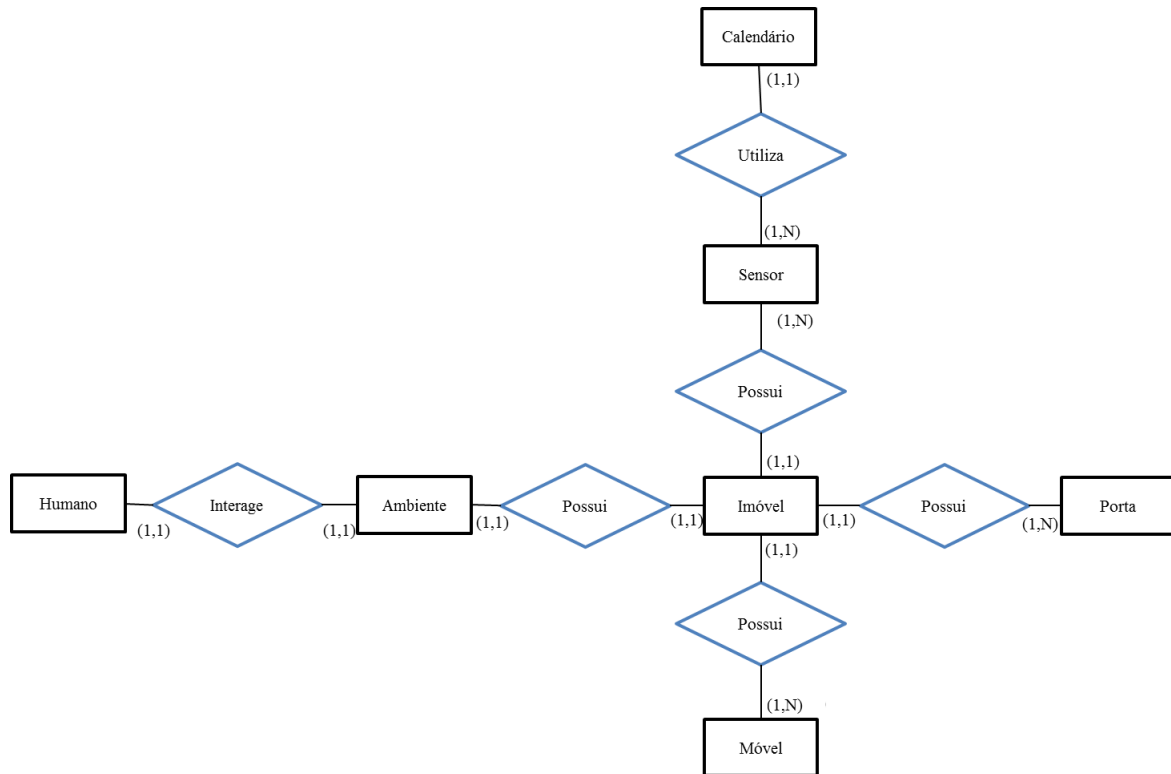


Figura 3.3: Arquitetura Conceitual do Modelo Proposto. Fonte: O Autor.

Após as etapas de análise de requisitos, *brain storm* e de criação do modelo computacional, o modelo criado foi detalhado utilizando o protocolo ODD.

3.6 Protocolo ODD

O detalhamento do modelo foi realizado utilizando o protocolo ODD (Overview, Design Concepts and Details). Esse protocolo foi proposto por Grimm et al. (2006) e tem como objetivo a padronização de descrições dos modelos baseados em agentes.

O protocolo ODD é dividido em três blocos, são eles: *Visão Geral*, *Conceitos de Projeto* e *Detalhes* (Figura 3.4). O bloco *Visão Geral* fornece uma visão geral do objetivo e da estrutura do modelo. Ele é formado por três elementos: *propósito*, que descreve a finalidade do modelo, *variáveis de estado e escala*, responsável por definir as entidades, o estado das variáveis e as escalas de representação espaço-temporal que compõem o modelo e, por fim, *visão de processo e escalonamento*, que define os processos do modelo.

Visão Geral	Propósito
	Variáveis de estado e escala
	Visão de processo e escalonamento
Conceitos de Projeto	Conceitos de Projeto
Detalhes	Inicialização
	Entrada
	Sub-modelos

Figura 3.4: Os sete elementos do protocolo ODD. Fonte:Grimm et al. (2006).

O bloco *Conceitos* de projeto possui um *checklist*, desenvolvido por Grimm et al. (2006), com o objetivo de auxiliar na fase de concepção do projeto e na comunicação interna dos modelos baseados em agentes. Esse *checklist* é formado por itens que, quando não se aplicam ao modelo, devem ser ignorados.

O último bloco do protocolo ODD é o *Detalhes*, importante para que os experimentos possam ser reproduzidos. Ele é subdividido em: *inicialização*, que define quais são os valores iniciais das variáveis de estado, *entrada*, que descreve como os dados de entradas são utilizados, gerados e obtidos e os *sub-modelos*, responsável por detalhar os processos listados no item visão de processo e escalonamento. A seguir o modelo Anjo da Guarda é detalhado seguindo o protocolo supracitado.

3.6.1 Propósito

O modelo possui como objetivo a simulação determinística de indivíduos em residências unipessoais realizando atividades domésticas rotineiras. Além disso, esse modelo deverá salvar informações sobre o comportamento diário desses indivíduos. Essas informações serão coletadas através de sensores distribuídos ao longo da residência.

3.6.2 Variáveis de estado e escalas

O modelo de simulação anjo da guarda é composto pelas entidades de baixo nível Humano, Móveis, Portas, Paredes e Sensores. O agente Humano movimenta-se pelo ambiente com o intuito de cumprir um conjunto fixo de atividades diárias, respeitando os limites espaciais do imóvel dados através do agente Paredes. Além disso, o Humano utiliza os agentes Portas para movimentar-se entre os cômodos da residência e os agentes Móveis para simular a realização das suas atividades domésticas. Já os agentes do tipo Sensores são

responsáveis por captar a movimentação do morador e salvar em um banco de dados informações como hora e data dessa movimentação.

O ambiente da simulação é dinâmico, podendo ser alterado por um usuário a qualquer momento, ou seja, a qualquer momento da simulação é possível modificar o ambiente inserindo móveis, sensores, portas e/ou paredes. Só não é possível inserir vários agentes do tipo humano, pois, como o modelo se propõe a simular pessoas que residem sozinhas, existe apenas um. A Tabela 3.1 mostra as variáveis de baixo nível do modelo juntamente com as suas descrições e tipos.

O tempo da simulação anjo da guarda é modelado por unidades discretas de execução denominadas *ticks*. O *tick* é uma unidade de tempo artificial e foi definido nesse modelo que um *tick* corresponde a um segundo simulado. Dessa forma, um segundo no modelo pode ser simulado por vários segundos ou por frações de segundos de processamento computacional, ajustáveis na simulação. A noção de tempo se expande para definições mais complexas, como a existência de horas, dias da semana, mês e ano. As simulações são sempre inicializadas em um determinado instante, representado por uma data e hora específica, e finalizados em outro instante. O padrão de medida adotado para tempo foi o segundo, para o espaço, metros e para a velocidade, metros por segundos.

Tabela 3.1: Parâmetros, descrições e tipos utilizados no modelo Anjo da Guarda - Fonte: O autor

Parâmetro	Descrição	Tipo
contadorMoveis	Identificador e contador dos móveis	inteiro
contadorPortas	Identificador e contador das portas	inteiro
contadorSensor	Identificador e contador dos sensores	inteiro
tb-memoriaPorta	Representa a memória do ser humano e armazena as coordenadas x e y referentes à posição das portas	table
tb-memoriaMovel	Armazena as coordenadas x e y referentes à posição dos móveis	table
tb-memoriaPortasMovel	Armazena os identificadores das portas que dão acesso a cada móvel	table
tb-memoriaLadoPortasMovel	Armazena os identificadores referentes ao lado das portas que cada móvel se encontra	table
diaSemana	Representa o dia da semana	string
Dia	Representa o dia do mês	inteiro
Mês	Representa o mês	inteiro
Ano	Representa o ano	inteiro
Horas	Representa as horas da simulação	inteiro
Minutos	Representa os minutos da simulação	inteiro
Segundos	Representa os segundos e seu valor é setado a partir da variável de ambiente <i>tick</i>	<i>tick</i>
mt-Adjacência	Representa o grafo formado pelas portas do imóvel	table
atividade1	Representa se o agente Humano está realizando ou não alguma atividade	booleano
tb-cotidiano	Representa os horários das atividades diárias do agente ser Humano. Diariamente esses horários são alterados seguindo uma distribuição pré-fixada baseada no arquivo de respostas do questionário.	table
CaminhoArquivo	Arquivo com as respostas do questionário. Ele serve como base para geração do cotidiano.	string

3.6.3 *Visão de processo e escalonamento*

O modelo Anjo da Guarda possui dois macroprocessos que podem ser divididos pela perspectiva dos agentes Humano e Sensores, descritos a seguir.

3.6.3.1 *Agente Humano*

A simulação anjo da guarda possui como um dos processos principais a realização de atividades domésticas pelo agente Humano. Sendo assim, o modelo possui uma tabela contendo uma lista fixa de atividades domésticas a ser realizadas diariamente pelo agente Humano, juntamente com o horário de inicialização de cada uma dessas atividades. Essa lista de atividades foi criada com base em pesquisas para identificação das principais atividades domésticas do ser humano e, os horários de cada atividade são gerados a partir do arquivo *txt* de respostas da aplicação do questionário, ou seja, se um usuário responde ao questionário afirmando que ele dorme entre 20 e 21 horas, na simulação o horário da atividade de dormir desse usuário será diariamente setada, seguindo uma distribuição triangular, com valores entre 20 e 21 horas.

É importante salientar que cada pessoa entrevistada possuirá uma lista de atividades diferenciada. Isso deve-se ao fato de que cada pessoa possui faixas de horários específicos para realização das suas atividades domésticas e, além disso, poderão existir algumas atividades que não são realizadas por algumas pessoas.

Depois de gerada a tabela contendo o cotidiano do agente Humano, esse agente verifica se a hora atual da simulação é a hora de realização de alguma das suas atividades, em caso positivo, ele se movimenta pelo ambiente com o intuito de cumprir essa atividade, e, em caso negativo, ele aguarda até o horário da sua próxima atividade. A Figura 3.5 mostra o fluxograma desse processo.

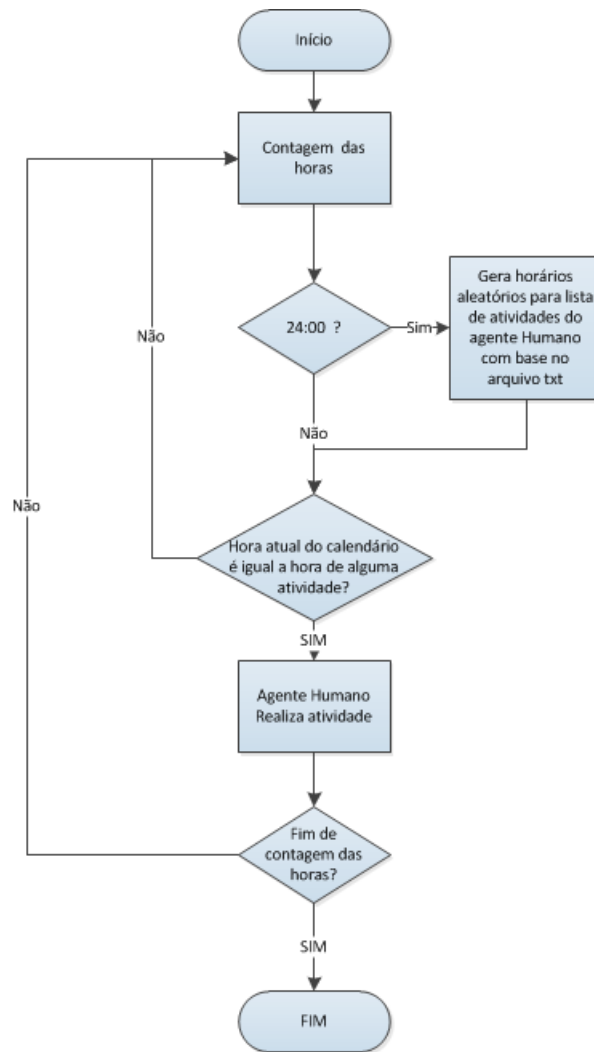


Figura 3.5: Fluxograma do processo Agente Humano. Fonte: O autor.

3.6.3.2 Agentes Sensores

O segundo macro processo da simulação Anjo da Guarda é o monitoramento do comportamento do morador através de agentes do tipo sensor. Esses sensores são configurados para captar, no raio de um metro, qualquer movimentação do agente humano e salvar em um banco de dados informações como hora e data dessa movimentação. Ou seja, partindo do princípio que nesse modelo foi definido que um patch possui um metro quadrado, toda vez que o agente humano passar num raio de um metro de distância de qualquer sensor, esse sensor será acionado e enviará para o banco de dados a seguinte string de informações: *“dia/mes/ano;horas:minutos:segundos;identificador do sensor;”*

Na Figura 3.6 pode-se observar um agente Humano, um agente do tipo Sensor, representado por um x , e oito agentes do tipo patch, representado pelos quadrados ao redor do sensor. Toda vez que o agente Humano passar por um dos oito patches, ou seja, pelo raio

de um patch do sensor, esse sensor é acionado e envia para o banco de dados a data e a hora do evento juntamente com o seu número de identificação.

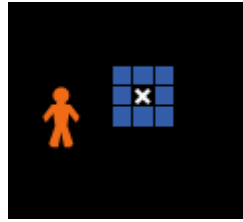


Figura 3.6: Acionamento do agente sensor pelo agente humano. Fonte: O autor.

3.6.4 Conceitos de Projeto

A seguir serão detalhados os itens do *checklist* de conceitos de projeto que se aplicam ao modelo Anjo da Guarda:

- *Adaptação*: O agente humano deve ser capaz de perceber alterações no ambiente como, por exemplo, à adição e/ou remoção das paredes, portas e móveis e reagir a cada uma dessas alterações;
- *Conhecimento*: o agente humano possui uma memória onde armazena a localização de todas as portas e móveis do imóvel;
- *Interações*: o agente humano interage com o ambiente verificando mudanças, pois se elas ocorrerem deverá reagir a elas, e com os agentes Móveis com o propósito de simular a realização das suas atividades domésticas;
- *Observação*: os resultados usados para testar o modelo dependem do posicionamento do sensor, ou seja, como o sensor capta a movimentação do morador poderá haver posições do sensor na qual o morador não passe em nenhum momento da simulação. Além disso, os resultados dependem também do arquivo txt de resposta do questionário, pois ele servirá como base para gerar os horários de cada uma das atividades.

3.6.5 Inicialização

A configuração inicial da simulação é definida pelo usuário, onde eles podem optar por desenvolver um modelo de imóvel novo ou por carregar um modelo pré-configurado. Sendo assim, os valores iniciais das variáveis de estado são baseados em um arquivo *txt* de

parâmetros, criado a partir das respostas da aplicação do questionário desenvolvido nessa pesquisa, e dos parâmetros definidos pelos usuários no início da simulação.

3.6.6 Entrada

O modelo é inicializado com um arquivo de dados, no formato *txt*, contendo as faixas de horários das atividades domésticas, informadas por usuários entrevistados. Essas faixas de horários são inseridas no arquivo *txt* na mesma sequência em que encontra-se no questionário. Cada horário no arquivo é separado por *!*. Dessa forma, esse arquivo possui o seguinte layout: “*horas.minuto!*”. Ele servirá como entrada para a geração do cotidiano do agente Humano.

O outro conjunto de dados que é fornecido ao modelo é gerado a partir das escolhas do usuário. Caso ele opte por configurar a simulação a partir de um dos modelos pré-configurados, os dados de entrada foram definidos para que o modelo se assemelhasse com imóveis reais. No caso do usuário optar pelo desenvolvimento de um novo imóvel, ele será responsável por inicializar os valores das variáveis de entrada (Tabela 3.1) imprescindíveis ao funcionamento do modelo.

Finalmente, os últimos dados de entrada a ser configurados, também necessitam do usuário, que são as coordenadas para posicionamento de agentes do tipo sensor e a configuração da *string* de conexão com um banco de dados MySQL. Essa *string* de conexão é responsável por determinar em qual local serão armazenadas as informações.

Sendo assim, a dinâmica do modelo varia conforme o tempo passa na simulação. As condições ambientais são uma entrada para o modelo. A saída do modelo corresponde a uma resposta aos parâmetros iniciais ou dados de entrada e possuiu os campos: “*dia/mes/ano;horas:mdo sensor;*” .

3.6.7 Submodelos

O objetivo do modelo anjo da guarda é a simulação de um indivíduo dentro da sua residência. Sendo assim, o principal processo desse modelo é a simulação da realização das atividades domésticas. Conforme explanado anteriormente, após pesquisas realizadas, foram identificadas as principais atividades domésticas dos seres humanos e, a partir dessas atividades levantadas, foi elaborado um questionário com o propósito de identificar as faixas de horários e quais das atividades são realizadas por cada usuário entrevistado. Esse questionário terá como saída um arquivo *txt* que servirá como entrada para o modelo

gerar o cotidiano de cada morador.

Ao agente humano identificar que está no horário para realização de uma das suas atividades, ele precisará se movimentar pelo ambiente com o propósito de cumprir essa atividade. A movimentação ocorre da seguinte maneira:

- As atividades são divididas em subatividades e essas, por sua vez, estão vinculadas a um móvel, ou seja, para o humano cumprir uma atividade é necessário que ele se direcione a um ou mais móveis;
- Para cumprir a subatividade é necessário que o agente humano se direcione do local no ambiente onde ele se encontra até o móvel destino. Como o ambiente é dinâmico e pode ser alterado a qualquer momento pelo usuário, o móvel vinculado à subatividade pode não estar presente no ambiente. Se isso ocorrer, o agente humano é capaz de verificar se existe algum móvel adicionado que tenha a mesma finalidade do móvel procurado inicialmente, por exemplo, se após ir ao vaso sanitário, não encontrar a pia do banheiro ele irá verificar se existe no ambiente a pia da cozinha e, se não existir nenhum móvel capaz de substituir o primeiro móvel procurado, que era a pia do banheiro, ele emite uma mensagem de log informando sobre a necessidade de um móvel para o cumprimento daquela atividade;
- Para que o agente humano consiga sair do local onde ele encontra-se e chegar até o móvel destino é necessário que as seguintes regras sejam seguidas:
 - O ambiente no qual o humano está inserido é formado por um imóvel e esse por sua vez é formado por cômodos, paredes, portas e móveis. Dessa forma, o agente humano possui uma memória na qual armazena o posicionamento de todos os móveis existentes no imóvel, juntamente com qual cômodo dá acesso a esses móveis, ou seja, quais portas dão acesso aos móveis, bem como a localização de cada uma das portas e sua localização atual. É importante salientar que as portas do imóvel formam um grafo que é representado no modelo através de uma matriz de adjacência. Sendo assim, para o agente humano chegar ao móvel destino, primeiramente ele analisa em qual cômodo ele está e depois verifica em qual cômodo está o móvel que ele deseja alcançar. Se ambos estiverem no mesmo local, o agente humano vai direto ao móvel e, se estiverem em cômodos diferentes, o agente humano verifica, através do algoritmo de Dijkstra (1959), qual o menor caminho de portas que ele deverá passar para conseguir chegar ao cômodo onde está localizado o móvel. Somente depois que ele chega à porta que dá acesso ao cômodo no qual se encontra o móvel é que ele direciona-se ao móvel propriamente dito;
 - O agente humano ao se movimentar deverá respeitar os limites impostos através do agente parede, ou seja, ele não deverá passar através dos agentes paredes.

Sendo assim, toda vez que o agente humano detectar a presença de um patch da cor azul à sua frente, ele deverá desviar para esquerda até não encontrar mais parede e seguida continua sua movimentação para chegar ao móvel ou porta.

Após conclusão do detalhamento do modelo, este foi implementado de forma iterativa e incremental com realização de testes no final de cada iteração. Os produtos gerados através dessa implementação estão descritos no capítulo seguinte.

Implementação do Modelo

A implementação do modelo explanado do capítulo anterior obteve como produtos dois softwares: o software Questionário e a simulação Anjo da Guarda. O software questionário foi elaborado com o objetivo de identificar as atividades rotineiras de uma pessoa em seu ambiente domiciliar e salvar essas informações em um arquivo de texto que servem como base para o comportamento do agente ser humano na simulação Anjo da Guarda. A implementação deste questionário foi realizada utilizando a linguagem Java devido a sua portabilidade.

Já a implementação da simulação, conforme descrito anteriormente, foi realizada utilizando o NetLogo e, para que funcione corretamente, é necessário que esteja instalado o pacote de extensão Netlogo-sql. Esse pacote além de permitir a conexão direta com os bancos de dados MySQL e PostgreSQL, possui suporte a outros bancos de dados através de drivers JDBC (*Java Data Base Connectivity*) compatíveis e possibilita a execução de comandos SQL (*Structured Query Language* - Linguagem de Consulta Estruturada), tanto DML quanto DDL.(BLOM; QUAKKELAAR; ROTTEVEEL, 2010)

Neste trabalho de pesquisa foi utilizado o MySQL que é um sistema de gerenciamento de banco de dados de código aberto, que utiliza a linguagem SQL para a realização de consultas. Segundo a ORACLE (2013) ele é o SGBD em código aberto mais popular entre os existentes, possuindo clientes como NASA, Youtube e Facebook.

4.1 O Questionário

Para a elaboração do questionário foram realizados diversos estudos com o propósito de identificar as principais atividades diárias dos seres humanos dentro dos seus lares. Após identificadas essas atividades, foi desenvolvido uma ferramenta que possibilita a gravação das respostas do entrevistado em um arquivo no formato *txt* e em um padrão capaz de ser interpretado pela ferramenta de simulação.

Com o intuito de melhorar a usabilidade e a dinâmica da ferramenta, as perguntas criadas foram subdivididas em seis tópicos, sendo eles: dados pessoais, dados referentes a atividade de dormir, dados referentes a refeições, dados referentes a entretenimentos, dados referentes a necessidades médicas e de higiene e dados referentes a lazer. A Figura 4.1 mostra a tela inicial da ferramenta desenvolvida. Já a ferramenta completa, com todas

as perguntas pode ser observada no Apêndice A desse trabalho.

Questionário

Arquivo

1. Dados Pessoais

1.1 Sexo: Masculino Feminino

1.2 Idade:

1.3 Cidade onde mora:

2. Dados referentes à atividade de dormir

2.1 A que horas costuma dormir de segunda à sexta-feira ?
Entre e Ex.: 21:50

2.2 A que horas costuma dormir nos feriados ?
Entre e Ex.: 21:50

2.3 A que horas costuma acordar de segunda à sexta-feira ?
Entre e Ex.: 21:50

2.4 A que horas costuma acordar nos feriados ?
Entre e Ex.: 21:50

2.5 Quantas vezes costuma levantar durante a noite para ir ao banheiro ?

2.6 Quantas vezes costuma levantar durante a noite para tomar remédio ?

2.7 Quantas vezes costuma levantar durante o ato de dormir para beber ou comer algo ?

2.8 Você costuma tirar cochilo(s) ao longo do dia ?

Figura 4.1: Software Questionário . Fonte: O Autor.

4.2 A Simulação Anjo da Guarda

Como segundo produto da implementação, foi gerada uma simulação chamada Simulação Anjo da Guarda. Com ela é possível escolher ou montar uma planta de uma residência, posicionar sensores em qualquer parte desses imóveis, visualizar o cotidiano de um indivíduo em diferentes plantas de apartamentos e dessa forma monitorar comportamentos. Para a utilização desse produto é necessário que alguns passos sejam realizados até que se inicie de fato a simulação. A Figura 4.2 ilustra um fluxograma com as etapas para manuseio do software que estão detalhadas no decorrer desse capítulo.

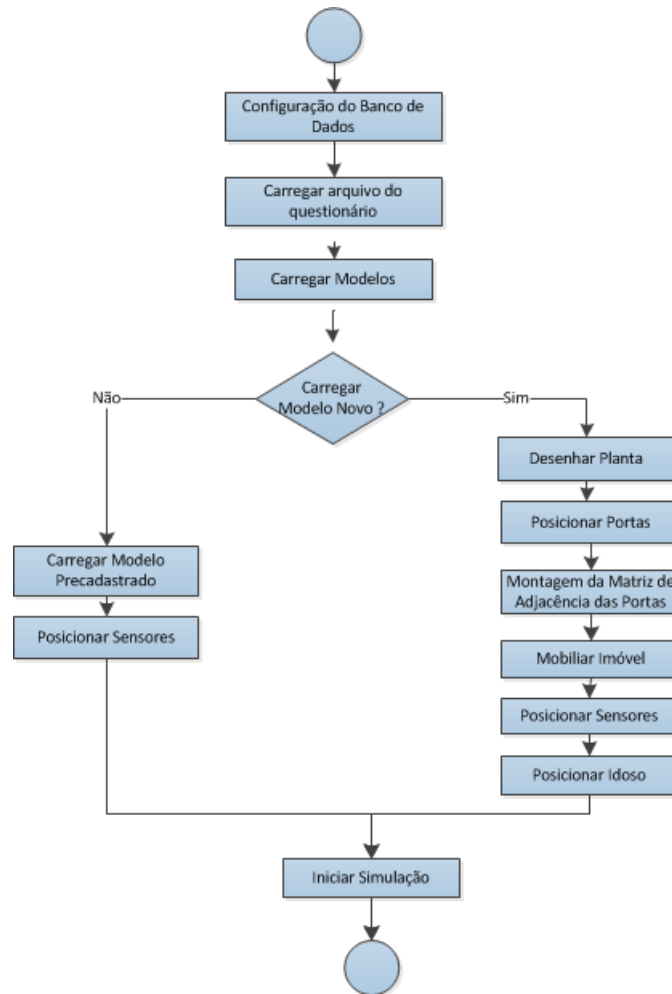


Figura 4.2: Fluxograma de utilização da Simulação Anjo da Guarda . Fonte: O Autor.

Conforme pode ser visto na Figura 4.2, a primeira etapa é a de configuração do banco de dados. Como citado anteriormente, o banco de dados utilizado foi o MySQL. Ele pode ser facilmente instalado utilizando o manual contido no site da ORACLE (2013). Depois de instalado o MySQL é necessário que seja executado o seguinte script:

```

1. CREATE SCHEMA `simulacao` DEFAULT CHARACTER SET utf8 ;
2. CREATE TABLE `simulacao`.`sim_sensor` ( `id` INT NOT NULL ,
`sim_data` VARCHAR(25) , `sim_id_sensor` INT , `sim_ativo` SMALLINT
NOT NULL DEFAULT 1 , PRIMARY KEY (`id`) ) ENGINE =
InnoDB DEFAULT CHARACTER SET = utf8;
  
```

Figura 4.3: Script de Criação da Base de Dados . Fonte: O Autor.

A primeira linha desse código é responsável pela criação do schema chamado *simulação*, com o padrão de caracteres *utf8* que é um tipo de codificação *Unicode* de comprimento

variável. Já a segunda linha cria, no schema *simulação*, uma tabela chamada *sim_sensor* possuindo os campos *id* (identificador do registro), *sim_data* (responsável por armazenar a data e hora na qual o sensor foi acionado), *sim_id_sensor* (armazena o número de identificação do sensor ativado) e *sim_ativo* (que informa se a linha foi ou não deletada).

Após realizadas essas instruções, que só precisarão ser realizadas uma vez, o aplicativo *simulação* poderá ser iniciado. Porém, continuando com a configuração do banco de dados, é necessário que seja procurado na paleta Code do Netlogo o trecho de código mostrado abaixo:

```
to conexao
  sql:configure "defaultconnection" [ ["host" "" ] ["port" 3306]
    ["user" "" ] ["password" "" ] ["database" "simulacao" ] ]
end
```

Figura 4.4: Código de Conexão com o Banco de Dados . Fonte: O Autor.

Esse trecho é responsável por estabelecer uma conexão entre a aplicação e o banco de dados. Para isso é necessário preenchê-lo nos locais onde encontram-se as “ ”, com as seguintes informações : a) *host*, ou seja, o endereço do servidor onde se encontra instalado o banco de dados. No caso do banco de dados estar instalado na mesma máquina na qual está sendo executada a simulação, deverá ser inserido o valor “*localhost*” e, em caso negativo, deverá ser colocado o IP da máquina destino; b) *user*, nesse campo deverá ser colocado o usuário criado para ter acesso ao banco e c) *password*, onde será colocada a senha do usuário escolhido. Com isso, a configuração do banco de dados é concluída.

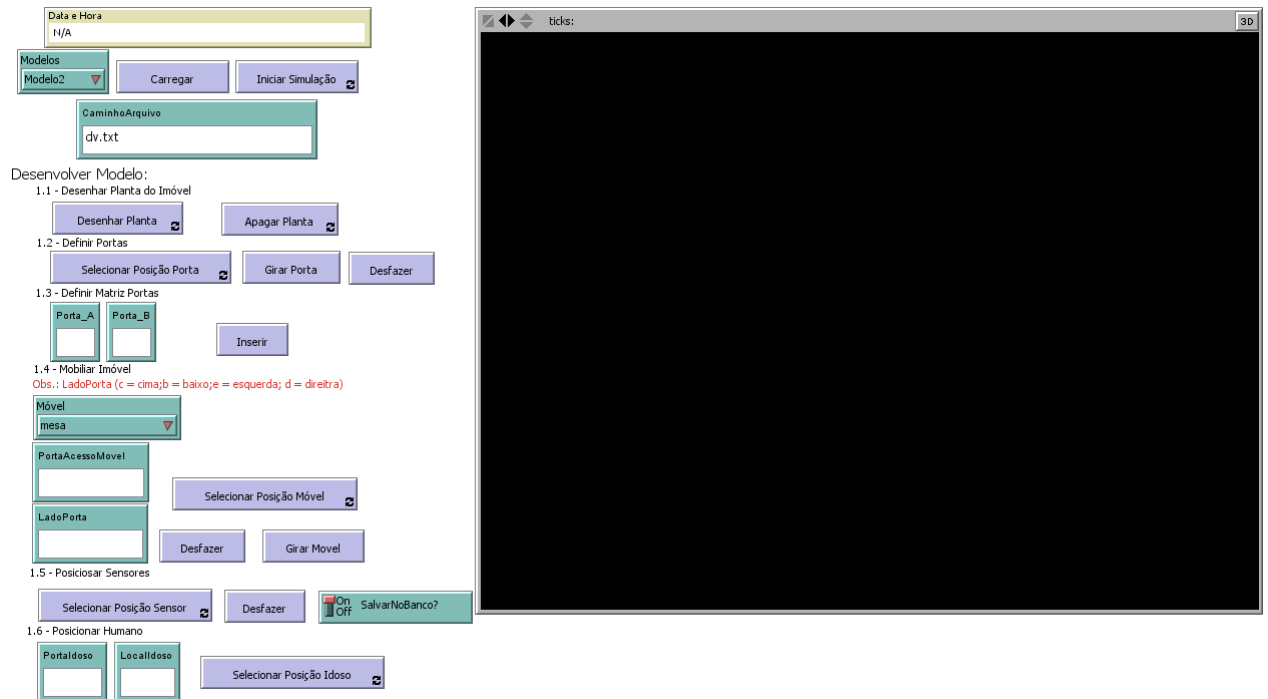


Figura 4.5: Tela inicial da Simulação Anjo da Guarda . Fonte: O Autor.

A Figura 4.5 mostra a tela inicial do sistema. Nela pode-se notar no topo um monitor Data e Hora. Como o próprio nome propõe, ele serve para ilustrar a contagem dos segundos, minutos, horas, dias, meses e anos durante a simulação. Além do monitor, é possível observar também uma série de componentes, como botões, interruptores, *combo boxes* e caixas de entradas que são responsáveis por preparar, começar, parar, construir ou alterar o modelo. Sendo assim, continuando com as atividades do fluxograma da Figura 4.2, é necessário nesse momento que o componente *CaminhoArquivo* seja configurado. Ele é responsável por armazenar o nome completo do arquivo que deverá ser carregado. Para o seu preenchimento é importante que o arquivo obtido pelo software *Questionário* esteja no mesmo diretório que o da simulação *Anjo da Guarda* e que ele seja preenchido colocando-se o nome completo do arquivo, ou seja, o nome do arquivo prosseguido pela sua extensão *.txt*. O preenchimento correto desse componente é imprescindível para o funcionamento da simulação, pois é a partir do nome do arquivo indicado que toda a rotina do agente é gerada.

Seguindo o fluxograma da Figura 4.2, a próxima etapa refere-se ao carregamento dos modelos. Conforme pode ser observado na Figura 4.6, o programa dá a opção de montar quatro modelos pré-definidos ou montar um novo modelo a ser desenvolvido pelo usuário.

Ao selecionar a opção de modelos pré-definidos, são carregadas diferentes plantas de apartamentos já mobiliadas com o indivíduo posicionado na porta de entrada do imóvel. Ape-

sar desses modelos estarem personalizados, eles podem ser editados conforme a necessidade do usuário.

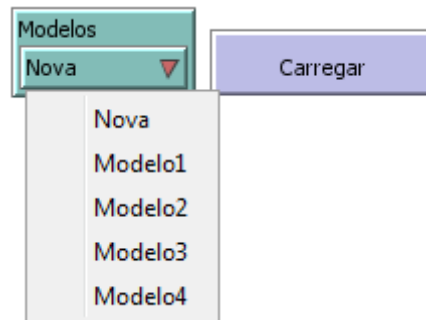


Figura 4.6: Carregar Modelos. Fonte: O Autor.

Após o carregamento de um dos modelos pré-configurados a simulação poderá ser iniciada, porém não será registrado nada do que estará acontecendo. Dessa forma, torna-se necessário o posicionamento dos sensores no apartamento, onde sempre que houver movimentação próxima a um desses dispositivos, ocorrerá o armazenamento dessa informação, caracterizada por data e hora da ação. Esse é um ponto chave da simulação porque é a partir dos sensores que todo o comportamento do morador será armazenado. Sendo assim, é importante que sejam escolhidos locais estratégicos pois os dados gerados por esses sensores deverão ser utilizados futuramente para entendimento e análise dos padrões gerados pelos hábitos do morador. A ferramenta auxilia na escolha desses locais pois ela além de permitir a visualização dos lugares onde o indivíduo costuma frequentar, também permite a fácil adição e remoção de sensores.

A Figura 4.7 ilustra os botões responsáveis pela manipulação dos sensores. Uma vez clicado o botão *Selecionar Posição Sensor*, pode-se marcar os locais aonde ficarão os sensores. Para isso, deve-se clicar com o botão esquerdo do mouse na posição desejada da planta que automaticamente os sensores aparecerão. Já o botão *Desfazer* é responsável por remover os sensores do último ao primeiro, em ordem decrescente, e por essa razão, sugere-se que seja marcado um novo sensor somente após ter a certeza do local do seu antecessor. Por último há ainda o interruptor *SalvarNoBanco?* que permite gravar na base de dados informações sobre a movimentação do agente Humano.



Figura 4.7: Posicionamento dos Sensores. Fonte: O Autor.

Caso a opção do usuário não seja carregar um modelo pré-definido, deve-se então escolher a opção de carregar um novo modelo, a ser desenvolvido pelo mesmo (Figura 4.6). A

primeira tarefa na elaboração de um modelo novo é a construção das paredes do imóvel. Para isso, a simulação tem disponíveis dois botões: o de *Desenhar Planta* e o de *Apagar Planta*. Ao clicar no primeiro, pode-se desenhar qualquer forma geométrica, embora exija habilidade com o mouse. O segundo serve para apagar os traços criados pelo primeiro botão. Nesse momento é importante que sejam deixados espaços de acesso entre os cômodos, isso porque o agente que representa o ser humano não consegue passar pelas paredes, ou seja, ele só consegue se movimentar em locais onde não hajam traços. Nesses espaços deixados entre os cômodos deverão ser colocadas as portas.



Figura 4.8: Posicionamento das Portas. Fonte: O Autor.

A Figura 4.8, mostra três botões que possibilitam a manipulação das portas. O botão *Selecionar Posição Porta* deverá ser selecionado quando o objetivo for preencher algum espaço da planta com uma porta. No botão seguinte, *Girar Porta*, a cada clique, faz-se uma rotação de 10 à direita, sempre na última porta inserida. E por último o botão *Desfazer* que apaga sempre em ordem decrescente as portas.

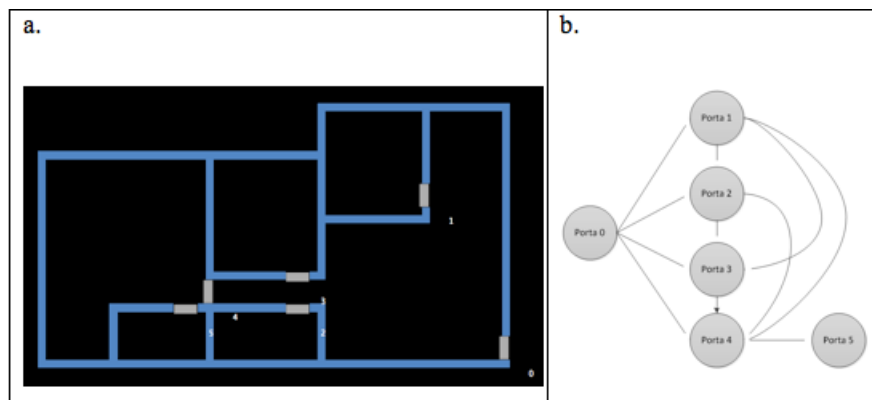


Figura 4.9: (a) Planta com Portas Posicionadas (b) Grafo Representativo das Portas Fonte: O Autor.

No exemplo apresentado na Figura 4.9, é possível notar que as portas de um imóvel, pintadas de cinza (Figura 4.9a), podem ser representadas através de grafos, onde as portas representam os vértices e a conexão entre elas formam as arestas (Figura 4.9b). Sendo assim, é necessário que esse grafo de portas seja representado computacionalmente na simulação Anjo da Guarda para que seja possível o funcionamento do algoritmo de mobilidade do morador entre os cômodos. Para isso, é utilizada a matriz de adjacência como forma de representação.

Uma matriz de adjacência é uma matriz $n \times n$, onde n é o número de vértices contidos no grafo a ser representado. Os elementos dessa matriz variam de acordo com as propriedades do grafo, porém de forma geral o valor a_{ij} corresponde à forma como um vértice v_i se relaciona com o vértice v_j . No caso dos grafos representativos das portas de um imóvel, eles são simples, não direcionados e sem peso nas arestas. Com isso, para sua representação como matriz, basta adicionar 1 no elemento a_{ij} quando houver adjacência entre o vértice v_i e v_j , ou seja ligação direta entre as portas p_i e p_j e 0 caso contrário. A Tabela 4.1 mostra a matriz de adjacência referente ao grafo de portas mostrado como exemplo na Figura 4.9.

Tabela 4.1: Matriz de Adjacência Fonte: O Autor.

	PORTA 0	PORTA 1	PORTA 2	PORTA 3	PORTA 4	PORTA 5
PORTA 0	0	1	1	1	1	0
PORTA 1	1	0	1	1	1	0
PORTA 2	1	1	0	1	1	0
PORTA 3	1	1	1	0	1	0
PORTA 4	1	1	1	1	0	1
PORTA 5	0	0	0	0	1	0

Na ferramenta de simulação, após completadas as fases de *Desenhar Planta* e *Posicionar Portas* deve-se dar entrada nos valores referentes à matriz de adjacência do grafo de portas. Para realização dessa atividade basta colocar o número das portas que possui adjacência nos componentes *Porta_A* e *Porta_B* e clicar no botão inserir (Figura 4.10). É importante salientar que os números das portas aparecem embaixo das mesmas e que, pelo fato do grafo não ser direcionado, uma vez colocado que a porta p_i possui conexão direta com a porta p_j , não é necessário colocar que p_j , se liga a p_i .

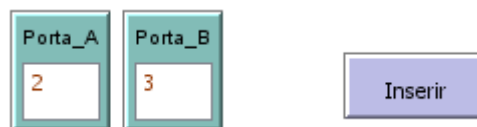


Figura 4.10: Componentes de Inserção da Matriz de Adjacência. Fonte: O Autor.

Obedecendo a sequência do fluxograma (Figura 4.2), é necessário agora que o imóvel seja mobiliado para que o morador consiga realizar as suas atividades rotineiras. A ferramenta possui seis artefatos para o cumprimento dessa tarefa. Com o *combobox Móvel*, é possível escolher qual dos móveis existentes se deseja adicionar na planta. O botão *Selecionar Posição Móvel* deverá ser selecionado para escolha da localização, contudo antes de utilizá-lo é importante que as entradas *PortaAcessoMovel* e *LadoPorta* estejam configuradas. Nesses campos deverá conter uma lista, separada por vírgula (,), dos números das

portas que fazem parte do cômodo no qual o móvel será inserido (*PortaAcessoMovel*) e das posições referentes a cada uma dessas portas (cima, baixo, esquerda ou direita) (*LadoPorta*).

Na Figura 4.11a é possível visualizar um móvel adicionado na planta elaborada na aplicação. Nota-se que as portas que dão acesso ao móvel citado são as portas 0,1,2,3 e 4, ou seja, exceto a porta 5, pois entre ela e o móvel existe a porta 4. Outra informação importante que se pode observar é que esse mesmo móvel está localizado no cômodo que fica à esquerda da porta 0, a direita da porta 1, em cima da porta 2, em baixo da porta 3 e a direita da porta 4. Essas informações deverão estar presentes no componente *LadoPorta* e são definidas da seguinte forma: *e* (esquerda), *d* (direita), *c* (cima) e *b* (baixo). A Figura 4.11b mostra os campos *PortaAcessoMovel* e *LadoPorta* preenchidos corretamente ao inserir a mesa na planta criada (Figura 4.11a).

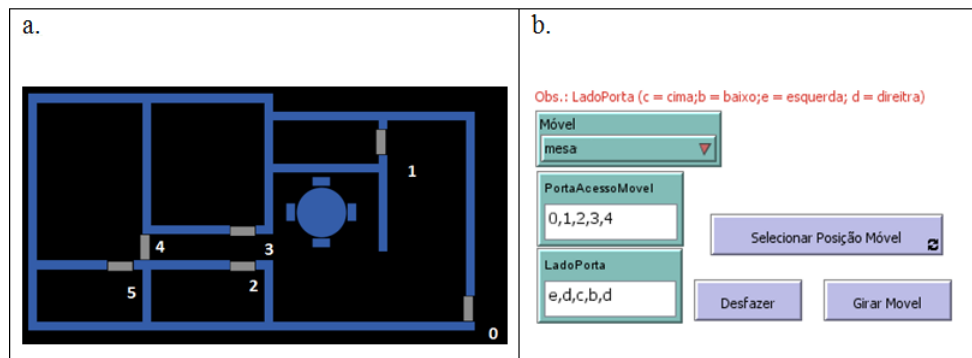


Figura 4.11: Adicionar Móveis no Imóvel. Fonte: O Autor.

Depois de inserido o objeto na planta, os botões *Desfazer* e *Girar Móvel* poderão ser utilizados. O primeiro botão apaga os móveis inseridos de forma decrescente e o segundo gira o último móvel inserido 10° à direita a cada clique, respectivamente.

Os sensores poderão agora ser adicionados no imóvel, seguindo instruções mencionadas anteriormente, e em seguida o morador deverá ser posicionado. Para o posicionamento do morador, assim como no posicionamento dos móveis, é necessário que seja inserido qual a porta mais próxima e em qual lado dela o idoso será fixado. Com isso são concluídas as fases de configuração da aplicação. Sendo assim, a simulação poderá ser iniciada e o comportamento do morador monitorado.

O algoritmo para movimentação do morador, apresentado no Capítulo 3, funciona da seguinte forma: toda vez que o calendário da ferramenta chega às 00:00 horas é gerado, seguindo uma distribuição pré-fixada, um novo conjunto de horários para as atividades diárias levantadas no questionário. Cada atividade do questionário possui um conjunto de subatividades e para cada subatividade possui um móvel vinculado, por exemplo, a

atividade ir ao banheiro possui como subatividades ir para a privada (móvel privada) e ir para pia do banheiro (móvel pia). O agente que representa o morador possui uma memória da localização de todos os móveis e portas do imóvel e, toda vez que a calendário marca o horário de uma das atividades, o agente procura, através do algoritmo de Dijkstra (1959), qual a porta que dá acesso ao móvel necessário para realização dessa atividade e por fim se direciona ao referido móvel.

4.3 Resultados

Com intuito de avaliar o modelo proposto, foi realizado um trabalho experimental com usuários com idade igual ou superior a 60 anos e que residem só. Para isso, inicialmente um profissional que trabalha na área de saúde com graduação em fisioterapia foi treinado para que ele fosse capaz de aplicar o questionário desenvolvido em usuários que satisfizessem as especificidades da pesquisa. Como quesito importante para o treinamento deste profissional, destaca-se: o enfoque do questionário é sempre para as atividades realizadas dentro dos domicílios. Esta informação é de grande relevância, pois podem existir usuários, por exemplo, que não costumam almoçar em seus domicílios e por isso, a resposta correta a ser marcada no questionário é que ele não almoça. O fato do entrevistador ser um profissional da área de saúde foi uma escolha estratégica, tendo em vista o contato diário destes profissionais com pessoas idosas. Sendo assim, três usuários foram entrevistados, suas respostas foram colocadas no software Questionário e o arquivo de dados para cada usuário gerado.

O questionário está subdividido em seis tópicos, sendo eles: *dados pessoais*, *dados referentes a atividade de dormir*, *dados referentes a refeições*, *dados referentes a entretenimentos*, *dados referentes a necessidades médicas e de higiene* e *dados referentes a lazer*. De cada um desses tópicos foi selecionada uma pergunta que servirá como base para posicionamento dos sensores. A Tabela 4.2 abaixo mostra as perguntas selecionadas e as repostas de cada idoso.

	IDOSO A DC 60 ANOS	IDOSO B DJ 86 ANOS	IDOSO C DV 90 ANOS
Qual hora costuma dormir de segunda a sexta-feira?	Entre 21:30 e 23:00	Entre 22:30 e 23:00	Entre 19:00 e 19:20
Qual hora você costuma almoçar?	Entre 12:00 e 12:30	Entre 12:00 e 12:30	Entre 12:30 e 13:00
Você costuma assistir TV? Se sim, em quais horários no dia?	Entre 08:30 e 09:00 Entre 12:00 e 13:00 Entre 20:00 e 20:40	Entre 08:00 e 09:00 Entre 12:00 e 12:30 Entre 18:00 e 18:30	Entre 12:00 e 13:00
Qual/Quais horário(s) você costuma tomar banho ao longo do dia?	Entre 06:20 e 06:50 Entre 11:00 e 12:00 Entre 16:00 e 16:30	Entre 07:00 e 07:30 Entre 22:00 e 22:30	Entre 15:00 e 16:00
Você costuma sair de casa diariamente? Se sim, em quais horários?	Entre 05:00 e 06:15 Entre 06:30 e 08:00	Entre 09:00 e 10:15 Entre 08:00 e 08:10 Entre 17:00 e 17:30	Entre 09:00 e 09:30

Tabela 4.2: Perguntas e Respostas do Questionário Fonte: O Autor.

Foi escolhida uma planta de apartamento preconfigurada pelo sistema que serviu como base para a simulação da rotina de todos os usuários. A escolha de uma mesma planta para realização de todas as simulações ocorreu pelo fato de que não foi possível ter acesso ao imóvel dos usuários entrevistados. Em decorrência disso, não foi possível verificar a quantidade e o posicionamento das portas, móveis e cômodos das residências, nem quais móveis existiam nelas. Além disso, o uso de diversas plantas diferentes, no caso desse trabalho experimental, não iria alterar os resultados, já que o mais importante é o posicionamento dos sensores.

O posicionamento dos sensores é o mais importante, no caso desse trabalho experimental, pois são eles os responsáveis por captar a movimentação do agente humano e enviar para o banco de dados o momento (data e hora) dessa movimentação, juntamente com a identificação de qual sensor foi acionado. Sendo assim, a sua posição é de extrema

importância, pois ele poderá ser inserido em locais onde o humano nunca passe. Nesse caso, o sensor vai ficar ocioso, sem captar nenhuma movimentação. Ou então, eles poderão ser inseridos em locais onde o humano realize diversas atividades, salvando no banco de dados ruídos, dificultando a análise dos dados. Em outras palavras, se um sensor for inserido na cama do usuário, com o intuito de saber qual horário ele vai dormir, se essa cama ficar muito próxima a porta do quarto, por exemplo, toda vez que o agente humano entrar no quarto, o sensor da cama vai captar como se ele tivesse deitado na cama. Nesse caso, para resolver esse problema, o sensor deverá ser posicionado em um local adequado para que capte apenas o morador indo deitar na cama.

Dessa forma, com o objetivo de confirmar as informações referentes a rotina do idoso, foram instalados sensores em locais estratégicos, a fim de acompanhar e ratificar na prática que a simulação foi capaz de representar as ações realizadas pelos usuários. Sendo assim, foram posicionados cinco sensores na planta, sendo eles colocados um no chuveiro, um na cama, um na porta de saída, um no sofá da sala e outro na mesa da sala. A Figura 4.12 ilustra a planta do apartamento de uma das simulações com os sensores posicionados. Os sensores são representados por um “x”.

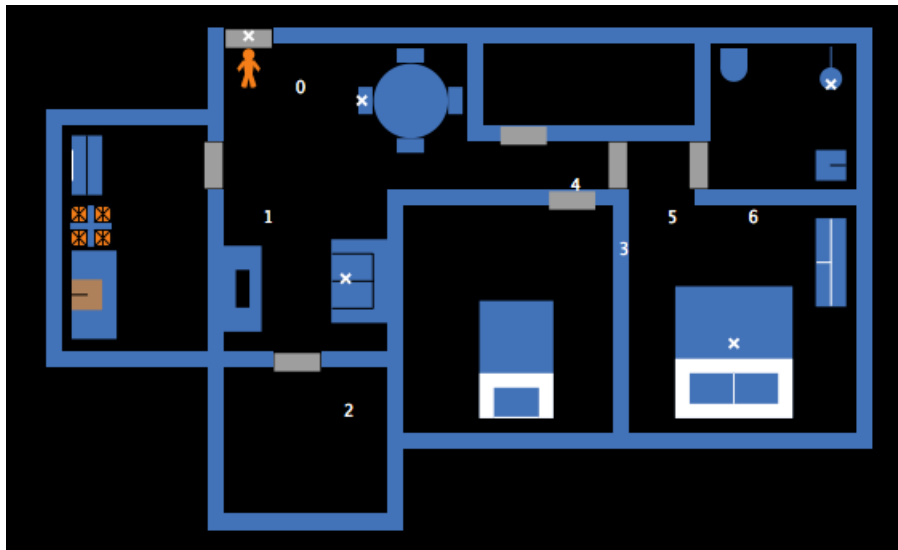


Figura 4.12: Apartamento com sensores posicionados. Fonte: O Autor.

Foram coletadas informações dos sensores referentes às atividades dos três idosos durante o período de um mês, do ponto vista do calendário da ferramenta e para isso foram necessárias duas horas e trinta minutos executando cada simulação. As informações geradas por cada sensor foram enviadas para o banco de dados. É importante salientar que para cada nova execução do aplicativo Anjo da Guarda, o banco de dados é zerado, evitando que sejam misturadas informações de duas simulações distintas. A partir disso foram gerados gráficos de frequência dos horários de acionamento de cada sensor no imóvel, para cada um dos idosos. Esses gráficos precisam ser consistentes com as respostas ob-

tidas através do questionário, comprovando que a simulação segue conforme informações colhidas previamente.

É importante ressaltar, que os gráficos de frequência desse trabalho foram obtidos através da ferramenta *R*, utilizando o método *bkde* com os parâmetros *bandwidth = 0.1* e *kernel = "normal"* para amaciamento de kernel. Esse método é não paramétrico e tem como objetivo a estimação de funções densidade. Cada observação é ponderada pela distância em relação a um valor central. Esse método foi escolhido pois ele é um interpolador que possibilita a estimação da intensidade dos acionamentos dos sensores. Sendo assim, o estimador de kernel é dado por:

$$f_h(x) = \frac{1}{(nh)} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (4.1)$$

Onde n é o estimador de Rosenblatt da função densidade, $h > 0$ é o parâmetro de alisamento e K é uma função real positiva satisfazendo:

$$\int_{-\infty}^{+\infty} K(t) dt = 1 \quad (4.2)$$

As Figuras 4.13, 4.14 e 4.15 representam os gráficos referentes ao sensor posicionado na cama do idoso. Esse sensor teve como propósito a confirmação de que os idosos realmente acionavam os sensores da cama no horário que foi dito como de dormir. Sendo assim, o idoso A respondeu que costuma dormir entre 21 e 23:00 horas. O gráfico da Figura 4.13 corrobora essa informação já que o maior pico de ativação do sensor cama está entre 21 e 23 horas. Porém, ainda na Figura 4.13 nota-se que existe frequência de acionamentos também entre 20:30 e 21 horas. Isso deve-se ao fato de que existe no questionário a pergunta: “A que horas costuma dormir no fim de semana (sábado e domingo)?”. Para essa pergunta o idoso A respondeu que costuma dormir entre 20:30 e 21:30 horas. Sendo assim, o gráfico corrobora também com essa informação fornecida pelo idoso A. Com intuito de auxiliar na análise dos gráficos, as respostas completas dos idosos entrevistados encontram-se no Apêndice C.

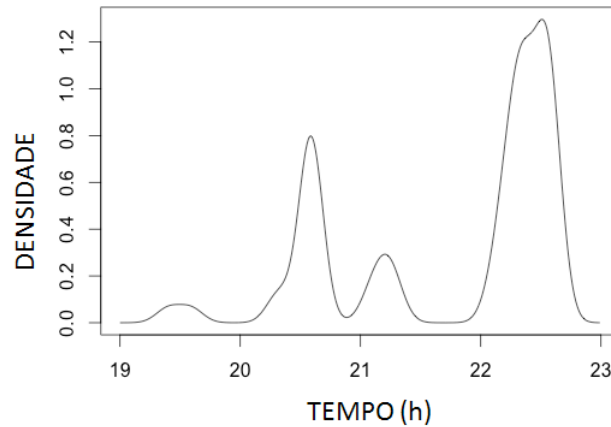


Figura 4.13: Sensor Cama Idoso A DC. Fonte: O Autor.

A Figura 4.14, que representa o sensor da cama do idoso B, também confirma o que foi informado previamente, mostrando que o idoso aciona esse sensor quase sempre entre 22 e 22:50 horas.

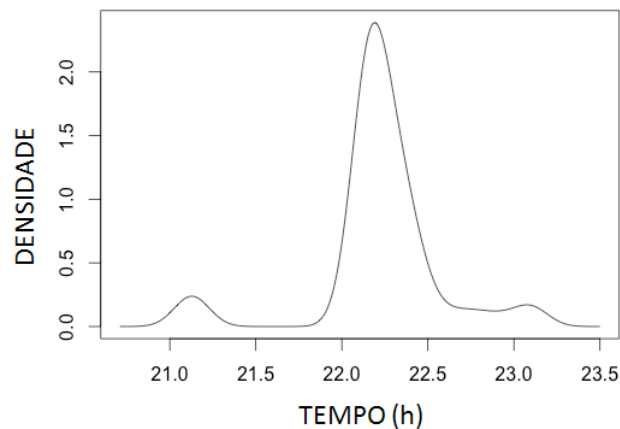


Figura 4.14: Sensor Cama Idoso B DJ. Fonte: O Autor.

Já o idoso C, informou que tem o hábito de dormir muito mais cedo do que os outros dois. Na Figura 4.15, pode-se observar justamente isso, diferente dos idosos A e B que costumam dormir em horários mais próximos à 22 horas a idoso C tem o sensor da cama acionado sempre entre 19 e 20 horas.

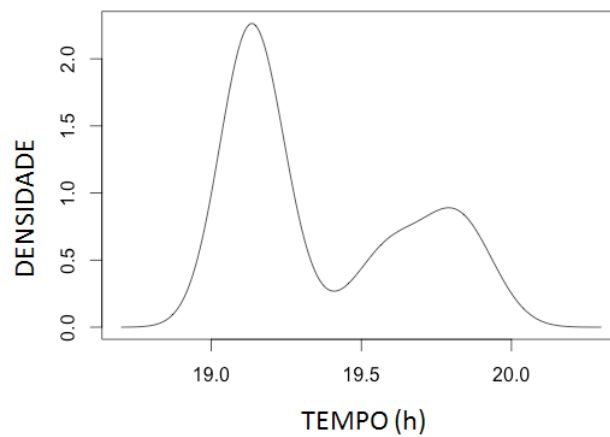


Figura 4.15: Sensor Cama Idoso C DV. Fonte: O Autor.

Em relação as refeições, os idosos A e B responderam que costumam almoçar entre 12 e 12:30 horas, enquanto que o C afirmou que costuma almoçar entre 12:30 e 13:00 horas. Esta informação pode ser visualizada e confirmada nos gráficos das Figuras 4.16, 4.17 e 4.18 que seguem abaixo. Eles mostram exatamente que o acionamento do sensor da mesa acontece nesse horário. Os picos de acionamento dos gráficos das Figuras 4.16 e 4.17 estão entre 12:00 e 12:50, sendo que o gráfico da Figura 4.17 apresentou ainda uma menor frequência de acionamentos entre 11:00 e 11:50. Isto deve estar associado a um posicionamento inadequado do sensor, pois atividades que não envolvem o ato de almoçar foram captadas, mostrando a importância do posicionamento destes.

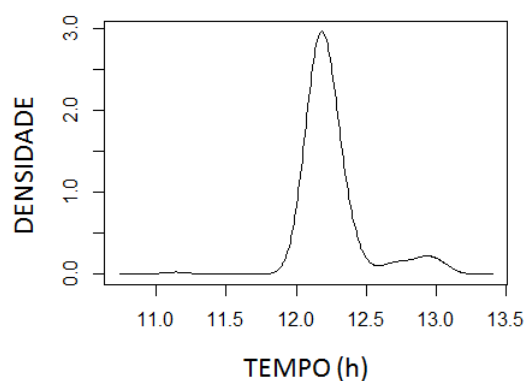


Figura 4.16: Sensor Mesa Idoso A DC. Fonte: O Autor.

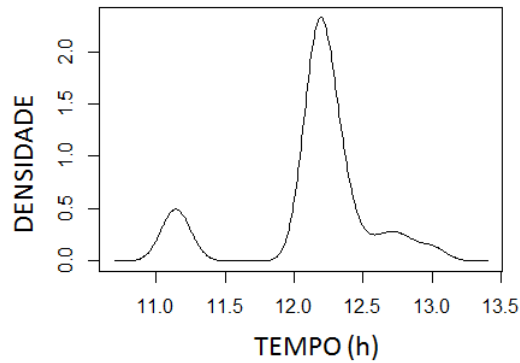


Figura 4.17: Sensor Mesa Idoso B DJ. Fonte: O Autor.

Pode-se observar no gráfico da Figura 4.18, que os picos de acionamento do sensor mesa ocorrem no intervalo de 12:50 e 13:00 horas, confirmando a informação coletada.

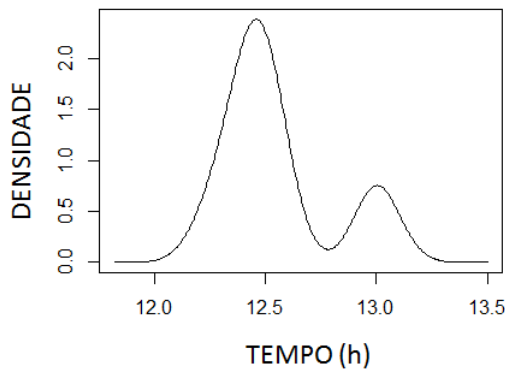


Figura 4.18: Sensor Mesa Idoso C DV. Fonte: O Autor.

Diante dos resultados visualizados nas Figuras 4.19, 4.20 e 4.21, pode-se concluir que a o sofá é um móvel bastante utilizado pelos três idosos, tanto para assistir televisão quanto para descanso. Desta maneira, esses gráficos são capazes de confirmar os dados coletados pelo questionário pois os maiores picos de acionamento deste sensor estão de acordo com os horários de uso ditos pelos idosos.

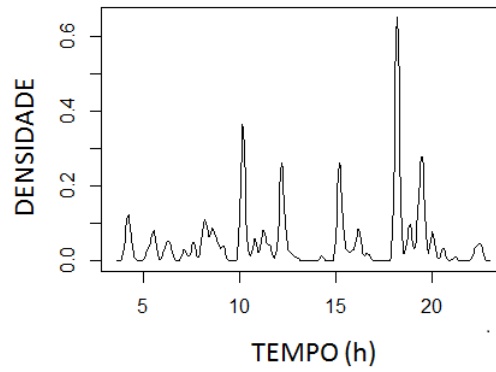


Figura 4.19: Sensor Sofá Idoso A DC. Fonte: O Autor.

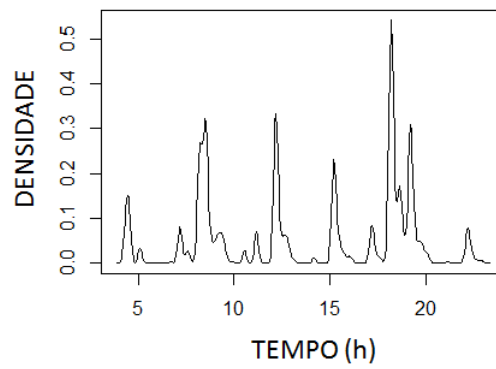


Figura 4.20: Sensor Sofá Idoso B DJ. Fonte: O Autor.

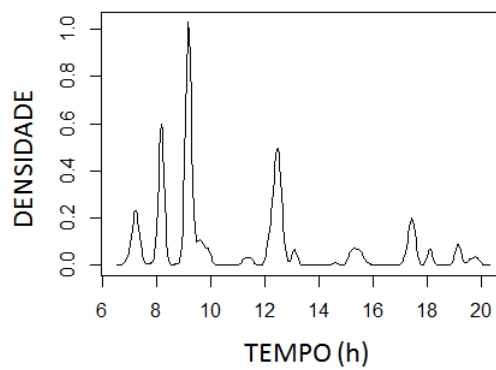


Figura 4.21: Sensor Sofá Idoso C DV. Fonte: O Autor.

Quanto ao acionamento do sensor colocado no box do banheiro, este indica total concordância de horário com as afirmações feitas pelos três idosos no questionário, principalmente no que se refere à frequência dos banhos, isso pode ser observado nas Figuras 4.22, 4.23 e 4.24. Na Figura 4.24 é possível observar também alguns acionamentos após 14:50, isso ocorre pois o algoritmo de geração dos horários das atividades permite horários próximos às faixas de horários respondidas pelos entrevistados. Este sensor pode ser visto como de extrema relevância pois devido a fragilidade do idoso, somada aos riscos gerados pelo box do banheiro que é um local molhado e escorregadio, aumenta a incidência de quedas.

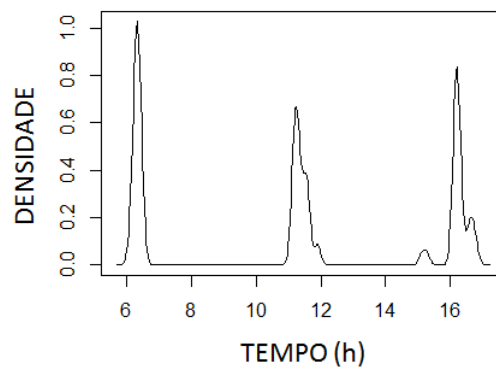


Figura 4.22: Sensor Chuveiro Idoso A DC. Fonte: O Autor.

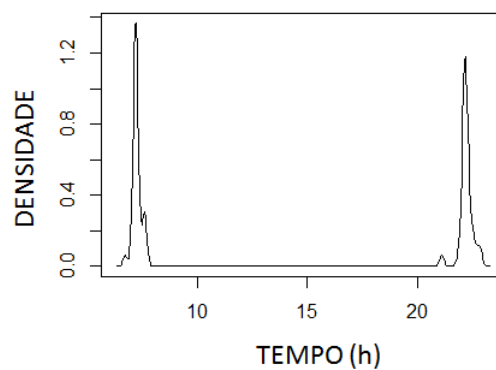


Figura 4.23: Sensor Chuveiro Idoso B DJ. Fonte: O Autor.

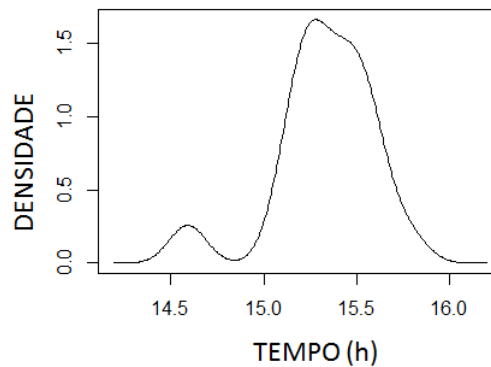


Figura 4.24: Sensor Chuveiro Idoso C DV. Fonte: O Autor.

Em resposta ao questionário, o idoso A afirmou que sai de casa entre 5:15 e 06:30 e 6:30 e 8:30 horas, o que pode ser visualizado no gráfico da Figura 4.25, que mostra o pico de acionamento sensor porta. Esse sensor é ativado toda vez que o indivíduo sai ou entra em casa. Este comportamento também pode ser observado nos gráficos das Figuras 4.26 e 4.27, que estão de acordo com as respostas dos idosos B e C. Porém, no gráfico da Figura 4.27 nota-se captação pelo sensor no horário entre 12 e 13:00 hora, que não está de acordo com a resposta do idoso C, o que também deve estar atrelado a um posicionamento inadequado do sensor. Ele captou atividades não associadas a ação de sair ou entrar em casa, mostrando outra vez a importância de um posicionamento correto.

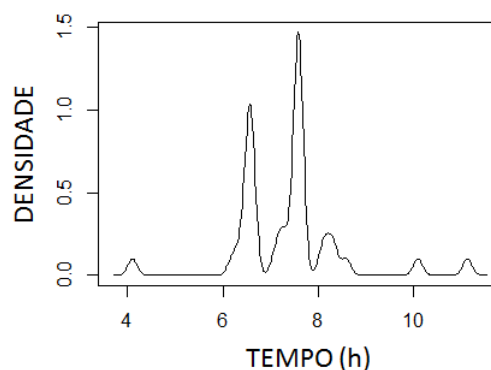


Figura 4.25: Sensor Porta Idoso A DC. Fonte: O Autor.

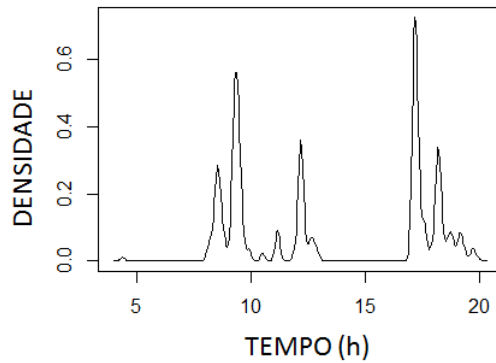


Figura 4.26: Sensor Porta Idoso B DJ. Fonte: O Autor.

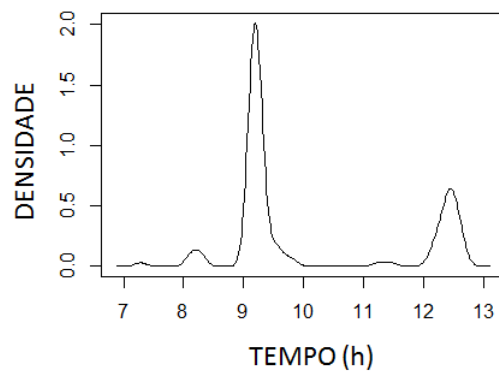


Figura 4.27: Sensor Porta Idoso C DV. Fonte: O Autor.

Desta forma, nota-se que a simulação foi capaz de gerar um padrão de comportamento com base nos questionários aplicados. Mostrando assim, que a proposta da simulação foi alcançada de forma eficaz. Sendo assim, conclui-se que os dados gerados através dessa pesquisa de mestrado podem ser utilizados para análise de algoritmos de reconhecimento de padrão.

Considerações finais

Em estudos realizados durante o desenvolvimento desse projeto de mestrado foi possível notar que existe, nas últimas décadas, um aumento do número de pessoas que residem sozinhas tanto no Brasil quanto no Mundo, devido à uma nova tendência de vida independente. Foi possível observar também a quantidade de acidentes nesses lares, levando assim a uma necessidade de atenção especial para essa realidade.

5.1 Conclusões

O modelo proposto neste trabalho possui como conceito básico a utilização de agentes para simulação dos padrões de comportamento de indivíduos, principalmente idosos, que residem sozinhos, através da aplicação de um questionário desenvolvido. Esse questionário possui perguntas que tem como objetivo principal conhecer o cotidiano de pessoas dentro das suas residências. Além disso, esse modelo contempla a possibilidade de construção de diferentes plantas de imóveis, a adição de móveis nessas plantas e o posicionamento de sensores, que são responsáveis por captar toda a movimentação, próxima a eles, dos moradores para execução das suas atividades vitais nos seus lares e salvar em um banco de dados.

Para validação do modelo, foi realizado um trabalho experimental onde o questionário desenvolvido nessa pesquisa foi aplicado em três pessoas com idade igual ou maior do que 60 anos e que moram em residências unipessoais. Após isso, as repostas de cada idoso foram colocadas no software Questionário e um arquivo no formato txt correspondente às informações de cada entrevistado foi salvo separadamente. Em seguida, para cada arquivo representativo das repostas de um idoso foi gerada uma simulação onde uma planta de imóvel pré-configurada pela ferramenta foi selecionada e sensores posicionados com o objetivo de verificar se os dados captados por eles condizem realmente com o que foi respondido por cada idoso.

Através dos resultados obtidos nesse estudo, pôde-se provar que é possível gerar padrões de comportamentos capazes de representar a forma de vida de indivíduos que residem sós, através de uma simulação baseada em agentes integrada com as repostas de um questionário facilmente aplicável.

Para simular o cotidiano dos três idosos durante o período de um mês, foram necessárias sete horas e meia de execução da simulação. Além disso, devido à considerável quanti-

dade de informações geradas no banco de dados, poderão haver instabilidade e paradas na ferramenta de simulação em decorrência das transações executadas no banco de dados. Outra informação importante é que quanto maior o tempo de execução, maior a necessidade de espaço livre em disco.

5.2 Contribuições

Nas pesquisas efetuadas nesta dissertação de mestrado não foram encontrados trabalhos correlatos que propusessem modelos computacionais com o objetivo de simular seres humanos realizando as suas atividades diárias dentro dos seus lares.

Os dados gerados nessa pesquisa servem como base para aplicação de diferentes algoritmos para reconhecimento de padrão e, com isso, tornar possível a análise de qual desses algoritmos obtém o melhor resultado, ou seja, escolher o melhor algoritmo capaz de aprender o comportamento diário dos usuários entrevistados. Através da escolha de um algoritmo eficiente e com a montagem de um sistema formado por sensores, redes de comunicação e computador nos lares unipessoais, será possível perceber antecipadamente comportamentos considerados fora dos padrões habituais e assim contatar, seja por envio de email, SMS ou outro meio qualquer, profissionais de saúde, familiares e/ou amigos, evitando que eventuais problemas se agravem.

Outra contribuição importante, é que a partir dessa ferramenta de simulação é possível também criar réplicas de plantas reais e, com isso, observar quais locais do imóvel são mais utilizados pelo morador e, dessa forma, escolher pontos estratégicos para o posicionamento de sensores na residência real.

5.3 Atividades Futuras de Pesquisa

Uma das propostas para atividades futuras desse trabalho é a simulação de enfermidades no cotidiano dos moradores monitorados, ou seja, nessa pesquisa o foco do questionário e da simulação foi em representar o cotidiano dos moradores nas suas casas mas sem nenhum tipo de patologia. Sendo assim, é extremamente relevante esse tipo de simulação pois com ela será possível testar previamente os algoritmos de reconhecimento de padrões para detecção de anormalidades.

Outra importante proposta de atividade futura é contemplar na simulação a visita de outras pessoas nas residências como familiares, amigos, ajudantes do lar entre outros.

Questionário



SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL E TECNOLOGIA INDUSTRIAL

Este questionário foi elaborado com o objetivo de identificar as atividades rotineiras de uma pessoa em seu ambiente domiciliar.

Data do preenchimento do questionário: ___/___/___ Horário: ___:___

1. Dados pessoais:

1.1 Sexo: Masculino () Feminino ()

1.2 Idade: _____

1.3 Cidade onde mora: _____

2. Dados referentes a atividade de dormir.

2.1 A que horas costuma dormir de segunda à sexta-feira?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado dorme, normalmente, durante a semana.

Entre _____ e _____

2.2 A que horas costuma dormir no fim de semana (sábado e domingo)?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado dorme, normalmente, durante finais de semana e feriado.

Entre _____ e _____

2.3 A que horas costuma acordar de segunda à sexta-feira?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado acorda, normalmente, durante a semana.

Entre _____ e _____

2.4 A que horas costuma acordar no fim de semana (sábado e domingo)?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado acorda, normalmente, durante finais de semana e feriado.

Entre _____ e _____

2.5 Quantas vezes costuma levantar durante o ato de dormir para ir ao banheiro?

Descrição: Essa pergunta visa identificar uma média da quantidade de vezes que o entrevistado, normalmente, costuma ir ao banheiro durante a noite.

2.6 Quantas vezes costuma levantar durante o ato de dormir para tomar remédio?

Descrição: Essa pergunta visa identificar uma média da quantidade de vezes que o entrevistado, normalmente, costuma tomar remédio(s) durante a noite.

2.7 Quantas vezes costuma levantar durante o ato de dormir para beber ou comer algo?

Descrição: Essa pergunta visa identificar uma média da quantidade de vezes que o entrevistado, normalmente, costuma levantar para comer ou beber algo durante a noite.

2.8 Em quais horários costuma tirar cochilos ao longo do dia?

Descrição: Essa pergunta visa identificar a quantidade de vezes e o(s) horário(s) que o entrevistado, normalmente, costuma tirar cochilo(s) ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre _____ e _____

2.10 Caso tenha respondido de forma afirmativa na questão 2.8, informar o local onde costuma tirar cochilos ao longo do dia.

Descrição: Essa pergunta visa identificar em qual móvel o entrevistado, normalmente, costuma tirar o(s) cochilo(s), se por ventura tenha respondido positivamente à pergunta 2.9.

Cama de Casal

Cama de Solteiro

Sofá

3. Dados referentes às refeições.

3.1. A que horas costuma preparar o café da manhã?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma preparar o café manhã. Essa pergunta refere-se ao preparo do café da manhã dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma preparar o café da manha fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.2. A que horas costuma tomar o café da manhã?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma tomar o café manhã. Essa pergunta refere-se à ação de tomar o café da manhã dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma tomar o café da manha fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.3. A que horas costuma lanchar no turno da manhã?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma lanchar no turno da manhã. Essa pergunta refere-se à ação de lanchar dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma lanchar fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.4. A que horas costuma preparar o almoço?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma preparar o almoço. Essa pergunta refere-se à ação de preparar o almoço dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele preparar o almoço fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.5. A que horas costuma almoçar?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma almoçar. Essa pergunta refere-se à ação de almoçar dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma almoçar fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.6. A que horas costuma lanchar no turno da tarde?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma lanchar no turno da tarde. Essa pergunta refere-se à ação de lanchar dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma lanchar fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre _____ e _____

3.7. A que horas costuma preparar o jantar/café da noite?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma preparar o jantar / café da noite. Essa pergunta refere-se à ação de preparar o jantar / café da noite dentro da residência. Se o entrevistado, não possuir esse

hábito ou se ele preparar o jantar / café da noite fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre e

3.8. A que horas costuma jantar/café da noite?

Descrição: Essa pergunta visa identificar a faixa de horário que o entrevistado, normalmente, costuma jantar. Essa pergunta refere-se à ação de jantar dentro da residência. Se o entrevistado, não possuir esse hábito ou se ele costuma jantar fora da sua residência, essa pergunta deverá ser deixada em branco.

Entre e

4. Dados referentes a entretenimento.

4.1 Em quais horários costuma assistir televisão?

Descrição: Essa pergunta visa identificar a quantidade de vezes e o(s) horário(s) que o entrevistado, normalmente, costuma assistir televisão em sua residência ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre e

Entre e

Entre e

4.2 Em quais horários costuma de ler jornal?

Descrição: Essa pergunta visa identificar a quantidade de vezes e o(s) horário(s) que o entrevistado, normalmente, costuma ler jornal em sua residência ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre e

Entre e

5. Dados referentes à necessidades médicas e de higiene.

5.1 Quais os horários que costuma tomar banho ao longo do dia?

Descrição: Essa pergunta visa identificar a quantidade de vezes e o(s) horário(s) que o entrevistado, normalmente, costuma tomar banho em sua residência ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre e

Entre e

Entre e

5.3 Quais os horários que costuma tomar medicamento ao longo do dia?

Descrição: Essa pergunta visa identificar a quantidade de vezes e o(s) horário(s) que o entrevistado, normalmente, costuma tomar medicamento(s) em sua residência ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre e

Entre e

Entre e

Entre e

6. Dados referentes à necessidades médicas e de higiene.

6.1 Em quais horários e dia da semana você costuma sair de casa?

Descrição: Essa pergunta visa identificar a quantidade de vezes, o(s) dia(s) da semana e o(s) horário(s) que o entrevistado, normalmente, costuma sair de casa ao longo do dia. Se o entrevistado não possuir este hábito a resposta deverá ficar em branco.

Entre e Dia da Semana:

Entre e Dia da Semana:

Entre e Dia da Semana:

OBRIGADO (A) PELA PARTICIPAÇÃO E COLABORAÇÃO.

NetLogo

NetLogo é uma linguagem de programação baseada em sistemas multiagentes e um ambiente favorável à modelagem de fenômenos naturais e sistemas complexos. Ela se adapta muito bem a modelagem de sistemas complexos que evoluem ao longo do tempo, sendo projetada para ser utilizada tanto na educação quanto na pesquisa.

Os usuários dessa ferramenta conseguem controlar milhares de agentes de forma independente, todos operando simultaneamente. Além disso, ela possui uma biblioteca com modelos prontos, editáveis e que podem servir para ambientação de novos usuários. Ela traz também o princípio de “baixo limiar”, ou seja, até mesmo pessoas que não possuem experiência em programação conseguem desenvolver um modelo.

Essa ferramenta foi desenvolvida em Java com o propósito de ser executada nos principais sistemas operacionais disponíveis. Outras vantagens importantes são: ela é gratuita; pode ser utilizada sem nenhuma limitação, ou seja, não possui nenhuma funcionalidade bloqueada para construção dos modelos; e possui uma grande quantidade de tutoriais e documentações disponíveis.

B.1 Histórico

A ferramenta Netlogo é de autoria de Uri Wilensky e foi desenvolvida no CCL (Centro de Aprendizagem Conectada). Ela é originada a partir da combinação das ferramentas StarLisp (LASSER C.; OMOHUNDRO, 1996) e Logo (PAPERT, 1980). Da ferramenta Logo foi aproveitado a facilidade de uso e o conceito de turtle (tartaruga), cuja a principal diferença é que no NetLogo o programador passa a possuir o controle sobre mais do que uma tartaruga. Já da ferramenta StarLisp, foram incorporados os conceitos de agentes e de concorrência.

O processo de desenvolvimento dessa ferramenta foi iniciado em 1999 e, a partir dessa data, a cada ano eram lançados de duas a três versões, em média. Mas foi em agosto de 2004 que foi lançada a versão 2.0.2, considerada como uma versão madura, confiável e estável.

Segundo Tisue e Wilensky (2004), existe uma série de evidências que mostram a crescente e ampla aceitação do NetLogo nas comunidades de pesquisa e educação. Apesar da

quantidade de usuários ter crescido bastante ao longo dos anos, a quantidade de relatórios de erros foi diminuindo e a velocidade de execução dos modelos foi aumentando a cada nova versão da ferramenta.

B.2 O Ambiente Netlogo

O ambiente de desenvolvimento NetLogo é formado por agentes móveis chamados de *turtles*, que são capazes de se movimentar no ambiente. Este, por sua vez, é formado por pequenos pedaços quadrados, chamados de *patches*, que também são agentes programáveis. Outro tipo de agente existente é o *observer* (observador). Existe apenas um observador no ambiente e ele é responsável por criar, supervisionar e controlar os demais elementos. Além desses três tipos de agente, o Netlogo possui ainda um quarto, que é chamado de *link*. Esse último agente tem como finalidade ligar outros dois agentes do tipo *turtle*. Sendo assim, se uma das duas tartarugas ligadas por um agente *link* morre esse agente morrerá também. Todos os agentes podem interagir uns com os outros e são capazes de realizar várias tarefas independentes e paralelas.

Para iniciar a utilização do NetLogo, é interessante que seja determinado, de acordo com os propósitos do modelo, o tamanho do ambiente no qual os agentes *turtles*, *patches* e *links* estarão inseridos. O NetLogo é dividido em três telas distintas com o objetivo de facilitar o desenvolvimento de modelos, são elas: *interface*, *info* e *code*.

A interface NetLogo é onde a simulação é visualizada e permite duas formas de interação, criação e controle dos modelos. A primeira maneira é através do *command center* (centro de comandos). Através dessa ferramenta, os usuários podem executar comandos de criação ou de controle a qualquer momento, até mesmo em tempo de execução das simulações. Ele é normalmente utilizado para criação de modelos simples ou para depuração de modelos mais complexos. Nele é possível que instruções sejam realizadas para qualquer um dos quatro tipos de agentes existentes. Para isso, é necessário que o tipo de agente seja escolhido no canto inferior esquerdo da janela do NetLogo (Figura B.1). Quando o agente escolhido é o *observer*, os comandos são executados no contexto global do sistema. Já quando o agente escolhido é o *link*, *turtle* ou *patch* os comandos são executados somente no contexto dos seus respectivos agentes.

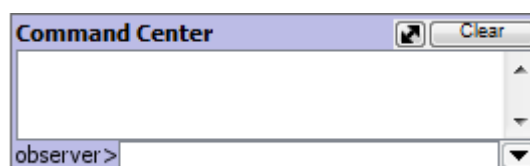


Figura B.1: Centro de Comandos do NetLogo. Fonte: O autor.

A segunda forma de interação com o ambiente NetLogo é através da interface. Conforme pode ser observado na figura B.2, ela é dividida em três partes: a azul, a verde e a bege. A parte azul são os botões de controle, ou seja, são botões responsáveis por preparar, iniciar ou parar os modelos. Esses botões têm como objetivo facilitar a execução automática de comandos e podem ser divididos em dois tipos: *apenas uma vez*, que é quando os botões são utilizados para realizar uma determinada ação apenas uma vez após ser pressionado e *para sempre*, que é quando, ao ser clicado, ações são realizadas repetidamente.

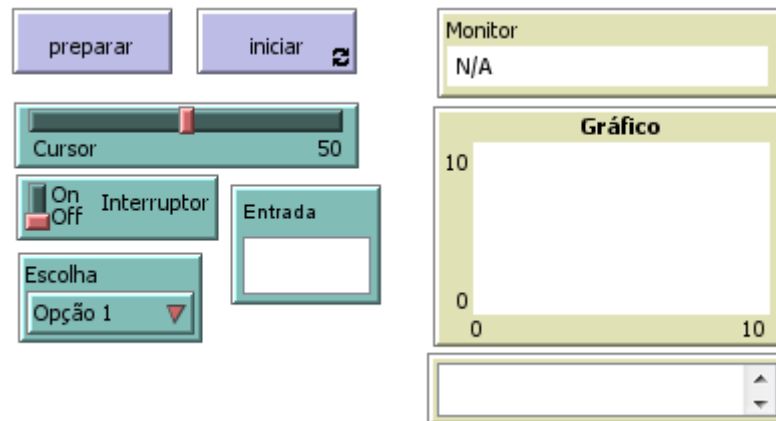


Figura B.2: Interface de Controle do NetLogo. Fonte: O autor.

Já os botões verdes são de configuração e são responsáveis por alterar os parâmetros dos modelos. Eles têm como finalidade testar variados cenários e hipóteses. Através deles é possível mudar variáveis e verificar de que forma elas influenciam no modelo, oferecendo assim a possibilidade de uma maior compreensão sobre os fenômenos modelados. Eles podem ser subdivididos em botões *cursor*, *interruptor*, *escolha* e *entrada* (Figura B.2). O botão *cursor* representa uma variável global que pode ser utilizada pelos agentes existentes no modelo. Com ele é possível determinar um intervalo de números reais e a forma de incremento dos elementos desse intervalo. Já o botão do tipo *interruptor*, representa visualmente uma variável booleana global que também pode ser acessada pelos agentes. O botão verde *escolha* é uma variável global que mostra aos usuários uma lista de opções das quais somente uma pode ser selecionada. Já o botão *entrada* permite que sejam inseridos valores do tipo numérico, alfanumérico ou do tipo cor, sendo que nesse último, o NetLogo mostra as opções de cores existentes no ambiente.

O último conjunto de botões da interface é o grupo de botões de cor bege. Estes botões têm como finalidade a apresentação de resultados, já que um dos propósitos da simulação é a obtenção de resultados, que são muitas vezes difíceis de serem alcançados em um laboratório. O NetLogo oferece três maneiras para demonstração dos resultados, são elas: *monitor*, *gráfico* e *saída*. O botão *saída* só pode ser adicionado uma vez por modelo e geralmente é utilizado para armazenar o *log* do sistema. Além de saída, existe também o botão *monitor*, que mostra em *tempo real* o valor de uma variável e o botão *gráfico*, que

permite a verificação do relacionamento entre duas variáveis do modelo.

O ambiente Netlogo, além da tela de *interface*, contém ainda a tela de *info* e a *code*. *Info* é a tela onde uma documentação sobre o modelo pode ser encontrada. Essa tela é utilizada para explicar as principais regras do modelo e ainda sugerir alguns experimentos. Já a tela *code*, é onde o código real do modelo é escrito na linguagem NetLogo.

B.3 A Linguagem NetLogo

A linguagem Netlogo é uma linguagem de programação estruturada e baseada em procedimentos. Os procedimentos são trechos de códigos que têm como objetivo a resolução de um problema específico. Existem dois tipos no NetLogo, o *to* e o *to-report*. A diferença entre eles é que o *to-report* no final da execução, retorna um valor para o ambiente enquanto que o *to* não retorna nada.

As variáveis existentes são divididas em: *globais*, *variáveis agentes* ou *locais*. As variáveis *globais* são acessíveis a todos os procedimentos e agentes existentes no modelo desenvolvido, enquanto que as variáveis *locais* podem ser utilizadas apenas no escopo do procedimento no qual elas foram criadas. As variáveis *agentes* são subdivididas em *turtle*, *patch* e *link* e servem para o armazenamento de valores dentro dos agentes. Neste tipo de variável cada agente possui o seu próprio valor, ou seja, cada novo agente do tipo *turtle* possui valores específicos para as variáveis do tipo *turtle*, por exemplo.

Outra funcionalidade importante do NetLogo é a possibilidade de criação de *breeds* (espécies) para os agentes. Com *breeds* torna-se possível a criação de propriedades e comportamentos diferenciados para agentes. Além disso, é possível também a criação de grupos de agentes de um mesmo tipo através da funcionalidade chamada *Agenteset*. Ela é extremamente importante, pois permite a formação de pseudogrupos com características específicas em comum.

Na área de programação, os comentários contribuem bastante para o entendimento e organização dos códigos fonte. Para a adição de comentários no Netlogo basta apenas iniciar a linha com “ ; ”.

B.4 Biblioteca de Modelos

A equipe de desenvolvimento do NetLogo investe quase o mesmo esforço no aprimoramento da ferramenta quanto no desenvolvimento de novos modelos para a biblioteca interna do

ambiente (TISUE; WILENSKY, 2004). Essa biblioteca já vem inclusa na ferramenta ou pode ser executada em um navegador de internet no site da comunidade NetLogo.

A biblioteca de modelos do NetLogo possui centenas de simulações que podem ser exploradas e editadas. Elas são bem diversificadas e abordam diversas áreas de conteúdo nas ciências naturais e sociais, como por exemplo, na biologia, medicina, psicologia social, matemática e ciência da computação.

Os modelos disponíveis são muito úteis para a ambientação de novos usuários já que os códigos das simulações são claros, bem comentados e possuem uma documentação detalhada contendo explicações sobre o assunto da simulação, as regras, sugestões de atividades e experiências, além de possíveis extensões.

B.5 Extensões

Anteriormente, nas primeiras versões do NetLogo, era utilizado um conceito que pode ser chamado de “ tudo em um ”, ou seja, todos os pacotes existentes vinham amarrados e integrados no mesmo ambiente. Porém, segundo Tisue e Wilensky (2004), este tipo de abordagem pode se transformar em “ tudo ou nada ”, pois corre-se o risco de que, se um componente não atender as necessidades de um usuário, esse usuário não será capaz de usar nenhum componente, porque estavam todos inter-relacionados.

Sendo assim, para evitar esse tipo de problema as novas versões passaram a permitir que os usuários importassem ou substituíssem as partes do NetLogo que não atendiam às suas necessidades. Essa solução permite que os usuário desenvolvam, na linguagem Java, o que eles necessitarem ou que busquem na comunidade soluções já prontas para auxiliar no desenvolvimento dos seus modelos.

Respostas do Questionário

Tabela C.1: Respostas do Questionário Aplicado. Fonte: O autor

	IDOSO A	IDOSO B	IDOSO C
Qual hora costuma dormir de segunda a sexta-feira?	Entre 21:30 e 23:00	Entre 22:30 e 23:00	Entre 19:00 e 19:20
A que horas costuma dormir no fim de semana (sábado e domingo)?	Entre 20:30 e 21:30	Entre 22:30 e 23:00	Entre 19:00 e 19:20
A que horas costuma acordar de segunda à sexta-feira?	Entre 04:00 e 04:30	Entre 04:30 e 05:00	Entre 07:00 e 07:30
A que horas costuma acordar no fim de semana (sábado e domingo)?	Entre 05:40 e 06:00	Entre 04:30 e 05:00	Entre 07:00 e 07:30
Quantas vezes costuma levantar durante o ato de dormir para ir ao banheiro?	1	2	3
Quantas vezes costuma levantar durante o ato de dormir para tomar remédio?	0	0	0
Quantas vezes costuma levantar durante o ato de dormir para beber ou comer algo?	0	0	0
Em quais horários costuma tirar cochilos ao longo do dia?	Entre 13:15 e 13:30	Entre 12:30 e 13:00	Entre 14:00 e 15:00
A que horas costuma preparar o café da manhã?	Entre 07:00 e 08:00	Entre 08:00 e 08:30	Entre 08:00 e 08:10

A que horas costuma tomar o café da manhã?	Entre 08:00 e 08:15	Entre 08:30 e 09:00	Entre 08:10 e 08:20
A que horas costuma lanchar no turno da manhã?	N.A.	N.A.	N.A.
A que horas costuma preparar o almoço?	Entre 10:00 e 10:30	Entre 10:30 e 11:00	Entre 09:00 e 09:20
Qual hora você costuma almoçar?	Entre 12:00 e 12:30	Entre 12:00 e 12:30	Entre 12:30 e 13:00
A que horas costuma lanchar no turno da tarde?	Entre 15:00 e 15:30	Entre 15:00 e 15:30	N.A.
A que horas costuma preparar o jantar/café da noite?	Entre 18:00 e 18:30	Entre 18:00 e 18:30	Entre 17:00 e 17:30
A que horas costuma jantar/café da noite?	Entre 19:30 e 20:00	Entre 19:00 e 19:30	Entre 17:30 e 18:00
Você costuma assistir TV? Se sim, em quais horários no dia?	Entre 08:30 e 09:00 Entre 12:00 e 13:00 Entre 20:00 e 20:40	Entre 08:30 e 09:00 Entre 12:00 e 13:00 Entre 18:00 e 18:30	Entre 12:00 e 13:00
Em quais horários costuma de ler jornal?	N.A.	N.A.	N.A.
Quais os horários que costuma tomar medicamento ao longo do dia?	N.A.	N.A.	N.A.
Qual/Quais horário(s) você costuma tomar banho ao longo do dia?	Entre 06:20 e 06:50 Entre 11:00 e 12:00 Entre 16:00 e 16:30	Entre 07:00 e 07:30 Entre 22:00 e 22:30	Entre 15:00 e 16:00
Você costuma sair de casa diariamente? Se sim, em quais horários?	Entre 05:00 e 06:15 Entre 06:30 e 08:00	Entre 09:00 e 10:15 Entre 08:00 e 08:10 Entre 17:00 e 17:30	Entre 09:00 e 09:30

Referências Bibliográficas

- ANDRADE, S. *Simulação Baseada em Agentes Para Alocação de Pessoal em Procedimento de Classificação de Risco na Emergência de um Hospital*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.
- BATISTA, A. S. . e. a. *Envelhecimento e Dependência: Desafios para a Organização da Proteção Social*. Brasília: MPS: Coleção Previdência Social, 2008.
- BLOM, J.; QUAKKELAAR, R.; ROTTEVEEL, M. Netlogo sql wrapper user manual. 2010.
- BORSHCHEV, A. Xj technologies: Anylogic 6. In: *Winter Simulation Conference*. ACM, 2005. p. 82. ISBN 0-7803-9519-0. Disponível em: <<http://dblp.uni-trier.de/db/conf/wsc/wsc2005.htmlBorshchev05>>.
- BUKSMAN, S. e. a. Quedas em idosos: Prevenção. In: *Projeto Diretrizes*. [S.l.: s.n.], 2008.
- CAMARGOS, M. C. S.; RODRIGUES, R. N.; MACHADO, C. J. Redes de apoio e estratégias de sobrevivência entre idosos que moram sozinhos. 2010.
- CAMARGOS, M. C. S.; RODRIGUES, R. N.; MACHADO, C. J. Idoso, família e domicílio: uma revisão narrativa sobre a decisão de morar sozinho. *Revista Brasileira de Estudos de População*, v. 28, n. 1, p. 217–230, 2011. ISSN 0102-3098.
- CARVALHO, A. A. *Padrão de Consumo dos Arranjos Familiares e das Pessoas Sozinhas no Brasil: Uma Análise por Gênero e Estratos Econômicos*. Dissertao (Mestrado) — Instituto Brasileiro de Geografia e Estatística (IBGE) ; Escola Nacional de Ciências Estatísticas (ENCE), RIO DE JANEIRO, 2010.
- CHORAFAS, D. N. *Agent Technology Handbook*. EUA: McGraw-Hill, 1998.
- COLLIER, N. *RePast: an extensible framework for agent simulation*. 2000. Disponível em: <<http://repast.sourceforge.net>>.
- COSER, A. *Utilização de Agentes Inteligentes no Trabalho Colaborativo via Internet*. Dissertao (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 1999.
- DAMÁSIO, A. *Descartes' Error Emotion, Reason and The Human Brain*. [S.l.]: Penguin Books; Reprint edition, 1994.
- DIJKSTRA, E. A note on two problems in connexion with graphs. *Numerische Mathematik*, Springer-Verlag, v. 1, n. 1, p. 269–271, 1959. ISSN 0029-599X. Disponível em: <<http://dx.doi.org/10.1007/BF01386390>>.

- FRANKLIN, S.; GRAESSER, A. Is it an agent, or just a program?: A taxonomy for autonomous agents. In: *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*. London, UK, UK: Springer-Verlag, 1997. (ECAI '96), p. 21–35. ISBN 3-540-62507-0. Disponível em: <<http://dl.acm.org/citation.cfm?id=648203.749270>>.
- GARCIA, A. C. B.; SICHMAN, J. S. *Sistemas Inteligentes, Fundamentos e Aplicações*. São Paulo: Manole, 2003. 269-306 p.
- GRIMM, V. et al. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, v. 198, n. 12, p. 115 – 126, 2006. ISSN 0304-3800. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304380006002043>>.
- HOLLAND, J. *Sistemas Complexos Adaptativos e Algoritmos Genéticos*. [S.l.: s.n.], 2003.
- JENNINGS, N. R. On agent-based software engineering. *Artificial Intelligence*, v. 117, n. 2, p. 277 – 296, 2000. ISSN 0004-3702. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0004370299001071>>.
- LASSER C.; OMOHUNDRO, S. M. *The Essential Starlisp Manual*. [S.l.]: Thinking Machines Corporation, 1996.
- LEDOUX, J. *The Emotional Brain - The Mysterious Underpinnings of Emotional Life*. [S.l.]: Simon & Schuster, 1996.
- MINAR, N.; BURKHART, R.; LANGTON, C.; ASKENAZI, M. The swarm simulation system: A toolkit for building multi-agent simulations. In: . [S.l.]: Santa Fe Institute, 1996.
- MORGADO, L. *Integração de Emoção e Raciocínio em Agentes Inteligentes*. Tese (Doutorado) — Universidade de Lisboa, Lisboa, 2005.
- MURTA, L. G. P. *CHARON: Uma Máquina de Processos Extensível Baseada em Agentes Inteligentes*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.
- ORACLE. Mysql. Disponível em: <http://www.oracle.com/br/products/mysql/index.html>. Acesso em: 09 Ago 2013. 2013.
- PAPERT, S. *Mindstorms: children, computers, and powerful ideas*. New York, NY, USA: Basic Books, Inc., 1980. ISBN 0-465-04627-4.
- PRASAD, M. N.; LESSER, V. Learning Situation Specific Coordination in Cooperative Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, v. 2, p. 173–207, 1999. Disponível em: <<http://mas.cs.umass.edu/paper-/274>>.

- PRESSMAN, R. S. *Engenharia de Software*. 6. ed. [S.l.]: McGraw-Hill, 2006. 720 p.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. [S.l.]: Prentice Hall, 2009.
- SAKURADA, N. Simulação baseada em agentes (sba) para modelagem de sistemas de operações. 2009.
- SIGNORETTI, A. *Agentes Inteligentes com Foco de Atenção Afetivo em Simulações Baseadas em Agentes*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Natal, 2012.
- SOMMERVILLE, I. *Engenharia de Software*. 6. ed. São Paulo, SP: Addison Wesley, 2003. 592 p. Tradução André Maurício de Andrade Ribeiro; Revisão técnica Kechi Hirama.
- SURANA, A.; KUMARA, S.; GREAVES, M.; RAGHAVAN, U. N. Supply-chain networks: a complex adaptive systems perspective. *International Journal of Production Research*, Taylor & Francis, v. 43, n. 20, p. 4235–4265, out. 2005. ISSN 0020-7543. Disponível em: <<http://dx.doi.org/10.1080/00207540500142274>>.
- TISUE, S.; WILENSKY, U. Netlogo: A simple environment for modeling complexity. In: *in International Conference on Complex Systems*. [S.l.: s.n.], 2004. p. 16–21.
- WEISZFLOG, W. Dicionário online michaelis. Disponível em: <http://http://michaelis.uol.com.br>. Acesso em: 15 Ago 2013. 2013.
- WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. Wiley, 2008. ISBN 9780470353479. Disponível em: <http://books.google.com.br/books?id=C4_9riKP2kQC>.
- WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, v. 10, p. 115–152, 1995.