



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
TRABALHO DE CONCLUSÃO EM ENGENHARIA DE
CONTROLE E AUTOMAÇÃO

MURILO DE OLIVEIRA PROVENZI

**PREDIÇÃO DE ATIVIDADES EM UM AMBIENTE
INTELIGENTE UTILIZANDO SISTEMAS
MULTI-AGENTES E REDES BAYESIANAS**

Porto Alegre
2016

MURILO DE OLIVEIRA PROVENZI

**PREDIÇÃO DE ATIVIDADES EM UM AMBIENTE
INTELIGENTE UTILIZANDO SISTEMAS MULTI-AGENTES E
REDES BAYESIANAS**

Trabalho de Conclusão apresentado à comissão de Graduação de Engenharia de Controle e Automação da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Engenheiro de Controle e Automação.

ORIENTADOR: Professor Marcelo Götz

Porto Alegre
2016

MURILO DE OLIVEIRA PROVENZI

**PREDIÇÃO DE ATIVIDADES EM UM AMBIENTE
INTELIGENTE UTILIZANDO SISTEMAS MULTI-AGENTES E
REDES BAYESIANAS**

Este Trabalho foi julgado adequado para
obtenção dos créditos da disciplina Trabalho
de Conclusão de Curso em Engenharia de
Controle e Automação e aprovado em sua
forma final pela Orientador e pela Banca
Examinadora.

Orientador: _____
Professor Marcelo Götz, UFRGS
Doutor(a) pela Universität Paderborn – Alemanha

Banca Examinadora:

Prof. Dr. Carlos Eduardo Pereira, UFRGS
Doutor pela Universidade de Stuttgart – Alemanha

Prof. Dr. João Cesar Netto, UFRGS
Doutor pela Université Catholique de Louvain – Bélgica

Prof. Dr. Rafael Antônio Comparsi Laranja, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Rs

Porto Alegre
2016

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Adalberto Luiz Provenzi e Claudia Regina de Oliveira Provenzi. Por sempre me darem amor e apoio, por me incentivarem a expandir meu mundo, por terem ido além para possibilitar essa realização e pela parceria em todos os momentos: muito, muitíssimo obrigado! Eu amo vocês.

Agradeço também ao meu orientador, Prof. Marcelo Götz, por aceitar me ajudar neste trabalho e por estar sempre disposto a ouvir com atenção os avanços e as dúvidas, retornando sugestões de grande valor. Obrigado!

À minha linda namorada, Caroline, pelo amor incessante, pela parceria e pela compreensão ao longo dos últimos meses, pelo incentivo e pela confiança nos momentos mais difíceis, obrigado! E obrigado por ter feito esse tempo todo ser muito mais leve e feliz. Eu te amo!

Aos meus irmãos, Francisco e Carolina, agradeço pelas brincadeiras, pelas risadas, por às vezes quererem até ajudar e pela compreensão da distância.

Agradeço também a todos os meus amigos, que fazem os dias terem mais tom de riso, e em especial ao Ezequiel, ao Túlio, ao Botto e ao Guilherme, pelas incontáveis conversas inenarráveis e incompreensíveis.

Aos professores membros da banca, obrigado pelo aceite.

E obrigado também a todos os outros familiares e amigos que contribuíram de alguma forma para a conclusão deste trabalho. Valeu!

RESUMO

A tecnologia presente em dispositivos como SmartPhones está transformando a vida das pessoas. Também está sendo incorporada em residências de variadas maneiras, e assim Ambientes Inteligentes estão se tornando cada vez mais acessíveis e disseminados. Este projeto foca em estudar e avaliar uma Rede Bayesiana para antecipar interações de um habitante com um Ambiente Inteligente. É usado um Sistema Multi-Agentes para modelar as funcionalidades do ambiente e permitir expansibilidade para diferentes funcionalidades. Para o estudo de caso, implementam-se o Sistema Multi-Agentes e a Rede Bayesiana em um Raspberry Pi, e utiliza-se o sistema embarcado ESP8266 para desenvolver dispositivos automatizados conectados à rede local, introduzindo sensoramento de posição e atuação sobre iluminação no ambiente. O sistema reduziu as interações do usuário com o ambiente em mais de 50 %, atuando através da Wi-Fi com atraso na casa de um segundo.

Palavras-Chave: Ambientes Inteligentes, Redes Bayesianas, Sistema Multi-Agentes, Automação Residencial, Internet das Coisas.

ABSTRACT

Technology in devices such as SmartPhones is transforming people's lives. It's also being incorporated into homes in various ways, and hence Smart Environments are becoming more accessible and real. This report focuses on studying and testing a Bayesian Network for anticipating inhabitant interactions with a Smart Environment. It also uses Multi-Agent Systems to model the environment's functionalities and allow for expandability. As a case study, the Multi-Agent System and the Bayesian Network are programmed and run in a Raspberry Pi, and the ESP8266 embedded system is used to develop automated devices connected to the local network, which introduces position sensing and lighting control into the environment. The system reduced the user's interactions with the environment by more than 50 % actuating over the Wi-Fi with a standard delay around one second.

Keywords: Smart Environments, Bayesian Networks, Multi-Agent Systems, Home Automation, Internet of Things.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE TABELAS	viii
LISTA DE ABREVIATURAS	ix
1 INTRODUÇÃO	1
2 REVISÃO BIBLIOGRÁFICA	3
2.1 Sistemas Multi-Agentes	3
2.1.1 SMA em Ambientes Inteligentes	4
2.2 Redes Bayesianas	6
2.2.1 Aplicações de Redes Bayesianas em Ambientes Inteligentes	7
2.3 Sistemas de computação embarcada	8
2.3.1 Raspberry Pi	8
2.3.2 ESP8266	9
3 MODELAGEM DE UM AMBIENTE INTELIGENTE UTILIZANDO SISTEMA MULTI-AGENTES E REDES BAYESIANAS	10
3.1 O ambiente automatizado	10
3.2 O Sistema Multi-Agentes	10
3.3 Rede Bayesiana para predição de atividades	13
4 ESTUDO DE CASO	17
4.1 Dispositivos Automatizados	17
4.2 Implementação do Sistema Multi-Agentes	19
4.2.1 JADE	21
4.2.2 Implementação do Agente Habitação	22
4.2.3 Implementação do Agente Iluminação	22
4.2.4 Implementação do Agente Banco de Dados	23
4.2.5 Agente ESP8266	23
4.3 Predição de atividade	23
5 RESULTADOS E DISCUSSÃO	26
6 CONCLUSÕES E TRABALHOS FUTUROS	30
REFERÊNCIAS BIBLIOGRÁFICAS	31
APÊNDICE	33

LISTA DE FIGURAS

2.1	Estrutura dos agentes de MavHome. Adaptada de (COOK; YOUNG-BLOOD; HEIERMAN, 2003).	5
2.2	Modelos de comunicação com os agentes ligados aos seus dispositivos. Adaptada de (MOERGESTEL et al., 2013).	5
2.3	Modelos de comunicação com os agentes em um mesmo ambiente de execução. Adaptada de (MOERGESTEL et al., 2013).	6
2.4	Proposta de estrutura de Rede Bayesiana para predição de atividade. Adaptada de (NAZERFARD; COOK, 2012).	8
3.1	Desenho da planta do ambiente automatizado.	11
3.2	Hierarquia de agentes e fluxos de informações.	12
3.3	Estrutura de camadas do Agente Iluminação.	13
3.4	Redes Bayesianas A e B.	14
4.1	Aplicativo de <i>SmartPhone</i> criado neste trabalho para controle dos atuadores.	18
4.2	Sensores e atuador conectados ao ESP2.	19
4.3	Diagrama de interfaces com usuário e entre dispositivos.	20
4.4	Arquitetura distribuída do JADE. Fonte: (FONSECA, 2011).	21
5.1	Frequentes alterações no abajur por perda de localização do habitante.	28

LISTA DE TABELAS

3.1	Estados das variáveis da Rede Bayesiana.	14
3.2	Probabilidades de DT0.	14
3.3	Distribuição de probabilidades de DT1.	15
3.4	Probabilidades de DT1 sob AT1.	15
4.1	Relação dos planos e dos comportamentos do agente Habitação. . . .	22
4.2	Relação dos planos e dos comportamentos do agente Iluminação. . .	22
4.3	Relação dos planos e dos comportamentos do agente BD.	23
4.4	Condição de iluminação do ambiente para cada atividade.	24
4.5	Distribuição de probabilidades entre as variáveis LT0 e LT1.	25
5.1	Quantidade de interações do habitante em um período de 4 dias. . . .	26
5.2	Horários das interações do habitante em um período de 4 dias. . . .	27
5.3	Percentuais de acertos das previsões das diferentes variáveis.	28
6.1	Distribuição de probabilidades da variável LT0 na Rede Bayesiana A. .	33
6.2	Distribuição de probabilidades da variável DT0 na Rede Bayesiana A. .	33
6.3	Distribuição de probabilidades da variável HT0 na Rede Bayesiana A. .	33
6.4	Distribuição de probabilidades da variável HT1 na Rede Bayesiana A. .	34
6.5	Distribuição de probabilidades da variável DT1 na Rede Bayesiana A. .	34
6.6	Distribuição de probabilidades da variável AT0 na Rede Bayesiana B. .	34
6.7	Distribuição de probabilidades da variável AT1 na Rede Bayesiana B. .	35
6.8	Distribuição de probabilidades da variável DT1 na Rede Bayesiana B. .	35
6.9	Distribuição de probabilidades da variável LT1 na Rede Bayesiana B. .	35
6.10	Distribuição de probabilidades da variável HT1 na Rede Bayesiana B. .	36

LISTA DE ABREVIATURAS

AAII	<i>Australian Artificial Intelligence Institute</i>
AI	Ambiente Inteligente
API	<i>Application Programming Interface</i>
DCP	Distribuição Conjunta de Probabilidades
FIPA	<i>Foundation for Intelligent Physical Agents</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IP	<i>Internet Protocol address</i>
JADE	<i>Java Agent Development Framework</i>
LDR	<i>Light Dependant Resistor</i>
POO	Programação Orientada a Objetos
RAM	<i>Random-Access Memory</i>
SMA	Sistemas Multi-Agentes
TCP	<i>Transmission Control Protocol</i>

1 INTRODUÇÃO

Nos últimos anos, vem acontecendo uma nova mudança na maneira como a tecnologia pode ser utilizada na vida das pessoas. Começando pela disseminação dos *SmartPhones*, em parte devida ao avanço das suas capacidades de processamento e de conectividade, são cada vez mais frequentes os exemplos de aplicações e serviços que são integrados à rotina de um indivíduo. Como exemplos e com funcionalidades semelhantes, os assistentes pessoais *Now*, da *Google*, e *Siri*, da *Apple* avisam o usuário sobre quanto tempo ele levará ao trabalho antes mesmo de ser solicitado. Além disso, quando a previsão do tempo indica chuva, lembram-no de levar um guarda-chuvas consigo (APPLE, 2016) (GOOGLE, 2016).

Uma das principais propostas deste tipo de aplicação é, de fato, antecipar-se ao usuário, entregando-o a informação que ele deseja antes que ele a solicite. O interessante é que esses serviços não estão restritos a *SmartPhones*, computadores, etc. Impulsionados principalmente pelo advento da Internet das Coisas, existem, por exemplo, dispositivos de automação residencial que aprendem e se adaptam à rotina dos habitantes para controlar a temperatura de maneira autônoma, procurando maximizar a relação conforto/consumo de energia, e que podem se conectar a dispositivos como lâmpadas inteligentes para acendê-las e apagá-las pró-ativamente quando o usuário iria fazê-lo (NEST, 2016). Essas novas possibilidades exponencializam aplicações de automação residencial para dar inteligência aos ambientes.

De fato, já no final do século XX havia estudos sobre as características que definiriam um ambiente inteligente, como adaptabilidade e pró-atividade (MOZER, 1999). Essas características podem ser alcançadas através de diferentes métodos: descoberta de padrões recorrentes de uso dos dispositivos (HEIERMAN III; COOK, 2003), redes neurais auto-adaptativas (ZHENG; WANG; BLACK, 2008) e Redes Bayesianas (NAZERFARD; COOK, 2012). Um ambiente com múltiplos dispositivos inteligentes que se comunicam para trocar informações, monitorando e controlando em harmonia o estado do mesmo, é tido como o nível mais alto de domótica segundo (MOERGESTEL et al., 2013).

O presente trabalho modela e implementa um ambiente inteligente que é capaz de antecipar as interações do habitante com seus dispositivos, fazendo-o através de uma arquitetura que permite o controle distribuído da atuação dos seus dispositivos automatizados. O objetivo principal do trabalho é:

- Antecipar as interações do habitante com o ambiente, de modo a reduzi-las em número.

Para alcançá-lo, os objetivos específicos são os seguintes:

- Estudar e aplicar um modelo de ambiente inteligente através do uso de agentes;
- Criar dispositivos automatizados com conectividade para comunicação;
- Estudar o uso de Redes Bayesianas para realizar inferência estatística das atividades do habitante, aplicando-as neste contexto.

O trabalho está organizado da seguinte maneira:

- No capítulo 2, está apresentada a revisão bibliográfica sobre os principais temas abordados neste trabalho: Sistemas Multi-Agentes e Redes Bayesianas. Também são introduzidas as principais características dos sistemas de computação embarcada utilizados no trabalho;
- No capítulo 3, trabalha-se em extrair os modelos de Sistemas Multi-Agentes e de Redes Bayesianas que atinjam o objetivo principal do trabalho;
- No capítulo 4, realiza-se o estudo de caso. São descritos os dispositivos automatizados e a implementação dos modelos montados no capítulo 3;
- No capítulo 5 são apresentados os resultados e é realizada uma discussão sobre suas causas e suas implicações;
- Por último, são apresentadas as conclusões obtidas após a realização do trabalho e levantadas opções de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta o estudo realizado sobre dois dos principais temas abordados neste trabalho: Sistemas Multi-Agentes e Redes Bayesianas. Os conceitos relacionados são definidos, e são mostradas características que os candidatam à utilização no presente projeto, além de trabalhos acadêmicos que os aplicam nesta área. Além dessas, também são apresentados os sistemas de computação embarcada que serão utilizados no trabalho.

2.1 Sistemas Multi-Agentes

A literatura não apresenta um conceito único para agentes, havendo algumas nuances em relação a como cada autor os define. Uma abordagem clara é a de (WOOLDRIDGE, 2002), que declara agentes como sistemas computacionais inseridos em algum ambiente com capacidade de tomar ações autônomas sobre ele para alcançar seus dados objetivos. Este referido ambiente pode ser computacional, físico, virtual, etc., e assim a definição permite uma certa versatilidade a esses sistemas, que são muito utilizados em Inteligência Artificial e em sistemas distribuídos em geral, como a própria Internet. As atividades executadas pelos agentes são chamadas de comportamentos. Além disso, é comum expressar que os agentes possuem crenças e desejos: como percebem seu ambiente e quais seus objetivos sobre ele, respectivamente (WOOLDRIDGE, 2002).

Essa definição é bastante ampla e, portanto, exige dos comportamentos algumas características para torná-los inteligentes, apresentadas a seguir (WOOLDRIDGE, 2002).

- Reatividade: percepção do ambiente onde estão inseridos, respondendo a alterações;
- Pró-atividade: iniciativas com comportamentos voltados aos objetivos;
- Habilidade social: interações com outros agentes, como solicitações, negociações e fornecimento de informações.

A definição geral de agentes torna-os semelhantes a objetos em Programação Orientada a Objetos (POO), mas existe uma diferença fundamental: estes não possuem controle sobre a execução de seus métodos, enquanto que um agente pode apenas solicitar a execução de um comportamento de outro, e este irá realizá-lo apenas se for de seu interesse. Essa diferença reforça a ideia de autonomia em agentes (WOOLDRIDGE, 2002).

Um Sistema Multi-Agentes (SMA) é uma estrutura de *software* em que é formada a sociedade dos agentes e que possibilita comunicação entre eles, que assim podem trabalhar em conjunto para atingir seus respectivos objetivos (PEIXOTO, 2012). Esta sociedade pode ser centralizada ou distribuída, possibilitando sua execução em diferentes ambientes de computação. Algumas das principais características de SMAs são escalabilidade, alcançada pela inclusão ou exclusão de agentes com diferentes propósitos, flexibilidade, permitindo a conexão de sistemas com diferentes arquiteturas, reusabilidade, pois um mesmo agente pode ser utilizado em ambientes diferentes, e facilidade no desenvolvimento de sistemas, uma vez que o problema original é dividido em diversos módulos com objetivos intermediários (FONSECA, 2011).

Para modelar um SMA, (WOOLDRIDGE, 2002) cita uma série de metodologias, dentre as quais destaca-se a do *Australian Artificial Intelligence Institute* (AAII), inspirada

em e desenvolvida a partir de técnicas de POO aliadas à aplicação de conceitos de agentes. Essa metodologia desenvolve modelos externos, que definem quais são os agentes e quais as relações entre eles, e internos, que definem quais as crenças e os desejos de cada um, e pode ser resumida nos quatro passos a seguir (WOOLDRIDGE, 2002).

- I Identificar as funcionalidades necessárias às aplicações no ambiente e extrair a hierarquia dos agentes;
- II Listar as responsabilidades associadas a cada funcionalidade e os serviços requeridos e disponibilizados por elas, bem como seus objetivos;
- III Para cada objetivo, determinar os planos que podem ser utilizados para alcançá-lo e as condições de contexto sob as quais são apropriados;
- IV Determinar a estrutura de crenças para cada plano e objetivo.

2.1.1 SMA em Ambientes Inteligentes

O conceito de Ambiente Inteligente (AI) na literatura apresenta algumas diferenças de autor para autor, mas parecem convergir para um ponto: a simples automatização dos dispositivos, como móveis, eletrodomésticos, aparelhos de multimídia, etc., em um dado ambiente não o torna inteligente. (COOK; DAS, 2007) define AIs como ambientes capazes de adquirir e utilizar informações sobre si próprio e sobre seus habitantes a fim de melhorar a experiência destes nele mesmo. Em (CALLAGHAN et al., 2004), o autor cita a definição de que esta inteligência se dá como uma abordagem mais amigável ao usuário, dando-o maior poder em serviços de suporte mais eficientes, e que o ambiente é capaz de reconhecer e responder à presença de um ou mais indivíduos de maneira fluida e até invisível.

Percebe-se que há uma semelhança entre os conceitos de AIs com o de agentes, principalmente o apresentado por (COOK; DAS, 2007). Além disso, um AI apresenta algumas das características potencializadoras da utilização de SMAs apresentadas no trabalho de (FONSECA, 2011): é intrinsecamente distribuído, dinâmico, possui diferentes resolvedores com diferentes protocolos de comunicação e as decisões são descentralizadas, com diferentes graus de autonomia.

Isso sugere a modelagem de um Ambiente Inteligente por um Sistema Multi-Agentes. De fato, o trabalho dos autores em (COOK; YOUNGBLOOD; HEIERMAN, 2003) aborda a residência *MavHome* como um agente racional que é capaz de prever e de se adaptar aos seus habitantes de modo a maximizar seu conforto e sua produtividade minimizando custos operacionais. No estudo, é apresentado um cenário em que diversos dispositivos estão automatizados, e dessa forma o agente *MavHome* é dividido em agentes de mais baixo nível que desempenham diferentes tarefas, como conjuntos de sensores e de atuadores (COOK; YOUNGBLOOD; HEIERMAN, 2003).

A estrutura interna dos agentes em (COOK; YOUNGBLOOD; HEIERMAN, 2003), com camadas física, de comunicação, de informações e de decisão bem definidas, e a hierarquia do sistema estão apresentadas na Figura 2.1.

Neste trabalho, a camada física é composta por dispositivos de hardware, como os sensores e atuadores, e por outros dispositivos em geral. As redes responsáveis pela comunicação entre e intra agentes também pertencem a ela. Já a chamada camada de comunicação define como será realizada a troca de informações entre os agentes. A camada de informação reúne e armazena conhecimento relevante à funcionalidade do

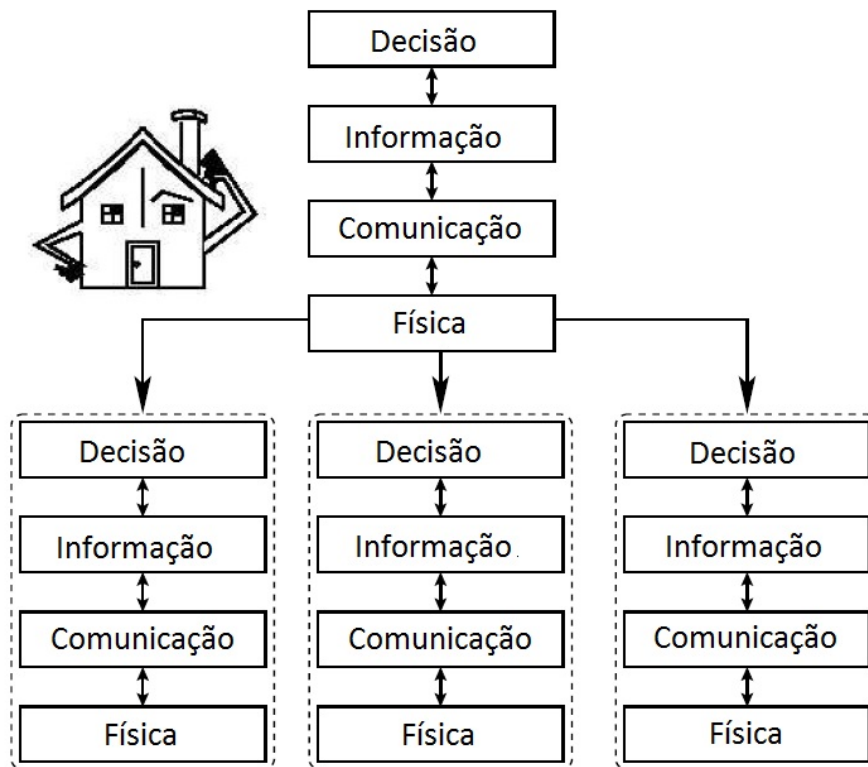


Figura 2.1: Estrutura dos agentes de MavHome. Adaptada de (COOK; YOUNGBLOOD; HEIERMAN, 2003).

agente e fornece informações úteis para tomada de decisão. Por último, a camada de decisão seleciona ações a serem executadas com base em informações da camada anterior. Essa arquitetura dividida permite que as tecnologias individuais de cada uma das estruturas sejam modificadas sem exigir atualização em todo o sistema ou em todo o agente (COOK; YOUNGBLOOD; HEIERMAN, 2003).

Em relação ao ambiente computacional em que os agentes são executados e como se dá o relacionamento deste com os dispositivos associados, existem duas opções, segundo (MOERGESTEL et al., 2013): cada agente pode rodar no mesmo sistema em que estão seus dispositivos, o que facilita esta integração mas exige capacidade de processamento distribuída (Figura 2.2), ou os agentes rodam no mesmo ambiente e separados fisicamente dos seus dispositivos, o que facilita a comunicação entre os agentes mas torna o sistema como um todo mais dependente da interface com os dispositivos (Figura 2.3). Assim, essas características devem ser consideradas quando se define como será a execução dos agentes.

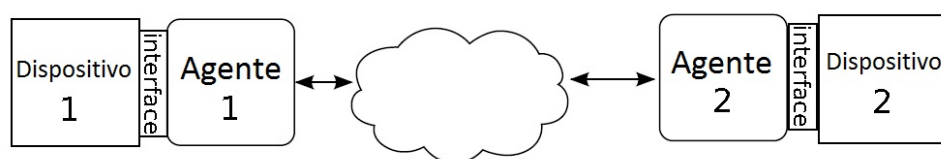


Figura 2.2: Modelos de comunicação com os agentes ligados aos seus dispositivos. Adaptada de (MOERGESTEL et al., 2013).

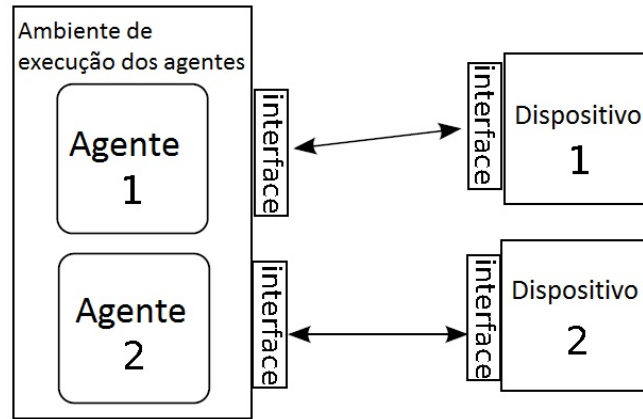


Figura 2.3: Modelos de comunicação com os agentes em um mesmo ambiente de execução. Adaptada de (MOERGESTEL et al., 2013).

2.2 Redes Bayesianas

O Teorema de Bayes apresenta uma relação de probabilidade condicionada entre dois ou mais eventos. Ou seja: sejam A e B dois eventos com suas respectivas probabilidades de ocorrência *a priori* conhecidas, o teorema fornece um método de calcular a probabilidade de que um destes eventos ocorra dado que sabe-se da ocorrência do outro, ou probabilidade *a posteriori* (BUSSAB; MORETTIN, 2010).

A Equação (2.1) apresenta a equação matemática fornecida pelo Teorema de Bayes em sua forma geral, que calcula a probabilidade de ocorrência do evento C_i , dados o conhecimento (evidência) ou a hipótese de ocorrência do evento A ($P(C_i | A)$), e considerando os outros C_j eventos não independentes a A. Na Equação (2.2) está apresentada uma forma simplificada da Equação (2.1) para apenas dois eventos A e B (BUSSAB; MORETTIN, 2010):

$$P(C_i | A) = \frac{P(C_i)P(A | C_i)}{\sum_{j=1}^n P(C_j)P(A | C_j)}, \quad (2.1)$$

$$P(A | B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B | A)}{P(B)}. \quad (2.2)$$

As Redes Bayesianas são um tipo de modelo probabilístico que relaciona as dependências entre um determinado número de variáveis. Em aplicações de aprendizado de máquina, possuem formato de grafo dirigido acíclico, em que são apresentados nós, os quais representam as variáveis, e arestas, que definem as dependências entre eles. As relações de dependência recebem graus de parentesco, e sua ausência entre duas variáveis é interpretada como independência (RUGGERI; KENETT; FALTIN, 2008). A Figura 2.4, na próxima seção, apresenta o grafo de uma rede, e ela será então detalhada mais adiante.

As Redes Bayesianas são um método eficiente de cálculo de probabilidades condicionais pois, com uma correta definição da estrutura de dependência dos nós, reduz-se a quantidade de dados estatísticos necessários para realizar os cálculos, o que facilita a tarefa de inferência. Define-se então uma única distribuição conjunta de probabilidades (DCP) sobre o conjunto de n variáveis X_i condicionadas aos seus respectivos conjuntos de dependência, π_i . Ela é dada pela Equação (2.3) (RUGGERI; KENETT; FALTIN, 2008).

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P_B(X_i | \pi_i). \quad (2.3)$$

A inferência em Redes Bayesianas pode ser chamada de preditiva, caso se tenha evidência de antepassados do nó de interesse, e diagnóstica, caso as evidências sejam sobre os descendentes. Para realizá-la, é necessário dispor da DCP para todos os nós de interesse de acordo com as relações de dependência, o que também pode exigir alta capacidade de processamento. A estrutura de uma Rede Bayesiana e os valores numéricos que preenchem sua DCP podem ser estimados através da experiência de um especialista no assunto ou calculados a partir de um histórico de dados (RUGGERI; KENETT; FALTIN, 2008).

2.2.1 Aplicações de Redes Bayesianas em Ambientes Inteligentes

As Redes Bayesianas podem ser utilizadas para auxiliar em diferentes tarefas de inferência em ambientes inteligentes. Em (ALESSANDRA et al., 2016), o ambiente inteligente é dividido em três camadas: baixa, com diferentes sensores espalhados pelo ambiente; intermediária, com fusão de dados, considerando confiabilidade e falhas de *hardware*, e inferência das condições do ambiente; e alta, com extração de contexto a partir das condições do ambiente e otimização da utilização dos dispositivos. Na camada intermediária, há uma Rede Bayesiana dinâmica (que leva em conta estados anteriores, além dos atuais), responsável pelo processo de inferência, que calcula o estado do seu mundo a partir de informações de contexto externas fornecidas e das leituras dos sensores, conforme a equação recursiva deduzida pelos autores (Equação (2.4)):

$$Bel(x_t) = \eta \cdot \prod_{e_t^i} P(e_t^i | x_t) \cdot \sum_{x_{t-1}} P(x_t | x_{t-1}, C_t) \cdot Bel(x_{t-1}). \quad (2.4)$$

Na Equação (2.4)), x_t é a condição atual, testada a cada instante, e x_{t-1} é a anterior, bem como e_t^i é o estado de cada sensor i no tempo atual, C_t são as condições de contexto, η é apenas uma constante de normalização e, finalmente, $Bel(x_t)$ é a confiança de que o estado se encontra no estado x_t . Os autores destacam que a complexidade computacional é dada por $O(m^2 + mn)$, em que n é o número de sensores e m o número de valores possíveis para x_t .

Já em (HSUEH et al., 2015) são usadas câmeras para acompanhar o movimento de habitantes dentro de um ambiente e Redes Bayesianas para identificar eventos anormais, mas não apresentam muitos detalhes de seus cálculos. (ALAM et al., 2016) também utilizam uma Rede Bayesiana dinâmica, esta para inferir atividades em um contexto *macro* (ex.: cozinhando) a partir de outros *micro* (ex.: caminhando; em silêncio; na cozinha), e que correlaciona a estrutura e as distribuições de probabilidades entre um ou mais indivíduos (por exemplo: calculando a probabilidade de um indivíduo estar jantando dado que algum outro indivíduo está).

Também há o estudo de (NAZERFARD; COOK, 2012), que propõe uma estrutura de Redes Bayesianas para predição de atividades, comparando-a com outras. A Figura 2.4 apresenta a proposta dos autores, em que os nós com X_t e X_{t+1} representam as atividades corrente e seguinte, respectivamente, e os nós com Y^i representam as características de contexto da atividade i de 1 a 3. As características Y_t^i são momento no dia, discretizado em seis intervalos de 4 horas de duração, localização dentro do ambiente inteligente e dia

da semana em que a atividade corrente está acontecendo. Y_{t+1}^i de 1 a 3 representam os respectivos valores destas variáveis para a próxima atividade.

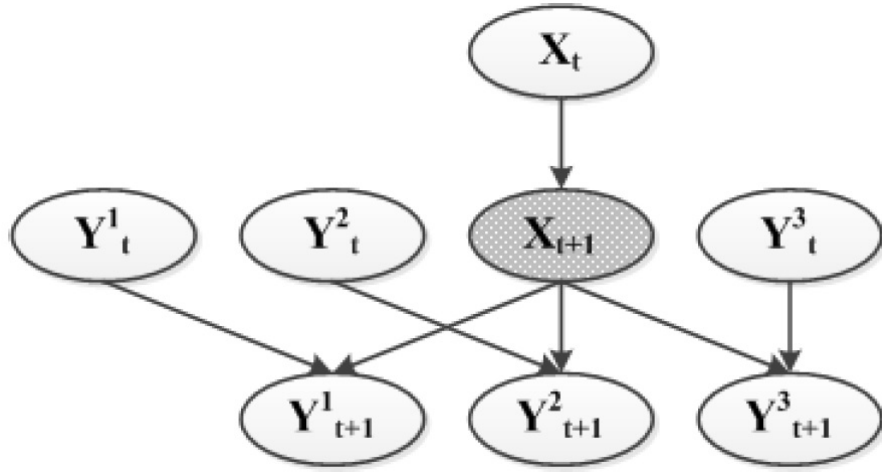


Figura 2.4: Proposta de estrutura de Rede Bayesiana para predição de atividade. Adaptada de (NAZERFARD; COOK, 2012).

A proposta dos autores é dividir o processo de predição em dois passos: primeiramente, a partir do estado atual t de cada uma das características Y^1 , Y^2 e Y^3 , inferem-se os seus respectivos estados mais prováveis no próximo momento ($t+1$). Então, dadas a atividade atual X_t , assumida conhecida (o trabalho não mostra como essa informação é estimada), e as características Y_{t+1}^1 , Y_{t+1}^2 , Y_{t+1}^3 estimadas para a atividade seguinte, esta é então inferida. Ou seja, de maneira independente são previstas as características esperadas para a próxima atividade, e com base nisto e na atividade atual é que ela será prevista. Destaca-se aqui que a estrutura apresentada não apresenta relação de dependência da atividade atual com suas respectivas características.

Ainda, este estudo apresenta a configuração do ambiente inteligente cujos dados foram utilizados para treinamento e validação dos diferentes algoritmos de predição, informando que há sensores de movimentação pelos cômodos e de contato em portas e armários. Suas conclusões são de que a estrutura com melhor desempenho em predição de atividades é a sua proposta, atingindo precisão de 70,33 %, além de prever com 97,03 %, 84,23 % e 52,79 % de precisão as características dia da semana, horário e local, respectivamente.

2.3 Sistemas de computação embarcada

Conforme será apresentado nos próximos capítulos, são utilizados dois sistemas de computação embarcada para o estudo de caso deste trabalho: o *Raspberry Pi* (PI, 2016) e o *ESP8266* (ESPRESSIF, 2016). Nas seções a seguir, são feitas breves apresentações das principais características destes sistemas.

2.3.1 Raspberry Pi

O *Raspberry Pi* se caracteriza como um computador de baixo custo, se comparado com computadores pessoais, e costuma ser utilizado para ensino e prototipagem. O modelo 3, utilizado neste trabalho por possuir conexão nativa à *Wi-Fi*, possui um processador

ARMv8 quad-core de 64 bits a 1,2 GHz e 1 GigaByte de *Random-Access Memory* (RAM), além de possuir interfaces USB, HDMI, pinos de entrada e saída digitais, dentre outros (PI, 2016).

Ele utiliza para memória *flash* um cartão de memória, onde deve ser instalado o sistema operacional de interesse. Existem várias opções desenvolvidas especificamente para este computador, destacando-se o *Raspbian*, versão de *Debian* que possui nativamente instalados diversos compiladores de diferentes linguagens de programação (PI, 2016). Vale destacar que no trabalho de (MOERGESTEL et al., 2013) são utilizados *Raspberry Pi* para implementar um SMA.

2.3.2 ESP8266

O *ESP8266-ESP07* é um sistema embarcado que possui um *chip* de conexão a *Wi-Fi* e um microcontrolador de 32-bits com 36 kiloBytes de RAM e 1 MegaByte de memória *flash*. Seu controlador disponibiliza 8 pinos digitais de entrada e saída, 1 pino de entrada analógica com conversor analógico-digital de 10 bits e uma interface serial, utilizada para programação e para comunicação. A fim de mantê-lo com baixo consumo de energia, sua alimentação e o funcionamento de seus pinos são a +3,3 V (ESPRESSIF, 2016).

Sua *Firmware* de fábrica o entrega como um escravo que responde a e executa comandos AT enviados à sua interface serial, e é dessa mesma forma que pode ser programada uma aplicação em sua memória *flash* (ESPRESSIF, 2016). Entretanto, existe uma biblioteca que permite programá-lo através da *Arduino IDE* utilizando códigos semelhante aos de *Arduino*, sendo que ela os converte para os comandos que o controlador reconhece (GROKHOTKOV, 2016).

Segundo os desenvolvedores do *ESP8266*, este *chip* tem sido muito utilizado por desenvolvedores da Internet das Coisas, devido aos seus baixo consumo e baixo custo (comparando-se com outros sistemas de conexão a redes em fio), além da facilidade de programação e capacidade de comunicação pela rede, uma vez que suas características permitem a criação de diversos produtos conectados à Internet (ESPRESSIF, 2016).

3 MODELAGEM DE UM AMBIENTE INTELIGENTE UTILIZANDO SISTEMA MULTI-AGENTES E REDES BAYESIANAS

Para implementar um Sistema Multi-Agentes em um Ambiente Inteligente, é realizado um estudo sobre o ambiente em questão e sobre qual a aplicação que se deseja realizar ao final, a fim de então extrair o modelo dos agentes. A etapa do desenvolvimento em que devem ser considerados os dispositivos de automação, como sensores e atuadores, depende dos recursos disponíveis, pois a modelagem por agentes é bastante flexível.

Assim, a seguir será apresentado o ambiente automatizado durante o desenvolvimento deste trabalho (os dispositivos serão detalhados no próximo capítulo). Então, será realizada a modelagem do SMA considerando a automação limitada disponível, mas de modo que o sistema possa ser facilmente expandido para englobar outros dispositivos e outras funcionalidades. Por último, será desenvolvido o modelo de inteligência a ser utilizado nos agentes deste SMA baseado em uma Rede Bayesiana, a fim de realizar a predição das atividades do habitante.

O capítulo 4, seguinte a este, apresenta o estudo de caso neste trabalho. Como mencionado, serão detalhados os dispositivos automatizados, bem como apresentadas as maneiras de interface com o usuário. Também consta naquele capítulo a implementação tanto do SMA quanto da Rede Bayesiana.

3.1 O ambiente automatizado

O ambiente automatizado utilizado no estudo de caso deste trabalho se trata de um dormitório com dimensões de aproximadamente 2,7 m x 4,7 m em que há um roupeiro, uma estante, uma cama e dois criado-mudos. A Figura 3.1 apresenta um desenho da planta deste quarto.

Neste ambiente, foram automatizados a lâmpada de teto e o abajur. Foram também posicionados sensores de presença e de movimento em duas localizações, sendo que o primeiro detecta em linha reta, e o segundo em um arco de aproximadamente 110 graus. São elas: na entrada do quarto, à frente do roupeiro, e acima da cama, sendo que nesta também foi colocado um sensor de iluminação. Todos esses dispositivos estão destacados em seus respectivos locais de instalação na Figura 3.1.

3.2 O Sistema Multi-Agentes

Uma das vantagens da utilização de um SMA em um Ambiente Inteligente é que o controle de cada um dos dispositivos automatizados não fica intrinsecamente dependente da tecnologia destes, pois o que é específico de cada dispositivo será apenas a comunicação com o agente que está relacionado à sua atividade. Ou seja, caso seja trocado algum sensor ou atuador por outro semelhante, atualiza-se a comunicação do mesmo com o agente correspondente, e este continuará oferecendo o mesmo serviço ao SMA. Assim, essa alteração será transparente às funcionalidades do AI.

Conforme apresentado, a aplicação sendo desenvolvida é a de antecipar as interações do habitante com o ambiente, configurando os estados dos atuadores automatizados de acordo com a previsão de atividade a ser desempenhada. Seguindo a metodologia do

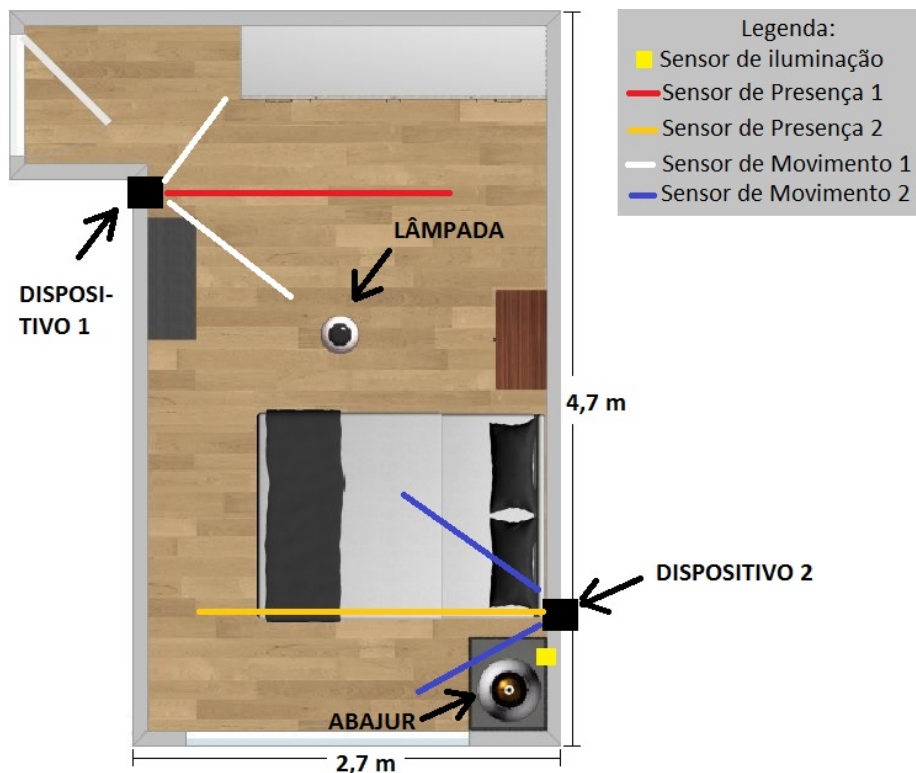


Figura 3.1: Desenho da planta do ambiente automatizado.

AII, a modelagem do SMA desenvolvido neste trabalho passa então por quatro etapas, começando pela definição das funcionalidades que são necessárias para implementar esta aplicação, a partir das quais é extraída a hierarquia dos agentes.

Antes disso, é válido destacar que a modelagem a seguir está relacionada, de certo modo, à automação que se tem disponível para realizar a aplicação proposta. Por isso, a estrutura de agentes terá poucos deles em quantidade. Entretanto, como esse sistema se apresenta como uma abstração dos serviços automatizados oferecidos no ambiente, a estrutura poderá ser expandida para acomodar a adição de outros dispositivos automatizados que ofereçam diferentes serviços.

Como descrito na Seção 2.1, inicia-se listando as funcionalidades desejadas no SMA:

- i Inferência de habitação: responsável por determinar em que local do ambiente o habitante se encontra;
- ii Controle de iluminação: responsável por controlar a iluminação do ambiente de acordo com a preferência do habitante;
- iii Interface com banco de dados: responsável por reunir as informações dos agentes e enviá-las ao banco de dados.

Essas funcionalidades induzem à criação de 3 agentes diferentes, nomeados respectivamente como Habitação, Iluminação e BD (Banco de Dados), e cuja comunicação está estruturada conforme apresentado na Figura 3.2.

O segundo ponto de desenvolvimento do SMA envolve listar, para cada funcionalidade, as responsabilidades associadas, os serviços requeridos e oferecidos e os objetivos.

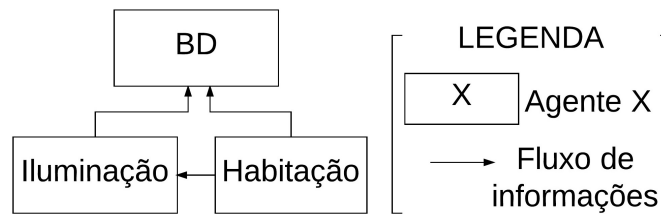


Figura 3.2: Hierarquia de agentes e fluxos de informações.

Então, são listados os planos para alcançar cada objetivo e quais as informações necessárias para isso. Destaca-se que nem todas as funcionalidades terão listados itens para todas as categorias, e que algumas podem parecer repetição da aplicação ou da própria definição da funcionalidade. De qualquer modo, essas informações explícitas são importantes pois ajudarão a extrair os planos dos agentes.

A Inferência de habitação informa o local e o estado do movimento do habitante quando identifica alguma mudança a quem houver solicitado. Além disso, envia ao BD essa informação, em conjunto com os dados crus de leitura dos sensores. O serviço necessário para essa funcionalidade é um catálogo para se registrar e ser encontrado por quem desejar conhecer a localização do habitante. Seus objetivos são: identificar as informações de habitação e informá-las a quem necessita. Para alcançá-los, os planos do agente correspondente, Habitação, devem envolver os seguintes pontos: leitura das informações dos sensores de presença e de movimentação e análise dos seus sinais para inferência da habitação; envio de mensagens aos agentes solicitantes dessa informação; envio de seus dados ao BD. A estrutura de crenças para essa funcionalidade (quais informações ela necessita) são os sinais dos quatro sensores.

O Controle de iluminação, por outro lado, tem por objetivos antecipar e configurar automaticamente a iluminação do ambiente de acordo com sua inferência de atividade e, também, mantê-la no estado em que o habitante a colocar manualmente, caso este altere o estado de algum dos atuadores. A fim de atingir esses objetivos, são necessárias as informações atuais de habitação. Desse modo, os serviços prestados são a antecipação dos estados dos dispositivos de iluminação e a manutenção dos mesmos caso o usuário os altere. Os planos do correspondente agente são os seguintes:

- quando o habitante alterar o estado de iluminação, não alterá-lo novamente até que alguma condição de contexto mude;
- inferir a próxima atividade com base nas condições do ambiente e alterar o estado dos atuadores conforme necessário;
- enviar seus dados ao BD.

As informações de contexto de que ele precisa são o estado atual da luz e dos métodos de entrada do habitante, o estado de habitação, e, além disso, outras condições de contexto, como horário, dia da semana e histórico de todas as variáveis, que serão repassadas ao seu algoritmo preditivo.

Por último, a Interface com banco de dados oferece o armazenamento de dados para os outros agentes com o objetivo de manter um histórico das informações envolvidas no sistema. Os planos são realizar a comunicação com o serviço de hospedagem de acordo com o respectivo

Comparando o resultado desta modelagem com a estrutura dos agentes de *MavHome*, apresentada em (COOK; YOUNGBLOOD; HEIERMAN, 2003), percebe-se uma certa equivalência: os planos representam a camada de decisão; já os as crenças e os serviços necessitados constituem a de informação; a de comunicação está presente nas trocas de mensagens entre os agentes; por último, os sensores, atuadores e o próprio ambiente de computação onde rodará o SMA constituem a camada física. A Figura 3.3 apresenta essa estrutura para o caso do agente Iluminação.

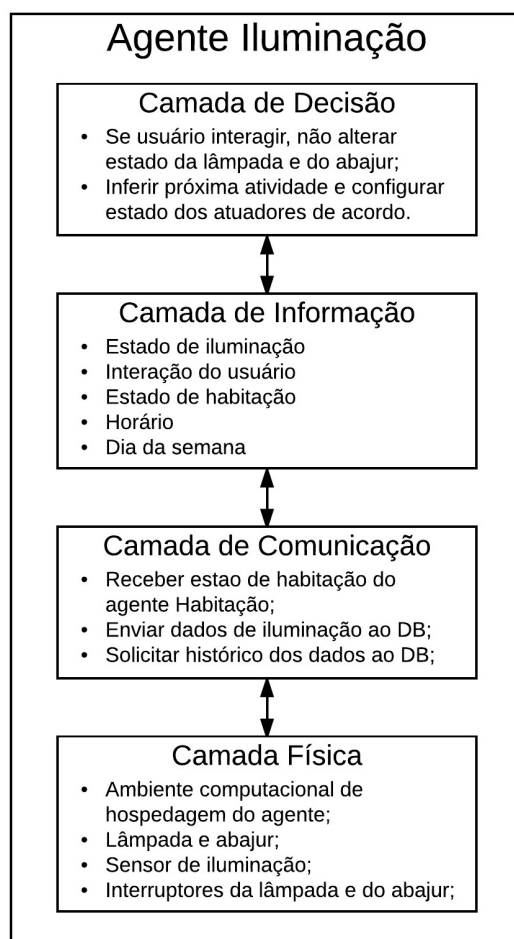


Figura 3.3: Estrutura de camadas do Agente Iluminação.

3.3 Rede Bayesiana para predição de atividades

Seguindo o objetivo de antecipar as interações do usuário com o ambiente, será utilizada a Rede Bayesiana desenvolvida para predição de atividades apresentada no trabalho de (NAZERFARD; COOK, 2012), pois segundo os autores ela mostrou mais precisão do que as demais que foram estudadas. Conforme apresentado, a rede mostrada na Figura 2.4 divide o processo de predição em dois passos. Seguindo esta ideia, e para reduzir o esforço de ensino à rede, ela foi dividida neste trabalho em duas redes A e B, conforme a Figura 3.4. A lógica, entretanto, não se altera: primeiro, dadas as características da atividade atual, são estimadas as da próxima atividade. Por sua vez, esta é então estimada a partir de suas respectivas características em conjunto com a atividade atual, assumindo que esta última seja conhecida.

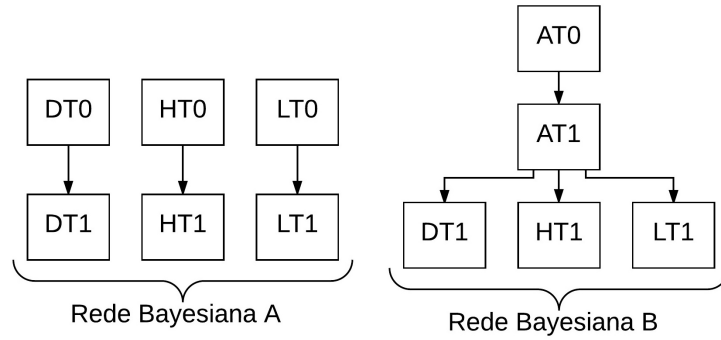


Figura 3.4: Redes Bayesianas A e B.

A nomenclatura dos nós segue o seguinte padrão: DT0 representa a característica Dia da Semana no tempo atual, enquanto que DT1 corresponde à mesma característica mas em relação à atividade futura. As iniciais H, L e A significam, respectivamente, horário, localização e atividade. Os estados que cada variável pode assumir estão representados na Tabela 3.1, sendo que o total de valores distintos de cada uma (m, n, o, p) está relacionado ao estudo de caso de interesse.

Tabela 3.1: Estados das variáveis da Rede Bayesiana.

DT0	$DT = \{d_1, d_2, \dots, d_m\}$
DT1	
HT0	$HT = \{h_1, h_2, \dots, h_n\}$
HT1	
LT0	$LT = \{l_1, l_2, \dots, l_o\}$
LT1	
AT0	$AT = \{a_1, a_2, \dots, a_p\}$
AT1	

As Tabelas 3.2 e 3.3 apresentam como são montadas as tabelas de probabilidades para as variáveis DT0 e DT1 respectivamente. Os números de linhas e de colunas dessas tabelas são determinados pela quantidade de estados diferentes que as variáveis podem assumir e, portanto, dependem da interpretação do contexto da aplicação. As Redes Bayesianas se baseiam em tabelas como essas para realizar os processos de inferência.

Tabela 3.2: Probabilidades de DT0.

d_i	$p(DT0 = d_i)$
d_1	p_{dT0_1}
d_2	p_{dT0_2}
...	...
d_m	p_{dT0_m}

Deste modo, a Rede Bayesiana A utiliza as variáveis respectivas às características no momento atual, cujos valores devem ser conhecidos, para inferir quais os valores que elas mais provavelmente assumirão a seguir. A Equação 3.1 mostra o valor conhecido da característica Dia da Semana atual, enquanto que a Equação 3.2 mostra como é encontrado o valor dessa característica no próximo momento, que assume o valor cuja probabilidade de ocorrência é a maior dentre as possíveis dado o valor conhecido da atual.

Tabela 3.3: Distribuição de probabilidades de DT1.

d_i	$p(DT1 = d_1 DT0 = d_i)$...	$p(DT1 = d_m DT0 = d_i)$
d_1	$p_{dT1_1}^{dT0_1}$...	$p_{dT1_m}^{dT0_1}$
d_2	$p_{dT1_1}^{dT0_2}$...	$p_{dT1_m}^{dT0_2}$
...
d_m	$p_{dT1_1}^{dT0_m}$...	$p_{dT1_m}^{dT0_m}$

$$DT0 = d_x \quad (3.1)$$

$$DT1 = d_y, \text{máx}(p(DT1 = d_y | DT0 = d_x)) \quad (3.2)$$

Para realizar a inferência dos estados dos nós nesta rede, a tabela de distribuição de probabilidades apresenta de maneira direta a esperança de cada estado, pois os nós de interesse se tratam, nos três casos, dos filhos daqueles cujo valor é conhecido. Essa rede realiza portanto uma tarefa simplesmente preditiva.

Realiza-se este procedimento equivalentemente para as outras variáveis que representam as características da atividade futura, e assim são obtidos os valores que a Rede Bayesiana considera mais prováveis para cada uma delas. Estes estados inferidos são assumidos como dados para o cálculo na Rede Bayesiana B.

A tabela de distribuição de probabilidades que relaciona as variáveis AT0 e AT1 é equivalente à Tabela 3.3, pois AT1, a variável de interesse, é descendente de AT0. Entretanto, o parentesco de AT1 com suas características é o contrário, pois nesses casos o nó de interesse passa a ser o antepassado. Assim, a Tabela 3.4 mostra como fica a distribuição de probabilidades para a relação de dependência entre as variáveis AT1 e DT1.

Tabela 3.4: Probabilidades de DT1 sob AT1.

a_i	$p(DT1 = d_1 AT1 = a_i)$...	$p(DT1 = d_m AT1 = a_i)$
a_1	$p_{dT1_1}^{aT1_1}$...	$p_{dT1_m}^{aT1_1}$
a_2	$p_{dT1_1}^{aT1_2}$...	$p_{dT1_m}^{aT1_2}$
...
a_p	$p_{dT1_1}^{aT1_p}$...	$p_{dT1_m}^{aT1_p}$

No caso da inferência do estado da variável AT1 na Rede Bayesiana B, diferentemente de como foi realizada na outra rede, a Tabela 3.4 não fornece a probabilidade de AT1 assumir algum estado dependendo do valor de DT1, mas o contrário, pois AT1 é um antepassado de DT1 no grafo desta Rede. Deste modo, a Rede Bayesiana neste caso mostra a sua utilidade, pois se usa das equações de probabilidades condicionais para realizar inferência diagnóstica, conforme apresentado na Seção 2.2.

No caso desta configuração, a Rede Bayesiana B utiliza propriedades de probabilidades condicionais para calcular a probabilidade de ocorrência de cada um dos estados a_i que AT1 pode assumir, seguindo as evidências de que $AT0 = a_x$, $DT1 = d_x$, $HT1 = h_x$ e $LT1 = l_x$ segundo mostra a Equação 3.3. O estado de AT1 que tiver a maior chance de ser verdadeiro será assumido como a próxima atividade que será realizada.

$$\begin{aligned}
p(AT1 = a_i, AT0 = a_x, DT1 = d_x, HT1 = h_x, LT1 = l_x) = \\
p(AT0 = a_x) \cdot p(AT1 = a_i | AT0 = a_x) \cdot p(DT1 = d_x | AT1 = a_i) \cdot \\
p(HT1 = h_x | AT1 = a_i) \cdot p(LT1 = l_x | AT1 = a_i)
\end{aligned} \tag{3.3}$$

4 ESTUDO DE CASO

Este capítulo aborda os passos seguidos para implementação do sistema. São apresentados os dispositivos automatizados, descrevendo os sensores, os atuadores e como é feita a comunicação com os mesmos. Então, a implementação do SMA é apresentada, incluindo a biblioteca utilizada para isso, bem como o sistema computacional em que fica hospedado. Por último, a aplicação da Rede Bayesiana no SMA é explicada. Todos os códigos desenvolvidos neste trabalho podem ser encontrados em (PROVENZI, 2016).

4.1 Dispositivos Automatizados

Conforme mostra a Figura 3.1, foram criados dois dispositivos automatizados, posicionando um na entrada do quarto e o outro próximo da cama. Nesse ponto, os objetivos foram permitir flexibilidade na comunicação e manter os custos baixos. Para isso, em cada unidade consta um *ESP8266* (em relação às suas posições, respectivamente ESP1 e ESP2), responsáveis pela lógica local de cada dispositivo, pelas interfaces com o habitante e pela comunicação com o SMA. Ambos os *chips* se conectam à rede Ethernet local através da *Wi-Fi*, tendo sido alocado um IP, *Internet Protocol Address*, fixo para cada. Conforme apresentado também na Seção 3.1, cada dispositivo possui conectados aos seus *ESP8266* um sensor de presença, um de movimento e um atuador: no ESP1, uma lâmpada; no ESP2, um abajur, e este também possui um sensor de iluminação ambiente.

Como sensor de presença, estava sendo utilizado um sonar *HC-SR04*. Este dispositivo (cujos sinais operam com nível alto em + 5 V, exigindo a utilização de *level shifters*) emite pulsos ultrassônicos, controla o tempo que cada pulso leva para retornar e mantém sua saída em nível lógico alto por este mesmo período (ELECTRFREAKS, 2016). Assim, no *ESP8266* é possível cronometrar este tempo a fim de se ter uma aproximação da distância percorrida pelo pulso, considerando como ocorrência de presença quando a distância estimada for menor do que a padrão na instalação do sensor (sendo, os respectivos dispositivo, a parede e a cama). Entretanto, foram percebidas instabilidades na saída destes sensores, e não foi possível retirá-las através de filtros pois suas durações frequentemente eram mais longas do que o período em que havia de fato presença à frente do sensor. Desse modo, sua utilização teve de ser descartada.

Como sensor de movimento, foi utilizado o *PIR HC-SR501*, que altera sua saída para nível lógico alto quando o mesmo detecta alguma movimentação, e sua saída permanece assim por um período calibrado em aproximadamente 30 segundos. O elemento receptor deste dispositivo é sensível na faixa de infravermelho e, portanto, não depende da iluminação do ambiente para funcionar (MPJA, 2016). O arco dentro do qual o sensor é sensível é fixo em aproximadamente 110° , e por isto as duas unidades foram posicionadas de modo a não terem seus campos de visão sobrepostos.

Para sensoramento da condição de iluminação do ambiente, foi utilizado um resistor dependente de luz (LDR, do inglês *Light Dependant Resistor*), cuja resistência, como o nome sugere, varia de acordo com a intensidade de luz que incide sobre ele (ELECTRIC4U, 2016). Foi montado um divisor de tensão utilizando o LDR, conectando este entre a tensão de alimentação e o pino de entrada analógica do *ESP8266* e um resistor fixo de 1 k Ω do mesmo pino à referência. Desse modo, podem ser identificadas faixas de leitura que representam diferentes condições de iluminação. Entretanto, neste trabalho

foram apenas consideradas duas situações no ambiente: iluminado ou escuro.

A atuação é feita com o auxílio de relés comuns que operam a 5 VDC e são apropriados para condução de corrente alternada em uma linha de 110 VAC. Para acionar suas bobinas, foram utilizados transistores NPN BC548 para dar ganho de corrente, pois os pinos do *ESP8266* não permitem condução maior do que 12 mA (ESPRESSIF, 2016).

Por último, há duas interfaces de interação do usuário com o ambiente: a fim de que ele possa controlar a lâmpada e o abajur como faria em dispositivos não automatizados, sua entrada física é identificada através de interruptores conectados aos pinos digitais do *ESP8266*; além disso, foi desenvolvido como parte deste trabalho um aplicativo para *SmartPhones*, executável em telefones com sistema operacional *Android*, que se conecta através da rede local a ambos os *ESP8266*. A Figura 4.1 mostra a sua interface com o usuário: há opções para controlar os estados dos atuadores de maneira independente, sempre retornando o estado em que ele se encontra (para que possa ser percebido quando há um erro na comunicação). Este aplicativo permite que o usuário possa, sem se aproximar dos seus interruptores, alterar o estado do abajur ou da lâmpada, como apagar a luz do quarto deitado na cama, por exemplo.

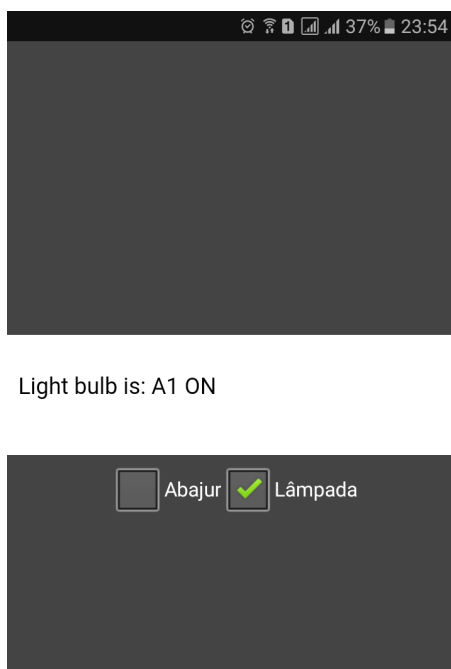


Figura 4.1: Aplicativo de *SmartPhone* criado neste trabalho para controle dos atuadores.

A Figura 4.2 mostra todos os sensores conectados ao *ESP2*, bem como seu atuador: um sensor de posição, um de movimento, um de iluminação, e o abajur. As únicas diferenças em relação ao *ESP1* são que este não possui o LDR e seu atuador é a lâmpada.

Uma das vantagens da utilização do *ESP8266* é que a comunicação com os dispositivos automatizados, conforme mencionado, é feita sem fios através da rede *Wi-Fi*, eliminando a necessidade de distribuir cabos pelo ambiente residencial. Mais do que isso, já que se conecta à rede Ethernet, permite encapsular toda a pilha do protocolo *TCP/IP* (*Transmission Control Protocol*), restando apenas a definição da camada de aplicação.

Conforme já foi apresentado, deseja-se que os *ESP8266* sejam acessados através de seu endereço de IP dentro da rede local. Desse modo, ele foi configurado como um servidor, monitorando os clientes que enviam solicitações a ele. Para isso, foi utilizada uma biblioteca que implementa o protocolo *Hypertext Transfer Protocol* (HTTP) para receber

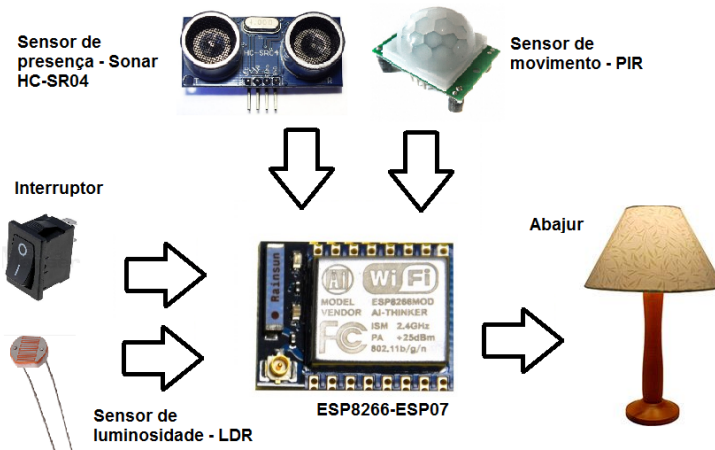


Figura 4.2: Sensores e atuador conectados ao ESP2.

os acessos. Neste caso, os clientes podem ser os agentes, conforme será apresentado na próxima seção, ou o usuário através do aplicativo de *SmartPhone*, ou qualquer sistema computacional conectado na mesma rede que conheça o endereço de IP dos dispositivos. Por questões de segurança de acesso, a comunicação cliente-servidor é validada por senha.

O ciclo de funcionamento das aplicações programadas em cada um dos *ESP8266* é bastante simples. No início de cada iteração, eles realizam a leitura do estado de cada um dos sensores. Caso tenham sido alterados os seus interruptores, os dispositivos alteram os estados dos seus respectivos atuadores. Então, verificam se há algum cliente tentando realizar acesso. Caso positivo, testam a senha e, em caso de sucesso, interpretam o comando. Se este for para definir o estado do atuador, isto é feito. A outra opção é solicitar o estado atual de todos os sensores. Por último, o retorno com as informações para o cliente é dado utilizando-se *HyperText Markup Language* (HTML), de modo que este deve interpretar a mensagem para retirar as informações.

4.2 Implementação do Sistema Multi-Agentes

Uma limitação do *ESP8266* é que ele não oferece capacidade de processamento o suficiente para rodar um ambiente com agentes. Dessa forma, os agentes deverão ser executados em outro sistema computacional. Uma opção de custo relativamente baixo, com bom poder de processamento, versatilidade e facilidade de conexão e de programação é o *Raspberry Pi* (PI, 2016), que, conforme apresentado na Seção 2.3, inclusive já foi utilizado para implementar um Sistema Multi-Agentes em um ambiente residencial (MOERGESTEL et al., 2013). É utilizado o sistema operacional *Raspbian*, pois, dentre outras vantagens, vem nativamente com compiladores de muitas linguagens diferentes instalados.

No trabalho de (MOERGESTEL et al., 2013), é utilizada uma plataforma chamada *JADE* para a implementação e execução do SMA no *Raspberry Pi*. Esta plataforma também é utilizada em (VARGAS, 2016) e (FONSECA, 2011) para trabalhar com agentes. Os três trabalhos destacam como este ambiente é apropriado para essa finalidade, e portanto também é utilizado aqui. Suas características estão descritas na seção 4.3.

Como o armazenamento do *Raspberry Pi* é feito em um cartão de memória *flash*, que possui espaço de armazenamento limitado e risco de se corromper, os dados serão hospe-

dados em um serviço de nuvem. Após uma pesquisa de opções, foi adotado o *ThingSpeak*, que, desenvolvido para incentivar o crescimento da Internet das Coisas, de modo gratuito permite o envio e a recuperação de infinitos dados em muitos canais, restringindo apenas o acesso a cada canal a um período mínimo de 15 segundos (MATHWORKS, 2016). Este servidor possui uma API (*Application Programming Interface*) na linguagem de programação *Java*, o que, conforme será mostrado na seção 4.3, é útil para o trabalho desenvolvido.

Em relação à comunicação dos agentes com seus dispositivos, como serão todos executados dentro de um ambiente no *Raspberry Pi*, caracteriza-se a estrutura de comunicação apresentada na Figura 2.3, e dessa forma a comunicação deve ser feita através da *Wi-Fi* utilizando o protocolo definido para os *ESP8266*. Entretanto, qualquer protocolo passível de implementação na linguagem de programação em que serão executados os agentes também pode ser utilizado.

A Figura 4.3 mostra a conexão entre os dispositivos ESP1 e ESP2 e o *Raspberry Pi*. O roteador define a estrutura da rede Ethernet, atribuindo IPs para os nós conectados a ela (como mencionado, no caso dos *ESP8266* são IPs fixos), montando uma rede local. Também é através dele que se alcança a rede externa, a fim de se acessar o serviço de armazenamento da nuvem. Deste modo, como é comum em redes Ethernet que possuem um roteador, passará por ele toda a comunicação entre dois nós, como *SmartPhone* e *ESP8266*, mesmo que eles estejam fisicamente próximos um do outro. Também estão apresentadas na Figura 4.3 todas as interfaces que o sistema possui com o habitante: os dois interruptores e o *SmartPhone*, que funcionam como entradas de dados, o abajur e a lâmpada.

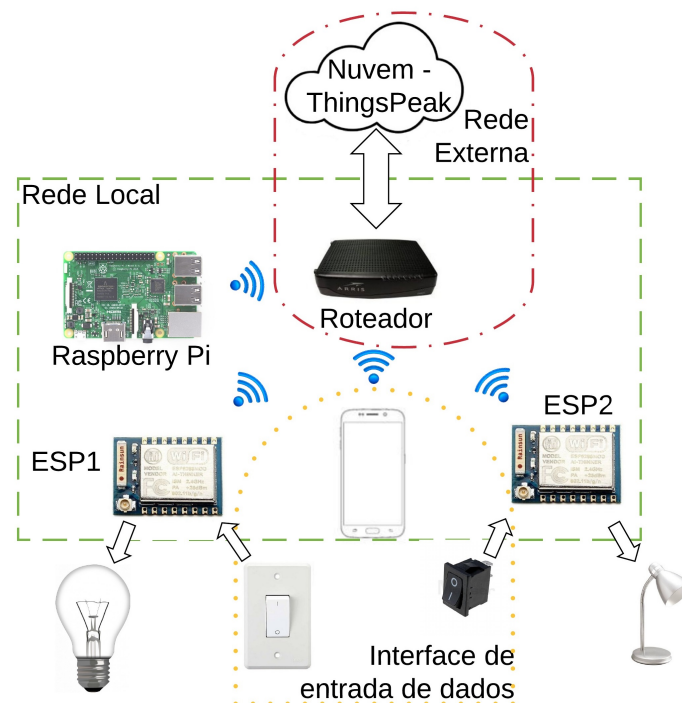


Figura 4.3: Diagrama de interfaces com usuário e entre dispositivos.

4.2.1 JADE

O JADE, *Java Agent DEvelopment Framework*, é, conforme seu nome explicita, um *Framework* utilizado para o desenvolvimento de sistemas multi-agentes. Este *Software* está de acordo com as especificações da FIPA (*Foundation for Intelligent Physical Agents*), o que significa que, dentre outras características, ele implementa a nomenclatura de agentes e sua comunicação de modo a permitir interoperabilidade (FONSECA, 2011). Além disso, o fato de ser baseado em *Java* é uma vantagem pois possibilita a utilização da grande gama de bibliotecas e de APIs desenvolvidos nesta linguagem para funcionalidades como, por exemplo, comunicação com servidores na nuvem.

A execução dos agentes em JADE é baseada em *containers*, sendo que em cada um destes podem estar contidos mais do que um agente. Uma característica importante de JADE é que ele permite a criação de um único ambiente SMA distribuído em múltiplas plataformas, desde que exista comunicação entre elas, conforme apresentado na Figura 4.4. Cada plataforma possui um *container* principal, em que existem sempre o *Agent Management System*, responsável por gerenciar o ciclo de vida de cada agente, e o *Directory Facilitator*, que permite que os diferentes agentes registrem seus serviços e procurem pelos já registrados, e também pode possuir *containers* distribuídos em diferentes computadores. Deste modo, caracteriza-se como uma opção que supre as necessidades de um ambiente inteligente. (FONSECA, 2011)

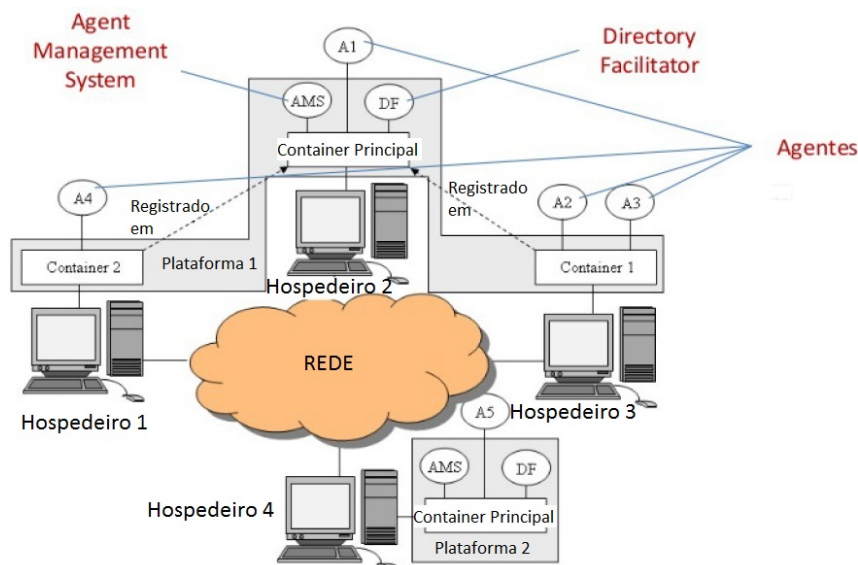


Figura 4.4: Arquitetura distribuída do JADE. Fonte: (FONSECA, 2011).

No JADE, a execução de cada agente se dá através da implementação de uma série de classes de comportamentos (*behaviours*), podendo principalmente ser cíclicas (*CyclicBehaviour* e *TickerBehaviour*) ou de execução única (*OneShotBehaviour*). Desse modo diferentes ações podem ser executadas continuamente, com uma determinada frequência ou a partir de algum gatilho. (FONSECA, 2011)

Em relação à comunicação entre os agentes, este *Framework* fornece uma abordagem interessante que permite que o agente filtre as mensagens que chegam a si a fim de extrair apenas as que são de seu interesse, podendo descartar as demais. Isto é feito através de instâncias da classe *Templates*, que pode ser configurado de acordo com tipo de mensagem, remetente, identificação da conversa, dentre outros. As mensagens seguem a definição *Agent Communication Language*, definida pela FIPA. Esta linguagem descreve,

dentre outras características, os tipos de mensagem, que podem ser *inform*, para envio de informação, *request*, para solicitações, *refuse*, para recusas, etc. (FONSECA, 2011)

Ainda em relação a métodos de comunicação, o JADE possui uma função de *Subscription* implementável, de modo que um agente pode solicitar a outro uma determinada informação, e este pode aceitar, fornecendo-a quando necessário, ou recusar o pedido. Com isso, o envio de dados é automático nas devidas condições (MOERGESTEL et al., 2013).

4.2.2 Implementação do Agente Habitação

O agente Habitação possui principalmente três ações: ler as informações dos sensores de presença e de movimento e inferir o estado de habitação do ambiente; enviar essa informação à sua lista de solicitantes; atualizar o banco de dados com as novas informações que foram identificadas. A Tabela 4.1 lista os planos referentes a estas ações comparando-os com os comportamentos que este agente terá implementados no JADE, incluindo nome e tipo. O comportamento do tipo *SubscriptionResponder* é um dos oferecidos para o serviço de *Subscription* mencionado, sendo este o que possui os métodos referentes ao envio da mensagem ao solicitante. A implementação em *Java* destes comportamentos e dos dos outros agentes podem ser encontrados em (PROVENZI, 2016).

Tabela 4.1: Relação dos planos e dos comportamentos do agente Habitação.

Plano	Comportamento
Leitura de sensores e inferência de habitação	<i>ReadmySensors – TickerBehaviour</i>
Envio da habitação aos solicitantes	<i>SubscriptionBehaviourResp - SubscriptionResponder</i>
Envio das informações ao banco de dados	<i>updateDB – TickerBehaviour</i>

4.2.3 Implementação do Agente Iluminação

Já o agente Iluminação tem seus comportamentos definidos a partir dos seus planos conforme apresentado na Tabela 4.2. Nela, percebe-se que a ação de respeitar as alterações feitas pelo usuário e a predição de atividade são implementada no laço principal de execução deste agente.

Tabela 4.2: Relação dos planos e dos comportamentos do agente Iluminação.

Plano	Comportamento
Respeitar alterações do usuário	<i>LightAgentLoop – TickerBehaviour</i>
Predição de atividade	<i>LightAgentLoop - TickerBehaviour</i>
Envio das informações ao banco de dados	<i>updateDB – TickerBehaviour</i>

Vale destacar que, embora não esteja presente nos planos, este agente possui um comportamento *SubscriptionBehaviourInit*, do tipo *SubscriptionInitiator*, que procura junto ao *Directory Facilitator* pelos agentes que fornecem o serviço de que necessita (neste caso, estimativa de habitação), e envia então uma solicitação de *Subscription*. Este comportamento também implementa o método que lida com as mensagens enviadas pelo respectivo *SubscriptionResponder* (MOERGESTEL et al., 2013).

4.2.4 Implementação do Agente Banco de Dados

Por último, o agente banco de dados tem seus comportamentos definidos a partir dos seus planos conforme apresentado na Tabela 4.3. Destaca-se que a comunicação com a nuvem é feita respeitando-se o período mínimo de 15 segundos entre cada envio de dados.

Tabela 4.3: Relação dos planos e dos comportamentos do agente BD.

Plano	Comportamento
Comunicação com a nuvem	<i>DBAgentLoop – TickerBehaviour</i>

4.2.5 Agente ESP8266

Ao longo da implementação dos agentes, foi identificada uma limitação no *ESP8266*: utilizando períodos menores do que 600 milissegundos entre os acessos pela rede de *Wi-Fi* a cada um destes dispositivos, constatou-se que ambas as unidades reiniciam-se após alguns minutos de execução. Suspeita-se que, por estarem confinados em caixas fechadas e sem resfriamento forçado, eles possam sofrer sobreaquecimento causado pelas frequentes conexões por *Wi-Fi*, tarefa que aumenta a demanda de corrente.

De qualquer modo, este problema não foi analisado em detalhes, pois não está no foco deste trabalho a alta frequência de acessos. Como apenas aumentar o período entre as conexões para em torno de 700 milissegundos faz com que não ocorram reinicializações, e como este intervalo de tempo não prejudica o desenvolvimento deste trabalho, esta foi justamente a solução adotada. Deste modo, mantendo-se nos agentes Iluminação e Habitação os métodos de leitura dos sensores, cada um destes teria que atualizar seus sensores com períodos de mais de um segundo, para que a combinação dos dois não resultasse em acessos demasiadamente frequentes.

Neste caso, seria um período alto demais para uma aplicação em que se deseja realizar antecipação às ações do usuário, pois aumentam as chances de ocorrerem casos em que o ele toma intervém sobre o ambiente antes de o sistema identificar as alterações de contexto, como por exemplo acender a luz na entrada do habitante no ambiente. Por isso, a fim de centralizar a leitura dos sensores de cada dispositivo em apenas um acesso ao mesmo, foi criado um agente chamado de *ESP8266*, responsável simplesmente por realizar as leituras no menor período de operação estável (700 milissegundos) e, então, repassar aos outros agentes suas respectivas informações.

Essa alteração no sistema multi-agentes implica somente em pequenas alterações nos agentes Habitação e Iluminação: ao invés do acesso aos seus meios físicos ser feito através de seus comportamentos, as rotinas que tratam os dados foram adaptadas para serem métodos que são executados na chegada de uma mensagem vinda do agente *ESP8266*.

4.3 Predição de atividade

No contexto deste trabalho, são definidas as seguintes atividades do habitante no ambiente identificáveis pela combinação de leitura de sensores e de outras informações: ausente; dormindo; perambulando; lendo; variados. Boa parte delas são auto-explicativas. Entretanto, vale destacar que a nomeada por perambulando representa a ação de levantar-se e sair do quarto no meio da noite. A Tabela 4.4 apresenta como deve ser a condição de

iluminação em cada uma destas atividades, e a caracterização dos atuadores no ambiente segue-a após o agente realizar a predição.

Tabela 4.4: Condição de iluminação do ambiente para cada atividade.

Atividade	Sensor Iluminação	Lâmpada	Abajur
Dormindo	Escuro	Apagada	Apagado
Perambulando	Apagada	TickerBehaviour	Aceso
Lendo	Iluminado	Indiferente	Aceso
Variados	Iluminado	Acesa	Indiferente
Ausente	Indiferente	Apagada	Apagado

Até aqui não foi esclarecido como é inferido o estado da variável Atividade Atual (AT0), apenas assumido que este valor é conhecido. De fato, essa variável não é inferida pelas Redes Bayesianas. Desta forma, foram utilizadas regras que verificam os valores de variáveis do contexto do trabalho, como condição de iluminação e estados dos atuadores, além das utilizadas na Rede Bayesiana, como localização, horário e dia, para inferir de maneira estática a atividade atual. Por exemplo, a regra abaixo mostra qual a condição em que se assume que o habitante está dormindo.

se o quarto estiver escuro **e**
o usuário estiver ausente ou na cama **e**
tratar-se de um dia da semana **e** o horário estiver entre 0 e 9 horas,
ou tratar-se de um dia de final de semana e o horário estiver entre 0 e 12 horas,
então o habitante está **dormindo**.

Em relação às outras variáveis da Rede Bayesiana, seus valores foram definidos dentro do contexto deste trabalho como se segue. O horário, ao contrário do apresentado no estudo de (NAZERFARD; COOK, 2012), divide o dia em intervalos de 3 horas, para haver um melhor encaixe na rotina do habitante do ambiente: 0h-3h, 3h-6h, 6h-9h, e assim sucessivamente até 21h-24h. Em relação ao dia da semana, este apenas recebe os valores de “dia de semana” ou de “final de semana”. Por último, a localização pode ser “inativo”, “cama”, “entrada” ou “ambos”, sendo que este representa que ambos os sensores de movimento estão com suas saídas ativas.

A fim de implementar a predição com Redes Bayesianas no agente desenvolvido na plataforma JADE, procurou-se por bibliotecas em *Java* que auxiliassem a sua criação e utilização. A única opção gratuita encontrada se chama *BayesFusion*, e esta apresentou o problema de necessitar de bibliotecas compartilhadas que a tornam dependente da arquitetura do processador do sistema. Como ela não possui uma versão que execute em processadores ARM, caso do *Raspberry Pi*, e seu código fonte não está disponibilizado para que seja compilada novamente, ela não pode ser utilizada (BAYESFUSION, 2015).

É então utilizada uma biblioteca na linguagem *Python* que implementa Redes Bayesianas e permite sua utilização para inferências específicas de acordo com evidências. Chamada de *libpgm* (LIBPGM, 2012), ela pode ser utilizada no *Raspberry Pi* sem problemas. Deste modo, o *script* de *Python* deverá ser executado a partir do agente implementado em *Java* e de modo que seja possível a troca de informações.

Esta biblioteca é de simples utilização: cria-se a Rede Bayesiana em um arquivo texto, seguindo um determinado padrão, na qual estão contidas as informações de nós, valores das variáveis (estados), dependências e distribuições de probabilidades. A biblioteca então disponibiliza métodos que permitem interpretar este arquivo texto como uma

Rede Bayesiana. Assim, podem ser feitas diferentes avaliações estatísticas. Por exemplo, definir os valores de determinadas variáveis para calcular a probabilidade de algumas outras assumirem determinados valores (ou seja, inferência a partir de evidências).

Para realizar a tarefa de inferência de atividade proposta neste trabalho, foram criados então dois arquivos texto, que caracterizam as Redes Bayesianas A e B apresentadas na Seção 3.3, e um *script* que utiliza a biblioteca. O programa em *Python* recebe como argumentos quais são os valores atuais de cada uma das 4 variáveis (DT0, HT0, AT0, LT0) e carrega ambas Redes Bayesianas. Então, inicialmente, a rede A infere a partir dos valores atuais das características de atividade, e independentemente entre si, quais são seus valores futuros. A seguir, ela define estes valores como a evidência para a rede B, utilizando também a atividade atual para inferir qual será a próxima atividade. Por último, ela retorna os estados de todas essas variáveis.

A execução do agente em JADE recebe o retorno das Redes Bayesianas e, a partir do valor retornado como de maior probabilidade para a próxima atividade, configura os estados dos atuadores de iluminação de acordo com a Tabela 4.4. Ou seja, caso a próxima atividade inferida seja Dormindo, então o agente irá desligar a lâmpada e o abajur; ou, caso o estado seja Lendo, ele verifica o estado de iluminação, acendendo o abajur caso o quarto esteja escuro.

Os dados estatísticos das distribuições de probabilidade das Redes Bayesianas aplicadas foram alimentados pelo desenvolvedor, utilizando-se de sua experiência, assim como mencionado na Seção 2.2. A Tabela 4.5 mostra a estatística que relaciona os estados das variáveis localização atual com futura. As outras tabelas das Redes Bayesianas se encontram em Apêndice. Entretanto, essas informações foram comparadas com as probabilidades condicionais recalculadas através de uma população de mais de duas mil entradas extraídas do banco de dados, e, em relação à Tabela 4.5, a diferença para o utilizado ficou abaixo de 5 % para cada valor. Os resultados das demais variáveis foram também semelhantes.

Tabela 4.5: Distribuição de probabilidades entre as variáveis LT0 e LT1.

l_i	$p(\text{LT1}=\text{Inativo} \mid \text{LT0} = l_i)$	$p(\text{LT1}=\text{Cama} \mid \text{LT0} = l_i)$	$p(\text{LT1}=\text{Entrada} \mid \text{LT0} = l_i)$	$p(\text{LT1}=\text{Ambos} \mid \text{LT0} = l_i)$
Inativo	0,4	0,3	0,2	0,1
Cama	0,3	0,4	0,1	0,2
Entrada	0,3	0,1	0,4	0,2
Ambos	0,1	0,2	0,4	0,3

Para o agente Iluminação respeitar as alterações do usuário e não insistir em colocar os atuadores em um estado indesejado em um determinado momento, antes de realizar a automação dos atuadores foi adicionada uma *flag* que bloqueia esta ação sempre que tiver ocorrido alguma interação do habitante com o ambiente. Para desbloquear a automação, esta *flag* tem seu valor desabilitado quando é alterado o estado de algum dos seguintes nós da Rede Bayesiana: AT0, DT0, HT0, LT0. A ideia por trás disto é que este agente Iluminação deve voltar a ter liberdade para atuar sobre os dispositivos quando o habitante passa a exercer outra atividade, ou quando ele se movimenta pelo quarto, ou quando já passou algum tempo desde que ele interagiu (características de hora e dia).

5 RESULTADOS E DISCUSSÃO

De maneira qualitativa, o objetivo de antecipação foi alcançado, pois o ambiente é capaz de configurar os estados dos atuadores conforme o habitante o faria na maioria dos cenários: quando ele está ausente, ambos ficam desligados, bem como enquanto dorme; quando se levanta durante a madrugada, apenas o abajur se acende; o mesmo vale quando o habitante se deita na cama durante o dia para ler; à noite, ao entrar no quarto, a lâmpada se acende e permanece assim durante sua estadia.

Porém, há comportamentos indesejados identificados no sistema: ao longo da manhã e da tarde, se o habitante ficar apenas na entrada do quarto, o ambiente não acende a luz pró-ativamente; quando o usuário decide dormir fora de horário, durante o dia, o ambiente tenta acender o abajur por acreditar que a pessoa deseja ler. O primeiro desses erros acontece pois a Rede Bayesiana estima, incorretamente, que mesmo que o habitante recém tenha entrado no quarto, ele logo irá sair. O segundo é causado por um limite na instrumentação, pois apenas com o sensor de movimento não é possível discernir qual atividade o usuário desempenha enquanto está na cama, e durante o mencionado horário o mais provável seria a atividade de leitura.

Ainda, a utilização de apenas sensores de movimento apresenta também outro problema: quando o habitante está imóvel por algum tempo, o sistema é levado a crer que ele não está mais presente. Isto se manifestou principalmente na atividade de leitura: enquanto o usuário está deitado à cama em horário em que provavelmente não estaria dormindo, o sistema desliga o abajur, acendendo-o novamente quando a pessoa se movimenta, repetindo essa sequência por repetidas vezes.

Como o principal objetivo do trabalho é reduzir o número de interações do habitante com os dispositivos automatizados no ambiente, foi configurada uma variável, também salva no banco de dados, que é ativada cada vez que há uma troca no estado de algum dos atuadores causada pelo usuário. A Tabela 5.1 compara a quantidade de vezes que essa variável (aqui chamada de Interações) acusou uma intervenção do usuário sobre o ambiente com a automação dos dispositivos ligada, ligando ou desligando algum dos atuadores através das predições das Redes Bayesianas, e desligada. O período de medição foi de quatro dias, com dias de semana e de final de semana.

Tabela 5.1: Quantidade de interações do habitante em um período de 4 dias.

Automação	Desligada	Ligada
Interações	65	28

Percebe-se que a automação da lâmpada e do abajur utilizando uma Rede Bayesiana reduziu em aproximadamente 56,9 % o número de interações do habitante com o ambiente. A fim de ajudar a entender o motivo dessas intervenções do usuário, a Tabela 5.2 separa quantas intervenções foram feitas em cada intervalo de horário, com e sem a automação. A coluna Entradas indica o número de envios ao banco de dados dentro de um dado intervalo, originadas por alteração em qualquer uma das variáveis contextuais do agente Iluminação. Deste modo, a Proporção indica a relação entre quantas interações o usuário fez sobre o ambiente com a quantidade de vezes que o sistema percebeu alguma alteração em seu contexto.

Tabela 5.2: Horários das interações do habitante em um período de 4 dias.

Automação	Desligada	Ligada		
Horário	Interações	Interações	Entradas	Proporção
0h às 3h	4	3	31	9,7 %
3h às 6h	5	0	2	0,0 %
6h às 9h	0	1	60	1,7 %
9h às 12h	0	2	106	1,9 %
12h às 15h	8	5	69	7,2 %
15h às 18h	13	0	10	0,0 %
18h às 21h	15	12	65	18,5 %
21h às 24h	20	5	68	7,4 %

Comparando o número de interações com a automação ligada e desligada, percebe-se que a antecipação do sistema ao usuário apresentou um bom desempenho, reduzindo consideravelmente o número de interações, em três períodos: das 3h às 6h, o habitante deve ter levantando-se durante a madrugada e não precisou acender nenhuma luz; das 15h às 18h e das 21h às 24h, períodos em que ele costumaria entrar no quarto para ler ou para alguma atividade diversa, ele também não precisou alterar o estado dos atuadores um grande número de vezes.

Já o período das 0h às 3h e o das 12h às 15h não apresentaram uma grande diferença de interações, enquanto que nos períodos das 6h às 9h e das 9h às 12h houve interações que não haviam ocorrido sem a automação. Entretanto, por serem poucas em número, não indicam alguma tendência. O período de pior desempenho foi o das 18h às 21h, pois quase 20 % das mudanças no contexto do ambiente foram causadas por interações do habitante. O próximo parágrafo entra em detalhes do que ocorreu neste horário.

Foi constatado que a implementação utilizada para bloqueio da automação não funciona quando o usuário deita-se na cama para dormir fora do horário padrão, pois o sistema se comporta da seguinte maneira: percebendo sua localização na cama fora do horário da madrugada, o agente Iluminação acende o abajur; o habitante, então, apaga-a; após o atraso do sensor de movimento, o bloqueio percebe que a característica de localização foi alterada, liberando a automação; após isso, quando o habitante se movimentar novamente, o abajur novamente é acendido, pois a Rede Bayesiana volta a inferir que a pessoa está realizando atividade de leitura. Esta sequência de acontecimentos ocorreu em três oportunidades, todas elas das 18h às 21h, resultando em 9 interações do usuário com o ambiente. Caso o bloqueio da automação tivesse permanecido ativo enquanto a pessoa estava deitada, teria havido 6 interações a menos (21,4 % do total) e a redução de interações do habitante com o ambiente teria chegado a 66,2 %.

Em relação ao que foi mencionado de o sistema perder a localização do habitante quando este fica imóvel por algum tempo, apagando a lâmpada ou o abajur, reacendendo algum desses quando ele se movimenta, a Figura 5.1 apresenta um intervalo de aproximadamente 6,5 minutos em que houve 10 alterações no estado do abajur realizadas pela automação. Esse tipo de comportamento ocorreu várias vezes ao longo dos quatro dias, principalmente em situações em que o habitante se encontrava lendo deitado na cama no horário das 9h às 12h, sendo este o principal motivo do grande número de entradas ao banco de dados neste período, mostrado na Tabela 5.2.

A Tabela 5.3 apresenta qual foi o percentual de acerto na predição de cada variável das Redes Bayesianas A e B. Percebe-se que o índice de acertos nas predições do dia da

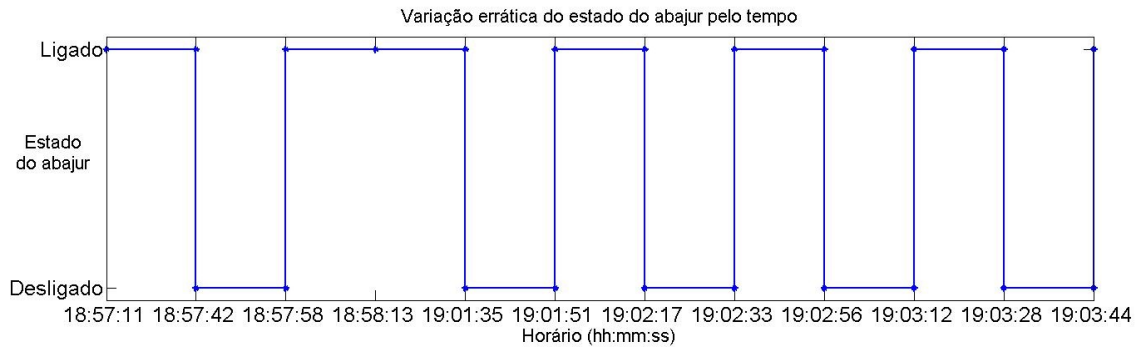


Figura 5.1: Frequentes alterações no abajur por perda de localização do habitante.

semana e do horário em que ocorrem a atividade seguinte foram altíssimos. Entretanto, esse resultado deve-se parcialmente ao fato de que, ao mesmo tempo, cada intervalo temporal é bastante abrangente, e, assim, a maior probabilidade é de que a próxima atividade ocorra na mesma faixa de tempo.

Por outro lado, o desempenho da variável referente à próxima localização foi bastante baixo, estando correta apenas uma a cada cinco vezes, aproximadamente. Este número é mais um indicativo do problema que é inferir a posição de um habitante utilizando apenas sensores de movimento, pois, como essa leitura varia bastante, acaba dificultando a tarefa de predição.

O baixo desempenho da variável localização acaba por reduzir a possibilidade de sucesso da predição de atividade. Assim, a predição do que o habitante estava prestes a fazer ficou com desempenho de 50,4 %. Vale destacar que este resultado está na mesma ordem de grandeza da redução do número de interações (56,9 %), o que indica que, quando ocorreu a correta predição, a tendência foi o ambiente configurar os seus atuadores de acordo com o desejo do habitante.

Tabela 5.3: Percentuais de acertos das predições das diferentes variáveis.

Variável	Índice de Acertos
DT1 – Dia da Semana	99,8 %
HT1 – Horário	96,0 %
LT1 – Localização	19,9 %
AT1 – Atividade	50,4 %

Os resultados do trabalho em que as Redes Bayesianas foram baseadas, conforme apresentado na Seção 2.2.1, foram respectivamente de 97,0 %, 84,2 %, 52,8 % e 70,3 %. Comparando-os com os mostrados na Tabela 5.3, percebe-se que, embora sejam numericamente diferentes, há uma certa relação entre eles: a predição do dia da semana e do horário são as que possuem maior índice de acertos, nesta ordem, sendo que a localização é a que apresenta o mais baixo resultado. Em relação à predição da próxima atividade, em ambos os casos ela apresenta valores intermediários em relação às outras variáveis.

Em relação ao Sistema Multi-Agentes, tendo obtido sucesso na implementação dos agentes modelados Iluminação, Habitação, BD e, devido à necessidade, *ESP8266*, os principais resultados a serem analisados tangem a questão temporal, devido às diferentes comunicações que acontecem. É importante que o sistema seja capaz de perceber alterações das condições do ambiente e atuar de acordo de uma maneira que não seja demasiadamente demorada. Caso contrário, o habitante não irá ficar aguardando, e ele

mesmo realizará as alterações nos estados dos atuadores. Ou seja, um aumento no tempo de atuação implica em um maior número de interações do habitante com o ambiente.

As medições realizadas no presente trabalho mostraram que estes atrasos ficaram entre 0,5 e 1,5 segundos, tempo que é aceitável para que o habitante não tenha que intervir sobre o ambiente. Os principais limitantes deste desempenho são o *chip ESP8266*, pois conforme apresentado exigiu que as conexões a ele fossem feitas a cada no máximo 700 ms, e a rede Ethernet, que faz com que ocasionalmente o acesso aos dispositivos para atuação seja demorada (na ordem de 1 segundo). A troca de mensagens entre os agentes no SMA não implicou em uma parcela significativa deste resultado, pois as trocas de mensagens mostraram levar apenas algumas dezenas de milissegundos para execução. Este resultado talvez fosse diferente caso os agentes estivessem rodando em plataformas separadas fisicamente. Por último, também foi constatado que, apesar da necessidade de carregar arquivos, a execução das Redes Bayesianas foi bastante rápida, na ordem de unidades de milissegundos, de modo que também não foi um fator relevante neste atraso.

6 CONCLUSÕES E TRABALHOS FUTUROS

O avanço da tecnologia aplicado à automação residencial tem fortalecido o conceito de Ambientes Inteligentes, de modo que a tendência é que os dispositivos automatizados possuam cada vez mais autonomia e capacidades cognitivas e de comunicação nos próximos anos. Inserido nesse contexto, este trabalho apresentou a modelagem de um Ambiente Inteligente por um Sistema Multi-Agentes que faz uso de Redes Bayesianas para antecipar-se ao seu habitante nas interações deste com os dispositivos automatizados.

Para o estudo de caso, foram utilizados sensores e atuadores acessíveis pela rede local através da *Wi-Fi*, tendo sido desenvolvido também um aplicativo para *Android* que possibilita as interações do usuário com o ambiente à distância. O Sistema Multi-Agentes e as Redes Bayesianas ficaram concentrados em um *Raspberry Pi*, que se comunica com os dispositivos pela rede Ethernet através do roteador.

Os resultados alcançados foram considerados satisfatórios, pois o sistema reduziu em 56,9 % o número de interações do habitante com a lâmpada e o abajur. Foi constatado que os horários que apresentaram melhor desempenho da automação foram: das 3h às 6h, horário comum para ir ao banheiro à noite; das 15h às 18h, comum para leitura e atividades diversas; e das 21h às 24h, também comumente possuindo atividades diversas.

Entretanto, foi constatado que, com a utilização apenas de sensores de movimento, a inferência da localização do habitante no ambiente apresentou comportamento instável e prejudicou no desempenho geral da predição de atividades. Foi percebido que uma melhor estimativa de onde a pessoa se encontra é muito importante para o desempenho da lógica preditiva.

Foi percebido que o uso de Redes Bayesianas da forma estudada ainda faz com que a automação seja de certa forma reativa, pois em geral as ações são apenas tomadas após a identificação da presença do habitante. Esta abordagem não seria muito útil em aplicações em que há uma constante de tempo envolvida à atuação, como por exemplo controle de temperatura, pois não possibilita que a atuação seja realizada algum tempo antes da chegada do habitante no ambiente.

Por último, foi constatado que a utilização da rede Ethernet local, especificamente através de *Wi-Fi* e com o protocolo TCP/IP, não resulta em atrasos grandes o suficiente para impossibilitar sua utilização no âmbito de automação residencial em tarefas que não exijam alta resolução temporal. Entretanto, em aplicações de tempo real, ou quaisquer outras nas quais o tempo seja uma variável crítica, esta não parece uma opção adequada.

Como trabalhos futuros, são feitas as seguintes sugestões:

- Implementar a utilização de sensores que detectam presença além dos de movimento, como câmeras, para inferência da localização do habitante e melhorar as condições de desbloqueio da automação;
- Implementar na Rede Bayesiana um sistema de aprendizado, para que possam ser atualizados periódica e dinamicamente os parâmetros das DCPs partir de dados, e a possibilidade de programar atuações para momentos futuros;
- Estudar outros períodos ou a utilização de uma variável contínua para representar o horário na Rede Bayesiana, e também acrescentar uma relação de dependência da localização com esta variável;

REFERÊNCIAS BIBLIOGRÁFICAS

- ALAM, M. A. U.; ROY, N.; MISRA, A.; TAYLOR, J. CACE Exploiting Behavioral Interactions for Improved Activity Recognition in Multi-Inhabitant Smart Homes. , [S.l.], p.539–548, 2016.
- ALESSANDRA, P.; PIERLUCA, F.; SALVATORE, G.; GIUSEPPE, R.; SAJAL, D. An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments. , [S.l.], v.1233, n.c, p.1–14, 2016.
- APPLE. “**Siri**”. Disponível em <<http://www.apple.com/ios/siri/>>. Acessado em 06/11/2016.
- BAYESFUSION. “**Academic Use of our Products**”. Disponível em <<http://www.bayesfusion.com/academic-users/>>. Acessado em 14/10/2016.
- BUSSAB, W. O.; MORETTIN, P. A. **Estatística Básica**. 6^a.ed. [S.l.]: Saraiva, 2010.
- CALLAGHAN, V.; CLARKE, G.; COLLEY, M.; HAGRAS, H.; CHIN, J. S. Y.; DOCTOR, F. Inhabited intelligent environments. **BT Technology Journal**, [S.l.], v.22, n.3, p.233–247, 2004.
- COOK, D. J.; DAS, S. K. How Smart are our Environments? An Updated Look at the State of the Art The Role of Physical Components in Smart Environments. **Journal of Pervasive and Mobile Computing**, [S.l.], v.3, p.53–73, 2007.
- COOK, D.; YOUNGBLOOD, M.; HEIERMAN, E. MavHome: an agent-based smart home. **Proceedings of the First IEEE International Conference On Pervasive Computing And Communications**, [S.l.], p.521 – 524, 2003.
- ELECFREAKS. “**Ultrasonic Ranging Module HC - SR04**”. Disponível em <<http://www.micropik.com/PDF/HCSR04.pdf>>. Acessado em 17/09/2016.
- ELECTRICAL4U. “**Light Dependent Resistor — LDR and Working Principle of LDR**”. Disponível em <<http://http://www.electrical4u.com/light-dependent-resistor-ldr-working-principle-of-ldr/>>. Acessado em 17/09/2016.
- ESPRESSIF. “**ESP8266EX Datasheet**”. Disponível em <<https://espressif.com/>>. Acessado em 17/09/2016.
- FONSECA, M. L. **Framework para construção e análise de sistemas de gestão de energia elétrica para consumidores de baixa tensão em redes elétricas inteligentes**. 2011. Tese (Doutorado em Ciência da Computação) — .
- GOOGLE. “**Meet the Google App**”. Disponível em <<https://www.google.com/search/about/>>. Acessado em 06/11/2016.
- GROKHOTKOV, I. “**ESP8266 core for Arduino**”. Disponível em <<https://github.com/esp8266/Arduino/>>. Acessado em 17/09/2016.

HEIERMAN III, E. O.; COOK, D. J. Improving home automation by discovering regularly occurring device usage patterns. **Data Mining, 2003. ICDM 2003. Third IEEE International Conference on**, [S.l.], p.537–540, 2003.

HSUEH, Y.; LIN, N.; CHANG, C.; LIE, W. Abnormal Event Detection Using Bayesian Networks at a Smart Home. , [S.l.], p.273–277, 2015.

LIBPGM. “**Welcome to libpgm!**”. Disponível em
<<http://pythonhosted.org/libpgm/index.html>>. Acessado em 29/10/2016.

MATHWORKS. “**Getting Started with ThingSpeak**”. Disponível em
<<https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>>. Acessado em 23/10/2016.

MOERGESTEL, L. V.; LANGERAK, W.; MEERSTRA, G.; NIEUWENBURG, N. V.; PAPE, F.; PUIK, E.; MEYER, J.-j. Agents in domestic environments. , [S.l.], 2013.

MOZER, M. C. An intelligent environment must be adaptive. **IEEE Intelligent Systems**, [S.l.], p.11–13, 1999.

MPJA. “**HC-SR501 PIR MOTION DETECTOR**”. Disponível em
<<https://www.mpja.com/download/31227sc.pdf>>. Acessado em 17/09/2016.

NAZERFARD, E.; COOK, D. J. Bayesian Networks Structure Learning for Activity Prediction in Smart Homes. , [S.l.], 2012.

NEST. “**Nest Thermostat**”. Disponível em
<<https://nest.com/thermostat/meet-nest-thermostat/>>. Acessado em 06/11/2016.

PEIXOTO, J. A. Manufatura Usando Arquiteturas Orientadas a Serviço e Sistemas Multi-Agentes. , [S.l.], 2012.

PI, R. “**RASPBERRY PI 3 MODEL B**”. Disponível em
<<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acessado em 25/09/2016.

PROVENZI, M. “**Smart Environment Project**”. Disponível em
<<https://github.com/mprovenzi/SmartEnvironmentProject>>. Acessado em 09/11/2016.

RUGGERI, F.; KENETT, R.; FALTIN, F. Bayesian Networks. In: **Encyclopedia of Statistics in Quality and Reliability**. [S.l.]: John Wiley & Sons, 2008.

VARGAS, M. F. d. Proposta de um sistema de assistência personalizada para ambientes inteligentes. , [S.l.], 2016.

WOOLDRIDGE, M. **An introduction to MultiAgent Systems**. [S.l.]: John Wiley & Sons, 2002.

ZHENG, H.; WANG, H.; BLACK, N. Human Activity Detection in Smart Home Environment with Self-Adaptive Neural Networks. , [S.l.], p.1505–1510, 2008.

APÊNDICE

As tabelas a seguir apresentam as distribuições de probabilidades das variáveis presentes nas Redes Bayesianas. Para a rede A, estão listadas as tabelas de LT0, DT0 e HT0, bem como de seus respectivos descendentes (à exceção de DT1, já mostrada no texto), LT1 e HT1. Para a B, também estão mostradas a LT1, a HT1 e a DT1, além da AT0 e da AT1.

Tabela 6.1: Distribuição de probabilidades da variável LT0 na Rede Bayesiana A.

l_i	$p(LT0 = l_i)$
Inativo	0,5
Cama	0,2
Entrada	0,2
Ambos	0,1

Tabela 6.2: Distribuição de probabilidades da variável DT0 na Rede Bayesiana A.

d_i	$p(DT0 = d_i)$
Dia de Semana	0,7
Final de Semana	0,3

Tabela 6.3: Distribuição de probabilidades da variável HT0 na Rede Bayesiana A.

h_i	$p(HT0 = h_i)$
0h a 3h	0,1
3h a 6h	0,1
6h a 9h	0,2
9h a 12h	0,05
12h a 15h	0,05
15h a 18h	0,1
18h a 21h	0,2
21h a 24h	0,2

Tabela 6.4: Distribuição de probabilidades da variável HT1 na Rede Bayesiana A.

d_i	$p(\text{HT1} = 0\text{h a } 3\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 3\text{h a } 6\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 6\text{h a } 9\text{h} \mid \text{HT0} = h_i)$
0h a 3h	0,7	0,2	0,1
3h a 6h	0,0	0,7	0,2
6h a 9h	0,0	0,0	0,7
9h a 12h	0,0	0,0	0,0
12h a 15h	0,0	0,0	0,0
15h a 18h	0,0	0,0	0,0
18h a 21h	0,1	0,0	0,0
21h a 24h	0,2	0,1	0,0
d_i	$p(\text{HT1} = 9\text{h a } 12\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 12\text{h a } 15\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 15\text{h a } 18\text{h} \mid \text{HT0} = h_i)$
0h a 3h	0,0	0,0	0,0
3h a 6h	0,1	0,0	0,0
6h a 9h	0,2	0,1	0,0
9h a 12h	0,7	0,2	0,1
12h a 15h	0,0	0,7	0,2
15h a 18h	0,0	0,0	0,7
18h a 21h	0,0	0,0	0,0
21h a 24h	0,0	0,0	0,0
d_i	$p(\text{HT1} = 18\text{h a } 21\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 21\text{h a } 24\text{h} \mid \text{HT0} = h_i)$	
0h a 3h	0,0	0,0	
3h a 6h	0,0	0,0	
6h a 9h	0,0	0,0	
9h a 12h	0,0	0,0	
12h a 15h	0,1	0,0	
15h a 18h	0,2	0,1	
18h a 21h	0,7	0,2	
21h a 24h	0,0	0,7	

Tabela 6.5: Distribuição de probabilidades da variável DT1 na Rede Bayesiana A.

d_i	$p(\text{DT1} = \text{Dia de Semana} \mid \text{DT0} = d_i)$	$p(\text{DT1} = \text{Final de Semana} \mid \text{DT0} = d_i)$
Dia de Semana	0,8	0,2
Final de Semana	0,3	0,7

Tabela 6.6: Distribuição de probabilidades da variável AT0 na Rede Bayesiana B.

a_i	$p(\text{AT0} = a_i)$
Dormindo	0,3
Perambulando	0,25
Lendo	0,15
Variados	0,1
Ausente	0,2

Tabela 6.7: Distribuição de probabilidades da variável AT1 na Rede Bayesiana B.

a_i	$p(AT1 = \text{Dormindo} \mid AT0 = a_i)$	$p(AT1 = \text{Perambulando} \mid AT0 = a_i)$	$p(AT1 = \text{Lendo} \mid AT0 = a_i)$
Dormindo	0,6	0,05	0,05
Perambulando	0,1	0,6	0,2
Lendo	0,2	0,3	0,4
Variados	0,2	0,2	0,1
Ausente	0,25	0,1	0,05

a_i	$p(AT1 = \text{Variados} \mid AT0 = a_i)$	$p(AT1 = \text{Ausente} \mid AT0 = a_i)$
Dormindo	0,15	0,15
Perambulando	0,05	0,05
Lendo	0,05	0,05
Variados	0,35	0,15
Ausente	0,2	0,4

Tabela 6.8: Distribuição de probabilidades da variável DT1 na Rede Bayesiana B.

a_i	$p(DT1 = \text{Dia de Semana} \mid AT1 = a_i)$	$p(DT1 = \text{Final de Semana} \mid AT1 = a_i)$
Dormindo	0,6	0,4
Perambulando	0,4	0,6
Lendo	0,5	0,5
Variados	0,3	0,7
Ausente	0,25	0,75

Tabela 6.9: Distribuição de probabilidades da variável LT1 na Rede Bayesiana B.

a_i	$p(LT1 = \text{Inativo} \mid AT1 = a_i)$	$p(LT1 = \text{Cama} \mid AT1 = a_i)$
Ausente	0,65	0,05
Dormindo	0,35	0,35
Perambulando	0,1	0,2
Lendo	0,35	0,35
Variados	0,3	0,1

Ausente a_i	$p(LT1 = \text{Entrada} \mid AT1 = a_i)$	$p(LT1 = \text{Ambos} \mid AT1 = a_i)$
Ausente	0,15	0,15
Dormindo	0,1	0,2
Perambulando	0,35	0,35
Lendo	0,15	0,15
Variados	0,3	0,3

Tabela 6.10: Distribuição de probabilidades da variável HT1 na Rede Bayesiana B.

d_i	$p(\text{HT1} = 0\text{h a } 3\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 3\text{h a } 6\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 6\text{h a } 9\text{h} \mid \text{HT0} = h_i)$
Ausente	0,05	0,05	0,15
Dormindo	0,2	0,2	0,1
Perambulando	0,2	0,35	0,15
Lendo	0,15	0,05	0,05
Variados	0,1	0,05	0,15
d_i	$p(\text{HT1} = 9\text{h a } 12\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 12\text{h a } 15\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 15\text{h a } 18\text{h} \mid \text{HT0} = h_i)$
15Ausente	0,25	0,2	0,15
Dormindo	0,05	0,05	0,1
Perambulando	0,05	0,05	0,05
Lendo	0,1	0,1	0,15
Variados	0,1	0,1	0,1
d_i	$p(\text{HT1} = 18\text{h a } 21\text{h} \mid \text{HT0} = h_i)$	$p(\text{HT1} = 21\text{h a } 24\text{h} \mid \text{HT0} = h_i)$	
Ausente	0,1	0,05	
Dormindo	0,15	0,15	
Perambulando	0,1	0,1	
Lendo	0,2	0,2	
Variados	0,2	0,2	