

CENTRO UNIVERSITÁRIO FEI
GUSTAVO MOLINA FIGUEIREDO

**ANÁLISE DE MÉTODOS PARA EXTRAÇÃO DE PADRÕES SEQUENCIAIS DE
COMPORTAMENTO EM CASAS INTELIGENTES**

São Bernardo do Campo

2017

GUSTAVO MOLINA FIGUEIREDO

**ANÁLISE DE MÉTODOS PARA EXTRAÇÃO DE PADRÕES SEQUENCIAIS DE
COMPORTAMENTO EM CASAS INTELIGENTES**

Dissertação de Mestrado apresentada ao Centro
Universitário FEI, como parte dos requisitos
necessários para obtenção do título de Mestre em
Engenharia Elétrica. Orientado pelo Prof. Dr.
Flávio Tonidandel.

São Bernardo do Campo

2017

Molina Figueiredo, Gustavo.

Análise de métodos para extração de padrões sequenciais de comportamentos em casas inteligentes / Gustavo Molina Figueiredo. São Bernardo do Campo, 2017.

93 f. : il.

Dissertação - Centro Universitário FEI.

Orientador: Prof. Dr. Flávio Tonodandel.

I. Extração de Padrões Sequenciais. 2. Casa Inteligente. 3. Automação Residencial. 4. Algoritmos. I. Tonodandel, Flávio, orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

Aluno: Gustavo Molina Figueiredo

Matrícula: 115105-9

Título do Trabalho: Análise de métodos para extração de padrões sequenciais de comportamento em casas inteligentes.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Flavio Tonidandel

Data da realização da defesa: 22/03/2017

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, 22 / 03 / 2017.

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. Flavio Tonidandel Ass.: _____

Prof. Dr. Plinio Thomaz Aquino Junior Ass.: _____

Prof. Dr. Leandro Augusto da Silva Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

VERSÃO FINAL DA DISSERTAÇÃO

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

A Deus, por me permitir viver esta experiência, à
minha família e a todos que me apoiaram nesta
empreitada.

AGRADECIMENTOS

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitário, mas que em todos os momentos é o maior mestre que alguém pode conhecer. Agradeço também pelos momentos difíceis, pois com eles me torno cada vez mais forte para enfrentar qualquer obstáculo sem medo.

À minha mãe, que sempre se preocupou, incondicionalmente, com seus três filhos, presença constante com seu jeito solidário, compreensivo e incentivador.

À minha família por todo amor, carinho, incentivo e principalmente pelo tempo concedido e pela compreensão, sem os quais a jornada seria bem mais difícil.

À minha namorada, melhor amiga e companheira de todas as horas, Cibele Nascimento, pelo carinho, compreensão e solidariedade.

A todos meus amigos, que compreenderam minha ausência em várias ocasiões nesta etapa da minha vida.

A todos os meus colegas do mestrado, pelos momentos de estudo, companheirismo e felicidade que passamos juntos durante esta jornada.

Aos meus companheiros de trabalho, que me proporcionaram inúmeros dias de ausência da empresa, para a elaboração deste trabalho e compreenderam tamanha a minha preocupação em realiza-lo.

Ao professor Flávio Tonidandel, pela amizade, orientação, apoio e incentivo no trato do tema.

Aos professores Dr. Plinio Thomaz Aquino Junior e Dr. Leandro Augusto da Silva, pelos conselhos e correções, apresentados em minha banca de qualificação, que auxiliaram na conclusão desta dissertação.

A todos os professores que proporcionaram minha formação acadêmica, desde a alfabetização até a graduação.

A todos que, de uma forma ou outra, contribuíram para a realização deste trabalho.

" O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis."

José de Alencar

RESUMO

Uma casa inteligente, também conhecida como *smart home*, possui diversos tipos de sistemas que envolvem desde dispositivos simples como os destinados para controle de iluminação, até os mais complexos, como dispositivos inteligentes que interagem com os moradores, e gradativamente novas tecnologias estão surgindo, convergindo para uma nova realidade em relação aos sistemas presentes em ambientes automatizados. Neste contexto, os moradores buscam mais conforto e uma melhor qualidade no interior da residência.

Para aumentar o conforto e qualidade de vida torna-se necessário o desenvolvimento de um sistema que se adapte ao comportamento dos habitantes levando-se em conta que cada habitante possui um comportamento (sequência de ações) diferente. Para que a casa inteligente aprenda de forma autônoma o comportamento de seus habitantes, é necessário que ela conheça os padrões de utilização de cada um.

Este trabalho apresenta um estudo sobre três algoritmos (GSP, SPADE e PrefixSpan) utilizados para a extração de padrões sequenciais. Serão apresentados os conceitos fundamentais de cada um e posteriormente será explicado como serão implementados para extrair os padrões sequenciais em casas inteligentes.

Após realização dos testes, notou-se que a quantidade de dados analisada influenciou no resultado dos algoritmos. Quando poucos dados foram analisados, o sistema aprendeu pouco sobre o comportamento do habitante. Conforme o número de dados analisados aumentou, o sistema aprendeu mais sobre o comportamento do habitante.

Com base nos resultados obtidos, concluiu-se que para o *dataset* utilizado neste trabalho, o melhor método para extração de padrões sequenciais de comportamento em casas inteligentes é o algoritmo PrefixSpan.

Palavras-chave: Extração de Padrões Sequenciais. Casa Inteligente. Automação Residencial. Algoritmos.

ABSTRACT

A Smart Home has different types of systems which involve simple devices like those provided for lighting control, up to the most complex, as smart devices that interact with the inhabitants, and gradually new technologies are emerging, converging on a new reality. In this context, the inhabitants are looking for more comfort and a better quality within the residence.

To increase the comfort and life quality of inhabitants, it is necessary to develop a system that adapts itself to the inhabitant's behavior taking into account that each inhabitant has a different sequence of actions (behavior). For the smart home to learn autonomously the behavior of its inhabitants it's necessary to understand the usage pattern of each one.

This work presents a study of three algorithms (GSP, SPADE, PrefixSpan) used for the extraction of sequential patterns. The fundamental concepts of each algorithm will be presented and later it will be explained how they are going to be implemented in order to extract the sequential patterns in smart homes.

After the tests, it was noticed that the amount of data analyzed influenced on the result of the algorithms. When a few data were analyzed, the system learned a little about the inhabitant's behavior. As the number of the data analyzed increased, the system learned more about the inhabitant's behavior.

Based on the results obtained, it was concluded that considering the dataset used in this work, the PrefixSpan algorithm is the best method of sequential patterns extraction in smart homes.

Keywords: Sequential Patterns Extraction. Smart Home. Residential Automation. Algorithms.

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVO	14
1.2 MOTIVAÇÃO	14
1.3 ORGANIZAÇÃO DO TRABALHO.....	15
2 REFERENCIAL TEÓRICO	16
2.1 CASAS INTELIGENTES	16
2.2 DOMÓTICA.....	24
2.3 INTEGRAÇÃO DE SISTEMAS RESIDENCIAIS	27
2.4 RECONHECIMENTO DE PADRÕES	28
2.5 AUTOMAÇÃO RESIDENCIAL INTELIGENTE	30
2.6 RELAÇÕES TEMPORAIS	31
3 TRABALHOS RELACIONADOS.....	33
4 ALGORITMOS DE DESCOBERTA DE PADRÕES DE SEQUÊNCIAS	39
4.1 MINERAÇÃO DE ITEMSETS FREQUENTES	39
4.2 ALGORITMO APRIORI.....	41
4.3 ALGORITMOS PARA EXTRAÇÃO DE PADRÕES DE SEQUÊNCIA	46
4.3.1 Estratégia de geração e teste de candidatos	47
4.3.2 O algoritmo GSP	47
4.3.3 O algoritmo SPADE.....	49
4.3.4 Estratégia de crescimento de padrões	51
4.3.5 O algoritmo PrefixSpan.....	52
5 ANÁLISE DA EXTRAÇÃO DE COMPORTAMENTOS SEQUENCIAIS	54

5.1 HOUSE_N.....	54
5.1.1 PlaceLab	54
5.1.2 Recursos do PlaceLab	56
5.2 DESCRIÇÃO DA BASE DADOS.....	57
5.3 ANÁLISE DOS ALGORITMOS	59
6 TESTES REALIZADOS	62
6.1 DATASET UTILIZADO PARA AS SIMULAÇÕES.....	63
6.2 HARDWARE UTILIZADO NOS TESTES.....	64
6.3 AVALIAÇÃO DA PRECISÃO DA EXTRAÇÃO	64
6.4 AVALIAÇÃO DA QUALIDADE DAS EXTRAÇÕES.....	76
7 CONCLUSÃO	82
7.1 TRABALHOS FUTUROS	83
7.1.1 Teste do sistema com outros datasets, com características e quantidades de habitantes diferentes	83
7.1.2 Uso de outros parâmetros no software para extração de padrões sequenciais	84
7.1.3 Teste do sistema em uma residência real	84
APÊNDICE A – LOCAIS E QUANTIDADES DE SENSORES INSTALADOS	85
APÊNDICE B – TIPOS, MECANISMOS E FUNCIONALIDADES DOS SENSORES INSTALADOS NO PLACELAB	86
APÊNDICE C – TIPOS DE SENSORES POSSÍVEIS A SEREM INCORPORADOS FUTURAMENTE	87
APÊNDICE D – DISPONIBILIDADE DE HARDWARE E DADOS	88
APÊNDICE E – ATIVIDADES REALIZADAS PELO HABITANTE DE 01/09/2007 A 14/09/2007	89
REFERÊNCIAS	90

1 INTRODUÇÃO

Atualmente, os habitantes das casas inteligentes, também conhecidas como *smart homes*, se interessam por mais conforto e uma melhor qualidade de vida no interior da residência. Visando o conforto é coerente afirmar que os sistemas mais eficientes são aqueles que se adaptam ao comportamento dos habitantes, no lugar dos habitantes terem de se adaptar ao sistema.

Segundo Weiser (1993, apud BOLZANI, 2010), uma casa inteligente é definida como uma "habitação incorporando uma rede de comunicações que conecta aparelhos elétricos e serviços que podem ser controlados, monitorados ou acessados remotamente". A definição continua declarando benefícios que uma casa inteligente pode ter, como por exemplo, a gestão da energia, ou controlar remotamente o sistema de aquecimento e resfriamento da casa para obter máxima eficiência de acordo com o número de habitantes na casa.

O projeto de uma casa inteligente depende das necessidades e dos estilos de vida dos habitantes. Geralmente, casas inteligentes oferecem conforto, segurança, controle remoto, e conservação de energia (BOLZANI, 2010).

Casas Inteligentes dependem de equipamentos e dispositivos de aquisição de dados para monitorar e posteriormente aprender os hábitos de cada habitante. Com as tecnologias que, nos dias atuais, já permitem um nível razoável de integração, a ideia de ter controle sobre certos equipamentos e algum nível de automação, visando comodidade e conforto do usuário, torna-se cada vez mais real (FUSSEK, 2010).

Baseado nas considerações acima, torna-se desejável o desenvolvimento de sistemas de fácil utilização pelos habitantes da residência (a complexidade dos controles dos sistemas presentes na casa com a automação não pode ser maior do que sem o uso dela) e, com o passar do tempo, cada vez mais adaptativos, evoluindo para o conceito de domótica inteligente, onde as regras de automação adaptam-se ao comportamento dos moradores (TONIDANDEL; TAKIUCHI; MELO, 2004).

Com base em estudos realizados pela empresa digitalSTROM, Baumann et al. (2012) mostra que é possível economizar energia levando em consideração o conforto dos habitantes de uma casa inteligente.

Para aumentar o conforto dos habitantes e reduzir o consumo de energia em uma casa inteligente, é necessário que ela seja capaz de aprender o comportamento de seus habitantes e

consiga prever suas necessidades futuras aplicando regras de automação. Um exemplo de tal regra poderia ser: *Se a luz foi acesa no quarto, as venezianas foram abertas e a luz foi acesa no banheiro, a luz do quarto deve ser apagada*. Um estudo realizado pela empresa CKW (2013) comprova que uma especificação manual de tais regras não é desejável devido ao elevado grau de interação e envolvimento dos habitantes.

Para que a regra de automação citada anteriormente possa ser aplicada, é necessário que a casa inteligente saiba as preferências e os padrões de comportamento de cada habitante. Os padrões de comportamento são essenciais para que a casa inteligente aprenda o que constitui um comportamento normal. Para reconhecer padrões, a casa inteligente analisa os dados de eventos advindos dos sensores (sensores de proximidade, vibração, movimento, temperatura, pressão, aceleração, luz e mudança de estado) instalados em objetos do dia a dia, tais como armários, gavetas, geladeira, portas, eletrodomésticos, conhecidos como *HVAC* (dispositivos de aquecimento e arrefecimento), sistemas de iluminação e das persianas (HARPER, 2003). Quando a análise dos dados termina, a mineração desses dados é feita a fim de encontrar padrões frequentes (padrões que ocorrem mais frequentemente do que outros) e padrões periódicos (padrões que ocorrem em um intervalo constante).

No contexto desse trabalho, um padrão é uma sequência de atividades realizada pelo habitante. Considerando – se uma casa inteligente onde o habitante acorda, acende a luz do quarto, abre a persiana, sai do quarto, entra no banheiro e acende a luz, temos um padrão constituído por três atividades / eventos:

- a) Acender a luz quarto;
- b) Abrir a persiana;
- c) Acender a luz do banheiro.

Se a casa inteligente sabe todos os padrões periódicos e frequentes de seus habitantes, pode prever as próximas ações e obter os melhores resultados na automatização de tarefas, dessa forma é possível economizar energia, tem uma segurança maior e evitar acidentes envolvendo pessoas com deficiência ou necessidades especiais.

Um algoritmo adequado, ou seja, de extração de padrões, permite que uma casa inteligente aprenda de forma autônoma os padrões de utilização (comportamento) dos seus habitantes.

Diversos algoritmos foram implementados para identificar sequências frequentes em bancos de dados de sequências. Uma das abordagens mais utilizadas para identificar sequências frequentes é a abordagem *Apriori*. De acordo com Agrawal e Srikant (1994), o algoritmo *Apriori* foi originalmente concebido com o objetivo de associar regras de mineração de dados e é baseado na propriedade *apriori* que diz que para um item ser frequente, todos os seus subconjuntos também devem ser.

Existem três algoritmos considerados bem-sucedidos na extração de padrões sequenciais: GSP (*Generalized Sequential Patterns*) introduzido em 1996 por Agrawal e Srikant, SPADE (*Sequential Pattern Discovery using Equivalence classes*) proposto por Zaki (2001) e PrefixSpan (*Prefix-projected Sequential pattern mining*) introduzido por Pei et al. (2001).

No contexto de uma casa inteligente, a sequência de atividades (eventos) é muito importante quando os algoritmos estão analisando os dados gerados de cada habitante. Para aprender os hábitos individuais, é fundamental saber se o eletrodoméstico X foi ligado antes ou depois do eletrodoméstico Y.

Nas próximas seções, são apresentados o objetivo, motivação e organização deste trabalho.

1.1 Objetivo

O objetivo deste trabalho é analisar e comparar o desempenho dos algoritmos GSP, SPADE e PrefixSpan na extração de padrões sequenciais em casas inteligentes e verificar a viabilidade de usá-los para extrair autonomamente o comportamento de seus habitantes.

1.2 Motivação

A motivação pela escolha deste tema baseia-se na grande utilidade que um sistema eficaz de extração de padrões sequenciais de comportamento pode ter em diferentes gamas de aplicações.

Neste trabalho, o sistema foi desenvolvido voltado à aplicação em um sistema de automação residencial inteligente, onde diferentes ações devem ser tomadas dentro da residência.

Esta aplicação, por si só, já justificaria o desenvolvimento do sistema, já que trará um considerável acréscimo ao nível de conforto dos habitantes de uma residência, que poderiam sentir-se incomodados por ações tomadas por regras de automação que não condizem com seus hábitos dentro dos ambientes. Estes níveis de conforto não são medidos, neste momento, por não serem o foco do trabalho.

Neste trabalho foi considerada apenas a extração de padrões sequenciais de comportamento de uma casa inteligente com um único habitante, mas um desenvolvimento futuro deste poderia contemplar casas inteligentes com dois ou mais habitantes. Assim, o sistema poderia ter aplicações comerciais extremamente úteis, como desenvolvimento de novos produtos para casas inteligentes de acordo com o comportamento do indivíduo identificado, ou para fins estatísticos, quanto ao perfil de comportamento de cada habitante que frequenta mais determinados cômodos da casa, a fim de adaptá-los a estes, entre outras aplicações.

1.3 Organização do trabalho

Este trabalho é dividido da seguinte forma:

Na seção 2, é exposto o referencial teórico relacionado com a pesquisa desenvolvida neste trabalho.

Na seção 3, foram expostos os trabalhos relacionados ao tema de extração de padrões sequenciais e, por fim, foram mostrados alguns trabalhos já desenvolvidos na área de automação residencial inteligente, que motivaram o desenvolvimento deste trabalho.

Na seção 4, são apresentados os algoritmos de extração de padrões sequenciais, explicando-se detalhadamente o seu funcionamento e parâmetros.

Na seção 5, é apresentada a formalização da proposta onde o *dataset* utilizado é apresentado e detalhado bem como o *software Smart Homer Pattern Miner* criado para os testes da lógica proposta para o sistema.

Na seção 6, são expostos os dados e parâmetros utilizados nas simulações realizadas no *software*, assim como gráficos e tabelas contendo os resultados das simulações, com suas análises e conclusões.

Na seção 7, são apresentadas as considerações finais sobre o trabalho e sugeridos alguns trabalhos futuros acerca deste tema.

2 REFERENCIAL TEÓRICO

A automação residencial vem tornando-se, com o passar dos anos, um assunto de interesse de um número cada vez maior de pessoas (BOLZANI, 2010). As casas inteligentes, também conhecidas como *smart homes*, já apresentam tecnologias que, nos dias atuais, permitem habitações mais confortáveis, visto que já permitem um nível razoável de integração e automação. Os sistemas de automação residencial, em sua maioria, têm como objetivo facilitar a interação dos habitantes com os itens controlados e monitorados por eles, como iluminação, climatização, entre outros (FUSSEK, 2010). Isto é conseguido por meio de “uma rede de comunicação que conecta os [...] aparelhos elétricos” (DOUNIS E CARAISCOS, 2009).

Este capítulo visa apresentar, de forma sucinta, conceitos importantes relacionados a este trabalho.

Serão mostradas, de forma resumida, algumas iniciativas acadêmicas e corporativas sobre residências-laboratório e automação residencial. Por fim, serão apresentados alguns trabalhos sobre mineração de padrões e detecção de anomalias baseados em padrões sequenciais.

A partir da próxima seção, são detalhados os itens expostos nesta introdução.

2.1 Casas Inteligentes

Segundo Weiser (1993, apud BOLZANI, 2010), o conceito de Casa Inteligente surgiu no início do século XX e está fundamentalmente ligado ao uso da tecnologia para melhorar a qualidade de vida das pessoas no ambiente residencial.

Na literatura, não existe ainda uma definição precisa e amplamente aceita, no entanto, existe um pensamento comum das funções que uma casa inteligente deve desempenhar: ela deve automatizar algumas tarefas rotineiras para aliviar o trabalho de seus habitantes; ser mais segura; proporcionar uma autonomia maior; aumentar o nível de conforto e utilizar de uma maneira eficaz o uso dos recursos energéticos por meio da integração (BOLZANI, 2010).

Nugent et al. (2008) enfatiza o aumento da qualidade de vida:

O objetivo de uma casa inteligente é monitorar as atividades de uma pessoa dentro de seu próprio ambiente de vivência e como ela interage com dispositivos domésticos, e baseado nessas interações e na atual sequência de atividades, o ambiente pode ser

controlado e adaptado para lhe proporcionar uma melhor experiência em seu dia-a-dia. Esta melhoria de vida pode ser alcançada provendo um maior nível de autonomia pessoal e de vida independente, o aumento da qualidade de vida ou o aumento da sensação de segurança.

Casas inteligentes oferecem uma melhor qualidade de vida por meio da introdução do controle automatizado de aparelhos e serviços de acessibilidade. O habitante de uma casa inteligente pode controlar eletrodomésticos e dispositivos remotamente, o que permite a execução das tarefas antes de chegar à casa (HELAL, 2005).

Outros autores dão uma ênfase maior à tecnologia em si:

Uma casa não é inteligente porque foi bem construída, nem por quão eficaz foram usados seus espaços; nem porque é sustentável, usando energia solar ou reaproveitamento de água, por exemplo. Uma casa inteligente de fato inclui essas coisas, mas o que a torna inteligente são as tecnologias interativas que ela contém (HARPER, 2003).

De acordo com Norbistrath et al. (2005), as residências inteligentes oferecem, por meio da combinação de seus equipamentos eletrônicos, avançados benefícios a seus ocupantes.

Analisando-se as definições acima, pode-se dizer que para a infraestrutura de uma *smart home*, destacam-se três principais ramos de aplicação: Segurança, Economia e Conforto.

Pensando na segurança, temos como exemplo as câmeras sensíveis ao movimento, capazes de enviar *e-mails* e transmitir em tempo real movimentações estranhas detectadas em horário inadequado. Há algoritmos de rastreamento cada vez mais modernos, capazes de ajustar o *zoom* e focar no rosto das pessoas que transitam pelo ambiente monitorado.

No campo da economia, verifica-se que o acionamento e desligamento automático dos aparelhos proporciona uma maior eficiência energética, pois, utiliza-se a energia elétrica apenas pelo tempo necessário, reduzindo o desperdício.

Visando o conforto dos moradores, a casa inteligente busca adaptar-se do modo mais natural possível ao dia-a-dia dos usuários. É possível, por exemplo, programar o aparelho de som para ligar pela manhã com um volume mais baixo que aumenta aos poucos, enquanto as cortinas do quarto se abrem.

A seguir, serão apresentadas algumas iniciativas acadêmicas sobre casas inteligentes e automação residencial.

Em Boulder, *University of Colorado*, a partir de 1991, foi desenvolvido o projeto “*Adaptive House*”, pelo professor Michael Mozer. Trata-se de uma casa inteligente que foi elaborada usando um sistema de redes neurais para controlar temperatura ambiente, aquecedor de água e iluminação, sem a programação prévia do usuário. Esse sistema, chamado de ACHE

(*Adaptive Control of House Environment*), visa economizar a energia respeitando o estilo de vida do usuário. ACHE monitora de forma constante o ambiente, observando todas as ações feitas pelos residentes. Por meio dos dados coletados por sensores remotos, padrões são registrados, processados e aprendidos, de forma que a casa possa se programar sozinha.

Segundo Mozer (2005), a *Adaptive House* possui mais de cinco quilômetros de condutores e 75 sensores, que monitoram inclusive portas e janelas, conforme Figura 1.

Se algum desconforto é notado em relação a um evento pré-programado, é dado um retorno de informação ao sistema, de forma que da próxima vez que o evento for acessado, o processo ocorra de forma mais adequada. Um exemplo dessa aplicação é feito quando o sistema percebe a presença de um usuário em determinado ambiente e controla a potência da luz. Caso o usuário interfira no processo, mudando a potência, ele reprograma seus dados para “acertar” da próxima vez, tentando seguir os hábitos do usuário em questão (MOZER, 2005).

Figura 1 - Planta da Adaptive House com a localização dos sensores.



Fonte: Autor “adaptada de” Mozer, 1998.

No Georgia *Institute of Technology*, em 1998, um grupo de pesquisadores interdisciplinares montou o projeto *Aware Home Research Initiative*, uma casa inteligente que funciona como um laboratório de desenvolvimento e avaliação para tecnologias domésticas (HARPER, 2003). O objetivo do projeto é criar um ambiente que perceba e ajude o usuário

quando o mesmo chegar à idade avançada, no sentido de aumentar a qualidade de vida, aumentando sua independência (HARPER, 2003).

A casa possui sensores no piso para diferenciar os passos dos diferentes habitantes, criando módulos baseados em seus comportamentos, utilizando modelos matemáticos e redes neurais. Nesse laboratório, muitos objetos foram mapeados com RFID (*radio frequency identification* – identificação por radiofrequência), de forma que praticamente todos os objetos essenciais possam ser localizados e reconhecidos pelo sistema.

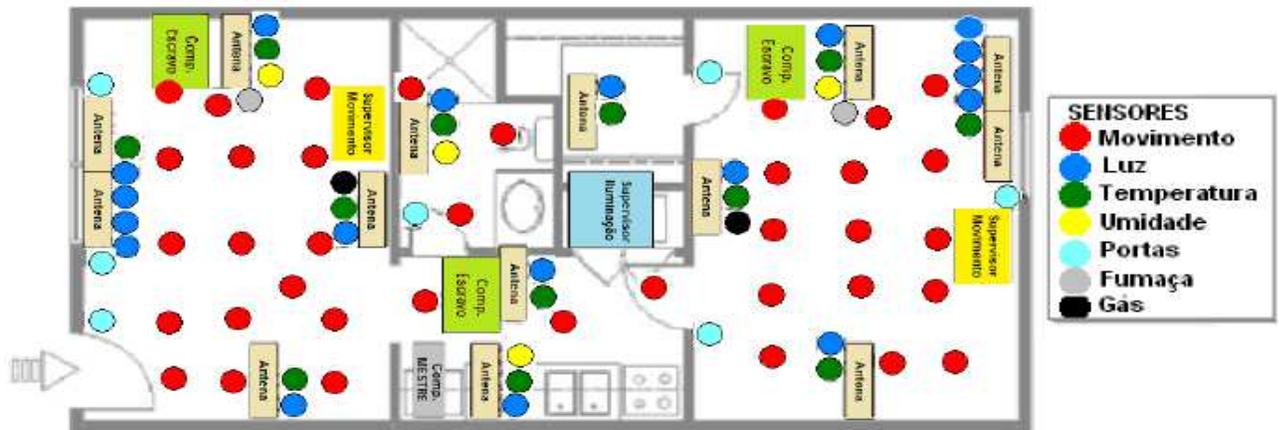
A interação com o usuário é feita via painéis de LCD (*liquid crystal display*) e sistemas de áudio e vídeo, de forma que os comandos também possam ser feitos por interpretação de falas e gestos. O projeto também apresenta robôs para dar assistência na manipulação motora, bem como pesquisas envolvendo entretenimento e sustentabilidade, investigando como novas tecnologias podem impactar na vida das pessoas nas casas.

Na *University of Texas*, em Arlington, foi idealizado em 2000 o projeto *MavHome* (*Managing an Adaptive Versatile Home*) com o objetivo de criar uma casa que age como um agente racional, tentando maximizar o conforto de seus habitantes, enquanto minimiza seus custos operacionais e provê segurança (BOLZANI, 2010). O sistema deve ser capaz de sentir e prever os hábitos dos ocupantes e o uso dos eletrodomésticos. O método é chamado de LeZi, um modelo de previsão probabilística de gerenciamento de conforto e uso dos eletrodomésticos. Esse sistema calcula a probabilidade de cada ação possível que ocorra na sequência observada do movimento, baseada em ações passadas e predizendo as mesmas (BOLZANI, 2010).

O projeto *MavHome* faz uso de várias tecnologias, como: robótica, inteligência artificial e banco de dados. O controle faz a automação de todas as luzes do ambiente e dos eletrodomésticos, assim como os sistemas de refrigeração e de aquecimento, ventiladores e persianas.

A Figura 2 ilustra a planta da casa e a localização dos sensores espalhados estrategicamente pelos cômodos.

Figura 2 – Planta da MavHome com a localização dos sensores.



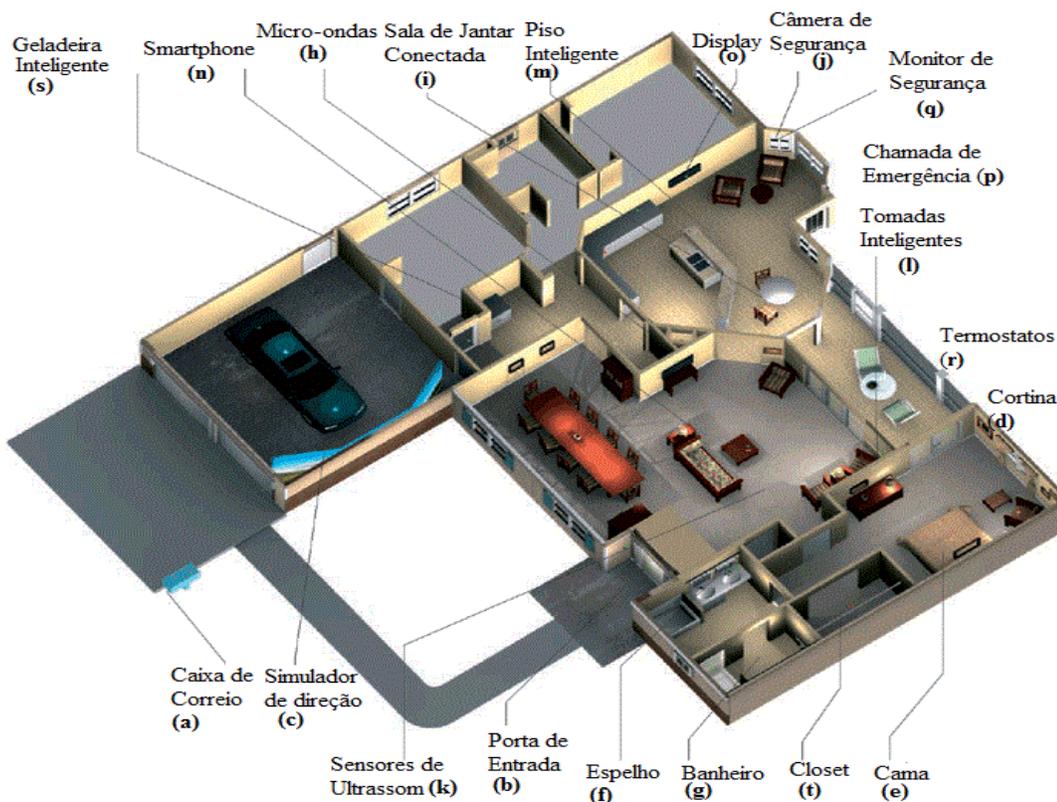
Fonte: Autor “adaptada de” Bolzani, 2010.

Na *University of Florida*, em 2005, um grupo de pesquisadores desenvolveu um projeto conhecido como *Gator Tech Smart Home* (DOMINGUES, 2013). Sua estrutura é baseada em vários dispositivos inteligentes instalados em pontos específicos da residência-laboratório.

Todos os componentes envolvidos no processo são montados com sensores e atuadores, conectados a um sistema operacional projetado para otimizar o conforto e a segurança especificamente de pessoas idosas e deficientes. Essa tecnologia também utiliza um sistema de rastreamento ultrassônico para localizar usuários e avaliar seus hábitos de locomoção, com a finalidade de controlar melhor o ambiente (HELAL et al., 2005).

Ainda segundo Helal *et al.*, 2005, o estudo tem como meta identificar todos os elementos estáticos do ambiente, como roupas, móveis, utensílios e eletrodomésticos. Com isso, será possível identificar, localizar e controlar a maioria dos objetos da casa. A Figura 3 ilustra a localização dos dispositivos inteligentes.

Figura 3 – Projeto Gator Tech com a localização dos dispositivos inteligentes.



Fonte: Autor “adaptada de” Domingues, 2013.

Alguns dos dispositivos inteligentes integrados ao sistema são listados no Quadro 1.

Quadro 1 – Dispositivos integrados ao Projeto Gator Tech.

a	Caixa de correio, que informa ao usuário quando existem correspondências.
b	Porta de entrada, que pode ser aberta automaticamente, com identificação dos usuários por meio de RFID, com comunicação audiovisual.
c	Simulador de direção de carro, instalado na garagem, para coletar dados das habilidades de pessoas idosas para pesquisa.
d	Cortinas das janelas, que podem ser programadas ou acionadas automaticamente de acordo com a iluminação, temperatura e privacidade.
e	Cama do quarto principal, com monitor de sono.
f	Espelho, com mensagens e lembretes, inclusive de remédios.
g	Banheiro, com sensor de término de papel, vazamentos, temperatura da água e biometria

	(temperatura e peso) do usuário.
h	Micro-ondas, que informa o usuário sobre tempos e preparos dos alimentos.
i	Sala de jantar conectada, onde é possível fazer refeições e lanches em videoconferência.
j	Câmeras de segurança, que monitoram o pátio e a frente da casa.
k	Sensores de ultrassom, para detectar o movimento e orientação dos ocupantes.
l	Tomadas inteligentes, que informam qual eletrodoméstico ou iluminação está ligado a elas, podendo ser controladas.
m	Piso Inteligente, para informar o movimento e posição precisos, bem como possíveis quedas.
n	<i>Smartphone</i> , o qual funciona como telefone e permite o controle remoto de luzes, eletrodomésticos e mídia da casa. Também pode receber e prestar informações sobre a casa para o usuário, quando o mesmo estiver ausente.
o	<i>Display</i> com assistência cognitiva, para informar sobre algum medicamento ou tarefa.
p	Chamada de emergência, na suspeita de algum problema mais sério.
q	Monitor de segurança, para informar o estado atual das portas e janelas.
r	Termostatos inteligentes, para condicionamento da temperatura.
s	Geladeira inteligente, para monitorar disponibilidade de alimentos e consumo, informando prazos de validade dos alimentos e criando listas de compras.
t	<i>Closet</i> inteligente, onde sugestões de combinações de roupas serão oferecidas baseadas em condições de tempo externas.

Fonte: Autor.

Além das iniciativas acadêmicas, temos também iniciativas corporativas. Dentre as empresas corporativas que contribuíram para o desenvolvimento de pesquisas em ambientes inteligentes, destacam-se: Microsoft, Philips e IBM.

Criado em 1997, pelo grupo de Computação Ubíqua da *Microsoft Research*, o *Easyliving* é um projeto que surgiu graças ao interesse do seu fundador, Bill Gates, pelo assunto, uma vez que ele possuía uma das casas mais automatizadas naquela época (BOLZANI, 2010). O projeto tinha como ideia principal a distribuição de computadores em vários ambientes de uma casa, de forma ubíqua, ligados em rede, como se fossem um só. A interação usuário-computador poderia

ser feita em todos os ambientes da casa usando gestos e voz, de forma natural (BRUMITT et al., 2000).

A criação do projeto *Easyliving* permitiu o desenvolvimento de vários sistemas de controle, interfaces e modelos de análise comportamental do usuário em ambientes inteligentes, como monitoramento dos usuários por rastreamento de vídeo, onde o sistema identifica a pessoa por meio da sua silhueta em tempo real, permitindo saber a posição do mesmo no ambiente (BRUMITT et al., 2000).

No ano 2000, na Holanda, a *Philips Electronics* lançou um projeto denominado *HomeLab*. O *HomeLab* é uma residência-laboratório, totalmente funcional, monitorada por câmeras, microfones escondidos e espelhos de duas faces, construída para estudar a forma como as pessoas interagem com os protótipos de tecnologia inteligente em um ambiente real vivendo suas rotinas diárias (AARTS, 2004). Os pesquisadores da empresa acreditam que ela permite avaliar de um modo mais realístico e natural a forma como as pessoas interagem com a tecnologia, dessa forma, é possível compreender melhor suas necessidades e motivações para a utilização da tecnologia e trazer produtos melhores para o mercado (BOLZANI, 2010).

HomeLab também funciona como um laboratório onde tudo o que acontece pode ser observado e gravado para uso posterior. Quando as pessoas estão andando na *HomeLab*, elas são assistidas por um líder de observação e uma equipe de psicólogos comportamentais, logo, os pesquisadores podem ver quais tecnologias as pessoas acham fácil e se sentem confortáveis em utilizar e quais são consideradas difíceis ou confusas (PHILIPS, 2016). A Figura 4 mostra o interior da sala do *HomeLab* e a sala de controle pode ser vista na Figura 5.

Figura 4 – Sala do HomeLab.



Fonte: Arrts, 2004.

Figura 5 – Sala de Controle.



Fonte: Arrts, 2004.

As pesquisas realizadas com o *HomeLab* permitiram o aumento do suporte que as casas podem oferecer ao usuário no seu dia-a-dia, principalmente para as pessoas idosas, de forma adequada e aceitável (PHILIPS, 2016).

A IBM, por meio dos seus centros de pesquisas, tem trabalhado em vários projetos ligados a ambientes inteligentes, projetando sistemas que possam interagir com o usuário por meio de voz, gestos e posição do corpo (KEKRE; THEPADE, 2009).

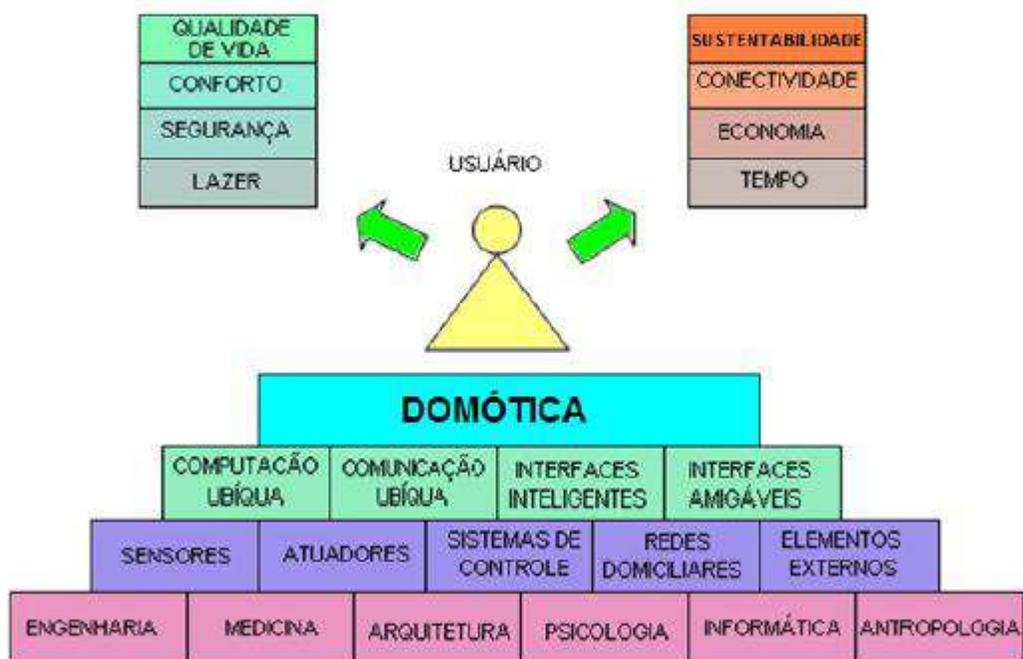
Desde 2010, a IBM também tem investido em serviços e produtos inteligentes para as residências, onde estes produtos estarão conectados em rede, criando uma possibilidade de relacionamento individual entre consumidores, provedores de serviços e indústria (DOMINGUES, 2013). A possibilidade de monitoramento remoto da casa de forma fácil também atrai o interesse dos usuários segundo as pesquisas. A tecnologia se desenvolve através do *Cloud Computing*, um sistema de rede aberto que se traduz em um ambiente inteligente, altamente flexível e eficiente para aplicativos de serviços que poderão ser conectados e usados remotamente, de forma bastante simples (ANDERSON, 2010).

2.2 Domótica

Segundo Angel (1993, apud BOLZANI, 2010), “a palavra domótica originou-se do latim *domus* que significa casa. O termo domótica, nasceu da fusão da palavra *domus* com robótica, ou seja, é uma ciência multidisciplinar que estuda a relação entre o homem e a casa”.

Diretamente ligada ao controle e automação de residências, a domótica, tem como objetivos fundamentais oferecer segurança e conforto (DOMINGUES, 2013). A Figura 6 permite visualizar a relação entre domótica e o usuário.

Figura 6 – A relação da domótica com outras ciências, tecnologias e serviços.

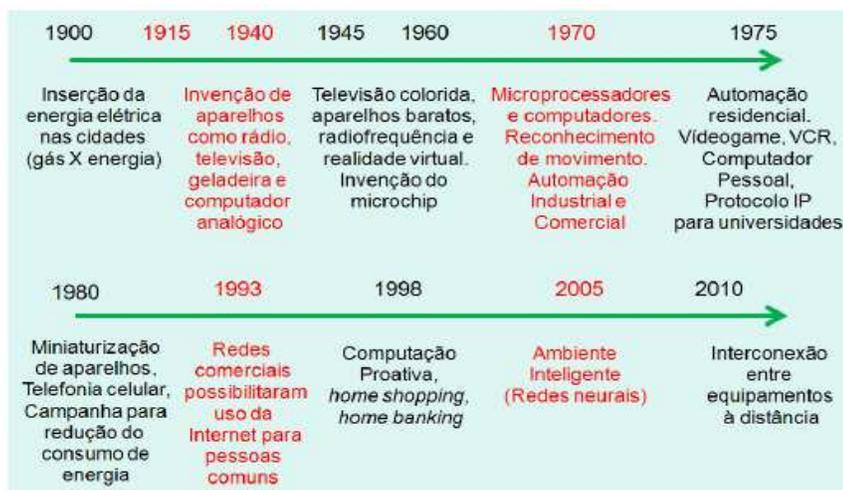


Fonte: Domingues, 2013.

O maior objetivo da domótica é simplificar a vida do habitante, facilitando a sua interação com o ambiente interno e externo. Ela permite a realização desde tarefas simples às de realização difícil, como realizar atividades de forma simultânea (ALDRICH, 2009).

A Figura 7 ilustra a linha do tempo da evolução da domótica até o ano de 2010.

Figura 7 – Linha do tempo da evolução da domótica



Fonte: Domingues, 2013

Existem inúmeras vantagens que podem ser observadas com a aplicação da domótica em casas inteligentes. Algumas vantagens são apresentadas no Quadro 2.

Quadro 2 – Vantagens da domótica.

Vantagens da domótica
A sua utilização de forma apropriada permite a gestão de gastos energéticos, através de funções de regulação da intensidade térmica e luminosa, conjuntamente com sensores de movimento, luz solar e temperatura e também levando em conta hábitos e horários.
Possibilitar usos mais apropriados do tempo.
Diminuir a execução e o tempo de tarefas repetitivas e mecânicas.
Praticidade.
Segurança para a residência e usuários.
Confiabilidade.
Ampliar as interações dos usuários à distância.
Lazer e entretenimento.
Prevenção de acidentes.
Valorização do imóvel.

Fonte: Autor.

O maior objetivo da domótica é simplificar a vida do habitante, facilitando a sua interação com o ambiente interno e externo. Ela permite a realização desde tarefas simples às de realização difícil, como realizar atividades de forma simultânea (ALDRICH, 2009).

Segundo Angel (1993, apud ALDRICH, 2009), os componentes de um sistema domótico, podem ser classificados como: controlador, atuador, sensor, meios de transmissão e elementos externos.

- a) **Controlador:** é a unidade central que gerencia o sistema. Nele, reside toda “inteligência” do mesmo e também as interfaces para que o usuário possa inserir e monitorar as informações envolvidas no processo.
- b) **Atuador:** é o dispositivo de saída responsável por receber uma ordem do controlador e realizar a ação determinada por ele.
- c) **Sensor:** é o dispositivo que está em permanente monitoramento do ambiente com

a finalidade de receber uma informação que será processada pelo controlador.

Quanto ao meio de transmissão, existem diversas tecnologias que podem ser aplicadas: fibra óptica, linha telefônica, rede elétrica, rede de computadores dedicada ou *wireless*.

Os elementos externos são os elementos e/ou sistemas instalados no ambiente, que serão controlados pelo sistema domótico.

Quanto ao nível de automação, os sistemas residenciais domóticos podem ser: autônomos, integrados ou complexos (SILVA, 2009).

Os sistemas autônomos são caracterizados por serem independentes, atuando apenas sobre um dispositivo eletroeletrônico único, ou seja, não há interação entre dispositivos. Sua ação é de Liga/Desliga. Um exemplo bem comum é o sensor de presença que aciona uma luz quando o usuário entra no local (DOMINGUES, 2013).

Os sistemas integrados são caracterizados pela existência de diferentes sistemas, projetados individualmente, mas com o funcionamento integrado entre eles (por exemplo, controle remoto universal). São equipamentos controláveis, com inteligência central ou distribuída. (BOLZANI, 2010).

Segundo Silva (2009), os sistemas complexos são personalizados de acordo com a necessidade do usuário. O sistema é gerenciador, servindo para programar o ambiente em questão. Uma infraestrutura adequada se faz necessária, onde a integração é feita por *software*. Neste caso, enquadram-se as casas inteligentes que são ambientes com grandes níveis de interação (ALDRICH, 2009).

2.3 Integração de Sistemas Residenciais

Atualmente, praticamente tudo pode ser automatizado em uma casa inteligente, permitindo seu controle e gerenciamento remotos (DOMINGUES, 2013). Segundo Bolzani (2010), o ambiente residencial pode possuir vários sistemas autônomos que realizam funções determinadas e responsáveis por diversos setores como podem ser vistos no Quadro 3.

O processo de integração desses sistemas faz com que o ambiente se torne inteligente. Dessa forma, um sistema poderia controlar outro. Um exemplo seria o caso de um incêndio, onde o sistema de ventilação e aquecimento poderia ser desligado e as portas abertas, juntamente com

o acionamento de alarmes e chamadas telefônicas para o corpo de bombeiros (DOMINGUES, 2013).

Quadro 3 - Sistemas autônomos.

Setor	Sistema
Comunicação	Centrais telefônicas, secretária eletrônica, identificador de chamadas, telefone celular.
Climatização	Ar condicionado, termostato, ventilador.
Áudio/vídeo	<i>Home theater</i> , som ambiente, multimídia.
Informática	Microcomputadores, impressoras, <i>scanners</i> .
Utilidades	Aspiração central, aquecedores, gás, sauna, bomba da piscina, irrigação.
Iluminação	Sensores de presença, <i>dimmers</i> (controles de potência de luz), luminárias.
Eletrodomésticos	Geladeira, lavadora de roupas, micro-ondas.
Segurança	Alarmes, circuito fechado de TV (CFTV), portas e fechaduras automáticas.
Energia	Controle de gastos de energia e uso horários de pico.
Água	Gestão de uso da água.

Fonte: Autor “adaptado de” Domingues, 2013.

2.4 Reconhecimento de Padrões

Reconhecimento de padrões é uma área da ciência cujo objetivo é a classificação de objetos dentro de um número de categorias ou classes (COOK, 2009). Esses objetos de estudos variam de acordo com o tipo de aplicação, podem ser imagens, sinais em forma de onda (como luz, rádio ou voz) ou quaisquer tipos de medidas que necessitem ser classificadas (DOMINGUES, 2013).

Um sistema completo de reconhecimento de padrões consiste de um sensor que obtém observações a serem classificadas ou descritas; um mecanismo de extração de características que

computa informações numéricas ou simbólicas das observações e um esquema de classificações das observações, que depende das características extraídas (SILVA, 2009).

Recentemente, devido ao avanço da tecnologia dos sensores, pode-se obter facilmente o consumo de energia dos eletrodomésticos, permitindo identificar o consumo individual de cada aparelho (COOK, 2009).

Usando sensores e com a utilização de um algoritmo adequado, que atenda aos requisitos de um sistema de reconhecimento de padrões, é possível que uma casa inteligente aprenda os padrões de utilização dos seus habitantes de forma autônoma, o que significa que os habitantes não precisam inserir nenhuma informação manualmente. Rashidi e Cook (2009) propuseram em seu artigo a utilização de uma versão modificada do algoritmo *Apriori* para detectar automaticamente padrões recorrentes com os dados coletados em casas inteligentes.

Tanbeer et al. (2009) propuseram o conceito de padrões periódicos frequentes, uma vez que as medidas de interesse, por exemplo o suporte, utilizadas na extração de padrões frequentes não fornecem informações sobre a ocorrência de formas de padrões em bases de dados. Além disso, na análise de dados, tais como a análise do comportamento de compra de um cliente em um supermercado, embora a exploração de padrões frequentes revele os padrões frequentes de ocorrência da compra, os varejistas podem estar interessados na natureza periódica (formas) de compra para a tomada de decisão.

Para resolver esta questão, Tanbeer et al. (2009) definiram o conceito de padrões periódicos como sendo padrões que ocorrem em um intervalo constante.

Rashidi e Cook (2009) diferenciam padrões periódicos de padrões frequentes (padrões que ocorrem mais frequentemente do que outros) visto que ambos são interessantes para posterior automação.

Sobre o tema de reconhecimento de padrões em casas inteligentes, vale ressaltar que há diversos exemplos na literatura. DeMaria (2002) menciona em seu artigo que é possível “executar scripts em [...] temporizadores [para controlar aparelhos diferentes] com funções de randomizador para simular uma pessoa se movendo ao redor da casa para cômodos diferentes, ligar e desligar luzes e eletrodomésticos [...]” e que “alguns temporizadores inteligentes podem aprender os padrões de uso e simular a utilização típica [...]”. Silva (2009) relatou o sucesso do projeto TAHI no Reino Unido: em um ambiente de uma casa inteligente, por meio de sensores de movimento, uma luz de *led* vermelha piscava toda vez que um padrão anormal de atividade era

detectado na casa de um idoso. Dounis e Caraiscos (2009) dizem que “é possível aprender a zona de conforto (série de comportamentos que adotamos por costume) a partir das preferências dos usuários”, mas não citam de que forma isso pode ser feito.

O reconhecimento de padrões é fundamental, pois, é a base para os algoritmos de mineração de padrões utilizados neste trabalho. Quando se fala de eventos ocorridos em uma casa inteligente, a sequência dos eventos é muito importante quando a análise dos dados gerados pelos habitantes é realizada. Levando-se em conta que o intuito é que a casa aprenda de forma autônoma os hábitos de cada habitante, é relevante saber, por exemplo, se o eletrodoméstico A foi ligado antes ou depois do eletrodoméstico B.

2.5 Automação Residencial Inteligente

Os sistemas de automação e controle são muito importantes para o mundo atual. Estão inseridos em praticamente todos os ambientes e atividades das mais diversas áreas. Segundo Mamede Filho (2010), por Automação entende-se a capacidade de se executar comandos, obter medidas, regular parâmetros e controlar funções automaticamente, sem a intervenção humana.

Automação também é sinônimo de integração, ou seja, pode ser produto de um ou mais sistemas que permite que um dispositivo seja controlado de modo inteligente, mais evoluído que o estado mais simples, tanto individualmente quanto em conjunto, para uma função específica (FUSSEK, 2010).

A Automação pode ser dividida em três ramos principais: Automação Industrial, Automação Comercial ou Predial e Automação Residencial.

A Automação Residencial (AR) consiste no uso da automação nas residências, fazendo uso de equipamentos eletroeletrônicos e eletromecânicos para controle de processos que levem ao conforto e segurança dos seus usuários.

Esta seção visa apresentar, de forma resumida, alguns trabalhos desenvolvidos na área de Automação Residencial Inteligente. Nos sistemas desenvolvidos nestes trabalhos, as regras de automação são criadas a partir do comportamento dos habitantes, o qual é capturado por meio de sensores e atuadores implantados na residência.

Em Tonidandel, Takiuchi, Melo (2004) foi apresentado um sistema denominado ABC (Automação Baseada em Comportamento), que visa reverter o processo normal de criação de

regras de automação em uma residência. Normalmente, o habitante cria as regras; no ABC, as regras são criadas pelo próprio sistema, com base no comportamento do habitante.

Sgarbi (2007) desenvolveu um sistema denominado ABC+ baseado em aprendizado de máquina capaz de adaptar-se ao comportamento de um habitante com base em dados obtidos por meio de sensores. O sistema tem a capacidade de aprender e adaptar regras para uma residência com um único morador, ou para residência onde podem ser aplicadas as mesmas ações sobre todos os moradores.

Visando complementar o funcionamento do sistema ABC+, Fussek (2010) propôs o Classificador Automático de Pessoas Baseado em Casos (CAPBC) que permite a adaptação de sistemas de automação residencial desenvolvidos para uso em residências com apenas um habitante para aquelas com duas ou mais pessoas.

Diferentemente dos trabalhos desenvolvidos anteriormente, este trabalho tem como objetivo demonstrar que é possível utilizar algoritmos de mineração de padrões sequenciais frequentes para extrair os dados de eventos ocorridos em casas inteligentes. Isso permite que no futuro, as casas inteligentes possam aumentar o conforto, economizar energia ou melhorar a segurança para idosos residentes em lares de vida assistida.

2.6 Relações Temporais

As atividades realizadas em uma casa inteligente não são instantâneas, mas têm tempos de início e fim distintos. Nota-se que há relações bem definidas entre intervalos de tempo para diferentes atividades. As treze relações temporais definidas por Allen (1983) definem a forma básica de representação entre quaisquer intervalos de dois eventos. Estas relações temporais desempenham um importante papel na identificação de atividades temporais que ocorrem em uma casa inteligente. As relações são mostradas na Figura 8. Essas relações podem descrever qualquer posição relativa entre dois intervalos com base no início e término de cada atividade.

Figura 8 – Relações temporais de Allen.

Exemplos	Relação Temporal
	A antes de B
	A coincide com B
	A depois de B
	B coincide com A
	A inicia B
	A iniciado por B
	A termina B
	A é terminado por B
	A inclui B
	A é incluído por B
	A simultâneo com B
	A durante B
	A idêntico a B

Fonte: Autor “adaptada de” Allen, 1983.

Considerando o *dataset* utilizado neste trabalho, pode-se observar as seguintes relações presentes na base de dados:

- a) A antes de B;
- b) A depois de B;
- c) A inclui B;
- d) A simultâneo com B;
- e) A durante B.

Essas relações, presentes na base, não influenciam os resultados dos algoritmos visto que os mesmos foram criados com base nas relações de Allen.

3 TRABALHOS RELACIONADOS

O rápido desenvolvimento das Tecnologias da Informação e Comunicação, as chamadas TIC, associadas à Automação (TICA), têm criado uma necessidade urgente de se definir teorias, métodos e conceitos para avaliar e analisar sua interação entre ambiente e usuários (PITONI, 2002). Muitos trabalhos têm sido realizados pela comunidade científica em laboratórios experimentais para o desenvolvimento de ambientes inteligentes no mundo todo, graças ao progresso das tecnologias envolvendo redes domiciliares com e sem fio, sensores, métodos computacionais e dispositivos inteligentes (DOMINGUES, 2013).

Esses laboratórios têm procurado trabalhar com espaços onde tecnologias a base de sensores, câmeras e outros equipamentos são instalados numa casa real com pessoas residindo nela. O objetivo da maioria desses projetos é observar o estilo de vida de seus ocupantes, fazendo uma análise das informações provenientes dos sensores que permita descrever os hábitos diários de seus usuários, onde sua modelagem pode proporcionar a antecipação das necessidades através de seu aprendizado, gerando comodidades (BOLZANI, 2010). Apesar de a proposta ser sempre a de gerar conforto e conveniência, muitas pesquisas tem focado especificamente a população de idade avançada ou com alguma incapacidade, no sentido de prover acessibilidade e autonomia.

Os esforços das pesquisas com ambientes inteligentes sempre têm se baseado em quatro pilares principais: Computação ubíqua, inteligência artificial, interatividade e percepção (SILVA, 2009). A computação ubíqua se refere à situação onde o usuário está rodeado de dispositivos invisíveis para ele; a inteligência artificial envolve a análise do contexto pelo sistema usado no que diz respeito ao comportamento e aprendizado do mesmo; a interatividade se refere à forma como o usuário se comunica com o sistema e vice-versa; e a percepção se refere à habilidade que o sistema tem de reconhecer objetos e pessoas e suas intenções (BOLZANI, 2010).

A seguir são mostradas algumas das principais pesquisas com laboratórios/ambientes inteligentes elaboradas pelo mundo, bem como os métodos utilizados para atender os objetivos comuns. A maioria delas ainda se encontra em andamento, permitindo novos avanços.

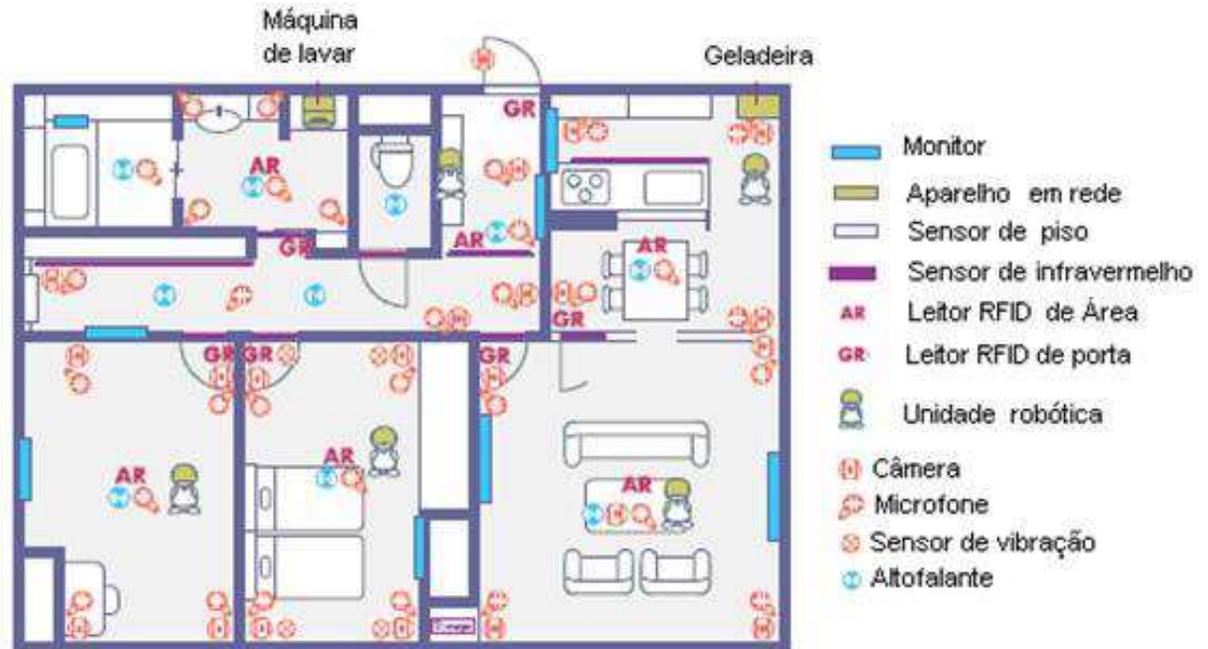
Na Suécia, o *Royal Institute of Technology* desenvolveu o projeto *comHOME*, com a finalidade de simular a casa do futuro. Seu objetivo era integrar soluções de comunicação através de vídeo, permitindo interatividade entre o usuário e o sistema de controle, cuja programação era feita através de interfaces, gestos e posicionamento, proporcionando aprendizado e adaptação do sistema em relação ao usuário (JUNESTRAND et al., 2001).

Na França, na Universidade de *Toulouse*, foi desenvolvido e testado em 2001, um ambiente inteligente experimental específico para suporte de pessoas idosas ou com incapacidade, cujo projeto foi batizado como PROSAFE. Os objetivos da experiência eram a detecção de comportamento anormal do usuário, interpretado como um acidente ou uma queda, e coleta de dados de usuários 24 horas por dia (CHAN et al., 2005). A habitação experimental foi equipada com sensores infravermelhos de movimento conectados a uma rede wireless, que podiam captar os movimentos e, através de um processamento, avaliar a evolução do processo. Mais tarde, em complemento ao PROSAFE, foi criado o projeto ERGDOM, com a característica de ser auto adaptativo, permitindo regular o conforto do ambiente em relação à temperatura, monitorar a segurança e fazer a gestão do consumo de energia do ambiente inteligente em estudo (CHAN et al., 2005).

Na Inglaterra, em *Portsmouth*, foi feito um projeto de pesquisa em conjunto (*University of Portsmouth, John Grooms Housing Association e Portsmouth City Council*), cujo objetivo era produzir soluções de assistência a pessoas com incapacidades, para viver independentemente. A ideia era realizar soluções integradas de arquitetura e tecnologia traduzindo-se em ambientes inteligentes (CHAPMAN; McCARTNEY, 2002). A instalação, composta por seis apartamentos, permitia que uma grande quantidade de dados fosse adquirida eletronicamente através de sensores, a fim de se fazer um estudo do comportamento do usuário mediante a utilização do espaço e produtos de ação remota e interativa em teste.

Na *University of Tokyo*, foi elaborado outro projeto chamado *Ubiquitous Home*, com a finalidade de realizar testes para a criação de novos serviços destinados à ligação de aparelhos eletrodomésticos, sensores e atuadores em redes domiciliares (YAMAZAKI, 2006). Nos ambientes foram instalados sensores, microfones e câmeras, além de unidades robóticas, para servirem de ligação entre o robô principal (a casa) e o usuário. O esquema da residência/laboratório pode ser visto na Figura 9.

Figura 9 – Disposição dos sensores na Ubiquitous Home.



Fonte: Autor “adaptada de” Yamazaki, 2006.

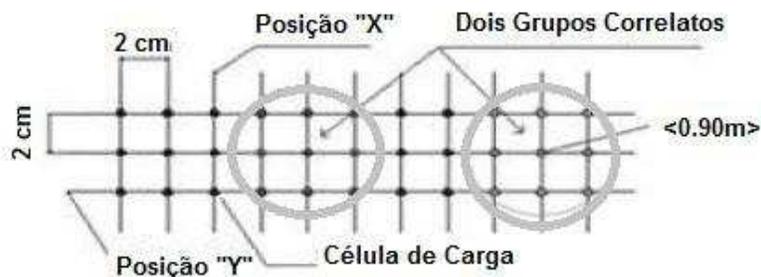
Na Nova Zelândia, na *Massey University*, está sendo feito um estudo sobre o reconhecimento do comportamento de um usuário num ambiente inteligente. Este processo está sendo analisado levando em conta três parâmetros: Tempo, ambiente e contexto racional. O projeto faz uso de um ambiente instrumentado com uma grande variedade de sensores (áudio, vídeo, sensores no corpo do usuário, e etiquetas de RFID). O objetivo maior é o desenvolvimento de um modelo confiável de detecção de comportamento dentro de um ambiente inteligente.

No Brasil, poucos estudos têm sido realizados envolvendo ambientes inteligentes. Trata-se de projetos de grande relevância, pois seu desenvolvimento proporcionou a criação de laboratórios com foco específico no tema e trabalhos de pesquisa considerados promissores. São citados aqui alguns de maior destaque.

No Instituto Militar de Engenharia (IME), foi proposto por Rosa et al. (2004) um trabalho cujo objetivo principal era a adaptação de uma casa inteligente onde seria mínima a interferência dos moradores, não havendo a necessidade de câmeras de vídeo, teclados ou acionamentos por reconhecimento de voz. O ocupante seria monitorado de forma não invasiva por agentes que, juntos tomariam decisões de comando de temperatura e luminosidade da casa. As condições de conforto e segurança se tornariam adequadas, monitorando-se o consumo de energia e mantendo-

se a privacidade do usuário. A modelagem da casa inteligente foi feita baseando-se em “pistas” deixadas pelos moradores, ou seja, suas ações características, proporcionadas por uma coleta de dados atualizada de acordo com o usuário. Os dados serão gerados por sensores de força instalados no chão da casa e serão aplicados em algoritmos de passos, que irão identificar através de uma rede neural, o usuário, juntamente com seus hábitos naquele ambiente. Dependendo do ocupante, este poderá ser classificado com morador, visitante ou invasor, o que permitirá ações específicas de tratamento (SILVA, 2009). Na Figura 10 pode-se observar a malha de sensores a serem instaladas nas zonas de entrada dos cômodos.

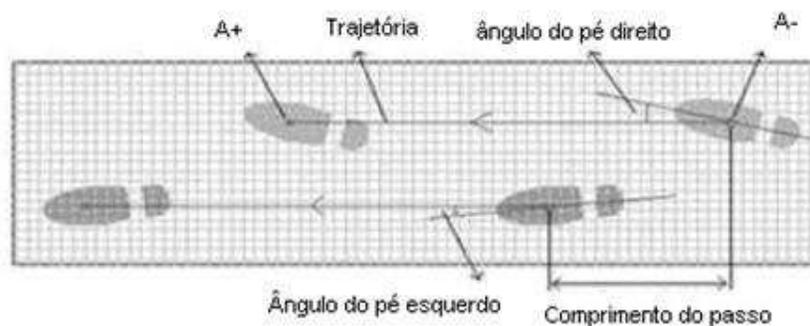
Figura 10 – Malha de sensores de força.



Fonte: Silva, 2009.

Para identificação do usuário, os parâmetros físicos a serem comparados pela posição do passo são a frequência, o peso, o ângulo do pé direito, o ângulo do pé esquerdo e o comprimento do passo. Na Figura 11 podem ser vistas essas características.

Figura 11 – Parâmetros dos passos do usuário.



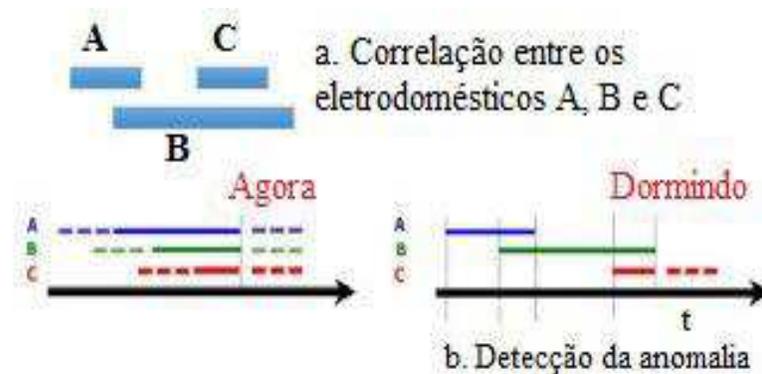
Fonte: Silva, 2009.

No Laboratório de Redes de Computadores (LAR) do Centro Federal de Educação Tecnológica do Ceará (CEFET-CE) foi desenvolvido o projeto PIMENTER, que se propõe a

transformar o uso do computador pessoal (PC) em uma espécie de robô estático, doméstico, inteligente e de acesso remoto (SILVA; OLIVEIRA, 2008). Com isso, foi criado o protótipo PIMENTER INTELLIGENT HOME (PIH), uma aplicação que faz um serviço de integração e controle de dispositivos que estariam ligados ao PC, como câmeras de vídeo, sensores e atuadores, instalados de acordo com a necessidade do usuário. O gerenciamento também pode ser feito via telefonia móvel por meio de uma página da *internet*. O aplicativo também promove o gerenciamento do processamento do PC (SILVA; OLIVEIRA, 2008).

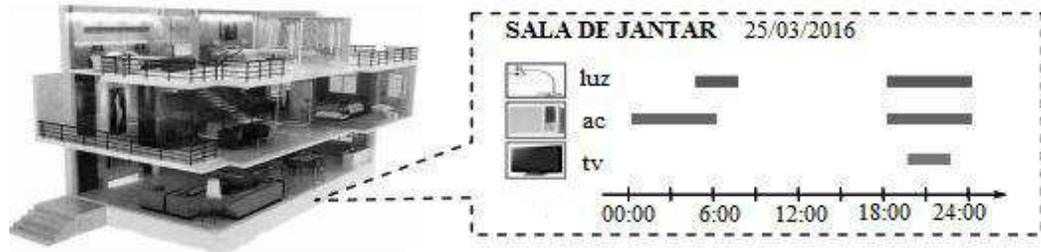
Na área de detecção de anomalias baseado em padrões sequenciais, vários métodos foram propostos para detectar anomalias nos eletrodomésticos de uma casa inteligente. A Figura 12 ilustra um exemplo da detecção de anomalias: a Figura 12a mostra o padrão normal de uso, onde o eletrodoméstico C é desligado antes do eletrodoméstico B (baseado no tempo da esquerda para a direita), ao passo que a Figura 12b mostra que em determinado tempo t , o eletrodoméstico C ainda está ligado e o eletrodoméstico B já foi desligado. A Figura 13 mostra a sequência de utilização diária de três eletrodomésticos em uma casa inteligente.

Figura 12 - Um exemplo de detecção de anomalias em eletrodomésticos de uma casa inteligente.



Fonte: Thai, 2015.

Figura 13 – Um exemplo de sequência de utilização diária.



Fonte: Autor “adaptada de” Thai, 2015.

Pitoni (2002) explora uma abordagem não supervisionada para determinar o número de eletrodomésticos que cada habitante de uma casa inteligente utiliza, incluindo o consumo de energia, em qualquer momento. Baumann et al. (2012) utiliza a análise de componentes principais para extrair recursos de sinais elétricos e classificá-los utilizando o *Support Vector Machine* (SVM). Aldrich (2009) desenvolveu um protótipo de *software* para entender o comportamento dos aparelhos eletrodomésticos. Chen et al. (2014) introduziu dois tipos de padrões de uso para descrever comportamentos representativos dos usuários.

Diferente dos trabalhos desenvolvidos, neste trabalho o foco recai sobre a automação de residências privadas e formas de uma casa inteligente decidir o que é um comportamento normal do seu habitante pela mineração de padrões periódicos e frequentes. Além disso, o habitante não será obrigado a especificar os seus padrões de uso e preferências visto que os mesmos serão aprendidos pelo sistema.

A abordagem para aprender os padrões de utilização escolhidos para este trabalho, utiliza três diferentes algoritmos de mineração de padrões para extrair os dados dos eventos ocorridos em casas inteligentes. As informações apresentadas neste trabalho são baseadas em uma base já existente de dados reais. A base de dados escolhida para este trabalho é a base do *House_n* do MIT (*Massachusetts Institute of Technology*) pois, a mesma satisfaz as características de uma residência real e será descrita com detalhes na Seção 5.

4 ALGORITMOS DE DESCOBERTA DE PADRÕES DE SEQUÊNCIAS

Mineração de padrões sequenciais representa uma importante classe de problemas de mineração de dados com grande variedade de aplicações. Minerar padrões sequenciais é uma atividade desafiadora, pois se lida com uma verificação cuidadosa de um grande número de possíveis combinações (PATEL, 2008).

Usa-se a mineração de padrões em diferentes contextos, tais como as organizações empresariais para estudar o comportamento dos clientes, na mineração de uso da *web* para encontrar *web logs* distribuídos em vários servidores e para a identificação e análise dos comportamentos de habitantes de uma casa inteligente que é o foco deste trabalho (NUGENT, 2008).

Nas próximas seções serão apresentados conceitos fundamentais de mineração de padrões sequenciais. Posteriormente, serão apresentados os algoritmos que serão utilizados para a mineração de padrões sequenciais em casas inteligentes.

4.1 Mineração de Itemsets Frequentes

No contexto de mineração de dados e mineração de padrões de sequências em casas inteligentes, a identificação de ações frequentes, também chamadas de *itemsets* frequentes, é um dos conceitos mais importantes. A seguir, serão apresentados algumas definições importantes.

- a) **Itemset:** Um *itemset* é um conjunto não vazio composto por uma ou mais ações realizadas pelo habitante representado por $I = \{i_1, i_2, \dots, i_n\}$ para $1 \leq i \leq n$.
- b) **Sequência:** Uma sequência é uma lista ordenada de ações dos habitantes (*itemsets*). Denota-se uma sequência S_i por $\langle s_i, s_j, \dots, s_x \rangle \in I$, onde $s_i, s_j,$ e s_x são *itemsets* e $S \supset I$. Os *itemsets* $s_i, s_j,$ e s_x também são considerados elementos da sequência onde $s_i \neq s_j \neq s_x \mid s_i \in S, s_j \in S, s_x \in S$.

Um *itemset* pode ser considerado como uma sequência com um único elemento.

O tamanho de uma sequência é determinado pelo número de instâncias, ou seja, pela quantidade de itens presentes na sequência. Uma sequência de tamanho 1 é chamada de 1 – sequência.

A Tabela 1 abaixo ilustra um exemplo de uma Base de dados de Ações (*itemsets*) dos habitantes (denotada por BA) que é composta por uma tabela de duas colunas, a primeira corresponde ao identificador da ação (denotada por TIA DAS OCORRÊNCIAS) e a segunda corresponde à ação propriamente dita.

Tabela 1 – Base de dados de Ações dos habitantes

TIA DAS OCORRÊNCIAS	AÇÕES (<i>ITEMSETS</i>)
1	Estudar, ler, assistir tv, ir ao dentista, viajar.
2	Ler, assistir tv.
3	Estudar, ler, escovar os dentes.
4	Ler, estudar, navegar na <i>web</i> .
5	Estudar, escovar os dentes, ir ao shopping.

Fonte: Autor.

- c) **Frequência:** É a ocorrência de um *itemset* na BA. É denotada por σ . Considerando-se a tabela acima: $\sigma(\{\text{ler, estudar}\}) = 3$.

Considera-se também o suporte da BA, chamada de s , que corresponde à fração da σ considerando-se toda a BA: $s(\{\text{ler, estudar}\}) = 3/5$, ou seja, o suporte $s = 0,6$. Com base no tamanho da BA e no suporte s define-se o suporte mínimo (denotado por minSup) que é o valor mínimo que um *itemset* deve ter para ser considerado frequente. Considerando o minSup da BA acima 0,5 conclui-se que os *itemsets* ($\{\text{ler, estudar}\}$) são frequentes.

- d) **Regra de associação:** Seja A um conjunto de ações: $A \subseteq I$, uma regra de associação é quando $A \rightarrow B$, onde $A \subseteq I$, $B \subseteq I$ e $A \cap B = \emptyset$. Considerando-se a tabela acima: $A = \{\text{Estudar, ler}\}$, $B = \{\text{Navegar na web}\}$, $\{\text{Estudar, ler}\} \rightarrow \{\text{Navegar na web}\}$.

e) **Confiança:** Considerando-se a regra de associação $A \rightarrow B$, confiança, denotada por c , é a probabilidade na base de ações que contém A e também contém B , ou seja, $C = P(B | A) = \sigma(A \cup B) / \sigma(A)$.

Considerando-se a tabela acima:

$\{\text{Estudar}\} \rightarrow \{\text{Ler}\}$

$\sigma(\{\text{Estudar}\} \rightarrow \{\text{Ler}\}) = 3$

$\sigma(\{\text{Estudar}\}) = 4$

$c = \frac{3}{4} = 0,75$.

Na literatura, são encontrados diversos algoritmos para descoberta e mineração de *itemsets* frequentes: *AIS*, *Apriori*, *Apriori-Tid*, *Apriori Hybrid*, *FPGrowth*, *Max – Miner*, entre outros. No próximo capítulo, será apresentado brevemente o clássico e pioneiro algoritmo *Apriori*, proposto por Agrawal e Srikant em 1994, que é fundamental para a compreensão dos demais algoritmos utilizados para mineração de padrões em casas inteligentes apresentados nesse trabalho.

4.2 Algoritmo *Apriori*

O algoritmo *Apriori*, proposto por Agrawal e Srikant em 1994, é o método mais utilizado na mineração de padrões frequentes em bases de dados. O nome do algoritmo é baseado no fato de que o seu método se utiliza das características de um padrão frequente já encontrado anteriormente (*prior*) para buscar mais padrões.

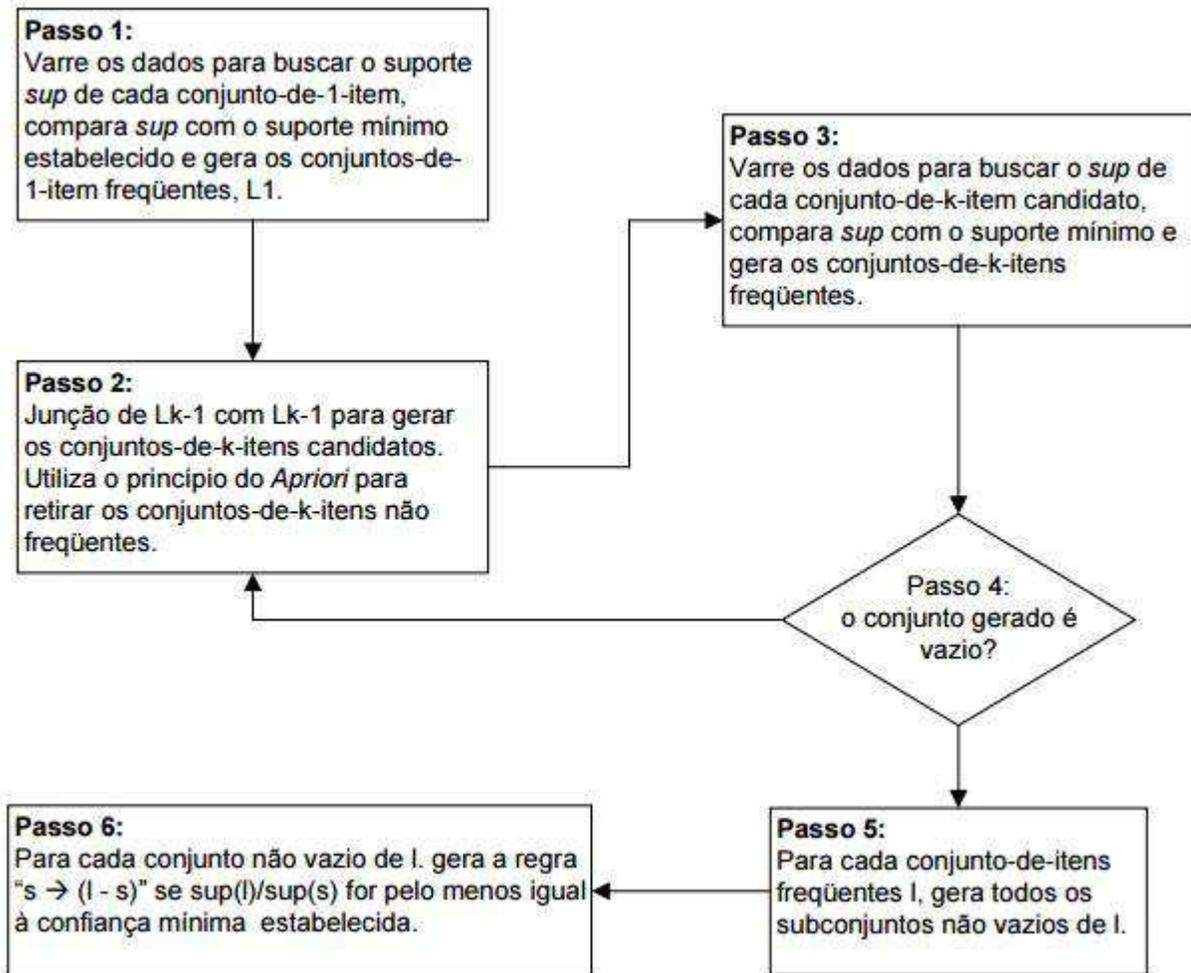
O algoritmo trata cada valor presente, de um atributo, como um item e conforme explicado na seção anterior, uma série de itens é denominada *itemset*.

O algoritmo faz diversas passagens sobre a base de dados para encontrar os *itemsets* frequentes, sendo que, em cada um destes passos, primeiro gera um conjunto de itens candidatos, e então percorre a base de dados para determinar se os candidatos satisfazem o suporte mínimo estabelecido (AGRAWAL, 1994). Na primeira passagem, o suporte para cada item individual (1-*itemsets*) é contado e todos aqueles que satisfazem o suporte mínimo são selecionados (AGRAWAL, 1996).

Em seguida, utilizam-se apenas esses itens frequentes (descartando-se os demais) para gerar as possíveis combinações entre eles e verificar a frequência de coincidências em que cada

um deles aparece junto a outro item (*2-itemsets*). Os *2-itemsets* que alcançaram o suporte mínimo definido, ou seja, os mais frequentes são utilizados para buscar os *3-itemsets* mais frequentes e assim sucessivamente. O algoritmo prossegue iterativamente, até que o conjunto de *k-itemsets* frequentes encontrado seja um conjunto vazio. Os passos descritos acima estão ilustrados na Figura 14.

Figura 14 – Flowchart do algoritmo Apriori.



Fonte: Autor "adaptada de" Pittoni, 2002.

Este algoritmo usa o princípio de que cada subconjunto de um conjunto de itens frequentes também deve ser frequente. Esta regra é utilizada para reduzir o número de candidatos a serem comparados no banco de dados. Todos os candidatos gerados que contenham algum subconjunto que não seja frequente são eliminados.

A Tabela 2, já apresentada anteriormente, ilustra um exemplo de funcionamento do *Apriori*.

Tabela 2 – Base de dados de Ações de um habitante

TIA DAS OCORRÊNCIAS	AÇÕES (<i>ITEMSETS</i>)
1	Estudar, ler, assistir tv, ir ao dentista, viajar.
2	Ler, assistir tv.
3	Estudar, ler, escovar os dentes.
4	Ler, estudar, navegar na <i>web</i> .
5	Estudar, escovar os dentes, ir ao shopping.

Fonte: Autor.

Nas Figuras 15 e 16, simula-se a execução do algoritmo *Apriori* para a lista de ações apresentada na Tabela 1. O algoritmo gera os conjuntos de ações frequentes com um suporte maior que o mínimo estabelecido de 40% (2/5). Já a Figura 16 apresenta o algoritmo gerando as regras que possuem uma confiança mínima de 70%.

Passo 1:**C1**

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte</i>
I1	Estudar	4/5 (80%)
I2	Ler	4/5 (80%)
I3	Assistir tv	2/5 (40%)
I4	Ir ao dentista	1/5 (20%)
I5	Viajar	1/5 (20%)
I6	Escovar os dentes	2/5 (40%)
I7	Navegar na <i>web</i>	1/5 (20%)
I8	Ir ao shopping	1/5 (20%)

L1

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte</i>
I1	Estudar	4/5 (80%)
I2	Ler	4/5 (80%)
I3	Assistir tv	2/5 (40%)
I6	Escovar os dentes	2/5 (40%)

Passo 2

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte</i>
{I1, I2}	Estudar, ler	3/5 (60%)
{I1, I3}	Estudar, assistir tv	1/5 (20%)
{I1, I6}	Estudar, escovar os dentes	2/5 (40%)
{I2, I3}	Ler, assistir tv	2/5 (40%)
{I2, I6}	Ler, escovar os dentes	1/5 (20%)
{I3, I6}	Assistir tv, escovar os dentes	0/5 (0%)

Passo 3

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte</i>
{I1, I2}	Estudar, ler	3/5 (60%)
{I1, I6}	Estudar, escovar os dentes	2/5 (40%)
{I2, I3}	Ler, assistir tv	2/5 (40%)

Passo 4: L2 não é vazio, repete passo 2.

Passo 2:**C3**

Figura 15 – Apriori gerando os conjuntos de itens frequentes.

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte</i>
{I1, I2, I3}	Estudar, ler, assistir tv	1/5 (20%)
{I1, I2, I6}	Estudar, ler, escovar os dentes	1/5 (20%)
{I1, I3, I6}	Estudar, assistir tv, escovar os dentes	0/5 (0%)
{I2, I3, I6}	Ler, assistir tv, escovar os dentes	0/5 (0%)

Fonte: Autor “adaptade de” Pittoni, 2002.

Passo 5

<i>Itemset ID</i>	<i>Ações (Itemsets)</i>	<i>Suporte (AB)</i>	<i>Suporte (A)</i>	<i>Confiança</i>
I1 I2	Estudar ler	60%	80%	75%
I1 I6	Estudar escovar o dente	40%	80%	50%
I2 I3	Ler assistir tv	40%	80%	50%
I2 I1	Ler estudar	60%	80%	75%
I6 I1	Escovar o dente estudar	40%	40%	100%
I3 I2	Assistir tv ler	40%	40%	100%

Passo 6:

Figura 16 – Apriori gerando as regras.

Regras	Suporte	Confiança
I1 \Rightarrow I2 Estudar \Rightarrow Ler	60%	75%
I2 \Rightarrow I1 Ler \Rightarrow Estudar	60%	75%
I6 \Rightarrow I1 Escovar o dente \Rightarrow Estudar	40%	100%
I3 \Rightarrow I2 Assistir tv \Rightarrow Ler	40%	100%

Fonte: Autor “adaptada de” Pittoni, 2002.

O algoritmo *Apriori* é apresentado na Figura 17. L_k corresponde ao conjunto de todos os conjuntos de k -*itemsets* frequentes e C_k é o conjunto de todos os candidatos participantes da análise, isto é, potencialmente frequentes, na passagem K (AGRAWAL E SRIKANT, 1994).

Figura 17 – Algoritmo Apriori.

```

Entrada: Base de dados, Suporte mínimo  $\text{minSup}$ 
Saída:  $U_k L_k$ 
1  $L_1 = \{\text{conjunto de 1- itemsets frequente}\}$ 
2 para ( $k=2; L_{k-1} \neq \emptyset; k++$ ) faça
3    $C_k = \text{apriori\_gen}(L_{k-1});$ 
4   para todas transações  $t \in D$  faça
5      $C_t = \text{VerificarCandidatosnaTransação}(C_{k,t})$ 
6     para todo  $c \in C_t$  faça
7        $c.\text{count} ++; t.\text{count} ++;$ 
8    $L_k = \{c \in C_k \mid c.\text{count} / t.\text{count} \geq \text{minSup}\};$ 
9 retorna  $U_k L_k$ 
10 fim

```

Fonte: Autor “adaptada de” Srikant, Agrawal, 1994.

O algoritmo *Apriori* funciona da seguinte maneira: primeiramente os conjuntos candidatos de tamanho k são gerados, tendo como base o conjunto de itens frequentes encontrados anteriormente (passo anterior $k - 1$) (linhas 1 a 3). O suporte de candidato “ c ” é calculado (linhas 4 e 5). O algoritmo prossegue iterativamente (linhas 6 e 7) até que o conjunto dos itens frequentes L_k (que atingirem o suporte mínimo determinado) é gerado (linha 8), cuja união constitui a saída do algoritmo (linha 9).

4.3 Algoritmos para Extração de Padrões de Sequência

A mineração de padrões sequenciais foi introduzida em (AGRAWAL; SRIKANT, 1994) com a apresentação do algoritmo *Apriori*. No entanto, esta tarefa de mineração de dados, também conhecida como MD, ganhou notabilidade com o algoritmo *Generalized Sequential Pattern* (GSP) apresentado em Agrawal e Srikant (1996).

Uma sequência temporal consiste em um conjunto de *itemsets* ordenados temporalmente. O tamanho de uma sequência é igual ao número de *itemsets* que possui.

Considerando-se uma base de ações (BA), em uma casa inteligente, composta por ações temporais, pode-se realizar a mineração de padrões sequenciais, cujo objetivo é descobrir todas as sequências frequentes de ações, ordenadas no tempo.

As seções a seguir descrevem alguns algoritmos utilizados na Extração de Padrões Sequenciais (EPS). O algoritmo GSP (AGRAWAL; SRIKANT, 1996) foi baseado na técnica *Apriori* descrita anteriormente, já o PrefixSpan (ASL et al., 2004) e o Spade (*Sequential Pattern Discovery using Equivalence classes*) (ZAKI, 2001) utilizam técnicas alternativas para contornar o problema das consecutivas varreduras no banco de dados.

4.3.1 Estratégia de Geração e Teste de Candidatos

A estratégia de Geração e Teste de candidatos consiste basicamente na repetição de duas etapas, Geração e Teste, até não ser mais possível a geração de sequências frequentes. Inicialmente os algoritmos que utilizam essa estratégia fazem uma varredura na base de dados encontrando os itens que são frequentes. A partir da etapa inicial, é feita a Geração, na qual os itens são combinados gerando as sequências candidatas. Com as sequências candidatas inicia-se a etapa de Teste. Nesta etapa, os suportes das sequências candidatas são calculados e verifica-se se as sequências são frequentes. As sequências frequentes são novamente combinadas com os *itemsets* frequentes gerando sequências candidatas maiores, novamente passando pela fase de Geração. Essas novas sequências geradas passam pela etapa de teste. Este *loop* é repetido até não ser mais possível a geração de sequências frequentes.

4.3.2 O Algoritmo GSP

O Algoritmo GSP (*Generalized Sequential Patterns*), introduzido em 1996 por Agrawal e Srikant, utiliza a estratégia de Geração e Teste de candidatos e se baseia na técnica do *Apriori*.

O pseudocódigo do algoritmo é apresentado na Figura 18.

Figura 18 – Algoritmo GSP.

Entrada: Base de dados, Suporte mínimo *minSup*
Saída: $\bigcup_{n=0}^k \mathcal{F}_n$

- 1 $\mathcal{F}_1 \leftarrow \{\text{itens frequentes de tamanho } 1\}$;
- 2 **para** $k \leftarrow 2$ até $\mathcal{F}_k.tamanho \neq 0$ **do**
- 3 $C_k \leftarrow$ candidatos *k*-sequencia ;
- 4 **para cada** sequencia $\mathcal{S} \in$ base de dados **do**
- 5 | incremente contador $\alpha \in C_k \subset \mathcal{S}$;
- 6 **fim**
- 7 $\mathcal{F}_k \leftarrow \{\alpha \in C_k | \alpha.sup \geq minSup\}$;
- 8 **fim**

Fonte: Autor “adaptada de” Srikant, Agrawal, 1996.

O GSP funciona da seguinte maneira: primeiramente é realizada uma varredura na base de dados, encontrando todos os itens frequentes que são as sequências unárias no resultado final (linha1). Então, enquanto for possível gerar sequências maiores (linha 2), o algoritmo combina as sequências geradas na iteração anterior com os itens frequentes, gerando, assim, sequências maiores (linha3). Os suportes destas novas sequências são computados (linhas 4 a 6) e verifica-se se são maiores ou iguais ao mínimo configurado pelo usuário (linha 7). Caso seja menor, a sequência é descartada. As sequências frequentes passam novamente pelo processo (linhas 2 a 8) até que a etapa de Geração deixe de gerar sequências frequentes. O resultado (Saída) é a união das n sequências frequentes geradas ($\bigcup_{n=0}^k \mathcal{F}_n$).

De acordo com Palencar e Rosell (2010), o GSP executa diversas buscas no banco de dados. O suporte mínimo de cada item é obtido na primeira busca, obtendo-se assim os itens frequentes, ou seja, aqueles que cumprem o requisito de suporte mínimo. As demais buscas geram novas sequências candidatas com bases nas sequências frequentes encontradas anteriormente, logo, a partir das 2-sequências frequentes são geradas as 3-sequências candidatas e este processo é repetido até que não sejam encontradas mais sequências frequentes (REIS, 2015).

Conforme citado na seção anterior, o GSP é baseado no algoritmo *Apriori*, logo apresenta como desvantagem a geração de uma grande quantidade de sequências candidatas o que influencia na velocidade de processamento dos dados dependendo do tamanho da base de dados. A Figura 19 exemplifica os passos realizados em cada etapa do GSP.

Figura 19 – Passos realizados em cada etapa do GSP.



Fonte: Autor “adaptada de” Kamat, Bakal, Nashipudi, 2013.

4.3.3 O Algoritmo SPADE

O algoritmo SPADE (*Sequential Pattern Discovery using Equivalence classes*), proposto por Zaki (2001), utiliza propriedades combinatórias para decompor o problema original em subproblemas menores, que podem ser resolvidos de forma independente. Ao contrário dos algoritmos baseados no *Apriori*, o SPADE encontra as sequências frequentes com apenas três varreduras no banco de dados.

Enquanto o algoritmo GSP, descrito na seção anterior, encontra os padrões considerando o banco de dados em seu formato horizontal, o SPADE cria uma lista com identificadores verticais, para facilitar a extração dos padrões frequentes.

A Figura 20 ilustra um banco de dados no formato horizontal (utilizado pelo GSP) com três campos. A tabela abaixo deve ser lida horizontalmente da esquerda para a direita, dessa forma, têm-se as ações que o habitante realiza em cada horário. Sabe-se, por exemplo, que o habitante estuda as 5:00 PM (linha 2) e liga a tv do quarto as 10:30 PM (última linha).

Figura 20 – Banco de dados de uma casa inteligente no formato horizontal.

idA	Ações
10	D
15	AB
20	AB
25	AD
15	AB
10	AB
10	D
20	B
25	A

idA: Instante da ação (horário);
Ação: Ação realizada pelo habitante.

idA 10 -> 8:30 am;
idA 15 -> 5:00 pm;
idA 20 -> 8:00 pm;
idA 25 -> 10:30 pm.

Ação A -> Ligar Tv do quarto;
Ação B -> Jantar;
Ação D -> Ir para a faculdade;
Ação AB -> Estudar;
Ação AD -> Dormir.

Fonte: Autor.

A Figura 21 mostra como o SPADE faz o retorno dos dados da Figura 20 após junção das listas criadas, dessa forma é possível observar todas as ações do habitante em uma única linha. Sabe-se, por exemplo, que o habitante escova os dentes as 11 PM, toma banho as 5 PM e navega na web as 10 PM.

Figura 21 – Retorno para o banco de dados horizontal.

Pares (Ação, idA)
(A,15) (A,20) (A,25) (B,15) (B,20) (D,10) (D,25)
(A, 15) (B, 15)
(A, 10) (B,10)
(A, 25) (B, 20) (D,10)

idA: Instante da ação (horário);
Ação: Ação realizada pelo habitante.

Ações:

A -> Escovar os dentes;
B -> Tomar banho;
D -> Navegar na web.

idA 10 -> 10 PM;
idA 15 -> 8:30 AM;
idA 20 -> 5 PM;
idA 25 -> 11 PM

Fonte: Autor.

O algoritmo SPADE apresenta um desempenho melhor que o GSP visto que apresenta otimizações que reduzem o consumo da memória e melhoram a sua eficiência.

O SPADE é baseado em listas que contêm informações sobre a localização dos padrões nas sequências e as consequências consecutivas desses padrões fazem com que a lista cresça rapidamente, podendo dessa forma, afetar o desempenho do algoritmo (CECÍLIA, 2011).

O pseudocódigo do algoritmo é apresentado na Figura 22.

Figura 22 – Algoritmo SPADE.

Entrada: Suporte mínimo minSup , *Id-List* das sequências

Saída: Sequência frequente F_s

```

1  $F_1 = 1$  – sequências frequentes;
2  $F_2 = 2$  – sequências frequentes;
3 Determinar  $E$  como conjunto de classes de
  equivalência para todas as 1 – sequências  $[Q]_{\exists 1}$ ;
4 para cada  $Q \in E$  do
5 Encontrar a sequência frequente  $F_s$ .
6 fim

```

Fonte: Autor “adaptada de” Grover, 2014.

O algoritmo SPADE funciona da seguinte forma: primeiramente são encontradas as 1 – sequências frequentes (linha 1). Na sequência o algoritmo encontra as 2 – sequências frequentes (linha 2). Posteriormente é feita a decomposição em classes de equivalência (linhas 3 e 4). Para cada classe de equivalência encontrada o algoritmo encontra a sequência frequente (linha 5).

4.3.4 Estratégia de Crescimento de padrões

A estratégia de Crescimento de Padrões foi introduzida por Pei et al. (2001) com a apresentação do algoritmo *Prefix-projected Sequential pattern mining* (PrefixSpan). Esta estratégia consiste na criação de subdivisões na base de dados baseando-se em prefixos ou sufixos frequentes. Estas subdivisões passam pelo mesmo processo, extraíndo, assim, seus prefixos frequentes e novamente subdividindo a base de dados (BD). Desta forma os padrões são encontrados, pois, vão crescendo à medida que os prefixos frequentes aparecem em uma determinada subdivisão da base.

O algoritmo PrefixSpan será apresentado em detalhes na próxima seção.

4.3.5 O algoritmo PrefixSpan

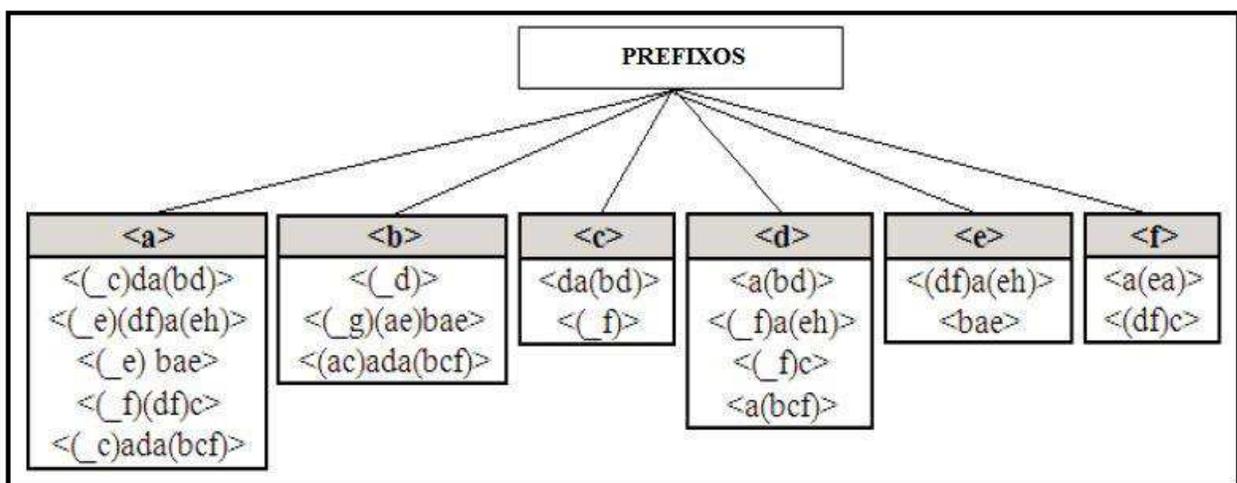
O algoritmo PrefixSpan utiliza uma técnica diferente da empregada no *Apriori*. Este algoritmo aplica o princípio de *Pattern-Growth* (Crescimento de Padrão) por meio da projeção de prefixos para geração de padrões sequenciais (REIS, 2015).

O prefixo de uma sequência consiste no conjunto formado por todas as subsequências que iniciam uma super-sequência. Por exemplo, seja $S = \langle a b c d \rangle$, os prefixos de S são: $\langle \rangle$, $\langle a \rangle$, $\langle a b \rangle$, $\langle a b c \rangle$ e $\langle a b c d \rangle$. Pós-fixos são as subsequências que terminam a sequência S . Por exemplo, seja $S = \langle a b c d \rangle$, os pós-fixos são: $\langle \rangle$, $\langle d \rangle$, $\langle c d \rangle$, $\langle b c d \rangle$.

A ideia do *Pattern-Growth* consiste em considerar que a projeção seja feita baseando-se nos prefixos das sequências que ocorrem frequentemente (CECÍLIA, 2011).

O PrefixSpan utiliza a estratégia de dividir e conquistar por meio de projeções da base de dados. Estas projeções dividem a BD (Base de dados) em porções menores, dessa forma, o espaço de busca é menor e a identificação e extração das sequências frequentes são realizadas de forma mais rápida e eficaz. A Figura 23 ilustra os prefixos $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle d \rangle$, $\langle e \rangle$ e $\langle f \rangle$ extraídos de uma base de dados projetada. Também é possível observar as projeções de cada prefixo em porções menores a fim de tornar o processo de extração mais rápido. O algoritmo PrefixSpan é apresentado na Figura 24.

Figura 23 – Base de dados projetada.



Fonte: Palencar e Rosell, 2010, p.12.

Figura 24 – Algoritmo PrefixSpan .

```

Entrada: Base de Dados  $S$  e  $minSup$ 
Saída: Conjunto de padrões sequencial.
1 inicio
2    $t \leftarrow PrefixSpan(<>, 0, S)$ ;
3   retorna  $t$ ;
4 fim
5 Sub-rotina  $PrefixSpan(\alpha, l, S|_{\alpha})$ 
   Dados:  $\alpha$  sequencia de padrões;  $l$  tamanho de  $\alpha$ ;  $S|_{\alpha}$  projeção da base de dados para  $\alpha$ .
6 inicio
7    $\beta \leftarrow$  Busca em  $S|_{\alpha}$  itens frequentes ;
8   se algum item  $\beta$  não pode ser montado no último elemento de  $\alpha$  e algum item  $\beta$  não
   pode ser adicionado a  $\alpha$  então
9     | exclui item de  $\beta$  ;
10  fim
11  para cada  $\beta$  faça
12    |  $\alpha' \leftarrow$  adiciona_ao_final( $\beta, \alpha$ ) ;
13  fim
14  para cada  $\alpha'$  faça
15    | Construa  $S|_{\alpha'}$ ;
16    |  $PrefixSpan(\alpha', l + 1, S|_{\alpha'})$ ;
17  fim
18 fim

```

Fonte: Autor “adaptada de” Pei et al., 2001.

O algoritmo PrefixSpan funciona da seguinte maneira: na execução do algoritmo, a partir de uma base sequencial são encontrados os prefixos mais frequentes (linha 7 a 9). Para todos os prefixos (linha 14), são geradas projeções da base (linha 15). A projeção da base de dados é feita sempre referente a um prefixo e é composta pelos pós-fixos das sequências que apresentam o prefixo ao qual a projeção é referente. O algoritmo, recursivamente, achará os prefixos frequentes nessas projeções (linha 16) e assim sucessivamente. Baseado nos prefixos frequentes que vão sendo encontrados dentro das projeções, as sequências vão aumentando.

O algoritmo PrefixSpan é mais eficiente do que os algoritmos baseados no algoritmo *Apriori*. O consumo de memória é estável ao se executar o PrefixSpan (CECÍLIA, 2011). O principal problema deste algoritmo é o tempo, que é consumido para se explorar as bases de dados projetadas, que podem ser muito grandes, dependendo do tamanho da base original.

5 ANÁLISE DA EXTRAÇÃO DE COMPORTAMENTOS SEQUENCIAIS

Considerando-se que, nos dias atuais a tarefa de mineração de padrões sequenciais em casas inteligentes é uma tarefa desafiadora, já que a busca pode exigir a análise de um alto número de padrões, torna-se muito complexa a simulação dos algoritmos GSP, SPADE e PrefixSpan em uma residência e situações reais, pois, uma simulação real dependeria de meses, talvez anos, de coletas de dados.

Além disto, há muitas variáveis a serem testadas, visando encontrar os critérios que tornarão a análise dos algoritmos eficientes para a extração dos padrões sequenciais. O tempo necessário para a coleta de dados suficientes, para avaliação de cada conjunto de critérios considerados, também seria impraticável.

Desta forma, a solução encontrada foi trabalhar com uma base já existente de dados reais. A base de dados escolhida para este trabalho é a base do *House_n* do MIT (*Massachusetts Institute of Technology*) de 2007, pois, a mesma satisfaz as características de uma residência real e será descrita a partir da próxima seção.

5.1 *House_n*

House_n é um grupo de pesquisa do MIT focado em como o *design* das casas inteligentes e suas respectivas tecnologias, produtos e serviços devem evoluir para atender melhor as oportunidades e os desafios do futuro. Os pesquisadores do MIT estão investigando métodos para a fusão de novas tecnologias com o *design* centrado nos habitantes. Eles estão gerando novas ideias, tecnologias e metodologias que suportam a criação de produtos e serviços inovadores que satisfaçam as necessidades futuras e emergentes dos habitantes (MIT, 2016).

5.1.1 *PlaceLab*

A missão do *House_n* é conduzir a pesquisa, projetando e construindo ambientes reais, denominados *living-labs* (laboratórios vivos) que são usados para estudar a tecnologia e as estratégias do projeto no contexto (MIT,2016).

O *PlaceLab* é uma iniciativa conjunta do MIT e da TIAX, LLC (consultoria empresarial que desenvolve novas tecnologias, protegidas por IP). É um condomínio residencial em *Cambridge, Massachusetts*, projetado para ser uma facilidade de pesquisa altamente flexível e multidisciplinar para o estudo científico dos habitantes e seus padrões sequenciais de ações em casas inteligentes.

Dezenas de sensores são instalados em todos os cômodos da casa (o condomínio é composto por residências de um dormitório). Esses sensores estão sendo usados para desenvolver aplicações inovadoras de interface com o usuário que ajudam os habitantes a controlar facilmente seu ambiente, economizar recursos, permanecer mental e fisicamente ativos e permanecer saudáveis. Além dos sensores serem usados para capturar as sequências de ações dos habitantes, eles também monitoram todas as atividades que ocorrem no ambiente para que os pesquisadores possam estudar cuidadosamente como os habitantes reagem a novos dispositivos e sistemas no contexto de casas inteligentes.

Cada casa inteligente do condomínio possui um único habitante, voluntário, que concorda em viver na casa por determinado período de tempo. Enquanto vivem nas casas, os habitantes não têm contato com os pesquisadores que têm a capacidade de monitorar todos os aspectos da rotina do habitante na casa, particularmente suas sequências de ações, além de monitorar as condições ambientais no interior e no exterior da casa.

Atualmente, o *PlaceLab* está sendo usado para investigar as seguintes questões sobre as sequências de ações de habitantes em casas inteligentes, entre outras:

- a) O que influencia o comportamento dos habitantes em casas inteligentes?
- b) Como a tecnologia pode ser eficaz no contexto doméstico por longos períodos de tempo?
- c) Até que ponto os padrões de ações sequenciais dos habitantes podem ser extraídos corretamente por algoritmos de extração de padrões sequenciais?

Há diversos projetos em andamento no *PlaceLab*. Um destes projetos é o desenvolvimento de um meio preciso, e não invasivo, de monitorar a ingestão de alimentos – uma ferramenta que seria útil para estudos de saúde envolvendo obesidade ou doenças relacionadas à dieta.

Outro projeto do *PlaceLab* é o desenvolvimento de um dispositivo que ajudaria as pessoas a aderir a horários de medicação complexos, dando-lhes lembretes personalizados ao longo do dia, por exemplo, piscar uma luz na geladeira quando for hora de tomar um determinado medicamento com comida.

Vídeos, artigos e teses, publicados entre 1999 e 2007, estão disponíveis para o público no *site* do MIT.

Em 2002, Intille escreveu um artigo chamado “*Designing a home of the future*”. Nesse artigo, Intille explica sobre as tecnologias e as estratégias de *design* que usavam detecção sensível ao contexto para capacitar as pessoas com informações, apresentando-as no momento e lugar certo. O artigo foi muito importante para o desenvolvimento de diversos recursos (explicados na próxima seção) que o *PlaceLab* utiliza atualmente.

5.1.2 Recursos do *PlaceLab*

Além de possuir uma rica infraestrutura para estudos, os pesquisadores da *House_n* contam com diversos recursos para estudo nas casas inteligentes. Os principais recursos do *PlaceLab* incluem:

- a) **Componentes interiores:** O interior do *PlaceLab* é formado por 15 componentes de armários pré-fabricados que podem ser reconfigurados de acordo com as necessidades dos pesquisadores.
- b) **Redes de sensores:** Cada um dos componentes interiores contém um micro controlador e rede de 25 a 30 sensores. Novos sensores podem ser adicionados a rede caso seja necessário.
- c) **Sensibilização ambiental:** Cada componente interior tem uma matriz de sensores ambientais. Incluindo monóxido de carbono (CO), gás carbônico (CO₂), temperatura e umidade. Sensores ambientais adicionais podem ser adicionados se necessário.
- d) **Sensores de estado:** Pequenos sensores, com fio e sem fio, instalados nos objetos que as pessoas tocam e usam, incluindo portas de armário e gaveta, controles, móveis, portas, recipientes de cozinha, entre outros. Eles detectam eventos do tipo ligar/desligar, aberto/fechado e movimentação dos objetos.

- e) **Detecção de áudio:** Microfones quase invisíveis são instalados em cada componente interior para capturar o áudio.
- f) **Captura de imagem e vídeo:** Um sofisticado sistema de captura de vídeo processa imagens capturadas por câmeras de luz visível e infravermelho, colocadas atrás de painéis de cada componente interior.
- g) **Comunicação por áudio:** Autofalantes estéreo são instalados em cada componente interior, permitindo que o áudio seja dirigido conforme necessário.
- h) **Iluminação:** A intensidade e a temperatura da cor da luz em cada cômodo podem ser controladas dinamicamente, permitindo que a luz seja usada como ferramenta de comunicação e para fins estéticos.
- i) **Controle ambiental:** O sistema de *HVAC* (dispositivos de aquecimento e arrefecimento) contém várias zonas, um permutador de calor e um sofisticado sistema para filtrar o ar dos ambientes.

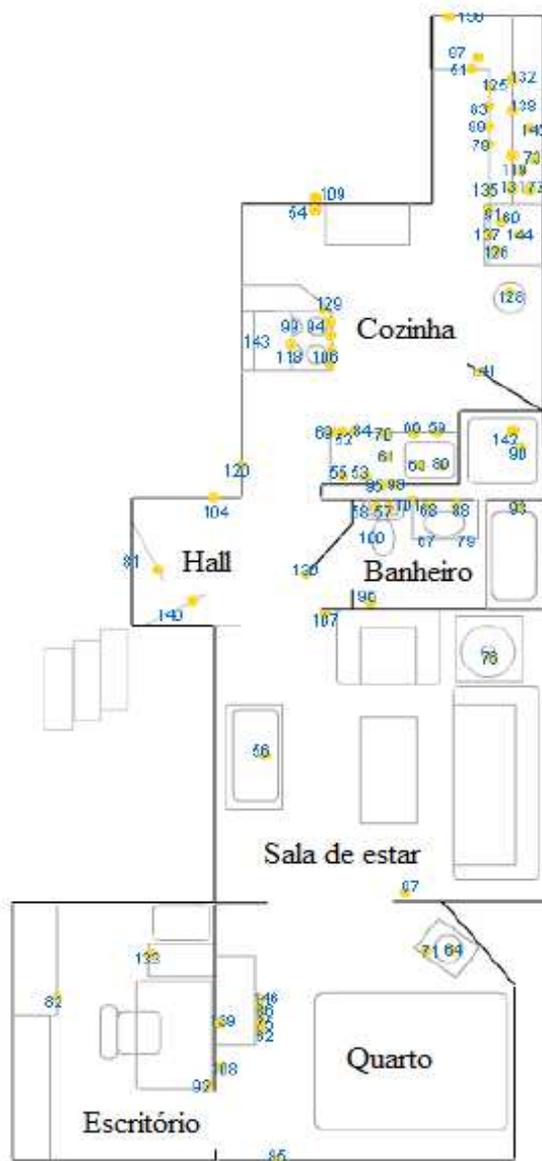
5.2 Descrição da base de dados

Conforme citado na seção anterior, a base de dados utilizadas neste trabalho é do *Massachusetts Institute of Technology* (MIT) de 2007. O *dataset* utilizado refere-se a uma casa inteligente (*PlaceLab*) com um único habitante vivendo nela por um período de catorze dias. O *dataset* é público e está disponível para download na página projeto *House_n* do MIT.

A casa está equipada com 77 sensores, de diferentes tipos: sensores de proximidade, sensores de aceleração (acelerômetros), sensores de vibração, sensores de movimentos, sensores de temperatura, sensores de pressão, sensores de luz e sensores de mudança de estado. O Apêndice A lista os locais que os sensores foram instalados e a quantidade de sensores instalados em cada local. Maiores detalhes sobre os sensores são encontrados no Apêndice B.

Todos os sensores foram instalados em objetos do dia a dia, tais como armários, gavetas, geladeira, portas, entre outros com o intuito de gravar eventos do tipo abrir/fechar (eventos de ativação/desativação dos sensores) enquanto o habitante executar suas tarefas diárias, dessa forma é possível obter a sequência de ações do mesmo. A Figura 25 ilustra os locais de instalação dos sensores.

Figura 25 – Locais de instalação dos sensores.

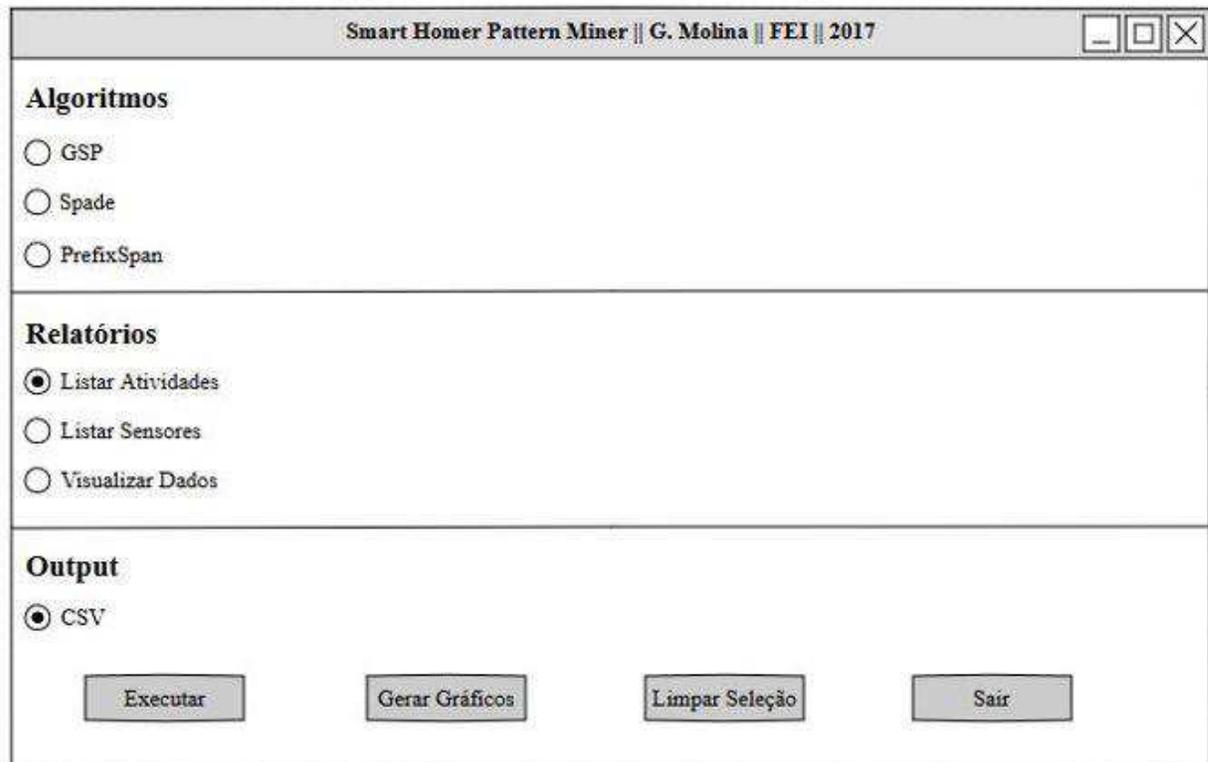


Fonte: MIT, 2016.

5.3 Análise dos algoritmos

Visando testar o comportamento, eficiência e desempenho dos algoritmos, para extrair padrões sequenciais, neste trabalho foi criado um *software* em computador denominado *Smart Homer Pattern Miner*. A interface do programa é mostrada na Figura 26.

Figura 26 – Interface do Smart Homer Pattern Miner.



Fonte: Autor.

Os objetivos do *software* desenvolvido são apresentados abaixo:

a) Visualização dos dados: Ao selecionar essa opção, o programa cria um arquivo data.CSV com os seguintes parâmetros: *action* (nome da ação realizada pelo habitante), *start time* (o horário em que a ação iniciou), *end time* (o horário em que a ação terminou). Por *default* (padrão) o arquivo gerado mostra as ações que o habitante realizou no dia. É possível visualizar as ações realizadas durante toda a semana ou visualizar as ações

realizadas durante todo o período que o habitante permaneceu no *PlaceLab* (01/09/2007 a 14/09/2007). A Figura 27 ilustra um exemplo do arquivo gerado.

b) Criação de rotina: Ao selecionar essa opção, o programa recebe os dados advindos dos sensores e gera um arquivo com todas as ações realizadas pelo habitante. Como as ações são ordenadas temporalmente é possível saber a sequência em que as mesmas foram criadas, logo pode-se aplicar os algoritmos GSP, SPADE e PrefixSpan para a extração dos padrões sequenciais.

c) Geração de gráficos: Com o arquivo .CSV que é gerado pelo *software* é possível gerar gráficos para observar e analisar o desempenho/eficiência de cada um dos algoritmos utilizados neste trabalho.

d) Listar Sensores: Ao selecionar essa opção, o programa gera o arquivo sensors.CSV que mostra o *ID* e a localização de casa sensor instalado no *PlaceLab*.

e) Listar atividades: Ao selecionar essa opção, o programa gera o arquivo activities.CSV que mostra todas as atividades cadastradas no *dataset* bem com o código de identificação de cada uma. A Figura 28 ilustra o arquivo gerado.

Figura 27– Tela do arquivo data. CSV.

09/01/2007			
Action	Start Time	End Time	
Preparing breakfast	8.15 AM	8.20 AM	
Bathing	8.25 AM	8.45 AM	
Grooming	8.46 AM	9.00 AM	
Going out for shopping	9.05 AM	4.00 PM	
Resting	4.00 PM	7.30 PM	
Preparing a beverage	7.35 PM	7.40 PM	
Watching Tv	7.40 PM	9.30 PM	
Talking on the phone	7.55 PM	8.07 PM	
Listening to music	9.35 PM	9.55 PM	
Sleeping	10.00 PM	6.00 AM	

Fonte: Autor.

Figura 28 – Tela do arquivo activities. CSV.

Heading	Category	Subcategory	Code
Employment related	Employment work at home	Work at home	1
Employment related	Travel employment	Going out to work	5
Personal needs	Eating	Eating	10
Personal needs	Personal hygiene	Toileting	15
Personal needs	Personal hygiene	Bathing	20
Personal needs	Personal hygiene	Grooming	25
Personal needs	Personal hygiene	Dressing	30
Personal needs	Personal hygiene	Washing hands	35
Personal needs	Personal medical	Taking medication	40
Personal needs	Sleeping	Sleeping	45
Personal needs	Talking on telephone	Talking on telephone	50
Personal needs	Resting	Resting	55
Domestic work	Preparing a meal	Preparing breakfast	60
Domestic work	Preparing a meal	Preparing lunch	65
Domestic work	Preparing a meal	Preparing dinner	70
Domestic work	Preparing a snack	Preparing a snack	75
Domestic work	Preparing a beverage	Preparing a beverage	80
Domestic work	Meal Cleanup	Washing dishes	85
Domestic work	Meal Cleanup	Putting away dishes	90
Domestic work	Putting away groceries	Putting away groceries	95
Domestic work	Clean house	Cleaning	100
Domestic work	Clean house	Doing Laundry	105
Domestic work	Clean house	Putting away laundry	110
Domestic work	Outdoor chores	Taking out the trash	115
Domestic work	Outdoor chores	Lawnwork	120
Educational	Home education	Home education	125
Educational	Travel study	Going out to school	130
Leisure	Leisure	Watching TV	135
Leisure	Leisure	Listening to music	140
Entertainment and social life	Travel social	Going out for entertainment	145
Sport	Travel sports	Going out to exercise	150
Purchasing goods and services	Travel services	Going out for shopping	155

Fonte: Autor.

A interface do *Smart Homer Pattern Miner* foi desenvolvida com o *software* PENCIL (ferramenta de protótipo de interface gráfica do usuário, conhecida como GUI, de código aberto). A linguagem de programação utilizada foi o C++ e para a compilação, foi utilizado o compilador Dev – C++. Os algoritmos GSP, SPADE e PrefixSpan foram implementados com base nos pseudocódigos apresentados na Seção 4.

Para testar o *software*, foram gerados arquivos data.CSV com diferentes parâmetros. Maiores detalhes sobre os testes e configurações de parâmetros serão apresentados na próxima seção.

6 TESTES REALIZADOS

Conforme explicado na seção anterior, foi usada a base de dados do MIT de 2007, com as ações realizadas pelo habitante do *PlaceLab* em um período de 14 dias (01/09/2007 a 14/09/2007). Todos os *datasets* disponíveis no site do MIT são públicos e apresentam as ações do habitante em períodos de 14 dias.

Devido ao propósito do uso dos algoritmos GSP, SPADE e PrefixSpan para a extração de sequências em um sistema de automação residencial real, os testes efetuados com o *dataset* focam os seguintes pontos principais: precisão da extração dos padrões e qualidade das extrações.

A precisão da extração de padrões se refere ao tempo necessário para identificação e extração dos padrões (levando-se em conta o consumo de memória do algoritmo).

A qualidade das extrações se refere à porcentagem de padrões que o algoritmo consegue identificar corretamente.

Conforme citado na Seção 5.3, quando o *software* recebe os dados advindos dos sensores e cria o arquivo.CSV com as ações realizadas pelos habitantes, as mesmas são ordenadas temporalmente, ou seja, o comportamento sequencial nesta base é definido pelos horários de início e término dos eventos. Os algoritmos GSP, Spade e PrefixSpan foram criados com base nas relações temporais de Allen (conforme citado na Seção 2.6), dessa forma, a questão de temporalidade é tratada naturalmente pelos algoritmos.

No contexto deste trabalho, extrair corretamente um padrão significa extrair sequencialmente as ações realizadas pelo habitante na ordem em que as mesmas ocorreram, ou seja, pode-se saber se o eletrodoméstico Y foi ligado antes ou depois do eletrodoméstico X.

O funcionamento ideal do sistema prevê a extração correta dos padrões sequenciais, de forma a possibilitar que seja possível identificar qual dos algoritmos (GSP, SPADE ou PrefixSpan) é mais interessante para ser aplicado em casas inteligentes.

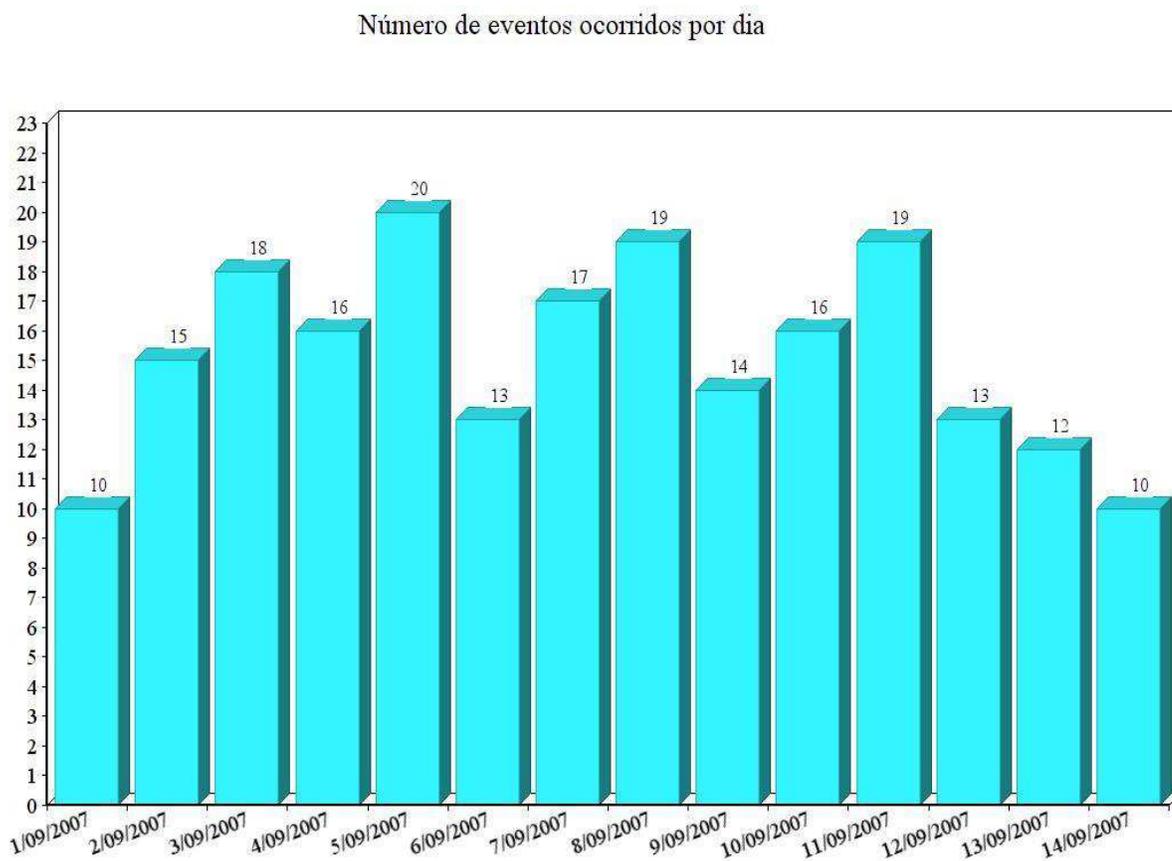
A partir da próxima seção são expostos todos os dados e parâmetros utilizados nas simulações dos testes.

6.1 Dataset Utilizado para as Simulações

Para que fosse possível comparar os resultados obtidos com diferentes parâmetros de extração, o mesmo *dataset* foi utilizado para todas as simulações. Desta forma, foi possível analisar o comportamento do *software* com o uso de diferentes critérios e valores de teste, sem a influência de particularidades entre diferentes *datasets*.

Antes dos testes serem iniciados, foi gerado o gráfico da Figura 29 que ilustra o número de eventos (ações) ocorridos por dia no *PlaceLab*.

Figura 29 - Número de eventos ocorridos por dia.



Fonte: Autor.

O *dataset* apresenta 212 eventos, sendo que a média é de 15 eventos por dia.

6.2 Hardware Utilizado nos Testes

Todos os testes foram realizados em um computador (PC) com as seguintes especificações:

- a) SISTEMA OPERACIONAL: Windows 8.1 x64
- b) CPU: AMD A4-1200, *dual core @ 1.0 GHz*
- c) RAM: 2 GB, DDR3 @1600 MHz
- d) HD: 320 GB SATA, 540 MB/s *sequential reading*
- e) VÍDEO: AMD Radeon HD 8180 @ 1.0 GHz

O código foi executado na linguagem de programação C++.

6.3 Avaliação da Precisão da Extração

A avaliação da precisão da extração de padrões se refere ao tempo necessário para identificação e extração dos padrões (levando-se em conta o consumo da memória). O teste foi dividido em dois momentos distintos para melhor visualização dos dados: em um primeiro momento foram analisados os tempos de execução (*run time*) de cada algoritmo e posteriormente foi analisado o consumo de memória de cada um.

Para a mensuração da memória, foi implementada uma função em C++ denominada `clock` (), da classe `ctime`, cuja finalidade é verificar se algum dos algoritmos explodem ou destoam o uso da memória.

- a) **Etapa 1:** Foi gerado o arquivo 01_09.CSV com todas as ações realizadas pelo habitante no dia 01/09/2007. Essa data foi escolhida, pois é o dia que possui menos eventos no *dataset*, ou seja, foi o dia em que o habitante realizou menos ações dentro do *PlaceLab*.
- b) **Etapa 2:** Foi gerado o arquivo 05_09.CSV com todas as ações realizadas pelo habitante no dia 05/09/2007. Essa data foi escolhida, pois é o dia que possui mais eventos no *dataset*, ou seja, foi o dia em que o habitante realizou mais ações dentro do *PlaceLab*.
- c) **Etapa 3:** Foi gerado o arquivo 10_base.CSV que corresponde a 10% do total de

todas as ações realizadas pelo habitante no período de 01/09/2007 a 14/09/2007. Foi calculada a média dos 10% da base utilizando-se dias diferentes para se chegar ao valor mais próximo. Ao utilizar 10% da base, foram totalizados 30 eventos, dessa forma é possível observar se há alguma mudança no comportamento dos algoritmos com relação às etapas anteriores.

d) **Etapa 4:** Foi gerado o arquivo 30_base.CSV que corresponde a 30% do total de todas as ações realizadas pelo habitante no período de 01/09/2007 a 14/09/2007. Foi utilizado o cálculo da média dos 30% da base, ou seja, foi utilizado o mesmo procedimento do cálculo dos 10%. Ao utilizar 30% da base, foram totalizados 69 eventos, mais do que o dobro de eventos do teste anterior, dessa forma é possível observar como os algoritmos se comportam com um número maior de eventos.

e) **Etapa 5:** Foi gerado o arquivo 50_base.CSV que corresponde a 50% do total de todas as ações realizadas pelo habitante no período de 01/09/2007 a 14/09/2007. Ao utilizar 50% da base, foram totalizados 109 eventos.

f) **Etapa 6:** A última etapa do teste consistiu na geração do arquivo 100_base.CSV que corresponde as ações realizadas pelo habitante no período de 01/09/2007 a 14/09/2007. A base possui 212 eventos no total.

Com o intuito de analisar a influência da quantidade de dados no resultado dos algoritmos, os testes foram fracionados em: 10% da base, 30% da base, 50% da base e 100% da base.

Em um primeiro momento, o desempenho dos três algoritmos foi comparado utilizando-se o suporte mínimo $\text{minSup} = 0,01$. Para os testes iniciais foram escolhidos padrões (sequências de ações) com o comprimento mínimo de 2 eventos (ações) e comprimento máximo de 7 eventos (ações).

No contexto desse trabalho, o comprimento representado por C será considerado o tamanho da sequência a ser analisada pelos algoritmos. Os valores de comprimento mínimo e máximo adotados foram escolhidos com base em diversas pesquisas realizadas pelo *House_n*.

Essas configurações de parâmetros são consideradas como representativas no *dataset* utilizado por que:

- a) O suporte mínimo minSup de 0,01 (ou 1%) permite que os algoritmos sejam executados mais rapidamente do que quando um suporte mínimo baixo (por exemplo 0,001 ou menos) é escolhido. Valores mais altos para o suporte mínimo (0,1 ou acima) não produzem padrões para o *dataset* utilizado e, portanto, não podem ser utilizados nos testes.
- b) Padrões frequentes constituídos por menos de dois eventos não são interessantes ao tentar descobrir o que constitui um comportamento normal.
- c) Padrões constituídos por mais de sete eventos sequenciais são improváveis de ocorrer no dia a dia.

Embora dificilmente ocorram padrões constituídos por mais de sete eventos sequenciais no dia a dia do habitante do *dataset* utilizado, o comprimento máximo dos padrões foi aumentado de 7 eventos para 10 eventos, a fim de se verificar o que ocorre com o tempo de execução dos algoritmos. Os resultados dos testes são apresentados a seguir.

Os gráficos das Figuras 30, 32, 34, 36, 38 e 40 mostram os resultados dos testes dos tempos de execução (*run time*) e os gráficos das Figuras 31, 33, 35, 37, 39 e 41 mostram o consumo de memória dos algoritmos.

Figura 30 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 01/09/2007.



Fonte: Autor.

Analisando-se a Figura 30 é possível observar que quando o comprimento = 2 – 7 e suporte mínimo $\text{minSup} = 0,01$ (ou 1%) os algoritmos GSP e Spade apresentam tempos de execução bem próximos, visto que o GSP foi executado em 0,9s e o Spade em 0,89s. Como a base de dados é pequena, apresenta apenas 10 eventos, a etapa de geração e contagem dos

candidatos não influenciou o tempo de processamento do algoritmo GSP. Devido a projeção de prefixos para geração de padrões sequenciais, o algoritmo PrefixSpan apresentou melhor desempenho (0,84s).

Ao alterar o valor do comprimento = 2 – 7 para comprimento = 2 – 10 e manter minSup = 0,01 o tempo de execução dos algoritmos permaneceu o mesmo. Isso ocorreu, pois, padrões constituídos por mais de sete eventos sequenciais no dia a dia do habitante do *dataset* utilizado dificilmente ocorrem.

Quando minSup = 0,001 e comprimento = 2 – 7 o tempo de execução dos três algoritmos aumentou: o GSP passou a ser executado em 0,92s, o Spade em 0,9s e o PrefixSpan 0,84s. Ao baixar o suporte mínimo, mais padrões sequenciais foram encontrados e, com isso, o tempo de execução dos algoritmos foi maior. As etapas de geração e contagem dos candidatos influenciaram no consumo de memória do algoritmo GSP que aumentou de 1,5 MB para 1,8 MB. A verticalização dos dados usada no algoritmo Spade também fez com que o consumo de memória passasse de 1,65 MB para 1,72 MB. O consumo de memória do algoritmo PrefixSpan se manteve estável.

Ao alterar o valor do comprimento = 2 – 7 para comprimento = 2 – 10 e manter minSup = 0,001, tanto o tempo de quanto o consumo de memória dos três algoritmos permaneceram o mesmo.

Figura 31 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 01/09/2007.



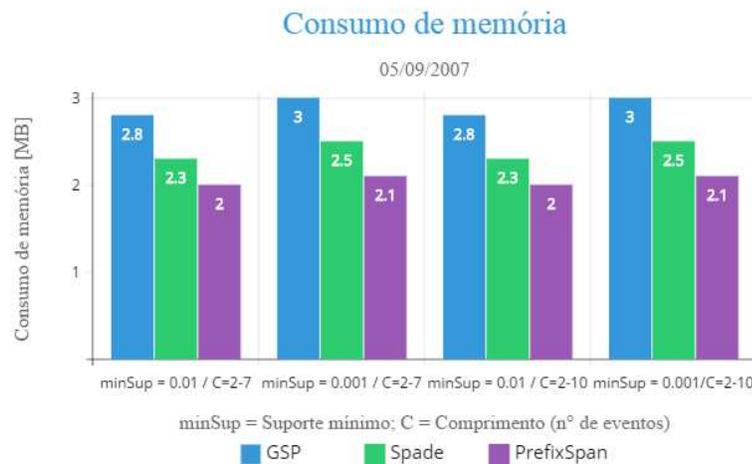
Fonte: Autor.

Figura 32 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 05/09/2007.



Fonte: Autor.

Figura 33 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 05/09/2007.



Fonte: Autor.

Analisando-se as Figuras 32 e 33 é possível observar que quando o número de eventos dobrou (20 eventos), o consumo de memória também dobrou e houve um aumento considerável no tempo de execução de todos os algoritmos.

O GSP foi executado em 1,36s consumindo 2,8 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,39s e foram consumidos 3,0 MB de memória. Ao alterar o valor do comprimento de

$C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. Nota-se que dos três algoritmos utilizados o GSP é o que apresenta um desempenho pior, pois, o GSP é baseado no algoritmo *Apriori*, logo apresenta como desvantagem a geração de uma grande quantidade de sequências candidatas o que influencia na velocidade de processamento dos dados.

O Spade foi executado em 1,24s consumindo 2,3 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,26s e foram consumidos 2,5 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. O algoritmo SPADE apresentou um desempenho melhor que o GSP devido a otimizações que reduzem o consumo da memória e melhoram a sua eficiência.

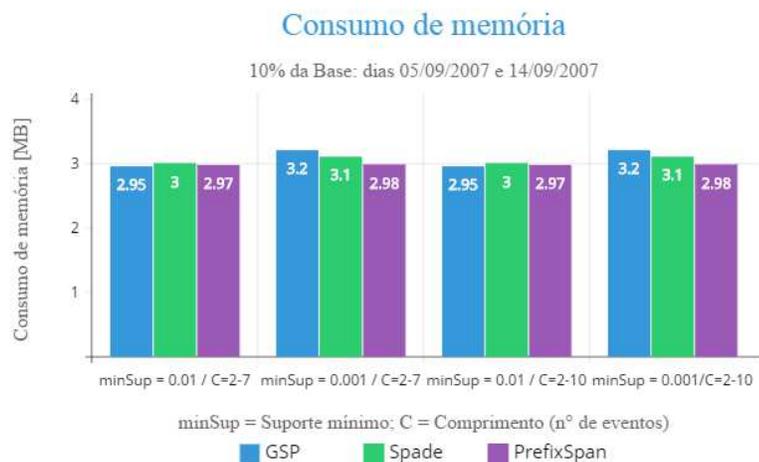
O PrefixSpan foi executado em 1,16s consumindo 2 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,17s e foram consumidos 2,1 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. O PrefixSpan utiliza a estratégia de dividir e conquistar por meio de projeções da base de dados. Estas projeções dividem a BD (Base de dados) em porções menores, dessa forma, o espaço de busca é menor e a identificação e extração das sequências frequentes são realizadas de forma mais rápida e eficaz. Dentre os 3 algoritmos o PrefixSpan foi o que apresentou o melhor desempenho.

Figura 34 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 10% da base.



Fonte: Autor.

Figura 35 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 10% da base.



Fonte: Autor.

Analisando-se as Figuras 34 e 35 é possível observar que quando os testes foram realizados em 10% da base, totalizando 30 eventos, o tempo de execução e o consumo de memória de todos os algoritmos aumentaram em relação ao teste anterior.

O GSP foi executado em 1,14s consumindo 2,95 MB de memória (com minSup = 0,01 e C = 2 – 7) e seu tempo e consumo de memória permaneceram os mesmos quando minSup = 0,01 e C = 2 – 10. Quando o minSup foi alterado para minSup = 0,001 e C = 2 – 7, o tempo de

execução subiu para 1,17s e foram consumidos 3,2 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. Embora o aumento no tempo de execução tenha sido pequeno, é possível perceber pelo aumento no consumo da memória que as etapas de geração e contagem de candidatos e poda baseada no suporte influenciam o desempenho do algoritmo.

O Spade foi executado em 1,06s consumindo 3 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,09s e foram consumidos 3,1 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. O aumento no consumo de memória foi causado pelas propriedades combinatórias que o algoritmo utiliza para decompor o problema original em subproblemas menores. Foi a primeira vez que o Spade consumiu quase a mesma quantidade de memória que o GSP, embora tenha apresentado tempos de execução diferentes.

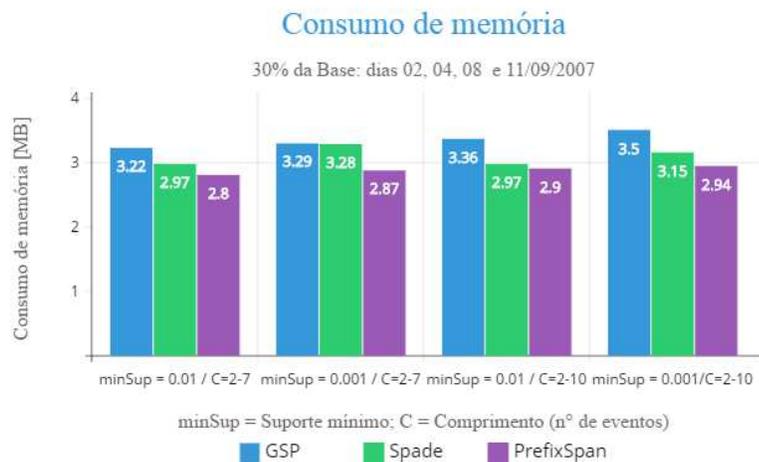
O PrefixSpan foi executado em 0,97s consumindo 2,97 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 0,99s e foram consumidos 2,98 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, tanto o tempo de execução quanto o consumo de memória permaneceram os mesmos. O PrefixSpan novamente apresentou o melhor desempenho.

Figura 36 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 30% da base.



Fonte: Autor.

Figura 37 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 30% da base.



Fonte: Autor.

Analisando-se as Figuras 36 e 37 é possível observar que quando os testes foram realizados em 30% da base, totalizando 69 eventos, o tempo de execução e o consumo de memória de todos os algoritmos aumentaram em relação ao teste anterior.

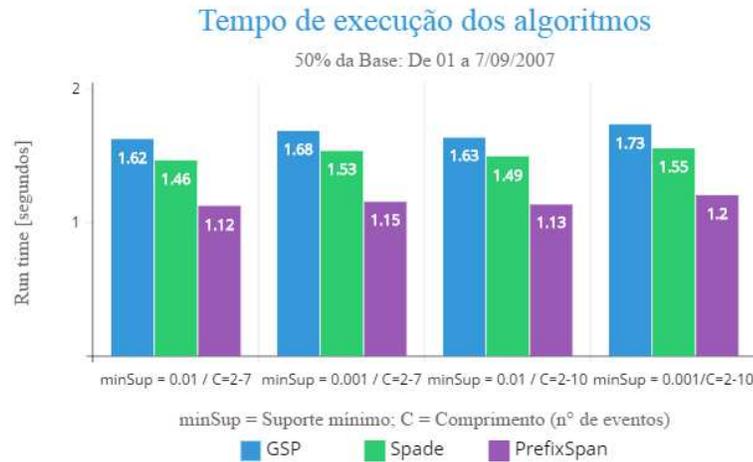
O GSP foi executado em 1,53s consumindo 3,22 MB de memória (com minSup = 0,01 e C = 2 – 7), seu tempo de execução permaneceu o mesmo quando minSup = 0,01 e C = 2 – 10, porém o consumo de memória aumentou para 3,29 MB. Quando o minSup foi alterado para

$\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,59s e foram consumidos 3,36 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, o tempo de execução permaneceu o mesmo, porém o consumo de memória aumentou para 3,5 MB. Embora o aumento no tempo de execução tenha sido pequeno, é possível perceber pelo aumento no consumo da memória que as etapas de geração e contagem de candidatos e poda baseada no suporte influenciam o desempenho do algoritmo.

O Spade foi executado em 1,29s consumindo 2,97 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$) e seu tempo e consumo de memória permaneceram os mesmos quando $\text{minSup} = 0,01$ e $C = 2 - 10$. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,35s e foram consumidos 3,28 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, o tempo de execução permaneceu o mesmo, porém o consumo de memória baixou para 3,15 MB. O aumento no consumo de memória foi causado pelas propriedades combinatórias que o algoritmo utiliza para decompor o problema original em subproblemas menores. A diminuição no consumo de memória de 3,28 MB para 3,15 MB ocorreu, pois, menos padrões sequenciais foram identificados e extraídos pelo algoritmo.

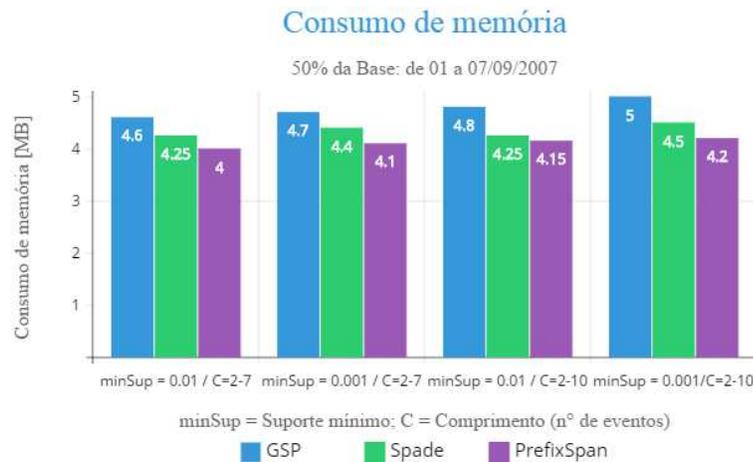
O PrefixSpan foi executado em 1,03s consumindo 2,8 MB de memória (com $\text{minSup} = 0,01$ e $C = 2 - 7$), seu tempo permaneceu o mesmo quando $\text{minSup} = 0,01$ e $C = 2 - 10$, porém seu consumo de memória teve um pequeno aumento, consumindo 2,87 MB. Isso ocorreu, pois foram identificados e extraídos mais padrões sequenciais do que no caso anterior. Quando o minSup foi alterado para $\text{minSup} = 0,001$ e $C = 2 - 7$, o tempo de execução subiu para 1,09s e foram consumidos 2,9 MB de memória. Ao alterar o valor do comprimento de $C = 2 - 7$ para $C = 2 - 10$ e manter $\text{minSup} = 0,001$, o tempo de execução permaneceu o mesmo, porém o consumo de memória aumentou para 2,94 MB. Isso ocorreu, pois foram identificados e extraídos mais padrões sequenciais do que no caso anterior. O aumento no consumo de memória ocorreu, pois foram identificados e extraídos mais padrões sequenciais do que no caso anterior, ou seja, o algoritmo precisou fazer mais projeções da base de dados. O PrefixSpan novamente apresentou o melhor desempenho.

Figura 38 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 50% da base.



Fonte: Autor.

Figura 39 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 50% da base



Fonte: Autor.

Analisando-se as Figuras 38 e 39 é possível observar que quando os testes foram realizados em 50% da base, totalizando 109 eventos, torna-se nítida a diferença de desempenho dos três algoritmos com relação ao tempo de execução de cada um.

O algoritmo PrefixSpan é o que apresenta melhor desempenho, pois, além de consumir menos memória em todos os casos, seus tempos de execução são melhores que os outros algoritmos em todos os casos. O algoritmo PrefixSpan é mais eficiente do que os algoritmos baseados no algoritmo *Apriori*. Seu consumo de memória foi estável.

O algoritmo Spade foi o que apresentou o segundo melhor desempenho, apresentando um consumo de memória estável e tempos de execução próximos a 1,5s. O Spade utiliza propriedades combinatórias para decompor o problema original em subproblemas menores, que podem ser resolvidos de forma independente, logo apresenta um tempo maior do que o algoritmo PrefixSpan. O algoritmo SPADE apresenta um desempenho melhor que o GSP visto que apresenta otimizações que reduzem o consumo da memória e melhoram a sua eficiência.

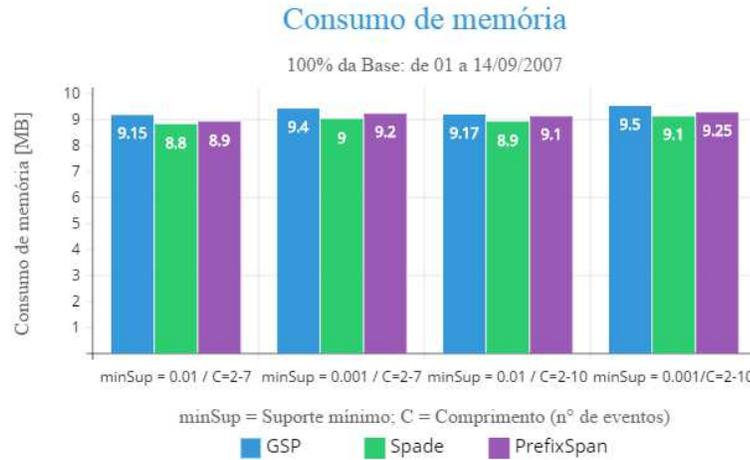
Dentre os três algoritmos testados, o GSP foi o que apresentou pior desempenho. Ele consumiu mais memória do que os outros algoritmos e apresentou tempos de execução próximos a 1,7s. O GSP é baseado no algoritmo *Apriori*, logo apresenta como desvantagem a geração de uma grande quantidade de sequências candidatas o que influencia na velocidade de processamento dos dados.

Figura 40 - Tempo de execução dos algoritmos GSP, Spade e PrefixSpan em 100% da base.



Fonte: Autor

Figura 41 - Consumo de memória dos algoritmos GSP, Spade e PrefixSpan em 100% da base.



Fonte: Autor.

Analisando-se as Figuras 40 e 41 percebe-se de forma clara que com relação ao tempo de execução, o algoritmo PrefixSpan é o melhor método para extração de padrões sequenciais de comportamentos em casas inteligentes. O algoritmo Spade apresentou um resultado mediano nos testes apresentando tempos de execução razoáveis e consumos estáveis de memória. O algoritmo GSP apresentou os piores resultados nos testes, apresentando os maiores tempos de execução e consumindo mais memória que os outros algoritmos.

Com os resultados obtidos conclui-se que o comportamento dos algoritmos é linear.

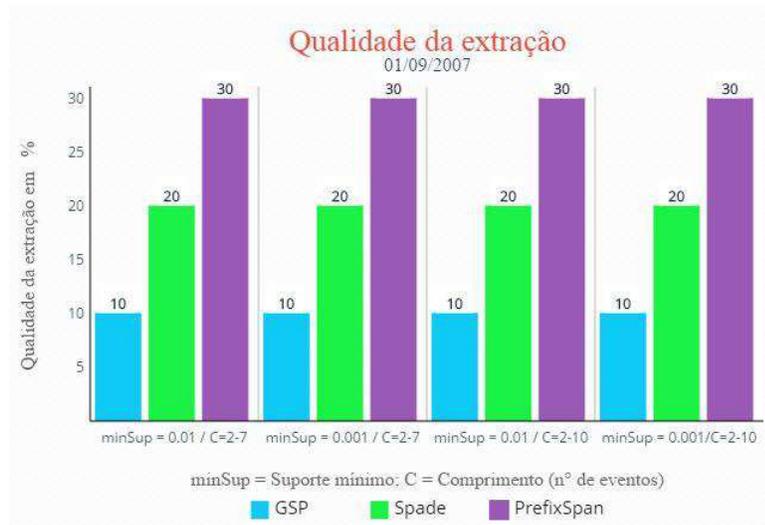
6.4 Avaliação da qualidade das extrações

Conforme citado anteriormente, a qualidade das extrações se refere a porcentagem de padrões que o algoritmo consegue identificar e extrair corretamente.

Após análise do consumo de memória dos algoritmos, o último teste focou na qualidade das extrações.

Em todos os testes as configurações de parâmetros do capítulo 5.3 foram mantidas. Os resultados obtidos são apresentados nos gráficos a seguir.

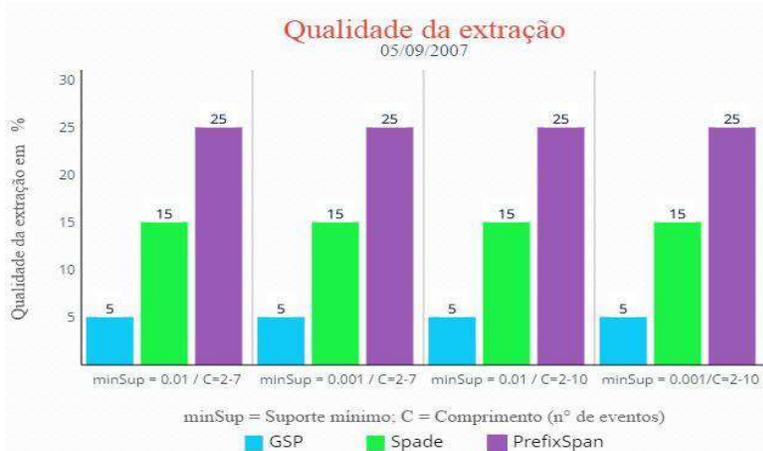
Figura 42 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 01/09/2007.



Fonte: Autor.

Analisando-se a Figura 42, nota-se que a qualidade das extrações obtidas foi baixa. Nesse dia o habitante realizou 10 ações, ou seja, o GSP identificou e extraiu corretamente 10% das ações realizadas pelo habitante, o Spade identificou e extraiu corretamente 20% das ações e o PrefixSpan identificou e extraiu corretamente 30% das ações.

Figura 43 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 05/09/2007.



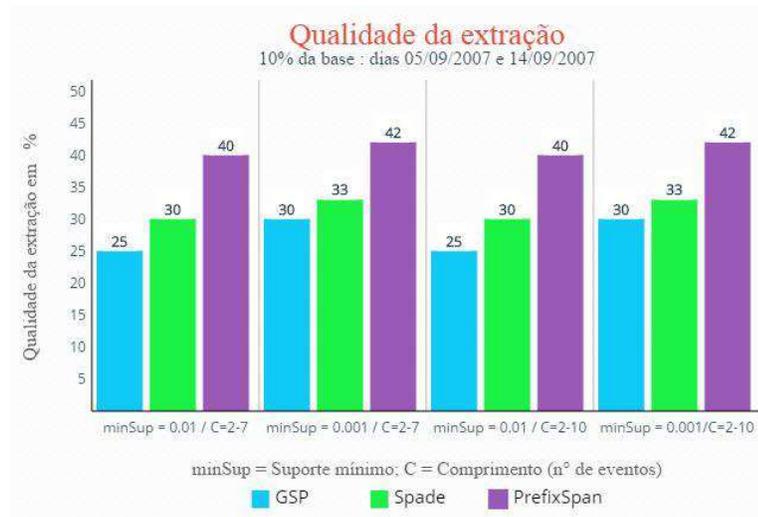
Fonte: Autor.

Analisando-se a Figura 43, nota-se que a qualidade das extrações obtida foi mais baixa do que no dia 01/09/2007. Todos os algoritmos apresentaram uma queda de 5% na qualidade de extração.

O GSP apresentou uma qualidade de extração de 5%, o Spade uma qualidade de extração de 15% e o PrefixSpan uma qualidade de extração de 25%.

Observando-se os resultados obtidos no dia 01/09/2007, 05/09/2007 e 13/09/2007 pode-se concluir que quando há poucos eventos sendo analisados, independente do valor do suporte mínimo e do comprimento, os três algoritmos apresentaram valores baixos de qualidade de extração.

Figura 44 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 10% da base.



Fonte: Autor.

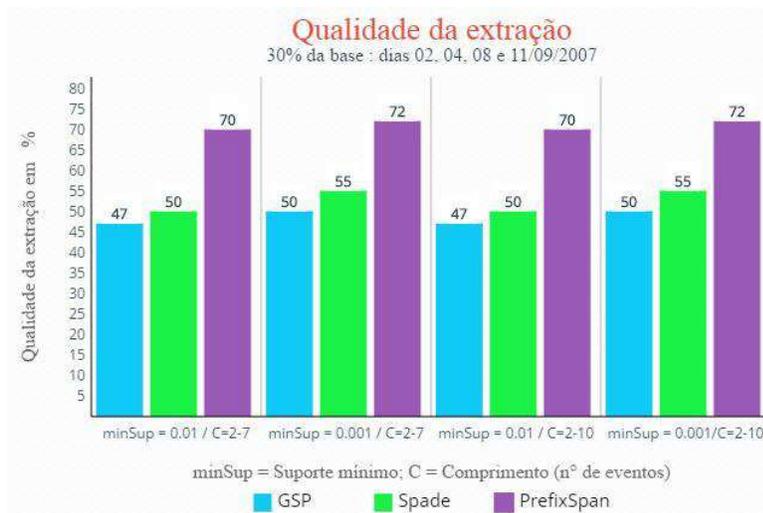
Observa-se na Figura 44 que ao utilizar 10% da base há um aumento considerável na qualidade de extração dos três algoritmos.

O GSP apresentou qualidade de extração de 25% quando o suporte mínimo utilizado foi $\text{minSup} = 0,01$ (o comprimento não alterou a qualidade) e 30% quando o $\text{minSup} = 0,001$, o que demonstra uma melhora significativa de desempenho.

O Spade apresentou um desempenho melhor que o GSP, pois, o mesmo apresentou qualidade de extração de 30% quando o suporte mínimo utilizado foi $\text{minSup} = 0,01$ (o comprimento não alterou a qualidade) e 33% quando o $\text{minSup} = 0,001$.

O PrefixSpan foi o algoritmo que apresentou melhor qualidade de extração, pois, o mesmo apresentou qualidade de extração de 40% quando o suporte mínimo utilizado foi $\text{minSup} = 0,01$ (o comprimento não alterou a qualidade) e 42% quando o $\text{minSup} = 0,001$. O algoritmo apresentou um desempenho 15% melhor do que os outros dias.

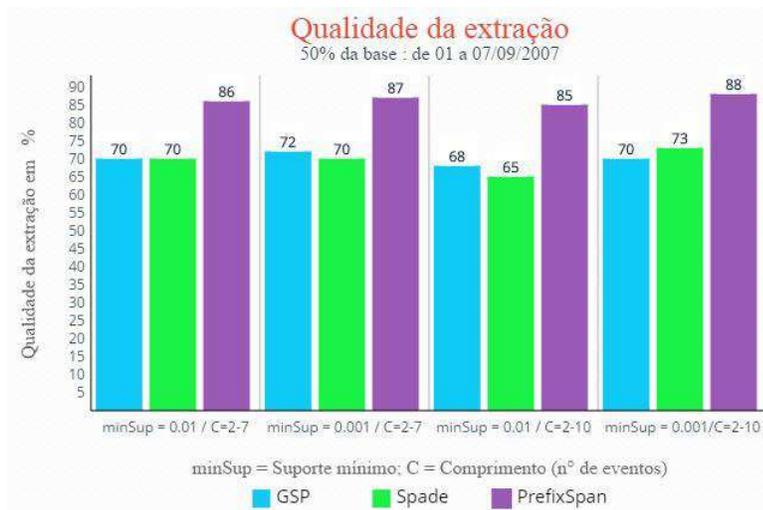
Figura 45 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 30% da base.



Fonte: Autor.

Nota-se na Figura 45 que ao utilizar 30% da base o desempenho dos algoritmos melhora. O GSP e o Spade apresentaram uma melhora de 20% na qualidade de extração. O PrefixSpan apresentou uma melhora de 40% na qualidade de extração, apresentando qualidade de extração de 70% quando $\text{minSup} = 0,01$ e 72% quando $\text{minSup} = 0,001$.

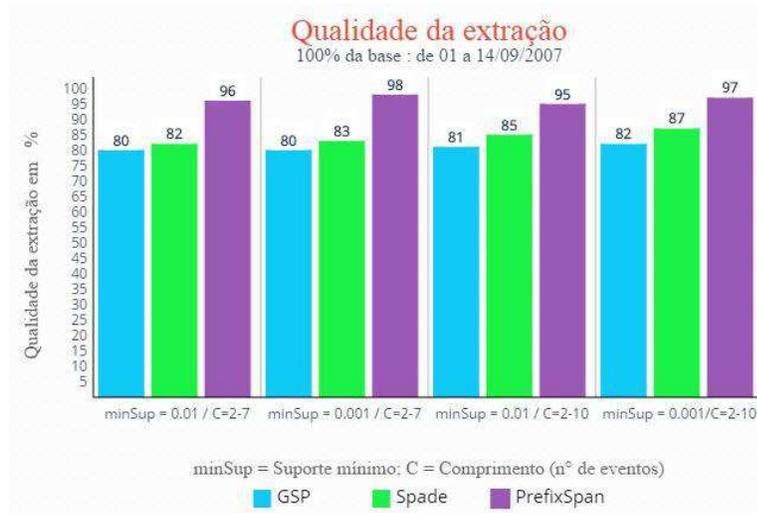
Figura 46 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 50% da base.



Fonte: Autor.

Nota-se na Figura 46 que ao utilizar 50% da base, totalizando 109 eventos, o desempenho dos algoritmos melhora. O GSP e o Spade apresentaram qualidades de extrações próximas a 70%. Em relação aos testes anteriores, observa-se que o GSP apresentou uma melhora significativa. Pode-se dizer que mesmo apresentando tempos de execução e consumo de memória maior que os outros algoritmos, o GSP apresentou um bom desempenho para identificar e extrair padrões corretamente. O PrefixSpan apresentou uma qualidade de extração média de 87%.

Figura 47 - Qualidade da extração dos algoritmos GSP, Spade e PrefixSpan em 100% da base.



Fonte: Autor.

Analisando-se a Figura 47, onde foi testada a qualidade das extrações em toda a base, totalizando 212 eventos, observa-se que o PrefixSpan apresentou um ótimo desempenho, conseguindo identificar e extrair praticamente todas as ações realizadas pelo habitante. O Spade apresentou o segundo melhor desempenho com qualidade de extração média de 85%. O GSP apresentou qualidade de extração média de 80%.

Após os testes, foi gerado o arquivo *activities.csv* para identificar quais foram as sequências de ações que os algoritmos não conseguiram identificar e extrair corretamente. O algoritmo GSP teve problemas para identificar e extrair as ações que ocorreram no banheiro (ações que utilizavam o sensor de temperatura e o sensor de proximidade). Os sensores de temperatura e de proximidade instalados no *PlaceLab*, são de alta precisão e apresentam valores contínuos que afetaram a qualidade de extração do GSP pois, o mesmo foi projetado para tratar valores numéricos. O Spade também teve problemas para identificar e extrair as ações que

ocorreram no banheiro, porém ao contrário do GSP, ele apenas teve problemas com o sensor de temperatura, ou seja, conseguiu identificar e extrair as ações que utilizaram o sensor de proximidade, pois, apresentou uma precisão maior do que o GSP. O PrefixSpan só teve problemas para identificar e extrair poucas ações que ocorreram na cozinha (utilizando o sensor de mudança de estado que utiliza valores binários).

Analisando-se os resultados obtidos nos testes conclui-se:

- a) Para o *dataset* utilizado neste trabalho, o melhor método para extração de padrões sequenciais de comportamento em casas inteligentes é o algoritmo PrefixSpan, pois o mesmo apresentou: melhores tempos de execução e consumo de memória além de ter obtido a melhor qualidade de extração (independente do valor do suporte mínimo e do comprimento utilizado).
- b) O algoritmo Spade utiliza propriedades combinatórias para decompor o problema original em subproblemas menores, que podem ser resolvidos de forma independente. Durante os testes, o Spade obteve resultados medianos: apresentou tempos de execução intermediários, consumo de memória estável e qualidade de extração média de 85%.
- c) Por ser baseado no algoritmo *Apriori*, possuir etapas de geração e contagem de candidatos e poda baseada no suporte, o GSP apresentou pior desempenho nos testes: piores tempos de execução e alto consumo de memória. Com relação à qualidade de extração, pode-se dizer que o GSP teve bons resultados, pois o mesmo apresentou qualidade de extração média de 80% no *dataset* utilizado neste trabalho.
- d) A quantidade de dados analisada influencia no resultado dos algoritmos. Quando poucos dados foram analisados, o sistema aprendeu pouco sobre o comportamento do habitante. Com 30 eventos (10% da base), o sistema já apresentou um resultado satisfatório e aprendeu mais sobre o habitante. Com 109 eventos (50% da base), já foi possível extrair bem o comportamento do habitante.

7 CONCLUSÃO

Sistemas de automação residencial devem ter como objetivo principal tornar mais simples a interação entre os habitantes e os recursos da residência gerenciados por ela.

Este trabalho consistiu no desenvolvimento de um sistema de extração de padrões sequenciais de comportamentos. Um sistema deste tipo possibilita que a casa inteligente aprenda de forma autônoma o comportamento de seus habitantes. Após análise dos resultados obtidos, concluiu-se que a casa conseguiu aprender de forma autônoma o comportamento de seu habitante. Levando em conta o que foi aprendido com o algoritmo PrefixSpan, pode-se dizer que houve qualidade para tomada de decisão.

Para os testes do sistema proposto, uma base já existente de dados reais foi utilizada pois, a mesma satisfaz as características de uma residência real.

Foram testados diversos parâmetros de configurações sobre o *dataset* utilizado com o objetivo de simular situações que poderiam ocorrer em situações reais. O sistema se comportou conforme o esperado, extraíndo os padrões sequenciais de comportamento do habitante, apresentando resultados diferentes para cada um dos algoritmos utilizados.

A variação do suporte mínimo e do comprimento, aplicados nos testes, mostrou que estes possibilitam aumentar ou reduzir a quantidade e qualidade das extrações. Foi verificado que os parâmetros ótimos no sistema dependem das características do *dataset* testado, ou seja, para a implantação do sistema em residências reais, seria necessário conhecer os hábitos dos moradores, a fim de se calibrar as variáveis do sistema.

Como principais pontos positivos obtidos neste trabalho, podem ser citados:

- a) Os testes foram realizados em uma base já existente de dados reais (pública) que satisfaz as características de uma residência real.
- b) Foi verificado que a manutenção dos valores de suporte mínimo e comprimento utilizados permite obter uma otimização nos resultados. Neste trabalho, não foi implementada uma política de manutenção automática destes parâmetros, mas há indícios que outros dados presentes no *software* de extração poderiam ser utilizados para ajustar automaticamente parâmetros do sistema, de forma a otimizar seus resultados.

Como pontos negativos, é possível citar:

c) Embora os testes tenham sido realizados com dados reais, o comportamento dos algoritmos foi simulado. Desta forma, particularidades que poderiam ser observadas em testes efetuados em residências reais não puderam ser verificadas no sistema. Ao testar os algoritmos em tempo real, é possível, por exemplo ver como cada algoritmo se comporta se um sensor falhar ou apresentar problemas durante a identificação e extração dos padrões sequenciais.

d) Foi utilizado um *dataset* único, nos testes, com apenas um habitante. Para ser possível uma avaliação melhor do sistema, seria interessante a realização de testes em outros *datasets*, com dois ou mais habitantes, e a execução de novos testes para observar modificações no comportamento das extrações e dos algoritmos.

Com base nos resultados obtidos e nos pontos positivos e negativos observados, na seção a seguir são apresentadas algumas sugestões de trabalhos futuros, com o objetivo de aperfeiçoar o sistema implementado neste trabalho.

7.1 Trabalhos Futuros

Após avaliados os resultados obtidos, novos trabalhos podem ser desenvolvidos, com o intuito de aperfeiçoar o sistema e resolver seus pontos negativos. Algumas sugestões são apresentadas a seguir:

7.1.1 Teste do sistema com outros *datasets*, com características e quantidades de habitantes diferentes

Neste trabalho, foi observado o comportamento dos algoritmos utilizando-se um *dataset* único, contendo as ações de um habitante de uma casa inteligente por um período de catorze dias. Foi verificado, nos testes onde os valores dos parâmetros (suporte mínimo e comprimento) foram modificados, que os resultados das extrações são alterados conforme o número de eventos (ações) ocorridos.

Seria interessante, porém, realizar testes utilizando outros *datasets*, alterando a quantidade de habitantes, as quantidades de ações de cada habitante e seus comportamentos, avaliando agora as alterações nas extrações dos padrões sequenciais conforme o *dataset* utilizado.

Também seria importante verificar comportamentos particulares do sistema, em situações específicas, que não foram geradas pelo *software*, que cria as situações a partir de parâmetros médios informados. Por exemplo, como o sistema se comportará quando houver a presença de visitantes na residência.

7.1.2 Uso de outros parâmetros no software para extração de padrões sequenciais

Conforme foi exposto anteriormente, o software *Smart Homer Pattern Miner* utiliza como parâmetros o suporte mínimo (minSup) e o comprimento para extração dos padrões sequenciais. Um teste interessante que poderia ser feito, é considerar o uso de outros parâmetros, como por exemplo a confiança.

Com o uso destes parâmetros, é provável que as extrações se tornem mais precisas.

Outro dado que poderia auxiliar o *Smart Homer Pattern Miner* na qualidade das extrações seria o uso das informações de um sistema de reconhecimento facial presente na entrada da residência, pois, certas extrações poderiam ser certificadas por este, e outras erradas, eliminadas.

7.1.3 Teste do sistema em uma residência real

Por fim, para confirmar o funcionamento do sistema proposto, na prática, são sugeridos testes em uma residência real.

Isto permitiria verificar tanto o comportamento dos habitantes, quanto dos parâmetros do sistema de extração dos padrões, agora utilizando-se dados reais nos testes.

Seria possível, também, posteriormente, fazer os testes enviando-se os resultados das extrações efetuadas pelo *Smart Homer Pattern Miner* para um sistema de automação implantado.

APÊNDICE A – LOCAIS E QUANTIDADES DE SENSORES INSTALADOS

LOCAL	QUANTIDADE
SALA DE ESTAR	
DVD	1
TV	1
LÂMPADAS	2
GAVETAS	2
ARMÁRIOS	2
INTERRUPTOR	2
PORTA	1
SALA DE JANTAR	
GAVETAS	2
ARMÁRIOS	2
INTERRUPTOR DE LUZ	1
COZINHA	
MICROONDAS	1
TORRADEIRA	1
GELADEIRA	1
FREEZER	1
GAVETAS	3
ARMÁRIOS	2
PIA	1
MÁQUINA DE CAFÉ	1
LIQUIDIFICADOR	1
PORTAS	1
JANELA	2
FOGÃO	1
FORNO	1
MÁQUINA DE LAVAR LOUÇA	1
SECADORA DE LOUÇA	1
INTERRUPTOR DE LUZ	2
BANHEIRO	
DESCARGA	1
SECADOR	1
PIA	1
CHUVEIRO	1
ARMÁRIO	2
GAVETA	2
PORTA	1
INTERRUPTOR DE LUZ	2
QUARTO	
LÂMPADAS	2
TV	1
DVD	1
VIDEO GAME	1
RÁDIO	1
ARMÁRIO	2
GAVETAS	3
JANELA	2
INTERRUPTOR DE LUZ	2
ESCRITÓRIO	
PORTA	1
GAVETAS	3
INTERRUPTOR DE LUZ	2
COMPUTADOR	1
IMPRESSORA	1
SCANNER	1
HALL	
PORTA	1
INTERRUPTOR DE LUZ	1
CAMPAINHA	1
VARANDA	
INTERRUPTOR	2
TOTAL DE SENSORES	77

Fonte: Autor.

**APÊNDICE B – TIPOS, MECANISMOS E FUNCIONALIDADES DOS
SENSORES INSTALADOS NO PLACELAB**

Tipo de sensor	Mecanismo	Funcionalidades
Proximidade / contato	Botão de ímã magnético	Porta aberta/fechada, gaveta aberta/fechada, armário aberto/fechado, janela aberta/fechada, geladeira aberta/fechada, micro-ondas aberto/fechado, torneira aberta/fechada.
Aceleração	Acelerômetro	Porta aberta/fechada, gaveta aberta/fechada, armário aberto/fechado, janela aberta/fechada, geladeira aberta/fechada, micro-ondas aberto/fechado, torneira aberta/fechada, mesa usada/não usada, sofá usado/não usado, cama usada/não usada, controle remoto usado/não usado.
Vibração	Sensor filme piezo	Mesa usada/não usada, cadeira usada/não usada, controle remoto usado/não usado.
Movimento	Sensor piroelétrico de presença	Sofá usado/não usado, cama usada/não usada, habitante se movimentando sim/não.
Temperatura	Sensor de temperatura	Lâmpada ligada/desligada, Fogão elétrico ligado/desligado.
Pressão / peso	Sensores de pressão	Sofá usado/não usado, cama usada/não usada, cadeira usada/não usada.
Luz	Sensores foto resistentes	Lâmpada ligada/ desligada, gaveta aberta/fechada, armário aberto/fechado, monitor do computador ligado/desligado.
Mudança de estado	Sensor de corrente/transformador	Televisão ligada/desligada, rádio ligado/desligado, Dvd ligado/desligado, lâmpada ligada/desligada, máquina de café ligada/desligada, torradeira ligada/desligada, máquina de lavar louça ligada/desligada, máquina de secar louça ligada/desligada, computador ligado/desligado, geladeira ligada/desligada, micro-ondas ligado/desligado, freezer ligado/desligado, congelador ligado/ desligado, impressora ligada/desligada, scanner ligado/desligado, liquidificador ligado/desligado.

Fonte: Autor.

**APÊNDICE C – TIPOS DE SENSORES POSSÍVEIS A SEREM
INCORPORADOS FUTURAMENTE**

Tipo de sensor	Descrição funcional
Interruptor mecânico	Interruptor mecânico convencional, como o botão de pressão. Interruptores feitos de plástico ou metal, que quando pressionados, geram “1s” lógicos ou “0s” caso contrário.
Interruptor piezo	Componentes flexíveis revestidos com uma película de polímeros. A medida que o filme piezo se curva, o sensor aciona uma saída lógica de “1s” e “0s” caso contrário.
Interruptor de vibração piezo	O mesmo sensor de filme piezo descrito para o interruptor piezo, onde uma massa em miniatura é usada para permitir que o interruptor possa detectar a vibração. A medida que o filme piezoelétrico vibra, o sensor aciona uma saída lógica de “1s” e “0s” caso contrário.
Interruptor de temperatura	O interruptor de temperatura consiste em um sensor de temperatura que possui uma saída lógica de “1s” sempre que a temperatura ultrapassar o limite pré-programado pelo usuário e “0s” caso contrário.
Interruptor fotoelétrico	O interruptor fotoelétrico consiste em um dispositivo com sensor fotoelétrico que detecta uma fonte de luz e gera “1s” lógicos sempre que a intensidade da luz é maior do que o limite pré-programado pelo usuário e “0s” caso contrário.
Interruptores de pressão	Consistem em sensores de pressão conectados a circuitos especiais para que gerem “1s” lógicos quando a pressão / peso é maior do que o limite pré-programado pelo usuário e “0s” caso contrário.
Sensores de movimento	Por exemplo o sensor piroelétrico. É feito de um material cristalino que gera uma carga elétrica superficial quando exposto à radiação de calor infravermelho do corpo humano. Quando a quantidade de radiação que atinge o cristal cruza o limiar pré-definido porque alguém está se movendo perto do sensor, ele gera “1s” lógicos, ou “0s” caso contrário.

Fonte: Autor.

APÊNDICE D – DISPONIBILIDADE DE *HARDWARE* E DADOS

Todas as especificações de design de *hardware* e dados coletados para este trabalho estão disponíveis gratuitamente para o público para fins de pesquisa. Entre em contato com Gustavo Molina Figueiredo [gustavo.molina17@yahoo.com.br] para obter informações sobre a aquisição dos dados. O *dataset* inteiro é de aproximadamente 15MB.

**APÊNDICE E – ATIVIDADES REALIZADAS PELO HABITANTE DE 01/09/2007 A
14/09/2007**

<i>Itemsets (ações)</i>	Quantidade de vezes que a ação foi realizada (período de 14 dias)
Trabalhar em casa	4
Sair para trabalhar	6
Comer	50
Usar o banheiro	90
Tomar banho	28
Higiene pessoal	35
Trocar de roupa	17
Lavar as mãos	42
Tomar remédio	0
Dormir	14
Falar no telefone	7
Descansar	4
Preparar o café da manhã	14
Preparar o almoço	10
Preparar a janta	10
Preparar um lanche	14
Preparar uma bebida	4
Lavar louça	48
Guardar a louça	48
Guardar mantimentos	2
Limpar a casa	2
Lavar a roupa	2
Guardar a roupa	2
Levar o lixo para fora	2
Molhar as plantas	4
Cuidar dos animais domésticos	14
Estudar em casa	6
Ir para a faculdade	10
Assistir Tv	12
Ouvir música	8
Sair para lazer	4
Trabalhar no computador	4
Ir para academia	10
Sair para compras	2
Atividade desconhecida	5

Fonte: Autor.

REFERÊNCIAS

AARTS, E.; RUYTER, B. **Ambient intelligence: visualizing the future**. The Netherlands: Eindhoven, 2004.

AGRAWAL, R., SRIKANT, R. Fast algorithms for mining association rules. In: INTERNATIONAL CONFERENCE VERY LARGE DATABASES., 1994, Santiago. **Proceedings...** Santiago: Vldb, 1994. p. 487-499.

AGRAWAL, R.; SRIKANT, R. Mining sequential patterns: generalizations and performance improvements. In: INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY., 1996, London. **Proceedings...** London: Springer-Verlag, 1996. p. 3-17.

ALDRICH, F. Smart Homes: past, present and future. In: HARPER, R. **Inside the smart home**. 2nd ed. London: Springer-Verlag, 2009. p. 17-36.

ALLEN, J. Maintaining knowledge about temporal intervals. **Communications of ACM**, v. 26, issue 11, p. 832-843, 1983.

ANDERSON B. **The IBM vision of a smarter home enabled by cloud technology**. IBM Corporation, Global Electronics Industry, USA, January 2010.

BAUMANN, C. et al. Smart energy case study. In: EMBEDDED SENSING SYSTEMS FOR ENERGY-EFFICIENCY IN BUILDINGS., 2012, New York. **Proceedings...** New York: BuildSys, 2012. p.36-38.

BOLZANI, Caio Augustus Morais. **Análise de arquiteturas e desenvolvimento de uma plataforma para residências inteligentes**. 2010. Tese (Doutorado em Engenharia Elétrica) - Universidade de São Paulo, São Paulo, 2010.

BRUMMITT et al. Easy Living: technologies for intelligent environments. In: HANDHELD AND UBIQUITOUS COMPUTING., 2000, Seattle. **Proceedings...** Seattle: Springer-Verlag, 2000. p. 12-29.

CECÍLIA, H. D. **Minerando padrões sequenciais para bases de dados de lojas virtuais**. 2011. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal de Ouro Preto, Ouro Preto.

CHAN et al. Assessment of activity of elderly people using a home monitoring system. **International Journal of Rehabilitation Research**, 2005, p.69-76.

CHAPMAN, K.; McCARTNEY, K. Smart Homes for people with restrict mobility. **Property Management**, v. 20, issue 2, p. 153-166, 2002.

CHEN, Y.-C., et al. Mining correlation patterns among appliances in smart home environment. In: V. TSENG, T. HO, Z.-H. ZHOU, A.P. CHEN, H.-Y. KAO. **Advances in Knowledge Discovery and Data Mining**, 2nd ed. London: Springer International Publishing, 2014, p. 222-233.

DEMARIA, M. Home smart home. **Network Computing**, issue 13, p. 55-60, 2002.

DOMINGUES, R. G. **A Domótica como tendência na habitação: aplicação em habitações de interesse social com suporte aos idosos e incapacitados**. 2013. Dissertação (Mestrado em Engenharia Urbana) – Universidade Federal do Rio de Janeiro, Rio de Janeiro.

DOUNIS, A; CARAISCOS, C. Advanced control systems engineering for energy and comfort management in a building environment - a review. **Renewable Sustainable Energy Reviews**, 13 (6–7), p. 1246–1261, 2009.

DUKE RESEARCH. **For researchers**. 2016. Disponível em: <<https://research.duke.edu/>>. Acesso em: 18 mai. 2016.

FARELLA, E; FALAVIGNA, M; RICCO, B. Aware and smart environments: The casattenta project. In: 3rd INT. WORKSHOP ADV. SENS. INTERFACES, 2009. **Proceedings...**p. 2-6, 2009.

FUSSEK, R. A. **Desenvolvimento de um classificador automático de pessoas para sistemas de automação residencial inteligente**. 2010. Dissertação (Mestrado em Engenharia Elétrica) – Centro Universitário da FEI, São Bernardo do Campo.

GARLAN, D. et al. Project aura: toward distraction-free pervasive computing. **IEEE Pervasive Computing**, v. 1, n. 2, p. 22–31, 2002.

HARPER, R. Inside the smart home: Ideas, possibilities and methods. In: HARPER, R. (Ed.). **Inside the Smart Home**. [S.l.]: Springer-Verlag, p. 1–13, 2003.

HELAL, S. et al. The gator tech smart house: a programmable pervasive space. **Computer**, p. 50–60, 2005.

HOPPNER, F. Finding informative rules in interval sequences. **Intelligent Data Analysis**, v. 6, n. 3, p. 237-255, 2002.

INTERTEK AND THE DEPARTMENT OF TRADE AND INDUSTRY. **Smart Home – A Definition**. Westminster: Department of Trade and Industry, 2003.

INTILLE et al. Using a live-in laboratory for ubiquitous computing research. **Pervasive Computer**, v. 3968, p. 349-365, 2006.

JUNESTRAND *et al.* Private and public digital domestic spaces. **International Journal of Human-Computer Studies**, p. 753-778, 2001.

KAMAT, S.; BAKAL, W.; NASHIPUDI, M. Comparative study of techniques to discover frequent patterns of web usage mining. **International Journal of Advance Computer Theory and Engineering**, v. 2, n. 3, p. 45-51, 2013.

KEKRE, H. B.; THEPADE, S. UbiComp: the future of computing technology. **Journal of Science**, 2009.

MAMEDE FILHO, J. **Instalações elétricas residenciais**. 8 ed. Rio de Janeiro: LTC 2010.

MIT. **House_n**. 2016. Disponível em: <http://web.mit.edu/cron/group/house_n/intro.html>. Acesso em: 15 de dez. 2016.

MOZER, M. Lessons from an adaptive home. In: COOK, D.; DAS, S. **Smart environments technologies, protocols, and applications**. Wiley InterScience, 2005. p. 273–294.

NORBISRATH, U. et al. Fujaba-based tool development for e-home systems. **Electronic Notes in Theoretical Computer Science**, v. 127, p. 89–99, 2005

NUGENT, C. et al. Editorial home automation as a means of independent living. **Automation Science and Engineering, IEEE**, v. 5, n. 1, p. 1–9, 2008.

PEI, J et al. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: 17th INTERNATIONAL CONFERENCE ON DATA ENGINEERING, 2001, New York. **Proceedings...** New York, p. 215-224, 2001.

PITONI, R. M. **Mineração de regras de associação nos canais de informação do direto**. 2002. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Rio Grande do Sul.

PHILIPS RESEARCH. **Our work**. 2016. Disponível em: <<http://www.philips.com/a-w/research/home.html>>. Acesso em: 15 de mai. 2016.

RASHIDI, P.; Cook, D. Keeping the resident in the loop: adapting the smart home to the user. **IEEE transactions on systems, man, and cybernetics - Part A: Systems and humans**, issue 39, n. 5, p. 949-959, 2009.

RASHIDI, P; COOK, D; HOLDER, L. Discovering activities to recognize and track in a smart Environment. **IEEE Transactions on knowledge and data engineering**, issue 23, n 4, p. 527-539, 2011. Disponível em: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3100559/>. Acesso em: 17 mar. 2016.

REIS, S. T. **Aprendizagem automática de padrões de negociação em sistemas multi – agente**. 2015. Dissertação (Mestrado em Engenharia) – Faculdade de Engenharia Universidade do Porto, Porto.

RUDDARRAJU, R.; HARO, A.; ESSA, I. Fast multiple camera head pose tracking. In: Vision Interface, 2003, Nova Scotia. **Proceedings...** Nova Scotia: AK Peters, 2003.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **Personal Communications**, v. 8, n. 4, p. 10–17, 2001.

SCHWEIZER, D. **Learning frequent and periodic usage patterns in smart homes**. 2014. M. S. thesis - University of Applied Sciences and Arts Northwestern Switzerland School of Business, Olten.

SGARBI, J. A. **Domótica inteligente: automação residencial baseada em comportamento**. 2007. Dissertação (Mestrado em Engenharia Elétrica) – Centro Universitário da FEI, São Bernardo do Campo.

SILVA, D. S. **Desenvolvimento e implementação de um sistema de supervisão e controle residencial**. 2009. Dissertação (Mestrado em Engenharia Elétrica) - UFRN, Rio Grande do Norte, 2009.

SILVA, F; OLIVEIRA, F; PIMENTER, K. **Um sistema para uso otimizado do computador pessoal: automação e outsourcing**. 2008. TCC (Bacharelado em Ciência da Computação) - Universidade Estadual do Ceará, 2008.

TANBEER et al. Discovering periodic-frequent patterns in transactional databases. In: ADVANCES IN KNOWLEDGE DISCOVERY AND DATA MINING. 2009, Bangkok. **Proceedings...** Bangkok, 2009.

TONIDANDEL, F; TAKIUCHI, M; MELO, E. Domótica inteligente: automação baseada em comportamento. In: CBA, 2004, Gramado. **Anais...** Gramado-RS, 2004.

VAINIO, M; VALTONEN, M; VANHALA, J. Proactive fuzzy control and adaptation methods for smart homes. **IEEE intel. syst.**, v. 23, issue 2, p. 42–49, Mar./Apr. 2008.

WU, S and CHEN, Y. Mining non-ambiguous temporal patterns for interval-based events. **IEEE transactions on knowledge and data engineering (TKDE'07)**, v. 19, issue 6, p. 742-758, 2007.

YAMAZAKI, T. Beyond the Smart Home. **IEEE international conference on hybrid information (ICHIT'06)**, v. 2, issue 2, p. 350-355, 2006.

ZAKI, M. An efficient algorithm for mining frequent sequences. **Machine learning**, v. 42, issue 1, p. 31–60, 2001.