



AUTOMAÇÃO RESIDENCIAL USANDO SOFTWARE OPENHAB

Hithallo Nicson Silva Guerra*¹, Josemar Alves dos Santos Junior¹, Ghunter Paulo Viajante¹, Marcos Antônio Arantes de Freitas¹ e Marcelo Escobar de Oliveira¹

IFG – INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS CAMPUS ITUMBIARA

Resumo – Este trabalho tem como objetivo o desenvolvimento de uma interface de automação residencial com OpenHAB. O OpenHAB é um *software* utilizado para disponibilizar uma interface intuitiva aos usuários de automação residencial e entregar a comodidade do controle por voz através de um assistente virtual inteligente. Ele é um servidor instalado em um Raspberry Pi 3 B+ que, através de uma conexão wi-fi, usando o protocolo MQTT, envia os comandos recebidos do assistente virtual inteligente ao tópico do *broker* MQTT. O sistema de automação residencial apresenta uma interface intuitiva para controle de dispositivos por aplicativo de celular ou página web, além de uma interface totalmente dedicada ao controle por telas configuráveis.

Palavras-Chave – Automação Residencial, Domótica, MQTT, OpenHAB, Raspberry Pi.

HOME AUTOMATION USING OPENHAB SOFTWARE

Abstract - This work aims to develop a home automation interface with OpenHAB. OpenHAB is a software used to provide an intuitive interface to home automation users and to provide voice control convenience through an intelligent virtual assistant. It is a server installed on a Raspberry Pi 3 B + that, through a wi-fi connection, using the MQTT protocol, sends the commands received from the intelligent virtual assistant to the topic of the MQTT broker. The home automation system features an intuitive interface for controlling devices by mobile application or web page, as well as an interface fully dedicated to the control of configurable screens.

Keywords – Home Automation, MQTT, OpenHAB, Raspberry Pi.

*hithallo_nicson2010@hotmail.com

I. INTRODUÇÃO

A automação residencial também conhecida como Domótica, veio para preencher a necessidade do homem de otimizar as atividades rotineiras. Praticidade, conforto e otimização de despesas são características que conquistam os usuários em busca de melhor qualidade de vida entregue pela automação residencial. Todavia, para obter as vantagens que a domótica proporciona, no Brasil, o usuário precisa de profissionais que desempenham a função de integrador de sistemas residenciais.

O Brasil tem uma alta demanda para integradores de sistemas residenciais devido à tecnologia usada não possuir o conceito de *plug-and-play* [1]. Em consequência da falta de padronização, o trabalho do integrador de sistemas se torna indispensável, pois o mesmo é responsável por tarefas como: a elaboração e acompanhamento do projeto de acordo com as necessidades do cliente; especificar, fornecer ou indicar os equipamentos e terceiros responsáveis pelo mesmo; além de programar e realizar os testes para consolidar o completo funcionamento do sistema.

A Associação Brasileira de Automação Residencial e Predial (AURESIDE), em 2020, afirmou que o mercado de automação residencial do Brasil pode crescer até 10 vezes mais que a média da economia do país. Para um período de cinco anos, é estimado um crescimento a uma taxa anual composta de 22% ao ano, chegando ao final deste período com um crescimento de 178% [2]. Este cenário poderia ser ainda mais favorável, mas para isso é preciso a desmitificação dos altos custos de implementação de sistemas de domótica no Brasil.

A AURESIDE firmou parceria com a maior feira especializada em persianas, portas, portões e sistemas de proteção solar da América Latina, a R+T South América. Esta decisão faz parte da principal aposta da AURESIDE para aumentar o número de casas automatizadas no Brasil no futuro [3]. O objetivo central desta ação é mudar as impressões dos brasileiros sobre as casas inteligentes a partir da maior difusão

de conhecimento entre usuários e desenvolvedores de sistemas de automação residencial.

Assim, com um alto grau técnico e o baixo custo de implementação, esse trabalho alinha-se com o cenário da domótica no Brasil, além de contribuir para difusão do conhecimento. Para desenvolver o mercado da automação residencial no Brasil, o setor também precisa de profissionais para ampará-lo. Uma forma de alcançar isso, é o fomento de pesquisas que contribuam para formação de profissionais hábeis a partir da superação dos desafios impostos pela implementação do projeto.

Este trabalho consiste em desenvolver uma interface de automação residencial em rede, de baixo custo, usando comandos de voz através de um assistente virtual inteligente com um *software* centralizador gratuito. A seção II. explora os elementos usados no sistema. Já seção III. é dedicada ao desenvolvimento do projeto desde a instalação até a elaboração das interfaces. Na seção IV são apresentados os resultados do projeto. E por fim, na seção V é apresentado a conclusão.

II. ELEMENTOS DO SISTEMA

Esta seção evidencia os estudos de caso, análises e comparações necessárias para escolha do *software*, além das definições de cada elemento que integra o sistema. Apresentando também a função individual de cada componente para mostrar uma visão pontual da automação, facilitando a compreensão da integração de todo sistema.

A. *Software OpenHAB*

O Open Home Automation Bus (OpenHAB), é um *software* de centralização do sistema de automação residencial [4]. Estruturado a partir de um código aberto, o OpenHAB possui alta compatibilidade, ou seja, integra vários dispositivos independente do fabricante.

Criado por uma organização filantrópica chamada OpenHAB Foundation, ele tem o intuito de difundir as capacidades das soluções residenciais gratuitas [4]. A fundação acredita que o mercado de soluções residenciais não se estabelece de forma mais consolidada devido aos problemas enfrentados na comunicação entre dispositivos diferentes. Problemas estes causados por interesses comerciais distintos.

Na automação residencial cada dispositivo é usado para alcançar um objetivo. Contudo, a expansão deste tipo de projeto pode gerar transtornos a respeito das capacidades dos dispositivos, sendo exigidas integrações com outros novos sistemas que não estavam previstos e que não são suportados pelos equipamentos anteriores. Neste ponto o OpenHAB entra com a solução, possibilitando a interconectividade com diferentes tipos de dispositivos e sistemas, dispensando custos adicionais com equipamentos mais atualizados [4].

A função desta plataforma no sistema é a integração, centralização e exibição das variáveis de controle da residência. Estas variáveis são exibidas nas interfaces de usuário e são alteradas de acordo com os comandos recebidos pelo assistente virtual ou por acionamentos manuais nas interfaces do OpenHAB. Depois de atualizado o estado das variáveis, o OpenHAB disponibiliza os mesmos comandos em tópicos definidos no *broker* MQTT.

Outras opções de *softwares* de código aberto foram cogitadas para o desenvolvimento deste trabalho, dentre elas destacaram-se: ThingsBoard e o Home Assistant. Então, para selecionar a plataforma mais adequada para este projeto foram analisados fatores relacionados à configuração, interface e suporte aos assistentes virtuais inteligentes.

Os *softwares* Home Assistant e ThingsBoard, apresentam interfaces altamente convidativas para comunicação entre o usuário e o sistema. Entretanto, o suporte aos assistentes virtuais do Home Assistant é disponibilizado apenas em versões pagas, o que também acontece com o ThingsBoard.

O OpenHAB conta com uma interface um pouco mais simples, mas ainda apresenta características relevantes para interação entre usuário e sistema. Mas em destaque aos outros, o suporte aos assistentes virtuais é totalmente gratuito assim como todos os recursos disponíveis.

Em consequência do exposto acima, o OpenHAB foi escolhido para o desenvolvimento deste trabalho, já que o projeto leva em conta critérios como baixo custo e a utilização dos assistentes virtuais para aumento da comodidade dos usuários.

B. *Raspberry Pi*

Raspberry Pi é um minicomputador de placa única, desenvolvido no Reino Unido pela Raspberry Pi Foundation com o objetivo de colocar a tecnologia da computação ao alcance de todos [5]. O modelo é versátil, pois apresenta um sistema aberto que permite várias aplicações convencionais aos computadores comuns, mas que também tem grande relevância em projetos de prototipagem.

Segundo a documentação do OpenHAB, os modelos recomendados para instalação da plataforma na placa Raspberry Pi são os modelos a partir do Raspberry Pi 2 B [4]. O modelo Raspberry Pi 2 B não tem conectividade Wi-fi nativa, indicando o uso obrigatório do cabo para conexão com a internet. Já o modelo Raspberry Pi 3 B pode ser implementado neste projeto, contudo optou-se pelo uso do modelo Raspberry Pi 3 B+.

As placas Raspberry Pi suportam sistemas operacionais como versões customizadas do Linux e Windows 10 IoT [6]. Entre as versões customizadas do Linux, será usada neste projeto o sistema autoconfigurável fornecido pelo OpenHAB para seu funcionamento no Raspberry Pi, chamado de OpenHABian.

O sistema OpenHABian fornece imagens completas pré-configuradas com OpenHAB, preparativos específicos do *hardware* e do OpenHAB para Raspberry Pi, além das ferramentas para instalar e configurar o OpenHAB baseados em Debian ou Ubuntu [4]. Devido a estas simplificações, com apenas uma etapa é possível configurar todo o sistema operacional e instalar o *software* OpenHAB.

C. *Assistente Virtual Inteligente*

Os assistentes virtuais são interfaces que buscam aprimorar o relacionamento de seres humanos com os computadores [8]. Estes agentes de *softwares* usam inteligência artificial para aprender a responder as instruções de forma mais adequada através de comandos de voz ou texto. Assim os assistentes

virtuais agregam velocidade e precisão ao solucionar questões em várias aplicações.

O mercado destes dispositivos tem dois concorrentes mais conhecidos, o Google Assistente e a Alexa. A escolha do melhor dispositivo depende do perfil do usuário [9] [10]. Cada assistente prioriza interações com outros aplicativos e dispositivos do seu fabricante, portanto, cabe ao usuário analisar seus outros equipamentos para definir qual assistente virtual entregará mais funcionalidades importantes a sua rotina.

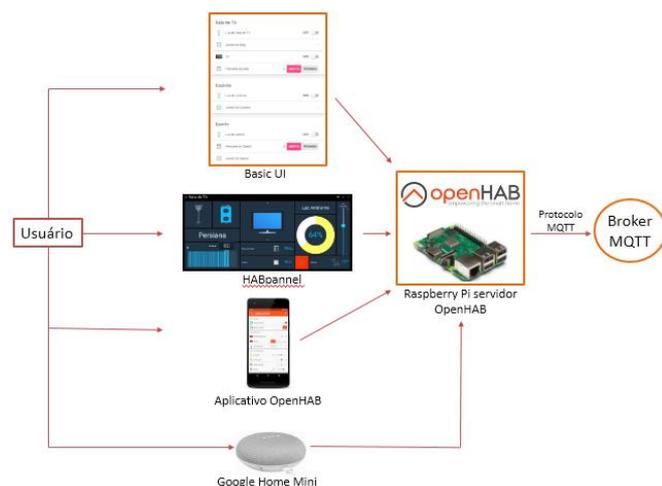
De acordo com [11], os quatro microfones da Echo Dot, tornam melhor a experiência de interagir com o assistente virtual Alexa, afirmando ainda que o *software* de inteligência também prioriza mais as respostas longas, tornando sua comunicação mais humanizada. Ainda assim, [11] defende que o Google Assistente entrega uma experiência mais completa devido o contexto das falas usados nos serviços do Google. Em concordância com [11], [9] afirma que o Google Assistente se sobressai por possuir um banco de dados maior, portanto, consegue realizar buscas mais precisas.

O valor de aquisição do Google Home Mini é em média R\$180,00 enquanto o *smart speaker* mais em conta da Amazon custa em média R\$250,00¹. Assim, o *smart speaker* Google Home Mini que usa o assistente virtual Google Assistente, foi o escolhido para o projeto. A Alexa apresenta respostas mais humanizadas, sugerindo um ambiente mais agradável, entretanto, o fator econômico e provável capacidade de buscas mais precisas foram priorizados para este projeto.

D. Arquitetura do Sistema

A partir das informações supracitadas é possível então visualizar a organização estrutural de toda automação. A Figura 1 apresenta a integração de todo o sistema, assim como os protocolos e *hardwares* utilizados.

Figura 1: Arquitetura do sistema de automação.



O usuário possui as interfaces e o próprio aplicativo do OpenHAB como ferramentas para o controle do sistema. Estas ferramentas podem receber comandos manuais através de

vários dispositivos como *tablets*, *smartphones*, computadores, ou através de comandos de voz proporcionados pelo assistente virtual. Cada comando efetuado pelo usuário é executado pelo OpenHAB, atualizando o estado da variável manipulada em todas as interfaces e por fim disponibilizando o comando ao *broker* MQTT.

III. METODOLOGIA E DESENVOLVIMENTO

Esta seção se dedica ao desenvolvimento do projeto, desde a instalação do sistema operacional OpenHABian, até o funcionamento dos conceitos do OpenHAB.

A. Instalação OpenHABian

Segundo os requisitos do OpenHAB, para instalação do OpenHABian na placa Raspberry Pi é necessário um modelo de Raspberry Pi 2 ou superior, um cartão SD de 16 GB ou maior, uma conexão com *internet* e uma fonte de alimentação estável. Ressaltando ainda que o procedimento não precisa de qualquer teclado ou monitor, o que se deve a capacidade de auto instalação do OpenHABian [4].

O procedimento de instalação pode ser consultado na documentação do *software*, onde estão previstas todas as ações cabíveis em caso de algum possível erro na instalação, além das variantes para configuração do sistema via Wi-fi [4]. Após instalado, será necessário o uso de um cliente *Secure Socket Shell* (SSH) para acesso ao console do OpenHABian que está instalado no Raspberry Pi.

A partir do cliente Putty, é possível configurar uma conexão com o OpenHABian no Raspberry Pi inserindo o *hostname*, a senha e o usuário [4]. Concluídas as etapas anteriores, as opções de configuração do OpenHAB agora estão acessíveis pelo console de configuração.

A documentação sugere a alteração de algumas senhas para evitar que vários usuários usem a mesma, e um sistema de *backup* contra perdas por corrupção do cartão SD. A ferramenta Advanced Maryland Automatic Network Disk Archiver (AMANDA) pode ser instalada no servidor, para garantir o *backup* de todo sistema a partir da definição de algumas etapas disponíveis em sua documentação [12].

B. Configurações OpenHAB

O OpenHAB é um marco na compatibilidade de dispositivos. Esta alta compatibilidade se deve ao seu funcionamento baseado em uma configuração generalista, sem tornar suas aplicações menos robustas. Para configurar um sistema de automação no OpenHAB, é necessário a definição de alguns conceitos usados por ele.

O conceito de Ligação é uma extensão do OpenHAB que integra o meio externo ao *software* de automação, como dispositivos microcontrolados, *smart speaks* e serviços de comunicação e gerenciamento. A partir da instalação das ligações, o OpenHAB tem acesso aos dispositivos que a ligação disponibiliza. Todas as ligações podem ser instaladas a partir da interface “Paper UI” na opção adicionais.

“Things” é um conceito definido como uma entidade que disponibiliza as funções dos dispositivos e serviços ao

¹ Estimativa de preços com base em valores de julho de 2020

OpenHAB. Um exemplo de “thing” usado neste projeto é a do tipo MQTT Broker. Esta é responsável por conectar o OpenHAB a um *broker* MQTT local ou *on-line*. Sua criação e gerenciamento também é feita na interface “Paper UI” em configurações.

Para conectar o OpenHAB a um *broker* MQTT local, é preciso instalar a partir do cliente SSH, um servidor local do *broker* MQTT Mosquitto no OpenHABian. Desta forma, com a ligação MQTT Binding, o OpenHAB tem acesso às funções do *broker*. Já para conectar o *software* a um *broker* on-line, basta fornecer o endereço e as credenciais de acesso nas configurações da thing MQTT Broker.

Os Canais ligam as diferentes funções que uma thing pode ter aos Itens. O tipo do canal é definido pelo tipo de funcionamento do item. Por exemplo, uma lâmpada pode ter comandos de acionamento, intensidade e cor, onde cada canal é dedicado a uma função. Os canais neste projeto diferem quais os tópicos que receberam as informações, além de definir o formato da informação enviada ou lida.

Os canais ligam as things aos Itens que por sua vez são os representantes virtuais dos dispositivos físicos. Cada item possui um *link* que define se a associação entre canal e item foi efetivada. Quando o *link* é estabelecido, os comandos enviados aos itens serão repassados as thing. Da mesma forma caso uma thing altere seu estado de funcionamento, o item receberá o *feedback* devido o *link* entre thing e item.

De acordo com a documentação do OpenHAB, os itens também podem ser criados na interface “Paper UI” ou a partir dos arquivos de textos, nos respectivos diretórios do OpenHAB. Estes arquivos de texto podem ser criados usando o console do cliente SSH, mas para melhorar a comodidade neste processo, o cliente WinSPC é usado neste projeto para ter acesso aos diretórios do OpenHAB no Raspberry Pi.

O WinSPC é um cliente gratuito para Windows usado para transferência de arquivos forma segura [13]. Para ter acesso aos diretórios do OpenHAB pelo cliente WinSPC, basta inserir a senha e o usuário, usados na conexão com o cliente Putty, e inserir como *hostname* o *localhost*. Os diretórios usados na configuração dos conceitos do OpenHAB são mencionados na documentação do *software* [4].

O método usado para criação dos itens nesse projeto foi através dos arquivos de configuração. A documentação do *software* exige que o arquivo de texto, usado para declaração dos itens, deve ter sua extensão do tipo *.items*. Então para representar virtualmente os dispositivos conectados na automação, os itens seguem uma regra de declaração como é mostrado na Figura 2:

Figura 2: Regra para declaração de itens no OpenHAB.

```
itemtype itemname "labeltext [stateformat]" <iconname>
(group1, group2, ...) ["tag1", "tag2", ...] {bindingconfig}
```

O “*itemtype*” é a coluna dedicada a declaração do tipo do item. Cada tipo de item suporta tipos de comandos distintos como ligar e desligar ou definir um valor percentual. Já a coluna “*itemname*” define a identificação daquele item para o

sistema, bem como a coluna “*labeltext [stateformat]*”. Contudo, essa última é usada apenas para nomear o item para os usuários do sistema.

Para auxiliar na identificação do item controlado pelo OpenHAB, o *software* usa a coluna “<*iconname*>” para exibir uma ilustração do item. O ícone declarado nesta coluna será exibido junto ao seu nome, definido em “*labeltext [stateformat]*”, nas interfaces “Basic UI” e “Classic UI”. Para isso, o OpenHAB conta com uma vasta biblioteca para escolha dos ícones em sua documentação.

Na coluna “(group1, group2, ...)” é onde se define os itens que são membros de grupos. Os grupos aglomeram itens para enviar o mesmo comando a todos os itens membros do grupo. Um grupo também pode ser usado para classificar os itens de acordo com a relevância para o usuário como, por exemplo, agrupar os itens de um cômodo em um painel dedicado na interface “Basic UI”.

Para que o assistente virtual consiga interagir com os itens, o OpenHAB usa as marcações na coluna “[“tag1”, “tag2”, ...]” para tal feito. O OpenHAB conta com uma compatibilidade nativa com o Google Assistant, portanto, não é necessário a instalação de uma ligação. Estas marcações são usadas para definir o tipo de dispositivo a ser reconhecido pelo assistente virtual.

Os tipos de marcações usadas neste projeto se resumem a *Swichable* e *Lighting*. A marcação do tipo *Swichable* identificara o item com esta marcação como um interruptor e conseqüentemente o assistente enviara comandos típicos deste tipo de dispositivo. Já a marcação *Lighting* definirá o item como uma lâmpada, permitindo que o assistente virtual envie comandos compatíveis como mudanças no brilho e na cor da lâmpada.

Por fim, a coluna “{bindingconfig}” é usada para a definição das ligações associadas ao item. É a partir deste conceito que o OpenHAB interage com os dispositivos externos, pois, uma vez que um item é associado a uma ligação, o estado do dispositivo é refletido no OpenHAB. Neste projeto, todos os itens estão associados a um tópico do *broker* a partir da ligação MQTT Binding.

C. Configurações de Interface

O OpenHAB disponibiliza algumas interfaces para os usuários e para o desenvolvedor da automação. A interface “Paper UI” é dedicada para o desenvolvimento do sistema, já as interfaces “Basic UI”, “Classic UI” e “HABpanel” são voltadas para o acesso dos usuários aos controles da automação. O acesso às interfaces de usuário pode ser feito por aplicativo ou através do endereço de IP do sistema e os dados de acesso.

A partir da interface “Paper UI” são instaladas as ligações dos dispositivos externos e definidas as things que responderam aos respectivos itens por meio dos canais associados a eles. É a partir desta interface que são definidas as configurações de comunicação e se tem o primeiro contato com os itens já configurados. Nesta interface também se tem acesso ao menu Regras, depois de instalada a ligação Rule Engine.

Este menu é uma interface mais simples para a criação das rotinas de automação, já que não está sujeita a erros de sintaxe.

Cada regra precisa de um gatilho, de uma atuação e, se necessário, também é possível condicionar esta regra a várias condições. As regras executam comandos de forma planejada para automatizar as rotinas do usuário de acordo com suas necessidades.

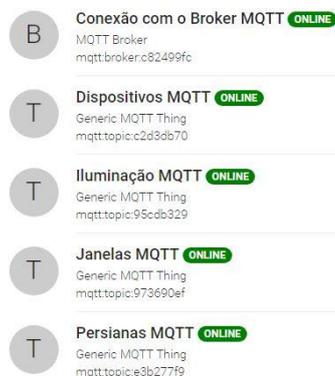
As regras de automação no OpenHAB podem ser feitas usando o console de comandos do OpenHABian ou criando arquivos de configuração. Neste trabalho, as configurações de things e regras foram desenvolvidas usando a interface “Paper UI”, já a criação da interface “Basic UI” e dos itens, foram elaborados os respectivos arquivos de configuração, seguindo a regra de estrutura da documentação do OpenHAB.

O “HABpanel” é uma interface otimizada para uso em tablets e telas touch screen. Com diferentes elementos de interação, esta interface disponibiliza os comandos de automação de forma clara e organizada para os usuários. Sua configuração é simples já que para enviar um comando a um item, basta associá-lo ao botão criado no “HABpanel”. Desta forma o tempo de projeto aqui é gasto apenas na personalização da interface.

IV. RESULTADOS DO PROJETO

Como toda a comunicação do sistema é baseada no protocolo MQTT, todas as things são do tipo Generic MQTT Thing. A Figura 3 mostra algumas things dedicadas ao funcionamento do projeto.

Figura 3: Things criadas para o sistema de automação.



Cada item do sistema está configurado de tal sorte que o arquivo de configuração para interface “Basic UI” não precisa de muitas outras definições. Esta configuração pode ser evidenciada na Figura 4 a seguir:

Figura 4: Arquivo de configuração dos itens do sistema.

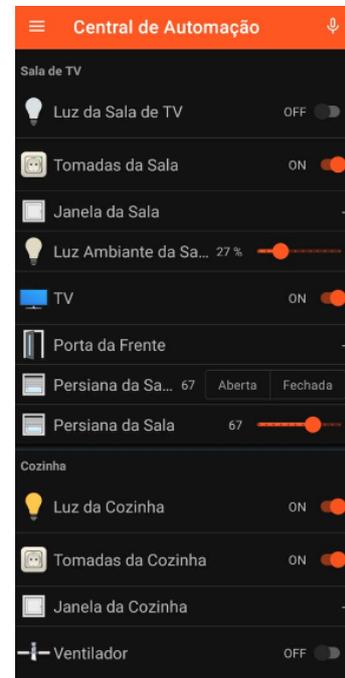
```

1 //Grupo para Rotinas de Automação
2 Group RotinaDormir "Dormir" <moon>
3 Group RotinaEntardecer "Entardecer" <sunset>
4 Group RotinaAmanhecer "Amanhecer" <sunrise>
5 Group RotinaCinema "Cinema" <screen>
6
7 //itens para Atuar Rotinas
8 Switch LDRotinas "Férias" <vacation> ["Switchable"]
9 {channel="mqtt:topic:5c235d1b:LDRotinas"}
10 Switch LDCinema "Protocolo Cinema" (RotinaCinema)
11 ["Switchable"] {channel="mqtt:topic:5c235d1b:RotinaCinema"}
12 Switch LDDormir "Protocolo Dormir" (RotinaDormir)
13 ["Switchable"] {channel="mqtt:topic:5c235d1b:RotinaDormir"}
14

```

Desta forma, o arquivo de configuração para interface “Basic UI” apenas agrupa os itens em quadros que correspondem aos cômodos da casa. O resultado pode ser exibido na interface “Basic UI” ou no aplicativo do OpenHAB como na Figura 5 a seguir:

Figura 5: Interface do aplicativo OpenHAB.



Por fim, a elaboração dos painéis do “HABpanel” finaliza os objetivos alcançados pelo projeto. O painel principal é mostrado na Figura 6.

Figura 6: Painel principal “HABpanel”.



Os painéis desenvolvidos visam a exposição harmônica das variáveis de automação, bem como o uso de controles intuitivos de acordo com o tipo de item acionado. Assim, a Figura 7 mostra o painel Sala de TV como síntese dos painéis.

Figura 7: Sala de TV.



V. CONCLUSÃO

A partir da metodologia empregada e dos estudos de casos executados, foi possível desenvolver uma interface de automação com alta compatibilidade, entregando comodidade para os usuários e uma interação amplamente acessível do ponto de vista econômico e adaptativo. O sistema é capaz de controlar variáveis de automação residencial através de comandos de voz sem custos adicionais com licenças ou outros serviços.

O baixo custo de implementação, a alta compatibilidade e a interação entre usuário e sistema, reforçam na desmistificação das dificuldades de implementação de um projeto de automação residencial. Para trabalhos futuros, pode ser explorado o uso de microcontroladores para manipular as variáveis de automação exibidas neste trabalho. Também existem ferramentas a serem exploradas como HABmin, que gera um histórico de forma gráfica das variáveis controladas, tornando possível um levantamento do consumo da residência para elaboração de um plano de economia de energia, por exemplo.

REFERÊNCIAS

- [1] ROVERI, M. Automação Residencial. 2012, 86 f. Monografia (Tecnólogo em Rede de Computadores) - Faculdade Politécnica de São Paulo, Santa Bárbara d'Oeste, SP. 2012.
- [2] AURESIDE. A Automação Residencial alavanca a demanda por eficiência. São Paulo, Brasil. Disponível em: <<http://www.aureside.org.br/noticias/a-automacao-residencial-alavanca-a-demanda-por-eficiencia>>. Acesso em: 20 mai. 2020.
- [3] SANTOS, T. Aureside firma parceria com principal Feira Internacional de Persianas, Sistemas de Proteção Solar e Automação Residencial da América Latina para mudar impressões de brasileiros sobre casas inteligentes. Disponível em: <<http://www.revistavarejobrasil.com.br/realidades-e-oportunidades-da-automacao-residencial-no-brasil-sao-apresentadas-na-rt-south-america/>>. Acesso em: 20 mai. 2020.
- [4] OPENHAB. Documents. Disponível em: <<https://www.openhab.org/docs/>>. Acesso em: 21 mai. 2020.
- [5] CIRIACO, Douglas. O que é Raspberry Pi? São Paulo, Brasil. Disponível em:

<<https://canaltech.com.br/hardware/o-que-e-raspberry-pi/>>. Acesso em: 22 mai. 2020.

- [6] FLIPFLOP. Raspberry Pi 3 Model B+ Anatel. Santa Catarina, Brasil. Disponível em: <<https://www.filipeflop.com/produto/raspberry-pi-3-model-b/>>. Acesso em: 22 mai. 2020.
- [7] CONTROLENET. O que é um servidor em computação? São Paulo, Brasil. Disponível em: <<https://www.controle.net/faq/o-que-sao-servidores>>. Acesso em: 23 mai. 2020.
- [8] BEHAR, P. A. et al. Um estudo sobre a introdução de um assistente virtual para suporte a escrita coletiva. *CINTED-UFRGS, Porto Alegre*, v. 5, n°1, p. 4. 2007.
- [9] LIRA, V. Alexa ou Google Home: qual a melhor assistente virtual? São Paulo, Brasil. Disponível em: <<https://www.promobit.com.br/blog/alexa-ou-google-home-qual-a-melhor-assistente-virtual/#:~:text=A%20escolha%20por%20optar%20por,melhorando%20a%20converg%C3%Aancia%20do%20aparelhos>>. Acesso em: 19 jun. 2020.
- [10] BELLONI, L. Qual o melhor smart speaker: Amazon Echo Dot ou Google Nest Mini?. Disponível em: <https://www.huffpostbrasil.com/entry/amazon-echo-dot-google-nest-mini_br_5e026245e4b05b08babb4fc2>. Acesso em: 18 jun. 2020.
- [11] GARRET, F. Amazon Echo Dot vs Google Nest Mini: saiba qual smart speaker é melhor. Disponível em: <<https://www.techtudo.com.br/noticias/2019/10/amazon-echo-dot-vs-google-nest-mini-saiba-qual-smart-speaker-e-melhor.ghtml>>. Acesso em: 19 jun. 2020.
- [12] AMANDA. How to backup your openHABian server using Amanda. Disponível em: <<https://onedrive.live.com/?id=CE755578E9202A78%21708&cid=CE755578E9202A78>>. Acesso em: 22 jun. 2020.
- [13] WINSOCP. Table of Contents. Disponível em: <<https://winscp.net/eng/docs/start>>. Acesso em: 22 jun. 2020.