

AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO

Jonas de Jesus Américo¹, Luciano Antunes¹

¹Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)
Av. Universitária, 1105 - CEP: 88806-000 – Criciúma – SC – Brasil

jonas.americo@hotmail.com, luc@unesc.net

Abstract. *Home automation in Brazil has difficulties to be disseminated, difficulties that range from the low evolution of this market to the high cost of implementation. With the high cost of equipment, an aggregation of the population does not have enough purchasing power to acquire a simple system for home automation. This population, especially those with special needs, with only 1% in the labor market. This research aims to apply the concepts of home automation, and to be able to bring comfort and safety at a low acquisition cost.*

Resumo. *A automação residencial Brasil, tem inúmeras dificuldades para ser disseminada, dificuldades essas que vão de baixa evolução desse mercado até ao custo elevado para a implantação. Com o alto custo dos equipamentos, uma parte considerável da população não tem poder aquisitivo suficiente para adquirir um simples sistema para automação residencial. População essa, em especial os com necessidades especiais, sendo que apenas 1% está no mercado de trabalho. Esta pesquisa tem como objetivo, aplicar os conceitos da domótica, e que possam trazer conforto e segurança visando o baixo custo para aquisição.*

1. Introdução

A domótica proporciona inúmeros benefícios aos usuários, dentre eles proporcionar um ambiente mais agradável e acolhedor, melhora a qualidade de vida, e segurança. Com o uso de alguns dispositivos é possível que se evite eventos perigosos. Com esses benefícios fazem da automação residencial uma diferença ou até uma tendência em novos empreendimentos imobiliários [SILVA et al, 2015].

A "Domótica" é uma filosofia aplicada ao projeto de redes para automação de residências, escritórios e comércio em geral. O termo é um neologismo formado pela raiz latina "domus" (casa) e pela palavra "robótica" (Santos Pinheiros, 2005). Com o seu avanço, apresenta o problema de uma perspectiva diferente, essa técnica vem evoluindo com o passar dos anos e trazendo e inovando técnicas aplicadas na resolução de problemas. Com a união da tecnologia da informação, eletrotécnica e eletrônica que faz tornar uma casa inteligente. É uma ferramenta que permite controlar sistemas, dispositivos de automação com o objetivo de aumentar a qualidade de vida e conforto do espaço doméstico. [“What is domotics - Discover | Easydom” [S.d.]]. Devido ao alto custo e complexidade no desenvolvimento de tecnologias proprietárias que promovem a inclusão social, o deficiente tem grandes dificuldades em suas atividades de vida diária e permanece excluído da sociedade e do trabalho. A utilização da automação permite a padronização e a flexibilização de sistemas, fazendo com que seja usada dentro do ambiente doméstico como incremento na qualidade de vida das pessoas [ALIEVI 2008].

De acordo com o último Censo realizado em 2010, aproximadamente 46 milhões, ou 24% da população no Brasil declarou que possui alguma limitação em algumas das habilidades investigadas, entre elas por exemplo, subir degraus ou caminhar, ouvir, enxergar, ou alguma deficiência intelectual/mental. Se considerando apenas as pessoas que possuem total ou grande dificuldade de ouvir, subir degraus ou caminhar, enxergar, é atingido cerca de 12,5 milhões de brasileiros, correspondendo a cerca de 6,7% da população. Segundo [Verdélío 2017] repórter do portal Agência Brasil, comenta que apenas 1% dos brasileiros com deficiência estão no mercado de trabalho. Em pesquisa feita sobre a média salarial de pessoas com necessidades especiais, o Portal Salário 2020, elaborou uma pesquisa com os dados oficiais fornecidos pelo *CAGED – Cadastro Geral de Empregados e Desempregados*, para obter a média salarial de acordo com os tipos de deficiência. Com os tipos de deficiência reportados, o resultado médio entre elas é de R\$: 1.695,05.

Conforme o levantamento bibliográfico, foram encontrados poucos projetos que visavam o baixo custo atualmente, tendo em vista o valor do dólar neste período. Na pesquisa de [CARDOSO 2018], *AUTOMAÇÃO RESIDENCIAL*, ressalta que o cenário da engenharia civil no Brasil, está crescendo gradativamente em ritmo acelerado, e que devido essas necessidades de criação de centros ou até mesmo a expansão dos mesmo de forma mais desenvolvida. Com essa rápida expansão tornou-se necessário a utilização da tecnologia nas necessidades da sociedade.

Levando em consideração a baixa complexidade para desenvolvimento, optou-se em utilizar o microcontrolador ESP32 que oferece por padrão tecnologias como Wi-Fi, Bluetooth. Microcontrolador [Kerschbaumer 2018], são circuitos integrados que possuem todos os componentes necessários para funcionar corretamente, apenas dependendo de uma fonte de alimentação externa, ou seja, pode-se afirmar que são computadores de um único chip. Por sua vez o microcontrolador, possui alguns recursos nativos como comunicação bluetooth e wireless, podendo ser comercializado tanto em placas com alguns componentes, dentre eles: porta micro USB, regulador de tensão, inúmeras portas de entrada e saída, portas essas que podem ser utilizadas por atuadores e sensores [ESPRESSIF, 2021]. Ele está disponível em duas variantes: single-core e dual-core. Mas a variante dual-core é mais popular porque não há diferença significativa de preço [Jain 2020]. A fim de facilitar a prototipação, existe uma gama de sensores e módulos que são compatíveis, por exemplo: sensor de temperatura e umidade, gás, módulos relés, buzzer, lâmpadas de led.

O engenheiro eletricista [MATTEDE, [201-?]], explica que sensores são basicamente dispositivos que têm como função detectar e responder com eficiência a estímulos. Depois que o sensor recebe o estímulo, é emitido um sinal que é capaz de ser convertido e interpretado por outros dispositivos.

O objetivo específico desta pesquisa consiste em: aplicar os conceitos da automação residencial; entender o funcionamento de sensores e do ESP32; integrar o dispositivo móvel com o ESP32; desenvolver os script de acordo com as especificações, para que seja possível o funcionamento correto dos sensores; prototipar uma maquete de uma residência para aplicação do sensores, com o objetivo de empregar o uso da plataforma ESP32 a um projeto de automação residencial de baixo custo a fim de trazer comodidade a pessoas com necessidades especiais.

2. Materiais e Métodos

Esta pesquisa é de forma aplicada, com base explicativa e tecnológica. Desenvolvendo um protótipo de automação residencial com base nos conceitos da domótica, a fim de trazer uma maior comodidade às pessoas que as utilizam. Fazendo o uso de uma maquete eletrônica simulando o acionamento de lâmpadas que representam qualquer equipamento eletrônico. A montagem deste protótipo foi dividida em duas partes, o módulo físico, que é o responsável por simular com os componentes físicos os equipamentos, e o módulo lógico, que é o responsável pelas configurações do software que gerencia todo o grupo físico. A programação dos componentes utilizou-se do ambiente de desenvolvimento da plataforma do arduino. De acordo com [Teixeira 2019], para que seja possível o funcionamento do ESP32 através da do ambiente de desenvolvimento da plataforma do arduino, é necessário seguir algumas etapas de configuração, que seria adicionar uma nova biblioteca fornecida pela fabricante do ESP32, que dá autonomia para reconhecer as novas placas.

Para desenvolvimento desta pesquisa, foi utilizado um computador com processador i7, de oitava geração, sistema operacional Linux Ubuntu versão 20.04 LTS, 16 GB de memória e um SSD de 240 GB. Configuração necessária o suficiente para rodar o simulador de android no próprio computador.

Na tentativa de diminuir custos sem perder a qualidade e conectividade do gerenciamento dos componentes, utilizou-se ESP32 como gerenciador principal. Foi levantado um estudo para analisar a viabilidade e o custo do projeto, então como resultado foram selecionados para aquisição os componentes: Relé Shield, protoboard, lâmpada, chicotes, sensor Mq-2 e Dht11, módulo Buzzer. O módulo relé shield necessita de 5V para conseguir armar seus relés, então foi necessário adquirir uma Fonte Ajustável Para Protoboard 3.3v 5v, que passa fornecer a tensão correta para o acionamento dos relés.

2.1. Desenvolvimento do grupo físico.

No desenvolvimento do grupo físico, foi criado uma maquete que simula o ambiente residencial de uma pessoa, esse ambiente é composto por 6 cômodos conforme figura 2, são eles: dois quartos, banheiro, sala, cozinha e garagem. Em cada cômodo é adicionado uma lâmpada led, que é acionada através do relé shield que recebe um sinal vindo da central controladora. Em um dos quartos, conta com um sensor de monitoramento da temperatura e umidade. Na cozinha foi adicionado um sensor MQ-2 e um módulo buzzer que é ativado e emite som, quando o sensor MQ-2 encontrar irregularidades no monitoramento do ambiente. Esses componentes são todos gerenciados pelo ESP32, que recebe comandos e envia informações por meio do Wi-Fi ao dispositivo móvel. Para desenvolvimento do script gerenciador do módulo físico, utilizou-se do ambiente de desenvolvimento integrado Arduino, em sua versão 1.8.13. No gerenciamento do Wi-Fi, utilizou-se a biblioteca na versão 1.2.7.



Figura 2- Maquete para prototipação

O ESP32 segundo [Boral 2020], é uma placa de microcontrolador de baixo custo, com Wi-Fi e Bluetooth integrados, baseado em um mecanismo de processador dual-core. O primeiro processador é um Xtensa LX6 (~240 MHz) com 512 KiB de memória, e o segundo é um processador ultra baixo consumo, que é projetado para funcionar quando estiver em modo de suspensão. BORAL (2020) ainda complementa que o ESP32 se sobressai sobre outras placas, devido a ter um suporte integrado a Wi-Fi e Bluetooth. A velocidade do wi-fi, pode chegar a 150 Mbps, o que é perfeito em um projeto de IoT.

Para comunicação entre o módulo lógico e módulo físico, utilizou-se da capacidade do próprio microcontrolador em fornecer um Web Server capaz de responder com eficiência às requisições solicitadas pelo aplicativo. Web Server recebe a requisição, processa e executa o comando e retorna a situação.

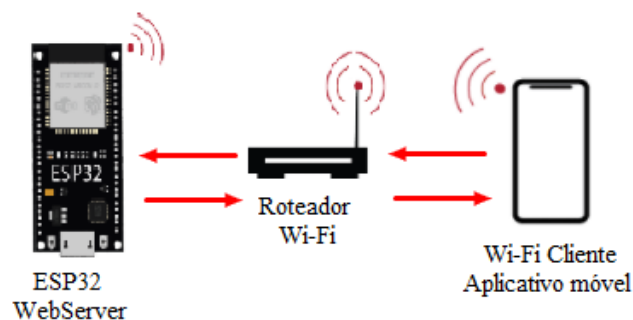


Figura 3 - Exemplo conectividade ESP32 como Web Server

O portal do NGINX, explica que um Web Server é capaz de armazenar e entregar o conteúdo de um site, e pode hospedar um único site ou até mesmo vários usando os mesmos recursos de software e hardware. Para que o ESP32 seja capaz de fornecer um Web Server, é necessário utilizar-se de duas bibliotecas específicas, “Wifi.h” e “WebServer.h”. Wifi.h fica responsável por fazer a configuração da conexão com a rede Wi-Fi disponível no ambiente, e o WebServer.h faz a parte em preparar os caminhos responsáveis pela execução dos comandos enviados pelo aplicativo.

2.1.1. Configuração conexão com a rede

Para configurar a biblioteca WiFi.h, é necessário primeiro incluir o mesmo no início do arquivo, basta adicionar `#include <WiFi.h>`, isso dará autonomia para que o projeto passe a utilizar todas as funcionalidades da biblioteca. Para ser possível a conexão com a rede Wi-Fi disponível, é necessário que se crie duas constantes do tipo char para armazenar o nome e senha da rede disponível, conforme figura 4.

```
const char *ssid = "NOME_DA_REDE";  
const char *password = "SENHA_DA_REDE";
```

4 - Exemplo de constante com nome e senha da rede disponível

Após criado as duas constantes é necessário configurar o acesso a rede, para isso, na função de setup do arquivo, é necessário declarar qual será o modo de utilização do Wi-Fi, existem três possibilidades de configuração, conforme o portal especializado em prototipagem com ESP32 Random Nerd Tutorials, explica os tipos, que são: *WIFI_STA*, *WIFI_AP*, *WIFI_STA_AP*. O primeiro faz com seja possível se conectar a um ponto de acesso, por exemplo Wi-Fi residencial, o segundo, permite que outros aparelhos se conectem à rede que o ESP32 tem a capacidade de prover, e o terceiro, permite se conectar a uma estação já conectada a outra rede. A fim de reprodução deste artigo, será utilizado a primeira opção. Conforme figura 5, após informado o modo de operação do Wi-Fi, é necessário iniciar a conexão com o Wi-Fi através da função *begin* disponível na biblioteca.

```
WiFi.mode(WIFI_STA);  
WiFi.begin(ssid, password);  
Serial.println("");  
while (WiFi.status() != WL_CONNECTED)  
{  
    delay(500);  
    Serial.print(".");  
}  
  
Serial.println("");  
Serial.print("Nome da rede conectado: ");  
Serial.println(ssid);  
Serial.print("IP da rede : ");  
Serial.println(WiFi.localIP());
```

Figura 5 - Exemplo de configuração da conexão via Wi-Fi ESP32

Após configurado entrará em loop e testará a conexão, se for bem sucedida será exibido no monitor serial da IDE, o nome da rede o IP em que o ESP32 está respondendo. Uma funcionalidade útil para evitar problemas com o endereço de IP que é atribuído ao ESP32, a biblioteca do próprio Wi-Fi, disponibiliza a opção de configurar um IP fixo na rede. Configurado ip fixo conforme figura 6, as requisições passam a ser feita para o endereço url: <http://192.168.0.113/>

```
IPAddress ip(192,168,0,113);  
IPAddress gateway(192,168,0,1);  
IPAddress subnet(255,255,255,0);  
  
WiFi.config(ip, gateway, subnet);
```

Figura 6 - Exemplo de configuração IP fixo na rede

Para que o Web Server entre em ação depois importando, é necessário que chame o construtor e informe qual a porta de operação que o Web Server ficará responsável por responder, conforme figura 7, o Web Server está informando que responderá na porta 80, e responderá no IP fixo selecionado acrescentando a porta, por exemplo figura 8.

```
WebServer server(80);
```

Figura 7 - Exemplo da declaração do Web Server

Configurado todo o processo de conexão com a rede, foi necessário criar rotas que ficam responsáveis por prover e receber informações necessárias para conseguir manipular todos os componentes disponíveis. Para se configurar uma rota, através do server declarado, é necessário declarar o método on, especificando o nome da rota e uma função que será executada quando for solicitado pelo aplicativo.

```
server.on("/", handleRoot);

void handleRoot ()
{
    server.send(200, "text/html", "Exemplo teste");
}
```

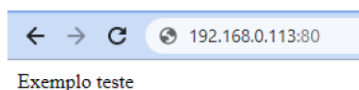


Figura 8 - Configuração da rota e resposta

Feito o processo de configuração do endereço responsável por prover e receber as informações, o ESP32 está apto configurar os componentes e sensores

2.1.2. Componentes e sensores

Sensores são componentes eletrônicos de acordo com [Costa 2020], que transforma informações físicas que estão em um ambiente, por exemplo, a temperatura/umidade de uma sala, quantidade de luz em um ambiente, em informações elétricas. Com isso, é possível tomar algumas decisões como ligar o ar condicionado quando a temperatura chegar em um determinado grau Celsius, ligar ou desligar luzes de acordo com a quantidade de luz incidente. Para atender o foco principal proposto nesta pesquisa, utilizou-se dos sensores de temperatura/umidade DHT11, gás MQ2, módulo Buzzer, e lâmpadas de led. Utilizou-se lâmpada de led para exemplificar a possibilidade de conseguir ligar e desligar qualquer equipamento eletrônico desejado em uma residência.

2.1.2.1. Sensor DHT11

O sensor DHT de acordo com o [Srivastava et al. 2018], é um módulo de controle de umidade e temperatura, que possui uma saída digital calibrada. Ele é capaz de prover um valor muito preciso no monitoramento, que garante uma alta confiabilidade e estabilidade a longo prazo, o tempo de processamento do sensor é de 4 ms, possui uma capacidade de transmissão de sinal de até 20 metros. O DataSheet [D-Robotics [S.d.]] do módulo explica que o sensor aceita tensão entre 3 a 5 VDC, sendo que o máximo aceito é 5,5 VDC. E sua capacidade de medição da umidade é de 20% a 90%, com precisão de

5% para mais ou para menos, e para temperaturas oferece a possibilidade de 0° a 50°C e com variação de 2°C para mais ou para menos.

2.1.2.2. Configuração DHT11

Para configurar o módulo DHT11 de maneira que ele funcione corretamente, é necessário selecionar um pino de entrada de dados da controladora ESP32. Nesta pesquisa, foi selecionado o pino 4, conforme exemplificado na figura 9. Para que o ESP32 passe a monitorar e receber as informações do módulo, é necessário configurar via script, e utilizou-se da biblioteca *DHT.h* em sua versão 1.4.2, ela é fornecida pela Adafruit. A biblioteca fornece um construtor que recebe como parâmetro qual o pino e qual o tipo de módulo do sensor, conforme mencionado anteriormente é o pino 4 e o DHT11. Informado o pino e qual o sensor, foi necessário inicializar o mesmo no setup do script, com o comando `"dht.begin()"`, depois de inicializado a biblioteca retorna alguns métodos, por exemplo: `"readTemperature"`, que é responsável por fornecer a informação da temperatura, e `"readHumidity"`, que fornece a informação da umidade no ambiente. Para disponibilizar essas informações para o cliente que fará o uso dele, criou-se uma rota exclusiva que responde pelo caminho: `/sensor/temperatura`. A rota retorna como resposta um json que contém os dados da umidade e temperatura.

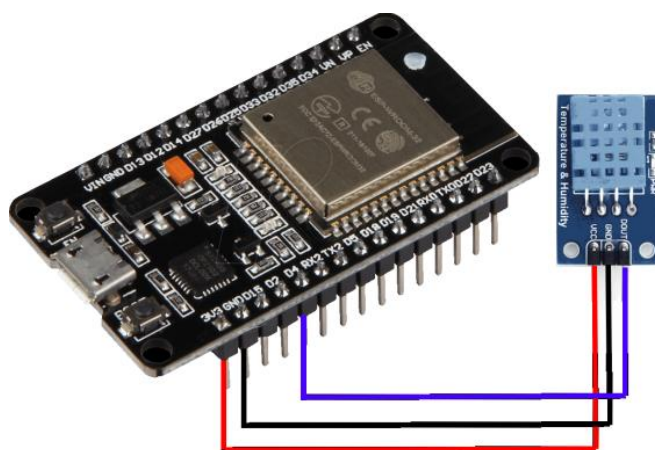


Figura 9 - Ligação módulo DHT11 no ESP32

2.1.2.3. Sensor MQ2 e módulo Buzzer passivo

O sensor MQ2 de acordo com [KOELLN 2017], é um dispositivo de segurança que tem como finalidade medir os gases. Os fabricantes recomendam para utilização em projetos de automação residencial. Possui a capacidade de detectar vários tipos de gases como por exemplo: propano, butano, petróleo liquefeito e gás natural. O datasheet ("MQ-2 Semiconductor Sensor for Combustible Gas"), explica que o sensor trabalha com uma tensão de 5.0V podendo variar 0.2V para mais ou para menos.

O módulo buzzer passivo tem funcionamento semelhante a um pequeno falante, utilizado para emitir sinais de alerta ou de informação. Funciona com uma tensão de 5V. Este módulo é utilizado em conjunto com o sensor MQ2, quando o sensor MQ2 encontra

alguma irregularidade no monitoramento, é disparado um som pelo módulo buzzer para alertar. Conforme figura 10, exemplifica o esquema de ligação entre os componentes.

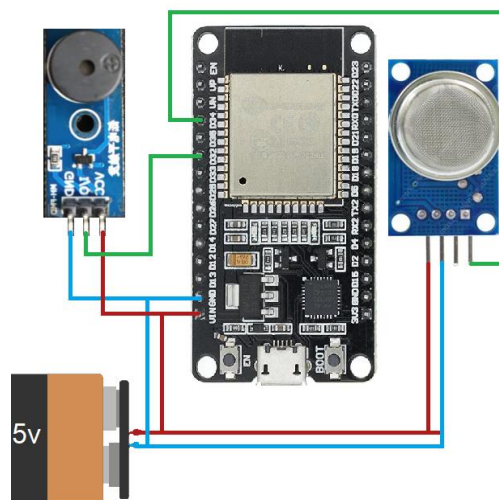


Figura 10 - Ligação do sensor MQ2 e o módulo Buzzer

2.1.2.3.1. Configuração MQ2

Para configurar o sensor MQ2, é necessário reservar um pino da placa controladora. Nesta pesquisa foi reservado o pino 34 conforme figura 10, para que o sensor possa se comunicar com a controladora. Para que o sensor MQ2 funcione de forma mais correta, é necessário que o mesmo seja calibrado, segundo [Leite 2016], existem quatro passos a serem seguidos para calibrar o sensor são eles:

1. Condicionar o sensor em local com ar limpo para definição do controle. Calibrar e descobrir o valor de resistência neste ambiente (R_0);
2. Condicionar o sensor em local que será afetado pelos gases, para caso encontre irregularidade possa ser tomado alguma ação;
3. Identificar no datasheet os dois ponto da curva do gás, para calcular a concentração de gás em PPM;
4. Calcular a concentração de gás em relação ao declive dessa linha.

No primeiro passo, é encontrado o R_0 . Com base no datasheet do MQ2, o valor de R_S em ar limpo sob umidade e temperatura é constante. Em condições de ar limpo, a proporção de R_S/R_0 é de aproximadamente 9,8. No segundo passo, depois que encontrado o R_0 , pode-se encontrar a razão R_S/R_0 . No terceiro passo, é calculada a concentração de PPM (partícula por milhão). Por fim, no passo quatro, é encontrada a concentração de cada gás específico.

2.1.2.4. Módulo relé e lâmpadas de led

Visando o baixo custo neste projeto, foi adquirido um módulo 8 Relés Opto Acoplados 5v, para simular ligar e desligar equipamentos eletrônicos. O engenheiro da computação [Oliveira 2018], comenta que os relés são capazes de controlar grandes circuitos com grandes correntes a partir de pequenas tensões ou correntes. Para exemplificar a ligação de equipamentos eletrônicos, foi utilizado como exemplo lâmpadas de led, que possuem funcionamento padrão na tensão de 3V a 3,2V.

2.1.2.4 Alimentação do sistema

Para alimentação do sistema num todo, foi utilizado de uma Fonte Ajustável Para Protoboard 3.3v e 5v. Ela é capaz de se acoplar a uma protoboard e fornece duas zonas com tensões distintas, uma operando a 3.3v, e outra operando em 5v, fornecendo corrente máxima de 700 mA. A tensão de 3.3v, foi utilizada para alimentação dos leds, na simulação dos equipamentos eletrônicos, e a de 5v para ligar os demais sensores. Para alimentação dessa fonte ajustável, foi utilizado um carregador bivolt que fornece 9v e 1A de corrente.

2.1.2. Custo de aquisição dos componentes

Na aquisição dos componentes utilizados nessa prototipação, foi utilizado o critério principal desta pesquisa, o baixo custo. Para encontrar os componentes com o maior baixo custo possível, recorreu-se à Internet para seleção dos mesmo. O custo total da prototipação foi de R\$ 212,21 conforme descrito na tabela 1.

Tabela 1 - Custo de aquisição dos componentes

Descrição	Quantidade	Valor (R\$)
Led Branco Frio Alto Brilho 3 mm	6	R\$ 1,80
Modulo Passive Buzzer 5V	1	R\$ 12,12
Protoboard 400 Pontos	1	R\$ 14,80
Fonte Ajustável Para Protoboard 3.3v 5v	1	R\$ 14,90
Cabo Wire Jumper 20cm 50 Fios	1	R\$ 15,90
Sensor de umidade e temperatura	1	R\$ 22,90
Sensor de Gás e Fumaça	1	R\$ 24,90
Esp32 Modulo Wi-Fi	1	R\$ 54,99
Modulo 8 Reles Relay Optoacoplados 5v	1	R\$ 49,90
Total		R\$ 212,21

O custo da fonte de alimentação, não entra na contagem, pois o protótipo é alimentado por uma fonte proveniente de outros equipamentos que não estão mais sendo utilizados, fonte essa que é capaz de fornecer 9V a 1 ampere. Em sua grande maioria, os itens que compõem este projeto, são importados, ou seja, seguindo a cotação do dólar, poderá sofrer diferença de valores para cada vez que for refazer a cotação.

2.2. Grupo lógico

Para o desenvolvimento do grupo lógico, foram aplicados dois princípios para a escolha da tecnologia que seria aplicada para o desenvolvimento. O primeiro princípio, é na facilidade de reaproveitamento dos códigos escritos, e a facilidade no desenvolvimento. Seguindo esses critérios, foi escolhido para o desenvolvimento a tecnologia *React Native*, pois ela funciona de forma híbrida, ou seja, com o mesmo código fonte escrito para compilar e executar em um celular Android, pode ser compilado e rodado para executar em celular com IOS. Segundo [Brito et al. 2018], por meio de um

estudo comparando tecnologias híbridas com nativas, foi aplicado critérios para analisar qual das tecnologias seria mais vantajoso, os critérios aplicados foram: aprendizagem, custo no desenvolvimento, emuladores e sua depuração, tempo de resposta, reconhecimento comercial, reutilização de código e por fim manutenção e atualização. Concluíram que a React Native apresenta vantagens sobre as outras tecnologias estudadas, como por exemplo, a equipe pode aplicar soluções Web baseadas em React apenas modificando os componentes de interfaces móveis para web.

Para desenvolvimento do aplicativo, foi utilizado a versão 0.64 do React Native. Como ambiente de desenvolvimento foi utilizado o Visual Studio Code com a versão 1.56.2. Para fins de teste e depuração do aplicativo, utilizado como depurador, um celular Motorola de modelo Moto G 8 Plus com a versão do Android 10.

O aplicativo desenvolvido para este protótipo é composto por 3 telas conforme figura 11, a tela de abertura, a tela inicial e uma tela para controle dos relays. Na tela de abertura, é apenas exibido um iniciar simples do aplicativo, na tela inicial, será exibido 3 funcionalidades, primeira é exibido a umidade e temperatura, segunda a opção de controlar os relés, que está sendo simbolizado por luzes e a terceira é a opção de comando de voz. Na funcionalidade do comando de voz, dando um comando por exemplo: “Ligar luz do ambiente um”, será disparado um comando para ligar a luz do ambiente configurado.

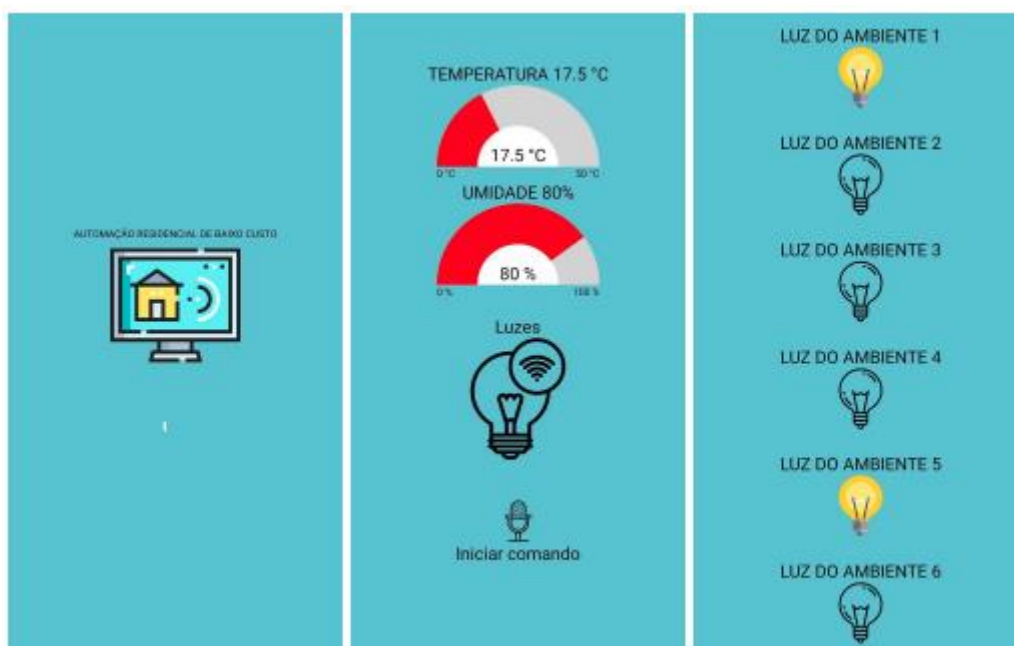


Figura 11 - Exemplo das telas do App criado para gerenciamento do protótipo

Os dispositivos androids contam com dois recursos que auxiliam pessoas com deficiência visual total, ou para quem tem baixa visão em ter uma maior acessibilidade. De acordo com [Park et al. 2019], a acessibilidade significa fornecer acesso igual a instalações, produtos, serviços e informações, independentemente da capacidade da pessoa. Para pessoas com deficiência total na visão, o smartphone conta com o recurso “TalkBack”, que auxilia na identificação de menus na tela, e para pessoas com baixa visão, possui o recurso “Selecionar para ouvir”. O React native, oferece o suporte a acessibilidade em seus layouts. Para este recurso ser ativado, é necessário configurar o

próprio smartphone em que será instalado o aplicativo. Em dispositivo android, basta ir em configurações, acessibilidade, entre na opção “Selecionar para ouvir” ou “TalkBack”, e ativar o recurso.

3. Resultados e Discussões

Os equipamentos foram adquiridos individualmente a fim de baixar o custo em comparação a modelos prontos. Com o objetivo demonstrar o funcionamento da aplicação foi utilizado uma maquete, que visa simular um ambiente real. Verificando o custo de aquisição dos componentes e comparando com o padrão pronto de automação hoje no mercado, fica comprovado que existe a possibilidade de aplicar um menor custo num projeto básico de automação residencial. Mas devo salientar que por esses equipamentos serem de baixo custo, não é apresentado quaisquer tipos de estudo sobre a confiabilidade e a probabilidade de falha em seu funcionamento se comparar com os módulos de automação residencial mais robusto que se encontram no mercado disponíveis.

Os projetos comerciais a venda semelhantes ao proposto nesta pesquisa, o mais próximo encontrado, tem como funcionalidade apenas o controle de lâmpadas, que tem um custo de R\$:482,00 contra R\$:224,16 desta pesquisa, que além da funcionalidade oferecida pelo mercado, é oferecido monitoramento de temperatura e do ar.

Observado também que em sua grande maioria dos projetos disponíveis para vendas, tem como foco a automação apenas em pontos de tomada, e outros que além das tomadas, oferece apenas o suporte para incrementar outros sensores, que elevará ainda mais o custo para aquisição.

Em uma aplicação real de automação residencial empregando o uso do ESP32, seria necessário que todos os dispositivos do sistema fossem acoplados em uma placa de circuito impresso, mas traria a desvantagem de não poder fazer futuras ampliações. Para aplicação de teste da efetividade dos componentes na automação, cada etapa de montagem foi testada de forma independente, desde a conexão com o Wi-Fi, a testagem de um led. Incluindo também a lógica de funcionamento de cada componente composto na automação. Durante a realização dos testes ficou evidente que os componentes precisam trabalhar de forma unificada, desta forma buscou-se a integração. Como resultado o ESP32 mostrou ser bastante versátil, e responde bem a lógica de cada componente aplicado.

Em teste com o sensor DHT11, responsável pelo monitoramento de temperatura e umidade, ele apresentou bom resultado em suas medições, o sensor possui uma precisão de +/- 2°C e de 5% na umidade segundo o seu próprio datasheet. Comparando as medições do DHT11 com o sensor de temperatura que é aplicado ao Ar condicionado Lg dual inverter, o sensor DHT11 apresentou a diferença de +/- 1°C no mesmo ambiente sob as mesmas condições.

O sensor MQ 2, que monitora a qualidade do ar e verifica a presença de, por exemplo, gás de cozinha no ambiente, juntamente com o módulo buzzer, utilizado para simular um sinal de emergência em caso de identificação de gás no ambiente. Com o componente calibrado em ambiente limpo livre de qualquer resquício de poluição, ao detectar minimamente a presença de gás GLP, o sensor entra em ação e a ESP32 processa a informação e dispara a sirene de alerta. O sensor por ser baixo custo se comparar com módulos mais sofisticados que existem no mercado, atendeu a demanda com eficiência na identificação de gás no ambiente.

Na simulação de ligar e desligar eletrônicos e lâmpadas presentes em uma residência, foi utilizado lâmpadas de led, para a exemplificação. No aplicativo na opção de luzes, é descrito as luzes de cada ambiente, e quando clicado em uma opção, o aplicativo manda uma requisição para ligar determinado ambiente, essa requisição é recebida pelos ESP32 e dado um pulso para que o relé arme e libere a corrente para ligar a lâmpada, ou qualquer equipamento que estiver ligado na saída do relé. Esse processo entre comunicação do aplicativo e o ESP32, mostrou-se bastante eficiente com pouca latência, utilizando a ferramenta de teste de API's, Insomnia versão 2021.3.0, é possível identificar através dos mesmo quantos milissegundos leva para se obter a resposta do comando enviando, e como resultados foi obtido valores entre 100 a 250 milissegundos a cada iteração. Teste esse realizado em uma rede local, contando com um roteador TP-Link modelo *ARCHER C5W* no início da residência e outro roteador Multilaser modelo Re047 atuando como repetidor no final da residência para amplificar o sinal.

O consumo total de energia seguindo os datasheet dos componentes é de 557,5 mA, conforme tabela 2.

Tabela 2 - Valores em mA dos componentes

Descrição	mA
Esp32 Modulo Wi-Fi	240
6 lâmpadas led	120
Modulo Passive Buzzer 5V	25
Módulo 8 relé Optoacoplador 5v	20
Sensor de Gás e Fumaça	150
Sensor de umidade e temperatura	2,5
Total	557,5

Tendo como base o consumo total dos equipamentos somando mais a resistência causada pelos fios que faz a ligação entre os componentes, a fonte ajustável para protoboard que trouxe a facilidade na alimentação, não foi capaz de suprir toda a demanda necessária, tendo em vista que ela entrega no máximo 700 mA para ser utilizado. Devido a esse déficit, quando todos os componentes estavam operando em seu máximo, a insuficiência causava instabilidade na tensão de todo o sistema, e vinha ocasionar falhar, ou até mesmo travar o ESP32. A solução encontrada foi desmembrar o módulo físico e duas partes, a primeira, o ESP32 funcionando com uma fonte de alimentação advinda de um carregador de celular sem uso, que atende às especificações de alimentação dele. E a segunda parte, o restante dos componentes, ligado a fonte ajustável, alimentada por um carregador específico, com saída de 9V a 1 A.

No quesito acessibilidade a aplicação num todo, os recursos de acessibilidade oferecidos pelo React Native, mostrou-se bem funcional, ele era capaz de indicar de forma legível a descrição dos label e botões disponíveis, facilitando o manuseio do aplicativo.

Na pesquisa de [CARDOSO 2018], *AUTOMAÇÃO RESIDENCIAL*, procurou evidenciar o baixo custo nos componentes utilizados. Automação essa que conseguia controlar 8 pontos da casa, ligar e desligar, e o monitoramento do ar, temperatura e

umidade. Como base para controle dos componentes, utilizou-se da plataforma arduino, tendo em vista que ele possui um baixo custo na aquisição. O custo para o desenvolvimento do protótipo foi de R\$ 163,30. O autor conclui que, o trabalho proposto correspondeu com a evidência do baixo custo, se comparado com outros sistemas semelhantes no mercado.

[Lofiego e Neves 2020], da faculdade de Tecnologia de Botucatu, propuseram um projeto de “*Automação Residencial: Acionamento remoto via Bluetooth, de uma forma simples e de baixo custo*”, onde o intuito é o baixo custo e uma aplicação simples para pessoas leiga terem uma maior facilidade no uso. Para prototipação o autor utilizou-se das tecnologias: Arduino Uno, módulo Shield Bluetooth, Shield Relés, Protoboard, chicotes, lâmpada e um Smartphone com sistema operacional Android. Com os componentes foi possível a automatização do ambiente. O autor afirma que o Arduino é uma boa alternativa no desenvolvimento de soluções residenciais. O custo com equipamentos foi de R\$: 325,00.

4. Conclusão

A partir dos conceitos aplicados no desenvolvimento deste protótipo de automação residencial, constatou-se que é possível aplicar um custo reduzido em um sistema que automatize a rotina simples do dia a dia de uma pessoa em sua residência. Apesar de não existir estudo que afirme a eficiência dos sensores aplicados neste projeto, eles mostraram resultado satisfatório pelo custo oferecido. Os componentes utilizados possuem bibliotecas que controlam e aproveitam o máximo da capacidade oferecida pelo hardware, facilitando que seja seguido as especificações necessárias para que seja aplicado o uso correto dele. O Datasheet do sensor MQ2, especifica os valores padrão para que fosse possível a calibração do mesmo em ambientes limpos, para que fosse possível a identificação mais exata do gás.

A integração entre a plataforma do ESP32 com o aplicativo por meio da tecnologia Wi-Fi, mostrou-se eficiente e sem delay em suas operações, o microcontrolador aplicado nos exemplos, possuem 25 pinos para entrada de informações, o que permite expandir mais o seu leque de sensor, o que reforça que o custo reduzido de uma central controladora não caracteriza baixo rendimento em sua operação.

No desenvolvimento deste protótipo, aplicando os conceitos principais desta pesquisa, obteve-se como custo total de R\$ 212,21, caracterizando ser possível a aplicação de baixo custo em relação aos valores praticados no mercado. Se comparando com a pesquisa de [Lofiego e Neves 2020], obteve-se uma redução de 65,3% no valor final e em relação a pesquisa de [CARDOSO 2018], houve um acréscimo de 29,95%, o que é justificado pela alta do dólar.

Como base nos conhecimentos adquiridos com a elaboração deste trabalho, bem como nos resultados obtidos propõe-se como trabalho futuros: a aplicação de relés independente com suporte ao *Módulo Wi-Fi ESP8266 Serial ESP-01S*, e com o ESP32 configurado como *WIFI_STA_AP*, que possibilitará ter mais flexibilidade na instalação dos componentes evitando excesso de fios; possibilitar o acesso a controle externo por meio da configuração do roteador, ou serviço MQTT; Possibilitar a integração com o Google Assistente ou Alexa da Amazon, para que o reconhecimento de comando de voz seja mais assertivo.

Referências

ALIEVI, C. A. (dez 2008). Automação residencial com a utilização de Controlador Lógico Programável.

Boral, S. (25 mar 2020). What is ESP32 and Why Is It Best for IoT Projects? - IoT Tech Trends. <https://www.iottechrends.com/what-is-esp32/>, [accessed on Jun. 1].

Brito, H., Gomes, A., Santos, A. e Bernardino, J. (27 jun 2018). JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development. In *Iberian Conference on Information Systems and Technologies, CISTI*. . IEEE Computer Society.

CARDOSO, R. D. M. (dez 2018). AUTOMAÇÃO RESIDENCIAL. FUNDAÇÃO CARMELITANA MÁRIO PALMÉRIO.

Cargos e Salários PCD 2020 - Tabela Salarial Pessoas Com Deficiência (2020). <https://www.salario.com.br/tabela-salarial/salarios-pcd-pessoas-com-deficiencia/>, [accessed on Jun. 1].

Costa, F. (16 set 2020). Arduino: como começar? Os componentes que você precisa conhecer. <https://fabiocosta.net/arduino/como-comecar-com-arduino/>, [accessed on Jun. 1].

D-Robotics ([S.d.]). DHT11 Humidity & Temperature Sensor.

ESP32 Useful Wi-Fi Library Functions (Arduino IDE) (2021). <https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/>, [accessed on Jun. 1].

ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems (2021). <https://www.espressif.com/en/products/socs/esp32>, [accessed on Jun. 1].

IBGE sai em defesa do orçamento do Censo 2021 | Agência de Notícias | IBGE (28 abr 2021). <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/30350-ibge-sai-em-defesa-do-orcamento-do-censo-2021>, [accessed on Maio 11].

Jain, R. (16 jan 2020). ESP32 Dual Core Programming with Arduino IDE. <https://circuitdigest.com/microcontroller-projects/esp32-dual-core-programming-using-arduino-ide>, [accessed on Jun. 1].

Kerschbaumer, R. (2018). Microcontroladores.

KOELLN, M. F. B. (2017). DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE DIÓXIDO DE CARBONO EM ESPAÇOS CONFINADOS DE UNIDADES DE PRÉ PROCESSAMENTO DE GRÃOS.

Leite, L. (30 nov 2016). Sensoriamento de gases em tempo real através de sensores MQ-(0-9)* | Juá Labs. <https://jualabs.wordpress.com/2016/11/30/sensoriamento-de-gases-em-tempo-real-atraves-de-sensores-mq-0-9/>, [accessed on Maio 25].

Lofiego, G. O. V. e Neves, A. D. (3 nov 2020). Automação Residencial: Acionamento remoto via Bluetooth, de uma forma simples e de baixo custo.

Mattede, H. ([S.d.]). O que são sensores e quais as suas aplicações? - Mundo da Elétrica. <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>, [accessed on Jun. 1].

Oliveira, E. (2018). Como usar com Arduino - Módulo Fotorresistor (Sensor) LDR - BLOG MASTER WALKER SHOP. <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-fotorresistor-sensor-ldr/>, [accessed on Jun. 11].

Pessoas com deficiência | Educa | Jovens - IBGE ([S.d.]). <https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/20551-pessoas-com-deficiencia.html>, [accessed on Jun. 1].

Srivastava, D., Kesarwani, A. e Dubey, S. (dez 2018). Measurement of Temperature and Humidity by using Arduino Tool and DHT11. *International Research Journal of Engineering and Technology*, v. 5, n. 12, p. 876.

Teixeira, G. (19 set 2019). Programar ESP32 com a IDE Arduino - Tutorial Completo - Blog UsinaInfo. <https://www.usinainfo.com.br/blog/programar-esp32-com-a-ide-arduino-tutorial-completo/>, [accessed on Jun. 1].

Teja, R. (17 fev 2021). Introduction to ESP32 | Specifications, ESP32 DevKit Board, Layout,. <https://www.electronicshub.org/getting-started-with-esp32/>, [accessed on Jun. 1].

Verdélío, A. (26 ago 2017). Apenas 1% dos brasileiros com deficiência está no mercado de trabalho | Agência Brasil. <https://agenciabrasil.ebc.com.br/direitos-humanos/noticia/2017-08/apenas-1-dos-brasileiros-com-deficiencia-esta-no-mercado-de>, [accessed on Jun. 1].

What is domotics - Discover | Easydom ([S.d.]). <http://www.easydom.com/en/discover/what-is-domotics>, [accessed on Jun. 1].