

UNIVERSIDADE FEDERAL DE OURO PRETO ESCOLA DE MINAS COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO - CECAU



VICTOR DE ABREU MERÇON

SISTEMA DE CONTROLE DE ILUMINAÇÃO E MONITORAMENTO DE CONSUMO ELÉTRICO RESIDENCIAL VIA APLICATIVO

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

VICTOR DE ABREU MERÇON

SISTEMA DE CONTROLE DE ILUMINAÇÃO E MONITORAMENTO DE CONSUMO ELÉTRICO RESIDENCIAL VIA APLICATIVO

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Adrielle de Carvalho Santana Ph.D.

Ouro Preto Escola de Minas – UFOP 01/2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO



Bibliotecário(a) Responsável: Angela Maria Raimundo - SIAPE: 1.644.803





FOLHA DE APROVAÇÃO

Victor de Abreu Merçon

Sistema de Controle de Iluminação e Monitoramento de Consumo Elétrico Residencial Via Aplicativo

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 27 de janeiro de 2022

Membros da banca

Profa. Dra. Adrielle de Carvalho Santana - Orientadora - UFOP

Prof. Dr. Bruno Nazário Coelho - Professor Convidado - UFOP

Prof. Dr. Paulo Marcos de Barros Monteiro - Professor Convidado - UFOP

Adrielle de Carvalho Santana, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 07/02/2022



Documento assinado eletronicamente por Adrielle de Carvalho Santana, PROFESSOR DE MAGISTERIO SUPERIOR, em 07/02/2022, às 12:09, conforme horário oficial de Brasília, com fundamento no art. 6°, § 1°, do Decreto nº 8.539, de 8 de outubro de 2015.



A autenticidade deste documento pode ser conferida no site <u>http://sei.ufop.br/sei/controlador_externo.php?</u> <u>acao=documento_conferir&id_orgao_acesso_externo=0</u>, informando o código verificador **0272428** e o código CRC **D0EC9AD7**.

 $\label{eq:referencia: Caso responde este documento, indicar expressamente o Processo n^{o} 23109.000963/2022-14$

SEI nº 0272428

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000 Telefone: 3135591533 - www.ufop.br

Não é nossa função controlar todas as marés do mundo, mas sim fazer o que pudermos para socorrer os tempos em que estamos inseridos, erradicando o mal dos campos que conhecemos, para que aqueles que viverem depois tenham terra limpa para cultivar. Que tempo encontrarão não é nossa função determinar.

J.R.R. Tolkien

AGRADECIMENTOS

Agradeço aos meus pais por todos ensinamentos e virtudes ensinados e minha irmã pela presença constante.

Agradeço também à Elisa por ser responsável pelos melhores momentos de alegria nos últimos 4 anos e pelo apoio incondicional.

Aos amigos de Nova Lima pelas amizades mais fieis e longevas que tenho, Bella, Jessé, Julinha, Lais, Librelon, Lloyd, Novy, Thiago e Yuri, vocês fazem parte da minha história.

Agradeço à Escola de Minas pelos conhecimentos e pelo aprendizado e aos amigos de Ouro preto por tornarem as noites de estudo mais proveitosas.

Agradeço a Geosol pela oportunidade da minha primeira experiencia profissional.

Por fim agradeço a gloriosa República Taberna e a todos meus irmãos taberneiros pelo acolhimento e por se tornarem minha segunda família.

RESUMO

A automação residencial é definida como o conjunto de técnicas que fazem uso da tecnologia para a automatização de uma habitação. Muitas vezes sendo referida como domótica a automação residencial tem como motivação o conforto dos usuários. Ambientes tecnologicamente interligados são desejáveis, e as pessoas buscam por mais conforto, praticidade, espaços inteligentes e a possibilidade de se conectar sem sair de sua casa. Apesar da evolução das técnicas e tecnologias relacionadas à automação residencial e criação de casas inteligentes, construir uma casa tecnológica e conectada está fora do alcance financeiro de muitas pessoas. Este trabalho tem como objetivo a criação de uma central de controle de iluminação em conjunto com o monitoramento do consumo energético, que seja de acessível e conectado a um aplicativo mobile. Para isso um estudo de como esse sistema deve ser construído foi feito e algumas ferramentas e componentes foram selecionados. Para o desenvolvimento do aplicativo mobile foi utilizado a plataforma Kodular, que além de ter acesso gratuito permite a criação de aplicativos com um backend complexo com relativa facilidade e pouco tempo. A central física tem como microcontrolador o módulo ESP32, que além de ter configurações robustas para o seu tamanho ainda possui conectividade Wi-Fi e Bluetooth embutidas. Com o desenvolvimento do trabalho e a montagem de um protótipo real as simulações de utilização foram feitas e o sistema funcionou corretamente no seu papel de controlar os grupos de iluminação e informar ao usuário o estado dos grupos assim como o consumo de anergia elétrica na residência, os resultados foram atingidos graças a programação correta tanto do aplicativo como do dispositivo físico, assim como a comunicação eficiente feita por meio do banco de dados. Um ponto importante a se considerar foi o custo total do desenvolvimento do protótipo, que apesar de não ser barato ficou abaixo de suas contrapartidas encontradas no mercado. Esse projeto mostra que com criatividade e dedicação é possível criar um sistema para automatizar a iluminação de uma residência sem custos elevados, necessários em sistemas de automatização residencial disponíveis no mercado no momento do desenvolvimento do trabalho.

Palavras-chaves: casa inteligente; automação residencial; iluminação; aplicativo mobile; ESP32

ABSTRACT

Home automation is defined as the set of techniques that uses technology to automate a home. Often referred to as home automation, it is motivated by the comfort of the users. Technologically interconnected environments are necessary, combined with the fact that people are looking for more comfort, practicality, smart spaces and the possibility of connecting without having to leave their homes. Despite the evolution of techniques and technologies related to home automation and the creation of smart homes, building a technological and connected home is beyond the financial reach of many people. This work aims to create a lighting control center together with an energy consumption monitoring, which is low cost and connected to a mobile application. For this reason, a study of how this system should be built was made and some tools and components were selected. For the development of the mobile application, Kodular platform was used, which allows the creation of applications with a complex backend with relative ease and little time, in addition to free access. The physical center has the ESP32 module as a microcontroller, robust configurations for its size and builtin Wi-Fi and Bluetooth connectivity. In order to develop the work and assembly a real prototype, simulations were made, and the system worked correctly in its function of controlling the lighting groups and informing the user the status of the groups as well as the consumption of electrical energy in the residence, the results were achieved thanks to the correct programming of both the application and the physical device, as well as the efficient communication made through the database. An important point to consider is that the total cost of developing the prototype was below their counterparts found in the market, even though it's not cheap. This project shows that with creativity and dedication it is possible to create a system to automate a home without high costs, necessary in home automation systems available on the market in the moment of work development.

Key-words: smart home; home automation; lighting; mobile app; ESP32

LISTAS DE ILUSTRAÇÕES

Figura 2.1 Fluxograma das atividades planejadas	17
Figura 2.2 Módulo WiFi ESP32 Bluetooth com descrição dos pinos	19
Figura 2.3 Esquema de um módulo relé de 8 canais, 5 Volts	20
Figura 2.4 Tela mostrada na aba Designer do Kodular	22
Figura 2.5 Tela mostrada na aba Blocks do Kodular	22
Figura 3.1 Wireframe do aplicativo	27
Figura 3.2 Tela inicial do aplicativo SisLux	28
Figura 3.3 Aba "Grupos" do menu	29
Figura 3.4 Aba "Energia" do menu	30
Figura 3.5 Aba "Configurações" do menu	30
Figura 3.6 Mensagem mostrada ao usuário enquanto procura por dispositivos Bluetooth	31
Figura 3.7 Lista de dispositivos encontrados	32
Figura 3.8 Mensagem para inserir nome da rede	32
Figura 3.9 Mensagem para inserir senha da rede	33
Figura 3.10 Tela da configuração da aba "Grupos"	34
Figura 3.11 Mensagem de configuração da tarifa de energia	35
Figura 3.12 Identificador da planilha	35
Figura 3.13 Diagrama de montagem do dispositivo físico	37
Figura 3.14 Fluxograma do funcionamento básico do dispositivo físico	38
Figura 4.1 Montagem do circuito definitivo	42
Figura 4.2 Configuração do dispositivo físico, visão do aplicativo, parte 1. Menu de	
configuração	43
Figura 4.3 Configuração do dispositivo físico, visão do aplicativo, parte 2. Procurando po	r
dispositivos Bluetooth	43
Figura 4.4 Configuração do dispositivo físico, visão do aplicativo, parte 3. Seleção do	
dispositivo Bluetooth	44
Figura 4.5 Configuração do dispositivo físico, visão do aplicativo, parte 4. Pareamento co	om o
dispositivo Bluetooth	44
Figura 4.6 Configuração do dispositivo físico, visão do aplicativo, parte 5. Fornecimento	do
nome da rede	45
Figura 4.7 Configuração do dispositivo físico, visão do aplicativo, parte 6. Fornecimento	da
senha da rede	45
Figura 4.8 Configuração do dispositivo físico, visão do aplicativo, parte 7. Confirmação d	la
$rac{1}{2}$	46
Figura 4.9 Dispositivo fisico anunciando que foi configurado	46
Figura 4.10 Inicio das leituras de corrente	4/
Figura 4.11 Atualização do consumo no banco de dados	48
Figura 4.12 Tela do aplicativo mostrando para o usuario a modificação do consumo	48
Figura 4.13 Tela de modificação dos grupos no aplicativo	49
Figura 4.14 Tela dos grupos apos modificação	49
Figura 4.15 Banco de dados apos alteração do estado do grupo	50
Figura 4.10 Tela dos grupos com a lampada que indica o estado do grupo 1 Iol alterada	
rigura 4.17 Communação do interruptor do grupo dois sendo pressionado e alteração no b	anco 5 1
ut uauos. Figura 4 18 Tala dos grupos com a lâmpada que indica o estado do grupo 2 foi alterado	
Figura 4.10 rela dos grupos com a lampada que muica o estado do grupo 2 los alterada Figura A 1 Declaração da variável global instante na tala inicial	
Figura A 2 Declaração da variával global LIDI, no tala inicial	
i iguia A.2 Decialação da valiavel global ONL lia tela illicial	

Figura A.4 Programação da função Consumo_mensal .61 Figura A.5 Programação para recuperar os dados obtidos através do protocolo de .61 comunicação HTTP pa tela incial parte 1 .62 Figura A.7 Programação para recuperar os dados obtidos através do protocolo de .63 comunicação HTTP parte 2 .63 Figura B.1 Declaração da variável global URL na tela menu .64 Figura B.3 Programação da função inicializa_grupos .66 Figura B.4 programação da função inicializa_grupos .66 Figura B.4 programação da função inicializa_grupos .67 Figura B.6 Programação do timer que controla a frequência de chamada de uma função	Figura A.3 Programação da inicialização da tela inicial	60
Figura A.5 Programação do clock da tela inical .61 Figura A.6 Programação para recuperar os dados obtidos através do protocolo de .62 comunicação HTTP na tela inicial parte 1 .63 Figura B. 7 Programação para recuperar os dados obtidos através do protocolo de .63 Figura B.1 Declaração da variável global URL na tela menu .64 Figura B.2 Programação da função inicializa, status .65 Figura B.5 Programação da função inicializa_grupos. .67 Figura B.6 Programação da função inicializa_consumo .68 Figura C.2 Programação da função controla a frequência de chamada de uma função	Figura A.4 Programação da função Consumo_mensal	61
Figura A.6 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela incial parte 1	Figura A.5 Programação do clock da tela inical	61
comunicação HTTP na tela incial parte 1	Figura A.6 Programação para recuperar os dados obtidos através do protocolo de	
Figura A.7 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP parte 2	comunicação HTTP na tela incial parte 1	62
comunicação HTTP parte 2 63 Figura B.1 Declaração da variável global URL na tela menu 64 Figura B.2 Programação da função inicializa, status 65 Figura B.3 Programação da função inicializa_grupos. 67 Figura B.5 Programação da função inicializa_grupos. 67 Figura B.5 Programação da função inicializa_grupos. 67 Figura B.6 Programação da função inicializa_consumo 68 Figura C.1 Programação da função consumo_mensal 69 Figura C.2 Programação da função consumo_mensal 70 Figura C.4 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 1 70 Figura D.2 Programação do controle do grupo 1 72 Figura D.2 Programação do controle do grupo 2 71 Figura D.3 Programação do controle do grupo 5 73 Figura D.4 Programação do controle do grupo 5 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura E.2 Programação do controle do grupo 7	Figura A.7 Programação para recuperar os dados obtidos através do protocolo de	
Figura B. Í Declaração da variável global URL na tela menu	comunicação HTTP parte 2	63
Figura B.2 Programação da inicialização da tela menu	Figura B.1 Declaração da variável global URL na tela menu	64
Figura B.3 Programação da função Inicializa_status	Figura B.2 Programação da inicialização da tela menu	64
Figura B.4 programação da função inicializa_grupos	Figura B.3 Programação da função Inicializa_status	65
Figura B.5 Programação da função inicializa_grupos	Figura B.4 programação da função adiciona_componentes	66
Figura B.6 Programação da função inicializa_consumo 68 Figura C.1 Programação da função consumo_mensal	Figura B.5 Programação da função inicializa_grupos	67
Figura C.1 Programação do timer que controla a frequência de chamada de uma função69 Figura C.2 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 1	Figura B.6 Programação da função inicializa consumo	68
Figura C.2 Programação da função Consumo_mensal	Figura C.1 Programação do timer que controla a frequência de chamada de uma função	69
Figura C.3 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 1	Figura C.2 Programação da função Consumo mensal	69
comunicação HTTP na tela menu parte 1 70 Figura C.4 Programação para recuperar os dados obtidos através do protocolo de 71 Figura D.1 Programação do controle do grupo 1 72 Figura D.2 Programação do controle do grupo 3 72 Figura D.3 Programação do controle do grupo 4 73 Figura D.4 Programação do controle do grupo 5 73 Figura D.5 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 7 74 Figura E.1 Programação do controle do grupo 7 74 Figura E.2 Programação do controle do grupo 7 74 Figura E.3 Programação do controle do grupo 7 74 Figura E.4 Programação do controle do grupo 7 75 Figura E.2 Programação do controle do grupo 7 75 Figura E.4 Programação do controle do grupo 7 76 Figura E.2 Programação do controle do grupo 7 76 Figura E.4 Programação do evento que é acionado após o usuário escolher um dispositivo 76 Figura F.1 Programação da configuração do nome do grupo 1 78 </td <td>Figura C.3 Programação para recuperar os dados obtidos através do protocolo de</td> <td></td>	Figura C.3 Programação para recuperar os dados obtidos através do protocolo de	
Figura C. ⁴ Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 2	comunicação HTTP na tela menu parte 1	70
romunicação HTTP na tela menu parte 2	Figura C.4 Programação para recuperar os dados obtidos através do protocolo de	
Figura D.1 Programação do controle do grupo 1 72 Figura D.2 Programação do controle do grupo 2 72 Figura D.3 Programação do controle do grupo 3 72 Figura D.4 Programação do controle do grupo 4 73 Figura D.5 Programação do controle do grupo 5 73 Figura D.7 Programação do controle do grupo 6 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do controle do grupo 8 74 Figura E.2 Programação do controle do grupo 8 74 Figura E.3 Programação do controle do grupo 8 74 Figura E.3 Programação do controle do grupo 8 74 Figura E.4 Programação do controle do grupo 8 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 Figura E.4 Programação do evento que é acionado após o usuário inserir dados sobre a rede Wi-Fi 77 Figura F.1 Programação da configuração que defina a visibilidade do grupo 1 ao usuário 78 Figura F.2 Programação da configuração que defina a visibilidade do grupo 2 ao usuário 78 Figura F.4 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 Figura F.4 Programaç	comunicação HTTP na tela menu parte 2	71
Figura D.2 Programação do controle do grupo 2 72 Figura D.3 Programação do controle do grupo 3 72 Figura D.4 Programação do controle do grupo 4 73 Figura D.5 Programação do controle do grupo 5 73 Figura D.6 Programação do controle do grupo 6 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do controle do grupo 8 74 Figura E.2 Programação do controle do grupo 8 74 Figura E.2 Programação do controle do grupo 8 75 Figura E.2 Programação do controle que fazer quando o escaneamento por novos 75 dispositivos for concluído 75 Figura E.4 Programação do evento que é acionado após o usuário escolher um dispositivo 76 Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 77 Figura F.1 Programação da configuração que defina a visibilidade do grupo 1 ao usuário	Figura D.1 Programação do controle do grupo 1	72
Figura D.3 Programação do controle do grupo 3 72 Figura D.4 Programação do controle do grupo 4 73 Figura D.5 Programação do controle do grupo 5 73 Figura D.6 Programação do controle do grupo 7 74 Figura D.7 Programação do controle do grupo 8 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do controle do grupo 8 74 Figura E.2 Programação do controle do grupo 7 75 Figura E.2 Programação do controle do grupo 6 75 Figura E.2 Programação do controle do grupo 7 75 Figura E.2 Programação do controle do grupo 7 75 Figura E.2 Programação do evento que fazer quando o escaneamento por novos dispositivos for concluído 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 76 Figura F.1 Programação da configuração do nome do grupo 1 78 Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário 78 78 Figura F.3 Programação da configuração do nome do grupo 3 78 Figura F.4 Programação da configuração do nome do grupo 3 78 Figura F.5 Programação da configuração que defina a visibilidade do grupo 3 ao usuário	Figura D.2 Programação do controle do grupo 2	72
Figura D.4 Programação do controle do grupo 4 73 Figura D.5 Programação do controle do grupo 5 73 Figura D.6 Programação do controle do grupo 6 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do controle do grupo 8 74 Figura E.2 Programação do botão de configuração de dispositivos 75 Figura E.2 Programação do evento que fazer quando o escaneamento por novos 75 figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 76 Figura E.4 Programação do evento que é acionado após o usuário escolher um dispositivo 76 76 Figura F.1 Programação do evento que é acionado após o usuário inserir dados sobre a rede Wi-Fi 77 Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário 78 78 Figura F.4 Programação da configuração do nome do grupo 2 78 Figura F.5 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 78 Figura F.4 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 79	Figura D.3 Programação do controle do grupo 3	72
Figura D.5 Programação do controle do grupo 5 73 Figura D.6 Programação do controle do grupo 6 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do controle do grupo 8 74 Figura E.2 Programação do controle do grupo 8 75 Figura E.2 Programação do controle do grupo 8 75 Figura E.2 Programação do evento que fazer quando o escaneamento por novos 75 Gispositivos for concluído. 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 76 Figura E.5 Programação do evento que é acionado após o celular parear com um dispositivo 76 Figura F.1 Programação da configuração do nome do grupo 1 78 Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário 78 Figura F.3 Programação da configuração que defina a visibilidade do grupo 2 ao usuário 78 Figura F.4 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 Figura F.4 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 Figura F.5 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 Figura F.7 Programação da configuraçã	Figura D.4 Programação do controle do grupo 4	73
Figura D.6 Programação do controle do grupo 6 73 Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do botão de configuração de dispositivos 75 Figura E.2 Programação que indica o que fazer quando o escaneamento por novos 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo 76 Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 77 Figura F.1 Programação da configuração do nome do grupo 1 78 Figura F.2 Programação da configuração do nome do grupo 2 78 Figura F.3 Programação da configuração que defina a visibilidade do grupo 2 ao usuário 78 Figura F.5 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 78 Figura F.7 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 4 ao usuário 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 5 ao usuário 79 Figura F.9 Programação da configuração que defina a visibilidade do grupo 5 ao usuário 79 <t< td=""><td>Figura D.5 Programação do controle do grupo 5</td><td>73</td></t<>	Figura D.5 Programação do controle do grupo 5	73
Figura D.7 Programação do controle do grupo 7 74 Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação do botão de configuração de dispositivos 75 Figura E.2 Programação que indica o que fazer quando o escaneamento por novos 75 Gispositivos for concluído 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo 76 Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 77 Figura F.1 Programação da configuração que defina a visibilidade do grupo 1 ao usuário	Figura D.6 Programação do controle do grupo 6	73
Figura D.8 Programação do controle do grupo 8 74 Figura E.1 Programação que indica o que fazer quando o escaneamento por novos 75 Figura E.2 Programação que indica o que fazer quando o escaneamento por novos 75 dispositivos for concluído. 75 Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo 75 Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo 76 Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 76 Figura F.1 Programação da configuração do nome do grupo 1 78 Figura F.2 Programação da configuração do nome do grupo 2 78 Figura F.3 Programação da configuração que defina a visibilidade do grupo 1 ao usuário 78 Figura F.4 Programação da configuração que defina a visibilidade do grupo 2 ao usuário 78 Figura F.5 Programação da configuração que defina a visibilidade do grupo 2 ao usuário 78 Figura F.6 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 3 ao usuário 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 4 ao usuário 79 Figura F.7 Programação da configuração que defina a visibilidade do grupo 5 ao usuário <t< td=""><td>Figura D.7 Programação do controle do grupo 7</td><td>74</td></t<>	Figura D.7 Programação do controle do grupo 7	74
Figura E.1 Programação do botão de configuração de dispositivos	Figura D.8 Programação do controle do grupo 8	74
Figura E.2 Programação que indica o que fazer quando o escaneamento por novos dispositivos for concluído	Figura E.1 Programação do botão de configuração de dispositivos	75
dispositivos for concluído	Figura E.2 Programação que indica o que fazer quando o escaneamento por novos	
Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo75 Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo 	dispositivos for concluído	75
Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 	Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo.	75
	Figura E.4 Programação do evento que é acionado após o celular parear com um dispositiv	0
Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi 		76
77Figura F.1 Programação da configuração do nome do grupo 1	Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-	Fi
Figura F.1 Programação da configuração do nome do grupo 1		77
Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário78 Figura F.3 Programação da configuração do nome do grupo 2	Figura F.1 Programação da configuração do nome do grupo 1	78
Figura F.3 Programação da configuração do nome do grupo 2	Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário	78
Figura F.4 Programação da configuração que defina a visibilidade do grupo 2 ao usuário78 Figura F.5 Programação da configuração do nome do grupo 3	Figura F.3 Programação da configuração do nome do grupo 2	78
Figura F.5 Programação da configuração do nome do grupo 3	Figura F.4 Programação da configuração que defina a visibilidade do grupo 2 ao usuário	78
Figura F.6 Programação da configuração que defina a visibilidade do grupo 3 ao usuário78 Figura F.7 Programação da configuração do nome do grupo 4	Figura F.5 Programação da configuração do nome do grupo 3	78
Figura F.7 Programação da configuração que defina a visibilidade do grupo 4 ao usuário79 Figura F.8 Programação da configuração que defina a visibilidade do grupo 4 ao usuário79 Figura F.9 Programação da configuração do nome do grupo 5	Figura F.6 Programação da configuração que defina a visibilidade do grupo 3 ao usuário	78
Figura F.8 Programação da configuração que defina a visibilidade do grupo 4 ao usuário79 Figura F.9 Programação da configuração do nome do grupo 5	Figura F.7 Programação da configuração do nome do grupo 4	79
Figura F.9 Programação da configuração do nome do grupo 5	Figura F.8 Programação da configuração que defina a visibilidade do grupo 4 ao usuário	79
Figura F.10 Programação da configuração que defina a visibilidade do grupo 5 ao usuário79 Figura F.11 Programação da configuração do nome do grupo 6	Figura F.9 Programação da configuração do nome do grupo 5	79
Figura F.11 Programação da configuração que defina a visibilidade do grupo 6 ao usuário79 Figura F.12 Programação da configuração que defina a visibilidade do grupo 6 ao usuário79 Figura F.13 Programação da configuração do nome do grupo 7	Figura F.10 Programação da configuração que defina a visibilidade do grupo 5 ao usuário.	79
Figura F.12 Programação da configuração que defina a visibilidade do grupo 6 ao usuário79 Figura F.13 Programação da configuração do nome do grupo 7	Figura F.11 Programação da configuração do nome do grupo 6	79
Figura F.13 Programação da configuração do nome do grupo 7	Figura F.12 Programação da configuração que defina a visibilidade do grupo 6 ao usuário.	79
	Figura F.13 Programação da configuração do nome do grupo 7	80
Figura F.14 Programação da configuração que defina a visibilidade do grupo 7 ao usuário80	Figura F.14 Programação da configuração que defina a visibilidade do grupo 7 ao usuário.	80

Figura F.15 Programação da configuração do nome do grupo 8	80
Figura F.16 Programação da configuração que defina a visibilidade do grupo 8 ao usuário	80
Figura G.1 Programação da configuração da tarifa de energia	81
Figura G.2 Programação de evento disparado após o usuário inserir dados sobre a tarifa de	
energia	81

LISTAS DE TABELAS

Tabela 1 Preços dos produtos oferecidos pela Houseasy	25
Tabela 2 Valores gastos para a montagem do dispositivo físico	52

LISTA DE ABREVIATURAS E SIGLAS

BLE	Bluetooth Low Energy
GPIO	General Purpose Input/Output
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
ІоТ	Internet of Things
LED	Light Emitting Diode
KB	Kilobyte
MHz	Megahertz
PWM	Pulse Width Modulation
RAM	Random Access Memory

1	INTRODUÇÃO	13
1.1	Contextualização	.13
1.2	Objetivos gerais e específicos	.15
1.3	Justificativa do trabalho	.15
1.4 2	Estrutura do trabalho REVISÃO DA LITERATURA	.16 17
2.1 2. 2. 2.2 2.2	Hardware 1.1 Módulos de microcontroladores 2.1.1.1 Arduino Uno R3 2.1.1.2 Modulo Wi-Fi ESP32 Bluetooth 30 pinos 1.2 Relés e módulos de relés 1.3 Sensor de corrente Software	.18 18 19 19 20 21 21 21
2	 2.2.1.2 Kodular 2.2 Banco de dados 2.2.2.1 Firebase 2.2.2.2 Google Sheets 2.3 Ambiente de desenvolvimento integrado do Arduino 	21 23 23 23 23
2.3 3	Sistema similar DESENVOLVIMENTO	.24 26
3.1	Aplicativo SisLux	26
3.2	Banco de dados	.35
3.3 4 5	Dispositivo físico SisLux RESULTADOS CONCLUSÃO	.36 41 54
REF	YERÊNCIAS	56
APE APÊ	ENDICE A – TELA INICIAL DO APLICATIVO SISLUX ENDICE B – TELA MENU DO APLICATIVO SISLUX (INICIALIZAÇÃO)	59 64
APE APÊ APÊ	ENDICE C – TELA MENU DO APLICATIVO SISLUX (ATUALIZAÇAO DOS DADOS) ÈNDICE D – TELA MENU DO APLICATIVO SISLUX (CONTROLE DOS GRUPOS) ÈNDICE E – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DO	69 72
DIS APÊ APÊ	POSITIVO FÍSICO) ÈNDICE F – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DOS GRUPO 78 ÈNDICE G – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DA TARIFA) 81	75)S))
APÊ APÊ APÊ	ÈNDICE H – PROGRAMAÇÃO APPS SCRIPT (CÓDIGO APLICATIVO) ÈNDICE I – PROGRAMAÇÃO APPS SCRIPT (CÓDIGO ESP32) ÈNDICE J – PROGRAMAÇÃO ESP32	82 83 85

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

A automação residencial é definida como o conjunto de técnicas que fazem uso da tecnologia para a automatização de uma habitação. Muitas vezes sendo referida como domótica a automação residencial tem como motivação o conforto dos usuários. É adequada para ambientes estáticos e altamente recomendada para realizar tarefas relativamente simples (COELHO e CRUZ, 2017).

As primeiras ideias de automatização residencial remontam da década de 80, quando surgiu a ideia de automatizar pequenas tarefas como a abertura de portões e o controle de iluminação utilizando controles remotos. Com o foco no conforto dos habitantes da residência, também foram introduzidos elementos que visavam a segurança do local com a aplicação de alarmes. Com a evolução das técnicas relacionadas à domótica surgiram 3 subáreas relacionadas a automação residencial:

- Automatização
- Segurança residencial
- Conforto

Na automatização os atuadores ativos são instalados em quaisquer partes móveis da edificação, portões, portas de garagens, venezianas e toldos, podendo utilizar sensores para a ampliação das funcionalidades, como, por exemplo, recolher um varal automaticamente quando for identificada chuva.

A segurança residencial pode ir além de alarmes contra intrusos, podendo funcionar como controle de acesso, monitoramento de ambientes por meio do controle de câmeras e acionamento automático de luzes para simular a presença de moradores.

Por fim, o Conforto e a personalização residencial, abrange todo controle simples de dispositivos, como controlar eletrodomésticos, regular a temperatura e controlar a iluminação. Paralelamente a isso surgiu a partir do início dos anos 2000 o conceito de casa conectada, que insere o uso da internet no cotidiano das pessoas e a possibilidade de conectar variados dispositivos à rede para envio de informações e monitoramento. Assim surge da fusão da automação residencial com a casa conectada (a casa inteligente), onde a comunicação e

interatividade entre os aparelhos e dispositivos da residência garante a difusão de informação e a automação dos ambientes da habitação (JR e FARINELLI, 2018).

O potencial de mercado apresentado pelos conceitos de ambientes inteligentes é grande, apesar de parecer algo futurista e fora de alcance. Estudos realizados nos Estados Unidos nos anos de 1998, 2002 e 2008 mostraram que o potencial mercado de casas inteligentes pelo menos dobrou a cada ano amostrado. Isso mostra o crescimento exponencial do mercado (PRUDENTE, 2011).

Esse potencial também é percebido no Brasil, onde nos últimos anos o número de fornecedores de equipamentos adequados a automação residencial aumentou consideravelmente (MURATORI, 2013). De acordo com Muratori (2013) :

"O consumidor já iniciou o seu processo de "descoberta" dos beneficios da Automação Residencial e é apenas uma questão de tempo a adoção cada vez mais intensa de sistemas integrados em nossas residências."

A energia se tornou um elemento chave da sociedade moderna, e o desenvolvimento de regiões ou países está intimamente atrelado ao fornecimento abundante e confiável de energia, usada para transformar recursos naturais em bens de consumo e valor econômico. Podemos perceber essa ligação ao analisarmos que nas últimas 3 décadas o consumo de energia em escala mundial teve quase 100% de aumento, embora os Estados Unidos, o maior consumidor de energia em escala global, tenha aumentado seu consumo em 25%, o que nos leva a entender que a grande porcentagem do crescimento do consumo energético ocorreu em países emergentes. Porém o desenvolvimento habilitado pela oferta de energia vem degradando o ambiente de forma acelerada. Com os combustíveis fósseis ocupando 90% da matriz energética global as emissões de gases poluentes seguem de perto o desenvolvimento e enriquecimento da sociedade. Por isso o desenvolvimento econômico sustentável está em foco na nossa atualidade, e medidas vêm sendo criadas para amenizar os danos ambientais provocados pelo avanço tecnológico. Como exemplo, em 1995 os carros modernos emitiam cerca de 1% dos poluentes em relação aos carros produzidos antes de 1970 (HINRICHS e KLEINBACH, 2014).

Com cerca de 25% do consumo energético do Brasil estando localizado em residências (MACÊDO, 2016), e os chuveiros, lâmpadas e refrigeradores consumindo cerca de 50% da energia dentro das residências (PROCEL, 2021), é possível chegar à conclusão que o monitoramento do consumo dentro das residências é vital para um crescimento sustentável da sociedade. A iluminação inteligente, seja por controle remoto, aplicativo mobile, por temporizadores, relés fotoelétricos e muitas outras técnicas, não contribui somente para o conforto dos residentes da construção como também para a economia de energia. O hábito de

deixar luzes acesas durante o dia onde tem iluminação solar o suficiente, ou mesmo onde não existam pessoas utilizando o ambiente contribuem muito para o aumento do gasto energético da residência, pois de acordo com PROCEL (2021) 9,67% do gasto energético de uma casa é atribuído à iluminação.

Com o conceito de casa inteligente ficando cada vez mais acessível, os esforços das empresas ligadas a esse segmento aumentam e buscam a simplificação dos sistemas já existentes para garantir o aceso a qualquer pessoa. Soluções que vão desde geladeiras programadas para enviar uma mensagem ao usuário quando produtos do cotidiano estão acabando até lâmpadas conectadas com assistentes virtuais, que habilitam o controle das mesmas através de um comando de voz, se mostram cada vez mais presentes na rotina dos possíveis consumidores. Dessa forma casas com tecnologias integradas vêm ganhando destaque, ainda mais após a pandemia causada pelo Covid-19 onde a prática de *home office* se tornou parte do cotidiano da maioria. Ambientes tecnologicamente interligados são necessários, e as pessoas buscam por mais conforto, praticidade, espaços inteligentes e a possibilidade de se conectar sem sair de sua casa (ZANATTA, 2021).

1.2 Objetivos gerais e específicos

Nesse trabalho tem-se como objetivo a criação de uma central de controle de iluminação para os diversos ambientes de uma residência em conjunto com o monitoramento em tempo real do consumo energético do local integrado a um aplicativo mobile, que será a plataforma onde o usuário poderá fazer o monitoramento da sua residência, assim como definir rotinas para a ativação ou desativação das luzes em determinados ambientes. Como objetivos específicos desse trabalho tem-se:

- Desenvolver uma plataforma mobile simples e intuitiva para o usuário;
- Construir uma central versátil de forma similar à dispositivos plug-and-play;
- Gastar a menor quantidade de recursos financeiros quanto o possível a fim de comparar o sistema desenvolvido com sistemas comerciais.

1.3 Justificativa do trabalho

O conceito de casas inteligentes se mostra cada vez mais presente, porém não mais perto de fazer parte do cotidiano da maioria das pessoas. Grande parte disso se deve ao elevado custo em se adquirir os equipamentos necessários. Como exemplo, uma lâmpada de LED que se conecta à rede Wi-Fi pode ser encontrada no mercado por volta de R\$100,00. Quando projetos são ampliados para toda uma residência os valores podem ficar bem altos.

Além disso alguns projetos necessitam de conhecimento técnico devido à complexidade exigida. Assim o objetivo desse trabalho é desenvolver um sistema que possa ser instalado em uma residência pequena ou média, que use tecnologias bem difundidas e conhecidas pelo público, que seja de acessível e que precise do mínimo de conhecimento técnico o possível do usuário.

Assim, justifica-se a realização deste trabalho.

1.4 Estrutura do trabalho

O trabalho em questão está organizado em 5 capítulos, cujos conteúdos são apresentados a seguir.

O primeiro capítulo trata da introdução. Nele são abordados a contextualização do tema do trabalho, os objetivos gerais e específicos que se espera alcançar com o trabalho, a justificativa para a realização do trabalho e pôr fim a estrutura do trabalho em si.

O segundo capítulo consiste na revisão da literatura pesquisada para a realização do trabalho. Aqui são encontradas uma revisão das principais tecnologias que permitem a construção de um sistema IoT, desde o *hardware* até o *software*, análise dos possíveis componentes físicos para a construção da central e plataformas de desenvolvimento de aplicações *mobile*, assim como as soluções semelhantes existentes atualmente.

O terceiro capítulo aborda a metodologia usada no desenvolvimento. Nele os materiais escolhidos são apresentados e a utilização de cada um para o desenvolvimento do sistema proposto é explicado.

No quarto capítulo é feita a apresentação dos resultados atingidos. Nele o funcionamento do sistema pronto é demonstrado e um comparativo entre o sistema desenvolvido e sistemas comerciais similares é feito.

No quinto e último capítulo tem-se uma síntese dos resultados do trabalho e das conclusões atingidas, assim como uma proposta para trabalhos futuros caso seja necessário.

2 REVISÃO DA LITERATURA

Para a realização desse trabalho foram utilizados diversos equipamentos para a construção do *hardware* e algumas ferramentas em forma de *software*. Dessa forma uma análise prévia das diversas técnicas foi necessária para um desenvolvimento competente.

Da mesma forma que o estudo prévio das diversas ferramentas e dos diversos equipamentos foi necessário, para manter a organização da evolução do trabalho e se orientar na ordem de execução das atividades foi criado um fluxograma, representado na Figura 2.1, que abrange desde o início do trabalho até a defesa do mesmo.



Figura 2.1 Fluxograma das atividades planejadas Fonte: Autoria própria

Com o tema em mente se iniciou-se a revisão bibliográfica do trabalho, onde foram definidas as ferramentas, técnicas e métodos mais adequados para o desenvolvimento do trabalho proposto.

Durante os estudos feitos na etapa de revisão bibliográfica uma série de ferramentas e componentes foram selecionados para a execução do trabalho. Devido a variedade de recursos selecionados dois grupos foram formados para facilitar o detalhamento.

Portanto nesse capitulo serão apresentados os conceitos teóricos e as pesquisas que foram utilizadas para a realização desse trabalho divididos nos seguintes tópicos:

- Hardware
 - o Módulos de microcontroladores
 - Arduino Uno R3
 - Modulo Wi-Fi ESP32 Bluetooth 30 pinos
 - o Relés e módulos de relés
 - Sensores de corrente
- Software
 - o Plataformas de desenvolvimento
 - Bubble.io
 - Kodular
 - o Banco de dados
 - Firebase
 - Google Sheets
 - o Ambiente de desenvolvimento integrado
- Sistemas similares

2.1 Hardware

2.1.1 Módulos de microcontroladores

Dois módulos microcontrolados foram estudados para uso neste trabalho, o Arduino Uno R3 e o ESP32. A seguir os dois serão apresentados.

2.1.1.1 Arduino Uno R3

As plataformas da família Arduino são ferramentas caracterizadas por serem *opensource* de fácil utilização destinada a qualquer pessoa interessada a desenvolver projetos de pequeno ou médio porte (MOTA, 2017).

O modelo Arduino Uno R3 é composto por um microcontrolador ATmega328 de 8 bits e possui 32 KB de Flash, 2 KB de RAM e um clock máximo de 20Mhz, além disso possui 14 pinos I/O digitais dos quais 6 podem ser usados como saída PWM e 6 pinos I/O analógicos. O modelo em questão pode ser alimentado com uma fonte de 7 à 12V (SOUZA, 2013).

2.1.1.2 Modulo Wi-Fi ESP32 Bluetooth 30 pinos

O módulo WiFi ESP32 Bluetooth representado na Figura 2.2 foi escolhido para ser um dos candidatos à microcontrolador responsável pelo controle da parte física do trabalho pois esse módulo apresenta diversas funcionalidades inclusas, como conectividade WiFi e BLE, que em outros casos teriam que ser adicionados ao sistema com a ajuda de outros módulos. Possui também um microcontrolador Xtensa Dual-Core 32-bit LX6 com arquitetura de 32 bits, com 4 MB de Flash, 520 KB de RAM, com um clock máximo de 240Mhz. Além disso possui 25 GPIOs com diversas funcionalidades, como portas analógicas PWM, interfaces Seriais entre outros.

Também é importante ressaltar que esse módulo é um dos mais encontrados no mercado brasileiro, tendo também um ótimo custo-benefício em razão das funcionalidades apresentadas (COELHO, 2020).



Figura 2.2 Módulo WiFi ESP32 Bluetooth com descrição dos pinos Fonte: (COELHO, 2020)

2.1.2 Relés e módulos de relés

O relé é um dispositivo elétrico capaz de modificar a atuação de forma rápida em diversos circuitos de saída, sem alterar o circuito principal. Dessa forma um relé pode ativar ou desativar circuitos secundários sem que o circuito principal seja alterado (Cunha, 2009).

Os relés podem ser facilmente instalados em módulos que já possuem os circuitos e componentes, como transistores, diodos e LEDs, necessários para o acionamento dos mesmos,

como pode ser observado na Figura 2.3, que ilustra um módulo relé de 8 canais acionado com 5 Volts.



Figura 2.3 Esquema de um módulo relé de 8 canais, 5 Volts Fonte: (ELETRÔNICA, 2021)

Alguns módulos possuem acopladores óticos que separam de forma mais eficiente o circuito principal e o circuito da carga. Esse método é chamado de isolamento galvânico. Nesse tipo de isolamento o circuito está fisicamente desconectado e os sinais são enviados através de radiação infravermelha a partir de um LED emissor infravermelho e coletados por um sensor correspondente (BRAGA, 2021).

2.1.3 Sensor de corrente

Para o monitoramento do consumo de energia elétrica o sensor de corrente ACS712, produzido pela *Allegro MicroSystems*, é uma opção. O sensor em questão tem a capacidade de medir uma faixa de corrente entre -30 e +30 ampères tanto em corrente alternada quanto em corrente contínua e sua leitura é feita por meio de um pino de saída analógica (Allegro MicroSystems, 2015). Essa escolha foi feita com base na faixa de corrente que se espera medir na aplicação residencial.

2.2 Software

2.2.1 Plataformas de desenvolvimento

2.2.1.1 Thunkable

A plataforma online Thunkable é uma ferramenta para o desenvolvimento de aplicativos *mobile* sem o uso de códigos de programação. A sua interface se baseia em um sistema conhecido como *drag and drop* que permite que o usuário selecione componentes para a construção do *frontend* da aplicação e desenvolva o *backend* com blocos funcionais.

Com isso é possível criar aplicações nativas para Android e IOS de forma que ao disponibilizar a aplicação ao público a leitura, a compreensão e a modificação por usuários menos familiarizados com as linguagens de programação requisitadas sejam possíveis.

2.2.1.2 Kodular

Da mesma forma que a plataforma Thunkable, apresentada anteriormente, o Kodular permite a construção de aplicativos mobile sem o uso de conhecimentos em linguagens de programação. De acordo com Kodular (2021):

> Kodular foi criado para fornecer um pacote acessível para que todos possam aprender e, posteriormente, criar aplicativos profissionalmente. Por isso, acreditamos fortemente que aprender a codificar deve ser um direito e ninguém deveria pagar para aprender o básico de programação. O Kodular Creator oferece uma das ferramentas mais completas da Internet para criar aplicativos sem codificação e, além disso, está sendo usado por milhares de alunos em todo o mundo. Todos devem ser capazes de transformar suas ideias em realidade, independentemente de sua renda. Alguém pode ter a próxima grande ideia revolucionária, mas às vezes sem dinheiro isso não pode ser feito.

Criado como uma evolução da plataforma Makeroid, que por sua vez foi baseada na plataforma App Inventor desenvolvida pelo MIT, o Kodular é um serviço gratuito onde aplicativos podem ser construidos arrastando os componetes pela tela e montando o *layout* do *frontend* usando a aba *Designer* mostrada na figura Figura 2.4.

0	Cre	ator	Projec	t Test Export Help				• • • • • • •
SisL	ux		Scr	aeen1 ⊙ Add Screen ⊙ Copy Screen ○ Remove Screen ⊗				Designer Blocks
Pale	te		Q	Viewer	⊑∎ Google Pixel 3 v	Ø	All Components 🗸 🧷 🗍	Screen1 Properties
L=D		Button	0				⊖ Screen1	Common properties
	~	Checkbox	0				Lottie1	About Screen
۲	Ģ	Circular Progress	0				O Clock1	- About Screen Background Color
© D	0	Custom Progress	0	IST-			Web1	#444444FF About Screen Light Theme
Ø	۵	Date Picker	0	2011-			Tiny_D81	About Screen Title
?	Ð	Floating Action Button	0					- Align Horizontal
D		Image	0					Center : 3 v
Ĉ		Label	0				>	Center : 2 V
2	**	Linear Progressbar	0	<		>		Background Color
G	5=	List Picker	0					- Background Image
\$	۰	Notifier	0					None ~ 🕑 🗹
騇	۲	Radio Button	0					Close Screen Animation
ង	*	Rating Bar	0					- Navigation Bar Color
		Slider	()	(• #000000FF
		Snackbar	()					Open Screen Animation
	0	Spinner	0					Screen Orientation
		6 (A) (A)	~					- Cintan

Figura 2.4 Tela mostrada na aba Designer do Kodular

Fonte: Autoria própria

A criação das funções que vão realizar todas as operações do aplicativo, ou seja, para a criação do *backend*, é usada a aba *Blocks* mostrada na Figura 2.5, utilizando blocos que representam funções ou logicas de programação o *backend* é construido em uma dinâmica que lembra a montagem de um LEGO.



Figura 2.5 Tela mostrada na aba *Blocks* do Kodular Fonte: Autoria própria

2.2.2 Banco de dados

No trabalho o banco de dados foi usado para o monitoramento do consumo de energia na residência e como meio de comunicação entre o aplicativo mobile e o sistema de *hardware* instalado na residência.

A ideia central é que o *hardware* envie, a cada intervalo de tempo predefinido, as informações necessárias para gerar o monitoramento do gasto de energia e, ao mesmo tempo, faça a leitura de *tags* presentes no banco de dados *online* para fazer o controle da iluminação.

2.2.2.1 Firebase

O Firebase é uma plataforma de banco de dados em tempo real desenvolvido pelo Google com o objetivo de integralizar o desenvolvimento do *backend* de aplicações mobile e *web* (FIREBASE, 2021). Constituído de um conjunto de produtos como, serviços de hospedagem, armazenamento em nuvem e banco de dados, o Firebase é gratuito com um certo limite de utilização. Para o trabalho em questão o banco de dados permite um total de 100 acessos simultâneos sem a cobrança da utilização. O grande diferencial dessa plataforma é permitir a inclusão de bancos de dados em projetos *web*, Android e iOS sem o conhecimento de como uma rede de bancos de dados funciona e com poucas linhas de código (UEDA, 2020).

2.2.2.2 Google Sheets

O Google Sheets é uma ferramenta do Google que funciona basicamente como o Microsoft Excel e de forma totalmente online e gratuita. Sozinha essa ferramenta não funciona como um banco de dados, porém ao usá-la em conjunto com outra ferramenta do Google, isso é possível.

A ferramenta em questão se trata do Apps Script que de acordo com Google(2021):

O Apps Script é a única plataforma com poucos códigos que facilita o desenvolvimento de soluções empresariais para integrar, automatizar e ampliar os recursos no Google Workspace. Com o Apps Script, os usuários empresariais podem desenvolver soluções personalizadas no Google Workspace, mesmo sem experiência profissional em desenvolvimento. Esse recurso está disponível para todas as pessoas que têm uma conta do Gmail.

Nessa ferramenta é possível criar um aplicativo *web* usando a linguagem HTML onde que possa receber dados por meio do método doPost do protocolo HTTP e assim salvar esses dados em lugares específicos do Google Sheets transformando-o em um banco de dados.

2.2.3 Ambiente de desenvolvimento integrado do Arduino

O Ambiente de desenvolvimento integrado, comumente conhecido como IDE do Arduino é um ambiente de desenvolvimento integrado onde o usuário tem acesso a formatação automática de sintaxe, correção de erros, inclusão de bibliotecas e vários recursos que o auxiliam a escrever códigos, além do carregamento de códigos diretamente na placa de desenvolvimento.

Apesar de ser inicialmente projetada para a programação de microcontroladores da família Arduino, com algumas alterações, é possível incluir as configurações necessárias para tornar possível a programação do microcontrolador ESP32 presente no Módulo WiFi ESP32 Bluetooth. Portanto, a escolha da IDE para a programação do *hardware* é ideal devido a ampla gama de conteúdo relacionado e à familiaridade com a plataforma.

2.3 Sistema similar

Buscando no mercado brasileiro uma solução similar à proposta neste trabalho para comparação, encontrou-se a Houseasy é uma *startup* brasileira sediada em Curitiba e que desde 2019 cria produtos para facilitar o acesso a automatização de ambientes a partir do conceito de IoT. Atualmente a empresa fornece 3 produtos que trabalham em conjunto para automatizar uma residência.

O produto Easy Mind faz papel da central na residência e é responsável por comunicar com os outros dois dispositivos que serão apresentados a seguir. Para fazê-lo funcionar basta liga-lo a uma tomada e configurar uma rede Wi-Fi nele. Para o controle de iluminação é usado o Easy Power, inserido dentro do orifício na parede que comporta interruptores ou tomadas onde a automatização é desejada. Assim o Easy Power é capaz de controlar até dois circuitos. Por fim tem-se o Easy Play, para substituir controles remotos. Esse aparelho usa comunicação infravermelho para automatizar televisões, ar-condicionados, sistemas de som, persianas elétricas e quaisquer outros aparelhos que são ativados com esse tipo de comunicação.

Para controlar todos os aparelhos a Houseasy disponibiliza um aplicativo para download nas versões Android e IOS. Com o aplicativo o usuário pode configurar todos os produtos da Houseasy e configurar ações e controles para cada aparelho ou circuito conectado.

Atualmente a Houseasy fornece esses produtos de forma individual e em pacotes com base na quantidade de cômodos que a pessoa deseja automatizar, como o trabalho pretende construir um sistema de acessível os valores dos produtos fornecidos pela *startup* podem ser conferidos na Tabela 1 para comparação.

Produto	Itens contidos no produto	Preço
EASY MIND	1 EASY MIND	R\$199,00
EASY POWER	1 EASY POWER	R\$69,90
EASY PLAY	1 EASY PLAY	R\$89,90
Kit Easy - 1 Cômodo	1 EASY MIND, 1 EASY	R\$237.90
	POWER, 1 EASY PLAY	
Kit Easy - 4 Cômodos	1 EASY MIND, 4 EASY	R\$539.90
	POWER, 2 EASY PLAY	1)
Kit Easy - 8 Cômodos	1 EASY MIND, 8 EASY	R\$ 722.90
	POWER, 4 EASY PLAY	

Tabela 1 Preços dos produtos oferecidos pela Houseasy Fonte: Autoria própria

3 DESENVOLVIMENTO

Nesse capítulo são apresentadas as etapas que foram exigidas para a construção desse trabalho. O principal objetivo desse trabalho foi desenvolver uma plataforma de controle de iluminação e monitoramento de consumo energético de uma residência que pudesse ser facilmente instalado pelo usuário com pouco ou nenhum conhecimento de eletrônica e programação. Além disso foi levado em consideração o custo total do projeto de forma que componentes baratos e que atendiam aos requisitos do projeto, assim como plataformas gratuitas, foram priorizados.

Dessa forma o aplicativo desenvolvido durante esse trabalho foi construído na plataforma online gratuita *Kodular*, que por meio de programação de blocos permitiu criar um aplicativo agradável ao usuário. O sistema físico foi construído com base no modulo ESP32 que possui um microcontrolador robusto o suficiente para suportar o código implementado por meio da IDE do *Arduino*, além de já possuir as tecnologias Wi-Fi e Bluetooth acopladas. A comunicação entre o aplicativo e a central física foi feita por meio do Google Sheets em conjunto com a ferramenta Apps Script.

A seguir serão detalhados todos os passos para a elaboração desse trabalho.

3.1 Aplicativo SisLux

Antes de começar a programação do aplicativo foi necessário a criação do *Wireframe* do mesmo. O *Wireframe* consiste do layout inicial do aplicativo, ou seja, uma representação linear da estrutura básica do aplicativo focada na interface do usuário. A sua construção tem como objetivo planejar as ações do usuário durante a utilização do aplicativo e, a partir desse planejamento se inicia a construção das funções que realmente dão vida ao aplicativo mobile. O *Wireframe* desenvolvido pose ser observado na Figura 3.1. Nele podemos ver as telas básicas da aplicação e como cada tela se comunica com as outras.

Para a construção do *Wireframe* foi utilizada a plataforma Fluid UI, onde é possível criar um layout incial de cada página do seu aplicativo e fazer as ligações de como as páginas se conectam.



Figura 3.1 *Wireframe* do aplicativo Fonte: Autoria própria

A partir desse planejamento se deu início a construção do aplicativo. A tela inicial do aplicativo consiste de uma animação de carregamento para o usuário mostrado na Figura 3.2. Enquanto essa animação mostra ao usuário que algo está acontecendo o aplicativo atualiza seus dados a partir do banco de dados construído, caso algum dispositivo esteja configurado na memória do aplicativo armazenada no celular do usuário.



Figura 3.2 Tela inicial do aplicativo SisLux Fonte: Autoria própria

Caso o usuário já tenha configurado algum dispositivo, o aplicativo fica nessa tela por alguns segundos antes de seguir para a próxima tela. Caso nenhum dispositivo tenha sido configurado, essa tela é exibida e quase que imediatamente a próxima tela é apresentada.

O menu principal do aplicativo é onde o usuário passará a maior parte do tempo durante a utilização do mesmo. Ela consiste de um *layout* em forma de *swap*, onde o usuário pode alternar entre as abas "Grupos", "Energia" e "Configurações" arrastando o dedo sobre a tela. A tela que se abre primeiro para o usuário é a tela que contém a aba "Grupos" representada na Figura 3.3.

Os grupos presentes na aba "Grupos" são agrupamentos de luzes que devem se acender simultaneamente por meio do comando do usuário. Por exemplo, se no grupo "Sala", mostrado na Figura 3.3, for composto de cinco pontos de iluminação, os cinco pontos devem acender ou apagar ao mesmo tempo.



Figura 3.3 Aba "Grupos" do menu Fonte: Autoria própria

Nessa aba o usuário consegue controlar a iluminação dos grupos configurados, ativando ou desativando as luzes ao pressionar o nome do grupo e conferindo o estado das luzes a partir dos ícones ao lado do nome dos respectivos grupos.

Por trás do *frontend*, o aplicativo, ao detectar que um grupo foi pressionado, envia uma solicitação através do protocolo de comunicação HTTP ao banco de dados utilizando o método doPost e aguarda o recebimento da confirmação da alteração do estado da iluminação. Paralelamente à ação do usuário, a cada determinado período de tempo o aplicativo faz uma requisição por conta própria para se certificar sobre os estados dos grupos. Dessa forma se o estado de algum grupo foi alterado pelos interruptores físicos isso é demonstrado na aba "Grupos" do aplicativo.

Ao arrastar a tela para o lado é apresentado ao usuário a aba "Energia" representada na Figura 3.4. Essa aba consiste de um *layout* interno e externo, ambos em forma de *swap*. O *layout* externo já foi apresentado e serve para o usuário navegar entre as abas "Grupos", "Energia" e "Configurações". Já o *layout* interno tem como função permitir ao usuário navegar entre as abas que correspondem aos meses do ano, e dessa forma, conseguir acompanhar o consumo de energia de sua residência mês a mês durante todo um ano.



Figura 3.4 Aba "Energia" do menu Fonte: Autoria própria

A aba "Energia" possui um sistema de atualização em conjunto com a aba "Grupos", ou seja, a mesma função que a cada determinado período de tempo faz as requisições para conferir e atualizar os estados dos grupos atualiza os valores de consumo e gastos na aba "Energia".

Por fim tem-se a aba "Configurações" mostrada na Figura 3.5, onde o usuário pode configurar o dispositivo físico para acessar a sua rede Wi-Fi, configurar o nome dos grupos e quais grupos devem aparecer na aba "Grupos" e configurar o valor da tarifa de energia elétrica.



Figura 3.5 Aba "Configurações" do menu Fonte: Autoria própria

Para configurar um dispositivo físico o usuário deve primeiramente verificar se a localização do seu celular está ativada e em seguida pode ligar ou não o Bluetooth. Esse segundo passo é opcional pois caso o Bluetooth estiver desabilitado o próprio aplicativo consegue ativar o mesmo.

Ao pressionar o botão "Configurar dispositivos" uma mensagem é apresentada ao usuário mostrando que o aplicativo está escaneando os dispositivos Bluetooth disponíveis e que não estão pareados ao celular, assim como mostra a Figura 3.6.



Figura 3.6 Mensagem mostrada ao usuário enquanto procura por dispositivos Bluetooth Fonte: Autoria própria

Ao encontrar os dispositivos que se encaixam nas descrições mencionadas, o aplicativo mostra ao usuário uma lista de dispositivos assim como na Figura 3.7. A seguir o usuário deve pressionar o nome do dispositivo que ele deseja configurar, lembrando que o aplicativo somente consegue configurar dispositivos correspondentes, ou seja, dispositivos SisLux.



Figura 3.7 Lista de dispositivos encontrados Fonte: Autoria própria

Após a seleção, o aplicativo pede permissão para parear com o dispositivo e ao confirmar o pareamento duas mensagens são exibidas ao usuário. Primeiramente uma mensagem onde é necessário digitar o nome da rede em que se deseja conectar assim como mostrado na Figura 3.8. Logo após essa mensagem outra é mostrada ao usuário, dessa vez é necessário inserir a senha da rede Wi-Fi. Esse segundo passo é representado na Figura 3.9.



Figura 3.8 Mensagem para inserir nome da rede Fonte: Autoria própria



Figura 3.9 Mensagem para inserir senha da rede Fonte: Autoria própria

Após inserir tanto o nome quanto a senha da rede, uma mensagem é montada com as duas informações e é enviada ao dispositivo físico via Bluetooth. O aplicativo, ao receber a confirmação que o dispositivo foi configurado, exibe uma mensagem de confirmação é exibida. Essa confirmação é o número de série do dispositivo, que é usado para acessar as informações corretas no banco de dados.

Caso o usuário esteja na aba "Configurações", representada pela Figura 3.5, e deseje configurar a aba "Grupos", ele deve pressionar o botão "Configurar grupos". Assim, uma tela de configuração mostrada na Figura 3.10 será apresentada.

4				
Grupo 1	Nome grupo 1		On	
Grupo 2	Nome grupo 2		On	
Grupo 3	Nome grupo 3		On	
Grupo 4	Nome grupo 4		On	
Grupo 5	Nome grupo 5		On	
Grupo 6	Nome grupo 6		On	
Grupo 7	Nome grupo 7		On	
Grupo 8	Nome grupo 8		On	
		۲	•	

Figura 3.10 Tela da configuração da aba "Grupos" Fonte: Autoria própria

Nessa tela é possível modificar os nomes dos grupos, assim como definir quais grupos estarão visíveis na aba "Grupos", para confirmar as modificações o usuário deve simplesmente pressionar a seta de retorno e as modificações estarão salvas.

Voltando à aba "Configurações", ainda é possível alterar a tarifa paga pelo usuário à concessionária de energia. Ao pressionar o botão "Configurar tarifa" uma caixa de texto é apresentada ao usuário, onde é possível digitar o valor da tarifa. Ao confirmar a modificação os valores pagos pela energia consumida são alterados. Essa ação está representada na Figura 3.11.


Figura 3.11 Mensagem de configuração da tarifa de energia Fonte: Autoria própria

A programação em blocos do aplicativo SisLux pode ser consultada por completo nos Apêndices A a G desse trabalho.

3.2 Banco de dados

O banco de dados foi criado a partir de uma planilha do *Google* Sheets. A partir da planilha criada foi identificado o identificador da planilha de acordo com a Figura 3.12. Esse identificador serve para selecionar e manipular a planilha pelo programa criado no *Apps Script*. $G = G = MONOGRAFIA_Desenvolviment \times MONOGRAFIA_Monitoramento \times G = Google Sheets and Apps Script \times G = C = C = MONOGRAFIA_Desenvolviment / ISSV9p-c7rqRaM1S7JYgpHbwMC5zSSJKFt8MW3sJjrc8jedit#gid=0$

Figura 3.12 Identificador da planilha

Fonte: Autoria própria

Com o identificador em mãos foram criados dois programas, o primeiro contém uma função que aceita o método de comunicação doPost e o segundo uma função que aceita o método doGet. A criação de dois programas foi necessária pois as funções de utilização de HTTP no aplicativo usam de forma nativa o método doPost enquanto a programação do dispositivo físico utiliza o método doGet. Uma vez que uma implementação não consegue acessar os dois métodos de forma simultânea a criação de dois códigos foi justificada. A programação do código utilizado para o aplicativo pode ser encontrado no Apêndice H,

S- C 0/7

enquanto a programação utilizada para a comunicação do dispositivo físico pode ser encontrada no Apêndice I.

3.3 Dispositivo físico SisLux

Para a construção do dispositivo físico primeiramente foram listados os componentes necessários para a concretização de todos os requisitos do sistema. Como o módulo escolhido para controlar o sistema foi o ESP32, cuja alimentação no pino de entrada pode variar entre 4,5 e 9 Volts, uma fonte de protoboard correspondente foi escolhida. O motivo de escolher uma fonte de protoboard foi sua facilidade de montagem em uma placa ilhada. A fonte escolhida também foi capaz de alimentar o módulo de relés e desde que o ESP32 e o módulo de relés estejam conectados no mesmo terminal neutro é possível o controle do módulo.

Para as conexões com os relés correspondentes no módulo, as conexões dos interruptores, do sensor de corrente, assim como a alimentação do sensor e do próprio módulo, foram usados conectores do tipo KRE por sua facilidade de ser montado em uma placa ilhada assim como para organizar a placa de modo correto.

Com a listagem de componente foi desenhado um diagrama no *software* EasyEDA para servir de guia durante a montagem da placa, representado pela Figura 3.13.

Os módulos relés e o sensor de corrente não foram representados no diagrama devido ao fato que foram usados componentes prontos e conectados externamente a placa que foi montada. Por esse motivo uma legenda foi colocada em cada barramento para facilitar a identificação.



Figura 3.13 Diagrama de montagem do dispositivo físico Fonte: Autoria própria

A programação do circuito foi realizada no ambiente IDE do *Arduino*. Como a princípio o projeto se mostrou mais complexo do que o planejado foi necessário a montagem de um fluxograma para definir os passos a serem tomados durante a execução do *script* pelo ESP32. O fluxograma citado é apresentado na Figura 3.14.



Figura 3.14 Fluxograma do funcionamento básico do dispositivo físico Fonte: Autoria própria

A comunicação com o banco de dados é feita através de uma conexão Wi-Fi. Porém essa comunicação não pode ser feita de forma imediata pois a rede da casa onde o dispositivo vai ser instalado precisa ser configurada primeiro. Para isso ao usar o aplicativo desenvolvido

38

e seguir os passos citados anteriormente é possível enviar o nome da rede e a senha por Bluetooth. Após receber essas informações o dispositivo envia um sinal de confirmação, reiniciando e voltando ao início do fluxograma da Figura 3.14.

Com o dispositivo configurado o programa consegue seguir para sua sequência principal, onde a comunicação com o banco de dados é habilitada e as demais funções do dispositivo são desbloqueadas. Isso não exclui a capacidade de ser reconfigurado, uma vez que a função onde as mensagens via Bluetooth são recebidas, rodam de forma paralela às funções principais.

Como o dispositivo precisa de alguns segundos entre o envio da requisição ao banco de dados e o recebimento das informações, foi necessário a utilização dos dois núcleos do microcontrolador de forma separada. Para isso foram criadas duas tarefas, cada uma rodando exclusivamente em um núcleo. Dessa forma a tarefa responsável por ler as interações externas não foi prejudicada pelo tempo em espera causado pela tarefa responsável pela atualização dos parâmetros que controlam a iluminação.

A primeira tarefa (tarefa 1) fica responsável por ler os interruptores físicos da residência, para isso é usada uma biblioteca que trata cada interruptor declarado como uma função, identifica o se o interruptor é pressionado e trata a informação para, após esses passos, gerar uma requisição que muda o valor do estado da iluminação no banco de dados. Ao receber a confirmação que o banco de dados foi corretamente alterado a iluminação correspondente é ativada ou desativada com base no estado anterior do banco de dados.

Sendo também responsável pela leitura do sensor de corrente, a tarefa 1 faz uso de uma biblioteca que indica a corrente com base em uma quantidade de leituras pré-definidas para chegar a um valor médio e reduzir as possíveis flutuações. Esse valor é armazenado em uma variável que é multiplicada pela tensão da rede, padronizada em 127 Volts. A seguir a função faz uma requisição ao banco de dados para obter a potência consumida anteriormente e ao receber o valor soma a potência consumida atual e faz outra requisição para atualizar o valor no banco de dados.

Essas operações no banco de dados são feitas nos espaços correspondentes aos meses do ano que são obtidos por meio da data fornecida pela própria rede. Isso possibilita o sistema ser instalado em diversos lugares com diferentes fusos horários sem grandes complicações para a aquisição de dados. Como essa função em específico executa dois acessos ao banco de dados, a sua chamada de forma constante poderia atrapalhar a leitura dos sensores, por isso, o valor da potência consumida é salvo a cada segundo em um vetor e somente a cada cinco minutos a comunicação com o banco de dados é feita.

A tarefa 2 é responsável por atualizar os parâmetros e identificar mudanças que foram feitas no banco de dados a partir do aplicativo, por isso, a cada segundo uma função que faz uma requisição que percorre todos os campos do banco de dados é chamada e todos os parâmetros são atualizados. Dessa forma se alguma iluminação foi desativada por meio do aplicativo, a mudança é feita no sistema físico.

Como a tarefa 1 é executada em um núcleo diferente da tarefa 2, em momento algum elas disputam tempo computacional e, dessa forma, o tempo de espera constante gerado pelas requisições seguidas da tarefa 2 não afetam o desempenho da tarefa 1.

Paralelamente a esses processamentos principais a função loop é executada em qualquer núcleo, onde estiver disponível. Esse método não diminui a forma de execução da função loop pois sua única função é perceber que uma mensagem foi recebida por Bluetooth e tratar a mensagem de forma adequada. Não ser prioridade na hora de ser processada não é determinante, pois as mensagens recebidas ficam armazenadas em um *buffer* interno até serem lidas.

A programação completa do dispositivo físico pode ser vista no Apêndice I.

4 RESULTADOS

A construção do sistema teve início na programação do aplicativo e na criação do banco de dados, a execução desses dois passos assim como seus testes foram feitas de forma simultânea tendo em vista que o funcionamento de um dependia completamente do funcionamento do outro.

Os primeiros testes envolvendo essa primeira parte tiveram como objetivo se certificar que quando um comando era executado pelo usuário o espaço dedicado para aquele comando no banco de dados tinha seu valor alterado corretamente. Com o funcionamento correto dessa parte foi possível montar quase todas as telas do aplicativo.

O próximo passo envolveu a montagem de um circuito em uma *protoboard* para testar a programação do sistema físico assim como a replicação do programa que possibilita o acesso do aplicativo ao banco de dados para permitir o acesso do dispositivo físico ao banco de dados. Algumas programações tiveram que ser feitas no programa do banco de dados, pois além de algumas funções de comunicação serem diferentes o aplicativo e o dispositivo usam métodos de comunicação diferentes.

Ao fazer os testes e garantir que o dispositivo físico também alterava os espaços correspondentes aos comandos executados, foi implementado a comunicação Bluetooth entre o aplicativo e o dispositivo. Diversos métodos foram testados e pôr fim a implementação que obteve mais sucesso envolveu a utilização do Bluetooth clássico no lugar do BLE.

Com todas as funções tanto do aplicativo quanto do dispositivo físico funcionando corretamente foi montado um circuito definitivo em uma placa de ilhas assim como ilustrado na Figura 4.1. Para os testes com o dispositivo montado foram utilizados LEDs para simular os grupos de lâmpadas, botões para simular os interruptores físicos e uma lâmpada ligada continuamente para simular o consumo de energia da residência.



Figura 4.1 Montagem do circuito definitivo Fonte: Autoria própria

Com o circuito montado os testes foram repetidos, porém durante essa segunda bateria de testes o sistema e o aplicativo foram usados como se um usuário estivesse utilizando os mesmos. O primeiro passo da utilização do sistema é a configuração da rede Wi-Fi no dispositivo físico esse passo pode ser observado no aplicativo nas Figura 4.2Figura 4.8 e na resposta do dispositivo físico na Figura 4.9.







Figura 4.3 Configuração do dispositivo físico, visão do aplicativo, parte 2. Procurando por dispositivos Bluetooth







Figura 4.5 Configuração do dispositivo físico, visão do aplicativo, parte 4. Pareamento com o dispositivo Bluetooth



Figura 4.6 Configuração do dispositivo físico, visão do aplicativo, parte 5. Fornecimento do nome da rede



Figura 4.7 Configuração do dispositivo físico, visão do aplicativo, parte 6. Fornecimento da senha da rede



Figura 4.8 Configuração do dispositivo físico, visão do aplicativo, parte 7. Confirmação da configuração





Fonte: Autoria própria

A seguir foi aguardado um período de cinco minutos com o dispositivo funcionando e durante esse tempo foram observadas as leituras da corrente. O tempo de espera de cinco minutos foi o necessário para o dispositivo completar a primeira bateria de leituras e modificar o valor no banco de dados. O início das leituras pode ser observado na Figura 4.10. Após os

cinco minutos os valores das leituras guardados foram somados e convertidos para Quilowattshora e atualizados no banco de dados. Essa etapa pode ser conferida na Figura 4.11. Com a atualização de dados verificamos no aplicativo se a mudança foi detectada. A tela do aplicativo que permitiu o usuário ver a modificação está representada na Figura 4.12.



Figura 4.10 Início das leituras de corrente



Figura 4.11 Atualização do consumo no banco de dados

Fonte: Autoria própria



Figura 4.12 Tela do aplicativo mostrando para o usuário a modificação do consumo Fonte: Autoria própria

Seguindo os testes, a tela de grupos do aplicativo foi modificada nas configurações (Figura 4.13) para atender aos testes, a modificação foi verificada na tela "Grupos", Figura 4.14. Em seguida o grupo 1 (Sala) foi pressionado e então o banco de dados foi verificado para observar se a mudança tinha sido feita no lugar certo. A verificação pode ser observada na

Figura 4.15. A alteração do estado no banco de dados também pode ser verificada pela tela do aplicativo onde a lâmpada de estado correspondente ao grupo pressionado muda de aparência (Figura 4.16). Simultaneamente o LED que representava o grupo pressionado estava sendo observado e o mesmo se acendeu conforme o desejado. O mesmo teste foi executado para o grupo 2 (Cozinha).

<u> </u>			
Grupo 1	Sala	On	
Grupo 2		On	-
Grupo 3	Nome grupo 3	Off	•
Grupo 4	Nome grupo 4	Off	
Grupo 5	Nome grupo 5	Off	0
Grupo 6	Nome grupo 6	Off	0
Grupo 7	Nome grupo 7	Off	0
Grupo 8	Nome grupo 8	Off	
		•	

Figura 4.13 Tela de modificação dos grupos no aplicativo

Fonte: Autoria própria



Figura 4.14 Tela dos grupos após modificação

	🗖 👗 Meu Dive-Google Dive x 👔 🕮 TCC-Raviba Google x 0 (0) Whatdapp x + +																		
C 🖓 🕼 C https://docs.google.com/typresdtheets/d/155V3p-27rgBaM157/lgpHowM25s/51RBMW33jRd/edl#gid=0)												
🔥 C.BRNANL - Googl. 🚱 Conchytoli - Asiste. 👌 (DFFCIAL) GeepVid. 📕 Mithad.com 📀 Vage PEOSEANA. 👌 Pio, Glide, Fight an., by Game Of Thrones 👔 Asistris Game Of Thrones 👔 Cubick Geosci 🛞 RepLit - Chine Java.							fevoritos												
	E TCC 🖈 🖸 🗠 Arguino Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda Aultima.edição foi feita há.3 minutos por VICTOR DE ABREU MERCON 🛹 🗏 🚺 di Compartilibrar 😢								V										
k.	い 여름 한 100% + R8 % ④ 401 123+ Padric (AL + 10 + B I 용 A 今田田田・吉・上・片・ヴィ GD 田田マ F・ Σ・								23										
AA14	• <i>fx</i>																		-
	A	В	С	D		E	F		6	н	<.> Z		AA	AB					
1	1		0	0	0	()	0	0		0	0	0	0					-
2		7																	a
4																			Ŭ
5	Valor alt	erado no h	anco de da	idos anós															•
6	pression	ar o grupo	1	aco apos															•
8																			•
9																			× .
10																			
11																De			
12																			+
14																			
15																			
16																			
17																			
19																			
20																			
21																			
22																			
23																			
25																			
26																			
27																			
28																			
30																			
31																		- -	
	+ ≣ Pa	ágina1 💌																	>
					n 🖬		(; N	•	🙏 🖩 🔌	0	🧿 🛧 (i 🥺 🧕	a 🦛 🖪 f	5	∧ POR . PTB	ල දා ක 16/0	16:33 01/2022

Figura 4.15 Banco de dados após alteração do estado do grupo



Figura 4.16 Tela dos grupos com a lâmpada que indica o estado do grupo 1 foi alterada Fonte: Autoria própria

De forma similar ao teste anterior o dispositivo físico foi testado, porém dessa vez o grupo que foi testado primeiro foi o grupo dois. Ainda com o dispositivo funcionado o botão que simula o interruptor do grupo 2 foi pressionado e o dispositivo físico identificou corretamente o botão pressionado e fez a mudança no banco de dados. A resposta do dispositivo

e a modificação do banco de dados pode ser verificada na Figura 4.17. Assim como no teste anterior a modificação do banco de dados pode ser visualizada na tela do aplicativo assim como na Figura 4.18. Como era esperado o LED que representava o grupo 2 foi ativado e desativado de acordo com o comando.



Figura 4.17 Confirmação do interruptor do grupo dois sendo pressionado e alteração no banco de dados.



Fonte: Autoria própria

Figura 4.18 Tela dos grupos com a lâmpada que indica o estado do grupo 2 foi alterada

Após os testes feitos e a confirmação de que o aplicativo estava funcionando corretamente em conjunto com o dispositivo físico, os dois componentes foram apresentados para outras pessoas que não tinham conhecimento do projeto. Explicações básicas foram dadas a respeito do funcionamento do sistema e a utilização do aplicativo e do dispositivo físico foi observada. Esse teste mostrou que o sistema de forma geral estava intuitivo o suficiente para um usuário leigo no assunto utilizar o mesmo.

Todos os códigos utilizados para a construção do projeto podem ser encontrados no seguinte link https://github.com/victor-mercon/Projeto-SisLux.git

Como um dos objetivos do trabalho era desenvolver um sistema de acessível foram pesquisados preços de mercado e foi obtida uma média. Os valores médios para a construção do sistema, assim como o custo total do sistema pode ser visto na Tabela 2. Os valores apresentados na tabela foram obtidos em dezembro de 2021.

Tabela 2 Valores gastos para a montagem do dispositivo físico

DESCRIÇÃO	QUANTIDADE	VALOR UNITARIO	VALOR TOTAL		
MÓDULO WIFI ESP32 BLUETOOTH 30 PINOS	1	R\$ 74,90	R\$ 74,90		
MÓDULO SENSOR DE CORRENTE 30A	1	R\$ 29,90	R\$ 29,90		
MÓDULO RELÉ 8 CANAIS 5V COM OPTOACOPLADOR	1	R\$ 59,90	R\$ 59,90		
FONTE AJUSTÁVEL PARA PROTOBOARD	1	R\$ 12,90	R\$ 12,90		
FONTE 9V 1A BIVOLT	1	R\$ 18,90	R\$ 18,90		
PLACA FENOLITE PERFURADA 10X10CM	1	R\$ 14,90	R\$ 14,90		
CABO FLAT COLORIDO 20 VIAS 26AWG	1	R\$ 17,90	R\$ 17,90		
LED DIFUSO 5MM	3	R\$ 0,30	R\$ 0,90		
CONECTOR BORNE KRE 2 VIAS	16	R\$ 1,90	R\$ 30,40		
TOTAL					

Fonte: Autoria própria

Em comparação ao sistema comercial pesquisado a solução proposta nesse trabalho apresenta uma modularidade que permite a adição de um módulo relê maior ou uma configuração de dois ou mais módulos sem aumentar o custo de forma considerável, uma vez que o sistema se encontra pronto e o a investimento extra seria na aquisição do módulo e na adequação do código do sistema físico e do aplicativo. O sistema comercial por sua vez tem a sua instalação facilitada uma vez que não é necessário modificar a fiação e os interruptores, além disso possui a função de controle de aparelhos que recebem sinal infravermelho.

Se compararmos os custos das duas alternativas para o controle exclusivo de iluminação, o sistema desenvolvido nesse trabalho é capaz de controlar até 8 grupos de lâmpadas a um custo de aproximadamente R\$260,00 enquanto para controlar os mesmos 8 grupos a solução comercial requer um investimento de pelo menos R\$722,90.

5 CONCLUSÃO

Este trabalho teve como objetivo a criação de uma central de controle de iluminação para os diversos ambientes de uma residência em conjunto com o monitoramento em tempo real do consumo energético do local integrado a um aplicativo mobile.

Em relação ao aplicativo desenvolvido foi colocado em foco a simplicidade que o usuário teria ao interagir com o mesmo, para isso, o aplicativo foi criado com aperas 3 telas principais de fácil navegação entre si e poucas funções. Além disso o conforto visual do usuário foi levado em conta e as telas do aplicativo contém um número baixo de objetos por tela, justamente para evitar a confusão do usuário.

A forma que o dispositivo físico foi construído permite ao usuário instalar na sua residência com poucas modificações na sua rede de iluminação. Ao ser instalado próximo ao padrão de energia da residência e as fiações correspondentes as luzes forem levadas até ele e os interruptores gangorra que são padrões para instalação forem substituídos por interruptores pulsantes o sistema fica pronto e esperando a configuração via aplicativo para funcionar.

Uma vez que o aplicativo não tem custo para o usuário e o sistema físico teve um custo de R\$260,60 para a construção do protótipo o custo total do desenvolvimento do protótipo se iguala ao custo do dispositivo físico. Por se tratar de um protótipo não se pode garantir que o preço de um possível produto derivado desse trabalho seja igual a esse, devido a novos recursos ou funcionalidades o valor pode aumentar ou ainda diminuir se as devidas otimizações forem aplicadas.

Com base nas informações apresentadas no capítulo anterior e no início desse capítulo podese concluir que o trabalho proposto foi executado com sucesso.

O sistema desenvolvido permite ao usuário monitorar o consumo energético de sua residência e controlar os grupos de lâmpadas de qualquer lugar desde que seu dispositivo móvel esteja conectado à internet.

Esse projeto mostra que com criatividade e dedicação é possível criar um sistema para automatizar uma residência sem custos elevados, necessários em sistemas de automatização residencial disponíveis no mercado.

O trabalho proposto é um protótipo e existem muitos aspectos que podem ser melhorados em trabalhos futuros. Por exemplo, expansores de portas podem ser utilizados para fazer o controle de mais grupos de iluminação permitindo o controle de casas maiores com implementações de usuários e localidade no aplicativo ou ainda o controle de condomínios de apartamentos onde cada usuário consegue controlar a iluminação de seu apartamento. Sensores de tensão também podem ser implantados para aumentar a precisão do cálculo da potência consumida e dispensar alterações no programa do dispositivo físico quando o sistema for instalado em locais diferentes tensões de rede.

Circuitos capazes de controlar a iluminação através de dimerização podem ser incluídos no dispositivo físico e programados no aplicativo para dar ao usuário mais controle e mais conforto ao utilizar os cômodos de sua residência.

Com a implementação da dimerização pode-se implementar também a criação de perfis de cenários para um cômodo, para que esse apresente um conforto maior dependendo da sua utilização ou do momento em que está sendo utilizado.

No dispositivo físico também é possível implementar interrupções para ler os interruptores para garantir uma resposta mais assertiva do dispositivo físico, pois assim ele não depende da varredura do *software*.

REFERÊNCIAS

ALLEGRO MICROSYSTEMS. Fully Integrated, Hall Effect-Based Linear Current Sensor IC. Massachusetts: Allegro MicroSystems, 2015.

BRAGA, N. C. Newton C. Braga, 2021. Disponivel em:

https://www.newtoncbraga.com.br/index.php/eletronica/52-artigos-diversos/12719-isolamento-com-acopladores-opticos-art2870. Acesso em: 14 nov. 2021.

COELHO, D. F. B.; CRUZ, V. H. D. N. **Edifícios Inteligentes:** uma visão das tecnologias aplicadas. Sao Paulo: Editora Blucher, 2017.

COELHO, Í. Qual módulo WiFi ESP32 é ideal para meu projeto?, 2020. Disponivel em: https://www.filipeflop.com/blog/qual-modulo-wifi-esp32-e-ideal-para-meu-projeto/. Acesso em: 25 agosto 2021.

CUNHA, L. Relés e Contatores. O Setor Elétrico, v. 45, p. 54-60, 2009.

ELETRÔNICA, A. &. **Arduino & Eletrônica**, 2021. Disponivel em: https://www.arduinoeeletronica.com.br/produto/modulo-rele-8-canais-5v/. Acesso em: 14 nov. 2021.

FIREBASE. **Firebase**, 2021. Disponivel em: https://firebase.google.com/docs/. Acesso em: 14 nov. 2021.

GOOGLE. **google**, 2021. Disponivel em: https://workspace.google.com/intl/pt-BR/products/apps-script/. Acesso em: 14 nov. 2021.

HINRICHS, R. A.; KLEINBACH, M. Energia e meio ambiente. 5^a. ed. São Paulo: Cengage Learning Brasil, 2014.

JR, S. L. S.; FARINELLI, F. A. **Domótica -Automação residencial e casas inteligentes com Arduino e ESP8626**. São Paulo: Editora Saraiva, 2018.

KODULAR. **Kodular**, 2021. Disponivel em: <https://docs.kodular.io/pricing/#commissions>. Acesso em: 14 nov. 2021.

MACÊDO, G. P. Medidor resiencial inteligente de energia életrica. [S.l.]. 2016.

MOTA, A. **Portal vida de silicio**, 2017. Disponivel em: <https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona/>. Acesso em: 13 nov. 2021.

MURATORI, J. A. Os desafios do mercado da Automação Residencial, 2013. Disponivel em: https://www.aecweb.com.br/cont/a/os-desafios-do-mercado-da-automacao-residencial_8192. Acesso em: 1 julho 2021.

PHILIPS. **PCF8574 remote 8-bit I/O expander for I2C bus**. Philips Semiconductors. [S.l.], p. 24. 2002.

PROCEL. Percentual de Consumo na Carga Residencial, 2021. Disponivel em: <https://app.powerbi.com/view?r=eyJrIjoiNmE3YjY2ZjctM2NkMi00MTYwLWE1OTktYW NmNTNkNDI1YWNkIiwidCI6IjhhMGZmYjU0LTk3MTYtNGE5My05MTU4LTIIM2E3Mj A2ZjE4ZSJ9>. Acesso em: 07 julho 2021.

PRUDENTE, F. Automação Predial e Residencial - Uma Introdução. São Paulo: Grupo GEN, 2011.

SOUZA, F. **Embarcados**, 2013. Disponivel em: https://www.embarcados.com.br/arduino-uno/#Alimentacao-da-placa-Arduino. Acesso em: 13 nov. 2021.

UEDA, D. Y. **APLICAÇÃO ANDROID COM FIREBASE PARA ADMINISTRAR O USO DE MEDICAÇÕES**. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. PATO BRANCO, p. 41. 2020.

ZANATTA, B. Casa 'Inteligente' é cada vez mais realidade. **O Estado de São Paulo**, v. 46478, Janeiro 2021.

APÊNDICE A – TELA INICIAL DO APLICATIVO SISLUX

initialize global (instante) to [0]

Figura A.1 Declaração da variável global instante na tela inicial Fonte: Autoria própria

initialize global URL to C * https://script.google.com/macros/s/AKfycbxs3lkG9...

Figura A.2 Declaração da variável global URL na tela inicial Fonte: Autoria própria



Figura A.3 Programação da inicialização da tela inicial Fonte: Autoria própria



Figura A.4 Programação da função Consumo_mensal







Figura A.6 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela incial parte 1 Fonte: Autoria própria



Figura A.7 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP parte 2 Fonte: Autoria própria

APÊNDICE B – TELA MENU DO APLICATIVO SISLUX (INICIALIZAÇÃO)

initialize global URL to thttps://script.google.com/macros/s/AKfycbxs3lkG9...



Figura B.1 Declaração da variável global URL na tela menu

Figura B.2 Programação da inicialização da tela menu

	to Inicializa_status			
do	set Status_G1 •	. Picture 🔹 to 🌘	call Tiny_DB1 • .Get Value	
			tag 🖡	" status_G1 "
			value If Tag Not There	" 🔲 "
	set Status_G2 •	. Picture 🔹 to 👔	call Tiny_DB1 • .Get Value	
			tag 🖡	" status_G2 "
			value If Tag Not There	" 🔲 "
	set Status_G3 •	.Picture 🔹 to 🛿	call Tiny_DB1 • .Get Value	
			tag 🌔	" status_G3 "
			value If Tag Not There	" 🔲 "
	set Status_G4 •	.Picture 🔹 to 🌘	call Tiny_DB1 • .Get Value	
			tag 🌔	" status_G4 "
			value If Tag Not There	* 🔲 *
	set Status_G5 •	. Picture 🔹 to 🌘	call Tiny_DB1 • .Get Value	
			tag 🌔	" status_G5 "
			value If Tag Not There	* 🔲 *
	set Status_G6 •	.Picture 🔹 to 🌘	call Tiny_DB1 • .Get Value	
			tag 🖡	" status_G6 "
			value If Tag Not There	" 🔲 "
	set Status_G7 •	. Picture 🔹 to 🌘	call Tiny_DB1 • .Get Value	
			tag 🌔	" status_G7 "
			value If Tag Not There	" 🔲 "
	set Status_G8 •	. Picture 🔹 to 🕼	call Tiny_DB1 • .Get Value	
			tag 🖡	" status_G8 "
			value If Tag Not There	" 🔲 "
	set Status_G9 •	. Picture 🔹 to 👔	call Tiny_DB1 • .Get Value	
			tag 🌔	" status_G9 "
			value If Tag Not There	"●"
	set Status_G10 •	. Picture 🔹 to	call Tiny_DB1 Get Value	
			tag	status_G10
	L		value If Iag Not There	
	set Status_G11 •	. Picture To	Call Tiny_DB1 Cet Value	
			tag	status_G11
			value If Tag Not There	
	set Status_G12 •	Picture to	Call Tiny_DB1 .Get Value	Catature Odda
			tag	status_G12
	set Status_G13	. Picture • to	Call Tiny_DB1 .Get Value	Later Oto I
			lag	status_G13
	set Status_G14 •	. Picture • to	Call Tiny_DB1 .Get value	Cotatua C14
			tag valuo lf Tag Not These	status_014
	set Statue G15	Picture to	value in ray Not There	
	Set Status_015		tan miy_bbr . Oet value	estatus G15 "
			value If Tag Not There	
	set Statue G16 -	Picture - to	call (Tiny DB1) Cot Value	
	Set Status_010		tag	f status G16
			value If Tag Not There	

Figura B.3 Programação da função Inicializa_status



Figura B.4 programação da função adiciona_componentes



Figura B.5 Programação da função inicializa_grupos



Figura B.6 Programação da função inicializa_consumo

APÊNDICE C – TELA MENU DO APLICATIVO SISLUX (ATUALIZAÇÃO DOS DADOS)



Figura C.1 Programação do timer que controla a frequência de chamada de uma função



Figura C.2 Programação da função Consumo_mensal



Figura C.3 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 1


Figura C.4 Programação para recuperar os dados obtidos através do protocolo de comunicação HTTP na tela menu parte 2

APÊNDICE D – TELA MENU DO APLICATIVO SISLUX (CONTROLE DOS GRUPOS)





whe	when G2 Click						
do	😂 if	C call (Finy_DB1	.Get	Value		
					tag u Disp *		
			value If T	ag Not	There of false		
		set Web	🔹 . URL	🔹 to	get (global URL -		
		call Web	Post	Text			
				text (call Web1 • .Build Request Data		
					list 🕴 😧 make a list 🕴 😧 make a list 🕴 🗧 apiCommand		
					setCellValue		
					🕐 make a list 🚺 🕺 celladress *		
					call (<u>Tiny_DB1 →</u> .Get Value		
					value if Tag Not There		
					🔿 make a list 👘 newalue i		
					Compare texts of Status_G2 • . Picture • E • 0 • (lampadaoff.png •		
					then a * 🖸 *		
		call Notif	eri 🔹 .Sh	iow Aleri			
				nouce			





Figura D.3 Programação do controle do grupo 3

whe	n G4	Click
do	🛛 if	call Tiny_DB1 • . Get Value
		value If Tag Not There + faise -
		set (Web1 • . URL • to t get (global URL •)
		call Web1 . Post Text
		text (call Web1 • .Build Request Data
		list 👖 😢 make a list 👖 🙂 make a list 👘 🕐 make a list 👘
		setCellValue)
		🖞 🙂 make a list 📮 🕻 celladress 🕯
		n an
		la call <u>Tiny_DB1 →</u> .Get Value
		value If Tag Not There +
		🕐 🕐 make a list 📬 "neuvalue"
		🔽 👘 🗘 compare texts 🕴 Status_64 🔹 🤉 Picture 🔹 🖅 🚛 Ampadaoff.png *
		cali Notifieri • Show Alert
		notice Configure o dispositivo primeiro





Figura D.5 Programação do controle do grupo 5



Figura D.6 Programação do controle do grupo 6

whe	n G7	.Click					
do	🖸 if	C call Tin	y_DB1 ▼ .Ge	t Value			
				tag ([*] Disp *			
		v	alue If Tag Not	There [false •			
		set Web1 🔹	. URL 🔹 to	get global URL 🔹			
		call Web1 •	.Post Text		_		
			text (call Web1 .Build Request Data			
				list	😟 make a list (🔉 😳 make a list 🖡	apiCommand)
							setCellValue
						🔉 🖸 make a list 🖡	celladress "
							join (G "
							call Tiny_DB1 . Get Value
							tag (<mark>"Disp</mark> "
							value If Tag Not There () *
						🔉 🗊 make a list	newvalue
						1	f compare texts (Status_G7 -). Picture -) E (* lampadaoff.png *
							then t 10 "
							else t "O"
		call Notifier1	.Show Ale				
			notic	Conligure o dispositivo primeiro			





Figura D.8 Programação do controle do grupo 8

APÊNDICE E – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DO DISPOSITIVO FÍSICO)







Figura E.2 Programação que indica o que fazer quando o escaneamento por novos dispositivos for concluído



Figura E.3 Programação do evento que é acionado após o usuário escolher um dispositivo



Figura E.4 Programação do evento que é acionado após o celular parear com um dispositivo



Figura E.5 Programação de evento disparado após o usuário inserir dados sobre a rede Wi-Fi

APÊNDICE F – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DOS GRUPOS)



Figura F.1 Programação da configuração do nome do grupo 1



Figura F.2 Programação da configuração que defina a visibilidade do grupo 1 ao usuário



Figura F.3 Programação da configuração do nome do grupo 2



Figura F.4 Programação da configuração que defina a visibilidade do grupo 2 ao usuário



Figura F.5 Programação da configuração do nome do grupo 3



Figura F.6 Programação da configuração que defina a visibilidade do grupo 3 ao usuário

whe	n nome_grupo_4 .Lost Focus	
do	call Tiny_DB1 .Store Value	
	tag 🌘	" NG4 "
	value To Store 🌘	nome_grupo_4 🔹 . Text 🔹

Figura F.7 Programação da configuração do nome do grupo 4



Figura F.8 Programação da configuração que defina a visibilidade do grupo 4 ao usuário

whe	n [nome_grupo_5 🔻] .Lost Focus	
do	call Tiny_DB1 .Store Value	
	tag 🌗	" NG5 "
	value To Store 🌘	nome_grupo_5 🔹 . Text 🔹

Figura F.9 Programação da configuração do nome do grupo 5

whe	n Switch_grupo_5 . Clicked		
is	Checked		
do	call Tiny_DB1 . Store Value		
	tag		" <u>SG5</u> "
	value To Store	l	(Switch_grupo_5 •). (is Checked •)

Figura F.10 Programação da configuração que defina a visibilidade do grupo 5 ao usuário









whe	n nome_grupo_7 .Lost Focus	
do	call Tiny_DB1 .Store Value	
	tag 🏮	" NG7 "
	value To Store 🌘	nome_grupo_7 🔹 . Text 🔹

Figura F.13 Programação da configuração do nome do grupo 7



Figura F.14 Programação da configuração que defina a visibilidade do grupo 7 ao usuário

whe	n [nome_grupo_8 🔹 .Lost Focus	
do	call Tiny_DB1 .Store Value	
	tag 🌔	" NG8 "
	value To Store 🌘	nome_grupo_8 🔹 . Text 🔹

Figura F.15 Programação da configuração do nome do grupo 8

_				
	when	Switch grupo 8 . Clicked		
	IS C	hecked		
6	do 🔒	call Tiny DB1 Store Value		
		tag	SG8	
		value To Store	Switch an	inc
		Value TO Store	 Switch_gro	up

Figura F.16 Programação da configuração que defina a visibilidade do grupo 8 ao usuário

APÊNDICE G – TELA MENU DO APLICATIVO SISLUX (CONFIGURAÇÃO DA TARIFA)



Figura G.1 Programação da configuração da tarifa de energia



Figura G.2 Programação de evento disparado após o usuário inserir dados sobre a tarifa de energia

APÊNDICE H – PROGRAMAÇÃO APPS SCRIPT (CÓDIGO APLICATIVO)

function doPost(data) {

// o objeto 'data' contem strings do seu post request

// define a planilha.

var sheet = SpreadsheetApp.openById('15SV9pc7rqRaM1S7JYgpHbwMC5zSSJKFt8MW3sJjRc8').getSheets()[0];

// metodo openById() descrito aqui:

 $//https://developers.google.com/apps-script/reference/spreadsheet/\ spreadsheet-app#openbyidid$

// metodo getSheets()[0] descrito:

//https://developers.google.com/apps-

script/reference/spreadsheet/spreadsheet#getsheets

//metodo analogo getSheetByName(name) descrito aqui: //https://developers.google.com/appsscript/reference/spreadsheet/spreadsheet#getsheetbynamename

// define um comando que voce quer executar.(string value)
var apiCommand = data.parameter.apiCommand;

```
// define uma string parao para resposta.
var sendBack = "comando api nao valido";
```

// "if" sessao para diferentes comandos

```
if (apiCommand == "setCellValue") {
```

```
var cell = sheet.getRange(data.parameter.celladress);
var col = cell.getColumn();
cell.setValue(data.parameter.newvalue);
```

```
sendBack =col.getA1Notation() + '/' + data.parameter.newvalue;
};
```

if (apiCommand == "getCellValue") {

```
var cell = sheet.getRange(data.parameter.celladress);
var col = cell.getColumn();
var cellvalue = cell.getValue();
```

```
sendBack = col.getA1Notation() + '/' + cellvalue;
};
// end "if" section
```

// "sendBack" e uma string, que retorna para nosso app
// more info https://developers.google.com/apps-script/guides/content
return ContentService.createTextOutput(sendBack);
}

APÊNDICE I – PROGRAMAÇÃO APPS SCRIPT (CÓDIGO ESP32)

```
var sheet = SpreadsheetApp.openById('15SV9p-
c7rqRaM1S7JYgpHbwMC5zSSJKFt8MW3sJjRc8').getSheets()[0];
function doGet(data){
     // define um comando que voce quer executar.(string value)
 var apiCommand = data.parameter.apiCommand;
    // define uma string parao para resposta.
 var sendBack = "comando api nao valido";
// "if" sessao para diferentes comandos
 if (apiCommand == "setCellValue") {
   var cell = sheet.getRange(data.parameter.celladress);
   cell.setValue(data.parameter.newvalue);
   sendBack = data.parameter.newvalue;
  };
 if (apiCommand == "getCellValue") {
   var cell = sheet.getRange(data.parameter.celladress);
   var cellvalue = cell.getValue();
   sendBack = cellvalue;
  };
 if (apiCommand == "scanCellValue") {
   var cell1 = sheet.getRange(data.parameter.celladress1);
   var cell2 = sheet.getRange(data.parameter.celladress2);
   var cell3 = sheet.getRange(data.parameter.celladress3);
   var cell4 = sheet.getRange(data.parameter.celladress4);
   var cell5 = sheet.getRange(data.parameter.celladress5);
   var cell6 = sheet.getRange(data.parameter.celladress6);
   var cell7 = sheet.getRange(data.parameter.celladress7);
   var cell8 = sheet.getRange(data.parameter.celladress8);
   var cellvalue1 = cell1.getValue();
   var cellvalue2 = cell2.getValue();
   var cellvalue3 = cell3.getValue();
   var cellvalue4 = cell4.getValue();
   var cellvalue5 = cell5.getValue();
   var cellvalue6 = cell6.getValue();
   var cellvalue7 = cell7.getValue();
   var cellvalue8 = cell8.getValue();
```

sendBack = cellvalue1 + "," + cellvalue2 + "," + cellvalue3 + "," + cellvalue4 + "," + cellvalue5 + "," + cellvalue6 + "," + cellvalue7 + "," + cellvalue8;

};
// end "if" section

```
// "sendBack" e uma string, que retorna para nosso app
// more info https://developers.google.com/apps-script/guides/content
return ContentService.createTextOutput(sendBack);
}
```

APÊNDICE J – PROGRAMAÇÃO ESP32

#include <button2.h></button2.h>	//biblioteca para controlar os botoes
#include <wifi.h></wifi.h>	//biblioteca para habilitar wi-fi
<pre>#include <httpclient.h></httpclient.h></pre>	//biblioteca para usar protocolo HTTP
#include "EmonLib.h"	//biblioteca para fazer a leitura do sensor de corrente
#include "time.h"	//biblioteca para definir data com informações da rede
<pre>#include "BluetoothSerial.h"</pre>	//biblioteca para habilitar o bluetooth
<pre>#include <preferences.h></preferences.h></pre>	//biblioteca para gravação de dados na memoria flash

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

#define VOLTAGE_AC 127.00	// 127 Volts
#define ACS_MPY 15.41	// ganho/calibracao da corrente
#define BUTTON_1_PIN 13	// pino ligado ao botão que controla o grupo 1
#define BUTTON_2_PIN 12	// pino ligado ao botão que controla o grupo 2
#define BUTTON_3_PIN 14	// pino ligado ao botão que controla o grupo 3
#define BUTTON_4_PIN 27	// pino ligado ao botão que controla o grupo 4
#define BUTTON_5_PIN 26	// pino ligado ao botão que controla o grupo 5
#define BUTTON_6_PIN 25	// pino ligado ao botão que controla o grupo 6
#define BUTTON_7_PIN 33	// pino ligado ao botão que controla o grupo 7
#define BUTTON_8_PIN 32	// pino ligado ao botão que controla o grupo 8
#define SENSOR_PIN 34	// pino ligado ao sensor de corrente
#define G1 15	// pino ligado ao rele que ativa/desativa o grupo 1
#define G2 2	// pino ligado ao rele que ativa/desativa o grupo 2
#define G3 4	// pino ligado ao rele que ativa/desativa o grupo 3
#define G4 16	// pino ligado ao rele que ativa/desativa o grupo 4
#define G5 17	// pino ligado ao rele que ativa/desativa o grupo 5
#define G6 5	// pino ligado ao rele que ativa/desativa o grupo 6
#define G7 18	// pino ligado ao rele que ativa/desativa o grupo 7
#define G8 19	// pino ligado ao rele que ativa/desativa o grupo 8
#define LedWiFi 22	// pino ligado ao led que indica comunicação via wifi
#define LedBluetooth 23	// pino ligado ao led que indica comunicação bluetooth
#define SeriesNumber "1"	// numero sequencial que indentifica o dispositivo

BluetoothSerial SerialBT;	//cria objeto bluetooth
Preferences preferences;	//cria instancia de gravação na memoria flash

const char* ntpServer = "pool.ntp.org"; const long gmtOffset_sec = 3600; const int daylightOffset_sec = 3600; //server ntp para recuperação da data

EnergyMonitor emon1; //criação de objeto para leitura do sensor de corrente

//variaveis utilizadas para o calculo de potencia

int array_Potencias[300]; int aux_potencias = 0; double Irms = 0; int mes; String mes_aux; double kwh; double kwh_atual;

//variaveis para conecção wi-fi

String ssid_pass; String ssid; String pass; int aux_wifi = 0; String GOOGLE_SCRIPT_ID = "AKfycbxm2TTT_XWXUSXdSRqAqcXUs5P1geKrwCT1wcv_hOnKXhsnrZA6YFgjXfkAvylvDFX";

//declaração dos objetos para controlar os botoes

Button2 button1, button2, button3, button4, button5, button6, button7, button8;

//declaração do objeto wi-fi

WiFiClientSecure client;

//declaração dos estados dos grupos

int state 1 = 0;

int state2 = 0; int state3 = 0; int state4 = 0; int state5 = 0; int state6 = 0; int state7 = 0; int state8 = 0;

//variaveis para o controle de tempo do acesso das funções

```
unsigned long startMillis1;
unsigned long currentMillis1;
const unsigned long period1 = 1000; //o valor e um numero em milisegundos
unsigned long startMillis2;
unsigned long currentMillis2;
const unsigned long period2 = 1000; //o valor e um numero em milisegundos
unsigned long startMillis3;
unsigned long currentMillis3;
const unsigned long period3 = 300000; //o valor e um numero em milisegundos
```

//definição das tasks que serão rodadas

TaskHandle_t Task1; TaskHandle_t Task2;

void setup() {
 startMillis1 = millis(); //tempo inicial 1
 startMillis2 = millis(); //tempo inicial 2
 startMillis3 = millis(); //tempo inicial 3

Serial.begin(115200); delay(50); SerialBT.begin("SisLux1"); //nome do dispositvo Bluetooth

preferences.begin("credentials", false);//inicializa o namespace onde as variaveis vão ser salvas na flash

ssid = preferences.getString("ssid", "");//carrega o valor da variavel ssid
pass = preferences.getString("password", "");//carrega o valor da variavel password

//defio o modo dos pinos usados
pinMode(G1, OUTPUT);
pinMode(G2, OUTPUT);
pinMode(G3, OUTPUT);

```
pinMode(G4, OUTPUT);
pinMode(G5, OUTPUT);
pinMode(G6, OUTPUT);
pinMode(G7, OUTPUT);
pinMode(G8, OUTPUT);
pinMode(LedWiFi, OUTPUT);
pinMode(LedBluetooth, OUTPUT);
button1.begin(BUTTON_1_PIN);
button1.setClickHandler(click);
button2.begin(BUTTON_2_PIN);
button2.setClickHandler(click);
button3.begin(BUTTON_3_PIN);
button3.setClickHandler(click);
button4.begin(BUTTON_4_PIN);
button4.setClickHandler(click);
button5.begin(BUTTON_5_PIN);
button5.setClickHandler(click);
button6.begin(BUTTON_6_PIN);
button6.setClickHandler(click);
button7.begin(BUTTON 7 PIN);
button7.setClickHandler(click);
button8.begin(BUTTON_8_PIN);
button8.setClickHandler(click);
//inicializa o objeto que irá ler o sensor de corrente
emon1.current(SENSOR_PIN, ACS_MPY);
                                                // Corrente: pino analógico, calibracao.
if (ssid == "" || pass == "") {
 Serial.println("sem valores salvos para ssid ou password");
}
else {
 WiFi.mode(WIFI STA);
 WiFi.begin(ssid.c_str(), pass.c_str());
 Serial.println("Connecting");
 while (WiFi.status() != WL_CONNECTED) {
  digitalWrite(LedWiFi, LOW);
  delay(250);
  Serial.print(".");
  digitalWrite(LedWiFi, HIGH);
  delay(250);
```

```
Serial.println("");
```

Serial.print("Connected to WiFi network with IP Address: "); Serial.println(WiFi.localIP());

//inicia e captura o tempo
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

ScanPost();

//cria a task que será exaecutada na função Task1code(), com prioridade 1 e execução co core 0

```
xTaskCreatePinnedToCore(
Task1code, /* função task */
"Task1", /* nome da task. */
10000, /* tamanho de Stack da task */
NULL, /* parametro da task */
10, /* prioridade da task */
&Task1, /* Task handle para monitorar a task criada */
0); /* vincula task ao core 0 */
delay(500);
```

//cria a task que será exaecutada na função Task2code(), com prioridade 1 e execução co core 1

xTaskCreatePinnedToCore(

```
Task2code, /* função task. */
  "Task2". /* nome da task. */
            /* tamanho de Stack da task */
  10000.
              /* parametro da task */
  NULL,
           /* prioridade da task */
  10,
              /* Task handle para monitorar a task criada */
  &Task2.
          /* vincula task ao core 0 */
  1);
 delay(500);
 if (aux wifi == 30) {
  Serial.println("SSID ou Senha errados");
Serial.println(WiFi.localIP());
}
```

}

//Task1code: chama a função para fazer leitura dos botões, define o mes atual e chama a função para fazer a leitura do sensor de corrente.

```
void Task1code( void * pvParameters ) {
   Serial.print("Task1 running on core ");
   Serial.println(xPortGetCoreID());
```

for (;;) {
 button1.loop();
 button2.loop();

```
button3.loop();
button4.loop();
button5.loop();
button6.loop();
button7.loop();
button8.loop();
struct tm timeinfo;
if (!getLocalTime(&timeinfo)) {
 Serial.println("Failed to obtain time");
 return;
}
mes = timeinfo.tm_mon;
switch (mes) {
 case 0:
  mes_aux = "Q";
  break;
 case 1:
  mes_aux = "R";
  break;
 case 2:
  mes_aux = "S";
  break;
 case 3:
  mes_aux = "T";
  break;
 case 4:
  mes_aux = "U";
  break;
 case 5:
  mes_aux = "V";
  break;
 case 6:
  mes_aux = "W";
  break;
 case 7:
  mes_aux = "X";
  break;
 case 8:
  mes_aux = "Y";
  break;
 case 9:
  mes_aux = "Z";
  break;
 case 10:
  mes_aux = "AA";
  break;
 case 11:
  mes_aux = "AB";
  break;
```

```
}
currentMillis2 = millis();
if (currentMillis2 - startMillis2 >= period2)
{
    startMillis2 = currentMillis2;
    scanCorrente();
    }
}
```

```
//Task2code: busca informações no banco de dados para atualizar o estados dos grupos
void Task2code( void * pvParameters ) {
    Serial.print("Task2 running on core ");
    Serial.println(xPortGetCoreID());
    for (;;) {
        currentMillis1 = millis();
        if (currentMillis1 - startMillis1 >= period1)
```

```
{

ScanPost();

startMillis1 = currentMillis1;

}

}
```

```
//loop: chama a função para ler a comunicação bluetooth se tiver alguma mensagem a ser lida
void loop() {
    if (SerialBT.available()) {
        btRead();
    }
    disableCore0WDT();
```

}

//btRead: le as informações passadas por blueto
oth e salva as informações na memoria flash void btRead() {

```
if (SerialBT.available()>0) {
```

```
ssid_pass = "";
```

ssid_pass = SerialBT.readStringUntil('/'); Serial.println(ssid_pass);

```
for (int i = 0; i < ssid_pass.length(); i++)
  Serial.println("Entrou for");
  if (ssid_pass.substring(i, i + 1) == ",")
   ssid = ssid_pass.substring(0, i);
   pass = ssid_pass.substring(i + 1);
   Serial.print("SSID = "); Serial.println(ssid);
   Serial.print("Password = "); Serial.println(pass);
   break;
  }
 }
  preferences.putString("ssid", ssid);
  preferences.putString("password", pass);
  Serial.println("*******");
  Serial.print("ssid_pass = ");
  Serial.println(ssid_pass); // Presenta ssid_pass.
  Serial.println("*******"):
  Serial.print("ssid = ");
  Serial.println(preferences.getString("ssid", "")); // Presenta ssid_pass.
  Serial.println("*******");
  Serial.print("pass = ");
  Serial.println(preferences.getString("password", "")); // Presenta ssid_pass.
  ESP.restart();
}
```

}

if(state3==0){

```
//click: indentifica qual botão foi pressionado e chama a função para alterar seu estado no
banco de dados
void click(Button2& btn) {
 if (btn == button1) {
  Serial.println("Grupo 1 clicked");
  if(state1==0){
   post("A",SeriesNumber,"1");
  }else{
   post("A",SeriesNumber,"0");
  }
 } else if (btn == button2) {
  Serial.println("Grupo 2 clicked");
  if(state2==0){
   post("B",SeriesNumber,"1");
  }else{
   post("B",SeriesNumber,"0");
  }
 else if (btn == button3) 
  Serial.println("Grupo 3 clicked");
```

```
post("C",SeriesNumber,"1");
  }else{
   post("C",SeriesNumber,"0");
 }else if (btn == button4) {
  Serial.println("Grupo 4 clicked");
  if(state4==0){
   post("D",SeriesNumber,"1");
  }else{
   post("D",SeriesNumber,"0");
  }
 }else if (btn == button5) {
  Serial.println("Grupo 5 clicked");
  if(state5==0){
   post("E",SeriesNumber,"1");
  }else{
   post("E",SeriesNumber,"0");
  }
 } else if (btn == button6) {
  Serial.println("Grupo 6 clicked");
  if(state6==0){
   post("F",SeriesNumber,"1");
  }else{
   post("F",SeriesNumber,"0");
  }
 else if (btn == button7) 
  Serial.println("Grupo 7 clicked");
  if(state7==0){
   post("G",SeriesNumber,"1");
  }else{
   post("G",SeriesNumber,"0");
  }
 }else if (btn == button8) {
  Serial.println("Grupo 8 clicked");
  if(state8==0){
   post("H",SeriesNumber,"1");
  }else{
   post("H",SeriesNumber,"0");
  }
}
}
```

//scanCorrente: faz a leitura da corrente e busca o ultimo consumo salvo no banco de dados e atualiza o consumo atual void scanCorrente() { Irms = emon1.calcIrms(1996); //Calcula IRMS Serial.println(Irms); array_Potencias[aux_potencias] = Irms * VOLTAGE_AC;

```
aux_potencias++;
 for (int i = 0; i < 300; i++) {
  Serial.print(array Potencias[i]);
  Serial.print(",");
 }
 Serial.println(" ");
 currentMillis3 = millis();
 if (currentMillis3 - startMillis3 >= period3) {
  kwh = 0;
  if (WiFi.status() == WL CONNECTED) {
   HTTPClient http;
   String url = "https://script.google.com/macros/s/" + GOOGLE_SCRIPT_ID +
"/exec?apiCommand=getCellValue&celladress=" + mes_aux + SeriesNumber;
   Serial.println("Making a request");
   led wifi blink();
   http.begin(url.c_str()); //Specify the URL and certificate
   http.setFollowRedirects(HTTPC STRICT FOLLOW REDIRECTS);
   int httpCode = http.GET();
   String payload;
   if (httpCode > 0) \{ //checa pelo codigo de retorno 
    payload = http.getString();
    kwh_atual = payload.toInt();
   } else {
    Serial.println("Error on HTTP request");
   }
   http.end();
  for (int i = 0; i < 300; i++) {
   array_Potencias[i] = array_Potencias[i]/1000;
  for (int i = 0; i < 300; i++) {
   kwh = kwh +array Potencias[i];
  kwh_atual = kwh_atual + kwh;
  if (WiFi.status() == WL_CONNECTED) {
   HTTPClient http;
   String url = "https://script.google.com/macros/s/" + GOOGLE_SCRIPT_ID +
"/exec?apiCommand=setCellValue&celladress=" + mes_aux + SeriesNumber +
"&newvalue=" + kwh_atual;
   Serial.println("Making a request");
   led_wifi_blink();
   http.begin(url.c_str()); //Specify the URL and certificate
   http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
   int httpCode = http.GET();
   String payload;
   if (httpCode > 0) { //Checa pelo codigo de retorno
    payload = http.getString();
    kwh_atual = payload.toInt();
   } else {
    Serial.println("Error on HTTP request");
```

```
}
http.end();
}
for (int i = 0; i < 300; i++) {
    array_Potencias[i] = 0;
}
aux_potencias = 0;
startMillis3 = currentMillis3;
}
</pre>
```

```
//post: atualiza no banco de dados o estados dos grupos e envia o comando correspondente a
partir do codigo de retorno para o rele correspondente
void post(String letra, String num, String Value) {
 if (WiFi.status() == WL CONNECTED) {
  HTTPClient http;
  String url = "https://script.google.com/macros/s/" + GOOGLE_SCRIPT_ID +
"/exec?apiCommand=setCellValue&celladress=" + letra + num + "&newvalue=" + Value;
  // Serial.print(url);
  Serial.println("Making a request");
  led_wifi_blink();
  http.begin(url.c str()); //especifica o url e o certificado
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
  int httpCode = http.GET();
  String payload;
  if (httpCode > 0) { //Checa pelo codigo de retorno
   payload = http.getString();
   Serial.println(httpCode);
   Serial.println(payload);
   if (letra == "A") {
     state1 = payload.toInt();
     digitalWrite(G1, (state1) ? HIGH : LOW);
   } else if (letra == "B") {
     state2 = payload.toInt();
     digitalWrite(G2, (state2) ? HIGH : LOW);
   } else if (letra == "C") {
     state3 = payload.toInt();
     digitalWrite(G3, (state3) ? HIGH : LOW);
   } else if (letra == "D") {
```

```
state4 = payload.toInt();
    digitalWrite(G4, (state4) ? HIGH : LOW);
   } else if (letra == "E") {
    state5 = payload.toInt();
    digitalWrite(G5, (state5) ? HIGH : LOW);
   } else if (letra == "F") {
    state6 = payload.toInt();
    digitalWrite(G6, (state6) ? HIGH : LOW);
   } else if (letra == "G") {
    state7 = payload.toInt();
    digitalWrite(G7, (state7) ? HIGH : LOW);
   } else if (letra == "H") {
    state8 = payload.toInt();
    digitalWrite(G8, (state8) ? HIGH : LOW);
   }
  }
  else {
   Serial.println("Error on HTTP request");
  }
  http.end();
 }
}
//ScanPost: busca informações no banco de dados para atualizar o estados dos grupos durante
a função setup
void ScanPost() {
 if (WiFi.status() == WL_CONNECTED) {
  HTTPClient http;
  String url = "https://script.google.com/macros/s/" + GOOGLE_SCRIPT_ID +
"/exec?apiCommand=scanCellValue&celladress1=A" + SeriesNumber + "&celladress2=B" +
SeriesNumber + "&celladress3=C" + SeriesNumber + "&celladress4=D" + SeriesNumber +
"&celladress5=E" + SeriesNumber + "&celladress6=F" + SeriesNumber + "&celladress7=G"
+ SeriesNumber + "&celladress8=H" + SeriesNumber;
```

```
// Serial.print(url);
Serial.println("Making a request");
led_wifi_blink();
```

http.begin(url.c_str()); //especifica a url e o certificado http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS); int httpCode = http.GET(); String payload; if (httpCode > 0) { //Checa pelo codigo de retorno payload = http.getString();

Serial.println(httpCode); Serial.println(payload); Serial.println(payload.charAt(0)); Serial.println(payload.charAt(2)); Serial.println(payload.charAt(4)); Serial.println(payload.charAt(6)); Serial.println(payload.charAt(8)); Serial.println(payload.charAt(10)); Serial.println(payload.charAt(12)); Serial.println(payload.charAt(12));

state1 = payload.charAt(0) - '0'; digitalWrite(G1, (state1) ? HIGH : LOW);

state2 = payload.charAt(2) - '0'; digitalWrite(G2, (state2) ? HIGH : LOW);

```
state3 = payload.charAt(4) - '0';
digitalWrite(G3, (state3) ? HIGH : LOW);
```

state4 = payload.charAt(6) - '0'; digitalWrite(G4, (state4) ? HIGH : LOW);

state5 = payload.charAt(8) - '0'; digitalWrite(G5, (state5) ? HIGH : LOW);

```
state6 = payload.charAt(10) - '0';
digitalWrite(G6, (state6) ? HIGH : LOW);
```

state7 = payload.charAt(12) - '0'; digitalWrite(G7, (state7) ? HIGH : LOW);

```
state8 = payload.charAt(14) - '0';
digitalWrite(G8, (state8) ? HIGH : LOW);
```

```
} else {
   Serial.println("Error on HTTP request");
}
http.end();
}
```

}

```
//led_wifi_blink: busca controla o led para indentificar comunicação via wifi
void led_wifi_blink(){
    int state=0;
    for(int i=0;i<10;i++){
        digitalWrite(LedWiFi, (state) ? HIGH : LOW);
        state != state;
        delay(100);
    }
}</pre>
```

```
//led_bluetooth_blink: busca controla o led para indentificar comunicação via bluetooth
void led_bluetooth_blink(){
    int state=0;
    for(int i=0;i<10;i++){
        digitalWrite(LedBluetooth, (state) ? HIGH : LOW);
        state != state;
        delay(100);
    }
}
```