



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
CURSO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA

CAELLISON BEZERRA MORAIS

**TECNOLOGIA EM CASA: APLICATIVO MÓVEL INTEGRADO AO
ARDUINO NA AUTOMAÇÃO RESIDENCIAL**

Caraúbas

2024

CAELLISON BEZERRA MORAIS

**TECNOLOGIA EM CASA: APLICATIVO MÓVEL INTEGRADO AO
ARDUINO NA AUTOMAÇÃO RESIDENCIAL**

Trabalho de conclusão de curso apresentado a
Universidade Federal Rural do Semi-Árido,
como requisito para a obtenção do título de
Bacharel em Ciência e Tecnologia.

Orientador: Prof. Dr. Zenner Silva Pereira

Caraúbas

2024

©Todos os direitos estão reservados à Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996, e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata, exceto as pesquisas que estejam vinculadas ao processo de patenteamento. Esta investigação será base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) seja devidamente citado e mencionado os seus créditos bibliográficos.

MM828 Moraes, Caellison Bezerra.
t Tecnologia em Casa: Aplicativo Móvel Integrado
 ao Arduino na Automação Residencial / Caellison
 Bezerra Moraes. - 2024.
 52 f.: il.

Orientador: Zenner Silva Pereira.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2024.

1. automação residencial. 2. aplicativo móvel.
3. microcontrolador arduino. I. Silva Pereira,
Zenner, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com
AACR2 e os dados fornecidos pelo(a) autor(a).
Biblioteca Campus Caraúbas
Bibliotecária: Dalvanira Brito Rodrigues
CRB: 15/700

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

CAELLISON BEZERRA MORAIS

**TECNOLOGIA EM CASA: APLICATIVO MÓVEL INTEGRADO AO
ARDUINO NA AUTOMAÇÃO RESIDENCIAL**

Trabalho de conclusão de curso
apresentado a Universidade Federal Rural do
Semi-Árido como requisito para obtenção do
título de Bacharel em Ciência e Tecnologia.

Defendido em: 17 / 04 / 2024.

BANCA EXAMINADORA



Documento assinado digitalmente
ZENNER SILVA PEREIRA
Data: 19/04/2024 20:23:55-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Zenner Silva Pereira (UFERSA)
Presidente



Documento assinado digitalmente
ANA TEREZA DE ABREU LIMA
Data: 20/04/2024 08:07:23-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dra. Ana Tereza de Abreu Lima (UFERSA)
Membro Examinador



Documento assinado digitalmente
JOSE JUNIOR ALVES DA SILVA
Data: 24/04/2024 11:25:00-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Jose Junior Alves da Silva (UFERSA)
Membro Examinador

LISTA DE FIGURAS

Figura 1 - Tipos de Placas Arduino.	18
Figura 2 - IDE do Arduino.	19
Figura 3- Interface de Designer do App Inventor	21
Figura 4 - Interface de programação App Inventor	23
Figura 5 - Espelhamento via QR Code do App em tempo real.	25
Figura 6 - Compilando app Android.	26
Figura 7 - Three Way convencional com dois Interruptores Paralelo.	28
Figura 8 - Three Way com Interruptor Paralelo com relé.	28
Figura 9 - Arduino Mega 2560.	30
Figura 10 - Fios e Jumpers.	30
Figura 11 - Resistores.	31
Figura 12 - Conexões da Protoboard.	32
Figura 13 - Conexão do Módulo Bluetooth HC-06.	33
Figura 14 – Interruptor Paralelo	33
Figura 15 - Interruptor físico em paralelo com o relé.	34
Figura 16 - Módulo Dimmer MC-8A.	35
Figura 17 - Forma de Onda da rede AC.	36
Figura 18- Controle de tempo em 0 V	36
Figura 19 - Lâmpada	37
Figura 20 - Interface da Tela Inicial/Login do App.	39
Figura 21 - Programação em Blocos da Tela de Login/Inicial.	40
Figura 22 - Interface da Tela do Three-Way do App.	40
Figura 23 – Blocos da tela do Three Way	41
Figura 24 - Blocos da tela do Three Way	41
Figura 25 - Blocos da tela do Three Way	41
Figura 26 - Blocos da tela do Three Way	42
Figura 27 - Blocos da tela do Three Way	42
Figura 28 - Blocos da tela do Three Way	43
Figura 29 - Interface da Tela de Controle de Brilho do App.	43
Figura 30 - Blocos da tela do Controle de Brilho.	44
Figura 31 - Blocos da tela do Controle de Brilho.	44
Figura 32 - Blocos da tela do Controle de Brilho.	44

Figura 33 - Blocos da tela do Controle de Brilho.	45
Figura 34 - Blocos da tela do Controle de Brilho.	45
Figura 35 - Blocos da tela do Controle de Brilho.	46
Figura 36 - Blocos da tela do Controle de Brilho.	47
Figura 37 - Blocos da tela do Controle de Brilho.	47
Figura 38 - Parte I do código completo.....	48
Figura 39 - Parte II do código completo.	49
Figura 40 - Parte III do código completo.	50
Figura 41 - Parte final do código completo.....	51
Figura 42 - Circuito para o Sistema Three-Way	52
Figura 43 - Circuito para o Controle de Brilho da Lâmpada	52
Figura 44 - Circuito Completo Three-Way e Dimmer	53

LISTA DE ABREVIATURAS E SIGLAS

IDE	Ambiente de Desenvolvimento Integrado
LED	Diodo Emissor de Luz
VCC	Tensão em Corrente Contínua
GND	Filtro Graduado de Densidade Neutra
QR CODE	Código de Resposta Rápida
MIT	Instituto de Tecnologia de Massachusetts
APP	Aplicativo
AC	Corrente Alternada
TRIAC	Tríodo de Corrente Alternada
ZC	Passagem pelo Zero
BPS	Bits por Segundo
IA	Inteligência Artificial
PCB	Placa de Circuito Impresso
STRING	Cadeia de Caracteres

Sumário

1.	INTRODUÇÃO	14
1.1	MOTIVAÇÃO	14
1.2	PROBLEMAS	14
1.3	OBJETIVO	15
2.	REFERÊNCIAL TEÓRICO.....	16
2.1	AUTOMAÇÃO RESIDENCIAL.....	16
2.1.1	Evolução Histórica da Automação Residencial	16
2.1.2	Importância e benefícios da automação residencial	17
2.2	ARDUINO	17
2.2.1	Origem do Arduino	18
2.2.2	Placas do Arduino	18
2.2.3	Arduino - IDE	19
2.3	MIT APP INVENTOR.....	21
2.4	BLUETOOTH.....	26
2.5	COMUNICAÇÃO SERIAL	27
2.6	THREE WAY.....	27
3.	METODOLOGIA.....	29
3.1	MATERIAIS.....	29
3.1.1	Arduino mega 2560	29
3.1.2	Fios e <i>Jumpers</i>	30
3.1.3	Resistores	31
3.1.4	Protoboard	31
3.1.5	Módulo Bluetooth HC-06	32
3.1.6	Interruptor Paralelo	33
3.1.7	Módulo Relé	34
3.1.8	Módulo <i>Dimmer</i> MC-8A	35

3.1.9	Lâmpada	37
3.2	MÉTODOS	38
4.	RESULTADO E DISCUSSÃO	39
4.1.1	Desenvolvimento do Aplicativo com o App Inventor:	39
4.1.2	Programação no Arduino IDE:	48
4.1.3	Montagem do Circuito para o Sistema <i>Three-Way</i> :	51
4.1.4	Montagem do Circuito para o Controle de Brilho da Lâmpada:	52
4.1.5	Testes e Ajustes:	53
5.	CONSIDERAÇÕES FINAIS	54
	REFERÊNCIAS	55
	APÊNDICE A – BLOCOS DE PROGRAMAÇÃO DA TELA DO <i>THREE WAY</i>	57
	APÊNDICE B – BLOCOS DE PROGRAMAÇÃO DA TELA DO <i>DIMMER</i>	58
	APÊNDICE C – CÓDIGO PARA A PROGRAMAÇÃO DO ARDUINO	59

AGRADECIMENTOS

A finalização deste trabalho de conclusão de curso representa não apenas o término de uma etapa acadêmica, mas também a concretização de um esforço conjunto, repleto de desafios e superações. Em primeiro lugar, expresso minha profunda gratidão a Deus, cuja graça e orientação foram fundamentais em todos os momentos desta jornada. Agradeço por sua infinita misericórdia que me concedeu força e determinação para superar os obstáculos ao longo do curso e especialmente, para concluir este trabalho.

À minha família, expresso minha eterna gratidão. O apoio incondicional de meus pais Pedro e Maria Elizangela foram a âncora que sustentaram minha jornada. Agradeço também aos demais membros da família e aos amigos que, de maneira direta ou indireta, contribuíram para essa conquista.

Ao meu orientador Dr. Zenner Silva Pereira, expresso minha sincera gratidão pela orientação, pela paciência e pelo comprometimento ao longo deste trabalho. Suas orientações foram fundamentais para o desenvolvimento e aprimoramento deste estudo, representando uma influência valiosa em minha formação acadêmica.

Por fim, agradeço a todos que, de alguma forma, fazem parte desta jornada. Este trabalho não seria possível sem a contribuição e apoio de cada um de vocês. A conclusão desta etapa é uma vitória coletiva, e celebro com profunda gratidão por todas as bênçãos e aprendizados ao longo do caminho. Que este seja apenas o início de muitas outras realizações e conquistas. Obrigado a todos que tornaram este momento possível.

“Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor, que um dia você chega lá. De alguma maneira você chega lá.”

Ayrton Senna

RESUMO

A automação residencial tem se tornado cada vez mais popular devido a sua capacidade de melhorar o conforto, segurança e eficiência energética das residências. Neste contexto o sistema proposto neste trabalho oferece uma solução acessível para automatizar o controle de iluminação em uma casa, com um aplicativo móvel que permite aos usuários controlarem remotamente a iluminação, proporcionando praticidade e facilidade de uso. Este projeto consistiu na implementação de dois sistemas de automação residencial para controle de iluminação. Na primeira parte, foi desenvolvido um sistema *three-way* que combina um interruptor físico convencional, um relé e um aplicativo móvel dedicado, permitindo o controle remoto da lâmpada tanto pelo interruptor físico quanto pelo aplicativo. Na segunda parte, foi projetado um sistema para ligar, desligar e controlar o brilho de uma lâmpada através do aplicativo, utilizando um módulo *dimmer* para ajustar a intensidade luminosa de acordo com as preferências do usuário, sendo compatível com lâmpadas LED dimerizáveis. Ambos os sistemas foram desenvolvidos com o uso do microcontrolador Arduino.

Palavras-Chave: automação residencial; aplicativo móvel; microcontrolador arduino.

ABSTRACT

Home automation has become increasingly popular due to its ability to improve the comfort, safety and energy efficiency of homes. In this context, the system proposed in this work offers a flexible and affordable solution for automating lighting control in a home, with a mobile app that allows users to remotely control lighting, providing practicality and ease of use. This project consisted of implementing two home automation systems for lighting control. In the first part, a three-way system was developed that combined a conventional physical switch, a relay and a dedicated mobile app, allowing remote control of the lamp by both the physical switch and the app. In the second part, a system was designed to turn on, off and control the brightness of a lamp, using a dimmer module to adjust the light intensity according to the user's preferences, being compatible with dimmable LED lamps. Both systems were developed using the Arduino microcontroller.

Keywords: home automation; mobile app; arduino microcontroller.

1. INTRODUÇÃO

1.1 MOTIVAÇÃO

A trajetória da evolução tecnológica, desde as primeiras invenções da humanidade, reflete a incessante busca por facilitar a vida cotidiana. Desde as rudimentares ferramentas de caça até a descoberta do fogo e a criação da roda, o ser humano sempre buscou maneiras de tornar suas tarefas mais eficientes.

A lógica por trás desse desenvolvimento foi a criação de inovações que contribuíssem para o progresso da sociedade, como o telefone, a luz elétrica, a televisão, o rádio e a internet. Ao longo das décadas, a velocidade e a necessidade de evolução tecnológica aumentaram exponencialmente, desafiando algumas gerações a acompanhar e se adaptar a esse ritmo acelerado (OWINTERACTIVE, 2020).

No cenário contemporâneo, a presença de tecnologia nas residências tornou-se uma realidade intrínseca ao cotidiano, transformando a forma como interagimos com nossos ambientes domésticos. A automação residencial desponta como uma vertente crucial desse avanço tecnológico, oferecendo soluções inovadoras para otimizar a gestão de recursos, aumentar a segurança e proporcionar maior conforto aos moradores.

Nesse contexto, a utilização do Arduino, plataforma de prototipagem eletrônica de código aberto, associada a aplicativos móveis, emerge como uma poderosa ferramenta na implementação de sistemas de automação residencial acessíveis e customizáveis. A junção dessas tecnologias permite a criação de soluções inteligentes, capazes de atender às necessidades específicas de cada lar, adaptando-se às demandas individuais dos usuários.

1.2 PROBLEMAS

Atualmente, a maneira mais comum de controlar a iluminação dos cômodos de uma casa é por meio do uso de interruptores físicos. No entanto, esse método tradicional apresenta limitações significativas. Ele exige a locomoção até o interruptor correspondente para ligar, desligar ou ajustar a intensidade da lâmpada, o que pode ser inconveniente, especialmente para pessoas com mobilidade reduzida, podendo resultar em um desperdício desnecessário de energia, já que as luzes podem permanecer acesas em cômodos desocupados.

1.3 OBJETIVO

Este trabalho propõe a implementação de um sistema *three-way* inovador que integra a plataforma Arduino, um aplicativo móvel e um controle físico do interruptor, além de incluir um sistema para o controle de intensidade luminosa. O objetivo principal é possibilitar o usuário a controlar remotamente as lâmpadas, eliminando a necessidade de deslocamento até os interruptores físicos. Essa abordagem visa proporcionar maior conveniência e flexibilidade no gerenciamento da iluminação residencial. Com a integração desses componentes, os usuários poderão ajustar a intensidade da luz e ligar ou desligar as lâmpadas de qualquer lugar, utilizando apenas um dispositivo móvel ou um controle físico, o que torna o processo mais simples e acessível. Assim, o sistema busca atender as necessidades do usuário moderno, que valoriza praticidade e tecnologia em seu dia a dia.

2. REFERÊNCIAL TEÓRICO

2.1 AUTOMAÇÃO RESIDENCIAL

O termo automação residencial, também conhecido como domótica, refere-se a uma tecnologia moderna que possibilita a gestão e controle de diversos recursos e dispositivos presentes em uma residência. Isso inclui sistemas de iluminação, climatização, segurança, entre outros, que podem ser monitorados e controlados de forma automatizada através de dispositivos eletrônicos e *softwares* específicos (SEBRAE, 2023).

2.1.1 Evolução Histórica da Automação Residencial

Quando se observa o desenvolvimento da tecnologia, um ponto de partida significativo pode ser identificado com o surgimento da máquina a vapor, que marcou o início da Revolução Industrial no século XVIII na Inglaterra. Esse período testemunhou a transição do sistema de produção feudal para o modelo capitalista. A partir desse ponto, avanços notáveis, como a eletricidade e sua disseminação generalizada, abriram caminho para novas ferramentas e necessidades, resultando, nos anos 60, no advento da automação industrial (SANTOS, 2017).

Durante a década de 60, empresas de tecnologia começaram a perceber a crescente demanda por tecnologia residencial. Inicialmente, porém, não se consideravam as particularidades desse mercado. Residências não requeriam a mesma imunidade a falhas que as indústrias, mas os produtos destinados a uso doméstico precisavam oferecer qualidade, uma interface amigável e intuitiva. Na década seguinte, ocorreu o lançamento dos primeiros módulos X-10 (SANTOS, 2017).

Com o lançamento dos primeiros módulos X-10, que foi a primeira tecnologia amplamente adotada para automação residencial, na década de 70, houve um avanço significativo na tecnologia voltada para o lar. Inicialmente, a automação residencial era limitada a funções básicas, como controle de iluminação e dispositivos elétricos. No entanto, com o avanço da eletrônica e da computação, novas possibilidades foram exploradas (BREDA, 2016).

Na década de 80 e 90, houve uma expansão dos sistemas de automação residencial, com a introdução de tecnologias como controle remoto infravermelho e sistemas de segurança integrados. Os consumidores começaram a ter acesso a dispositivos mais sofisticados, como sistemas de áudio distribuído e até mesmo sistemas de vigilância por vídeo.

Com o advento da internet e a popularização da conectividade Wi-Fi no início dos anos 2000, a automação residencial deu um salto significativo. Surgiram dispositivos inteligentes controlados por aplicativos móveis e assistentes de voz, permitindo que os usuários controlassem seus sistemas domésticos de qualquer lugar do mundo.

Em 2005, o lançamento do Arduino teve um impacto marcante na evolução da automação residencial. Democratizando o acesso à tecnologia, o Arduino permitiu que indivíduos com pouca ou nenhuma experiência em eletrônica e programação desenvolvessem seus próprios sistemas inteligentes para casa.

Nos últimos anos, a inteligência artificial (IA) e a aprendizagem de máquina começaram a desempenhar um papel importante na automação residencial. Os dispositivos agora podem aprender as preferências dos usuários e antecipar suas necessidades, tornando os sistemas mais intuitivos e personalizados.

2.1.2 Importância e benefícios da automação residencial

Ambientes automatizados oferecem comodidade a indivíduos com agendas agitadas, contribuindo para a redução do estresse e possibilitando que as pessoas desfrutem mais plenamente de seus lares. Essas são melhorias simples que não exigem um grande investimento de tempo ou esforço por parte dos moradores. Os proprietários das residências podem configurar uma série de dispositivos automatizados sem a necessidade de interagir com *softwares* complexos. Isso abrange sistemas de câmeras de segurança, controle de climatização, gerenciamento de iluminação e muito mais (PERIN, 2022).

2.2 ARDUINO

O Arduino é uma plataforma eletrônica de código aberto que combina hardware e *software* de fácil utilização. As placas Arduino têm a capacidade de captar diversos tipos de entradas, como luz em um sensor, pressionar um botão ou até mesmo receber mensagens de aplicativos para, em seguida, transformá-las em saídas que executam ações específicas. Por meio da linguagem de programação Arduino, que se baseia no conceito de "Fiação" e do *software* Arduino (IDE), que se apoia na estrutura do Processamento, é possível instruir a placa a realizar tarefas específicas de acordo com as suas necessidades e projetos (ARDUINO, 2018).

2.2.1 Origem do Arduino

Em 2005, um grupo de cinco pesquisadores - Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, se uniram para dar origem ao Arduino. Com o propósito central de criar um dispositivo acessível em termos de custos e fácil de programar, tornando-o assim disponível para estudantes e entusiastas de projetos amadores (THOMSEN, 2014).

A primeira placa Arduino foi equipada com um microcontrolador da Atmel e programada através de um Ambiente de Desenvolvimento Integrado (IDE) que utilizava uma linguagem baseada em C/C++. Além disso, os circuitos de entrada e saída podiam ser facilmente conectados a um computador por meio de um cabo USB. Essa abordagem descomplicada democratizou o acesso à eletrônica e à programação, estabelecendo as bases para a proliferação de projetos criativos e inovadores em todo o mundo (THOMSEN, 2014).

2.2.2 Placas do Arduino

As placas do Arduino em sequência de lançamento (figura 1).

Figura 1 - Tipos de Placas Arduino.



Fonte: MAKER HERO (2014)

- **void loop ():**

Depois que a função `setup ()` é concluída, o programa entra na função `loop ()`. Esta função é executada repetidamente em um loop infinito enquanto a placa Arduino estiver ligada. É onde o comportamento principal do programa é implementado. O código dentro da função `loop ()` é executado continuamente, a menos que seja interrompido ou o programa seja reiniciado. Aqui, podem ser definidas as ações que deseja que o seu programa execute repetidamente, como ler dados de sensores, processar informações, controlar dispositivos, enviar dados pela porta serial, etc.

Na interface da IDE do Arduino, encontra-se ícones que representam funcionalidades importantes, entre eles estão:

- **Verificar**

Este botão compila o código-fonte do seu programa para verificar se há erros de sintaxe. Ele não carrega o código para a placa, apenas verifica se o código está escrito corretamente.

- **Upload**

Este botão carrega o código compilado para a placa Arduino conectada ao computador. Após a verificação bem-sucedida, pode-se usar o botão de upload para transferir seu código para a placa.

- **Debug**

A IDE do Arduino não possui uma funcionalidade de depuração integrada como em IDEs mais avançadas. Em vez disso, é possível usar instruções de depuração no código, como impressões de debug, para monitorar o comportamento do programa durante a execução.

- **Selecionar a Porta**

Este menu suspenso permite selecionar a porta serial à qual a placa Arduino está conectada. É necessário escolher a porta correta para carregar o código com sucesso para a placa.

- **Plotar Serial**

Esta é uma ferramenta que permite visualizar os dados enviados pela porta serial em forma de gráfico. É útil para visualizar leituras de sensores ou outras informações analógicas ao longo do tempo.

- **Monitor Serial**

Esta é uma janela na qual pode-se ver mensagens de depuração, leituras de sensores e outras informações que seu programa envia pela porta serial. É uma ferramenta valiosa para depurar e entender o comportamento do seu código em tempo real.

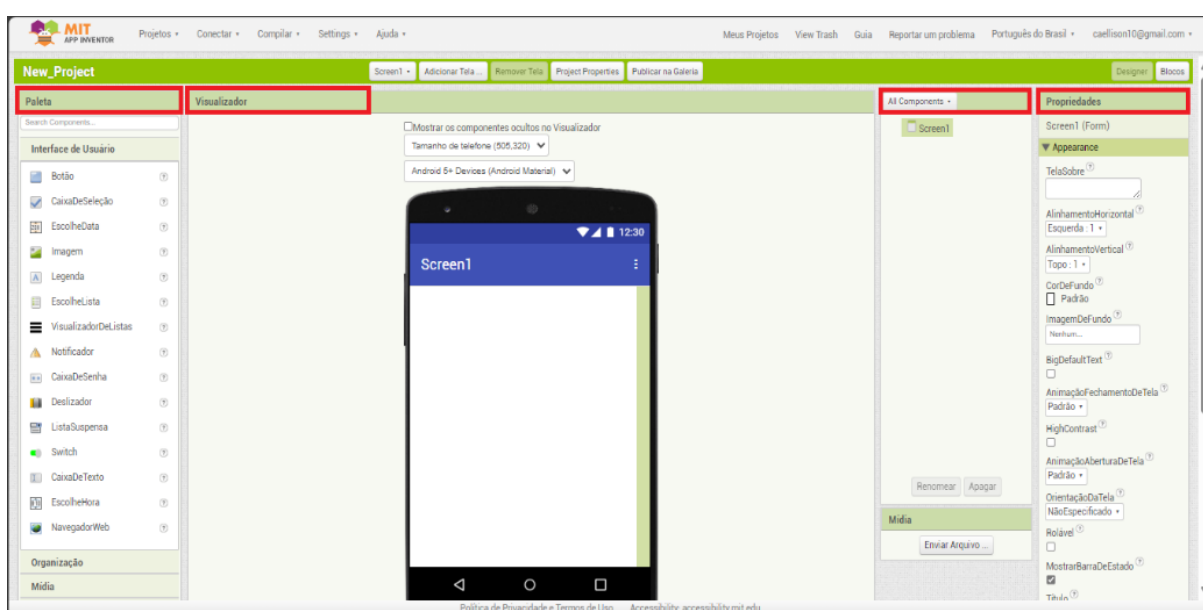
2.3 MIT APP INVENTOR

O MIT App Inventor é uma ferramenta inicialmente desenvolvida pelo Google e atualmente mantida pelo Instituto de Tecnologia de Massachusetts (MIT). Ela possibilita a criação de aplicativos para smartphones que funcionam no sistema operacional Android (TECH TUDO, 2023).

Um dos motivos de se utilizar o App Inventor é a praticidade, pois ele torna o desenvolvimento de aplicativos móveis mais acessível a pessoas sem conhecimento avançado de programação. A plataforma foi projetada para permitir que até mesmo iniciantes pudessem criar aplicativos para dispositivos Android usando uma interface de arrastar e soltar, bem como blocos de programação visual (TECH TUDO, 2023).

A interface de designer do MIT App Inventor é dividida em quatro partes principais para facilitar o desenvolvimento de aplicativos móveis, conforme ilustrado na Figura 3: a Paleta, o Visualizador, os Componentes e as Propriedades.

Figura 3- Interface de Designer do App Inventor.



Fonte: App Inventor (2024).

- **Paleta:**

A paleta é onde se encontra todos os componentes que podem ser adicionados a interface de usuário. Esses componentes incluem botões, listas, imagens, sensores, entre outros. Cada componente na paleta tem uma função específica e pode ser arrastado para o visualizador para criar a interface do seu aplicativo.

- **Visualizador:**

O visualizador é onde se projeta a interface do aplicativo arrastando e soltando componentes da paleta. Onde pode-se posicionar e redimensionar os componentes para criar a aparência desejada para o seu aplicativo. O visualizador permite uma visualização em tempo real do layout do aplicativo conforme são feitas alterações.

- **Componentes:**

A seção de componentes exibe uma lista de todos os componentes que foram adicionados ao projeto. Onde é possível selecionar um componente específico para acessar suas propriedades e eventos. Também pode-se renomear, copiar ou excluir componentes nesta seção.

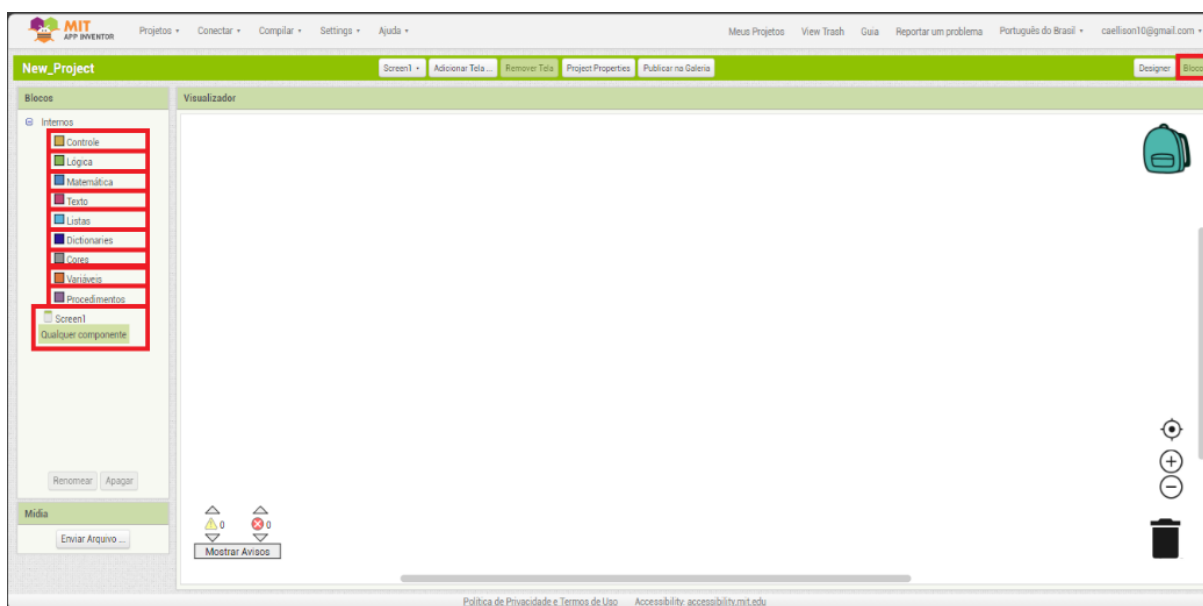
- **Propriedades:**

As propriedades são características individuais de cada componente, como cor, tamanho, texto, entre outras. Na seção de propriedades, pode-se personalizar os componentes do aplicativo ajustando suas propriedades conforme necessário. Isso inclui definir o texto de um botão, a cor de fundo de uma tela, a imagem de um componente, entre outros aspectos visuais e funcionais do seu aplicativo.

Essas quatro partes trabalham em conjunto para projetar a interface do seu aplicativo no MIT App Inventor, permitindo a criação de aplicativos personalizados de forma intuitiva e eficiente.

Além da interface de designer, o MIT App Inventor também oferece uma interface de programação baseada em blocos, onde pode-se definir o comportamento do seu aplicativo através da lógica de programação visual. Nesta interface, os blocos representam diferentes ações, eventos e operações que podem ser conectados para criar a funcionalidade do seu aplicativo. Na figura 4 cada bloco está destacado:

Figura 4 - Interface de programação App Inventor.



Fonte: App Inventor (2024).

- **Blocos de Controle:**

Os blocos de controle são responsáveis por orientar o fluxo de execução do programa. Eles incluem estruturas como "if", que permitem executar determinados blocos de código se uma condição for verdadeira, e "else", que especifica o que fazer caso essa condição não seja atendida. Por exemplo, imagine um aplicativo de clima que exibe uma mensagem diferente dependendo da temperatura. Se a temperatura for maior que 25°C, exiba "Está quente!". Se não, exiba "Está frio!". Além disso, os blocos de controle incluem loops, como "for" e "while", que permitem repetir blocos de código várias vezes, e "switch", que seleciona um bloco de código com base no valor de uma variável.

- **Blocos de Lógica:**

Os blocos de lógica ajudam a tomar decisões com base em condições e comparar valores. Eles incluem operadores lógicos como "and", "or" e "not", que permitem combinar condições. Por exemplo, em um aplicativo de quiz, pode-se usar o operador "and" para verificar se o usuário respondeu corretamente a todas as perguntas e se também completou o quiz dentro

do tempo limite. Além disso, os comparadores como "=" e ">" são úteis para comparar valores, como verificar se um número é igual a outro ou se é maior que outro número.

- **Blocos de Matemática:**

Os blocos de matemática são utilizados para realizar operações aritméticas e funções matemáticas. Eles incluem operações básicas como adição, subtração, multiplicação e divisão, além de funções mais avançadas como raiz quadrada e potência. Por exemplo, em um aplicativo de calculadora, pode-se usar os blocos de matemática para calcular a soma de dois números ou encontrar a raiz quadrada de um número.

- **Blocos de Texto:**

Os blocos de texto são utilizados para manipular cadeias de caracteres. Eles incluem operações como concatenação, que combina duas ou mais strings, e substituição, que substitui parte de uma string por outra. Por exemplo, em um aplicativo de edição de texto, pode-se usar os blocos de texto para permitir que o usuário insira texto em um campo de entrada e, em seguida, manipule esse texto de várias maneiras, como adicionando formatação ou substituindo palavras específicas.

- **Blocos de Listas e Dicionários:**

Os blocos de listas e dicionários são usados para armazenar e manipular conjuntos de dados. As listas são coleções ordenadas de itens, enquanto os dicionários são coleções de pares chave-valor. Por exemplo, em um aplicativo de lista de tarefas, pode-se usar uma lista para armazenar as tarefas a serem concluídas e um dicionário para armazenar informações adicionais sobre cada tarefa, como sua prioridade ou prazo.

- **Blocos de Cores:**

Os blocos de cores permitem especificar cores para elementos visuais. Eles são úteis ao personalizar a aparência do aplicativo, como definir a cor de fundo de um botão ou a cor do texto em um rótulo.

- **Blocos de Variáveis:**

Os blocos de variáveis são utilizados para armazenar valores temporários. Eles incluem operações como definir uma variável, atribuindo um valor a ela, e obter uma variável,

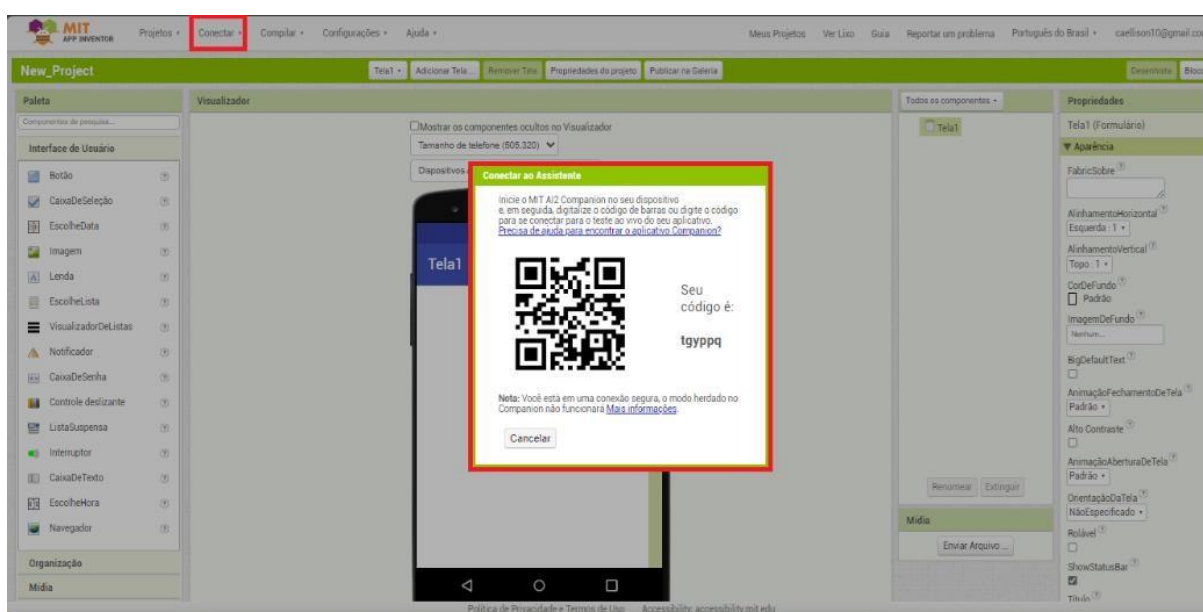
recuperando o valor armazenado nela. Variáveis são úteis quando precisa armazenar informações que podem mudar ao longo do tempo, como o resultado de um cálculo.

- **Blocos de Procedimentos:**

Os blocos de procedimentos são utilizados para agrupar conjuntos de blocos de código que realizam uma determinada tarefa ou ação em um único bloco. Isso permite que os desenvolvedores organizem e reutilizem funcionalidades complexas nos aplicativos.

Esses são os blocos disponíveis na interface de programação do MIT App Inventor. Combinando esses blocos de maneiras diferentes, pode-se criar uma ampla variedade de funcionalidades para o seu aplicativo, desde operações simples até aplicativos complexos e interativos. Além disso no App Inventor é possível testar os aplicativos em desenvolvimento de forma prática e eficiente. Utilizando o aplicativo MIT AI2 Companion que está disponível na google play para dispositivo móvel, os desenvolvedores podem estabelecer uma conexão direta com o projeto em andamento no ambiente de programação. Isso é feito através da leitura de um QR Code, como mostrado na figura 5, ou pela conexão via cabo USB. Uma vez conectado, o aplicativo em desenvolvimento é espelhado em tempo real no dispositivo, permitindo que os desenvolvedores interajam com ele para testar funcionalidades e verificar o comportamento do aplicativo conforme as alterações são feitas no projeto. Essa abordagem simplifica o processo de teste e depuração, garantindo que os aplicativos sejam refinados e aprimorados de maneira eficaz durante o processo de desenvolvimento.

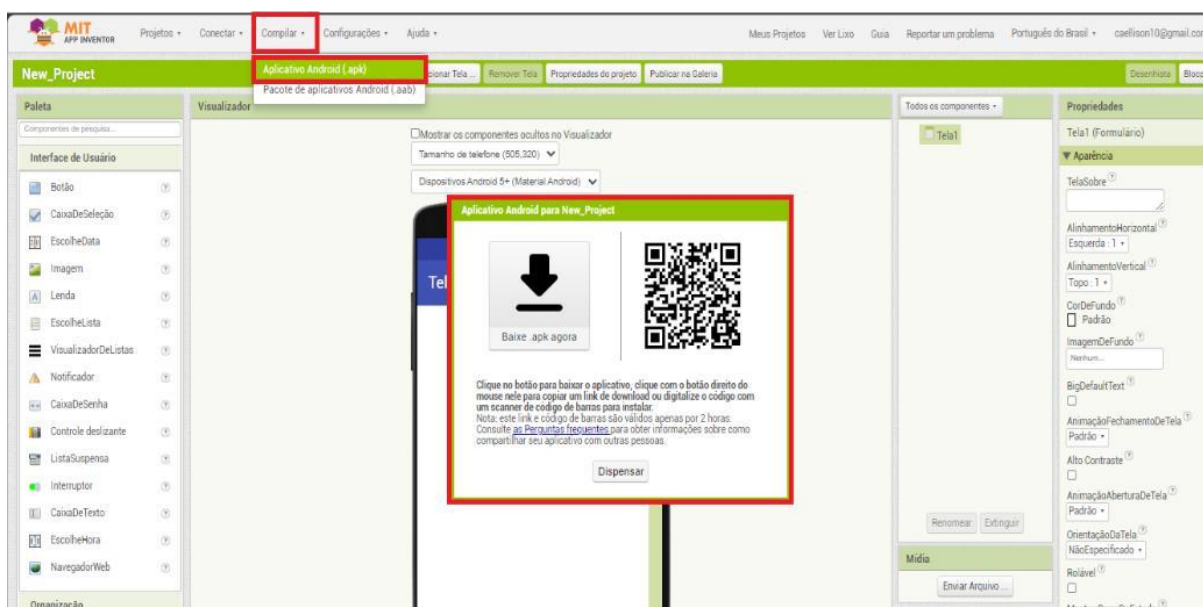
Figura 5 - Espelhamento via QR Code do App em tempo real.



Fonte: App Inventor (2024).

Quando o aplicativo estiver pronto para ser usado fora do ambiente de desenvolvimento, o MIT App Inventor oferece a opção de compilar e instalar diretamente no dispositivo Android através do aplicativo MIT AI2 Companion. Ao selecionar essa opção, o App Inventor gera um arquivo APK do seu aplicativo, como destacado na Figura 6, que contém todas as informações necessárias. Em seguida, o Companion no dispositivo Android recebe e instala automaticamente esse arquivo, permitindo que o aplicativo seja executado independentemente do ambiente de programação.

Figura 6 - Compilando app Android.



Fonte: App Inventor (2024).

2.4 BLUETOOTH

O Bluetooth é uma tecnologia de comunicação sem fio que permite a transferência de dados e a conexão de dispositivos eletrônicos a curtas distâncias. É comumente utilizado por sua facilidade de uso, eficiência e ampla compatibilidade, além de oferecer segurança, conexões de curto alcance e emparelhamento simples, sendo amplamente utilizado em diversas aplicações, incluindo automação (LUZ, 2018).

2.5 COMUNICAÇÃO SERIAL

A comunicação serial é um método de transmissão de dados em que os bits de informação são enviados sequencialmente, um após o outro, através de um único canal de comunicação. Isso contrasta com a comunicação paralela, onde múltiplos bits de dados são transmitidos simultaneamente em canais separados.

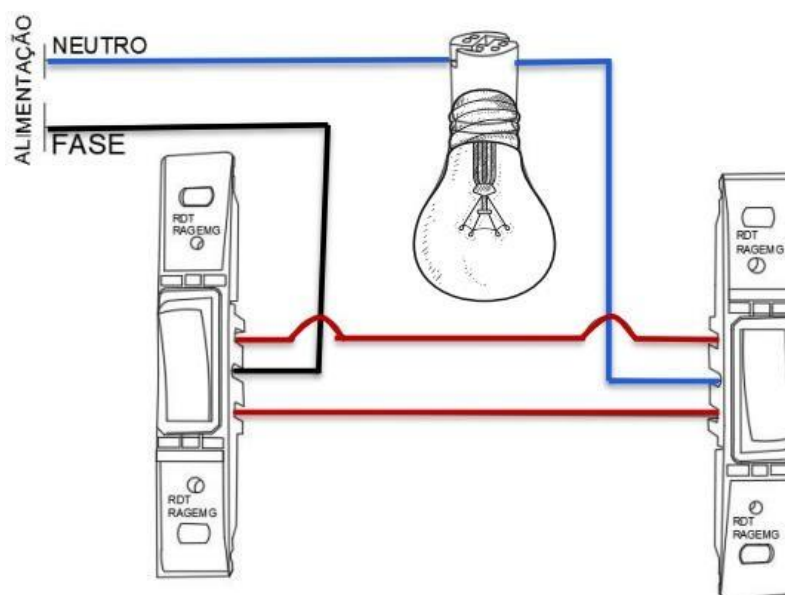
Na comunicação serial, os bits de dados são enviados um de cada vez, seguindo uma ordem específica, e são reconstruídos na ordem correta no destino. Isso é feito utilizando protocolos de comunicação que especificam a estrutura dos dados transmitidos, como a ordem dos bits, a taxa de transmissão (baud rate), os métodos de detecção de erros e outros parâmetros.

A finalidade da comunicação serial é transferir dados de forma eficiente e confiável entre dispositivos eletrônicos, mesmo quando há limitações de recursos, como largura de banda limitada, número limitado de pinos de comunicação ou distâncias longas entre os dispositivos.

2.6 THREE WAY

Um circuito *three-way* convencional com dois interruptores paralelos oferece o controle de uma lâmpada de dois locais diferentes. Nesse tipo de arranjo, ambos os interruptores estão conectados em paralelo com a lâmpada, conforme a figura 7, permitindo que ela seja ligada ou desligada independentemente da posição de qualquer um dos interruptores. Por exemplo, ao entrar em um corredor, é possível acender a luz tanto no início quanto no final do corredor, com qualquer um dos interruptores. Se um interruptor estiver na posição de ligado, é possível desligar a luz no outro interruptor e vice-versa, independentemente da posição de cada interruptor (STRAUB, 2021).

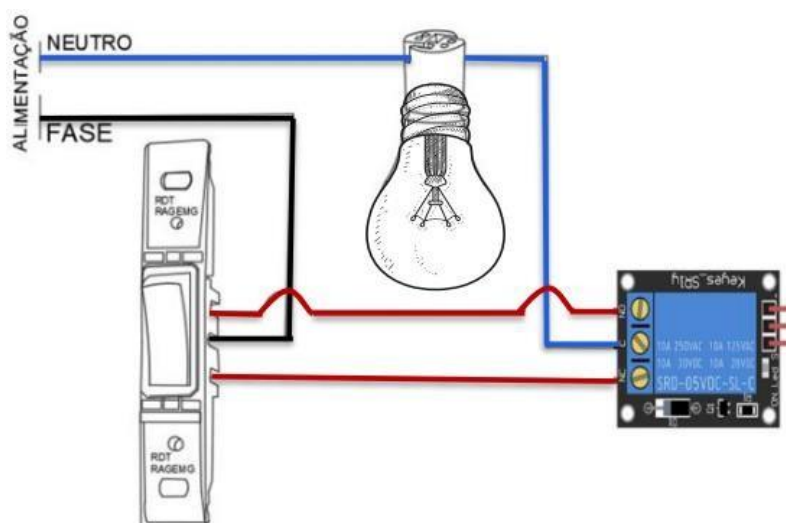
Figura 7 - *Three Way* convencional com dois Interruptores Paralelo.



Fonte: Autoria própria (2024).

Por outro lado, um circuito *three-way* com um interruptor paralelo e um aplicativo móvel oferece um controle mais flexível e moderno sobre a iluminação. Nesse caso, um dos interruptores é substituído por um dispositivo inteligente, como um módulo relé, representado na figura 8, controlado por um aplicativo móvel. Enquanto o interruptor físico paralelo oferece a funcionalidade básica de ligar e desligar a lâmpada, o aplicativo móvel permite controlar remotamente a lâmpada através de um smartphone ou tablet (STRAUB, 2021).

Figura 8 - *Three Way* com Interruptor Paralelo com relé.



Fonte: Autoria própria (2024).

3. METODOLOGIA

Aviso de Segurança: Este projeto envolve o uso de uma tensão de 220V/ 127 V, que é considerada alta para aplicações domésticas e comerciais, apresentando um risco significativo de choques elétricos e danos físicos e materiais graves. Recomenda-se que apenas pessoas com experiência e conhecimento adequados realizem este projeto. Certifique-se de entender completamente os procedimentos de segurança e de ter as habilidades necessárias antes de prosseguir com a execução.

3.1 MATERIAIS

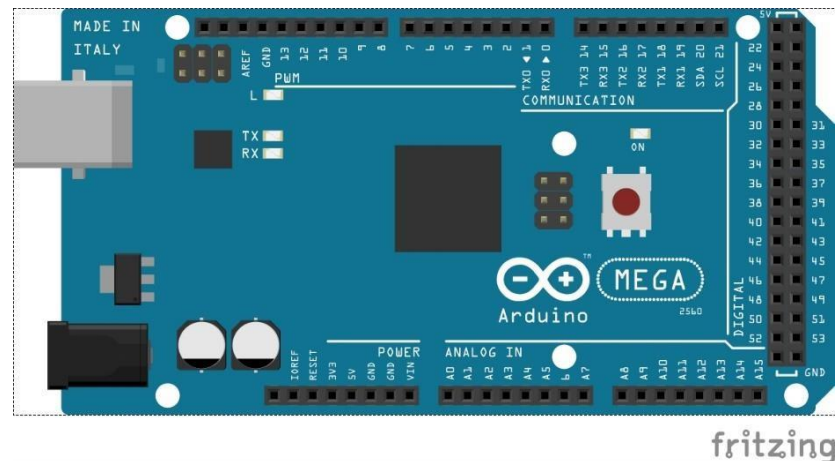
No projeto, foram utilizados os seguintes materiais: Arduino Mega 2560, *Jumpers*, Fios, Resistores, Protoboard, Módulo Bluetooth HC-06, Interruptor paralelo, Módulo Relé, Módulo *Dimmer* MC-8^a e duas Lâmpadas.

3.1.1 Arduino mega 2560

No projeto a função do Arduino é atuar como o cérebro do sistema de automação residencial, coordenando e controlando as operações dos dispositivos eletrônicos envolvidos, como relés, interruptores, módulos de comunicação e sensores. Ele executa o código programado para implementar as funcionalidades desejadas, como o controle remoto da iluminação, ajuste de brilho e comunicação com o aplicativo móvel.

A versão utilizada foi o Arduino Mega 2560, representado na figura 9, e a vantagem está na sua capacidade expandida de entrada/saída e recursos adicionais em comparação com versões mais básicas do Arduino. Com seus numerosos pinos de entrada/saída e memória aprimorada, o Arduino Mega 2560 oferece mais flexibilidade e poder de processamento, permitindo a conexão e controle de um maior número de componentes eletrônicos. Isso é especialmente útil em projetos complexos de automação residencial, nos quais há múltiplos dispositivos a serem controlados e interligados.

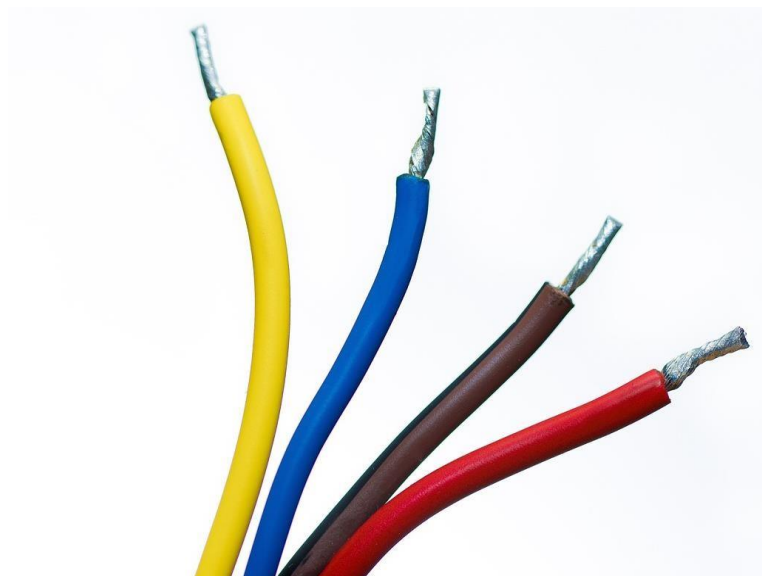
Figura 9 - Arduino Mega 2560.



Fonte: Fritzing (2024).

3.1.2 Fios e *Jumpers*

Fios e *jumpers*, representados na figura 10, desempenham papéis fundamentais na construção de circuitos eletrônicos, facilitando a conexão entre componentes e permitindo a transmissão de energia elétrica e sinais. Enquanto os fios são utilizados para estabelecer conexões permanentes ou de longo prazo, roteando os circuitos de maneira organizada e oferecendo flexibilidade na montagem, os *jumpers* são empregados em conexões temporárias, configurando hardware, selecionando modos de operação ou facilitando testes e depurações.

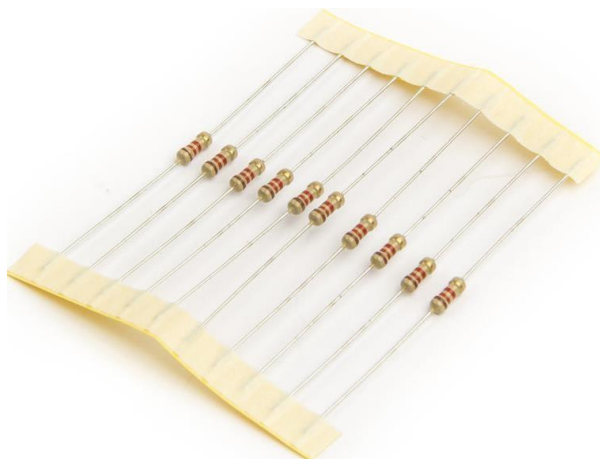
Figura 10 - Fios e *Jumpers*.

Fonte: Pixabay (2015).

3.1.3 Resistores

Os resistores são componentes eletrônicos essenciais em circuitos elétricos, responsáveis por controlar a corrente que flui pelo circuito, limitando-a conforme necessário. Além de limitar a corrente, eles são usados para dividir a tensão, filtrar sinais, terminar linhas de transmissão e polarizar transistores. Sua versatilidade os torna indispensáveis em várias aplicações na eletrônica, garantindo o funcionamento adequado e seguro de dispositivos e sistemas. Um método comum para identificar o valor da resistência de um resistor é através das cores de suas faixas, como mostra na figura 11, um sistema padronizado utilizado internacionalmente que indica o valor nominal da resistência e sua tolerância.

Figura 11 - Resistores.



Fonte: ARDUINO (2024).

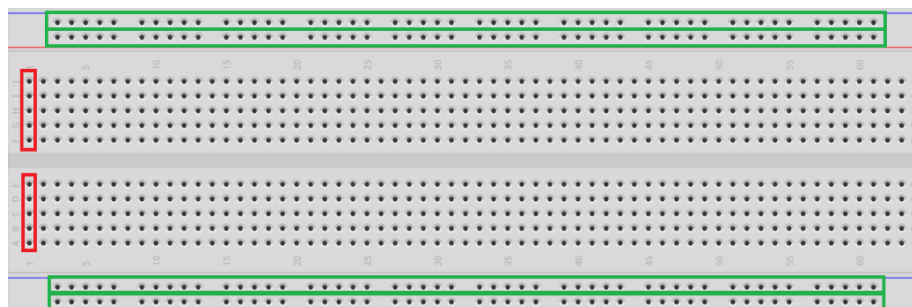
3.1.4 Protoboard

Uma protoboard, também conhecida como breadboard, é uma placa de circuito com uma série de furos condutores em uma matriz. Ela é usada para montagem temporária e teste de circuitos eletrônicos sem a necessidade de soldagem. Cada fileira de furos está conectada eletricamente entre si, tanto na vertical quanto na horizontal, seguindo um padrão pré-determinado.

Para ligar componentes em uma protoboard, você insere os terminais dos componentes (como resistores, LEDs, fios, capacitores, etc.) nos furos da placa. Os furos em cada fileira estão conectados de acordo como destacada na figura 12, o que permite a passagem de corrente entre os componentes. Fios jumper também são usados para fazer conexões entre os componentes,

permitindo criar circuitos mais complexos. Esses fios podem ser inseridos nos furos para estabelecer conexões entre diferentes partes do circuito conforme necessário.

Figura 12 - Conexões da Protoboard.



Fonte: Fritzing (2024).

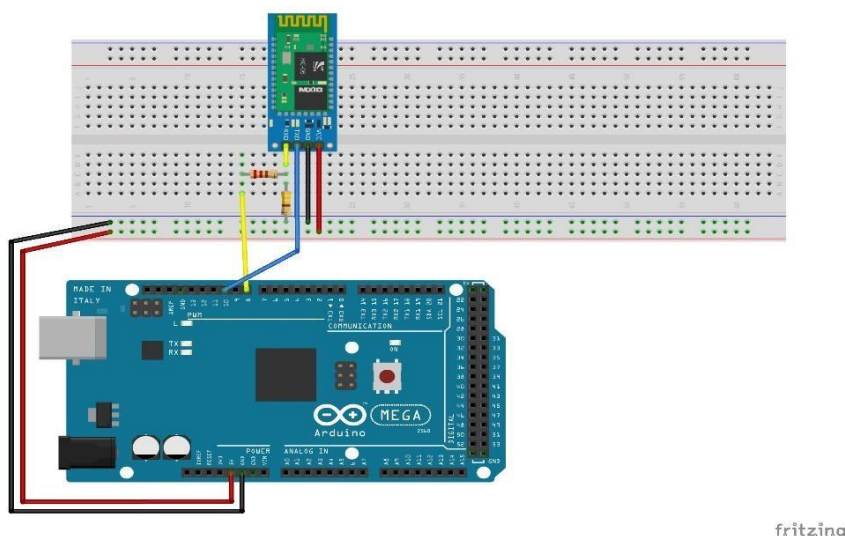
3.1.5 Módulo Bluetooth HC-06

O módulo Bluetooth HC-06 é um componente que permite a comunicação sem fio entre dispositivos, como um smartphone ou tablet, e um microcontrolador, como o Arduino, por exemplo. No projeto, ele estabelece uma conexão sem fio entre o aplicativo móvel e o Arduino, permitindo que o aplicativo envie comandos para o Arduino, que então interpreta esses comandos e controla o estado do relé para ligar ou desligar a iluminação.

O pino RX (Receptor) do módulo Bluetooth HC-06 é a entrada de dados, onde ele recebe informações do Arduino ou de outro dispositivo conectado. No entanto, como o módulo HC-06 opera com uma tensão de 3,3 volts nesse pino, é necessário reduzir a tensão antes de enviar dados do Arduino para evitar danos ao módulo, pois os pinos do Arduino trabalham numa tensão de 5 volts.

Foi utilizado um divisor de tensão, apresentado na figura 13, composto por dois resistores: um de 220 ohms e outro de 440 ohms para reduzir a tensão do sinal de saída do Arduino de 5 volts para aproximadamente 3,3 volts, garantindo que o módulo Bluetooth HC-06 receba uma tensão segura em seu pino RX.

Figura 13 - Conexão do Módulo Bluetooth HC-06.



Fonte: Autoria própria (2024).

3.1.6 Interruptor Paralelo

Um interruptor paralelo representado na figura 14, também conhecido como interruptor de três vias, é um dispositivo elétrico utilizado para controlar uma única luz ou dispositivo a partir de dois ou mais pontos diferentes. Ele permite que a luz seja ligada ou desligada de qualquer posição dos interruptores conectados em paralelo. A instalação requer um cabo de três condutores entre os interruptores, e sua função de controle de iluminação em três vias permite que os usuários ativem ou desativem a luz de diferentes pontos da área em que ela está instalada, sem a necessidade de retornar ao ponto inicial.

Figura 14 – Interruptor Paralelo



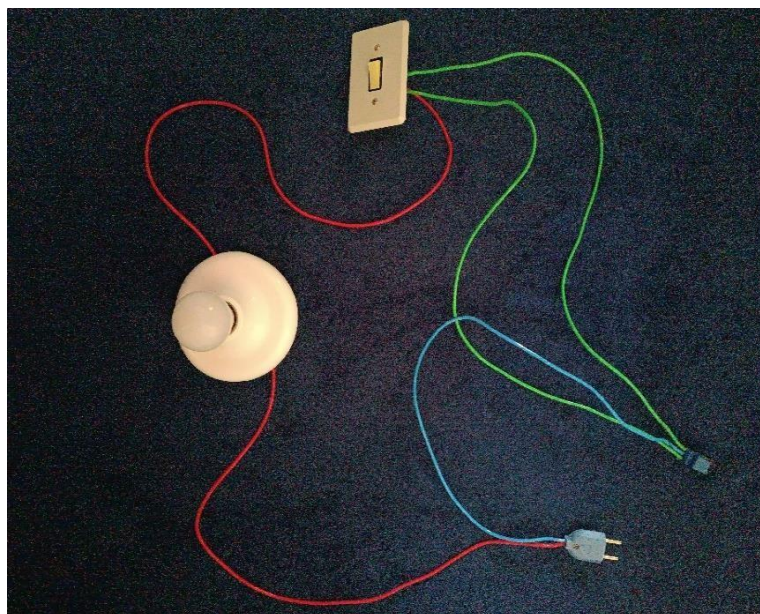
Fonte: Quintel (2024).

3.1.7 Módulo Relé

O relé atua como um interruptor controlado eletronicamente para ligar ou desligar a lâmpada. Ele recebe um sinal do módulo Bluetooth conectado ao Arduino que, por sua vez, controla a ativação da bobina do relé. Esse sinal é enviado por meio do aplicativo móvel desenvolvido como parte do projeto. Quando a bobina do relé é energizada pelo Arduino, o comutador do relé fecha os contatos, permitindo que a corrente elétrica flua para a lâmpada, ligando-a. Quando a bobina do relé é desenergizada, os contatos do relé abrem, interrompendo a corrente e desligando a lâmpada.

O interruptor físico em paralelo com o relé, como mostra a figura 15, oferece uma alternativa de controle manual. Ele permite que a lâmpada seja ligada ou desligada manualmente, independentemente do sinal recebido pelo Arduino. Quando o interruptor físico é acionado, ele alterna o estado do circuito elétrico, fechando ou abrindo os contatos do relé, e assim, ligando ou desligando a lâmpada, respectivamente. Isso proporciona uma operação flexível do sistema, permitindo o controle tanto por meio do aplicativo quanto pelo interruptor físico.

Figura 15 - Interruptor físico em paralelo com o relé.



Fonte: Autoria própria (2024).

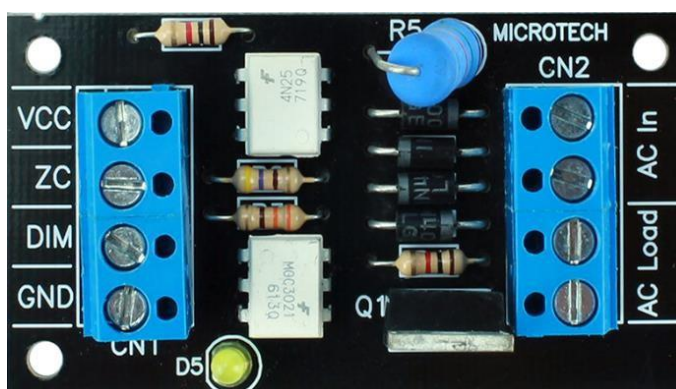
3.1.8 Módulo *Dimmer* MC-8A

O Módulo *Dimmer* MC-8A representado na figura 16, é um dispositivo eletrônico projetado para regular a quantidade de energia enviada a uma lâmpada, permitindo ajustar a luminosidade de maneira gradual, alcançando efeito de dimerização. (STRAUB, 2020).

A operação desse *dimmer* é viabilizada por um controlador Triac BT137 integrado na placa. Esse componente é comumente utilizado para modular a corrente alternada, possibilitando a dimerização de uma ou mais lâmpadas. Dessa forma, ele possibilita o controle elétrico da intensidade luminosa, com as lâmpadas incandescentes ou com lâmpadas de LED dimerizáveis. (STRAUB, 2020).

Alguns componentes do módulo estão ligados aos circuitos de alta tensão, portanto, tocar em algum deles é equivalente a tocar diretamente na tomada. É importante ter cuidado com as superfícies e materiais condutores. Recomenda-se isolar a parte inferior do módulo para garantir a segurança adequada. (STRAUB, 2020).

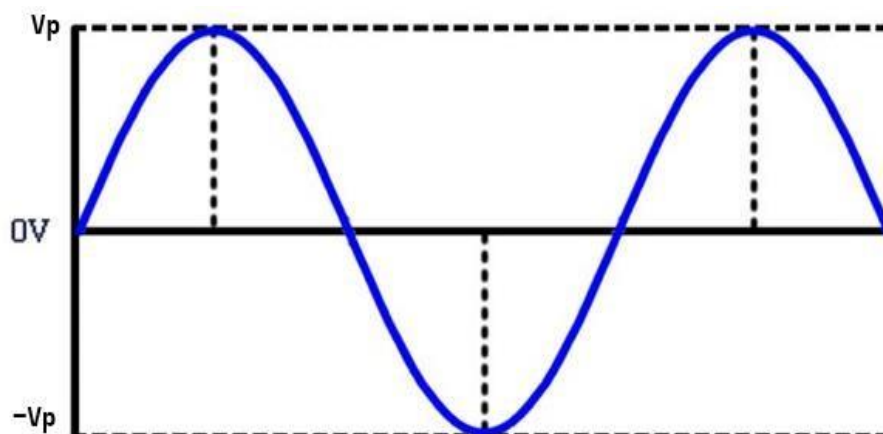
Figura 16 - Módulo *Dimmer* MC-8A.



Fonte: Usinainfo (2023).

Para compreender o sistema de funcionamento do Módulo *Dimmer*, é essencial ter uma compreensão clara de alguns conceitos básicos de elétrica. Um desses conceitos é o sistema de funcionamento da rede AC. A instalação elétrica de corrente alternada das residências no padrão brasileiro opera com uma frequência fixa de 60Hz, onde o valor da tensão alterna entre positivo e negativo 60 vezes por segundo. Essa alternância, representada na figura 17, ocorre em torno de um ponto comum de 0V, que é utilizado como referência para o controle da lâmpada e é sinalizado com um pulso para o Arduino através do pino Zero Crossing. (STRAUB, 2020).

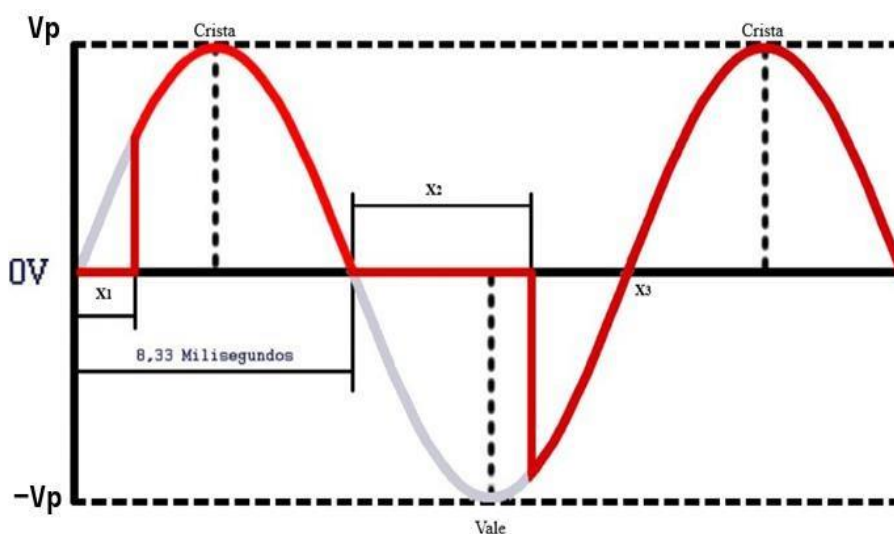
Figura 17 - Forma de Onda da rede AC.



Fonte: Adaptado de Usinainfo (2020).

A lâmpada se apaga sempre que atinge $0V$, mas logo em seguida se reacende. Esse ciclo de ligar e desligar ocorre 120 vezes por segundo, tão rapidamente que nossa visão não percebe o piscar e parece que a luz está sempre acesa. Teoricamente, é o tempo em que a lâmpada permanece em $0V$ que determina sua luminosidade. Quanto mais longa for essa duração, mais tempo a lâmpada fica apagada a cada ciclo, resultando em uma menor luminosidade perceptível. Considerando que cada semiciclo da senoide dura em média 8,33 milésimos de segundo, mesmo se a lâmpada ficar apagada o suficiente para iluminar seu mínimo, esse intervalo é tão pequeno que se torna imperceptível ao olho humano, como se a luz estivesse sempre acesa. (STRAUB, 2020).

Figura 18- Controle de tempo em 0 V.



Fonte: Adaptado de Usinainfo (2020).

A teoria de controle do tempo em 0V do *Dimmer* é simples de entender. Para uma análise mais clara, considere as cristas e o vale na figura 18: as cristas representam os valores positivos da senoide, enquanto o vale corresponde ao valor negativo. (STRAUB, 2020).

Na primeira crista, identifica-se um valor X_1 que representa o tempo em que a senoide permaneceu em 0V. Quanto menor for esse tempo, maior será o brilho da lâmpada. No vale, observa-se que o tempo em 0 foi significativamente maior, indicado pela variável X_2 , e conseqüentemente, nesses casos, o brilho da lâmpada será menor. (STRAUB, 2020).

Ao analisarmos a segunda crista, percebemos que o tempo em 0V, expresso por X_3 , foi praticamente nulo. O valor de 0 correspondeu apenas ao instante em que a senoide mudou seu valor de negativo para positivo, representando o brilho máximo da lâmpada. (STRAUB, 2020).

3.1.9 Lâmpada

Representada na figura 19, as lâmpadas são dispositivos elétricos essenciais que convertem energia elétrica em luz. Desde as tradicionais lâmpadas incandescentes até as modernas lâmpadas de LED, esses dispositivos desempenham um papel vital em nossa vida cotidiana, proporcionando iluminação para ambientes residenciais, comerciais e industriais. Sua utilidade abrange desde a iluminação básica de espaços até aplicações mais especializadas, como iluminação de rua, sinalização e iluminação de veículos.

Figura 19 - Lâmpada.



Fonte: Pixabay (2016).

3.2 MÉTODOS

O projeto proposto foi dividido em duas partes distintas, cada uma focada na implementação de sistemas de automação residencial para o controle de iluminação.

Na primeira parte do projeto, foi montado um sistema *three-way* que combinou um interruptor físico convencional com um relé e um aplicativo móvel dedicado, desenvolvido na plataforma App Inventor. Este sistema permitiu o controle remoto de uma lâmpada tanto através do interruptor físico quanto pelo aplicativo instalado em dispositivos móveis. A integração do relé ao circuito elétrico preexistente possibilitou a ativação e desativação da lâmpada de forma remota, fornecendo redundância de controle através do interruptor físico. Essa abordagem garantiu um controle flexível da iluminação dentro do ambiente residencial.

Para a segunda parte do projeto, desenvolveu-se um sistema capaz de ligar, desligar e controlar o brilho de uma lâmpada incandescente. Esse sistema foi projetado visando oferecer maior flexibilidade e controle sobre a intensidade luminosa da lâmpada, contribuindo para economia de energia e conforto visual. Utilizando um módulo *dimmer* MC-8A, conseguiu-se ajustar a intensidade luminosa conforme as preferências do usuário. Além disso, o sistema foi projetado para ser compatível com lâmpadas LED dimerizáveis, permitindo o controle de brilho de maneira semelhante às lâmpadas incandescentes. O mesmo aplicativo móvel desenvolvido para o sistema *three-way* foi também empregado para controlar tanto a ativação e desativação quanto o ajuste do brilho das lâmpadas, garantindo uma solução unificada e intuitiva para o usuário.

No desenvolvimento de ambos os sistemas, o microcontrolador Arduino desempenhou um papel central, facilitando a integração dos componentes eletrônicos e a programação das funcionalidades específicas. A comunicação entre o aplicativo móvel e o Arduino foi realizada através de um módulo Bluetooth, enquanto o controle de brilho da lâmpada na segunda parte do projeto foi alcançado por meio de um módulo *dimmer*.

4. RESULTADO E DISCUSSÃO

Os resultados obtidos, conforme a metodologia deste projeto na implementação dos sistemas de automação residencial para o controle de iluminação, estão ligados às etapas fundamentais. Essas etapas incluem:

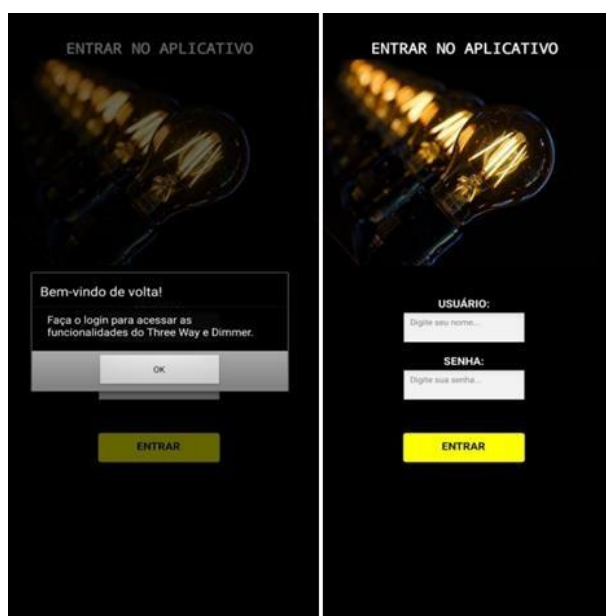
4.1.1 Desenvolvimento do Aplicativo com o App Inventor:

Utilizando a plataforma App Inventor, foi criado um aplicativo móvel dedicado para controlar os sistemas de automação residencial. Esse aplicativo foi projetado para funcionar em dispositivos móveis, como smartphones e tablets, e permite aos usuários controlarem remotamente as lâmpadas conectadas aos sistemas.

O aplicativo foi desenvolvido com três telas distintas: a tela inicial, que serve como tela de login, a tela do sistema *three-way* e a tela do sistema *dimmer*. Representadas pela figura 20, figura 22 e figura 29, respectivamente.

- **Tela Inicial/Login: (Figura 20).**

Figura 20 - Interface da Tela Inicial/Login do App.



Fonte: Autoria própria (2024).

Esta tela dá as boas-vindas aos usuários ao abrir o aplicativo, fornecendo informações sobre seu funcionamento e permitindo o acesso às funcionalidades principais após o login. Após fazer o login, o usuário será direcionado para a tela correspondente ao sistema *three-way*.

Figura 21 - Programação em Blocos da Tela de Login/Inicial.



Fonte: Autoria própria (2024).

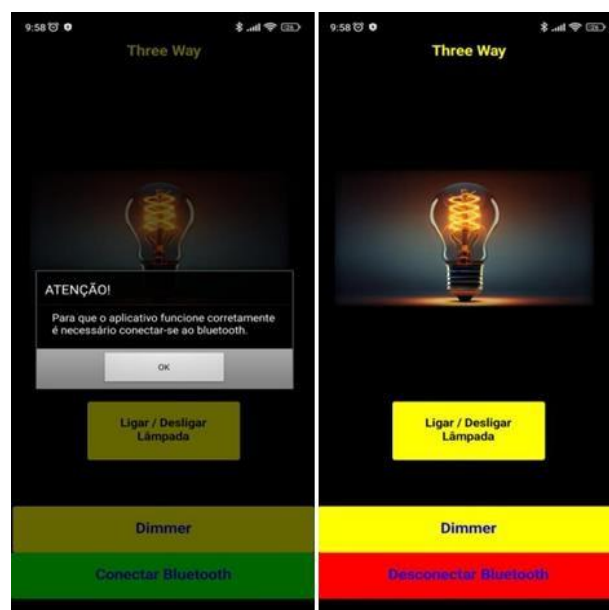
A programação em bloco é bastante intuitiva, como demonstra na figura 20, ao abrir o aplicativo, a tela de login ('Screen1') é exibida com uma caixa de mensagem dizendo "Bem-vindo de volta!" e uma mensagem abaixo que orienta o usuário a fazer login para acessar as funcionalidades do *Three Way* e *Dimmer*.

Quando o usuário clica no botão "Entrar", o aplicativo verifica se o nome de usuário e a senha inseridos correspondem aos valores esperados. O programa compara o nome de usuário e a senha com as seguintes condições:

Se o nome de usuário for "automação" ou "AUTOMAÇÃO" e a senha for "3way" ou "3WAY", então a tela do sistema *three-way* ('Screen3') é aberta, dando acesso às funcionalidades do sistema. Se não, uma mensagem de erro é exibida, solicitando que o usuário insira as informações corretas para acessar o sistema.

- **Tela do Sistema *Three-Way*: (Figura 22).**

Figura 22 - Interface da Tela do *Three-Way* do App.

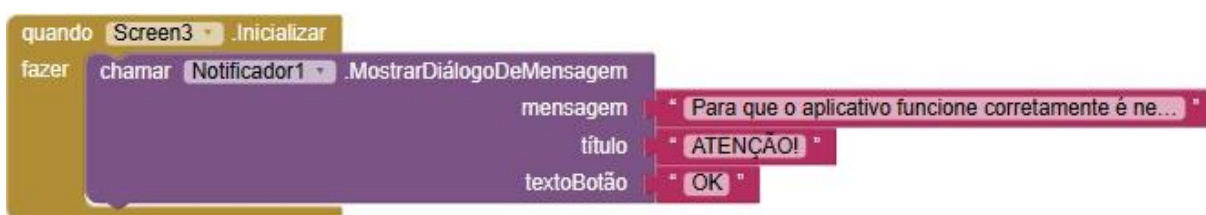


Fonte: Autoria própria (2024).

Nesta tela, o usuário tem acesso aos controles do sistema *three-way* de uma lâmpada. Para garantir o funcionamento correto do sistema, é necessário conectar-se ao Bluetooth. Essa conexão permite o controle remoto da lâmpada através do aplicativo.

O processo de programação detalhado em cada bloco, estão ilustrados nos passos abaixo:

Figura 23 – Blocos da tela do *Three Way*.



Fonte: Autoria própria (2024).

1. Ao iniciar a 'Screen3' (tela do sistema *three-way*): (Figura 23).

- Uma mensagem é exibida notificando que é necessário conectar-se ao Bluetooth para o correto funcionamento do aplicativo.

Figura 24 - Blocos da tela do *Three Way*.

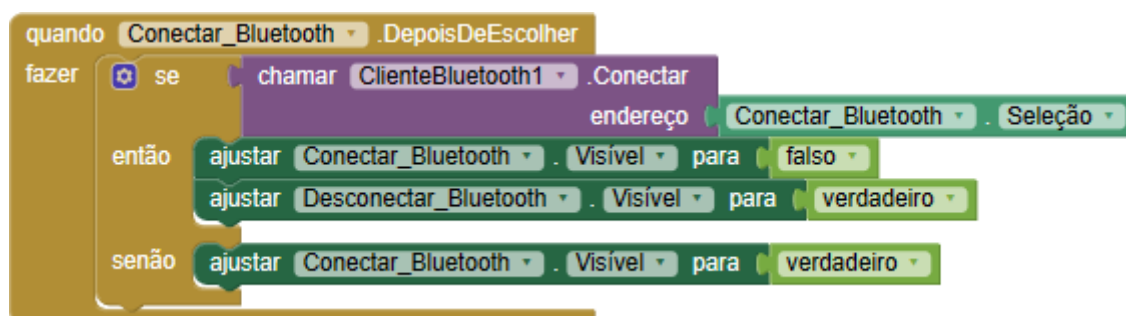


Fonte: Autoria própria (2024).

2. Antes de clicar no botão "Conectar Bluetooth": (Figura 24).

- Os elementos da tela são ajustados para facilitar a conexão.

Figura 25 - Blocos da tela do *Three Way*.

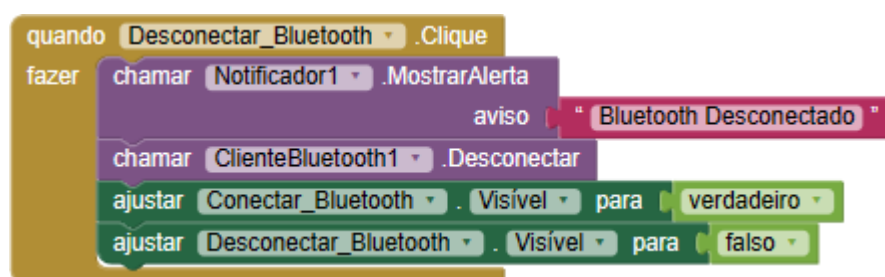


Fonte: Autoria própria (2024).

3. Quando o usuário clica no botão "Conectar Bluetooth" e escolhe um dispositivo disponível para conexão: (Figura 25).

- Se a conexão for bem-sucedida:
- Então, o botão "Conectar Bluetooth" desaparece e o botão "Desconectar Bluetooth" é exibido, indicando que o Bluetooth está conectado e pronto para uso.
- Se não (a conexão falhar):
- O botão "Conectar Bluetooth" permanece visível para permitir outra tentativa de conexão.

Figura 26 - Blocos da tela do *Three Way*.

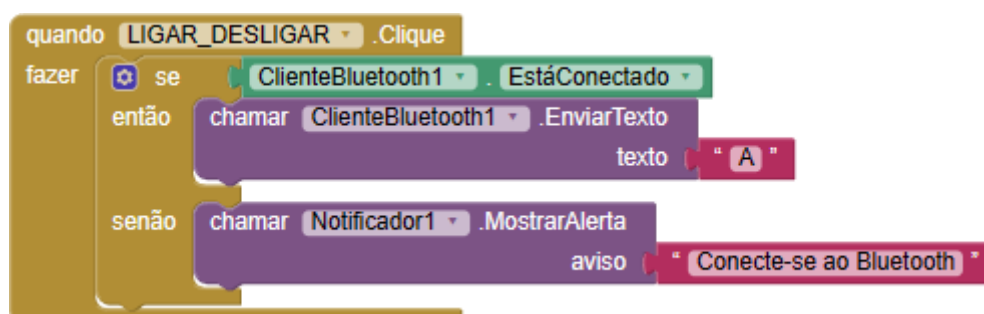


Fonte: Autoria própria (2024).

4. Quando o usuário clica no botão "Desconectar Bluetooth": (Figura 26).

- Uma notificação é exibida informando que o Bluetooth foi desconectado.
- O botão "Conectar Bluetooth" volta a ser exibido.

Figura 27 - Blocos da tela do *Three Way*.



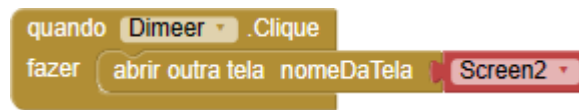
Fonte: Autoria própria (2024).

5. Quando o usuário clica no botão "Ligar/Desligar Lâmpada": (Figura 27).

- O aplicativo verifica se o Bluetooth está conectado.

- Se estiver conectado:
- Uma mensagem "A" é enviada para o módulo Bluetooth emparelhado com o Arduino.
- Se não (o Bluetooth não estiver conectado):
- Uma mensagem é exibida no aplicativo, informando ao usuário para conectar-se ao Bluetooth para que o aplicativo funcione corretamente.

Figura 28 - Blocos da tela do *Three Way*.



Fonte: Autoria própria (2024).

6. Quando o botão "*Dimmer*" for clicado: (Figura 28).

-O aplicativo redirecionará o usuário para a 'Screen2', que é a tela do controle de brilho.

- **Tela de Controle de Brilho: (Figura 29).**

Figura 29 - Interface da Tela de Controle de Brilho do App.



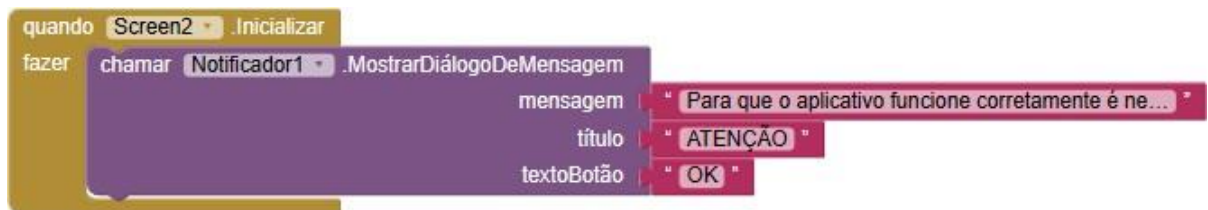
Fonte: Autoria própria (2024).

Nessa terceira interface, o usuário pode ajustar o ligar, desligar e o brilho da lâmpada conforme suas preferências. Para garantir a funcionalidade desse sistema, é necessário

estabelecer uma conexão Bluetooth. Essa conexão permite que o usuário controle a lâmpada remotamente e ajuste o brilho conforme sua necessidade.

O processo de programação detalhado em cada bloco, estão ilustrados nos passos abaixo:

Figura 30 - Blocos da tela do Controle de Brilho.



Fonte: Autoria própria (2024).

1. Ao Iniciar a Screen 2 (Tela do Controle de Brilho): (Figura 30).

- Mostrar mensagem para conectar-se ao Bluetooth.

Figura 31 - Blocos da tela do Controle de Brilho.

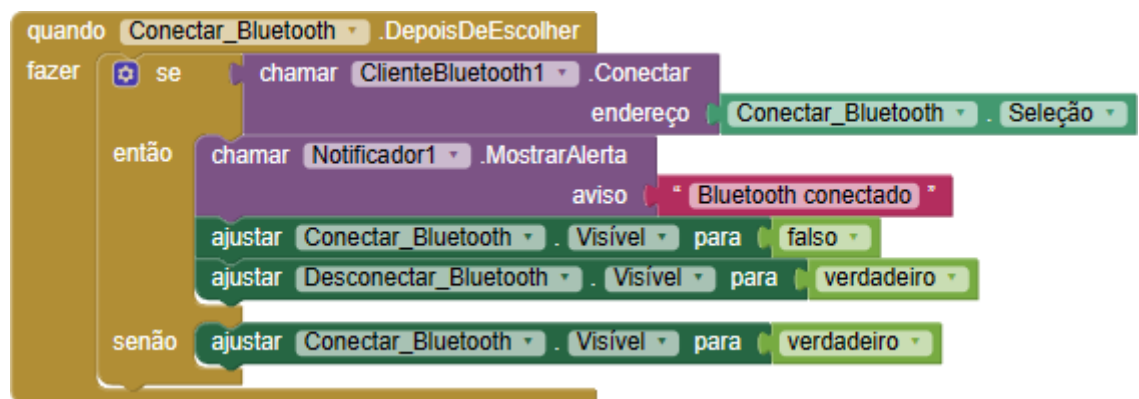


Fonte: Autoria própria (2024).

2. Antes de Clicar no Botão "Conectar Bluetooth": (Figura 31).

- Ajustar a interface para facilitar a conexão.

Figura 32 - Blocos da tela do Controle de Brilho.

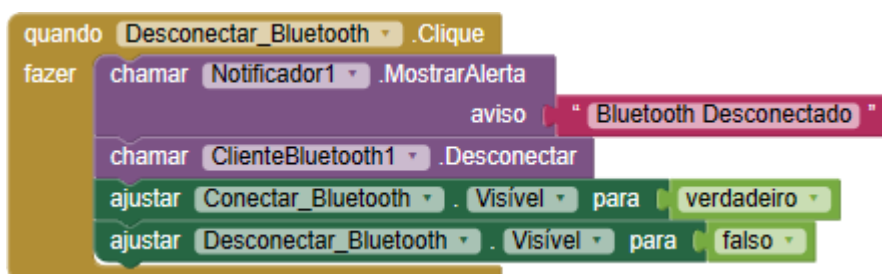


Fonte: Autoria própria (2024).

3. Quando o Usuário Clica no Botão "Conectar Bluetooth" e Seleciona um Dispositivo Disponível: (Figura 32).

- Se a conexão for bem-sucedida:
- Ocultar o botão "Conectar Bluetooth".
- Mostrar o botão "Desconectar Bluetooth".
- Se não:
- Manter visível o botão "Conectar Bluetooth" para outra tentativa.

Figura 33 - Blocos da tela do Controle de Brilho.

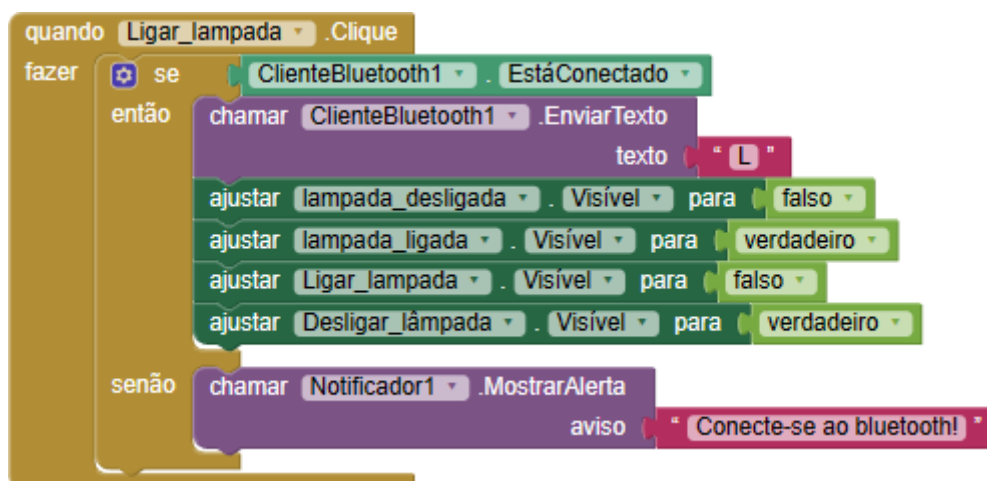


Fonte: Autoria própria (2024).

4. Quando o Usuário Clica no Botão "Desconectar Bluetooth": (Figura 33).

- Exibir notificação informando que o Bluetooth foi desconectado.
- Ocultar o botão "Desconectar Bluetooth".
- Mostrar novamente o botão "Conectar Bluetooth".

Figura 34 - Blocos da tela do Controle de Brilho.

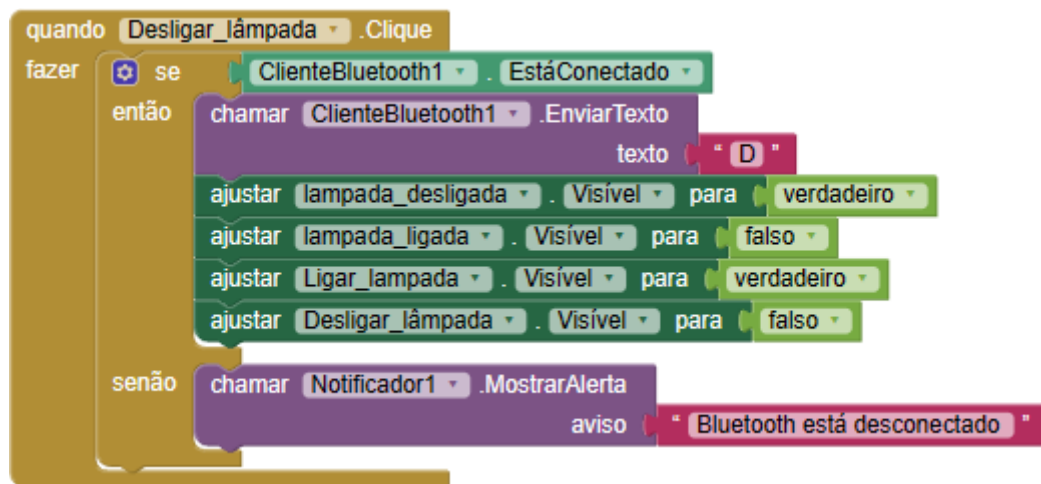


Fonte: Autoria própria (2024).

5. Ao Clicar no Botão "Ligar Lâmpada": (Figura 34)

- Verificar se o Bluetooth está conectado.
- Se estiver conectado:
 - Enviar o comando "L" via Bluetooth para ligar a lâmpada.
 - Alterar a imagem para mostrar a lâmpada ligada.
- Se não estiver conectado:
 - Exibir notificação para conectar-se ao Bluetooth.

Figura 35 - Blocos da tela do Controle de Brilho.

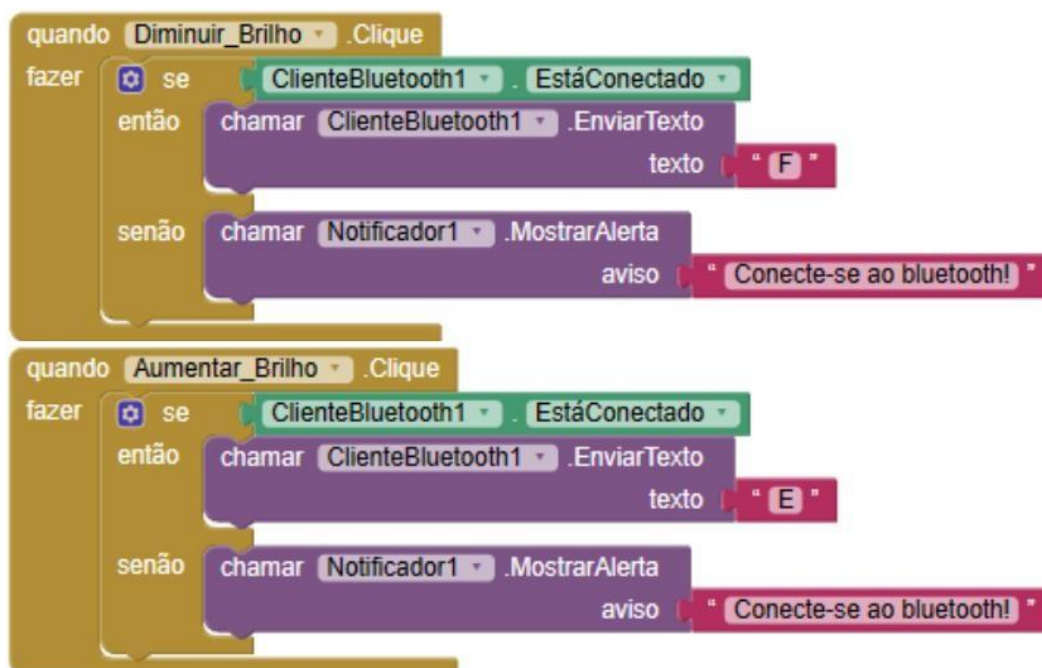


Fonte: Autoria própria (2024).

6. Ao Clicar no Botão "Desligar Lâmpada": (Figura 35).

- Verificar se o Bluetooth está conectado.
- Se estiver conectado:
 - Enviar o comando "D" via Bluetooth para desligar a lâmpada.
 - Alterar a imagem para mostrar a lâmpada desligada.
- Se não estiver conectado:
 - Exibir notificação para conectar-se ao Bluetooth.

Figura 36 - Blocos da tela do Controle de Brilho.

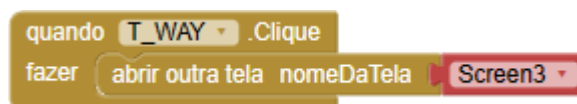


Fonte: Autoria própria (2024).

7. Ao Clicar nos Botões "Aumentar Brilho" ou "Diminuir Brilho": (Figura 36)

- Verificar se o Bluetooth está conectado.
- Se estiver conectado:
 - Enviar o comando correspondente ("E" para aumentar ou "F" para diminuir) via Bluetooth para ajustar o brilho.
- Se não estiver conectado:
 - Exibir notificação para conectar-se ao Bluetooth.

Figura 37 - Blocos da tela do Controle de Brilho.



Fonte: Autoria própria (2024).

8. Ao Clicar no Botão "Three Way": (Figura 37).

- Redirecionar o usuário para a tela do sistema *three-way* (Screen 3).

4.1.2 Programação no Arduino IDE:

Abordando detalhadamente o desenvolvimento da programação no Arduino, criada para controlar a luminosidade de uma lâmpada e o acionamento de um relé. Será explicado passo a passo, desde a configuração dos pinos até a interpretação dos comandos recebidos via comunicação serial. Vale ressaltar que os // que estão nas figuras são comentários que servem para explicar também a função do código.

Figura 38 - Parte I do código completo.



Fonte: Autoria própria (2024).

Explicando a figura 38, que é a primeira parte do código, é utilizado a biblioteca SoftwareSerial para comunicação serial via *software* e define os pinos utilizados, incluindo os pinos para o *dimmer*, o relé e a comunicação Bluetooth. Um objeto SoftwareSerial chamado bluetooth é criado para essa comunicação. As variáveis globais são definidas para armazenar o valor da luminosidade e o estado do *dimmer*.

A função zeroCross é definida e chamada quando há uma interrupção na detecção da corrente alternada (AC) ao cruzar o zero. Isso possibilita a sincronização do controle do *dimmer* com a rede elétrica. Dentro desta função, é verificado se o *dimmer* está ligado, garantindo que o controle ocorra somente quando o dispositivo estiver ativo. Em seguida, a luminosidade é limitada entre 10 e 90 usando a função constrain(), garantindo que permaneça em uma faixa segura. O tempo de delay necessário para ajustar a luminosidade é calculado com base na

luminosidade atual, multiplicando um valor constante (8300 microssegundos, ajustado para a frequência de 60Hz da rede elétrica) pela proporção da luminosidade em relação ao máximo brilho. Posteriormente, um delay é realizado utilizando `delayMicroseconds()` com o valor calculado. Por fim, o pino do *dimmer* é alternado entre ligado e desligado com pequenos delays adicionais para assegurar o controle preciso da luminosidade.

Figura 39 - Parte II do código completo.



Fonte: Autoria própria (2024).

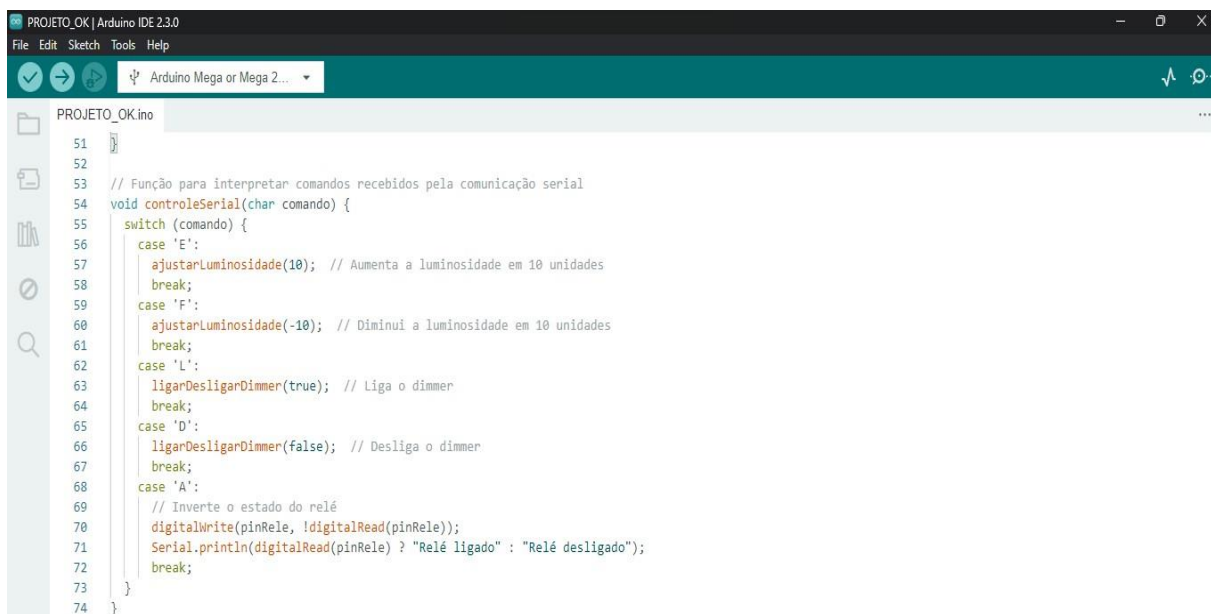
A segunda parte do código representado pela figura 39, contém duas funções e parte da função `setup()`. A função `ajustarLuminosidade(int delta)` é responsável por ajustar a luminosidade com base em um delta recebido como parâmetro. Ela adiciona o valor do delta à variável `luminosidade`, garantindo que o resultado esteja dentro do intervalo entre 10 e 90 usando a função `constrain()`. Em seguida, imprime uma mensagem indicando se a luminosidade foi aumentada ou diminuída, junto com o novo valor da luminosidade na porta serial.

A segunda função, `ligarDesligarDimmer(bool ligar)`, é usada para ligar ou desligar o *dimmer*. Ela recebe um parâmetro booleano que indica se o *dimmer* deve ser ligado (`true`) ou desligado (`false`). Dependendo do valor recebido, atualiza o estado da variável `dimmerLigado` e imprime uma mensagem indicando se o *dimmer* está ligado ou desligado na porta serial.

Na função `setup()`, são realizadas as configurações iniciais do Arduino. A comunicação serial é iniciada com uma taxa de transmissão de 9600 bps tanto para a porta serial padrão quanto para o módulo Bluetooth. Os pinos `PINO_DIM` e `pinRele` são configurados como saída,

sendo o primeiro usado para controlar o *dimmer* e o segundo para o relé. O relé é inicialmente desligado ao definir o estado do pino pinRele como LOW. Por fim, é configurado uma interrupção para o pino digital 2 (INT0) do Arduino, que será acionada no evento de subida do sinal (RISING), associada à função zeroCross().

Figura 40 - Parte III do código completo.

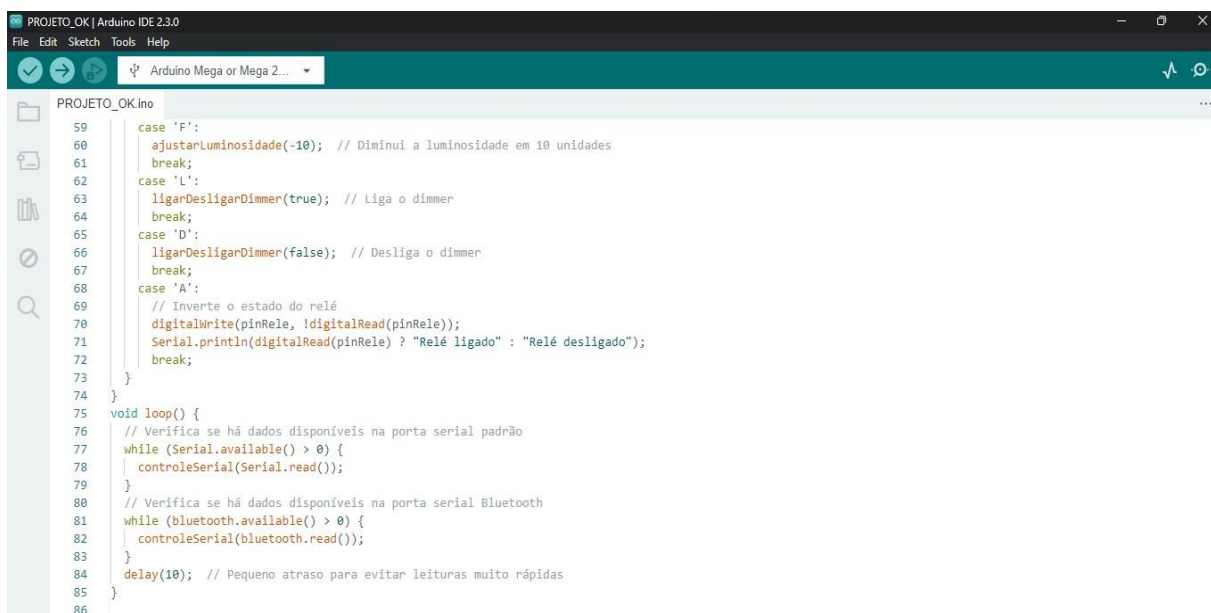


```
51
52
53 // Função para interpretar comandos recebidos pela comunicação serial
54 void controleSerial(char comando) {
55     switch (comando) {
56         case 'E':
57             ajustarLuminosidade(10); // Aumenta a luminosidade em 10 unidades
58             break;
59         case 'F':
60             ajustarLuminosidade(-10); // Diminui a luminosidade em 10 unidades
61             break;
62         case 'L':
63             ligarDesligarDimmer(true); // Liga o dimmer
64             break;
65         case 'D':
66             ligarDesligarDimmer(false); // Desliga o dimmer
67             break;
68         case 'A':
69             // Inverte o estado do relé
70             digitalWrite(pinRele, !digitalRead(pinRele));
71             Serial.println(digitalRead(pinRele) ? "Relé ligado" : "Relé desligado");
72             break;
73     }
74 }
```

Fonte: Autoria própria (2024).

Partindo para terceira parte do código demonstrado pela figura 40. A função controleSerial(char comando) é responsável por interpretar os comandos recebidos pela comunicação serial, tanto pela porta serial padrão quanto pela porta serial Bluetooth. Utilizando uma estrutura de switch-case, ela analisa cada comando recebido e executa a ação correspondente. Quando recebe o comando 'E', a função aumenta a luminosidade em 10 unidades, enquanto o comando 'F' diminui a luminosidade em 10 unidades. Os comandos 'L' e 'D' ligam e desligam o *dimmer*, respectivamente, e o comando 'A' inverte o estado do relé. Além disso, a função fornece feedback ao usuário, imprimindo mensagens indicativas das ações realizadas, como "Luminosidade aumentada" ou "*Dimmer* ligado", garantindo uma interação no monitor serial da IDE Arduino.

Figura 41 - Parte final do código completo.

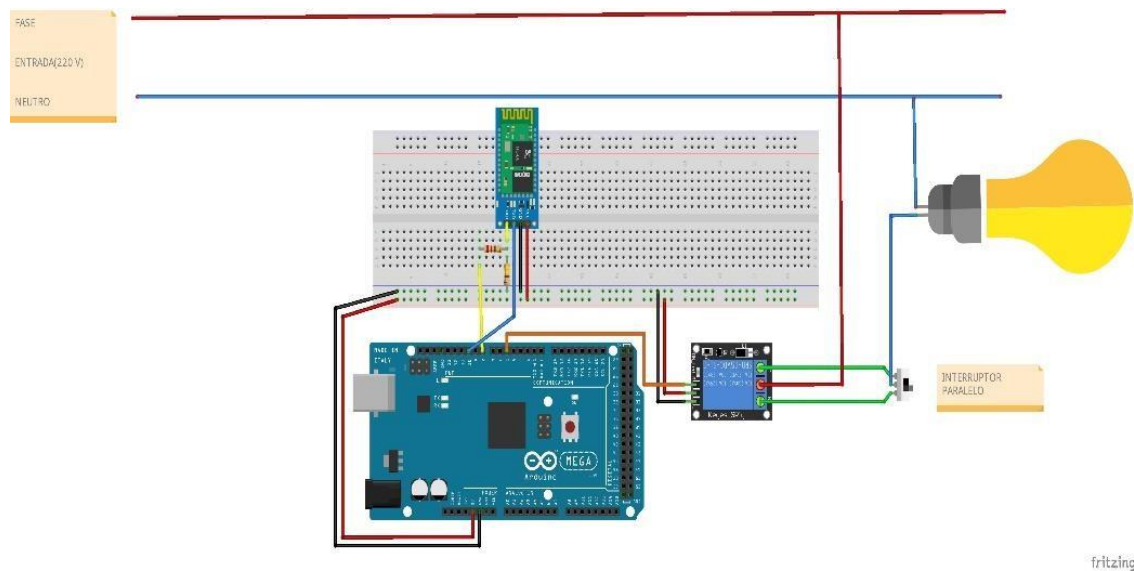


Fonte: Autoria própria (2024).

Por fim, a parte final do código apresentado na figura 41. No void loop (), o programa verifica continuamente se há dados disponíveis tanto na porta serial padrão quanto na porta serial Bluetooth. Se houver dados disponíveis na porta serial padrão, o programa chama a função controleSerial() para interpretar e processar os comandos recebidos. Da mesma forma, se houver dados disponíveis na porta serial Bluetooth, a função controleSerial() é chamada para processar os comandos recebidos. Após esse processo de verificação e processamento dos dados nas duas portas seriais, é introduzido um pequeno atraso de 10 milissegundos utilizando a função delay (10). Esse atraso é necessário para evitar leituras muito rápidas e para garantir uma operação mais estável do programa.

4.1.3 Montagem do Circuito para o Sistema *Three-Way*:

O circuito para o sistema *three-way* foi montado utilizando um interruptor paralelo, um relé, um Arduino, um módulo Bluetooth HC-06, uma protoboard e *jumpers* para fazer as conexões necessárias, como mostra na figura 42. Após a montagem do circuito, ele está pronto para funcionamento. Com essa configuração, é possível controlar remotamente uma lâmpada, por meio do aplicativo móvel conectado ao módulo Bluetooth ou interruptor físico para alternar o estado da lâmpada.

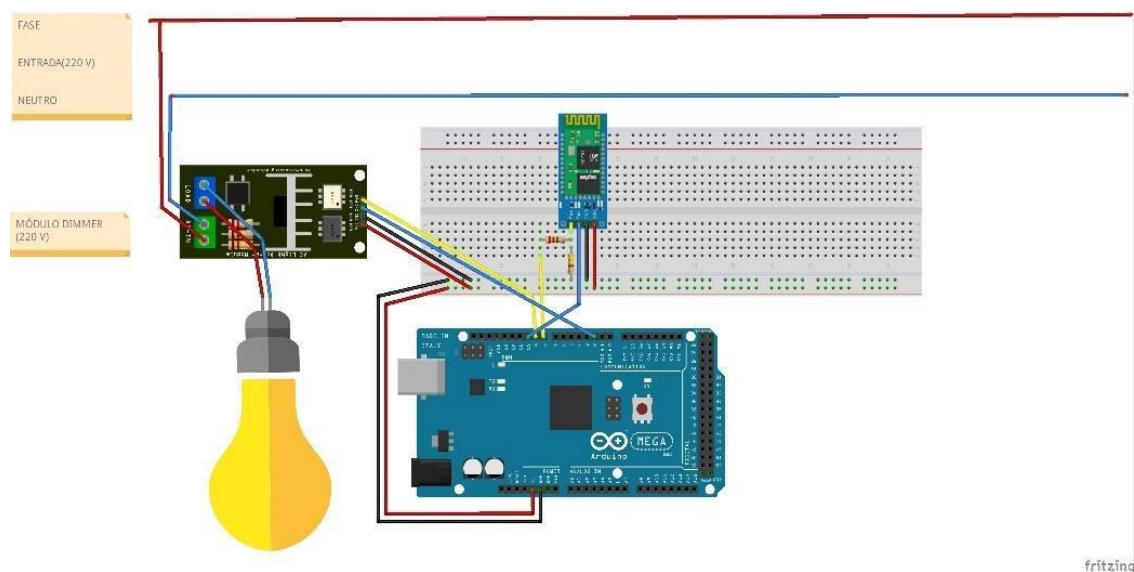
Figura 42 - Circuito para o Sistema *Three-Way*.

Fonte: Autoria própria (2024).

4.1.4 Montagem do Circuito para o Controle de Brilho da Lâmpada:

Para o controle de brilho da lâmpada, representado na figura 43, foi montado um circuito utilizando um módulo *dimmer*, o microcontrolador Arduino, uma lâmpada, os *jumpers* e o mesmo modulo Bluetooth do sistema *three-way*. Esse circuito permite ajustar a intensidade luminosa da lâmpada de acordo com as preferências do usuário.

Figura 43 - Circuito para o Controle de Brilho da Lâmpada.

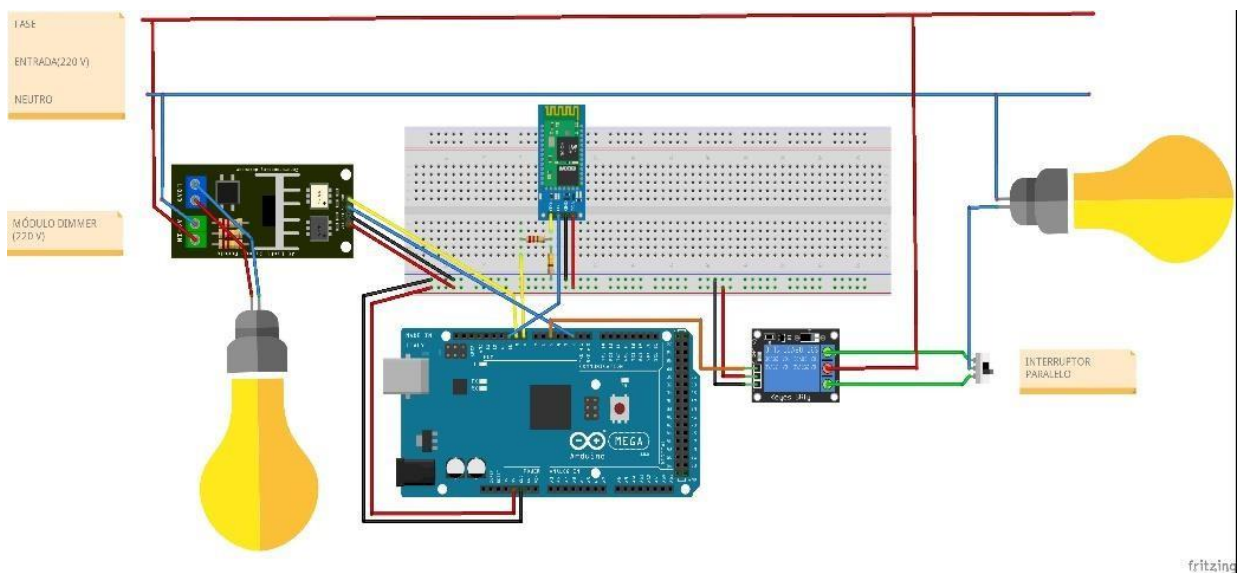


Fonte: Autoria própria (2024).

4.1.5 Testes e Ajustes:

Após a montagem dos dois circuitos e a programação do Arduino, foram realizados testes para validar o funcionamento dos sistemas conforme o planejado. Durante essa etapa, ocorreram ajustes para garantir a integração entre o aplicativo móvel e a junção dos circuitos eletrônicos, conforme a figura 44, visando assegurar o funcionamento correto. Esses testes permitiram identificar e corrigir problemas de funcionamento dos dispositivos, garantindo assim que o sistema operasse de maneira eficaz.

Figura 44 - Circuito Completo *Three-Way* e *Dimmer*.



Fonte: Autoria própria (2024).

Os sistemas implementados atenderam às expectativas do projeto, fornecendo soluções eficazes para o controle de iluminação residencial. A utilização de tecnologias como o Arduino e o Bluetooth possibilitou uma implementação versátil e eficiente, ao passo que a integração com um aplicativo móvel proporcionou uma experiência intuitiva para o usuário.

5. CONSIDERAÇÕES FINAIS

- Conclusão

Em conclusão, este trabalho demonstrou alguns benefícios da automação residencial, especialmente no que diz respeito ao controle de iluminação. Através da implementação de sistemas baseados em tecnologias como Arduino integrado a aplicativos móveis, é possível não apenas oferecer praticidade aos usuários, mas também alcançar redução no consumo de energia elétrica. Além disso, o uso do Arduino proporciona uma excelente ferramenta para a familiarização e aprendizado na criação de projetos de automação residencial, contribuindo para o desenvolvimento de habilidades técnicas e criativas.

- Potenciais melhorias no projeto

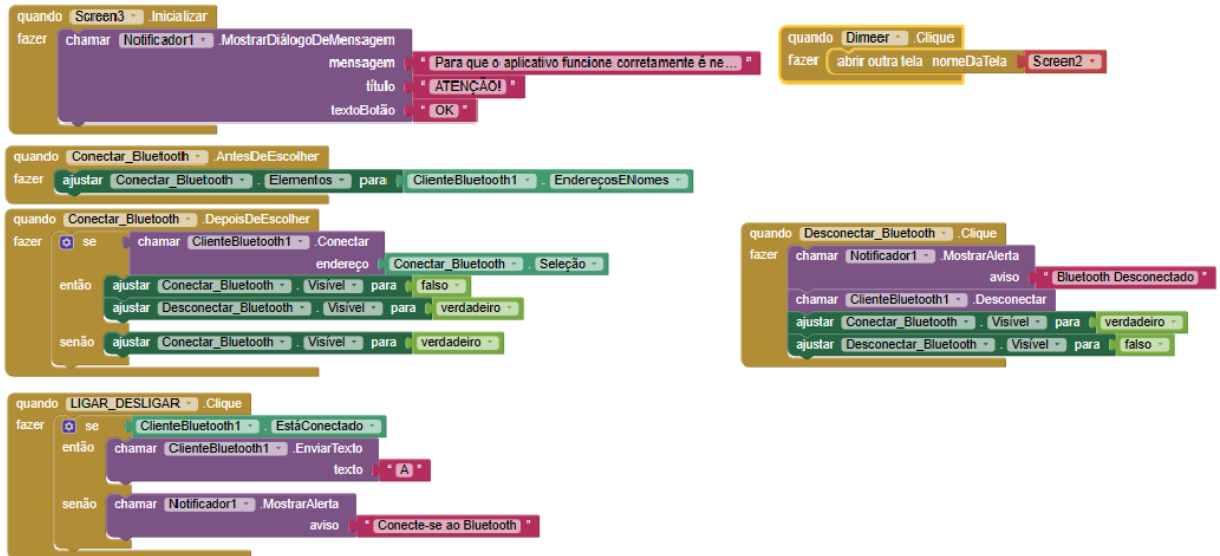
Além das funcionalidades demonstradas, há espaço para melhorias significativas no projeto de automação residencial apresentado, como a simplificação do projeto com o uso de um Arduino mais básico, a integração de placas de circuito impresso (PCBs) para a simplificação do circuito e a incorporação de comunicação Wi-Fi para ampliar o alcance, pois existe uma limitação na curta distância da conexão Bluetooth.

REFERÊNCIAS

- BOLZANI, Caio Augustus Moraes. Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes – Edição Revisada – São Paulo, 2010.
- Coletânea Estudos em engenharia e inovação / Thiago Santana de Oliveira, Fabiana Aquino de Moraes Rêgo, Mirian Nunes de Carvalho Nunes e Carolina Gomes Araújo Garreto (Org). São Luís - Editora Pascal, 2023.
- RIBEIRO, Carlos Eduardo. **DOMÓTICA**: Viabilidade da Automação Residencial. 2018. Trabalho de Conclusão de Curso - Curso de Engenharia Elétrica - Centro Universitário Sul de Minas, Minas Gerais, 2018.
- QUESADA, Ricardo Carvalho. Controle e automação de processos industriais / Ricardo Carvalho Quesada. – Londrina: Editora e Distribuidora Educacional S.A., 2017.
- Automação residencial: o que é e como investir em casas inteligentes?. SEBRAE, 2023. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/automacao-residencial-o-que-e-e-como-investir-em-casas-inteligentes,75659ebcdd4c5810VgnVCM1000001b00320aRCRD>. Acesso em: 21. out. 2023.
- PERIN, Erick. A importância da automação residencial para pessoas comuns. OKTOPUS, 2022. Disponível em: <https://oktopus.cloud/blog/porque-automatizar-minha-casa>. Acesso em: 21, Out. 2023.
- SANTOS, Vieira. A evolução da automação residencial. VIEIRA SANTOS, 2017. Disponível em: <https://vieirasantos.com/a-evolucao-da-automacao-residencial/>. Acesso em: 21. out. 2023.
- LUZ, Helder José. Sistema de Automação Residencial/Predial utilizando Arduino e sistema operacional Android / Helder José Santos Luz. - 2018.
- Crie aplicativos aberto para todos com o Google App Inventor. Tech Tudo. Disponível em: <https://www.techtudo.com.br/tudo-sobre/google-app-inventor/>. Acesso em: 25. out. 2023.
- BREDA, Danilo Costa Automação Residencial: Aplicação de Redes Neurais Artificiais para Controle de Climatização. / Danilo Costa Breda; orientador: Prof. Ms. Rodolfo Barros Chiaramonte. Marília, SP: [s.n.], 2016.
- COSTA, Érika Isabel da. Desenvolvimento de um protótipo para automação residencial usando dispositivos controlados via smartphone / Érika Isabel da Costa. 2019. 54f. Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Interdisciplinar em ciência e tecnologia. Caraúbas, 2019.
- STRAUB, Matheus Gebert. Interruptor paralelo no controle de lâmpada com arduino e relé. USINAINFO, 2021. Disponível em: <https://tecnoblog.net/responde/referencia-site-abnt-artigos/>. Acesso em 12. fev. 2024.

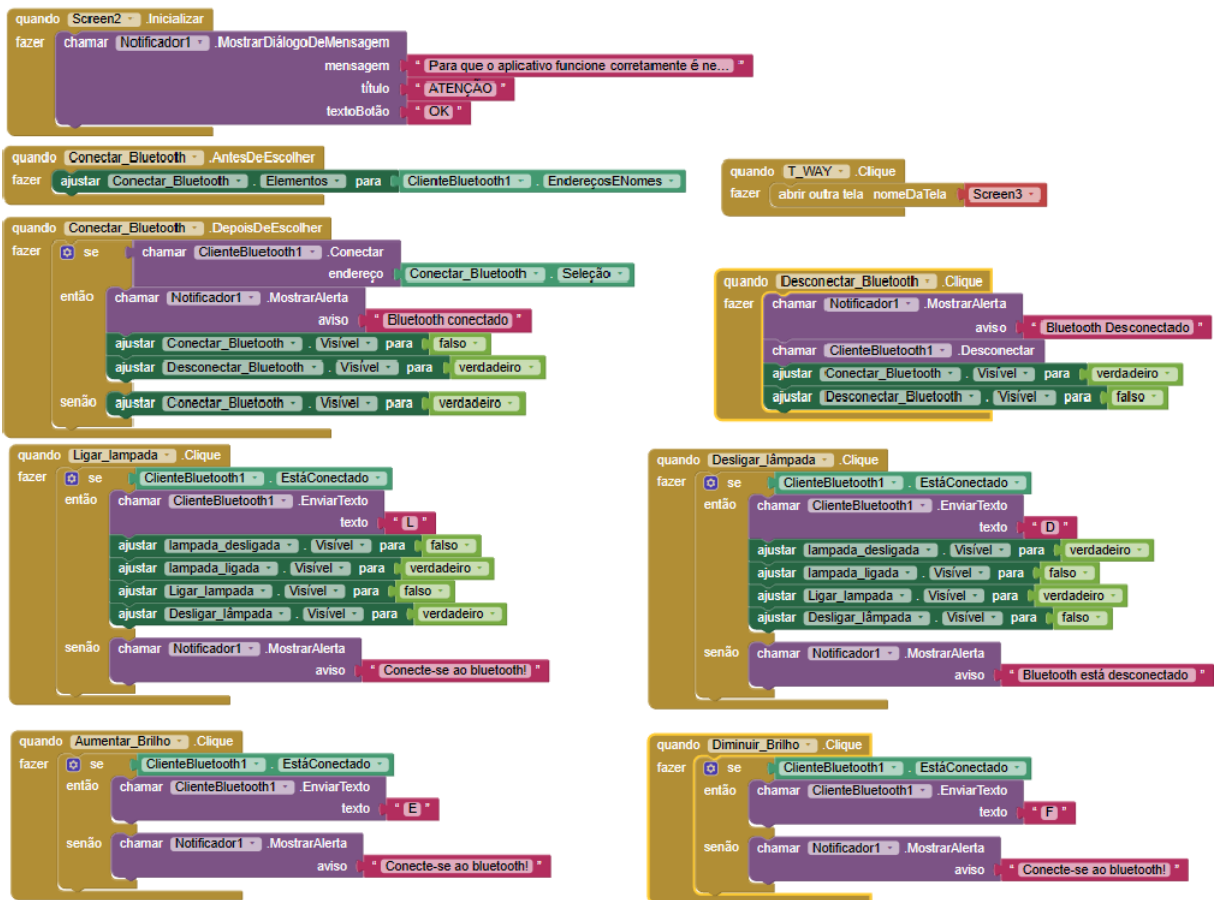
STRAUB, Matheus Gebert. PROJETO ESP32 DIMMER – AUTOMAÇÃO PARA CONTROLE DE LUZ. USINAINFO, 2020. Disponível em: <https://www.usinainfo.com.br/blog/projeto-esp32-dimmer-automacao-para-controle-de-luz/>. Acesso em 16. fev. 2024.

APÊNDICE A – BLOCOS DE PROGRAMAÇÃO DA TELA DO *THREE WAY*



Fonte: Autoria própria (2024).

APÊNDICE B – BLOCOS DE PROGRAMAÇÃO DA TELA DO *DIMMER*



Fonte: Autoria própria (2024).

APÊNDICE C – CÓDIGO PARA A PROGRAMAÇÃO DO ARDUINO



```

1  #include <SoftwareSerial.h> // Inclui a biblioteca SoftwareSerial para comunicação serial via software
2
3  // Define os pinos utilizados
4  #define PINO_DIM 9 // Pino ao qual o dimmer está conectado
5  #define pinRele 5 // Pino ao qual o relé está conectado
6  #define pinBluetoothTX 8 // Pino de transmissão do Arduino, logo é conectado ao Rx do Bluetooth.
7  #define pinBluetoothRX 10 // Pino de recepção do Arduino, logo é conectado ao Tx do Bluetooth.
8
9  // Cria um objeto SoftwareSerial para comunicação Bluetooth
10 SoftwareSerial bluetooth(pinBluetoothRX, pinBluetoothTX);
11
12 // Variáveis globais
13 volatile long luminosidade = 10; // Variável para armazenar o valor da luminosidade
14 bool dimmerLigado = false; // Flag para indicar se o dimmer está ligado
15
16 // Função chamada na interrupção da cruz zero
17 void zeroCross() {
18   if (dimmerLigado) {
19     luminosidade = constrain(luminosidade, 10, 90); // Limita a luminosidade entre 10 e 90
20     long t1 = 8300L * (100L - luminosidade) / 100L; // Calcula o tempo de delay necessário
21     delayMicroseconds(t1);
22     digitalWrite(PINO_DIM, HIGH); // Liga o pino de dimmer
23     delayMicroseconds(6);
24     digitalWrite(PINO_DIM, LOW); // Desliga o pino de dimmer
25   }
26 }
27
28 // Função para ajustar a luminosidade com base em um delta
29 void ajustarLuminosidade(int delta) {
30   luminosidade = constrain(luminosidade + delta, 10, 90); // Ajusta a luminosidade com base no delta
31   Serial.println(delta > 0 ? "Luminosidade aumentada || " : "Luminosidade diminuída || ");
32   Serial.println("luminosidade= " + String(luminosidade));
33 }
34
35 // Função para ligar ou desligar o dimmer
36 void ligarDesligarDimmer(bool ligar) {
37   dimmerLigado = ligar; // Atualiza o estado do dimmer
38   Serial.println(dimmerLigado ? "Dimmer ligado" : "Dimmer desligado");
39 }
40
41 void setup() {
42   Serial.begin(9600); // Inicializa a comunicação serial
43   bluetooth.begin(9600); // Inicializa o módulo Bluetooth
44   pinMode(PINO_DIM, OUTPUT); // Configura o pino do dimmer como saída
45   pinMode(pinRele, OUTPUT); // Configura o pino do relé como saída
46   digitalWrite(pinRele, LOW); // Inicializa o relé desligado
47
48   // Configura a interrupção na cruz zero (Zero Cross Detection)
49   // A interrupção está associada ao pino digital 2 (INT0) no Arduino Uno
50   attachInterrupt(0, zeroCross, RISING);
51 }
52
53 // Função para interpretar comandos recebidos pela comunicação serial
54 void controleSerial(char comando) {
55   switch (comando) {
56     case 'E':
57       ajustarLuminosidade(10); // Aumenta a luminosidade em 10 unidades
58       break;
59     case 'F':
60       ajustarLuminosidade(-10); // Diminui a luminosidade em 10 unidades
61       break;
62     case 'L':
63       ligarDesligarDimmer(true); // Liga o dimmer
64       break;
65     case 'D':
66       ligarDesligarDimmer(false); // Desliga o dimmer
67       break;
68     case 'A':
69       // Inverte o estado do relé
70       digitalWrite(pinRele, !digitalRead(pinRele));
71       Serial.println(digitalRead(pinRele) ? "Relé ligado" : "Relé desligado");
72       break;
73   }
74 }

```

```
75 void loop() {  
76   // Verifica se há dados disponíveis na porta serial padrão  
77   while (Serial.available() > 0) {  
78     controleSerial(Serial.read());  
79   }  
80   // Verifica se há dados disponíveis na porta serial Bluetooth  
81   while (bluetooth.available() > 0) {  
82     controleSerial(bluetooth.read());  
83   }  
84   delay(10); // Pequeno atraso para evitar leituras muito rápidas  
85 }  
86
```

Fonte: Autoria própria (2024).