



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Paulo José Duarte

**Sistema de Controle e Automação de Iluminação via Smartphone com
Ênfase em Eficiência Energética**

Araranguá
2021

Paulo José Duarte

**Sistema de Controle e Automação de Iluminação via Smartphone com
Ênfase em Eficiência Energética**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Dr. Roderval Marcelino

Araranguá
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Duarte, Paulo José

Sistema de Controle e Automação de Iluminação via
Smartphone com Ênfase em Eficiência Energética / Paulo
José Duarte ; orientador, Roderval Marcelino, 2021.
26 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2021.

Inclui referências.

1. Engenharia de Computação. 2. Automação Residencial. 3.
Eficiência Energética. 4. Controle de Iluminação. I.
Marcelino, Roderval. II. Universidade Federal de Santa
Catarina. Graduação em Engenharia de Computação. III. Título.

Paulo José Duarte

**Sistema de Controle e Automação de Iluminação via Smartphone com
Ênfase em Eficiência Energética**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 27 de setembro de 2021.

Prof. Fabrício De Oliveira Ourique, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Dr. Roderval Marcelino
Orientador

Prof. Anderson Luiz Fernandes Perez, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Jim Lau, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof^a. Eliane Pozzebon, Dr^a.
Avaliador Suplente
Universidade Federal de Santa Catarina

Sistema de Controle e Automação de Iluminação via Smartphone com Ênfase em Eficiência Energética

Paulo José Duarte*

Roderval Marcelino†

2021, Setembro

Resumo

A energia elétrica é um fator essencial nas nossas vidas e devemos ter a preocupação da quantidade que estamos utilizando, principalmente em locais em que a fonte de energia não é renovável. A iluminação representa de 30% a 50% de toda a carga elétrica em prédios comerciais, se diminuirmos a quantidade de iluminação, ou a intensidade da iluminação, reduzimos consideravelmente o uso de energia elétrica. Com essa percepção, o presente trabalho retrata o desenvolvimento de um modelo de baixo custo com objetivo de controlar e automatizar uma lâmpada a fim de buscar a máxima eficiência energética, equilibrando a quantidade necessária de iluminação da lâmpada de acordo com a luz natural do ambiente, além da possibilidade de gerenciar e monitorar em tempo real a quantidade de energia elétrica que o sistema está utilizando através de um aplicativo que pode ser acessado por *smartphone*. Para a realizar os testes, foi construído um modelo para representar uma sala comercial e com a utilização de uma lâmpada LED e uma lâmpada incandescente, foi medido a quantidade de energia elétrica de cada uma das lâmpadas utilizando o sistema e também da forma comum, ou seja, no máximo. Os testes resultaram em uma economia de energia de 40% para lâmpadas LED e 47% para lâmpadas incandescentes. Conforme o esperado, o teste nos mostra a eficácia da utilização da automação residencial para a economia de energia.

Palavras-chaves: Automação Residencial. Eficiência Energética. Controle de Iluminação.

*paulo.jose.duarte@grad.ufsc.br

†roderval.marcelino@ufsc.br

Smartphone Lighting Control and Automation System with Emphasis on Energy Efficiency

Paulo José Duarte*

Roderval Marcelino†

2021, September

Abstract

Electricity is an essential factor in our lives and we must be concerned about the amount we are using, especially in places where the energy source is not renewable. Lighting represents 30% to 50% of the entire electrical load in commercial buildings, if we reduce the amount of lighting, or the intensity of lighting, we considerably reduce the use of electrical energy. With this perception, this work portrays the development of a low-cost model in order to control and automate a lamp in order to seek maximum energy efficiency, balancing the necessary amount of lamp lighting according to the natural light of the environment, in addition to the possibility of managing and monitoring in real time the amount of electricity that the system is using through an application that can be accessed by smartphone. To carry out the tests, a model was built to represent a commercial room and with the use of an LED lamp and an incandescent lamp, the amount of electrical energy of each of the lamps was measured using the system and also in the common way, that is, at most. The tests resulted in energy savings of 40% for LED lamps and 47% for incandescent lamps. As expected, the test shows us the effectiveness of using home automation for energy savings.

Key-words: Home Automation. Energy Efficiency. Lighting Control.

*paulo.jose.duarte@grad.ufsc.br

†roderval.marcelino@ufsc.br

1 Introdução

Especialmente em áreas onde a fonte de energia não é renovável e também pela energia estar cada vez mais cara, há sempre a preocupação da quantidade gasta de energia e como encontrar uma forma de economizar.

Porém a energia é muito mais do que apenas um valor em uma fatura, energia é um fator essencial nas nossas vidas, contamos com a energia para ter conforto, produtividade, lazer. Para sustentar essa qualidade de vida, precisamos a todo momento utilizar essa energia de maneira sustentável, a gestão de recursos energéticos não é apenas reduzir o consumo, mas também usufruir de maneira inteligente e eficiente (CALÇA, 2014).

De acordo com Pádua (2006), de 30% a 50% de toda a carga elétrica de um prédio comercial, se deve a iluminação. Podendo haver a possibilidade de acionar a luz ambiente de forma mais inteligente, através de sensores de presença.

A automação residencial, que também é referenciada como domótica, consiste na integração entre variados tipos de equipamentos como motores e sensores conversando entre si. Com isso podemos dizer que automação residencial é controlar de forma rápida e fácil os aparelhos, luzes, cortinas entre outras coisas apenas com um comando (CALÇA, 2014).

Sensores podem monitorar diversos fatores do ambiente como luminosidade, movimento, temperatura e umidade. O controle dessas funcionalidades é feito via *softwares* em aparelhos eletrônicos, como celulares ou computadores, mas em alguns casos também são utilizados *hardwares* dedicados, dependendo da complexidade do projeto (WILSON et al, 2017).

A automação em prédios e residências iniciaram nos anos 80, com o objetivo de aumentar o conforto, a segurança e a economia de energia elétrica (PÁDUA, 2006). Mesmo após 40 anos do início da automação residencial, o assunto ainda é tratado como artigo de luxo pela grande parte dos consumidores (LEMOS, 2016).

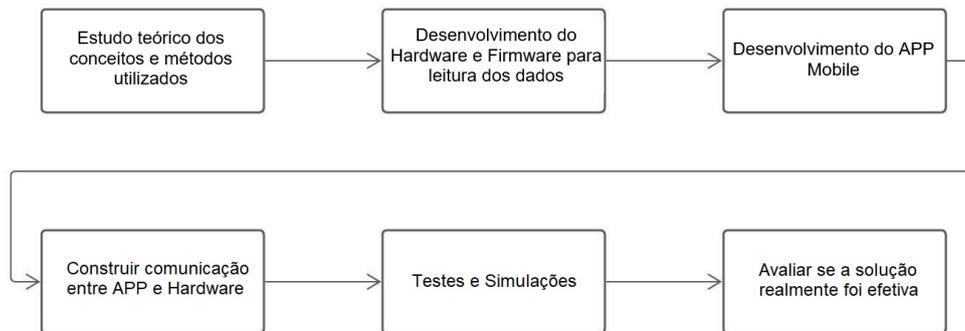
No Brasil, apesar de ser considerado de difícil acesso para a população em geral, segundo a Aureside (Associação Brasileira de Automação Residencial) o interesse de brasileiros por automação residencial vem crescendo. Uma pesquisa mostra que 78% dos entrevistados estariam interessados em automação residencial, número maior do que a média que ocorre no mundo, de 66% (CALÇA, 2014).

Nos dias de hoje, com o avanço tecnológico, é possível fazer inúmeras atividades com o celular e isso alavancou ainda mais as possibilidades de se fazer automação residencial, através de alguns protocolos de comunicação que foram desenvolvidos justamente para trabalhar com projetos de Internet das coisas (*IoT – Internet of Things*), já que é necessário lidar com alguns fatores como baixa largura de banda, alta latência e instabilidade de comunicação (TORRES et al, 2016).

O presente trabalho tem o objetivo de desenvolver um sistema de automação residencial para controle da intensidade da luz de uma lâmpada de acordo com a quantidade de luz ambiente e gerenciado por *Smartphone* a fim de deixar a casa mais eficiente energeticamente. O desenvolvimento iniciou na construção do *hardware* e o *firmware* para fazer a leitura dos dados de luminosidade do ambiente e controlar a intensidade da luz automaticamente, em um segundo momento foi criado o aplicativo, que se comunicará com o sistema e com isso poder controlar manualmente a luminosidade do local, através do celular, é possível visualizar em tempo real a quantidade de energia que está sendo utilizada

e com isso avaliar se realmente a solução irá fazer diferença significativa no consumo energético. Podemos observar na Figura 1, a metodologia utilizada para o desenvolvimento do projeto.

Figura 1 – Diagrama da metodologia do desenvolvimento.



Fonte: Próprio autor.

Para o desenvolvimento deste trabalho, o mesmo foi dividido em 5 seções, sendo eles:

1- Fundamentação Teórica: aqui serão explicadas todas as tecnologias bases para entendimento do trabalho;

2- Revisão Sistemática da Literatura: nesta seção foram descritas as principais informações sobre o estado da arte da plataforma utilizada, sobre os componentes, protocolo e sobre o ambiente de desenvolvimento do aplicativo;

3- Metodologia e Desenvolvimento: aqui estão explicitadas todas as informações de metodologias que foram usadas para o desenvolvimento do projeto descrito;

4- Resultados e Discussões: nesta seção estão as discussões com base nos resultados obtidos;

5- Conclusão e Trabalhos Futuros: aqui podemos observar as principais dificuldades encontradas e também todo o conhecimento adquirido no decorrer do desenvolvimento do projeto, além de sugerir aplicações futuras baseadas neste trabalho.

2 Fundamentação Teórica

2.1 Eficiência Energética

Já há algumas décadas existe uma busca de um modelo de desenvolvimento que possa ser considerado sustentável para a humanidade. Essa busca foi muito impulsionada por vários problemas de caráter ambiental e social, como o aquecimento global, desastres ecológicos mais frequentes, existência de grande parte da população vivendo na pobreza e o caráter extremamente consumista da sociedade atual, tem trazido uma conscientização dessas interferências significativas dos sistemas humanos sobre os sistemas naturais, levando a um desequilíbrio e possíveis impactos irreversíveis sobre os referidos sistemas (ROMÉRO; REIS, 2012).

Investir na eficiência energética colabora não só para tornar o ambiente mais sustentável, como também para se gastar menos no longo prazo. Quanto menos energia for utilizada, menor será a necessidade de produzi-la, levando assim uma grande vantagem para o planeta, já que grande parte da eletricidade gerada atualmente é proveniente da queima de combustíveis fósseis. Esses combustíveis emitem gases de efeito estufa como o dióxido de carbono, um dos principais causadores do aquecimento global. Existem outras formas de se obter energia elétrica, por meio de hidrelétricas, que ocorre no alagamento de grandes áreas, e de usinas nucleares que geram resíduo atômico (BORGES; DORES, 2010).

De acordo com Roméro e Reis (2012), no setor elétrico existem vários desenvolvimentos que visam diminuir o impacto atmosférico ambiental negativo causado por usinas baseados no uso de carvão mineral e derivados do petróleo. No setor industrial, existem mudanças tecnológicas que podem ter um impacto significativo na conservação de energia, desde o uso de motores e iluminação mais eficientes e da automação e implementação de novas soluções para o processamento e gerenciamento de processos. Já nos Setores residencial e comercial, incentivos financeiros têm sido ou podem ainda ser criados para influenciar consumidores individuais na demanda de produtos de maior eficiência energética, como aparelhos domésticos, sistemas de iluminação, aquecimento e refrigeração, entre outros.

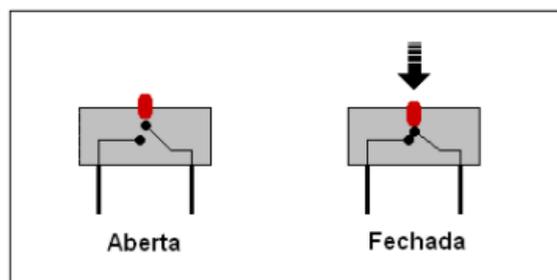
2.2 Sensores

Sensores são dispositivos sensíveis à alguma forma de energia presente no ambiente, essa energia pode ser térmica, luminosa, cinética, mensurando alguma grandeza física e relacionando as informações como temperatura, pressão, velocidade, corrente, aceleração, posição, entre outras (WENDLING, 2010).

2.2.1 Sensores Mecânicos

De acordo com Wendling (2010), sensores mecânicos são aqueles sensores que senseiam movimentos, posições ou presença utilizando recursos mecânicos como: Chaves Fim-de-Curso que são sensores que atuam sobre um circuito da forma aberta/fechada quando alguma ação mecânica acontece no seu elemento atuador. Na figura 2, podemos ver o funcionamento do sensor mecânico Chave de Fim-de-Curso.

Figura 2 – Exemplo de funcionamento de um sensor Chave Fim-de-Curso.

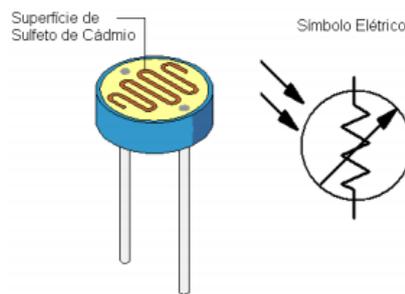


Fonte: WENDLING, 2010.

2.2.2 Sensores Fotoelétricos

Sensores fotoelétricos são sensores muito mais velozes que sensores mecânicos, já que eles atuam com luz, não existe inércia e nem peças móveis que quebram ou desgastam. Existem diversos tipos de sensores fotoelétricos e um deles é o fotoresistor (LDR) que possui uma superfície de Sulfeto de Cádmio (CdS), que é um composto químico onde sua resistência elétrica depende da quantidade de luz incidente (WENDLING, 2010). Na Figura 3, observamos a composição de um sensor fotoresistor LDR e seu funcionamento.

Figura 3 – Funcionamento do sensor fotoresistor LDR.

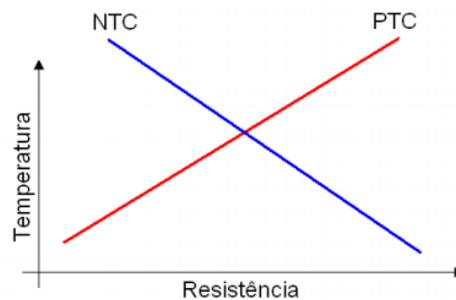


Fonte: WENDLING, 2010.

2.2.3 Sensores Térmicos

Sensores térmicos, assim como sensores fotoelétricos, possuem uma gama enorme de tipos de sensores que atuam em função da variação de temperatura. Um desses sensores o NTC (*Negative Temperature Coefficient*) que funciona da seguinte maneira: Quando a temperatura aumenta a resistência diminui. Outro sensor muito parecido é o PTC (*Positive Temperature Coefficient*), que quando a temperatura aumenta a resistência aumenta também (WENDLING, 2010). Podemos ver como se comporta a resistência dos sensores de temperatura NTC e PTC, observando a Figura 4.

Figura 4 – Resistência dos sensores NTC e PTC em função da temperatura.



Fonte: WENDLING, 2010.

2.2.4 Sensores de Corrente

Sensores de corrente são muito utilizados para monitorar a energia elétrica de um equipamento. Existem dois tipos de sensores de corrente, o invasivo, que precisa estar conectado ao equipamento e a corrente precisa fluir pelo sensor, e o sensor não invasivo, que não há necessidade de interromper o circuito para instalar o sensor, basta envolver um dos fios de energia. O funcionamento de um sensor não invasivo se deve ao fato da Lei de Ampère, quando uma corrente flui em um condutor, é gerado um campo magnético proporcional a corrente, como a corrente alternada varia no tempo, logo ela vai gerar um campo magnético que também varia no tempo e, de acordo com a Lei de Faraday, um campo magnético que varia no tempo, induz uma tensão, que por sua vez, uma corrente elétrica proporcional (THOMSEN, 2015).

Figura 5 – Sensor de corrente não invasivo SCT 013-000.



Fonte: THOMSEN, 2015.

2.3 Microcontroladores

Segundo Santos e Junior (2019), microcontroladores são equipamentos programáveis que possuem pequeno porte e baixo custo. Normalmente é composto com uma unidade processadora, memórias, entradas e saídas, controle temporal, conversores analógicos e digitais entre outros. Sua função é administrar as ações e eventos remotamente, controlando todo o sistema embarcado.

Microcontroladores são considerados um computador completo desenvolvido em um circuito integrado único (ARAUJO et al, 2019). Mas existem uma enorme quantidade de modelos de microcontroladores no mercado e cada um com sua própria característica, como por exemplo: ESP32, PIC32, NodeMCU, Atmel AVR, entre outros.

Arduino é uma plataforma de prototipagem eletrônica programável que permite processar entradas e saídas entre o dispositivo e os componentes externos conectados junto dele. A placa do Arduino é composta de um microcontrolador Atmel AVR, um cristal ou oscilador e um regulador linear de 5 volts. Alguns modelos de Arduino possuem também uma saída USB para conexão com o computador, permitindo o *upload* e recuperação dos

dados. (McRoberts, 2011). No presente trabalho, será utilizado a plataforma Arduino Uno para ser o principal controlador.

2.4 Protocolos de Comunicação

Para tornar a IoT viável e também permitir a conexão de centenas de milhares de coisas, é necessário que todas essas coisas sejam de baixo custo, implicando assim em baixo custo de processamento, armazenamento e comunicação. Os protocolos para a comunicação entre os componentes devem lidar com alguns fatores como baixa largura de banda, alta latência e instabilidade da comunicação (TORRES et al, 2016).

Os protocolos possibilitam a transmissão da informação entre as camadas e entre os equipamentos além de ser responsáveis por dividir os dados em pacotes que são transmitidos pela rede, dentro de cada pacote deve existir o endereçamento do destino, uma numeração para transformar esse pacote em único, estabelecer um canal fechado entre transmissor e receptor, controle de erros de comunicação e a correção dos mesmos, retransmissão ou confirmação e uma conversão de código com adequações de pacotes enviados ou recebidos (MUNIZ et al, 2018).

Com uma grande variedade de sistemas operacionais e também uma enorme gama de *hardwares* diferentes, precisava ser criada uma padronização dos protocolos de envio e recebimento de dados. Para comunicação entre 2 dispositivos a ISO (*International Organization for Standardization*) trouxe o modelo OSI (*Open System Interconnection*) que contava com sete camadas: Aplicação, Apresentação, Sessão, Transporte, Rede, Enlace e Física. Já o modelo TCP/IP, que foi criado pelo DoD (Departamento de Defesa dos Estados Unidos) tornou-se mais popular. Esse modelo possui apenas quatro camadas: Aplicação, Transporte, Internet e *Host/rede*. (MUNIZ et al, 2018).

O padrão IEEE 802.11, que é conhecido por *WiFi* que é um acrônimo para *Wireless Fidelity* que significa um conjunto de especificações para redes locais sem fio (*Wireless Local Area Network* (WLAN)), por ser muito popular, o IEEE 802.11 possui vários padrões próprios de rede sem fio, como 802.11(a,b,g,n,ac), assim ele vai se reestruturando para não ser substituído por outra tecnologia que atenda melhor a evolução de IoT. Esses padrões trabalham em uma faixa definida de 2,4GHz a 5GHz, e essas redes possuem um número finito de elementos conectados. O ponto de acesso que é chamado de roteador *wireless*, é um dispositivo que funciona interligando a rede *ethernet* na rede *WiFi*, com isso, promovendo para o cliente acesso à internet ou rede local (MUNIZ et al, 2018).

2.5 Desenvolvimento Mobile

Desde o lançamento do primeiro iPhone em 2007, pela Apple, a tendência *mobile* tem crescido exponencialmente no mundo todo e no Brasil não foi diferente. Com isso surgiu uma corrida para desenvolvimento de aplicativos, que basicamente é comandado por duas plataformas: Android e iOS (CRUZ; PRETUCELLI, 2018).

Para produzir aplicações realmente nativas para cada plataforma, surgiram ferramentas que permitem os desenvolvedores utilizar conhecimentos prévios em desenvolvimento *web* e contornar algumas limitações, uma dessas ferramentas é o *React Native* que foi desenvolvida pelo *Facebook*, utiliza em seu núcleo o *React* que é uma biblioteca *JavaScript* e se baseia na criação de interfaces através de JSX, uma extensão para o *JavaScript* que possibilita escrever sintaxe XML em meio aos códigos da linguagem (CRUZ; PRETUCELLI, 2018).

3 Estado da Arte

Nos últimos anos a automação residencial é um tema muito discutido em trabalhos científicos. As tecnologias utilizadas variam muito, desde projetos em que o usuário é apenas um elemento não participativo, até aplicações onde a participação do usuário é fundamental para fazer o controle do sistema. Porém em todos os trabalhos é realizado a busca da otimização, a otimização tem várias formas, podendo ser a otimização do tempo do usuário ou a otimização de recursos. Um desses recursos é a energia elétrica, com isso encontramos alguns trabalhos que fizeram um enfoque em energia elétrica, utilizando a automação residencial.

Em Souza et al (2019) foi feito comparações entre diferentes microcontroladores e lâmpadas a fim de verificar quais opções são mais relevantes para o consumidor, sendo importante ressaltar que apenas ao fato de trocar as lâmpadas fluorescentes da residência para lâmpadas LED tem um impacto significativo na quantidade de energia elétrica utilizada. Os autores concluem que fazendo a troca para esses equipamentos e que os mesmos sejam em quantidade relevante para uma casa popular, proporcionam ganho na qualidade de energia, redução de distorções harmônica e redução no consumo de energia.

Penteado et al (2019) faz a utilização de um Arduino que faz a leitura de um sensor de umidade do solo para controlar um irrigador automático que quando necessário abre sua válvula para irrigar a horta, também foi implementado um alimentador de animais que vai colocar a comida na hora e na quantidade certa. Essa automação contou também com sensores de corrente em cada uma das aplicações que foram instalados com o propósito de monitorar a energia elétrica gasta pelos equipamentos, essa monitoração ocorre em uma aplicação *web*, onde os dados são enviados via ESP32 que é um microcontrolador que possui módulo *WiFi*. Os autores concluem que o projeto se mostrou muito útil, com sensores e dispositivos funcionando de forma satisfatória tendo apenas alguns problemas na comunicação do ESP32 e aplicação *web*.

Araujo (2021) apresenta um projeto construído a partir de uma série de sensores, tais como: luminosidade, temperatura, umidade, raios UV e ultrassônico, controlados pelo microprocessador NodeMCU. A comunicação entre os periféricos foi construída a partir do protocolo MQTT, utilizando o *broker cloudMQTT*, que pode ser acessado via *web*, possibilitando a aplicação ser controlada remotamente. O objetivo do projeto é automatizar uma residência focando no baixo custo, ele mostra que seu projeto custou aproximadamente 269 reais e conclui que é possível automatizar uma residência sem precisar elevar o custo do projeto.

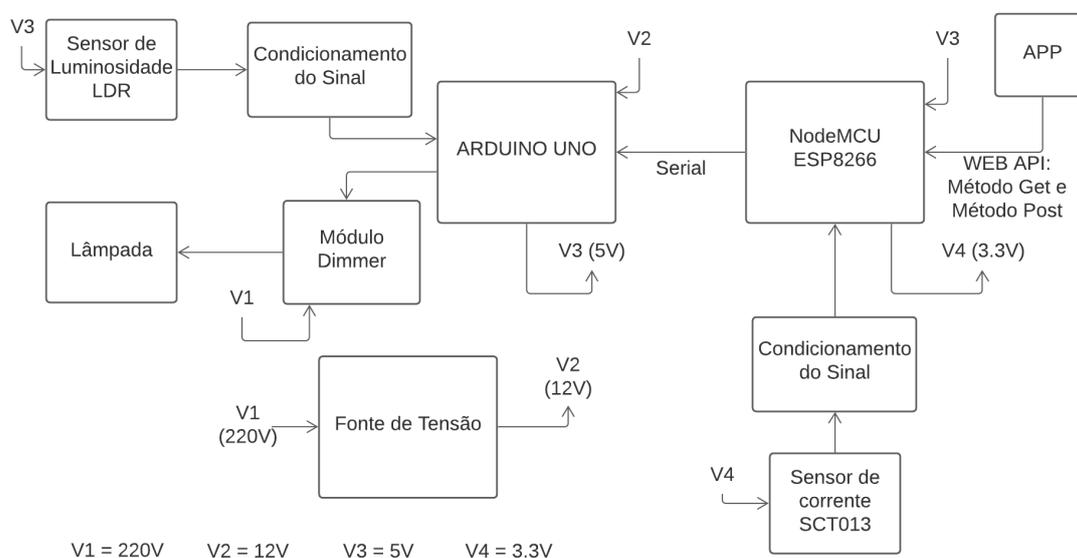
Com um objetivo semelhante, de construir um projeto de baixo custo, Almeida et al (2021) desenvolveu uma aplicação capaz de monitorar o consumo de energia elétrica em tempo real de forma segura e não invasiva, utilizando sensores de corrente e tensão para coletar os dados que são processados através de um Arduino. Com um custo de apenas 320 reais para o projeto, conclui-se que é possível monitorar a energia com um baixo custo de aplicação e reforça que é possível utilizar a tecnologia para diminuir o consumo de energia elétrica.

Observando os estudos citados anteriormente, é possível concluir que a utilização de sensores, juntamente de uma boa estratégia de controle pode gerenciar e otimizar o consumo de energia, além de podermos dar conforto para o usuário que irá desfrutar do sistema.

4 Desenvolvimento

A partir dos resultados que foram obtidos no Estado da arte, juntamente com pesquisas de custos e de disponibilidade imediata, foram escolhidos os componentes a serem utilizados no presente trabalho. Após a definição dos componentes foi feita uma análise do projeto para definição das etapas a serem desenvolvidas. Primeiramente foi construído um diagrama de blocos para a para definição da arquitetura do projeto, podemos observar na Figura 5, neste diagrama está apresentado todos os componentes escolhidos juntamente com a metodologia de transmissão de informações entre os sistemas.

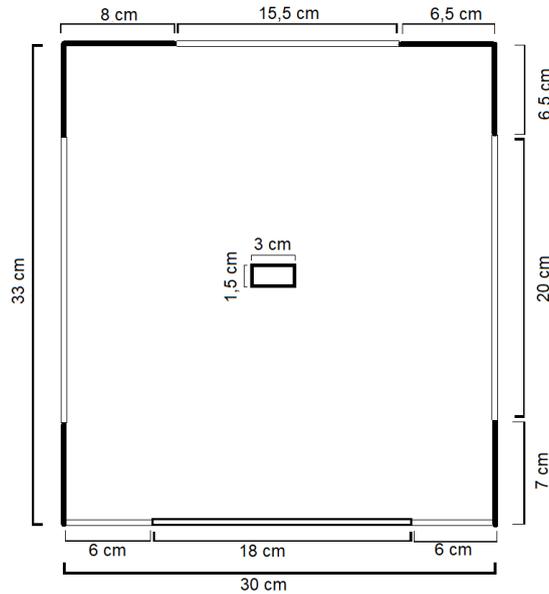
Figura 6 – Diagrama de blocos do projeto.



Fonte: Próprio autor.

Para representar uma sala comercial, foi construído uma pequena maquete utilizando uma caixa de papelão, onde foram feitas aberturas na caixa para representar portas e janelas, possibilitando a entrada da luz. Também foi construído uma entrada no teto para prender a lâmpada. Na Figura 6, podemos observar as medidas da maquete, foi utilizado uma representação em escala de 1:20, possibilitando assim testes e obtenção de dados.

Figura 7 – Plano horizontal da sala comercial.



Fonte: Próprio autor.

4.1 Hardware

Para ser a principal forma de controle do nosso sistema, o Arduino Uno foi o escolhido, ele é o responsável por fazer a leitura e tratar dos dados do sensor fotoresistor LDR, também é responsável pelo acionamento da lâmpada através do envio de sinais para o módulo *dimmer*. O Arduino Uno também faz conexão serial com o microcontrolador NodeMCU, ele possui o módulo *WiFi* ESP8266 integrado. O ESP8266 é um chip *WiFi* de baixo custo desenvolvido pela *Espressif Systems* e utiliza o protocolo TCP/IP (PARIHAR, 2019). A utilização de dois microprocessadores se deve ao fato de que ambos têm suas limitações próprias, mas em conjunto se completam. Para fazer a programação do Arduino e do NodeMCU foi utilizado a IDE (*Integrated Development Environment*) do Arduino.

O Arduino IDE é um *software open-source* que foi construído para facilitar a criação de códigos e o *upload* para a placa. O *software* pode ser utilizado com qualquer placa Arduino (ARDUINO, 2021). Para trabalhar com o NodeMCU, foi necessário instalar um módulo genérico criado pela comunidade de maneira não oficial através da própria interface do Arduino IDE, possibilitando assim a programação do NodeMCU.

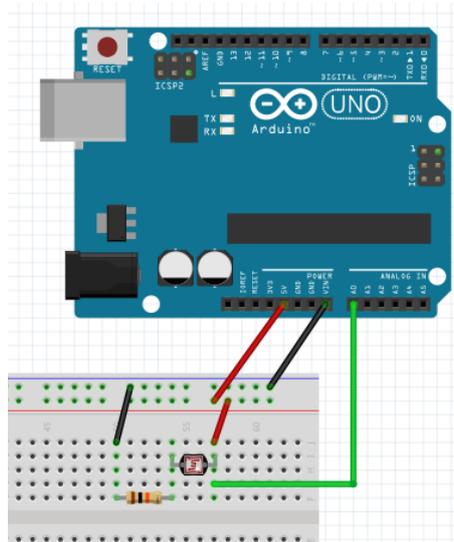
O projeto foi dividido em algumas áreas de atuação, que vão ser explicados em específico na continuidade do trabalho. A divisão foi feita no seguinte formato: sensor fotoresistor LDR, Módulo *dimmer*, Sensor de Corrente, Servidor *web* e Conexão Arduino x NodeMCU. Através do *software* Fritzing, foi construído uma simulação da disposição real do projeto.

4.1.1 Sensor Fotoresistor LDR

O sensor fotoresistor LDR consegue alterar sua resistência de acordo com a luz presente no ambiente, sendo assim podemos medir a tensão de saída desse resistor utilizando

a porta analógica do Arduino, como observado na Figura 7, o sensor LDR é conectado diretamente em 5V e na saída temos um resistor de $10k\Omega$ entre o sensor LDR e o terra. Para medirmos a tensão de saída, devemos colocar na porta analógica do Arduino um *jumper* até o lado conectado em 5V.

Figura 8 – Disposição do Arduino com e o sensor LDR.



Fonte: Próprio autor.

Na programação fazemos a leitura dos dados através da função *analogRead()*, essa função nos retorna valores de 0 até 1023, quanto maior o valor maior a quantidade de luminosidade presente no ambiente. Para a utilização desse projeto, foi feita a inversão do sinal, transformando o valor de leitura, sendo o maior valor para a menor quantidade de luminosidade no ambiente. Utilizando a função *map()*, transformamos o valor da luminosidade que pode ir de 0 até 1023, para um valor de 10 a 100, essa transformação é feita pois o módulo *dimmer* trabalha com valores de 0 a 100.

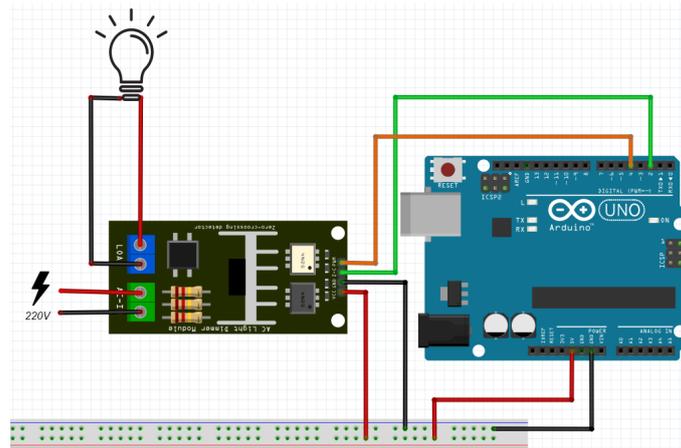
A função *map()* faz um remapeamento de uma variável que pertence a um intervalo de valores para um novo intervalo de valores, ela apenas utiliza números inteiros e não gera números fracionados. A sintaxe dessa função é *map(valor, deMenor, deMaior, paraMenor, paraMaior)* possuindo os seguintes parâmetros: *valor* é o número a ser mapeado, *deMenor* é o menor limite do intervalo atual do valor, *deMaior* é o maior limite do intervalo atual do valor, *paraMenor* é o menor limite do intervalo alvo, *paraMaior* é o maior limite do intervalo alvo. Essa função irá retornar apenas o valor mapeado para o novo intervalo (ARDUINO, 2020).

4.1.2 Módulo *Dimmer*

O módulo *dimmer* é o responsável por receber o sinal do Arduino e controlar a intensidade da lâmpada. Esse módulo contém um transistor Triac BT137 que é utilizado para chavear a corrente alternada, esse chaveamento permite que a corrente seja interrompida através de uma regulagem de disparos, que é controlada pelo Arduino. O módulo possui dois blocos com quatro terminais em cada, sendo um bloco voltado para conexões AC,

que são um par de fios recebida da tomada e um par de fios enviados para a lâmpada e o outro bloco é para conexões DC, vindas do Arduino, que são identificadas como: GND que é conectado ao terra do Arduino, DIM que é o gatilho que define se pode ou não passar corrente entre as conexões AC, ZC é quem detecta o momento que a tensão recebida da tomada AC passada pelos 0V e VCC que é a entrada de tensão conectada aos 5V do arduino (CAMPOS, 2016). Podemos ver como é a montagem do Módulo *dimmer* junto com o Arduino na Figura 8.

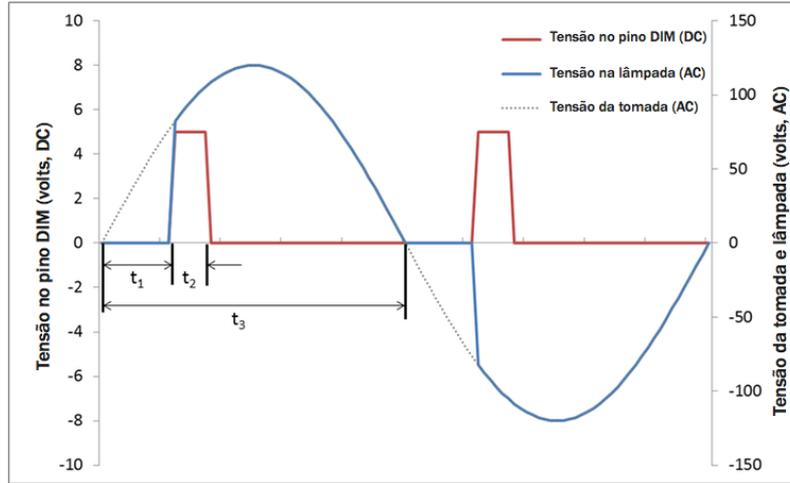
Figura 9 – Disposição do Arduino com o módulo *dimmer* e a lâmpada LED dimerizável.



Fonte: Próprio autor.

Sempre que a tensão alternada chega a 0v ela é sinalizada pelo módulo para o terminal ZC, para uma rede elétrica de 60Hz isso deve ocorrer aproximadamente a cada 8,33 milissegundos. A Figura 9 mostra o funcionamento durante um ciclo AC que se repete aproximadamente 60 vezes por segundo. A duração de t_1 é o elemento que define a luminosidade da lâmpada, quanto mais longa, menor será a luminosidade produzida, o limite máximo que t_1 pode crescer é aproximadamente o valor de t_3 , ou seja meio ciclo AC, caso for igual ou maior a meio ciclo AC a lâmpada não irá acender. A duração de t_2 não importa muito já que o terminal DIM funciona como um gatilho e não como um temporizador, quando disparado, o TRIAC é acionado e ficará assim até passar por 0V novamente, reiniciando o ciclo (CAMPOS, 2016).

Figura 10 – Funcionamento do módulo *dimmer* em um ciclo AC.



Fonte: CAMPOS, 2016.

Na prática utilizamos a interrupção externa INT0 que fica no pino 2 do Arduino, ela vai interromper a função *loop()* ou qualquer outra e executará a função *zeroCross()*. Dentro da função *zeroCross()* encontramos o valor do tempo t_1 , utilizando a Fórmula (1), onde $(t_3 - t_2) * 100 \approx 8200$, $MaxLuminosidade = 100$ e $Luminosidade = ValorSensorLDR$ chegando assim na Fórmula (2), que é a fórmula final de t_1 . Com o valor de t_1 , é esperado um *delay* em microsegundos com esse valor, para que aí, o terminal gatilho DIM, seja acionado e após 6 microsegundos ser desativado.

$$t_1 = ((t_3 - t_2) * 100) * (MaxLuminosidade - Luminosidade) / 100 \quad (1)$$

$$t_1 = (8200) * (100 - Luminosidade) / 100 \quad (2)$$

4.1.3 Sensor de Corrente

Para fazer o gerenciamento da quantidade de energia elétrica que está sendo utilizada pela lâmpada, foi utilizado um sensor SCT 013-000, que é um sensor de corrente não invasivo. O sensor SCT 013-000 possui uma especificação de 100A na bobina do primário para 50mA na bobina do secundário, isso significa que ele possui um fator K de 2000, que é referente a quantidade aproximada de espiras no seu trafo (ALMEIDA et al, 2021). O valor RMS ou valor eficaz é de 100A, utilizando a Fórmula (3), é descoberto o valor da corrente de pico na bobina primária e com isso, é possível descobrir a corrente de pico na bobina secundária, com a Fórmula (4).

$$CorrenteDePicoPrimária = CorrenteRMS * \sqrt{2} = 100 * \sqrt{2} = 141.4A \quad (3)$$

$$CorrenteDePicoSecundária = CorrenteDePicoPrimária / NumeroDeEspiras = 141.4 / 2000 = 0.0707A \quad (4)$$

Como estamos medindo corrente alternada, o valor que pode ser lido pelo microcontrolador é de metade da tensão de referência, para realizar a leitura do sensor de corrente

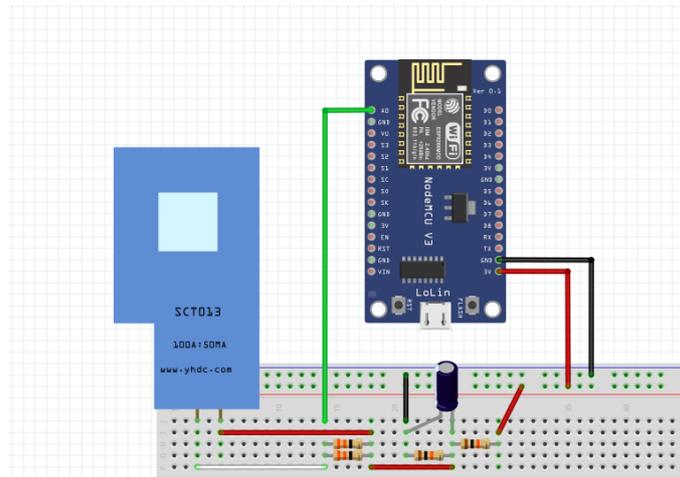
estamos utilizando o NodeMCU, ele possui tensão de referência de 3.3V, então podemos afirmar que teremos uma tensão no sensor de aproximadamente 1.65V, utilizando a lei de ohm, se faz necessário um resistor de carga, que vai ter um valor de 23Ω de acordo com a Fórmula (5), como o menor resistor que eu tinha a disposição era de 33Ω , eu utilizei dois resistores de 33Ω em paralelo no meu circuito, fazendo associação de resistores obtemos 16.5Ω , de acordo com a Fórmula (6).

$$R_{carga} = 1.65/0.0707 = 23\Omega \quad (5)$$

$$R_{eq} = (R1 * R2)/(R1 + R2) = (33 * 33)/(33 + 33) = 16.5\Omega \quad (6)$$

Como o NodeMCU faz a leitura de uma corrente alternada, a faixa de leitura vai ser de - 1.65V até 1.65V, porém, é necessário fazer a transformação para uma faixa em que o NodeMCU consiga trabalhar, ou seja, de 0 a 3.3V, para fazer essa transformação, foi construído um divisor de tensão, utilizando 2 resistores de $10k\Omega$ e um capacitor de *Offset* de $100\mu F$. Podemos observar na Figura 10, como fica a montagem desse sensor com o NodeMCU.

Figura 11 – Disposição do NodeMCU com o sensor de corrente.



Fonte: Próprio autor.

Já na programação do NodeMCU, precisamos utilizar a biblioteca *EmonLib.h*, que é uma biblioteca criada exclusivamente para atuar com monitoramento de energia. Com essa biblioteca podemos usar a classe *EnergyMonitor*, dentro dessa classe foi usado duas funções, primeiramente a função *current(inPinI, ICal)* dentro do *setup()*, onde *inPinI* é o terminal analógico que foi realizada a leitura do sensor e *ICal* é o valor de calibração, que é calculado de acordo com a Formula (7), dentro do *loop()*, utilizamos a função *calcIrms(SAMPLES)* para calcular o valor da corrente, onde *SAMPLES* é o quantidade de amostras que o irão ser lidas, para esse valor foi utilizado 1480 amostras, pois é o valor sugerido na biblioteca (ARDUINO, 2021).

$$ICal = \text{NúmeroDeEspiras}/R_{carga} = 2000/16.5 = 121.21 \quad (7)$$

4.1.4 Servidor Web

Como já falado anteriormente, o grande diferencial da placa NodeMCU é possuir o módulo *WiFi* ESP8266 nativo, podendo se conectar facilmente à redes sem fio com padrão 802.11 b/g/n. Para conseguirmos fazer a conexão é necessário a utilização da biblioteca *ESP8266wifi.h*. Dentro do código, para definirmos um IP fixo para a conexão, criamos 3 variáveis com a classe *IPAddress*, uma variável contendo o IP que desejamos nos conectar, que no meu caso foi 192.168.1.100, outra variável contendo o *gateway* padrão, ao qual para o meu roteador é 192.168.1.1 e a última variável contendo a máscara de rede, ou *subnet*, que foi definida 255.255.255.0, com essas 3 variáveis em mãos utilizamos a função *wifi.config(ip, gateway, subnet)*. Após configurar a conexão, nos conectamos a rede utilizando a função *wifi.begin(ssid, password)*, onde *ssid* é nome e *password* é a senha da rede em que iremos nos conectar. Na Figura 11, observamos parte do código utilizado para se conectar e criar um servidor *web* pelo NodeMCU.

Figura 12 – Código utilizado para conexão e criação de servidor web no NodeMCU.

```
IPAddress ip(192, 168, 1, 100);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

// Variáveis de Server e Status do LED
ESP8266WebServer server(80);

void setup() {
  // Inicia Conexão WiFi
  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password); //senha e password defidos por variável
  Serial.println("");

  // Aguarda Conexão
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Configura Handles do Server e Inicia Server
  server.on("/", handle_OnConnect);
  server.on("/ledon", handle_ledon);
  server.on("/ledoff", handle_ledoff);
  server.on("/auto", handle_auto);
  server.on("/manual", handle_manual);
  server.on("/countplus", handle_countplus);
  server.on("/countsub", handle_countsub);
  server.on("/corrente", handle_corrente);
  server.onNotFound(handle_NotFound);
  server.begin();
  Serial.println("Servidor HTTP iniciado!");
}
|
```

Fonte: Próprio autor.

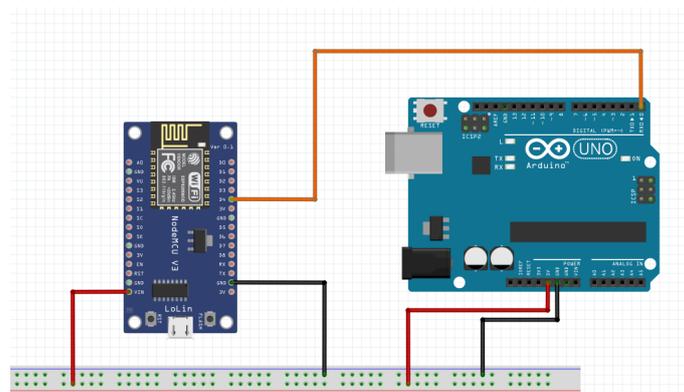
Após realizar a conexão com a rede, criamos o servidor *web* com o NodeMCU, para isso foi utilizado a biblioteca *ESP8266WebServer.h*. Com essa biblioteca, podemos definir uma variável chamada *server*, com a porta definida em 80, utilizando a função *server.on(caminho, função)* nós configuramos as rotas do servidor, onde no primeiro parâmetro passamos o caminho da rota e no segundo parâmetro passamos a função que irá ser executada quando essa rota for acessada. Para acessarmos a rota, é feito uma chamada, essa chamada pode ser feita via API pelo aplicativo ou por um navegador qualquer, no projeto foram utilizadas oito rotas. Após a configuração das rotas iniciamos o servidor utilizando a função *server.begin()*.

4.1.5 Conexão Arduino x NodeMCU

Para transmitir as informações do NodeMCU para o Arduino foi utilizado comunicação serial. A interface serial UART (*Universal Asynchronous Receiver Transmitter*) do Arduino é responsável por enviar informações e receber informações de outro microcontrolador. Para isso são utilizados os pinos nomeados de TX (*Transmitter*) e RX (*Receiver*). No Arduino Uno esses pinos são respectivamente, D1 e D0 e também são utilizados pelo USB para o *upload* dos programas (TANNUS, 2018).

Neste trabalho apenas recebemos dados no Arduino, com isso, utilizamos apenas a porta RX. Já no NodeMCU, como apenas publicamos as mensagens, utilizamos a porta TX. No NodeMCU temos três portas TX, no projeto foi utilizado a porta D4, que corresponde a porta TXD1, podemos ver na Figura 12 a disposição do Arduino juntamente com o NodeMCU.

Figura 13 – Disposição da comunicação serial Arduino x NodeMCU.



Fonte: Próprio autor.

Na programação, toda vez que o aplicativo envia uma requisição para o NodeMCU, ele envia uma mensagem UART para o Arduino, essa mensagem contém apenas um caractere. Caso seja feita uma requisição para acender a luz, o nodeMCU enviará a letra 'P', caso a requisição seja feita para desligar é enviado a letra 'D', para requisições de modo automático e manual, respectivamente 'A' e 'Z', quando estamos em modo manual, podemos escolher a intensidade da lâmpada, essa intensidade pode ir de 1 até 9, o NodeMCU envia para o Arduino o valor escolhido em forma de caractere, por exemplo '5'. Podemos ver na Figura 13, a requisição para acender a lâmpada. Já pelo lado do Arduino, foi criada uma interrupção externa, para que, quando uma mensagem serial chegar, ele interromper qualquer outra função e executar a leitura dessa mensagem. O Arduino vai tratar cada mensagem enviada, armazenando em variáveis correspondentes e utilizando a informação da maneira correta.

Figura 14 – Função que envia o comando de ligar por meio da comunicação UART.

```
void handle_ledon() {  
    Ligado = HIGH;  
    Serial1.println("P");  
    server.send(200);  
}
```

Fonte: Próprio autor.

4.2 Aplicativo

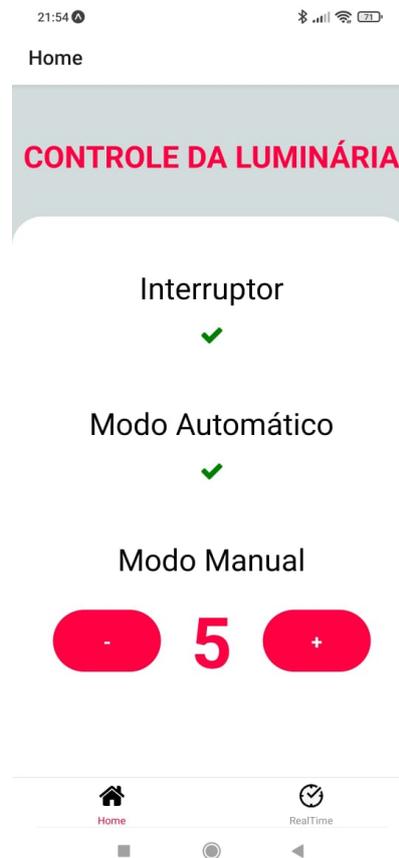
Para desenvolver o aplicativo foi necessário fazer diversas escolhas de *softwares* e *frameworks*. Para ser o ambiente de desenvolvimento foi escolhido o *Microsoft Visual Studio*, que é uma um editor leve, porém, muito poderoso para edição de códigos (MICROSOFT, 2021). Na criação do código, foi utilizado *React Native*, que é uma biblioteca *JavaScript*, juntamente com a plataforma *Expo*, que é um *framework* desenvolvido para aplicações em *React*, onde permite-se desenvolver, criar, testar e facilmente iterar com aplicativos *IOS*, *Android* e *web app* (EXPO, 2021).

Esta seção foi recortada em três subseções, sendo elas: *Home Page*, *RealTime Page* e Comunicação com NodeMCU. Para criar a navegação entre as páginas foram utilizadas as bibliotecas *@react-navigation/bottom-tabs* e *@react-navigation/native*, elas possibilitam que o aplicativo alterne entre páginas, utilizando uma barra de navegação na parte inferior da página. Dentro das duas primeiras subseções, na Figura 14 e na Figura 15, podemos observar como se comporta essa barra de navegação.

4.2.1 Home Page

Na Figura 14 podemos visualizar a *Home Page*, que foi criada com o intuito de ser o controlador geral do aplicativo, nela temos o "Interruptor" com uma *Checkbox*, que ao ser selecionada envia o comando para ligar a lâmpada e quando é retirado a seleção envia o comando para desligar a lâmpada. Da mesma maneira, temos o "Modo Automático", que também possui um *Checkbox*, que quando selecionado, envia o comando para o sistema utilizar o sensor de luminosidade e se controlar automaticamente, já quando não está selecionado, utiliza o "Modo Manual", que são dois botões logo abaixo, sendo um botão para aumentar a intensidade da lâmpada e o outro para diminuir a intensidade da lâmpada. Para monitoramento do nível manual da lâmpada existe um indicador com um número de 1 até 9 entre os botões. O *Checkbox* de Modo Automático e os botões de Modo Manual funcionam apenas quando o Interruptor está selecionado, o Modo Manual também só funciona caso o Modo Automático esteja desligado.

Figura 15 – Página *Home* do Aplicativo.



Fonte: Próprio autor.

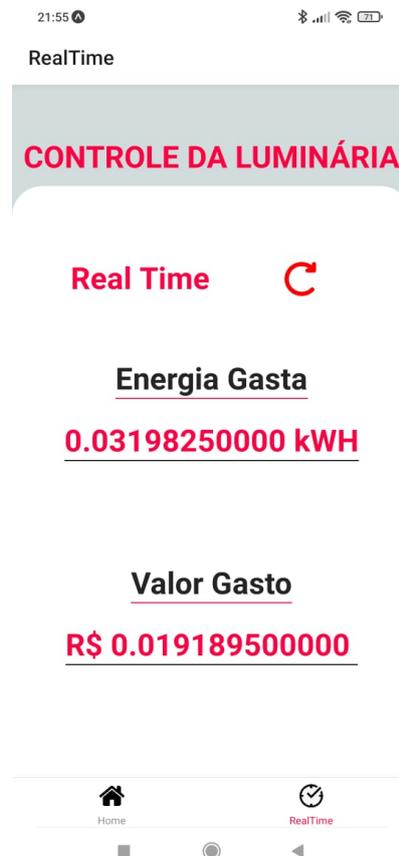
4.2.2 *RealTime Page*

Na Figura 15 podemos ver o *layout* da pagina *RealTime*, que tem como objetivo observar a quantidade de energia elétrica gasta em tempo real pela lâmpada. No canto superior direito temos um botão para recarregar a página, assim, recebendo o acumulado de corrente que foi medida, mais abaixo vemos a quantidade de energia elétrica gasta em kWh e também o valor gasto em reais. Para indicar a quantidade de energia elétrica utilizada foi feita a operação matemática observada na Fórmula (8), sendo os valores i = Corrente Acumulada, V = Tensão da Tomada = 220V e $t = 1h = 3600s$, o valor de 1000 que está multiplicando o tempo no divisor faz a transformação da resposta em WH para kWh. O valor de t está em segundos pois o NodeMCU faz a leitura e adiciona o valor na Corrente Acumulada a cada 1 segundo. Para visualizarmos o valor em reais que o sistema está gastando, foi utilizado uma taxa de 0.6, que significa que o preço de cada kWh é 0.6 reais, podemos ver o cálculo na Fórmula (9).

$$EnergiaGasta[kWh] = (i[A] * V[V]) / (t[s] * 1000) \quad (8)$$

$$ValorGasto[R\$] = EnergiaGasta[kWh] * Taxa = EnergiaGasta[kWh] * 0.6 \quad (9)$$

Figura 16 – Página *RealTime* do Aplicativo.



Fonte: Próprio autor.

4.2.3 Conexão com NodeMCU

Sempre que acontece um evento no aplicativo, é realizado uma chamada para uma função específica, cada função faz a chamada em um determinado caminho, que foi definido dentro do NodeMCU. Utilizando a biblioteca *Axios*, que é um projeto *open source* que possibilita fazer requisições HTTP (AXIOS, 2021), fazemos as requisições para o servidor utilizando o método *GET* para apenas dar um *ping* no servidor em determinada rota, apenas no envio do valor de intensidade do modo manual, é feito uma requisição no método *POST*, pois assim, além de pingar na rota do servidor enviamos a mensagem desejada para o NodeMCU. Na página de *RealTime* a única chamada que fazemos é para atualizar os valores, nesta requisição é utilizado o método *GET*, porém não apenas para pingar o servidor, mas também para pegar o valor acumulado da corrente que o servidor envia. Na Figura 16, vemos um exemplo de requisição.

Figura 17 – Função para aumentar a intensidade da lâmpada manualmente.

```
export default function Home(){
  const[ligar, setSelected] = useState(false)
  const[Automatico, Selected] = useState(false)
  Connect = "http://192.168.1.100/"
  const [counter, setCounter]=useState(5)

  function plus(){
    if ((counter != 9) && (!Automatico) && (ligar)){
      return (
        setCounter(counter+1),
        axios.post(Connect+"countplus", {
          Count: {counter}
        })
        .then(function (response) {
          console.log(response);
        })
        .catch(function (error) {
          console.log(error);
        })
      )
    }
  }
}
```

Fonte: Próprio autor.

5 Resultados e Discussões

Para a realização dos testes foi escolhido 4 configurações de intensidade e luz. Na Tabela 1 vemos cada um dos modos, que ocorreram entre os dias 01/09/2021 e 04/09/2021, apenas em horário comercial, todos foram submetidos a 8 horas de trabalho, iniciando as 10:00h e indo até as 18:00h. Utilizando o sensor de corrente, foi realizado a coleta de informações de energia elétrica. Na Figura 17, vemos o modelo utilizado para simular a sala comercial.

Tabela 1 – Modos utilizados nos testes.

NOME	TIPO DE LÂMPADA	MODO DE USO	DATA
MODO 1	LÂMPADA LED 9W	AUTOMÁTICO	01/09/2021
MODO 2	LÂMPADA LED 9W	COMUM (MÁXIMO)	02/09/2021
MODO 3	LÂMPADA INCANDESCENTE 70W	AUTOMÁTICO	03/09/2021
MODO 4	LÂMPADA INCANDESCENTE 70W	COMUM (MÁXIMO)	04/09/2021

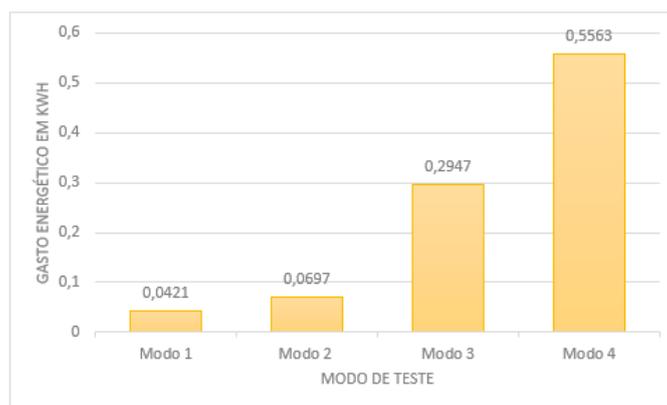
Figura 18 – Modelo de sala comercial utilizado para testes.



Fonte: Próprio autor.

Para efeito de comparação, foi utilizado lâmpadas LED e lâmpadas incandescentes, assim analisamos se o sistema se comporta bem e melhora a eficiência energética para ambos os tipos de lâmpada. Através dos testes realizados, foi construído o gráfico presente na Figura 19, observamos que o sistema se comportou como o esperado, economizando energia no modo automático, que se adapta conforme a luminosidade natural presente no ambiente, em relação ao modo comum, que mantém a lâmpada no máximo 100% do tempo.

Figura 19 – Resultado dos testes em kWh.



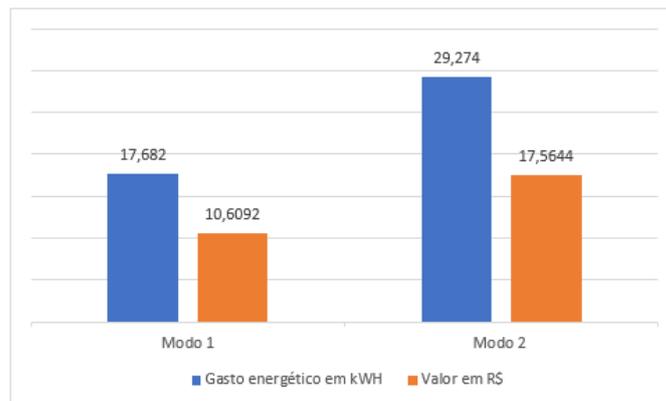
Fonte: Próprio autor.

Apesar de que com a lâmpada incandescente foi economizado 47% de energia no Modo 3 em comparação com o Modo 4 enquanto o Modo 1 economizou 40% em relação ao Modo 2, ainda assim se torna inviável trabalhar com lâmpadas incandescentes, já que as lâmpadas LED se mostram muito mais eficientes e econômicas. Por esse motivo as análises

a seguir serão feitas apenas utilizando lâmpadas LED.

Podemos pensar que uma lâmpada ligada por 8 horas e que possui apenas uma diferença de 0,0276 kWh do Modo 2 para o Modo 1, não faz diferença na prática, mas quando estamos falando de maiores intervalos de tempo e maiores quantidades de lâmpadas, essa diferença começa a se tornar mais expressiva. Como a maquete utilizada está sendo representada em escala de 1:20 e utiliza 1 lâmpada, em uma sala comercial de tamanho real, existiriam 20 lâmpadas, levando em consideração que essa sala comercial fica aberta 8 horas por dia e existem aproximadamente 21 dias úteis em um mês, podemos observar no gráfico da Figura 19, a quantidade de energia e a quantidade em reais que são gastos a mais pelo Modo 2 em relação ao Modo 1. Os valores em reais da energia elétrica variam bastante em cada estado do Brasil, levando em consideração uma média de todos os estados brasileiros, foram feitas as medidas utilizando a relação de que 1 kWh é equivalente a R\$ 0,60.

Figura 20 – Gasto mensal energético e de valor em reais de uma sala comercial com 20 lâmpadas.



Fonte: Próprio autor.

Apesar de existir uma economia de aproximadamente 11,6 kWh por mês e que quando convertemos para reais essa diferença acaba sendo de aproximadamente R\$ 7,00 ao mês, não podemos pensar apenas em quanto isso influenciará no bolso, temos que levar em consideração que existem milhares de salas comerciais no Brasil trabalhando a anos sem pensar em eficiência energética e que resulta em milhares de kWh que poderiam estar sendo economizados, com cada um fazendo a sua parte, podemos cada vez menos utilizar energias de fontes não renováveis.

Com a Tabela 2, é possível observar o custo total do projeto, que apesar de ter sido idealizado como um projeto de baixo custo, teve um orçamento total de aproximadamente R\$ 394,93, onde grande parte do custo deve-se ao fato da utilização de dois microcontroladores e de um sensor de corrente não invasivo. Componentes como resistores, capacitores, *protoboards*, *jumpers* entre outros itens de menor valor, foram estimados, somados e adicionados no campo "Itens diversos".

Tabela 2 – Custos do Projeto.

Produto	Custo
Arduino	R\$ 89,90
NodeMCU	R\$ 39,90
Módulo Dimmer	R\$ 47,90
Sensor de Corrente	R\$ 72,90
Sensor LDR	R\$ 0,90
Lâmpada LED Dimerizável	R\$ 56,44
Lâmpada Incandescente	R\$ 9,99
Itens diversos	R\$ 77,00
Total	R\$ 394,93

6 Conclusão e Trabalhos Futuros

Neste trabalho foi apresentado o desenvolvimento de um protótipo de sistema para controle e automação de uma lâmpada com o objetivo de economizar energia elétrica, para gerenciar e mensurar a energia que estava sendo consumida, foi utilizado um sensor de corrente que envia as informações para um *smartphone*, podendo acompanhar o gasto energético em tempo real. Para realização dos testes foi utilizado um modelo em escala e o sistema se mostrou eficaz, pois economizou cerca de 40% de energia em lâmpadas LED e 47% de energia em lâmpadas incandescentes.

Apesar de que os resultados sejam positivos, para uma maior confiabilidade em um ambiente real, se faz necessário a utilização de mais sensores e também de mais lâmpadas, assim podendo constatar a economia com um maior grau de certeza. Também é importante ressaltar que lâmpadas LED já são muito econômicas e que o investimento para um sistema desse em uma sala pequena que irá utilizar apenas uma lâmpada pode não compensar financeiramente, mas quando pensamos a longo prazo e em uma grande quantidade de lâmpadas, o sistema se mostra muito eficaz.

Para aumentar a confiabilidade, uma implementação interessante seria a criação de um banco de dados, obtendo informações de um maior intervalo de tempo, podendo analisar dados de vários dias, pois dependendo do clima a luminosidade do ambiente pode ser diferente, com o banco de dados criado, poderia ter uma seção no aplicativo para monitorar dados retroativos, assim podendo realmente ver o quanto de energia foi economizado em um intervalo de tempo.

Para diminuir o custo do projeto, uma opção interessante seria a substituição do Arduino e do NodeMCU pelo microcontrolador ESP32, juntamente com a utilização de uma fonte de tensão de 5V para alimentar o módulo *dimmer*. Outra opção seria substituir o sensor de corrente utilizado no projeto por um sensor de corrente invasivo, pois é um sensor de menor custo.

Os trabalhos futuros podem utilizar desta seção e utilizar todas essas abordagens que foram comentadas, além da possibilidade de aumentar a quantidade de eletrônicos, podendo assim controlar mais aparelhos, como ar-condicionado, televisão entre outros. O código

desenvolvido no trabalho está disponível em <https://github.com/PaulinhoDuarte/Controle-de-Lampada-por-Smartphone> para visualização e *download* além de poder ser refatorado para possíveis trabalhos futuros.

7 Referências

ALMEIDA, R. et al. SISTEMA DE MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA UTILIZANDO A TECNOLOGIA ARDUINO - Brazilian Journal of Development, 2021. Disponível em: <<https://www.brazilianjournals.com/index.php/BRJD/article/view/33251>>. Acesso em 26 ago. 2021.

ARAÚJO, Jonathas Formiga de. Automação residencial com objetos inteligentes: qualidade de vida e controle de energia. 2020. Monografia (Graduação em Engenharia da Computação) - Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2020. Disponível em: <<https://repositorio.uniceub.br/jspui/handle/prefix/15107>>. Acesso em 26 ago. 2021.

ARAUJO, W. M., CAVALCANTE, M. M., SILVA, R. O. VISÃO GERAL SOBRE MICROCONTROLADORES E PROTOTIPAGEM COM ARDUINO. 2019. Disponível em: <<http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/view/1357/0>>

ARDUINO. Arduino IDE 1.8.15. 2021. Disponível em: <<https://www.arduino.cc/en/software>>. Acesso em: 29 ago. 2021.

ARDUINO. EmonLib Library. 2021. Disponível em: <<https://www.arduino.cc/reference/en/libraries/emonlib/>>. Acesso em: 29 ago. 2021.

ARDUINO. map() function. 2020. Acesso em: 29 ago. 2021. Disponível em: <<https://www.arduino.cc/reference/pt/language/functions/math/map>>.

AXIOS. Biblioteca AXIOS, 2021. Disponível em: <<https://github.com/axios/axios>> Acesso em: 01 set. 2021.

BORGES, L. P., DORES, R. C. AUTOMAÇÃO PREDIAL SEM FIO UTILIZANDO BACNET/ZIGBEE COM FOCO EM ECONOMIA DE ENERGIA. 2010. Disponível em: <<http://www.ene.unb.br/adolfo/Monographs/Graduation/TG10%20Luc%C3%ADlia%20P.%20Borges%20e%20Rodrigo%20C.%20Dores.pdf>>

CALÇA, Karin. Bem Morar, interação entre o Design de Interiores e a Automação para o bem estar em uma residência. Julho, 2014. Disponível em: <http://www.aureside.org.br/_pdf/artigo17.pdf>

CAMPOS, A. Arduino e dimmer com Triac: criando um amanhecer artificial, 2016. Disponível em: <<https://br-arduino.org/2016/03/arduino-triac-dimmer.html>>. Acesso em: 29 ago. 2021.

Cruz, V.; Pretucelli, E. TECNOLOGIAS WEB PARA O DESENVOLVIMENTO MOBILE NATIVO. SIMTEC - Simpósio de Tecnologia da Fatec Taquaritinga, v. 4, n. 1, p. 15, 14 maio 2018.

EXPO. 2021. Disponível em: <<https://expo.dev/>> Acesso em: 01 set. 2021.

LEMONS, Alan. Automação Residencial. 2016. Disponível em: <<https://repositorio.unesp.br/bitstream/handle/11449/156883/000905380.pdf>>

McRoberts, Michael. Arduino básico; [tradução Rafael Zanolli]. São Paulo : Novatec Editora, 2011.

MICROSOFT. VISUAL STUDIO CODE, 2021. Disponível em: <<https://code.visualstudio.com/docs>> Acesso em: 01 set. 2021.

PÁDUA, Ivo. Caracterização de Edifícios Inteligentes: um caso exemplo. 2006. Disponível em: <http://www.biblioteca.pucminas.br/teses/EngEletrica_PaduaIH_1.pdf>

Parihar, Yogendra Singh. (2019). Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products. 6. 1085. Disponível em: <<https://www.researchgate.net/publication/337656615>>. Acesso em: 29 ago. 2021.

PENTEADO, A. et al. SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA CONTROLE DE IRRIGAÇÃO, ALIMENTAÇÃO ANIMAL E MONITORAMENTO DE ENERGIA - Revista TechnoEng, 2019. Disponível em: <http://www.cescage.com.br/revistas/index.php/RTE/article/view/842>. Acesso em 25 ago. 2021.

Romero, M. A., & Reis, L. B. (2012). Eficiência energética em edifícios. Barueri: Manole.

SANTOS, J. W., JUNIOR, R. C. L. SISTEMA DE AUTOMATIZAÇÃO RESIDENCIAL DE BAIXO CUSTO CONTROLADO PELO MICROCONTROLADOR ESP32 E MONITORADO VIA SMARTPHONE. 2019. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/12133/1/PG_COAUT_2019_1_02.pdf>

SOUSA, A. et al. AUTOMAÇÃO RESIDENCIAL E EFICIÊNCIA ENERGÉTICA: UM ESTUDO DE CASO - Brazilian Journal of Development, 2019. Disponível em: <<https://www.brazilianjournals.com/index.php/BRJD/article/view/2899>>. Acesso em 24 ago. 2021.

TANNUS, A. Arduino: Comunicação Serial, 2018. Disponível em: <<http://45.4.96.19/bitstream/ae/1855/1/ARDUINO%20-%20COMUNICA%C3%87%C3%83O%20SERIAL.pdf>> Acesso em: 01 set. 2021.

THOMSEN, ADILSON. 2015. Medidor de corrente não invasivo com Arduino. Disponível em: <<https://www.filipeflop.com/blog/medidor-de-corrente-sct013-com-arduino/>>

TORRES, A., ROCHA, A., SOUZA, J. Análise de Desempenho de Brokers MQTT em Sistema de Baixo Custo. 2016. Disponível em: <<https://sol.sbc.org.br/index.php/wperformance/article/view/9727/9623>>

VINICIUS ANGELO DE O. MUNIZ., L. L. B. M. M. R. F. O. A. PROTOCOLOS DE COMUNICAÇÃO PARA INTERNET OF THINGS (IOT). Intercursos Revista Científica, [S. l.], 2019. Disponível em: <<https://revista.uemg.br/index.php/intercursosrevistacientifica/article/view/3712>>. Acesso em: 13 jun. 2021.

WENDLING, Marcelo. Sensores. 2010. Disponível em: <<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4-sensores-v2.0.pdf>>

WILSON, C., Hargreaves, T., Hauxwell-Baldwin, R. Benefits and risks of smart home Technologies. abril, 2017. <<https://www.sciencedirect.com/science/article/pii/S030142151630711X>>