

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
Etec TRAJANO CAMARGO
Eletroeletrônica

DIEGO SILVA MOTA
MARCOS JOSÉ FRUTUOSO
VICTOR PRADO RIBEIRO

ARMÁRIO AUTOMATIZADO DE MEDICAMENTOS

Limeira
2021

DIEGO SILVA MOTA
MARCOS JOSÉ FRUTUOSO
VICTOR PRADO RIBEIRO

ARMÁRIO AUTOMATIZADO DE MEDICAMENTOS

Projeto de Trabalho de Conclusão de Curso **apresentado ao Curso Técnico** em Eletroeletrônica da Etec Trajano Camargo orientado pelo Prof. Carlos Alberto Serpeloni Barros **como requisito parcial para obtenção do título de técnico em Eletroeletrônica.**

Limeira
2021

AGRADECIMENTOS

Diego Silva Mota

Primeiramente, agradeço a Deus pelo dom da vida, por me permitir sonhar alto e capacitar-me para almejar meus sonhos diariamente.

Á todo corpo docente do curso Técnico em Eletroeletrônica da Etec Trajano Camargo por todo aprendizado compartilhado no decorrer do curso, sem medir esforços para que o conhecimento chegasse a nós com a melhor qualidade em meio a pandemia da COVID19.

Ao Engenheiro Matheus Colis, amigo que o estágio me proporcionou, pela doação de um microcontrolador Arduino Leonardo, um servo motor, além de vários outros componentes para projetos com Arduino, que permitiu a realização do presente Trabalho de Conclusão de Curso.

Á empresa CPFL Energia que me deu grande suporte ao longo dos últimos oito meses, em especial, aos grandes colegas de trabalho Adailton, Davi, Jonathan, Leandro, Raul, Rodrigo e Vinícius.

Á minha família, sendo eles meus pais Juraci e Valdineire, irmão Matheus e minha namorada Amanda, por todo apoio e suporte para que eu enfrentasse o desavio de retornar ao curso técnico e concluir um sonho que havia sido “pausado” em 2015 para que eu pudesse me dedicar integralmente a graduação na área da saúde.

A quem não mencionei, mas esteve presente ao meu lado eu quero lembrar que não estão esquecidos: vocês foram imensamente importantes para concluir meu curso.

Marcos José Frutuoso

Agradeço a minha família, em especial Mãe que me motivou a continuar aprendendo e desenvolvendo novos conceitos junto a pessoas importantes desse curso.

Muito à Jesus e Nossa Senhora, pelas bênçãos em minha vida diante das dificuldades para que eu pudesse superar a covid e me permitirem hoje estar aqui realizando este projeto.

Ao Diego e o Victor, colegas de curso envolvidos nesse projeto de Trabalho de Conclusão de Curso, que durante difícil pandemia, unimos esforços para enfrentar os desafios impostos e juntos concluir esse novo projeto que vai facilitar a vida dos idosos.

A quem não mencionei, mas esteve presente ao meu lado eu quero lembrar que não estão esquecidos: vocês foram imensamente importantes para concluir meu curso.

Victor Prado Ribeiro

Hoje não posso esquecer o papel que Deus teve ao longo do meu percurso. Agradeço ao Senhor pela força que colocou no meu coração para lutar até alcançar esta grande meta na minha vida.

À Etec Trajano Camargo eu só posso demonstrar minha gratidão e reconhecimento porque sem todos os recursos que ela oferece não seria fácil.

A todos os professores eu agradeço a orientação repleta de conhecimento, sabedoria e paciência.

À minha família e a todos os meus amigos eu deixo uma palavra de gratidão por todo conforto e ânimo que me deram. Vocês são maravilhosos!

À empresa SEMEQ que me apoiou durante minha trajetória ao longo desse último ano, em especial Fernando Martins que me orientou e auxiliou.

A quem não mencionei, mas esteve presente ao meu lado eu quero lembrar que não estão esquecidos: vocês foram imensamente importantes para concluir meu curso.

“De todas as coisas seguras, a mais segura é a dúvida.”

BERTOLT BRECHT

RESUMO

A população idosa tem aumentado significativamente no Brasil. Concomitantemente, o uso de medicamentos para o controle de doenças crônicas não transmissíveis (DCNT) aumentaram, sendo a região Sudeste a que tem maior percentual fazendo o uso de fármacos para tratamento de DCNT. Em determinados momentos, faz-se necessário a presença de alguém ao lado da pessoa idosa para assessorá-la quanto ao momento correto de ingerir as medicações visto que a ingestão de medicamentos nos horários corretos são fundamentais para sua eficácia. Porém, em 2019, uma questão a nível mundial alterou por completo o comportamento da população. Trata-se da COVID19, que fez com que reduzisse inclusive esse contato presencial de filho, netos e amigos com os pais, tios, colegas, amigos de idade mais avançada. Com as mudanças de comportamento social, a cultura do Home Office cresceu, aumentando a procura pela Domótica, popularmente conhecida como automação residencial. Pensando em facilitar o acesso ao ensino-aprendizagem e a criação de projetos de automação, em 2005, na Itália, foi criado o Arduino®, microcontrolador programável através de linguagem C++. Devido ao aumento da utilização de agentes farmacológicos no Brasil, crescimento da população idosa, além da redução do contato presencial entre as pessoas em decorrência da COVID19, objetivo principal do presente trabalho foi, através da utilização de Arduino®, desenvolver um armário automatizado de medicamentos de baixo custo acionado por efeito audiovisual em datas e horários pré-estabelecidos, lembrando o usuário de ingerir os medicamentos e, em tempo real, permitir que um responsável acompanhe através de notificações no smartphone ou computador.

Palavras chaves: Domótica, Armário de Medicamentos, Automação.

ABSTRACT

The elderly population has increased significantly in Brazil. Concomitantly, the use of drugs for the control of chronic non-communicable diseases (NCDs) increased, with the Southeast region having the highest percentage using drugs for the treatment of CNCDs. At certain times, the presence of someone beside the elderly person is necessary to advise them on the correct time to ingest medications, since taking medications at the correct times is essential for its effectiveness. However, in 2019, a global issue completely changed the behavior of the population. This is COVID19, which made it even reduce this face-to-face contact between children, grandchildren and friends with parents, uncles, colleagues, older friends. With the changes in social behavior, the Home Office culture grew, increasing the demand for Home Automation, popularly known as home automation. Thinking about facilitating access to teaching-learning and the creation of automation projects, in 2005, in Italy, Arduino® was created, a programmable microcontroller using C++ language. Due to the increased use of pharmacological agents in Brazil, the growth of the elderly population, in addition to the reduction in face-to-face contact between people as a result of COVID19, the main objective of this study was, through the use of Arduino®, to develop an automated medicine cabinet low-cost triggered by audiovisual effect at pre-established dates and times, reminding the user to ingest the medication and, in real time, allowing a responsible person to follow it through notifications on the smartphone or computer.

Key words: Home Automation, Medicine Cabinet, Automation.

LISTA DE FIGURAS

Figura 1 –	Placa Arduino Leonardo.....	15
Figura 2 –	Placa Arduino Uno.....	16
Figura 3 –	Display LCD 16x2 <i>backlight</i>	16
Figura 4 –	Módulo I2C.....	17
Figura 5 –	Módulo RTC DS1307.....	18
Figura 6 –	<i>Buzzer</i>	18
Figura 7 –	Diodo emissor de luz de diferentes cores.....	19
Figura 8 –	Pinagem do módulo ESP8266-12E NodeMCU.....	20
Figura 9 –	Interface de programação do software IDE.....	20
Figura 10 –	Montagem do circuito para gravação do calendário no módulo RTC DS1307 e visualização no display LCD.....	25
Figura 11 –	Programação utilizada para testar a gravação do calendário no RTC1307 e visualização no display LCD.....	26
Figura 12	Circuito montado para gravação do despertador no Arduino Leonardo em funcionamento após o horário pré-estabelecido..	27
Figura 13.A –	Programação realizada para acionar as portas 9 e 11 em horários pré-determinados.....	27
Figura 13.B –	Programação realizada para acionar as portas 9 e 11 do Arduino em horários pré-determinados.....	28
Figura 14 –	Programação realizada para acionar as portas 8, 9 e 11 em horários pré-determinados.....	28
Figura 15 –	Montagem do circuito do despertador com horários de acionamento para as duas portas.....	29
Figura 16 –	Diagrama de conexão do led e resistor em série elaborado no Circuit Wizard®.....	29
Figura 17 –	Montagem do circuito de acionamento do servomotor com funcionamento através do potenciômetro.....	30
Figura 18 –	Programação realizada para acionamento do servo motor nos ângulos de 0° e 90° através da utilização de um potenciômetro de 10KΩ.....	31
Figura 19 –	Montagem para teste da programação do ESP8266.....	32

Figura 20.A –	Programação da conexão IoT DO NodeMCU ESP8266.....	32
Figura 20.B –	Programação da conexão IoT DO NodeMCU ESP8266.....	33
Figura 21 –	Visualização do monitor serial do IDE.....	33
Figura 22 –	Página HTML com o status das portas D3 e D4 do microcontrolador ESP8266 em estado LOW.....	34
Figura 23 –	Página HTML com o status das portas D3 e D4 do microcontrolador ESP8266 em estado high.....	34
Figura 24 –	Cotação dos palitos de madeira e display LCD 16x2.....	35
Figura 25 –	Vista frontal do protótipo.....	35
Figura 26 –	Respectivamente, vistas laterais, super, inferior e posterior do protótipo.....	36
Figura 27 –	Vista em 3D do protótipo.....	36
Figura 28 –	Colagem dos palitos de madeira no papelão.....	37
Figura 29 –	Utilização de massa F.12 para correção e calafetação das frestas entre os palitos.....	37
Figura 30 –	Processo de acabamento: antes e depois após lixar as madeiras.....	37
Figura 31 –	Montagem do armário.....	38
Figura 32 –	Pintura da caixa com verniz.....	38
Figura 33 –	Ligações e conexões do circuito.....	39
Figura 34 –	Protótipo em funcionamento na visão frontal.....	39
Figura 35 –	Disposição do circuito eletrônico na parte posterior do protótipo.	40
Figura 36 –	ESP8266 conectado na protoboard colada na plataforma juntamente com o relé.....	40
Figura 37 –	Protótipo em cima da plataforma de MDF.....	41
Figura 38 –	Último teste do equipamento antes da apresentação final.....	41
Figura 39 –	Programação final do despertador para acionamento do LED, relé e <i>buzzer</i>	42

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	Doenças crônicas e utilização de fármacos.....	11
1.2	Pandemia do coronavírus (COVID19) e o distanciamento social.....	11
1.3	Domótica.....	13
1.4	Arduino.....	13
2	OBJETIVO.....	14
2.1	Objetivo principal.....	14
2.2	Objetivo específico.....	14
3	FUNDAMENTAÇÃO.....	14
4	MATERIAIS E MÉTODOS.....	21
4.1	Pesquisa de campo.....	23
4.2	Princípio de acionamento.....	24
4.3	Construção.....	24
4.3.1	Programação Arduino e pinagem.....	25
4.3.2	Armário.....	35
5	RESULTADOS.....	43
5.1	Pesquisa de campo.....	43
5.2	Protótipo.....	46
6	CONSIDERAÇÕES FINAIS.....	46
	REFERÊNCIAS.....	48
	APENDICE 1.....	50

1. INTRODUÇÃO

Ao passar dos anos, o crescimento da população idosa tem aumentado significativamente no Brasil. Segundo o IBGE, atualmente, a proporção de pessoas com mais de 65 anos é 1 a cada 10 brasileiros. Em 2060, essa projeção será de 1 a cada 4 brasileiros (CENSO, 2022). Esses valores estão relacionados diretamente a natalidade e a expectativa de vida da população (G1, 2018). Estudos demonstraram que no ano 2000, das 14.536.029 pessoas com 60 anos ou mais no Brasil, 74% residiam na Região Sudeste (46,3%) ou Nordeste (27,27%) (SILVA & JOAQUIM, 2004). Junto com o aumento da idade fisiológica da população brasileira, tem aumentado também as doenças crônicas, consequentemente a utilização de medicamentos.

1.1 Doenças Crônicas e Utilização de Fármacos

A Pesquisa nacional de Saúde (2013), levantou o uso de medicamentos para o tratamento de doenças crônicas não transmissíveis (DCNT) no Brasil. São DCNT doenças cardiovasculares, doenças respiratórias crônicas, diabetes, câncer e outras. Esse grupo de patologias foram as principais causas de morte no mundo, correspondendo a 68% dos óbitos em 2012 (SCHMIDT *et al.*, 2011). Devido a crescente nos números de DCNT, concomitantemente o uso de medicamentos para o controle dessas doenças aumentaram, sendo a Região Sudeste a que tem maior percentual de pessoas acima de 18 anos fazendo o uso de fármacos para tratamento de DCNT (TAVARES *et al.*). Com o passar da idade, é corriqueiro que filhos ou outras pessoas com ou não grau parentesco passem a cuidar das pessoas idosas nas atividades cotidianas como gerenciar e/ou lembrá-las de tomar seus medicamentos em horários pré-determinados quando a mesma faz uso de fármacos.

1.2 Pandemia do Coronavírus (COVID-19) e o distanciamento social

Em determinados momentos, faz-se necessário a presença de alguém ao lado da pessoa idosa para alertá-la sobre o momento correto de ingerir suas medicações. Porém, em 2019, uma questão a nível mundial alterou por completo o comportamento da população e fez com que reduzisse inclusive esse contato presencial de filho, netos e amigos com os pais, tios, colegas, amigos de idade mais avançada.

Trata-se da COVID-19, uma doença infecciosa causada pelo novo coronavírus (SARS-CoV-2), onde os principais sintomas são febre, cansaço e tosse seca. Algumas pessoas podem apresentar dores, congestão nasal, dor de cabeça, conjuntivite, dor de garganta, diarreia, perda de paladar ou olfato, erupção cutânea na pele e discoloração dos dedos das mãos ou dos pés. Geralmente são sintomas leves e começam gradualmente e cerca de 80% das pessoas infectadas se recuperam da doença sem precisar de tratamento hospitalar, no entanto, um em cada seis pessoas infectadas por COVID-19 fica gravemente doente e desenvolve dificuldade para respirar (OPAS, 2019).

Pessoas idosas e as que têm outras condições de saúde como pressão alta, problemas cardíacos e do pulmão, diabetes ou câncer, têm maior risco de ficarem gravemente doente. Mas, qualquer pessoa pode pegar a COVID-19 e ficar gravemente doente (OPAS, 2019). Devido a pandemia e a gravidade do vírus, o mundo tomou medidas mais restritivas, dentre elas a quarentena, onde o isolamento social passou a ser adotado. Na fase inicial, da pandemia, as primeiras curvas de tendência epidemiológicas apontavam os idosos como a população mais exposta a letalidade do vírus, conseqüentemente, o contato com pessoas acima dos 65 anos ficara mais restritas para evitar o contágio (PEIXOTO *et al.*, 2020; BEZZERA *et al.*, 2020).

Durante o enfrentamento da pandemia, muitos protocolos sanitários foram criados na tentativa de conter a disseminação do vírus e suas possíveis mutações. Por meio de protocolos da Organização Mundial da Saúde (OMS), Organização Nacional da Saúde (ONS) e municipais, a rotina das pessoas sofreram muitas alterações. Muitas empresas desfizeram do trabalho presencial e adotaram por um modelo de trabalho a distância por meios remotos denominado *Home Office* (escritório em casa). Essa tendência cresceu muito e passou a ser adotadas nas instituições de ensino, no comércio com vendas online, sistemas *delivery* (sistema remoto de compras com entrega em casa – alimentação, roupas, produtos de higiene pessoal, dentre outros). Tudo isso para evitar o contato entre as pessoas e o contágio pelo vírus e conter a alta taxa de pessoas infectadas, superlotação em leitos hospitalares e principalmente óbitos.

1.3 Domótica

Com as mudanças de comportamento social devido a pandemia, a cultura do *Home Office* cresceu. As pessoas passaram a conviver por muito mais tempo dentro de casa em relação ao período sem pandemia. Dessa forma, a necessidade e a busca por conforto aumentaram, conseqüentemente gerando crescimento da procura pela Domótica, popularmente conhecida como automação residencial (JORNAL AO VIVO DE BRASÍLIA, 2020; EUREKA, 2021).

A domótica pode ser definida como a utilização da eletricidade, da eletrônica e das tecnologias da informação no ambiente residencial, permitindo realizar a sua gestão de modo local ou remoto, oferecendo uma vasta gama de aplicações nas áreas da segurança, conforto, comunicação e gestão de energia (MARIOTONI & ANDRADE, 2007; LINS & MOURA, 2010).

É possível exemplificar atuações da Domótica em situações como poder ligar e desligar as lâmpadas de uma residência através de um pulso no interruptor ou através de um smartphone de modo presencial na instalação ou remotamente à distância (por exemplo: em outro estado), desde que o smartphone esteja conectado a uma rede de *wireless* com *internet* e os módulos de automação da residência também). Outro bom exemplo é a atuação automática de ar-condicionado na residência em um determinado horário pré-programado para quando o dono da residência chegue em casa, a casa esteja na temperatura desejada, ou então programar as luzes do jardim para acenderem determinados dias da semana em determinados horários. Há uma infinidade de atuação da Domótica (MARIOTONI & ANDRADE, 2007; LINS & MOURA, 2010).

1.4 Arduino

No entanto, por mais que haja infinitudes de possibilidades com a Domótica e a procura esteja em constante crescimento no mercado (JORNAL AO VIVO DE BRASÍLIA, 2020), ainda não é tão acessível financeiramente. Pensando em facilitar o acesso ao ensino-aprendizagem e a criação de projetos de automação, em 2005, na Itália, um grupo de 5 pesquisadores criou o Arduino, cujo objetivo era elaborar um dispositivo que pudesse ser ao mesmo tempo barato, funcional e fácil de programar, sendo acessível a estudantes e projetistas amadores, assim foi criada uma placa composta por um microcontrolador Atmel, circuitos de entrada/saída e que pode facilmente ser conectada a um computador e programada via IDE (*Integrated*

Development Environment, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C++, sem a necessidade de equipamentos extras além de um cabo USB (SILVA, 2014; ROCHA, 2016).

O Arduino foi uma grande revolução na área da eletroeletrônica. Atualmente, muitos projetos pioneiros surgem através de criações iniciais utilizando Arduino, incluindo a Domótica, onde o Arduino é muitas vezes empregado como uma alternativa de baixíssimo custo (SILVA, 2014; ROCHA, 2016).

2. OBJETIVO

2.1 Objetivo Principal

Utilizando microcontrolador Arduino® e linguagem C++, desenvolver um armário automatizado de medicamentos de baixo custo acionado por efeito audiovisual em datas e horários pré-estabelecidos, lembrando o usuário de ingerir os medicamentos e, em tempo real, permitir que um responsável acompanhe através de notificações no *smartphone* ou computador.

2.2 Objetivo Específico

- Propiciar autonomia à pessoa idosa não sendo necessário excessiva comunicação oral de alguém próximo para lembrá-la que está no horário de tomar os medicamentos;
- Permitir que pessoas (filhos, neto, cuidador) possam monitorar à distância, através de *smartphone* ou computador com conexões *wireless*, horários de disparo, abertura e fechamento das gavetas, reduzindo a preocupação e gerando maior comodidade;
- Reduzir o contato presencial excessivo (quando possível) não expondo pessoas idosas, gestantes, hipertensas e portadoras de diabetes, contribuindo com os protocolos de redução da taxa de contágio pela COVID19.

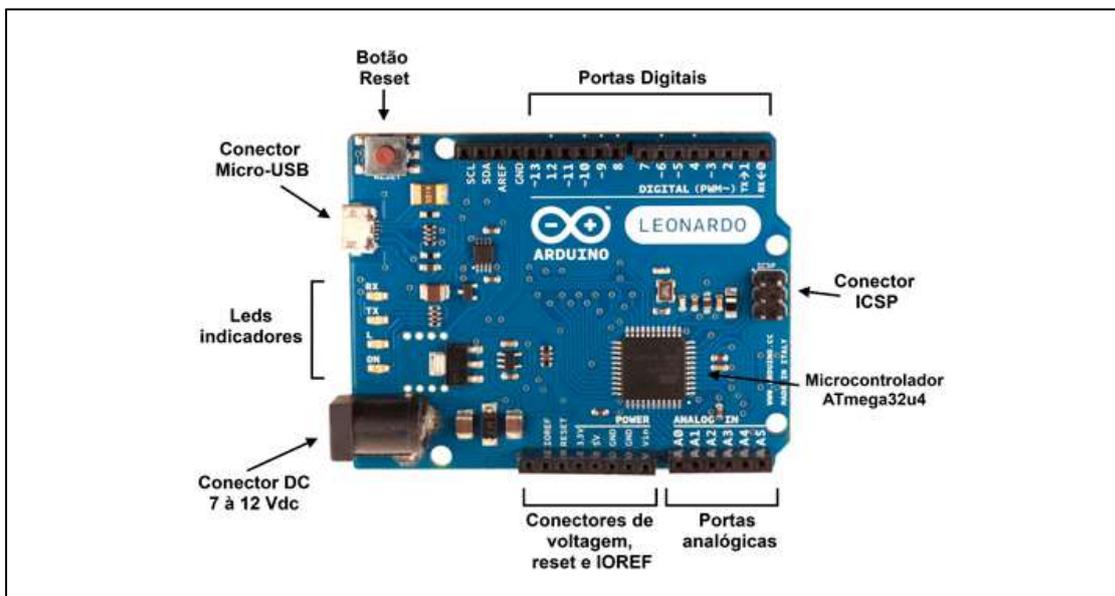
3. FUNDAMENTAÇÃO

Para realização do projeto, foi necessário conhecer e entender mais a fundo sobre os microcontroladores Arduino, os módulos e *Shield*.

O Arduino foi criado em 2005, na Itália, com a finalidade de ser um dispositivo acessível e de fácil programação, permitindo seu acesso para estudantes e projetistas amadores (THOMSEN, 2014). Ele é composto por um microcontrolador Atmel, é composto por circuitos de entradas e saídas, digitais e analógicas, além de fácil conexão com a porta USB de um computador, onde a programação é realizada através do software IDE (*Integrate Development Environment*), do inglês, Ambiente de Desenvolvimento Integrado.

Dois modelos clássicos de microcontrolador são o Arduino Uno e o Arduino Leonardo. Ambas as placas são muito similares entre si, contendo um microcontrolador SMD ou Circuito Integrado ATmega32u4 e comunicação USB embutido na placa. Possuem 20 pinos de entrada e saída, dos quais 7 podem ser usados como saídas PWM (*Pulse Width Modulation*), 12 portas analógicas. A principal diferença entre os dois está na conexão USB da placa, pois o Leonardo utiliza uma conexão micro USB (muito comum em aparelhos celulares também), enquanto o Uno apresenta conexão USB A/B (muito comum em impressoras).

FIGURA 1 – PLACA ARDUINO LEONARDO.



Fonte: FilipeFlop. 2021.

FIGURA 2 – PLACA ARDUINO UNO.

Fonte: Multilógica-shop, 2021.

Em muitas aplicações de automação é necessário a existência de uma tela para facilitar a visualização do processo que está ocorrendo em um evento programado, como por exemplo, na indústria, a interface homem-máquina. Na automação com Arduino, essa aplicação pode ser realizada através da utilização de display LCD. Esses módulos são especificados por sua capacidade gráfica de comunicação, que são os números de caracteres por linha (Ex:1, 2 e 3) e o número de linhas (ex: 8, 12, 16, 20, 24 e 40) (PUHLMANN, 2015).

Um modelo comumente utilizado em projetos de automação com Arduino é o display LCD 16x2 – *BackLight* (luz de fundo) azul. Ele utiliza o controlador HD44780 e pode ser operado em 4 ou 8-bits paralelamente. A conexão desse modelo de display é realizada através de 16 pinos.

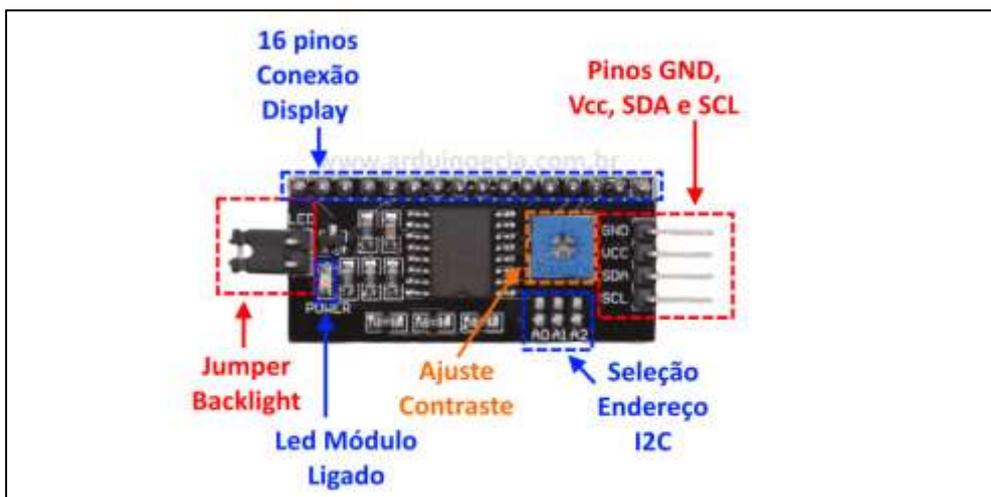
FIGURA 3 – DISPLAY LCD 16X2 BACKLIGHT AZUL.

Fonte: Usinainfo, 2021.

Para reduzir a quantidade de conexões nos pinos do display LCD 16x2, é possível utilizar o módulo I2C. Com ele, reduz-se de 16 para 4 pinos a conexão do display com o microcontrolador Arduino, sendo dois pinos de alimentação (Vcc e GND) e dois de interface I2C (SDA e SCL).

No módulo, tem um potenciômetro que permite ajustar o contraste do display. Também contém um jumper na lateral oposta permitindo que a luz de fundo (backlight) seja controlada pelo programa ou permaneça apagada.

FIGURA 4 – MÓDULO I2C.



Fonte: Arduino e cia, 2021.

Por padrão, o módulo vem configurado de fábrica com o endereço 0x27, no entanto, pode ser alterado através dos pinos A0, A1 e A2 conforme o quadro abaixo:

QUADRO 1 – QUADRO VERDADE ENDEREÇO DO MÓDULO I2C.

Endereço	A0	A1	A2
0x20	0	0	0
0x21	1	0	0
0x22	0	1	0
0x23	1	1	0
0x24	0	0	1
0x25	1	0	1
0x26	0	1	1
0x27	1	1	1

Fonte: Arduino e cia, 2021.

Vasta gama dos microcontroladores funciona independente do tempo real, pois não é uma necessidade eminente. No entanto, caso o programador deseje que sua

programação seja controlada pelo microcontrolador em determinado tempo real (horas e dias), pode-se utilizar o módulo de *Real Time Clock* (RTC), ou simplesmente módulo RTC. Dentre a gama de módulos RTC, tem-se o modelo DS1307, que funciona através da interface I2C de dois fios e retorna uma informação de 7 bytes. Esse módulo é um relógio de tempo real com calendário completo e mais de 56 bytes de SRAM, sendo capaz de fornecer informações como segundo, minutos, dia, data, mês e ano.

FIGURA 5 – MÓDULO RTC DS1307.



Fonte: Shopee, 2021.

Quando se fala em despertador, sistema de automação com microcontrolador com efeito sonoro, não se pode deixar de inserir no projeto um buzzer, esse nada mais é do que um pequeno dispositivo externo com dois pinos para conectá-lo na alimentação e no GND. Seu interior apresenta um piezoelétrico e quando uma corrente é aplicada ao buzzer, ela faz com que o disco de cerâmica (piezoelétrico) se contraia ou se expanda, essa variação faz com que ao redor do disco vibre gerando o som emitido pelo componente.

FIGURA 6 – BUZZER.



Fonte: ByteFlop, 2021.

Além do efeito auditivo, outro efeito que caminha ao lado é o visual. Na eletrônica, quem faz esse papel muitas vezes é o *light-emitting diode* (diodo emissor de luz) – LED. É um semicondutor e a tensão de trabalho é em torno de 2V a 3V e corrente de trabalho de 0,02 amperes (tendo pequenas variações de acordo com a cor). Tem um baixo consumo de energia elétrica, alto rendimento, vida útil longa e causa baixo impacto a natureza devido seus materiais semicondutores, como o silício, por exemplo (HELERBROCK, 2021). Seu funcionamento ocorre quando, polarizado corretamente (Anodo (+) e catodo (GND)) e há passagem de corrente elétrica contínua por seu material, são capazes de emitir luz (MATTEDE, 2021).

FIGURA 7 – DIODO EMISSOR DE LUZ DE DIFERENTES CORES.

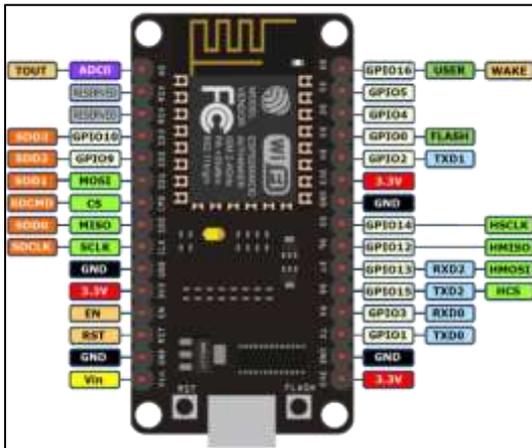


Fonte: Mundo da Elétrica, 2021.

Atualmente, com a crescente na Domótica, muito se fala em dispositivos inteligentes, possibilidade de ligar e desligar aparelhos elétricos a distância através de smartphone, acompanhar remotamente o estado dos equipamentos (ligado ou desligado). Dentre os inúmeros módulos que surgiram recentemente para explorar a onda da Internet das Coisas (IoT), um dos que mais se destacam é o ESP8266, muito utilizado em projetos de automação com microcontrolador (CURVELLO, 2015).

Com o avanço tecnológico, foi desenvolvido o módulo WiFi ESP8266-12E NodeMCU, que é uma placa que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão de 3,3V. Esse equipamento pode ser programado através da linguagem de programação LUA ou então através da própria IDE do Arduino. Esse módulo contém 11 pinos de entrada e/ou saída, além de conversor analógico-digital.

FIGURA 8 – PINAGEM DO MÓDULO ESP8266-12E NodeMCU.



Fonte: Eletrogate, 2021.

Para realização das programações nos microcontroladores Arduino, é necessário a utilização de um software. Trata-se do IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado), onde utilizando uma linguagem baseada em C++, sem a necessidade de equipamentos extras além de um cabo USB compatível com o modelo da placa Arduino sendo utilizada, é possível construir a programação lógica e realizar o upload no dispositivo (SILVA, 2014; ROCHA, 2016). É possível realizar o *download* da ferramenta gratuitamente através do arduino.cc/em/software.

FIGURA 9 – INTERFACE DE PROGRAMAÇÃO DO SOFTWARE IDE.



Fonte: Acervo pessoal do projeto, 2021.

4. MATERIAIS E MÉTODOS

Para o desenvolvimento do presente trabalho, foram utilizados uma gama elevada de materiais e componentes. Desses, tiveram materiais que foram necessários realizar a aquisição através da compra, outros foram através de doações ou empréstimo voluntário de integrantes do grupo e terceiros. Na tabela abaixo, estão listados os materiais utilizados para realização do protótipo:

TABELA 1 – MATERIAIS UTILIZADOS NA REALIZAÇÃO DO PROJETO.

QNTD.	MATERIAL	DESCRIÇÃO
1	Arduino Leonardo	Acionamento do buzzer, LED's e display LCD).
1	Arduino Uno	Acionamento do servo motor.
1	Servomotor	Abertura e fechamento da porta.
1	Módulo RTC1307	Armazenamento digital do calendário.
1	Display LCD 16x2	Visualização data e hora em tempo real.
1	Módulo I2C	Redução de 16 para 4 pinos de conexão o display.
1	ESP8266	Repassar as informações recebidas para o IP do roteador em que está conectado.
1	Buzzer	Emissão do efeito sonoro.
2	Diodo Emissor de Luz (LED)	Emissão do efeito visual.
1	Potenciômetro 10K Ω	Controle do servomotor.
2	Resistor 220 Ω	Dividir a corrente que passa pelos LED's
1	Fim de curso	Informar abertura e fechamento da porta.
1	Protoboard	Ensaio e montagem dos circuitos.
1	Kit de fios para <i>jumper</i> s	Realização das conexões nas montagens.
200	Palitos de madeira	Construção física do protótipo.
1	Caixa de papelão	Construção física do protótipo.
160g	Cola para madeira	Construção física do protótipo.
6	Parafusos	Instalação das portas e display LCD.
4	Porcas	Fixação das dobradiças.
1	Tubo de estanho	Soldar os terminais.
1	Massa F.12	Calafetação do protótipo.
1	Lixa para ferro	Desbastar o excesso de massa F.12
2	Dobradiças	Sustentação, abertura e fechamento portas.
1	Fonte 12Vcc, 1,5A	Alimentação dos microcontroladores.
1	Chave interruptor	Ligar e desligar a alimentação do protótipo.
1	Tubo de cola quente	Realização da colagem dos microcontroladores, servo motor, fim de curso e maçaneta das portas.

Fonte: Acervo pessoal do projeto, 2021.

O custo para realização do protótipo foi de R\$ 170,60 (cento e oitenta e um reais e sessenta centavos). O valor foi esse, graças a materiais doados por terceiros ou emprestados por integrantes do grupo. Na tabela 2, constaram os valores pagos nos materiais.

TABELA 2 – CUSTO DOS MATERIAIS UTILIZADOS NO PRESENTE PROTÓTIPO.

QNTD.	MATERIAL	PREÇO (R\$)
1	Arduino Leonardo	**
1	Arduino Uno	*
1	Servomotor	**
1	Módulo RTC1307	22,90
1	Display LCD 16x2	39,50
1	Módulo I2C	24,90
1	ESP8266	28,00
1	Buzzer	5,00
2	Diodo Emissor de Luz (LED)	0,80
1	Potenciômetro 10K Ω	*
2	Resistor 220 Ω	*
1	Fim de curso	7,00
1	Protoboard	*
1	Kit de fios para <i>jumpers</i>	*
200	Palitos de madeira	10,00
1	Caixa de papelão	***
160g	Cola para madeira	10,00
6	Parafusos	2,50
4	Porcas	0,50
1	Tubo de estanho	*
1	Massa F.12	16,00
1	Lixa para ferro	*
2	Dobradiças	12,00
1	Fonte 12Vcc, 1,5A	*
1	Chave interruptor	*
1	Tubo de cola quente	3,50
TOTAL		R\$ 170,60

Legenda: *Materiais do componente do grupo; **Materiais doados; ***Materiais gratuitos. Fonte: Acervo pessoal do projeto, 2021.

Para construção do protótipo também foram necessárias ferramentas específicas, desde software para construção das programações até ferramentas

eletromecânicas. Na tabela 3, estão descritas as ferramentas e a descrição de aplicação das mesmas no presente protótipo.

TABELA 3 – FERRAMENTAS UTILIZADAS.

QNTD.	MATERIAL	DESCRIÇÃO
1	software Arduino IDE®	Programação dos microcontroladores.
1	software web Adafruit®	Interface entre os microcontroladores e o smartphone.
1	Revolver de cola quente	Realização da colagem dos microcontroladores, servo motor, fim de curso e maçaneta das portas.
1	Ferro de solda	Para soldar os condutores.
1	Fim de curso	Informar abertura e fechamento da porta.
1	Furadeira	Realização de furos nos palitos de madeiras.
1	Micro retifica	Realização dos cortes e furos nos palitos de madeira e papelões.
1	Régua de 30 cm	Para realização das medidas para recortes.
1	Lápis	Para realizar as marcações.
1	Chave philips	Para parafusar o display LCD.
1	Chave de fenda	Para parafusar as dobradiças.

Fonte: Acervo pessoal do projeto, 2021.

4.1 Pesquisa de campo

Após o surgimento da ideia, para entender a opinião do público, aceitação do projeto e de obter sugestões para enriquecerem o armário automatizado, foi realizada uma pesquisa de campo contendo cinco perguntas de modo remoto através da ferramenta *Google Forms* (adendo 1).

A pesquisa de campo foi divulgada através da rede social *WhatsApp*. Os integrantes da equipe divulgaram em seus nichos de contatos de trabalho, escola e familiar, envolvendo pessoas de ambos os sexos, entre 16 e 45 anos, abrangendo pessoas com ensino fundamental, médio, técnico e superior completos e incompletos.

Os dados pessoais dos participantes como nome, e-mail e número do documento de identidade não foram necessários, tornando a pesquisa totalmente segura e sigilosa.

4.2 Princípio de funcionamento

Inicialmente seria um gaveteiro automatizado, no entanto, pensando na viabilidade de confecção do protótipo, o projeto tornou-se um armário automatizado, pois conta com duas portas com dobradiça ao invés de gavetas. Além disso, pensando simular que o produto permita atender diferentes perfis de públicos, as duas portas apresentam diferentes mecanismos de funcionamento, sendo:

Porta 1: ao ser acionado, dispara um efeito sonoro e de iluminação ao lado da porta do armário e, tanto para abri-la quanto fechá-la, será necessário rotacionar um potenciômetro, respectivamente, no sentido anti-horário e horário.

Porta 2: ao ser acionada, dispara um efeito sonoro e de iluminação ao lado da porta do armário e, para abri-la, será necessário aplicar força com as mãos (como uma porta de armário convencional);

Para determinar o horário de funcionamento, primeiramente, o usuário escolhe qual gaveta vai querer programar, em sequência, ajusta o horário (exemplo: 7h00 da manhã), quando der o determinado horário, um *buzzer* será disparado com sonorização intercalada a cada 1 segundo e um *light-emitting diode* (diodo emissor de luz) – LED de cor vermelha também será acionado oscilando entre acender e apagar a cada um segundo.

No mesmo instante em que for disparado o efeito audiovisual, através de comunicação *Wireless* entre o módulo ESP8266 e um roteador de internet, será possível acompanhar no smartphone ou *desktop* informando que o porta-remédios foi acionado e qual é porta acionada.

Também pelo smartphone, é possível acompanhar em tempo real se a porta foi aberta e fechada pelo usuário, dando a entender se ele tomou ou não o medicamento.

Quanto a pessoa que está usufruindo do equipamento, ela poderá visualizar o horário atual através de um display LCD no equipamento, identificar qual a porta que deve ser aberta através de um LED piscando em frente a porta.

Pretendia-se que o efeito audiovisual fosse desligado apenas após a sequência de a porta ser aberta e fechada, no entanto, não foi possível tal *performance* na programação realizada.

4.3 Construção

A construção do protótipo deu-se por etapas que ocorreram paralelamente. Sendo a elaboração das programações em linguagem C++ e a confecção do armário.

4.3.1 Programação Arduino e Pinagem

A linguagem utilizada na programação do microcontrolador Arduino e no ESP8266 foi a linguagem C++, uma linguagem de programação de nível médio desenvolvida na década de 1980 (PACIEVITCH, 2021). A ferramenta utilizada foi o software Arduino IDE® disponibilizado e adquirido gratuitamente através do site da empresa Arduino (disponível em: <https://www.arduino.cc/>).

Inicialmente, o módulo I2C foi soldado no display LCD 16x2 com para reduzir de 16 para apenas 4 terminais de conexões do *display* (SDA, SCL, GND e VCC). Na sequência, utilizando fios para *jumpers* e uma placa *protoboard*, foi realizada as conexões na placa Arduino Leonardo com o RTC 1307 e o módulo I2C. Os pinos (-) e (+) do módulo I2C e RTC1307 foram conectados nas portas de alimentação GND e 5Vcc do Arduino. Os pinos SCL e SDA do módulo I2C e RTC1307 foram conectados, respectivamente, nas portas SCL e SDA do Arduino (Figura 10).

FIGURA 10 – MONTAGEM DO CIRCUITO PARA GRAVAÇÃO DO CALENDÁRIO NO MÓDULO RTC DS1307 E VISUALIZAÇÃO NO DISPLAY LCD.



Fonte: Acervo pessoal do projeto, 2021.

Após essa etapa inicial de montagem, foi realizada a programação no IDE (Figura 10) para gravar o calendário (data e hora) no RTC e permitir a visualização através do display.

FIGURA 11 – PROGRAMAÇÃO UTILIZADA PARA TESTAR A GRAVAÇÃO DO CALENDÁRIO NO RTC1307 E VISUALIZAÇÃO NO DISPLAY LCD.

```

1  #include <LiquidCrystal_I2C.h>    // Inclui a Biblioteca
2  #include <RTClib.h>              // Inclui a Biblioteca
3
4  RTC_DS1307 rtc;
5  LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia o display 16x2 no endereço 0x27
6
7  void setup() {
8      rtc.begin();                 // Inicia o módulo RTC
9
10     rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Ajuste Automático da hora e
    data
11     //rtc.adjust(DateTime(2019, 11, 29, 10, 23, 00)); // Ajuste Manual (Ano, Mês,
    Dia, Hora, Min, Seg)
12
13     lcd.init();                  // Inicia o Display
14     lcd.backlight();            // Inicia o Backlight
15 }
16
17 void loop() {
18     DateTime now = rtc.now();
19
20     lcd.setCursor(1, 0);         // Posiciona o cursor na primeira linha
21     lcd.print("Hora: ");         // Imprime o texto "Hora: "
22     lcd.print(rtc.now().hour()); // Imprime a Hora
23     lcd.print(":");              // Imprime o texto entre aspas
24     lcd.print(rtc.now().minute()); // Imprime o Minuto
25     lcd.print(":");              // Imprime o texto entre aspas
26     lcd.print(rtc.now().second()); // Imprime o Segundo
27
28     lcd.setCursor(0, 1);        // Posiciona o cursor na segunda linha
29     lcd.print("Data: ");         // Imprime o texto entre aspas
30     lcd.print(rtc.now().day());  // Imprime o Dia
31     lcd.print("/");             // Imprime o texto entre aspas
32     lcd.print(rtc.now().month()); // Imprime o Mês
33     lcd.print("/");             // Imprime o texto entre aspas
34     lcd.print(rtc.now().year()); // Imprime o Ano
35
36     delay(1000);                // Aguarda 1 segundo e reinicia
37     lcd.clear();
38 }
39

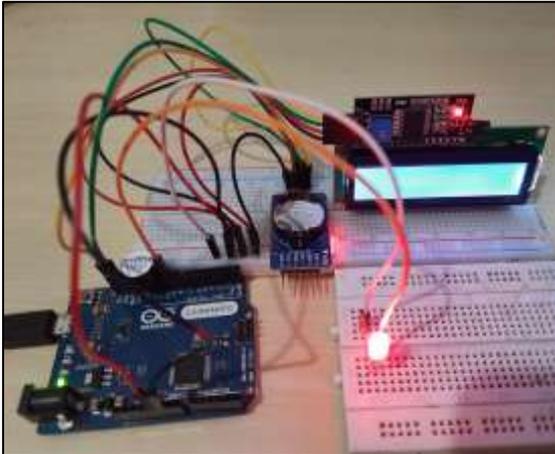
```

Fonte: Acervo pessoal do projeto, 2021.

Após a etapa acima, foi realizada uma nova montagem, dessa vez foram adicionados ao diagrama um LED e o *buzzer* para o acionamento do despertador. Alterou muito pouco em relação à anterior, pois foi mantida e apenas foi acrescida a conexão do *buzzer*, onde o pino (-) foi ligada a porta GND próxima as portas SCL e SDA, enquanto o pino (+) conectado à porta 11. Já o (+) do LED foi conectado na porta 9 e o (-) no GND do microcontrolador (Figura 12).

Na sequência, uma nova programação foi realizada, dessa vez para acionar o sistema de alarme. Uma vez que o calendário está gravado no RTC, agora foi configurado o microcontrolador para acionar as portas 9 e 11 em horários pré-estabelecidos na programação, no caso, o horário definido para atuação do alarme foi às 9 horas e 25 minutos (Figuras 13.A e 13.B).

FIGURA 12 – CIRCUITO MONTADO PARA GRAVAÇÃO DO DESPERTADOR NO ARDUINO LEONARDO EM FUNCIONAMENTO APÓS O HORÁRIO PRÉ-ESTABELECIDO.



Fonte: Acervo pessoal do projeto, 2021.

FIGURA 13.A – PROGRAMAÇÃO REALIZADA PARA ACIONAR AS PORTAS 9 E 11 EM HORÁRIOS PRÉ-DETERMINADOS.

```

1
2  #include <LiquidCrystal_I2C.h>    // Inclui a Biblioteca
3  #include <RTCLib.h>              // Inclui a Biblioteca
4
5  RTC_DS1307 rtc;
6  LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia o display 16x2 no endereço 0x27
7
8  #define buzzer 11 // Define o buzzer como pino 11
9  #define led1 9 // Define o led da porta 1 como pino 9
10
11 int Hor; // Inicia a variável Hor
12 int Min; // Inicia a variável Min
13 int Seg; // Inicia a variável Seg
14
15 void setup() {
16   pinMode(buzzer, OUTPUT); // Definido o estado da porta (saída)
17   pinMode(led1,OUTPUT); // Definido o estado da porta (saída)
18   rtc.begin(); // Inicia o módulo RTC
19   lcd.init(); // Inicia o Display
20   lcd.backlight(); // configura a porta serial
21 }
22
23 void loop() {
24   DateTime now = rtc.now();
25   Hor = rtc.now().hour(); // Chama o horário de Hor
26   Min = rtc.now().minute(); // Chama o minuto de Min
27   Seg = rtc.now().second(); // Chama os segundos de Seg
28
29   // Define o horário do despertador e executa o som do buzzer
30   if ( Hor == 9 && Min == 25) {
31     digitalWrite(led1,HIGH);
32     digitalWrite(buzzer, HIGH);
33     delay(1000);
34     digitalWrite(led1,LOW);
35     digitalWrite(buzzer,LOW);
36   }
37   lcd.setCursor(1, 0); // Posiciona o cursor na primeira linha
38   lcd.print("Hora: "); // Imprime o texto "Hora: "
39   lcd.print(Hor); // Imprime a Hora
40   lcd.print(":"); // Imprime o texto entre aspas
41   lcd.print(Min); // Imprime o Minuto
42   lcd.print(":"); // Imprime o texto entre aspas
43   lcd.print(Seg); // Imprime o Segundo
44

```

Fonte: Acervo pessoal do projeto, 2021.

FIGURA 13.B – PROGRAMAÇÃO REALIZADA PARA ACIONAR AS PORTAS 9 E 11 DO ARDUINO EM HORÁRIOS PRÉ-DETERMINADOS.

```

45  lcd.setCursor(0, 1);           // Posiciona o cursor na segunda linha
46  lcd.print("Data: ");          // Imprime o texto entre aspas
47  lcd.print(rtc.now().day());   // Imprime o Dia
48  lcd.print("/");              // Imprime o texto entre aspas
49  lcd.print(rtc.now().month()); // Imprime o Mês
50  lcd.print("/");              // Imprime o texto entre aspas
51  lcd.print(rtc.now().year());  // Imprime o Ano
52
53  delay(1000);                  // Aguarda 1 segundo e reinicia
54  lcd.clear();
55  }
56

```

Fonte: Acervo pessoal do projeto, 2021.

Após os testes, foram acrescentadas mais dez linhas de programação para desenvolver também o acionamento do LED da segunda porta (Figura 14), essa definida como porta 8 do Arduino. O acionamento das portas 9 e 11 foram mantidas às 9 horas e 25 minutos, enquanto às portas 8 e 11 foram programadas para serem acionadas às 9 horas e 35 minutos (Figura 15). As linhas 1 a 7 e após o colchete da linha 46, mantêm-se idênticas as das Figuras 13.A e 13.B.

FIGURA 14 – PROGRAMAÇÃO REALIZADA PARA ACIONAR AS PORTAS 8, 9 E 11 EM HORÁRIOS PRÉ-DETERMINADOS.

```

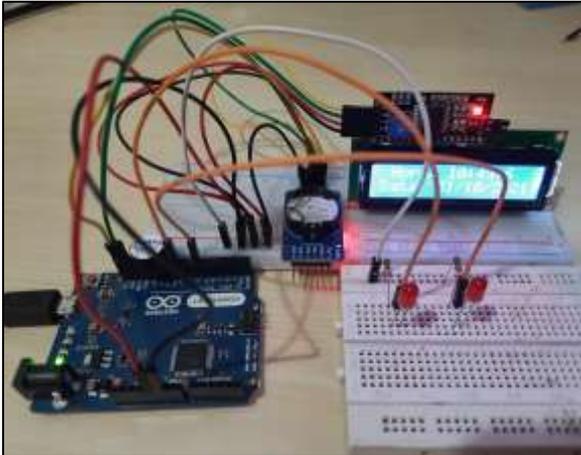
8   #define buzzer 11 // Define o buzzer como pino 11
9   #define led1 9 // Define o led da porta 1 como pino 9
10  #define led2 8 // Define o led da porta 2 como pino 8
11
12  int Hor;           // Inicia a variável Hor
13  int Min;           // Inicia a variável Min
14  int Seg;           // Inicia a variável Seg
15
16  void setup() {
17    pinMode(buzzer, OUTPUT); // Definido o estado da porta (saída)
18    pinMode(led1, OUTPUT);   // Definido o estado da porta (saída)
19    pinMode(led2, OUTPUT);   // Definido o estado da porta (saída)
20    rtc.begin();             // Inicia o módulo RTC
21    lcd.init();              // Inicia o Display
22    lcd.backlight();         // configura a porta serial
23  }
24
25  void loop() {
26    DateTime now = rtc.now();
27    Hor = rtc.now().hour();   // Chama o horário de Hor
28    Min = rtc.now().minute(); // Chama o minuto de Min
29    Seg = rtc.now().second(); // Chama os segundos de Seg
30
31    // Define o horário do despertador e executa o som do buzzer
32    if ( Hor == 9 && Min == 25) {
33      digitalWrite(led1, HIGH);
34      digitalWrite(buzzer, HIGH);
35      delay(1000);
36      digitalWrite(led1, LOW);
37      digitalWrite(buzzer, LOW);
38    }
39    // Define o horário do despertador e executa o som do buzzer
40    if ( Hor == 9 && Min == 35) {
41      digitalWrite(led1, HIGH);
42      digitalWrite(buzzer, HIGH);
43      delay(1000);
44      digitalWrite(led1, LOW);
45      digitalWrite(buzzer, LOW);
46    }

```

Fonte: Acervo pessoal do projeto, 2021.

No modelo para atender horários em duas portas diferentes, foi mantido as conexões já realizadas anteriormente, sendo acrescentado um LED a mais, que por sua vez o terminal (+) foi conectado à porta 8 e o (-) no GND do equipamento.

FIGURA 15 – MONTAGEM DO CIRCUITO DO DESPERTADOR COM HORÁRIOS DE ACIONAMENTO PARA AS DUAS PORTAS.



Fonte: Acervo pessoal do projeto, 2021.

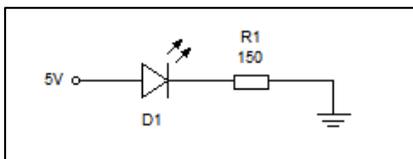
É importante frisar que, para não queimar os LED's devido sobrecarga de tensão e corrente elétrica, foi adicionado um resistor de 150Ω em série entre cada terminal (-) do LED e o GND da protoboard (imagem 16). A equação utilizada para calcular o resistor ideal foi:

$$V_{\text{alimentação}} - V_{\text{led}} / A_{\text{led}}$$

Onde:

- $V_{\text{alimentação}}$ = Tensão de alimentação (5 volts);
- V_{led} = Tensão de trabalho do LED (2 volts);
- A_{led} = Corrente de trabalho do LED (0,02 Amperes).

FIGURA 16 – DIAGRAMA DE CONEXÃO DO LED E RESISTOR EM SÉRIE ELABORADO NO CIRCUIT WIZARD®.



Fonte: Acervo pessoal do projeto, 2021.

Depois da realização com êxito nos testes de acionamento dos alarmes através do microcontrolador, foi desenvolvida a programação para o acionamento automático das gavetas através do acionamento do servo motor. O sistema conflitou a realização da programação da abertura automática das gavetas do acionamento do servo motor e acionamento do *buzzer* led. Então o grupo realizou a programação de abertura e

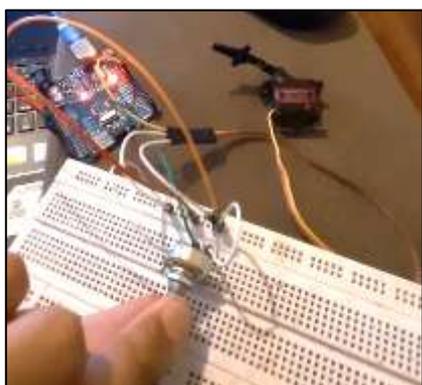
fechamento da porta em um microcontrolador separado, nesse caso, em um Arduino Uno.

Pensando na funcionalidade e garantia de fidedignidade das informações dos eventos transmitidos em tempo real para o smartphone, foi definido, a título de simulação e demonstração no protótipo, o armário automático será acionado através de um potenciômetro. Ao atingir determinada resistência, a porta analógica A0 do Arduino receberá a informação, o microcontrolador irá processar, interpretar e na sequência encaminhar o comando para o servo motor através da porta digital 9, e de acordo com a resistência, o servo motor irá rotacionar em uma angulação determinada. Permitindo assim a abertura e/ou fechamento da porta.

Inicialmente, foram utilizados dois trimpots ajustados com resistências suficientes para rotacionar o servo motor, respectivamente, em ângulos de 0° e 90° , além de um interruptor de alavanca paralelo para fazer a seletividade em qual ângulo disparar o servo motor, no entanto, devido aos trimpots utilizados não serem novos, após alguns testes, passaram a apresentar falha nos dispositivos. Dessa forma, o potenciômetro foi mantido de forma oficial no protótipo.

O circuito com o servo motor, foi o menos complexo para construção física. Os pinos de alimentação (-) e (+) do servo motor, foram conectados, respectivamente, nas portas GND e 5VCC do Arduino, enquanto pino de sinal foi conectado na porta 9~ do Arduino. O potenciômetro, responsável por definir a posição do servo motor conforme variação da sua resistência, os dois pinos laterais foram polarizados sendo conectado um pino na porta GND do microcontrolador, enquanto o outro foi conectado no 5VCC. Já o pino central, foi ligado a porta analógica A0 do Arduino, sendo tal porta responsável por receber o sinal.

FIGURA 17 – MONTAGEM DO CIRCUITO DE ACIONAMENTO DO SERVOMOTOR COM FUNCIONAMENTO ATRAVÉS DO POTENCIÔMETRO.



Fonte: Acervo pessoal do projeto, 2021.

FIGURA 18 – PROGRAMAÇÃO REALIZADA PARA ACIONAMENTO DO SERVO MOTOR NOS ÂNGULOS DE 0° E 90° ATRAVÉS DA UTILIZAÇÃO DE UM POTENCIÔMETRO DE 10KΩ.

```

1  #include <Servo.h>;      // Inclui a Biblioteca Servo.h
2
3  Servo meuservo;         // Inicializa o servo no modo de teste
4  int angulo = 0;         // Ajusta o ângulo inicial do Servo
5  int potencio = A0;      // Inicializa o pino analógico para o potenciômetro
6
7  void setup() {
8  meuservo.attach(9);    // Define que o Servo está conectado a Porta 9
9  }
10
11 void loop() {
12 angulo = analogRead(potencio); // Faz a leitura do valor do potenciômetro
13 angulo = map(angulo, 0, 1023, 0, 179); // Associa o valor do potenciômetro ao valor
do ângulo
14 meuservo.write(angulo); // Comando para posicionar o servo no ângulo
especificado
15 delay(5);
16 }
17

```

Fonte: Acervo pessoal do projeto, 2021.

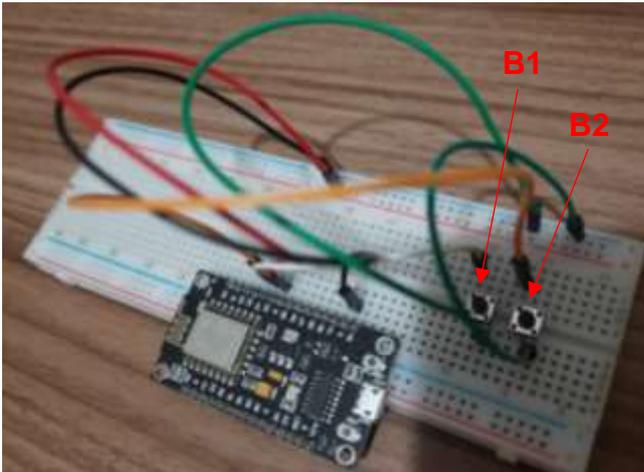
3.4.2 Programação ESP8266 e Pinagem

A comunicação entre o Arduino e o *smartphone* será realizada pelo NodeMcu, também popularmente conhecido como ESP8266®. As informações são repassadas através da conexão Wireless do NodeMcu com a rede WiFi. Será gerado um IP para essa comunicação e, através desse número, ao acessá-lo como página da web, será possível realizar a leitura do estado das portas D3 e D4, respectivamente “Porta fechada” ou “Porta aberta” e “Alarme ligado” ou “Alarme desligado”. Será possível, graças a linguagem HTML que será adicionada na programação, onde o estado de alto (HIGH) ou baixo (LOW) serão convertidos em mensagem de texto e impressa em tempo real na página web do IP gerado.

Para realizar os testes, a montagem do circuito está representada na Figura19. A alimentação do circuito foi através do cabo USB tipo C, a porta D3 foi conectada ao terminal do *push button* e o outro terminal foi ligado ao GND da placa. A porta D4, foi conectada ao segundo *push button* e o outro terminal também foi no GND. Ambos os terminais foram utilizados na função normal aberto e as portas D3 e D4 configuradas como entrada (input) na função *Input_Pullup* (função entrada com resistor interno).

Para construção da programação no ESP8266 NodeMCU (Figuras 20.A e 20.B), devido sua elevada complexidade por conter linguagem HTML, foi necessário recorrer a apostila *Internet of Things – Iniciando IoT com ESP8266* da UNIVAP disponibilizada na integra (PERA, 2021).

FIGURA 19 – MONTAGEM PARA TESTE DA PROGRAMAÇÃO DO ESP8266.



Fonte: Acervo pessoal do projeto, 2021.

FIGURA 20.A – PROGRAMAÇÃO DA CONEXÃO IoT DO NodeMCU ESP8266.

```

1  #include <ESP8266WiFi.h>           //Inclusão da biblioteca ESP8266WiFi
2
3  const byte Porta = 2;              //Definindo a pino D4 como Porta
4  const byte Alarme = 0;            //Definindo a porta D3
5  const char* ssid = "PROJETO DE ICC"; //Nome da rede WiFi = "*****"
6  const char* password = "tcc1234"; //Senha da rede WiFi = "*****"
7
8  WiFiServer server(80); //Shield irá receber as requisições das páginas (o padrão
  WEB é a porta 80)
9
10 void setup() {
11     pinMode(Porta, INPUT_PULLUP); // definindo o pino D4 como entrada com
    resistor interno
12     pinMode(Alarme, INPUT_PULLUP); // definindo o pino D3 como entrada com
    resistor interno
13     Serial.begin(115200); //Velocidade de atualização de escaneamento da página
14
15     //Conexão na rede WiFi
16     Serial.println();
17     Serial.print("Conectando a ");
18     Serial.println(ssid);
19
20     WiFi.begin(ssid, password);
21
22     while (WiFi.status() != WL_CONNECTED) {
23         delay(500); //delay tentativas de conexão
24         Serial.print("."); //Aparecer "..." quando estiver conectando
25     }
26     Serial.println("");
27     Serial.println("WiFi conectado!"); //Mensagem na tela de scan ao conectar no WiFi
28
29     // Inicia o servidor WEB
30     server.begin();
31     Serial.println("Server iniciado");
32
33     // Mostra o endereço IP
34     Serial.println(WiFi.localIP());
35
36 }

```

Fonte: Acervo pessoal do projeto, 2021.

FIGURA 20.B – PROGRAMAÇÃO DA CONEXÃO IoT DO NodeMCU ESP8266.

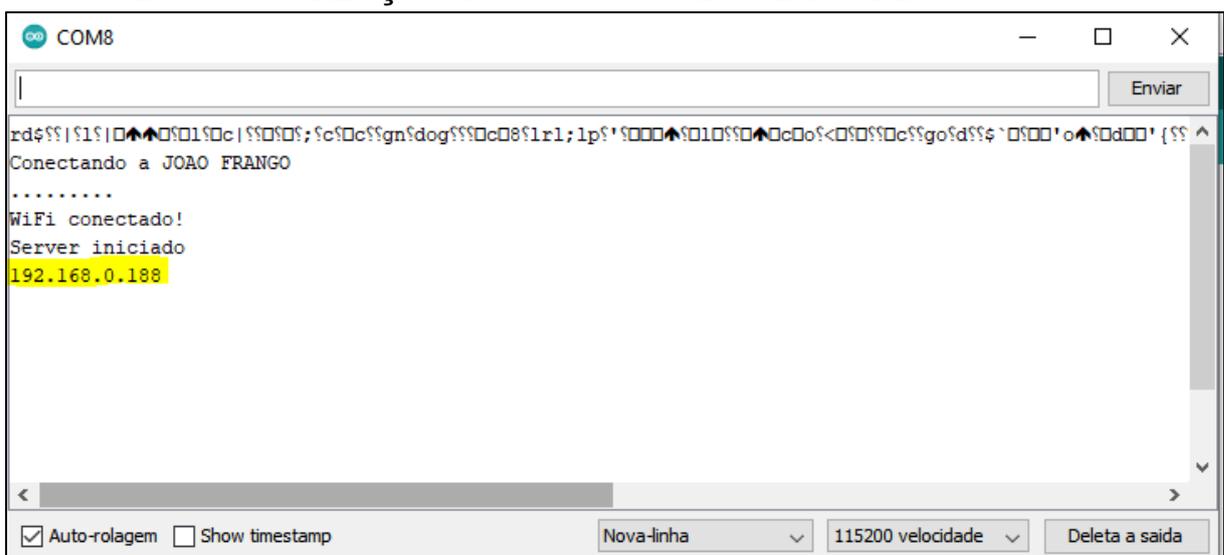
```

37
38 void loop() {
39   WiFiClient client = server.available();
40   if(!client) {
41     return;
42   }
43   while (!client.available()){
44     delay(1);
45   }
46   client.println("<!DOCTYPE HTML>");
47   client.println("<html>");
48
49   client.println("HTTP/1.1 200 OK");
50   client.println("Content-Type: text/html");
51   client.println("");
52
53   client.println("<meta http-equiv='refresh' content='2'>");
54   client.println("<h1>ARMARIO AUTOMATIZADO DE MEDICAMENTOS</h1>"); //Titulo da página
HTML
55   client.println("<h2>Status de Acionamento:<h2>"); //Substituto da página HTML
56   if(digitalRead(Porta) == LOW) {
57     client.println("<p>PORTA ABERTA<p>"); //Estado da mensagem da porta em LOW
58   } else {
59     client.println("<p>PORTA FECHADA<p>"); //Estado da mensagem da porta em HIGH
60   }
61   client.println("<meta http-equiv='refresh' content='0'>");
62   if(digitalRead(Alarme) == LOW) {
63     client.println("<p>ALARME LIGADO<p>"); //Estado da mensagem do alarme em LOW
64   } else {
65     client.println("<p>ALARME DESLIGADO<p>"); //Estado da mensagem do alarme em HIGH
66   }
67   client.println("</html>");
68 }
69

```

Fonte: Acervo pessoal do projeto, 2021.

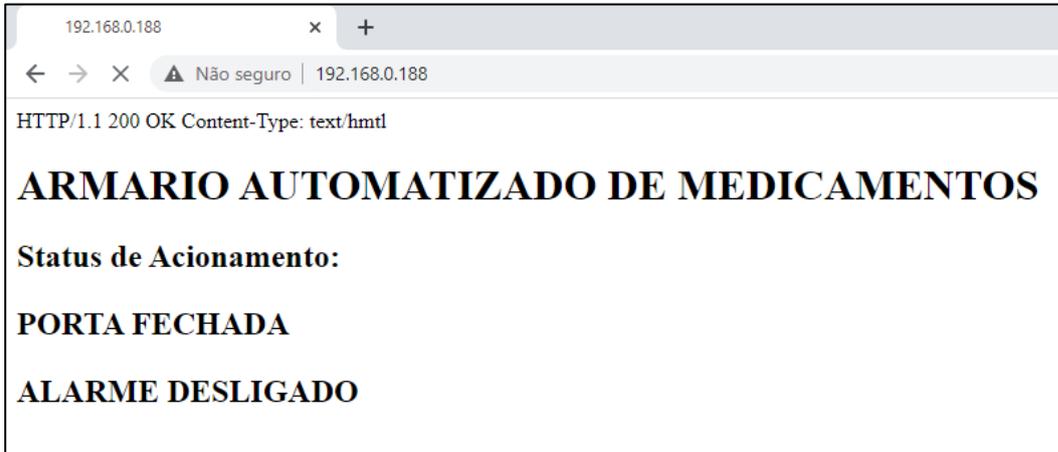
Após carregar a programação no dispositivo, foi aberto o monitor serial (lupa) do IDE. Após clicar no botão *reset* no ESP, logo conectou na rede WiFi e gerou um número de IP (Figura 21).

FIGURA 21 – VISUALIZAÇÃO DO MONITOR SERIAL DO IDE.

Fonte: Acervo pessoal do projeto, 2021.

Através desse número de IP, foi acessado a página *Web* através do navegador Google Chrome® e foi possível se deparar com a tela HTML gerada pelo ESP8266.

FIGURA 22 – PÁGINA HTML COM O STATUS DAS PORTAS D3 E D4 DOMICROCONTROLADOR ESP8266 EM ESTADO LOW.



Fonte: Acervo pessoal do projeto, 2021.

Quando a porta D3 e a porta D4 saem do estado 0 e passam para estado 1, instantaneamente, a informação é enviada via WiFi para o roteador e o a visualização do status dessas portas são atualizados simultaneamente, podendo ser visualizada na página.

FIGURA 23 – PÁGINA HTML COM O STATUS DAS PORTAS D3 E D4 DOMICROCONTROLADOR ESP8266 EM ESTADO HIGH.



Fonte: Acervo pessoal do projeto, 2021.

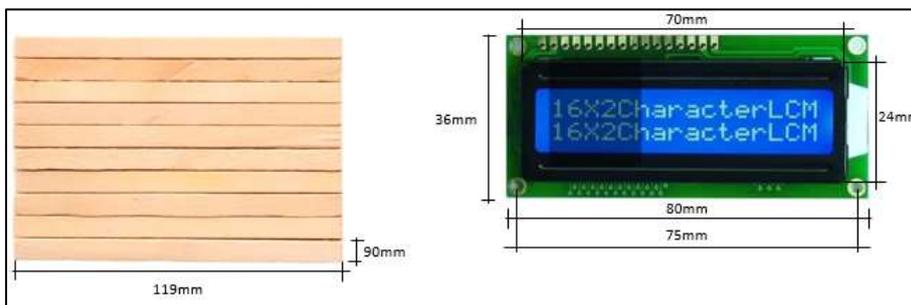
Para a versão final do protótipo, no lugar dos *push button*, foram inseridos um fim de curso como interruptor na porta D4 (porta) e um relé na porta D3 (alarme). No fim de curso, a conexão do borne comum (COM) ao GND e o borne normal aberto (NA) no D4. Foi inserido o relé na porta D3, pois este, é disparado pelo Arduino Leonardo, que tem saída de 5Vcc, e como o ESP8266 trabalha em 3,3Vcc e faz-se

necessário o disparo através do GND de referência do NodeMCU, o Arduino Leonardo acionará o relé, que estará com o terminal comum conectado ao GND do ESP, enquanto o terminal normal aberto estará conectado na porta D3, assim, fazendo o acionamento sempre que a bobina do relé for energizado.

4.3.2 Armário

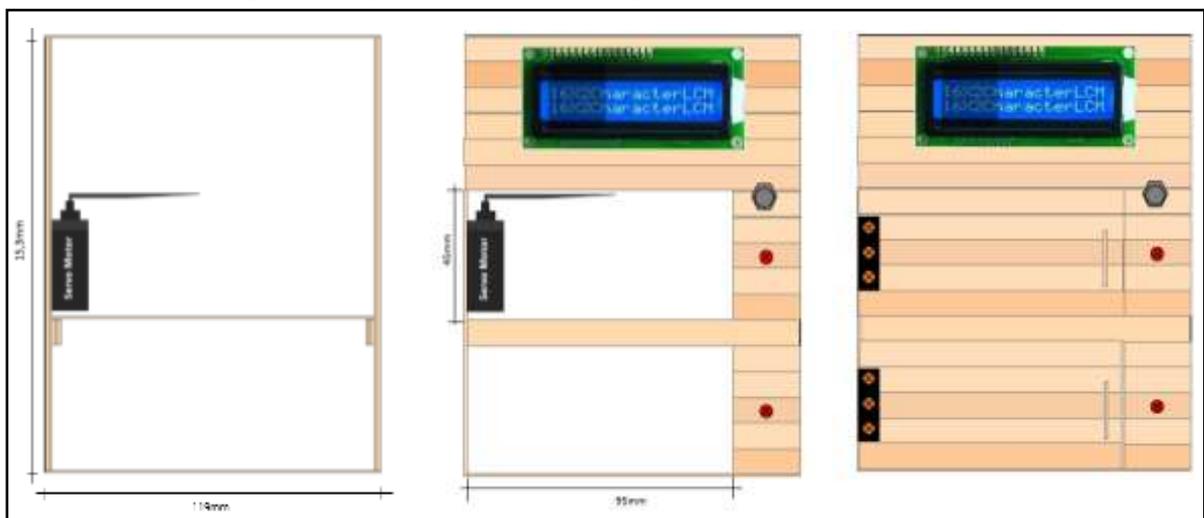
Quanto a construção do armário, inicialmente foi realizado alguns desenhos das cotações em papel sulfite A4. Algumas discussões foram realizadas, onde chegou-se à conclusão para que de fato fosse desenvolvido um armário (contendo portas), ao invés de gaveteiro (com gavetas), devido aos mecanismos de funcionamento e os componentes disponíveis pelo grupo para a confecção. Além, disso definiu-se os materiais para construção. Na sequência, o desenho foi transferido para o programa Microsoft PowerPoint para permitir uma melhor visualização do layout e das cotações.

FIGURA 24 – COTAÇÃO DOS PALITOS DE MADEIRA E DISPLAY LCD 16X2.



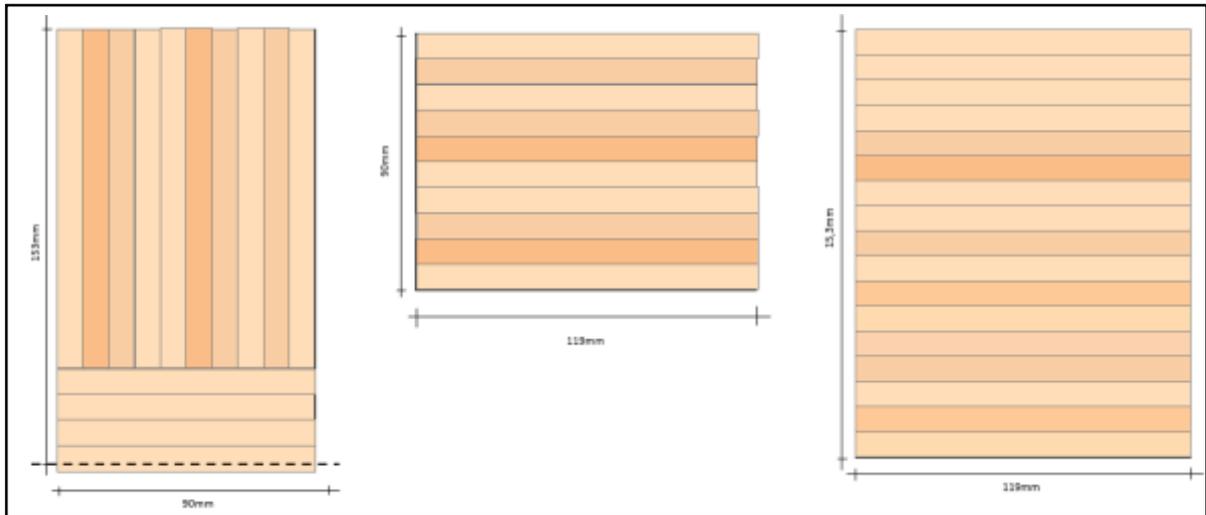
Fonte: Google Imagens, 2021.

FIGURA 25 – VISTA FRONTAL DO PROTÓTIPO.



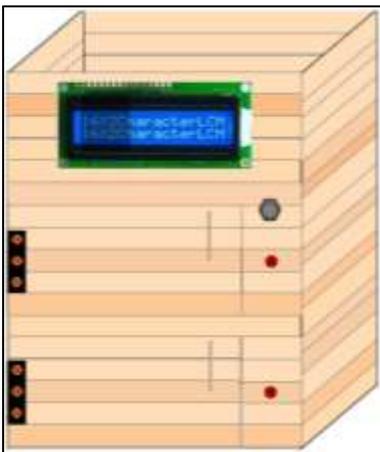
Fonte: Acervo pessoal do projeto, 2021.

FIGURA 26 – RESPECTIVAMENTE, VISTAS LATERAIS, SUPER, INFERIR E POSTERIOR DO PROTÓTIPO.



Fonte: Acervo pessoal do projeto, 2021.

FIGURA 27 – VISTA EM 3D DO PROTÓTIPO.



Fonte: Acervo pessoal do projeto, 2021.

Após o *template* ilustrativo com o *layout* e cotações do protótipo, deu-se início a construção do equipamento. A primeira etapa foi utilizando régua de 30 cm, esquadro de construção civil e lapiseira com ponta 0,7mm, realizada as cotações nos papelões e em seqüências os palitos foram colados utilizando cola para madeira Almafex®. Os cortes necessários dos palitos de madeira foram realizados utilizando micro retifica Dremel® (Figura 28).

Após 12 horas de secagem da cola, utilizando os dedos da mão como ferramenta, foi passada massa F.12 para correção e calafetação das frestas entre os palitos (Figura 28). Quando a secou, o excesso foi desgastado utilizando lixa para ferro número 80, permitindo um ótimo acabamento (Figura 29).

FIGURA 28 – COLAGEM DOS PALITOS DE MADEIRA NO PAPELÃO.



Fonte: Acervo pessoal do projeto, 2021.

FIGURA 29 – UTILIZAÇÃO DE MASSA F.12 PARA CORREÇÃO E CALAFETAÇÃO DAS FRESTAS ENTRE OS PALITOS.



Fonte: Acervo pessoal do projeto, 2021.

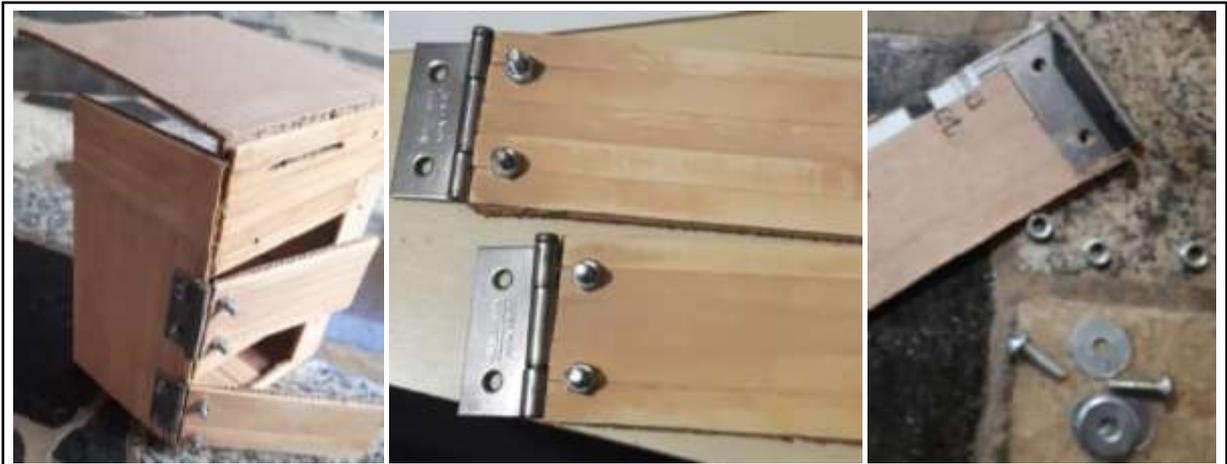
FIGURA 30 – PROCESSO DE ACABAMENTO: ANTES E DEPOIS APÓS LIXAR AS MADEIRAS.



Fonte: Acervo pessoal do projeto, 2021.

Por fim, após as colagens, calafetação e processo de lixamento, as partes foram colocadas, formando a estrutura do protótipo. Para prender as portas, dobradiças foram utilizadas, sendo fixadas com o uso de parafusos e portas, não sendo necessário ruelas.

FIGURA 31 – MONTAGEM DO ARMÁRIO.



Fonte: Acervo pessoal do projeto, 2021.

Os furos para instalação da porta, do potenciômetro, LED's e passagem da fiação de conexão do display foram realizados utilizando furadeira Bosh® e a micro retifica já mencionada. Para iniciar os furos sem danificar as madeiras devido sua fragilidade foi utilizada broca para madeira 2mm e, posteriormente, brocas de 4mm, 6mm e 8mm de acordo com a necessidade. O acabamento externo foi realizado utilizando verniz acrílico fosco Acrilex® com o auxílio de um pincel Pintacora TIGRE® modelo 815, tamanho 8.

FIGURA 32 – PINTURA DA CAIXA COM VERNIZ.



Fonte: Acervo pessoal do projeto, 2021.

Com o armário semifinalizado, foi realizado um teste das ligações e conexões eletrônicas, além da instalação do servo motor e ajuste refinado do equipamento com a porta.

FIGURA 33 – LIGAÇÕES E CONEXÕES DO CIRCUITO.



Fonte: Arquivo pessoal do projeto, 2021.

Por fim, após as etapas anteriores, foi finalizada a colagem dos palitos de madeira restantes, onde foram repetidos os processos de calafetação com a massa F.12, lixamento, pintura com verniz para que pudesse receber em definitivo os microcontroladores e todas as conexões eletrônicas necessárias para o funcionamento.

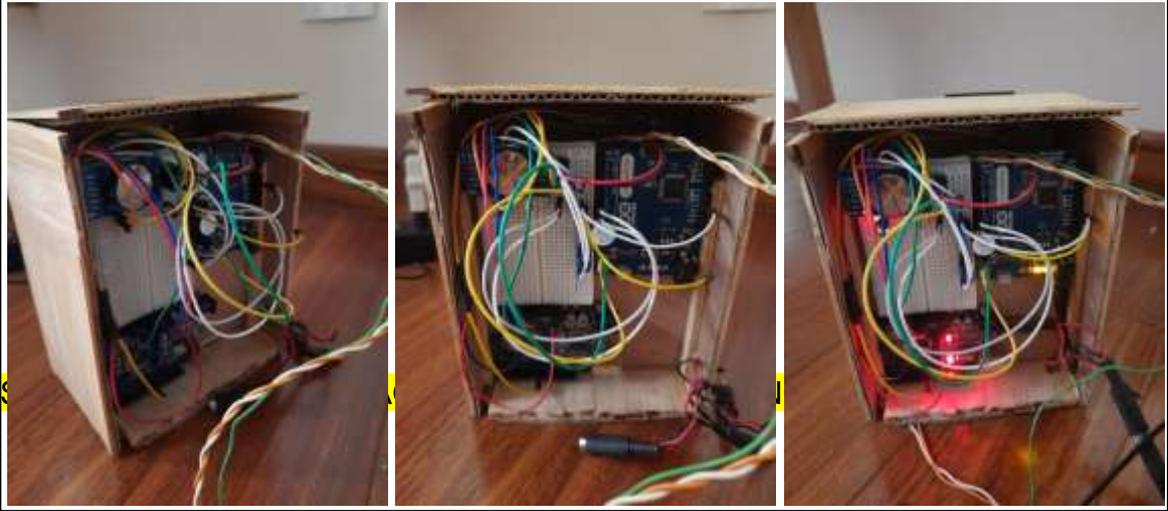
FIGURA 34 – PROTÓTIPO EM FUNCIONAMENTO NA VISÃO FRONTAL.



Fonte: Arquivo pessoal do projeto, 2021.

Na parte posterior, optou-se em não fechar o protótipo com um tampão, pois a ideia é justamente mostrar as conexões do circuito eletrônico do equipamento.

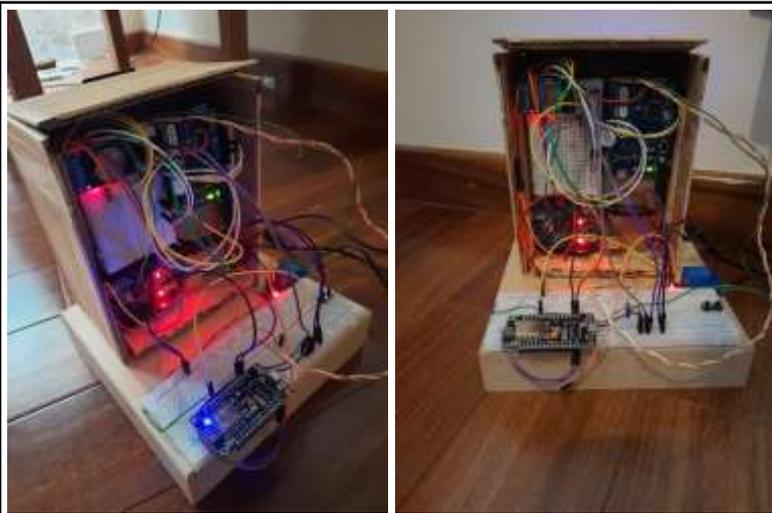
FIGURA 35 – DISPOSIÇÃO DO CIRCUITO ELETRÔNICO NA PARTE POSTERIOR DO PROTÓTIPO.



Fonte: Arquivo pessoal do projeto, 2021.

Para uma melhor disposição, o protótipo foi inserido em uma bancada (Figura 37). Nela também foi inserido a protoboard com o NodeMCU e o relé, permitindo-os ficarem externos e que a banca avaliadora do trabalho final, possa visualizar com maior facilidade (Figura 36).

FIGURA 36 – ESP8266 CONECTADO NA PROTOBOARD COLADA NA PLATAFORMA JUNTAMENTE COM O RELÉ.



Fonte: Arquivo pessoal do projeto, 2021.

FIGURA 37 – PROTÓTIPO EM CIMA DA PLATAFORMA DE MDF.



Fonte: Arquivo pessoal do projeto, 2021.

Após o protótipo montado e em cima da plataforma, foi realizado um teste final para conectar o ESP8266 no roteador do celular de um dos integrantes dos grupos para analisar a possibilidade de uma demonstração real na apresentação final para a banca, além de testar o acionamento das portas D3 e D4 do NodeMCU através do acionamento do fim de curso e do relé.

FIGURA 38 – ÚLTIMO TESTE DO EQUIPAMENTO ANTES DA APRESENTAÇÃO FINAL.



Fonte: Arquivo pessoal do projeto, 2021.

Por fim, foi realizado um último ajuste na programação em linguagem C++ no Arduino Leonardo, onde, a porta 6 foi adicionada no modo OUTPUT para acionar o relé nos horários pré-estabelecidos para acionamento do alarme (Figura 39). Após a linha 48, toda a programação manteve-se idêntica as das Figuras 12.A e 12.B.

FIGURA 39 – PROGRAMAÇÃO FINAL DO DESPERTADOR PARA ACIONAMENTO DO LED, RELÉ E BUZZER.

```

1  #include <LiquidCrystal_I2C.h>    // Inclui a Biblioteca
2  #include <RTClib.h>              // Inclui a Biblioteca
3
4  RTC_DS1307 rtc;
5  LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicia o display 16x2 no endereço 0x27
6
7  #define buzzer 11 // Define o buzzer como pino 11
8  #define led1 9 // Define o led da porta 1 como pino 9
9  #define led2 8 // Define o led da porta 2 como pino 8
10 #define rele 6 // Define o rele como pino 6
11
12 int Hor; // Inicia a variável Hor
13 int Min; // Inicia a variável Min
14 int Seg; // Inicia a variável Seg
15
16 void setup() {
17   pinMode(buzzer, OUTPUT);
18   pinMode(led1, OUTPUT);
19   pinMode(led2, OUTPUT);
20   pinMode(rele, OUTPUT);
21   rtc.begin(); // Inicia o módulo RTC
22   lcd.init(); // Inicia o Display
23   lcd.backlight(); // configura a porta serial
24 }
25
26 void loop() {
27   DateTime now = rtc.now();
28   Hor = rtc.now().hour(); // Chama o horário de Hor
29   Min = rtc.now().minute(); // Chama o minuto de Min
30   Seg = rtc.now().second(); // Chama os segundos de Seg
31
32   // Define o horário do despertador e executa o som do buzzer
33   if ( Hor == 20 && Min == 04) {
34     digitalWrite(led1, HIGH);
35     digitalWrite(buzzer, HIGH);
36     digitalWrite(rele, HIGH);
37     delay(1000);
38     digitalWrite(buzzer, LOW);
39     digitalWrite(led1, LOW);
40     digitalWrite(rele, LOW);
41     delay(1000);
42   }

```

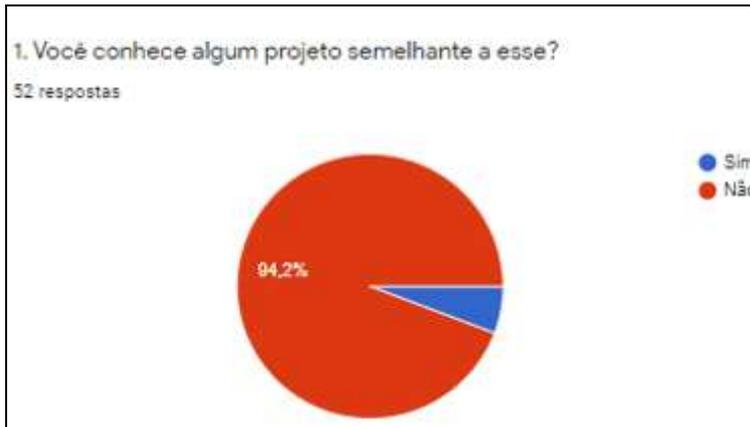
Fonte: Arquivo pessoal do projeto, 2021.

5.RESULTADOS E DISCUSSÃO

5.1 Pesquisa de Campo

Foram obtidas 52 respostas, onde 94,2% responderam não conhecer algum projeto semelhante.

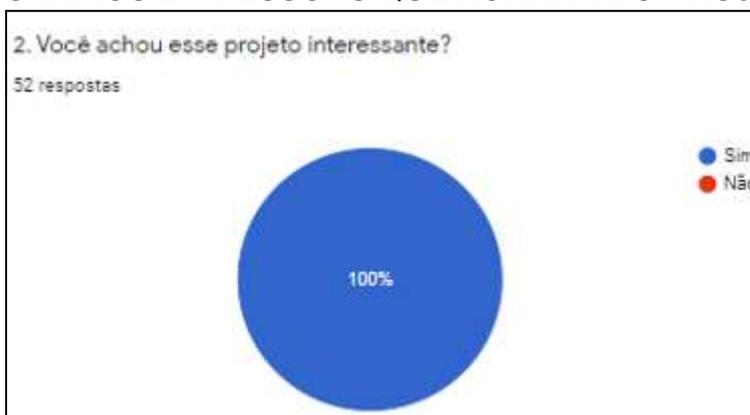
GRÁFICO 1 – PESSOAS QUE CONHECEM ALGUM PROJETO SIMILAR.



Fonte: Acervo pessoal do projeto, 2021.

Todos os avaliados acharam o projeto interessante.

GRÁFICO 2 – PESSOAS QUE ACHARAM O PROJETO INTERESSANTE.



Fonte: Acervo pessoal do projeto, 2021.

Dentre os avaliados, 92,3% responderam que comprariam o produto para si ou algum familiar.

GRÁFICO 3 – PESSOAS QUE TEM INTERESSE NA COMPRA DO PRODUTO.



Fonte: Acervo pessoal do projeto, 2021.

Quanto a pergunta do quanto a pessoa estaria disposta a pagar pelo produto, houve uma maior heterogeneidade, pois 34,6% responderam que pagariam entre R\$ 1,00 e R\$ 100,00; 34,6% pagariam entre R\$ 101,00 e R\$ 200,00; 19,2% estariam dispostos a pagar entre 201,00 e R\$ 300,00, enquanto apenas 9,6% responderam que pagariam entre R\$ 300,00 ou mais. Apenas uma pessoa, respectivamente 1,92%, respondeu que não estaria disposto a pagar pelo produto (R\$ 0,00).

GRÁFICO 4 – PREÇO QUE AS PESSOAS ESTÃO DISPOSTAS A PAGAR PELO PRODUTO.



Fonte: Acervo pessoal do projeto, 2021.

Referente a pergunta aberta (dissertativa) não obrigatória sobre sugestões para o projeto, 23,07% dos avaliados responderam, totalizando 12 respostas. Dessas, dois responderam não terem sugestões, assim ficando 10 respostas válidas.

TABELA 4 – RESPOSTAS ABERTAS OBTIDAS ATRAVÉS DA PESQUISA DE CAMPO.

Nº	Respostas
1.	Não.
2.	Uma gaveta para cada remédio.
3.	R\$ 380,00
4.	Projeto muito interessante, boa sorte!
5.	Colocaria um relógio para aparência estética.
6.	O alarme audiovisual é mais interessante do que propriamente a gaveta abrir.
7.	Acho que deveria ter alguma outra forma também além da internet, porque muitos idosos não tem internet. Minha vó, por exemplo, não tem.
8.	Precificar ou detalhar a complexidade do produto a fim de informar o consumidor/avaliador do valor do produto.
9.	Ele auxilia no lembrete, mas não dá certeza se a pessoa realmente tomou. Então não tenho certeza se eu estaria disposto a investir.
10.	Nenhuma.
11.	Timer dos horários para tomar os remédios.
12.	Talvez realizar algum tipo de programação ou separação para diferentes tipos de remédios a serem tomados num mesmo horário, descrevendo cada um deles, ao invés de ser um aviso geral do tipo “tomar remédios da manhã” seja interessante, contado quem em algumas situações o responsável por dar o medicamento não saiba exatamente quais sejam esses.

Fonte: Acervo pessoal do projeto, 2021.

Das respostas obtidas, as respostas número 2, 5, 6, 11 e 12 foram acatadas para realização do protótipo. A resposta 8 também, sendo essa respondida após a finalização do protótipo.

Quanto a resposta 7, de fato é um problema, porém, que pode ser solucionado utilizando um chip de operadora no microcontrolador do Arduino para enviar as notificações via *Wirelles* ou *Short Message Service* – Serviço de Mensagens Curtas (SMS). Segundo a Agência Brasil (2021), atualmente, 152 milhões de pessoas tem acesso à internet no Brasil (OLIVEIRA, 2021), um aumento de 18 milhões em relação ao ano anterior (FARIAS, 2020), o que demonstra crescente no acesso a internet no Brasil, conseqüentemente, em um futuro não muito distante, grande maioria da população brasileira terá acesso a *internet*, tornando viável ter o armário automatizado de medicamentos em sua residência em caso de necessidade.

5.2 Protótipo

Foi possível desenvolver um armário automatizado de medicamentos que permite programar previamente os horários em que os alarmes audiovisuais possam ser acionados. Podendo ser adicionado qualquer data e hora futura dentro dos próximos 100 anos (a contar da data em que foi gravado o calendário no módulo RTC).

Como sugerido na pesquisa de campo, foi possível adicionar um display LCD para visualizada da data e hora em tempo real no equipamento.

O ESP8266 foi capaz de receber os sinais digitais da porta do armário e da porta do Arduino Leonardo e enviá-la para uma página HTML, permitindo a visualização do estado atual do equipamento (aberto/fechado e ligado/desligado) na web através do acesso do IP gerado na conexão entre o NodeMCU e o roteador/provedor de internet.

Devido conflitos de programação, não foi possível utilizar o Arduino Leonardo para também acionar o servo motor no horário pré-estabelecido, sendo necessário acrescentar o Arduino Uno exclusivamente para essa função.

6. CONSIDERAÇÕES FINAIS

Tinha-se como objetivo principal do projeto, desenvolver um armário automatizado de medicamentos de baixo custo acionado por efeito audiovisual em datas e horários pré-estabelecidos, lembrando o usuário de ingerir os medicamentos e, em tempo real, permitir que um responsável acompanhe através de notificações no *smartphone* ou computador.

Pode-se dizer que este objetivo foi alcançado, pois foi possível programar o horário no equipamento e o mesmo ser acionado no horário desejado, além de enviar as informações para uma página HTML. No entanto, o horário só pode ser programado através da IDE sendo necessário a utilização de um computador e conexão USB com o Arduino, não sendo possível definir o horário do *smartphone*.

Por tratar-se de um protótipo, só é possível acessar a página HTML gerada, quando os aparelhos celulares ou notebook estiverem conectados no mesmo provedor de internet, mostrando limitação do projeto quanto ao objetivo específico de uma pessoa acompanhar em tempo real do trabalho ou escola o status do armário, por exemplo. Mas, essa situação pode ser contornada através de programações em linguagem C++ específicas para que o equipamento tenha um IP fixo e o roteador que

está provendo a internet para o NodeMDU esteja configurado para aceitar acesso externo.

Quanto a redução do contato presencial, o protótipo demonstrou ter atingido este objetivo, pois uma vez programado os horários de acionamento, os mesmos ficam fixo independentemente de falta ou não de energia elétrica, graças ao módulo RTC com uma bateria externa de 3V.

Por fim, é possível concluir que o protótipo construído, atingiu grande parcela dos objetivos propostos. Além disso ele proporcionou uma vasta gama de aprendizado aos integrantes da equipe em relação a conhecimento em microcontroladores, acessórios e linguagem C++.

Sugere-se que outras turmas possam dar sequência nessa ideia e realizar aperfeiçoamentos nas programações já elaboradas, para que de fato, o usuário possa ter maior autonomia e um terceiro possa, à distância, receber informações do status da ferramenta e programar os despertadores à distância, sem a necessidade de conexão USB com o equipamento.

REFERÊNCIAS

BEZERRA, Anselmo César Vasconcelos, *et al.* "Fatores associados ao comportamento da população durante o isolamento social na pandemia de COVID-19." **Ciência & Saúde Coletiva**, p. 2411-2421., 2020.

CURVELO, Andre. Apresentando o módulo ESP8266. 2015. **Embarcados**. Disponível em <https://www.embarcados.com.br/modulo-esp8266/>. Acesso em: 23 de nov. de 2021.

DA SILVA, Vania Candida e JOAQUIM Aloísio. "Determinantes de condições dos idosos nas Regiões Sudeste e Nordeste do Brasil, 1970 e 2000." **Anais ABESP**, p. 1-21., 2016.

DE OLIVEIRA, Pedro Ivo. Brasil tem 152 milhões de pessoas com acesso a internet. **Agência Brasil**. Disponível em <https://agenciabrasil.ebc.com.br/geral/noticia/2021-08/brasil-tem-152-milhoes-de-pessoas-com-acesso-internet>. Acesso em: 25 de out. de 2021.

EUREKA. **Eureka**, c2021. Smart Home: aumenta o consumo de itens de automação residencial durante a pandemia. Disponível em <https://www.eurekaht.com.br/smart-home-aumenta-o-consumo-durante-pandemia/>. Acesso em: 28 de jul. de 2021.

FARIAS, Liliane. Brasil tem 134 milhões de pessoas de usuários de internet aponta pesquisa. **Agência Brasil**. Disponível em <https://agenciabrasil.ebc.com.br/geral/noticia/2020-05/brasil-tem-134-milhoes-de-usuarios-de-internet-aponta-pesquisa>. Acesso em: 25 de out. de 2021.

HELERBROCK, Rafael. "O que é LED?". **Brasil Escola**. Disponível em: <https://brasilecola.uol.com.br/o-que-e/fisica/o-que-e-led.htm>. Acesso em 23 de novembro de 2021.

IBGE. Idosos indicam caminhos para uma melhor idade: As decisões diante do envelhecimento populacional **Revista Retrato**, 19 de mar. de 2019. Disponível em <https://censo2022.ibge.gov.br/2012-agencia-de-noticias/noticias/24036-idosos-indicam-caminhos-para-uma-melhor-idade.html>. Acesso em: 28 de jul. de 2021.

JORNAL NACIONAL. Daqui a 20 anos, Brasil vai ter mais idosos do que crianças, afirma IBGE. **G1, Jornal Nacional**, Rio de Janeiro, 25 de jul. de 2018. Disponível em <http://g1.globo.com/jornal-nacional/noticia/2018/07/daqui-20-anos-brasil-vai-ter-mais-idosos-do-que-criancas-afirma-ibge.html>. Acesso em: 28 de jul. de 2021.

LINS, Vitor e MOURA Waldson. "Domótica: Automação Residencial." **UNIBRATEC**, 2010. Disponível em <https://silo.tips/download/domotica-automacao-residencial>. Acesso em: 28 de jul. de 2021.

MATTEDE, Henrique. O que é um LED?. **Mundo da Elétrica**. c2021. Disponível em <https://www.mundodaeletrica.com.br/o-que-e-um-led/>. Acesso em: 23 de nov. de 2021.

OPAS. Folha informativa sobre COVID-19. **Organização Panamericana De Saúde**, c2021. Disponível em <https://www.paho.org/pt/covid19>. Acesso em: 28 de jul. de 2021.

PACIEVITCH, Yuri. Linguagem de programação C++. **Info Escola**. Disponível em <https://www.infoescola.com/informatica/cpp/>. Acesso em: 27 de out. de 2021.

PEIXOTO, Sérgio Viana, *et al.* "Comportamentos em saúde e adoção de medidas de proteção individual durante a pandemia do novo coronavírus: iniciativa ELSI-COVID-19." **Cadernos de Saúde Pública**, 2020.

PHULMANN, Henrique Frank Werner. Módulo de Display LCD. 2015. **Embarcados**. Disponível em <https://www.embarcados.com.br/modulo-de-display-lcd/>. Acesso em: 23 de nov. de 2021.

ROCHA, Igor Dantas. Domótica via dispositivos móveis utilizando a plataforma Arduino. 2016. 64f. (Trabalho de Conclusão de Curso - Monografia), Curso de Bacharelado em Engenharia Elétrica, Centro de Engenharia Elétrica e Informática, **Universidade Federal de Campina Grande** – Paraíba - Brasil, 2016. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/18647>. Acesso em: 07 de set. de 2021.

SILVA, Ademir Cavalcante e. Projeto de um sistema de domótica utilizando Arduino. 2014. 40f. (Trabalho de Conclusão de Curso - Monografia), Curso de Bacharelado em Engenharia Elétrica, Centro de Engenharia Elétrica e Informática, **Universidade Federal de Campina Grande** – Paraíba - Brasil, 2014. Disponível em <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/18341>. Acesso em: 07 de set. de 2021.

TAVARES, Noemia Urruth Leão, *et al.* "Uso de medicamentos para tratamento de doenças crônicas não transmissíveis no Brasil: resultados da Pesquisa Nacional de Saúde, 2013." **Epidemiologia e Serviços de Saúde**, p. 315-323; 2015.

THOMSEN, Adilson. O que é Arduino?. 2014. **Filipe Flop**. Disponível em <https://www.filipeflop.com/blog/o-que-e-arduino/>. Acesso em: 23 de nov. de 2021.

VB. Pandemia aumenta procura por automação residencial no DF. **Jornal ao Vivo Brasília**, Brasília, 06 set. de 2020 Disponível em: <https://www.aovivodebrasil.com.br/pandemia-aumenta-procura-por-automacao-residencial-no-df/>. Acesso em: 07 de set. de 2021.

PERA, Bruno. Iniciando IoT com ESP8266. Univap. Disponível em <https://www1.univap.br/bruno.pera/uploads/ELETRONICA/MICROPROCESSADORE S/IOT%20com%20ESP8266.pdf>. Acesso em: 23 de nov. de 2021.

APENDICE 1 – PERGUNTAS DA PESQUISA DE CAMPO



Seção 1 de 2

GAVETEIRO AUTOMATIZADO DE MEDICAMENTOS

O Trabalho de Conclusão de Curso (TCC) denominado "Gaveteiro Automatizado de Medicamentos" trata-se de um sistema de automação residencial (domótica) que permite pré-estabelecer e programar os horários de abertura dos gaveteiros de medicamentos seguido por aviso audiovisual de maneira automatizada, lembrando a pessoa idosa de tomar o(s) medicamento(s).

Simultaneamente, ao abrir a gaveta, é disparando um recado via wireless (internet) para os smartphones dos filhos, netos e/ou cuidadores informando o momento em que a gaveta foi aberta (modo automático) e quando foi fechada (modo manual). Assim, permitindo ao familiar monitorar de maneira remota (do trabalho, escola, reunião, viagem etc.) os horários de ingestão dos medicamentos pela pessoa idosa, além de propiciar maior autonomia desse público (dentro da sua residência), pois não será necessário alguém ficar lembrando-a de forma direta a todo instante que está na hora de tomar o respectivo remédio.

Os horários programados para abertura das gavetas podem ser realizados a distância. Se por ventura, o sistema autônomo programado falhar, é possível abrir a(s) gaveta(s) manualmente de modo físico ou remoto.

Discentes: Diego Silva Mota; Marcos José Frutuoso; Victor Prado Ribeiro.
Orientador: Prof. Eng. Carlos Alberto Serpeloni Barros.
Curso: Técnico em Eletroeletrônica.
Escola: Etec Trajano Camargo, Limeira-SP.
Contato: diego.mota5@etec.sp.gov.br

☰

Deseja responder ao questionário de 5 perguntas? O tempo estimado é de 2 minutos. *

Sim

Não

Após a seção 1 Continuar para a próxima seção ▼

Seção 2 de 2

Perguntas de 1 a 5



Agradecemos por responder ao questionário. Sua participação é muito importante para o desenvolvimento do nosso projeto de TCC.

1. Você conhece algum projeto semelhante a esse? *

- Sim
- Não

2. Você achou esse projeto interessante? *

- Sim
- Não

3. Você compraria esse produto? (Para você ou algum familiar) *

- Sim
- Não

4. Quanto você estaria disposto a pagar por esse produto? *

- R\$ 0,00
- R\$ 1,00 a R\$ 100,00
- R\$ 101,00 a R\$ 200,00
- R\$ 201,00 a R\$ 300,00
- R\$ 300,00 ou mais

5. Você tem alguma sugestão? Se sim, qual? (Resposta opcional)

Texto de resposta longa