

Estudo de Caso: sistema para auxílio de pessoas idosas

Leonardo Adolfo Pimenta Junqueira
Graduando em Engenharia de Software – Uni-FACEF
adolfoleo625@gmail.com

Carlos Eduardo de França Roland
Mestre em Desenvolvimento Regional – Uni-FACEF
roland@facef.br

Resumo

Esse artigo apresenta o projeto de aplicação das Tecnologias Digitais de Informação e Comunicação no auxílio de pessoas de idade avançada a terem uma vida mais saudável e segura. Ao observarmos parentes de mais idade foi notado que as famílias tinham certas inseguranças sobre deixar idosos sozinhos, com medo que eles possam sofrer algum tipo de acidente dentro de sua casa e não tenha ninguém para socorrê-los. Outro ponto observado foi a ausência de controle sobre a medicação dos idosos, trazendo riscos para a saúde deles. Com base nessas evidências foi elaborado o projeto de conclusão de curso em Engenharia de Software com o objetivo de associar os recursos da plataforma Arduino para automação de processos de controle do uso de medicamentos e da tecnologia de Internet das Coisas para monitoramento de idosos. A plataforma Arduino disponibiliza *hardware* que opera com base em microcontroladores apropriados para prototipagem de sistemas de automação e controle, como provas de conceito. É possível desenvolver *software* para o dispositivo, que monitora os horários de uso dos medicamentos e gere alertas luminosos e sonoros para que os pacientes se mediquem; e, com adição de componente de comunicação Wifi, criar um dispositivo móvel para sinalização de necessidades e urgências.

Palavras-chave: Arduino; ESP8266; medicação de idosos; monitoramento de idosos; NodeMCU.

Abstract

This article presents the project to apply Digital Information and Communication Technologies to help elderly people lead healthier and safer lives. When we observed older relatives, it was noticed that families had certain insecurities about leaving elderly people alone, afraid that they might suffer some type of accident inside their home and there would be no one to help them. Another point observed was the lack of control over the elderly's medication, posing risks to their health. Based on this evidence, the final course project in Software Engineering was prepared with the objective of combining the resources of the Arduino platform for automating processes for controlling the use of medicines and Internet of Things technology for monitoring the elderly. The Arduino platform provides hardware that operates based on microcontrollers suitable for prototyping automation and control systems, as proofs of concept. It is possible to develop software for the device, which monitors medication usage times and generates light and round alerts for patients to take medication; and,

with the addition of a Wifi communication component, create a mobile device to signal needs and emergencies.

Keywords: *Arduino; ESP8266; medication for the elderly; monitoring the elderly; NodeMCU.*

1 Introdução

Nos dias atuais, o aumento da expectativa de vida e o envelhecimento da população trazem novos desafios para as famílias e profissionais da saúde. Entre esses desafios, destaca-se a preocupação com a segurança e o bem-estar de pessoas idosas que, muitas vezes, residem sozinhas. Situações de risco, como quedas ou esquecimentos em relação à medicação, são fontes de apreensão para familiares e cuidadores. Diante desse cenário, a tecnologia pode oferecer soluções promissoras para garantir mais segurança e autonomia aos idosos?

Este artigo explora como a combinação da plataforma Arduino com módulo de comunicação sem fio pode criar sistemas inteligentes para apoio e monitoramento de idosos. A criação dessa classe de sistema é importante por diversos motivos: em primeiro lugar, porque oferece um mecanismo eficiente de lembrete para realização da medicação prescrita, evitando problemas decorrentes de esquecimentos; em segundo, porque pode reduzir significativamente o tempo de resposta em casos de emergências proporcionando mais tranquilidade para as famílias. Por fim, esse tipo de tecnologia é útil porque pode ser implementado de forma discreta, sem interferir na privacidade do idoso, mas garantindo um monitoramento eficaz e contínuo, sem a necessidade de intervenção constante de terceiros.

Com o uso de sensores, alertas sonoros e algoritmos de controle de atuadores, o sistema proposto visa melhorar a qualidade de vida dos idosos, proporcionando-lhes mais segurança em suas rotinas diárias e, ao mesmo tempo, aliviando as preocupações dos seus familiares e cuidadores.

2 Bases teóricas do projeto

Nessa seção estão descritas informações que fundamentam a criação dos protótipos que melhorem a qualidade de vida e auxiliem idosos. Assim, estão comentados tópicos sobre a saúde, a tecnologia e qualidade de vida; Arduino; comunicação sem fio com ESP8266.

2.1 Saúde e Tecnologia

A tecnologia e a saúde estão intrinsecamente interligadas, com o progresso tecnológico, impulsionando continuamente a evolução da medicina e dos cuidados à saúde. Essa relação se manifesta em diversas áreas, desde o desenvolvimento de novos medicamentos e equipamentos médicos até a aplicação de inteligência artificial e telemedicina. “O setor saúde, fortemente influenciado pelo paradigma da ciência positiva, tem sido sensível à incorporação tecnológicas do tipo material, para fins terapêuticos, diagnósticos e de manutenção da vida” (Lorenzetti *et al*, 2012).

A relação entre tecnologia e saúde é dinâmica e promissora. O desenvolvimento e a aplicação de novas tecnologias continuarão a revolucionar a medicina e a melhorar a saúde da população. A busca por soluções inovadoras e eficazes, aliada à superação dos desafios existentes, garantirá um futuro mais saudável para todos.

2.2 Qualidade de Vida

A qualidade de vida na terceira idade é um tema complexo e multifacetado, influenciado por diversos fatores, como saúde física e mental, bem-estar social, autonomia e independência. Segundo Minayo (2012), “existe uma pequena parcela de idosos sem rendimentos próprios e que não é capaz de atender às suas necessidades básicas, além de vivenciar sérios problemas de saúde e dependências físicas e mentais”.

A saúde é um componente fundamental da qualidade de vida, especialmente na terceira idade. Doenças crônicas, como diabetes, hipertensão e doenças cardiovasculares, são comuns entre os idosos e podem afetar significativamente sua capacidade de realizar atividades cotidianas e sua qualidade de vida. O envelhecimento pode levar à fragilidade física e mental, à perda de autonomia e à dependência de outras pessoas. O isolamento social, a falta de acesso a serviços de saúde e a pobreza também são desafios que podem afetar a qualidade de vida dos idosos (Albuquerque, 2003).

Considerando alguns dos desafios citados, a tecnologia tem um papel importante na promoção da qualidade de vida na terceira idade. Telemedicina, robótica, sistemas de monitoramento remoto e dispositivos de assistência pessoal são exemplos de tecnologias que podem ajudar os idosos a viver de forma mais independente e segura.

3 Automação com Arduino

A automação visa executar tarefas repetitivas e complexas em diversos setores, como industrial, residencial e comercial através de equipamentos eletromecânicos, eletrônicos, pneumáticos, ou hidráulicos. Ela utiliza sistemas de controle para monitorar e manipular processos, substituindo o trabalho manual e aumentando a eficiência, a produtividade e a segurança.

O Arduino é uma plataforma de prototipagem eletrônica de código aberto, composta por *hardware* e *software*. O *hardware* é baseado em microcontroladores Atmel AVR, enquanto o *software* é uma linguagem de programação simplificada, derivada do C/C++ (McRoberts, 2018).

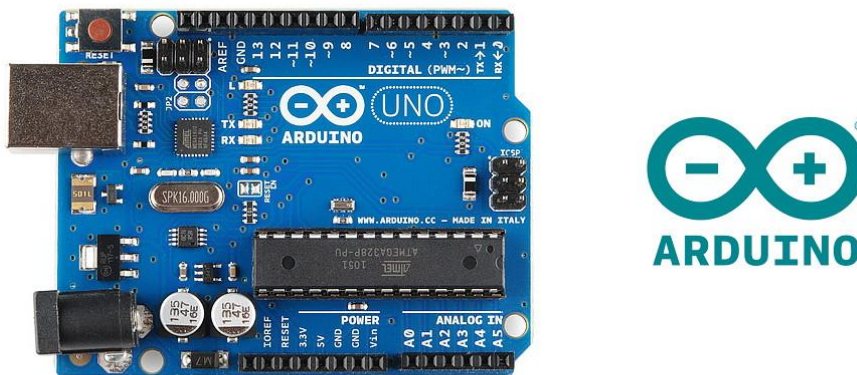
3.1 Hardware Arduino

A plataforma Arduino desbrava o mundo da automação, abrindo um universo de possibilidades. Seja na automação residencial, na vanguarda da indústria, no ensino ou na criação de robôs futuristas, o Arduino é a ferramenta ideal para transformar uma visão em realidade.

A placa Arduino (Figura 1) é um verdadeiro universo de possibilidades criativas para os entusiastas da eletrônica. No seu coração, pulsa um poderoso microcontrolador, o maestro que dá vida aos projetos.

Por meio de um conjunto versátil de entradas e saídas digitais e analógicas, a placa se conecta a uma infinidade de sensores, atuadores e outros dispositivos, abrindo um leque de oportunidades para dar vida às ideias. Com a comunicação serial, a placa Arduino se torna um cidadão global do mundo digital, podendo se conectar a computadores, *smartphones* e outros dispositivos, expandindo ainda mais suas capacidades. A memória interna armazena o programa e dados, garantindo que os projetos funcionem com precisão e confiabilidade e, para alimentar essa maravilha tecnológica, a placa conta com um sistema de alimentação robusto e eficiente (Arduino e Cia, 2024).

Figura 1 - Arduino Uno



Fonte: Arduino e Cia (2024)

Em suma, a placa Arduino é uma plataforma completa e poderosa que coloca o projetista no comando da sua criatividade. Com seus diversos recursos e flexibilidade, pode-se dar vida a projetos incríveis, desde simples experimentos até sistemas complexos e inovadores.

3.2 Aplicações da Automação com Arduino

O Arduino é uma plataforma versátil que revoluciona a interação com o ambiente ao nosso redor. Na Domótica, ele é o cérebro de uma casa inteligente, integrando controle de iluminação, eletrodomésticos e sistemas de segurança, tudo para oferecer um ambiente que se ajusta automaticamente às preferências e necessidades dos moradores. No campo da robótica, o Arduino é o coração de protótipos de robôs, desde os que realizam tarefas domésticas até os projetados para

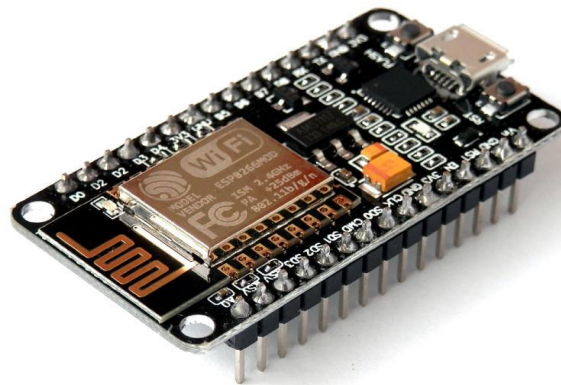
missões espaciais, todos operando com autonomia ou comandos remotos. Na indústria, ele se destaca na prototipagem de automação, otimizando processos e elevando a eficiência e segurança das operações. O Arduino é uma ferramenta inestimável, agilizando a criação de protótipos eletrônicos e transformando ideias em realidade tangível. E na educação, ele se torna um recurso didático poderoso, incentivando o aprendizado prático de eletrônica, programação e automação, preparando os alunos para os desafios tecnológicos do futuro (Makerhero, 2023).

4 ESP8266 e Comunicação Sem Fio

O ESP8266 (Figura 2) é um microcontrolador altamente popular em projetos de Internet das Coisas (IoT) devido à sua capacidade de se conectar à internet via Wi-Fi. Este módulo integra um protocolo de comunicação em rede TCP/IP, permitindo que dispositivos se conectem a redes Wi-Fi padrão 802.11 b/g/n1 (Embarcados, s.d. online).

Da mesma fonte, o ESP8266 é baseado em um processador de 32 bits e memória flash integrada, podendo executar tarefas complexas e transmitir dados para servidores remotos ou outros dispositivos na rede, tornando-o ideal para aplicações que exigem conectividade constante com a internet.

Figura 2 - Esp8266



Fonte: Magalhães (2024)

Uma das principais aplicações do ESP8266 é em sistemas de automação residencial, onde ele pode ser utilizado para controlar dispositivos remotamente através de uma conexão Wi-Fi. Por exemplo, o ESP8266 pode ser programado para monitorar sensores de temperatura e umidade e enviar esses dados para um servidor na nuvem. A partir daí, os dados podem ser acessados por um aplicativo móvel, permitindo que os usuários monitorem e controlem o ambiente de suas casas em tempo real. Além disso, o ESP8266 pode receber comandos via internet para acionar dispositivos como lâmpadas, termostatos e fechaduras eletrônicas (Embarcados, 2024).

Outra aplicação importante do ESP8266 é em projetos de monitoramento e coleta de dados. Em ambientes industriais, por exemplo, o ESP8266

pode ser utilizado para monitorar o funcionamento de máquinas e enviar alertas em caso de falhas ou necessidade de manutenção. A capacidade de se conectar à internet permite que os dados coletados sejam armazenados em servidores remotos, facilitando a análise e a tomada de decisões baseadas em dados em tempo real. Essa conectividade robusta e flexível faz do ESP8266 uma ferramenta essencial para desenvolvedores que buscam criar soluções inovadoras e eficientes em IoT (Embarcados, 2024).

4.1 Protocolo MQTT de Plataformas em Nuvem

O protocolo MQTT (*Message Queuing Telemetry Transport*) é amplamente utilizado em plataformas de nuvem para comunicação eficiente entre dispositivos de Internet das Coisas (IoT). Desenvolvido inicialmente em 1999 para monitoramento de oleodutos via satélite, o MQTT é um protocolo leve e de fácil implementação, ideal para redes com largura de banda limitada e alta latência. Ele opera sobre o protocolo TCP/IP e utiliza um modelo de publicação e subscrição, onde dispositivos (clientes) podem publicar mensagens em tópicos específicos e outros dispositivos podem se inscrever nesses tópicos para receber as mensagens (Banks; Gupta, 2014).

Uma das principais vantagens do MQTT em plataformas de nuvem é sua eficiência na transmissão de dados. O protocolo minimiza o uso de largura de banda e recursos do dispositivo, o que é fundamental para dispositivos IoT que frequentemente operam com restrições de energia e conectividade (Banks; Gupta, 2014). Além disso, o MQTT oferece três níveis de qualidade de serviço (QoS) para garantir a entrega confiável das mensagens: no máximo uma vez (QoS 0), pelo menos uma vez (QoS 1) e exatamente uma vez (QoS 2). Essa flexibilidade permite que desenvolvedores ajustem o protocolo conforme as necessidades específicas de suas aplicações. O ESP8266, um microcontrolador amplamente utilizado em projetos de IoT, é frequentemente combinado com o protocolo MQTT para comunicação com a internet (Banks; Gupta, 2014).

O ESP8266 pode se conectar a redes Wi-Fi e utilizar o MQTT para enviar e receber dados de servidores na nuvem. Por exemplo, em um projeto de automação residencial, o ESP8266 pode monitorar sensores de temperatura e umidade e publicar esses dados em um tópico MQTT. Um servidor na nuvem, como o AWS IoT ou o Azure IoT, pode então processar esses dados e enviar comandos de volta ao ESP8266 para controlar dispositivos como aquecedores ou sistemas de ventilação. Essa integração permite a criação de sistemas IoT eficientes e escaláveis, aproveitando a robustez do MQTT e a versatilidade do ESP8266 (Hunkeler; Triminton; Stanford-Clark, 2008).

4.2 Framework Wiring, IDE Arduino e Linguagem C++

O *framework* Wiring é uma plataforma de prototipagem eletrônica de *hardware* livre que inclui uma linguagem de programação, um ambiente de desenvolvimento integrado (IDE) e um microcontrolador. Desenvolvido por Hernando Barragán em 2003, o Wiring foi projetado para facilitar a criação de *software* que controla dispositivos conectados a uma placa eletrônica, permitindo a criação de diversos dispositivos interativos (Wiringproject, 2024).

A linguagem Wiring é uma camada fina sobre C++, simplificando a escrita de programas embarcados.

A IDE Arduino é uma ferramenta essencial para programar microcontroladores da plataforma Arduino. Baseada na linguagem Wiring, a IDE Arduino permite escrever, compilar e carregar programas para as placas Arduino de forma intuitiva e eficiente. A IDE é compatível com Windows, Mac OS X e Linux, e inclui uma vasta biblioteca de exemplos e tutoriais que ajudam iniciantes e profissionais a desenvolverem seus projetos de forma rápida e eficaz.

A linguagem C++ é amplamente utilizada no desenvolvimento de programas para microcontroladores devido à sua eficiência e flexibilidade. No contexto do Arduino, o C++ é a base para a linguagem Wiring, que adiciona funções específicas para facilitar a interação com o hardware. A combinação de C++ com a IDE Arduino e o *framework* Wiring permite a criação de projetos complexos de forma simplificada, tornando a prototipagem eletrônica acessível a uma ampla gama de usuários, desde estudantes até profissionais da área de tecnologia.

5 Metodologia de Desenvolvimento

Para o desenvolvimento e progresso do projeto, foi adotada a IDE Arduino, garantindo maior eficiência e flexibilidade no projeto. No lado do *hardware*, o Arduino UNO foi escolhido como a base de controle dos sensores e atuadores. Para a implementação da lógica de controle, foi utilizada C++, a linguagem de programação padrão da plataforma Arduino. C++ permite o controle direto dos pinos, possibilitando o gerenciamento eficiente dos componentes como sensores de movimento, sensores de queda e módulos sonoros. O código escrito em C++ garante precisão no controle de tempo para alertas e na leitura dos sinais dos sensores, resultando em um sistema confiável para o monitoramento dos idosos.

Além disso, o uso do C++ facilita a criação de rotinas eficientes de comunicação entre os diferentes sensores do Arduino. Sua compatibilidade com bibliotecas específicas da plataforma torna o desenvolvimento mais simples e direto, permitindo maior controle sobre as entradas e saídas do sistema. A linguagem também oferece baixo consumo de recursos, o que é essencial em dispositivos embarcados, garantindo que o sistema funcione continuamente e sem interrupções, mesmo em ambientes de operação constante (Arduino e Cia, 2024).

6 *Dispenser* de Remédios

O *dispenser* de remédios é um dispositivo eletrônico projetado para ajudar na administração correta de medicamentos, especialmente para pessoas que precisam seguir regimes de medicação rigorosos. Ele organiza e dispensa medicamentos em horários programados, reduzindo a chance de esquecimentos ou erros de dosagem. Com um sistema de compartimentos, o *dispenser* armazena os medicamentos conforme a prescrição médica e emite alertas sonoros ou visuais para lembrar o usuário de tomar o remédio na hora certa.

Esses dispositivos são particularmente úteis para idosos e pacientes com condições crônicas que necessitam de múltiplas doses diárias de diferentes medicamentos. Além de facilitar o seguimento do tratamento, os *dispensers* de remédios podem ser conectados a *smartphones* ou sistemas de monitoramento, permitindo que cuidadores ou profissionais de saúde acompanhem o uso dos

medicamentos em tempo real. Isso melhora a aderência ao tratamento e ajuda a prevenir complicações associadas à administração incorreta de medicamentos.

Além de beneficiar os pacientes, os *dispensers* de remédios também aliviam a carga sobre os cuidadores, proporcionando-lhes maior tranquilidade ao garantir que o paciente receba a medicação corretamente. No entanto, a eficácia do dispositivo depende de uma programação precisa e de um bom entendimento por parte do usuário sobre seu funcionamento. Por isso, é essencial que o uso do *dispenser* seja acompanhado por orientações claras e suporte técnico quando necessário.

6.1 Bandeja de Remédios

A bandeja de remédios foi pensada para permitir o controle dos insumos disponibilizados pelo sistema. Nela, o *dispenser* depositará o remédio em forma de comprimido na bandeja. Com um sensor de proximidade, será possível controlar a abertura e fechamento da porta da bandeja. Além disso, o sistema poderá registrar o horário e a quantidade de comprimidos dispensados, permitindo um monitoramento preciso do consumo e ajudando a evitar erros de dosagem. Esse controle automatizado é especialmente útil para o gerenciamento de medicamentos de idosos, aumentando a segurança e a eficiência no cuidado com a saúde.

7 Diagramas e Componentes

Os diagramas elétricos são representações gráficas que mostram a disposição e a conexão dos componentes em um circuito elétrico. Eles são essenciais para a compreensão e a execução de projetos elétricos, pois permitem visualizar como os componentes estão interligados e como a corrente elétrica flui através do circuito.

Na Figura 3 vê-se a visão virtual do circuito físico do protótipo feito no TINKERCad, aplicativo usado para projetos de circuitos como Arduino. Nele observa-se como é feita a montagem do circuito com as ligações físicas dos componentes e cabos. Observa-se o protótipo do *dispenser* de medicamentos.

Nele é usado um sensor de distância para determinar a abertura e fechamento da porta da bandeja. Caso seja detectada mudança de estado da porta, novos remédios serão deixados no *dispenser*. Observa-se também o botão que é responsável por mudar o estado da aplicação de *stand-by* para inativo; e a chave (com *knob*) para ligar e desligar o protótipo; o servo motor é responsável pela abertura da porta da bandeja, e para recarregar os medicamentos; o piezo ou *buzzer* é responsável por emitir um sinal sonoro de alerta.

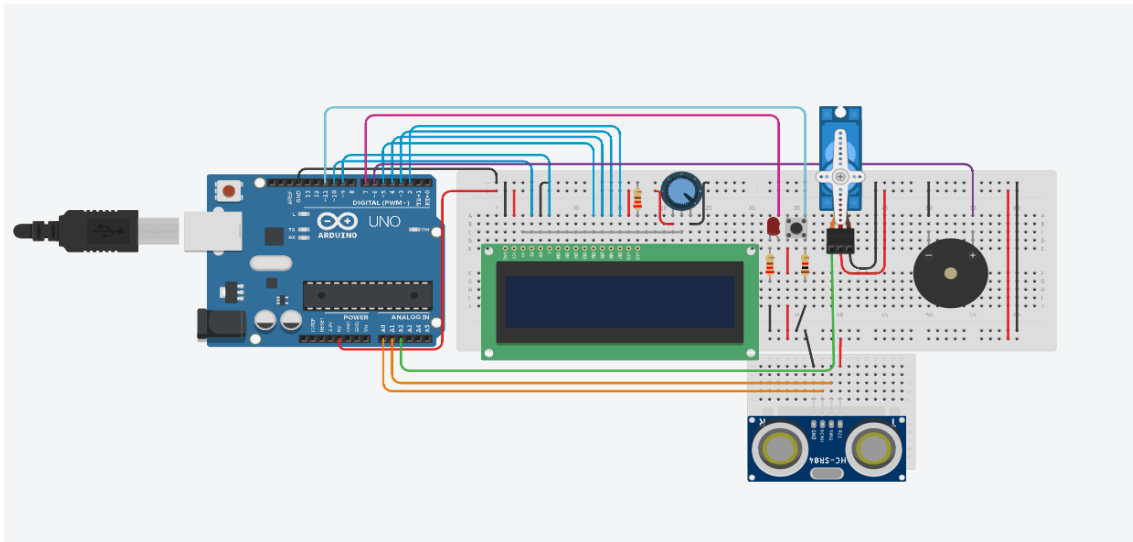
A codificação do *sketch* da lógica de controle está disponível no link do github listado nas referências. Nele está o código e mais imagens do *dispenser* funcionando, como prova de conceito. Além disso, o projeto está hospedado na plataforma de simulação TINKERCad da Autodesk.

7.1 Componentes

Os componentes eletrônicos são os elementos fundamentais em qualquer circuito ou dispositivo eletrônico, desempenhando funções específicas que permitem o funcionamento do sistema como um todo. Entre os mais comuns estão

resistores, que controlam o fluxo de corrente; capacitores, que armazenam e liberam energia; diodos, que permitem a passagem de corrente em uma única direção; transistores, que amplificam ou comutam sinais; e microcontroladores, que funcionam como o cérebro do sistema, controlando outros componentes e processando informações. Juntos, esses componentes formam a base de todo tipo de tecnologia, desde simples brinquedos eletrônicos até complexos sistemas de comunicação.

Figura 3 - Vista virtual do protótipo físico



Fonte: os autores

7.1.1 Botões

O botão é um componente eletrônico fundamental em muitos circuitos, atuando como um interruptor que permite ao usuário iniciar ou interromper uma ação específica. Em sistemas eletrônicos, como os baseados em Arduino, os botões são usados para enviar sinais que aciona ou desativa outras partes do circuito. Esses sinais são gerados quando o botão é pressionado, fechando um circuito e permitindo a passagem de corrente elétrica. Os botões podem ser configurados para realizar diversas funções, desde acionar motores até iniciar processos em interfaces de usuário (Makerhero, 2024a).

7.1.2 Servo motor

O servo motor é um tipo de motor elétrico que oferece controle preciso de posição angular, sendo amplamente utilizado em robótica, automação e modelagem. Diferente dos motores comuns, que giram continuamente, o servo motor é projetado para mover seu eixo até uma posição específica, mantendo essa posição com alta precisão. Ele funciona através de sinais *Pulse Width Modulation* (PWM), onde a largura do pulso determina o ângulo de rotação do motor. Devido à sua precisão e confiabilidade, os servomotores são ideais para aplicações que requerem

movimentos controlados, como o posicionamento de braços robóticos ou o ajuste de superfícies móveis (Makerhero, 2024d).

7.1.3 Tela LCD

A tela *Liquid Crystal Display* (LCD) é um componente essencial para a exibição de informações em dispositivos eletrônicos, permitindo que dados sejam apresentados de forma clara e visível ao usuário. As telas LCD são compostas por pixels que, ao serem ativados por uma corrente elétrica, exibem imagens ou texto. Em projetos de Arduino, as telas LCD são frequentemente usadas para mostrar dados em tempo real, como leituras de sensores, status de sistemas ou instruções de operação. Elas são altamente eficientes em termos de consumo de energia e podem ser encontradas em uma variedade de tamanhos, tornando-as ideais para uma ampla gama de aplicações, desde pequenos dispositivos portáteis até grandes painéis de exibição (Eletrogate, 2023).

7.1.4 Led

LED são dispositivos semicondutores que emitem luz quando uma corrente elétrica passa por eles. Ao contrário das lâmpadas incandescentes tradicionais, os LEDs são extremamente eficientes, convertendo a maior parte da energia elétrica em luz, com mínima dissipação de calor. Eles estão disponíveis em diversas cores e intensidades, determinadas pelos materiais semicondutores utilizados em sua fabricação. Devido à sua durabilidade, baixo consumo de energia e tamanho compacto, os LEDs são amplamente utilizados em uma variedade de aplicações, desde *displays* eletrônicos e iluminação residencial até indicadores de status em circuitos eletrônicos.

7.1.5 Piezo ou *Buzzer*

Os dispositivos piezoelétricos (piezo) ou *buzzers* são amplamente empregados em circuitos eletrônicos para a geração de som, cada um com características e aplicações específicas que influenciam sua escolha em projetos. O dispositivo opera com base no efeito piezoelétrico, onde materiais específicos vibram ao serem estimulados eletricamente, produzindo som em diversas frequências. Essa capacidade de gerar tons variados torna o piezo ideal para aplicações que demandam controle sobre a tonalidade, como alarmes personalizados ou sistemas de notificação sonora com diferentes notas. Contudo, o piezo necessita de um circuito externo, como um oscilador, para funcionar adequadamente, o que pode adicionar complexidade ao projeto (Samesky, 2024).

Por outro lado, o *buzzer* é um dispositivo que geralmente contém um oscilador interno, emitindo um som constante e fixo ao receber uma tensão. Sua simplicidade de operação o torna uma escolha prática para aplicações onde um alerta sonoro simples e repetitivo é suficiente, como em alarmes convencionais, timers e sinais de notificação. A principal vantagem do *buzzer* reside na facilidade de implementação, pois não requer circuitos adicionais para gerar som (Samesky, 2024).

7.1.6 Resistores

Um resistor é um componente eletrônico que tem a função de limitar ou controlar a corrente elétrica em um circuito. Ele faz isso oferecendo resistência à passagem da corrente, transformando parte da energia elétrica em calor, um fenômeno conhecido como efeito Joule. Os resistores são essenciais para proteger outros componentes do circuito contra sobrecargas de corrente e para ajustar os níveis de tensão e corrente conforme necessário. Eles são amplamente utilizados em diversos dispositivos eletrônicos, desde simples circuitos de iluminação até complexos sistemas de computação (Makerhero, 2024b).

7.1.7 Sensor de Distância

O sensor de distância ultrassônico HC-SR04, amplamente utilizado em projetos com Arduino, opera emitindo ondas ultrassônicas que, ao refletirem em um objeto, retornam ao sensor, permitindo o cálculo da distância com base no tempo de retorno. Com um alcance de 2 cm a 4 metros e precisão de aproximadamente 3 mm, ele é ideal para aplicações que necessitam de medição precisa de distâncias ou detecção de obstáculos. A integração com o Arduino é simples, utilizando pinos digitais para envio e recepção dos pulsos elétricos, e um código básico pode ser implementado para medir e exibir a distância, tornando-o uma ferramenta eficaz e acessível para projetos de automação e robótica (Makerhero, 2024c).

7.2 Sketch

O *sketch* é a implementação do código da lógica das funções importantes para o *dispenser*. Dentre elas, as que mais se destacam do projeto são os blocos `setup()` e `loop()` apresentados a seguir. O código fonte do projeto e os artefatos criados estão disponíveis em Adolfo (2024).

7.2.1 Funções básicas: `setup()` e `loop()`

As funções `setup()` e `loop()` no código do Arduino são a base de qualquer programa feito para ele. Ambas sendo necessárias para a estruturação do *sketch*. A função `setup()` é responsável por configurar o modo de operação das portas de Entrada/Saída do microcontrolador. Na Figura 4, observa-se as configurações para que o Arduino opere com sensores e atuadores.

Na função `loop()` são implementadas as regras de operação da aplicação. A Figura 5 mostra com mais clareza os trechos de regras e chamadas de outras funções. É possível ver como foram implementados o fechamento da bandeja de remédios e o abastecimento com medicamentos. Observa-se ainda, como é feito o desligamento e *reset* dos componentes do circuito.

A função `loop()` é o centro lógico da aplicação, onde se capta os diferentes sinais dos sensores (ultrassônico e botão) para decidir, por estruturas condicionais, como operar os atuadores.

Figura 4 - Função setup()

```
void setup() // Função usada para configurar os pinos, iniciar o serial e conectar os servos
{
  Serial.begin(9600); // Inicia o serial na frequência 9600
  pinMode(piezo, OUTPUT); // Define o piezo como saída
  pinMode(led, OUTPUT); // Define o LED como saída
  pinMode(trigPin, OUTPUT); // Define o trigPin como saída
  pinMode(echoPin, INPUT); // Define o echoPin como entrada
  delay(250); // Atraso de 150 milissegundos
  pinMode(buttonPin, INPUT); // Define o botão como entrada
  servo.attach(servoPin); // Conecta o pino do servo
  LCD.begin(16, 2); // Inicia o LCD
  LCD.setCursor(0, 0); // Define o cursor do LCD
}
```

Fonte: Os Autores

7.2.2 Função checkDistance()

A função `checkDistance()` mostrada na Figura 6, é utilizada para detectar a distância utilizando um sensor ultrassônico. Inicialmente, o pino `trigPin` é configurado para nível baixo (LOW), seguido por um atraso de 2 milissegundos. Em seguida, o pino é configurado para nível alto (HIGH) por 10 milissegundos, e depois novamente para LOW, criando um pulso de 10 milissegundos. O tempo de retorno do pulso é medido pela função `pulseIn()` no pino `echoPin`, armazenando o valor na variável `pingTime`. Finalmente, a distância ao alvo é calculada multiplicando `pingTime` por 0,034 e dividindo por 2, resultando o valor de `targetDistance`.

7.2.3 Funções Adicionais

Algumas funções foram necessárias para teste do *dispenser*, como por exemplo a função `displayDistance()` referenciado na Figura 7 e a função `resetDisplay()` para limpar a tela LCD como mostra a Figura 8.

A função `displayDistance()` é projetada para exibir a distância medida no display LCD. Ela posiciona o cursor na primeira linha e imprime a mensagem "Status: Em espera ". Em seguida, move o cursor para a segunda linha, onde exibe a mensagem "Distancia: ", seguida pelo valor da variável `targetDistance` e a unidade "cm". Isso permite que o usuário veja claramente o *status* e a distância medida em tempo real.

Por outro lado, a função `resetDisplay()` é usada para limpar o display. Ela posiciona o cursor na primeira linha e imprime espaços em branco para apagar qualquer texto existente. O mesmo processo é repetido para a segunda linha. Essa função é útil para garantir que o display esteja limpo antes de exibir novas informações, evitando sobreposição de texto e garantindo uma leitura clara.

Figura 5 - Função Loop()

```
void loop() // Função principal do código
{
  in11 = digitalRead(buttonPin); // Lê o botão
  delay(1000); // Atraso de 1 segundo
  if (in11 == HIGH){ // Se o botão estiver em nível alto
    if (buttonOnOff == 0){ // Se o estado de buttonOnOff for 0, muda para 1
      buttonOnOff = 1;
      Serial.print("Inativo");
    }
    else if(buttonOnOff == 1){ // Se o estado de buttonOnOff for 1, muda para 0
      buttonOnOff = 0;
      Serial.print("Ativo");
    }
  }
  if (buttonOnOff == 0){ // Se o estado de buttonOnOff for 0
    checkDistance(); // Checa a distância
    displayDistance(); // Exibe a distância
    if (targetDistance <= 25){ // Se a targetDistance for menor ou igual a 25
      resetDisplay(); // Reseta o display
      LCD.setCursor(0, 0);
      LCD.print("Status: Ativo");
      LCD.setCursor(0, 1);
      LCD.print("Dispensando");
      alert(); // Toca o Alerta
      servo.write(0); // Move o servo
      blink(); // Faz o LED piscar
      servo.write(90);
      countdown(5); // Faz a contagem regressiva
      checkDistance(); // Checa a distância novamente
    }
    else{
      servo.write(90); // Reseta o servo e o LED
      digitalWrite(led,LOW);
    }
  }
  else if (buttonOnOff == 1){ // Se o estado de buttonOnOff for 1
    servo.write(90); // Reseta todos os componentes
    digitalWrite(led,LOW);
    LCD.setCursor(0, 0);
    LCD.print("Status: Inativo");
    LCD.setCursor(0, 1);
    LCD.print("          ");
  }
}
```

Fonte: Os Autores

Figura 6 - checkDistance()

```
void checkDistance(){ // Função usada para detectar a distância
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);
  pingTime = pulseIn(echoPin, HIGH);
  targetDistance = 0.034 * pingTime / 2;
}
```

Fonte: Os Autores

Figura 7: displayDistance()

```
void displayDistance(){ // Função usada para exibir a distância e imprimir informações
  LCD.setCursor(0, 0);
  LCD.print("Status: Em espera ");
  LCD.setCursor(0,1);
  LCD.print("Distancia: ");
  LCD.print(targetDistance);
  LCD.print(" cm");
}
```

Fonte: Os Autores

Figura 8: resetDisplay()

```
+
void resetDisplay(){ // Função usada para resetar o display
  LCD.setCursor(0,0);
  LCD.print("                "); // Limpa a primeira linha
  LCD.setCursor(0,1);
  LCD.print("                "); // Limpa a segunda linha
}
```

Fonte: Os Autores

8 Considerações finais

Destacamos a importância do desenvolvimento de soluções tecnológicas voltadas para o cuidado e monitoramento de idosos, um público que vem crescendo em número e que demanda cada vez mais atenção especial. A utilização de plataformas como o Arduino e o ESP8266 demonstrou-se eficiente para a criação de sistemas acessíveis, personalizáveis e de baixo custo, capazes de melhorar significativamente a qualidade de vida desses indivíduos. Além disso, o projeto reforça a necessidade de uma abordagem interdisciplinar, combinando conhecimentos de engenharia de *software*, eletrônica e ciências da saúde para enfrentar os desafios associados ao envelhecimento da população.

Este trabalho também ressalta a relevância da usabilidade e da simplicidade na concepção de dispositivos destinados a usuários com pouca familiaridade com tecnologias digitais. A criação de interfaces intuitivas e de sistemas de fácil manutenção é essencial para garantir que essas inovações cumpram seu papel de forma eficaz. Por fim, é importante sublinhar o potencial expansivo deste projeto, como a implementação de um dispositivo portátil de chamada por socorro, com ESP8266 para acesso sem fio a sistemas de comunicação de pânico, que pode servir de base para futuras pesquisas e desenvolvimentos em outras áreas da saúde e bem-estar, ampliando ainda mais o impacto positivo que a tecnologia pode ter na vida das pessoas.

Pode-se afirmar que a integração das Tecnologias Digitais de Informação e Comunicação, como a plataforma Arduino e o módulo ESP8266, oferece soluções inovadoras e eficazes para melhorar a qualidade de vida dos idosos. Ao automatizar processos de controle de medicação e implementar sistemas de monitoramento remoto, pode-se proporcionar maior segurança e autonomia para

pessoas idosas, além de tranquilidade para suas famílias. O desenvolvimento desses sistemas confirmou a importância da tecnologia na promoção do bem-estar e na prevenção de riscos à saúde, evidenciando como a inovação tecnológica pode ser uma aliada poderosa no enfrentamento dos desafios do envelhecimento populacional. Portanto, este projeto não apenas apresenta um avanço técnico, mas também um impacto social significativo, ao contribuir para um futuro onde os idosos possam viver com mais conforto e segurança.

Agradecimentos

Quero expressar minha profunda gratidão ao professor Carlos Roland, cujo apoio e orientação foram fundamentais para a realização deste trabalho. Sua dedicação, conhecimento e disponibilidade para ajudar em todas as etapas do projeto foram imprescindíveis para que eu pudesse alcançar este resultado. Agradeço também a toda a equipe de docentes do Uni-Facef, que com seu comprometimento e excelência no ensino, contribuíram significativamente para a minha formação acadêmica e pessoal.

Quero também agradecer ao Gabriel Ferreira que começou comigo esse projeto, mas por forças maiores não pudemos concluir juntos. Desde o começo sempre apoiou o projeto com tudo que pode com seu conhecimento e expertise no assunto. Esse trabalho é uma homenagem à nossa amizade que surgiu há 4 anos quando entramos no curso de Engenharia de Software durante a pandemia.

Não poderia deixar de agradecer à minha família, que esteve ao meu lado durante toda essa jornada acadêmica. Seu apoio incondicional, compreensão e incentivo foram fundamentais para que eu pudesse me dedicar plenamente a este trabalho. A paciência e o amor que recebi de vocês foram a base sobre a qual construí minha trajetória até aqui. Este trabalho é dedicado a vocês, que sempre acreditaram em mim e me proporcionaram as condições necessárias para que eu pudesse chegar a este momento.

Referências

ADOLFO, Leonardo. **TCC-Prova-De-Conceito**. 2024. Disponível em: <https://github.com/LeonardoAdolfo/TCC-Prova-De-Conceito>. Acesso em: 26 set. 2024.

ARDUINO E CIA. **O que é Arduino (e para que serve)**. Disponível em: <https://www.arduinoocia.com.br/o-que-e-arduino-e-para-que-serve/>. Acesso em: 26 set. 2024.

BANKS, Andy; GUPTA, Rahul. **MQTT Version 3.1.1**. 2014. OASIS Standard.

ALBUQUERQUE, Sandra Márcia Ribeiro Lins. **Qualidade de vida do idoso: a assistência domiciliar faz a diferença?** 2003. Casa do Psicólogo.

ELETROGATE. **Apostila Kit Arduino Advanced**. 2023. Disponível em: <https://conteudo.eletrogate.com/obrigado-apostila-arduino-advanced>. Acesso em: 26 set. 2024.

EMBARCADOS. **ESP8266: Apresentação do módulo para projetos.** Disponível em: <https://embarcados.com.br/modulo-esp8266/>. Acesso em: 26 set. 2024.

HUNKELER, Urs; TRIMINTON, Hong Linh; STANFORD-CLARK, Andy. **MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks.** In: Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on. IEEE, 2008. p. 791-798.

HIPÓLITO, José Guilherme; SILVA, Miqueias de Jesus da. **Automação residencial com arduino.** 2018.

ITIGIC. **Programação em Arduino: linguagens, programas e primeiros passos.** 2022. Disponível em: <https://itigic.com/pt/programming-in-arduino-languages-programs-and-first-steps/>. Acesso em: 21 set. 2024.

LORENZETTI, Jorge et al. **Tecnologia, inovação tecnológica e saúde: uma reflexão necessária.** Texto & Contexto-Enfermagem, v. 21, p. 432-439, 2012.

MAGALHÃES, Roberto. **Introdução ao ESP8266 e ESP32 no Arduino IDE.** COMPRACO, 10 jun. 2024. Disponível em: <https://compraco.com.br/blogs/tecnologia-e-desenvolvimento/introducao-ao-esp8266-e-esp32-no-arduino-ide>. Acesso em: 26 set. 2024.

MAKERHERO. **O que é Arduino, para que serve e primeiros passos.** 2023. Disponível em: <https://www.makehero.com/blog/o-que-e-arduino/>. Acesso em: 21 set. 2024.

MAKERHERO. **Botão: o que é, tipos, função e aplicações.** 2024a. Disponível em: <https://www.makehero.com/guia/componentes-eletronicos/botao/>. Acesso em: 26 set. 2024.

MAKERHERO. **Guia de Componentes Eletrônicos: Resistor.** 2024b. Disponível em: <https://www.makehero.com/guia/componentes-eletronicos/resistor/>. Acesso em: 26 set. 2024.

MAKERHERO. **Sensor Ultrassônico HC-SR04.** 2024c. Disponível em: <https://www.makehero.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>. Acesso em: 26 set. 2024

MAKERHERO. **O que é servo motor? Funcionamento, tipos e como testar.** 2024d. Disponível em: <https://www.makehero.com/blog/o-que-e-servo-motor/>. Acesso em: 26 set. 2024.

MCROBERTS, Michael. **Arduino básico.** Novatec Editora, 2018

MINAYO, Maria Cecília de Souza. **O envelhecimento da população brasileira e os desafios para o setor saúde.** Cadernos de Saúde Pública, v. 28, p. 208-210, 2012.

NOVAES, Hillegonda Maria D. **Avaliação de programas, serviços e tecnologias em saúde**. Revista de Saúde Pública, v. 34, n. 5, p. 547-549, 2000.

SAME SKY. **Buzzer Basics - Technologies, Tones, and Drive Circuits**. Disponível em: <https://www.sameskydevices.com/blog/buzzer-basics-technologies-tones-and-driving-circuits>. Acesso em: 26 set. 2024.

SHAHIDIAN, Shakib. **Electrónica e Automação com Arduino**. Aparentamentos para os alunos. Universidade de Évora, 2013.

WIRINGPROJECT. **Wiring Framework**. GitHub. Disponível em: <https://github.com/WiringProject/Wiring/>. Acesso em: 21 set. 2024.