

Protótipo IoT: Ventilador Automatizado

Felipe Campolina¹, Gabriel Martins¹, Leandro Guido¹, Marcelo Augusto¹

¹Instituto de Ciências Exatas e Informática (ICEI) –
Pontifícia Universidade Católica (PUC-MG)

1. Introdução

1.1. Problema Identificado

A problemática central deste projeto concentra-se na gestão eficaz das temperaturas noturnas em Belo Horizonte, uma região caracterizada por uma notável variação térmica, com altas temperaturas durante o dia que dão lugar a noites mais frescas. Esta variação pode afetar adversamente tanto o conforto térmico dos habitantes quanto o consumo energético necessário para manter uma temperatura interna adequada durante a noite. Conforme indicado por Pinto et al. [Pinto et al. 2017], esta situação representa um desafio significativo, necessitando de soluções inovadoras para o controle e ajuste da temperatura interna nas horas noturnas, com o objetivo de assegurar o conforto dos residentes enquanto minimiza o uso excessivo de energia.

1.2. Hipóteses de Solução

Para abordar o desafio das variações térmicas, propomos uma solução multifacetada que integra a Internet das Coisas (IoT), desenvolvimento de um aplicativo móvel em Flutter, e o emprego do Arduino como microcontrolador, enriquecido com um módulo Wi-Fi para transmissão de dados. O projeto visa transformar um ventilador comum em um dispositivo automatizado inteligente, capaz de ajustar sua velocidade em resposta às mudanças de temperatura ambiente, especialmente durante a noite. A automação será alcançada através do uso de sensores de temperatura conectados ao Arduino, que, equipado com um módulo Wi-Fi, facilitará a comunicação sem fio e a transmissão de dados em tempo real entre o ventilador e o aplicativo móvel desenvolvido em Flutter. Este aplicativo permitirá aos usuários não apenas monitorar as condições térmicas remotamente, mas também controlar o ventilador com precisão, personalizando suas configurações para atender às preferências individuais e garantindo um ambiente adaptado e confortável diante das flutuações térmicas diárias. Esta abordagem integrada destaca a combinação de hardware e software para promover uma melhor qualidade de vida através da tecnologia.

1.3. Objetivos

Em primeiro lugar, o projeto visa proporcionar maior conforto ao usuário ao adaptar um ventilador para operar de forma automatizada, sem a necessidade de intervenção manual para desligar ou reduzir sua velocidade, mesmo durante a noite. Isso elimina a inconveniência de ter que acordar e levantar para ajustar as configurações do ventilador, garantindo um sono contínuo e sem interrupções. Em segundo lugar, busca-se promover a economia de energia elétrica, otimizando o uso do ventilador de acordo com as condições climáticas.

1.4. Estrutura do Documento

Este documento está meticulosamente organizado para proporcionar uma compreensão abrangente e detalhada do projeto de protótipo IoT de ventilador automatizado. Inicia-se com uma introdução, que delinea o escopo do documento. Em seguida, procede-se da seguinte forma:

Seção 2 - Revisão Bibliográfica: Nesta seção, realiza-se uma revisão literária extensa, enfatizando o papel fundamental do Arduino em múltiplas aplicações de automação e controle, além de discutir a importância das soluções baseadas em IoT para o desafio proposto. Esta revisão abrange estudos anteriores que investigam a sinergia entre hardware e software em sistemas de automação doméstica e industrial, juntamente com a utilização de tecnologias de comunicação sem fio em projetos IoT.

- *Subseção 2.1 - Relação com as Disciplinas Estudadas:* Estabelece-se um diálogo entre os projetos mencionados na revisão bibliográfica e os conceitos fundamentais das áreas de arquitetura de computadores, redes de computadores e sistemas operacionais. Essa discussão evidencia como o entendimento desses campos é vital para desenvolver soluções inovadoras e eficazes em IoT, fornecendo um embasamento teórico para as tecnologias e métodos empregados no desenvolvimento do ventilador automatizado.

Seção 3 - Metodologia: Nesta parte, detalha-se a metodologia utilizada no desenvolvimento e na avaliação do protótipo IoT, organizando-a em subseções específicas para promover uma compreensão clara e aprofundada.

- *Subseção 3.1 - Cronograma do Projeto:* Apresenta-se o cronograma do projeto, destacando as fases de desenvolvimento, os marcos importantes e a sequência prevista de atividades. Este cronograma fornece uma estrutura temporal para o projeto, orientando-se na gestão eficaz do tempo e no cumprimento de prazos.
- *Subseção 3.2 - Metodologia de Desenvolvimento:* Detalha-se a abordagem metodológica adotada para o desenvolvimento do projeto, incluindo estratégias de planejamento, execução e monitoramento. Esta seção explica como as decisões metodológicas sustentam o desenvolvimento eficiente do protótipo e a consecução dos objetivos do projeto.
- *Subseção 3.3 - Desenvolvimento do Hardware:* Foca-se nos aspectos técnicos relativos ao desenvolvimento do hardware do projeto. Descrevem-se os componentes selecionados, como o Arduino UNO, módulo relé, módulo Bluetooth/Wi-Fi, sensor de temperatura e protoboard, e detalha-se o processo de montagem, programação e teste do circuito.
- *Subseção 3.4 - Desenvolvimento do Software:* Aborda-se o desenvolvimento do software, incluindo o aplicativo móvel criado com Flutter. Especificam-se as funcionalidades principais, como controle de ligar/desligar, ajuste de velocidade, modo automático, timer de desligamento e relatório de funcionamento. Discute-se também o design da interface do usuário e os procedimentos de teste adotados.
- *Subseção 3.5 - Método de Avaliação:* Explica-se o método de avaliação utilizado para testar a eficácia e a viabilidade do sistema. Detalham-se como os dados sobre a variação de temperatura e as mudanças de velocidade são coletados e analisados, descrevendo-se o processo de medição e os critérios de avaliação aplicados.

Esta estrutura detalhada visa fornecer uma visão holística e profunda do projeto, desde a fundamentação teórica até a aplicação prática e avaliação, facilitando o entendimento e a replicabilidade do trabalho.

2. Revisão Bibliográfica

Nos últimos anos, o Arduino tem desempenhado um papel crucial no desenvolvimento de soluções de automação e controle, oferecendo uma abordagem flexível e acessível para uma variedade de aplicações. De Lima [de Lima 2013] conduziu um estudo abrangente sobre o controle de temperatura utilizando a placa Arduino. Este trabalho não apenas demonstra a eficácia do Arduino em sistemas de baixo custo, mas também fornece insights valiosos sobre a integração de hardware e software para alcançar objetivos de controle específicos.

Bueno [Bueno 2015] expandiu ainda mais os limites do Arduino ao utilizar a plataforma para controle de iluminação e temperatura via celular. Sua pesquisa destaca não apenas a versatilidade do Arduino em diferentes cenários de automação residencial, mas também ressalta a importância da conectividade e integração com dispositivos móveis.

Santos [Santos 2018] explorou as capacidades do Arduino em ambientes industriais ao utilizar um Shield Ethernet para monitoramento de luminosidade, controle de temperatura e outros dispositivos. Este estudo prático demonstra como o Arduino pode ser uma solução viável para aplicações industriais que requerem precisão e confiabilidade.

Embora não diretamente relacionado ao Arduino, o relatório técnico de Luz e Lott [Luz and Lott 2024] sobre o controle de temperatura de um sistema aletado com ventilação forçada destaca a importância do controle preciso da temperatura em sistemas de refrigeração. Esses conceitos são fundamentais em muitas aplicações onde o Arduino é frequentemente utilizado.

O MakerHero fornece um guia detalhado sobre a transformação de um ventilador convencional em um dispositivo controlado por Bluetooth, utilizando o Arduino [MakerHero 2024]. O autor destaca a acessibilidade ao conforto tecnológico a baixo custo, inclusão de pessoas com deficiência e o desenvolvimento da cultura maker. O projeto detalha os materiais e etapas necessários, incluindo segurança, desmonte do ventilador, montagem do circuito e programação do Arduino. O código permite controlar a velocidade do ventilador via Bluetooth. O autor enfatiza a importância da reutilização de eletrônicos e incentiva os leitores a explorarem e ampliarem o projeto, contribuindo para a redução do lixo tecnológico e o crescimento da comunidade maker.

Margolis [Margolis 2011] oferece uma compreensão abrangente do universo Arduino com o "Arduino Cookbook". Esta referência é crucial para iniciantes e especialistas, abrangendo uma ampla gama de projetos e técnicas. O livro serve como um guia prático que capacita os usuários a expandirem suas habilidades em automação e controle, destacando a adaptabilidade do Arduino em superar desafios de engenharia.

Mwamburi [Mwamburi et al. 2017] apresentam um sistema inovador de controle remoto para ventiladores baseado em Arduino e GSM. Este trabalho exemplifica como a integração de comunicações móveis com o Arduino pode levar a soluções de automação residencial mais eficientes e acessíveis. O projeto destaca a importância da comunicação

sem fio na implementação de sistemas de controle remoto, promovendo a conveniência e a eficiência energética.

Abubakar e Jalo [Abubakar and Jalo 2023] desenvolveram um sistema de controle de temperatura e umidade baseado em Arduino para controle de velocidade de ventiladores. Esta abordagem inovadora para a regulação climática em ambientes internos não só demonstra a versatilidade do Arduino em gerenciar variáveis ambientais, mas também sublinha a aplicabilidade do Arduino em criar ambientes de vida mais confortáveis e sustentáveis.

Essas referências evidenciam claramente a versatilidade e eficácia do Arduino como uma plataforma para automação e controle em uma variedade de cenários, desde sistemas de baixo custo até aplicações industriais e residenciais complexas. A revisão bibliográfica apresentada ilustra a amplitude de aplicações do Arduino em diversos contextos, desde automação residencial até soluções industriais, enfatizando seu papel em projetos que abordam aspectos fundamentais da arquitetura de computadores, redes de computadores e sistemas operacionais. As referências discutidas demonstram não apenas a adaptabilidade e eficácia do Arduino, mas também como essa plataforma pode ser a base para a implementação de soluções computacionais avançadas.

2.1. Relação com as disciplinas estudadas

Projetos como o controle de temperatura e umidade para a regulação da velocidade de ventiladores [Abubakar and Jalo 2023], e o sistema de controle remoto baseado em Arduino e GSM para ventiladores [Mwamburi et al. 2017], exemplificam a interação entre a arquitetura de computadores e redes de computadores. Esses projetos dependem de hardware específico (arquitetura de computadores) e comunicação sem fio (redes de computadores) para funcionar corretamente, destacando a necessidade de conhecimentos integrados dessas áreas para desenvolver soluções eficazes.

Além disso, a implementação de sistemas de controle e automação, como os descritos nas referências, frequentemente exige a utilização de sistemas operacionais, seja em um nível mais básico, como os firmwares que rodam diretamente nos microcontroladores do Arduino, ou em sistemas mais complexos que facilitam a comunicação entre dispositivos através de redes. Isso ressalta a relevância dos sistemas operacionais na gestão de recursos de hardware e na execução eficiente de tarefas de software, crucial para a realização de projetos interdisciplinares que envolvem múltiplas áreas da computação.

Portanto, os estudos e projetos mencionados nesta revisão não apenas confirmam a viabilidade do Arduino como uma plataforma para exploração acadêmica e inovação tecnológica, mas também destacam a importância de uma abordagem integrada que abrange arquitetura de computadores, redes de computadores e sistemas operacionais. Essa sinergia entre as áreas é fundamental para o desenvolvimento de soluções computacionais que sejam não apenas inovadoras, mas também robustas, eficientes e capazes de atender às crescentes demandas de automação e controle em diversos setores.

3. Metodologia

3.1. Cronograma

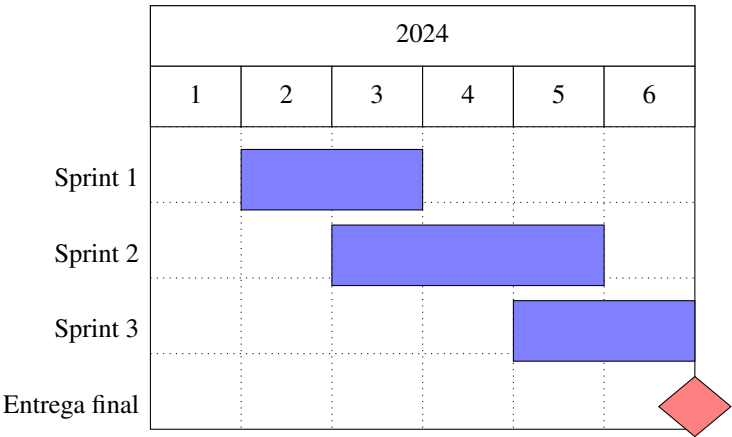


Figure 1. Cronograma

O cronograma (Figura 1) mostra as sprints do projeto ao longo de 6 meses, e também quando ocorrerá a entrega final. Além das entregas de sprints, reuniões semanais ao longo do semestre serão feitas para desenvolver o projeto, resolver problemas e tomar decisões em conjuntos.

3.2. Metodologia de Desenvolvimento

A metodologia de desenvolvimento para este projeto é dividida em duas principais seções: desenvolvimento do hardware e desenvolvimento do software. Abaixo, cada seção é detalhada, descrevendo os componentes utilizados, o processo de desenvolvimento e os testes realizados.

3.3. Desenvolvimento do Hardware

- **Arduino UNO:** O cérebro do projeto, controla as funcionalidades do sistema.
- **Módulo Relé:** Permite o Arduino controlar cargas de alta potência, como o ventilador.
- **Módulo Bluetooth/Wi-Fi:** Oferece conectividade sem fio para comunicação com o aplicativo móvel.
- **Sensor de Temperatura:** Monitora a temperatura ambiente para ajuste automático do ventilador.
- **Protoboard:** Utilizado na fase de prototipagem para testar conexões e layouts de circuito.

3.3.1. Processo de Desenvolvimento

O desenvolvimento do hardware segue os seguintes passos:

1. **Montagem do Circuito:** A montagem inicia-se pela colocação do Arduino UNO no centro do protoboard. Em seguida, o Sensor de Temperatura é conectado ao Arduino utilizando cabos jumper, garantindo que a pinagem corresponda aos pinos

de entrada analógica do Arduino. O Módulo Relé é então interligado ao Arduino, conectando o pino de controle do relé a um pino digital do Arduino configurado como saída. Resistores e LEDs são posicionados para indicar o status operacional do sistema; os LEDs são conectados através de resistores apropriados para limitar a corrente, evitando danos. O Módulo Bluetooth/Wi-Fi é conectado via comunicação serial para permitir a comunicação sem fio. Toda a montagem é cuidadosamente realizada seguindo o esquemático de design, com atenção especial à organização e à minimização de interferências.

2. **Programação do Arduino:** A programação começa com o desenvolvimento de um sketch no Arduino IDE, que inicializa componentes e define as variáveis necessárias. O foco principal é ler os dados do sensor de temperatura e, com base nesses dados e nas preferências do usuário recebidas do aplicativo Flutter, controlar o estado e a velocidade do ventilador utilizando um módulo relé. A comunicação entre o Arduino e o aplicativo Flutter é realizada através do módulo Wi-Fi. Utilizamos um formato simples para as mensagens, facilitando o envio de comandos e a recepção de configurações do aplicativo, assim como o retorno de informações de status e dados de temperatura para o aplicativo. Esta abordagem permite ajustes em tempo real e feedback sobre o ambiente para o usuário. Comentários no código ajudam a esclarecer a funcionalidade implementada e facilitam futuras atualizações ou manutenções do sistema. Este método garante uma interação eficaz entre o hardware controlado pelo Arduino e o software do aplicativo Flutter, proporcionando uma experiência de usuário fluida e responsiva.
3. **Testes:** A fase de testes começa com a verificação da montagem do circuito para assegurar que todas as conexões estão corretas e seguras. Testes funcionais são realizados para garantir que o Arduino está corretamente lendo a temperatura ambiente e controlando o ventilador como esperado. Isso inclui ajustar o sensor de temperatura para diferentes valores usando uma fonte de calor controlada e verificar a resposta do sistema. Testes de comunicação sem fio são conduzidos para assegurar que o aplicativo móvel e o Arduino podem se comunicar eficientemente, sem interrupções ou latências significativas. Esses testes são críticos para garantir que o sistema opere de forma confiável em condições reais de uso.

3.4. Desenvolvimento do Software

- Flutter
- Firebase

3.4.1. Processo de Desenvolvimento

O desenvolvimento do software inclui:

1. **Design da Interface do Usuário (UI):** A interface do usuário é projetada para ser intuitiva e fácil de navegar, enfatizando uma experiência de usuário fluida. Utilizamos o Flutter para criar uma UI moderna que apresenta componentes visuais claros para todas as funcionalidades principais, incluindo botões para ligar/desligar o ventilador, ajustar a velocidade, ativar o modo automático (noturno), configurar o modo automático, definir um timer e visualizar relatórios de

funcionamento. O design também incorpora visualizações gráficas para os relatórios, mostrando linhas do tempo com a temperatura ambiente e as velocidades do ventilador ao longo do período selecionado. A UI é estruturada em torno de uma navegação simples, com uma barra inferior para acesso rápido às diferentes seções do aplicativo.

2. **Implementação das Funcionalidades:** O desenvolvimento do aplicativo começa com a implementação das funcionalidades chave:

- *Ligar/Desligar e Trocar Velocidade:* Implementamos botões que enviam comandos ao Arduino para controlar o ventilador de forma direta.
- *Modo Automático (Noturno):* Uma tela de configuração permite ao usuário definir uma temperatura desejada para ativação automática do ventilador. O aplicativo, então, se comunica com o Arduino para ajustar o funcionamento com base na leitura do sensor de temperatura.
- *Timer:* Uma funcionalidade de timer é desenvolvida para permitir ao usuário programar um desligamento automático do ventilador após um intervalo de tempo definido.
- *Relatório de Funcionamento:* Implementamos uma seção de relatórios que exibe gráficos com a evolução da temperatura e da velocidade do ventilador ao longo do tempo, utilizando Firebase para armazenar e recuperar os dados históricos.

A integração com Firebase facilita o armazenamento de configurações e a coleta de dados operacionais, permitindo uma experiência de usuário rica e personalizada.

3. **Testes:** A fase de testes é essencial para assegurar a qualidade do aplicativo. Realizamos uma série de testes para verificar a compatibilidade em diferentes dispositivos e sistemas operacionais, a usabilidade da interface do usuário e a estabilidade das funcionalidades implementadas. Testes de unidade e de interface são aplicados para identificar e corrigir bugs, enquanto testes de integração asseguram que a comunicação entre o aplicativo e o hardware do Arduino ocorra sem problemas. Feedback dos usuários é coletado durante a fase de testes beta para fazer ajustes finais e melhorar a experiência do usuário.

3.5. Método de Avaliação

O sistema baseado em Arduino é projetado para registrar continuamente os dados referentes à variação da temperatura ambiente e às alterações nas velocidades do ventilador durante o período de operação. Esses dados críticos são essenciais para o monitoramento e análise do desempenho do sistema em tempo real. Após a coleta, os dados são enviados para o aplicativo desenvolvido em Flutter, que é responsável por processar e organizar essas informações em duas linhas do tempo distintas. Essas linhas do tempo fornecem uma representação visual clara e intuitiva das variações de temperatura e das mudanças de velocidade do ventilador ao longo do tempo, permitindo aos usuários compreender facilmente o comportamento do sistema.

Para avaliar a eficácia e a viabilidade do projeto de maneira objetiva, adotamos um modelo de medição baseado em evidências empíricas. Esse modelo envolve a realização de uma série de testes controlados, concebidos para simular diversas condições ambientais e avaliar a resposta do sistema. Durante esses testes, um ar-condicionado foi utilizado

para alterar de forma controlada a temperatura ambiente, criando cenários variados para o sistema responder. Paralelamente, um termo higrômetro digital é empregado para verificar a exatidão das leituras de temperatura realizadas pelo sensor acoplado ao Arduino. Essa abordagem permite uma comparação direta entre os dados de temperatura e velocidade registrados pelo sistema e as representações geradas no aplicativo.

Os dados coletados durante os testes são meticulosamente anotados e submetidos a uma análise comparativa com as linhas do tempo visualizadas no aplicativo. Essa análise é crucial para identificar qualquer discrepância ou incoerência no funcionamento do sistema, assegurando que as informações apresentadas no aplicativo reflitam com precisão as condições reais de operação. Através desse processo rigoroso de validação, podemos confirmar a confiabilidade e a precisão do sistema, garantindo que o projeto atenda aos critérios de desempenho estabelecidos.

References

- Abubakar, A. and Jalo, S. M. (2023). Arduino-based temperature and humidity control system for fan speed. Technical report, Independent University, Bangladesh.
- Bueno, G. N. D. M. (2015). Controle de iluminação e temperatura pela plataforma arduino via celular. Master's thesis, Universidade Tecnológica Federal do Paraná.
- de Lima, G. F. (2013). Controle de temperatura de um sistema de baixo custo utilizando a placa arduino. *IX Congresso de Iniciação Científica do IFRN*.
- Luz, A. P. and Lott, A. B. M. (2024). Controle de temperatura de um sistema aletado com ventilação forçada. Technical report, Universidade Federal de Minas Gerais.
- MakerHero (2024). Controle um ventilador pelo bluetooth do seu smartphone.
- Margolis, M. (2011). *Arduino Cookbook*. O'Reilly Media.
- Mwamburi, L. M. et al. (2017). Design and implementation of a remote controlled fan based on arduino and gsm.
- Pinto, J. L. C., Silva, F. S. d., Elmiro, M. A. T., and Nobrega, R. A. d. A. (2017). Análise comparativa dos mapeamentos de temperaturas obtidos por imagens termais e medidas de campo em belo horizonte – mg. *Revista Geografias*, page 107–118.
- Santos, R. V. d. (2018). Uso do arduino e shield ethernet para monitoramento de luminosidade, controle de temperatura e dispositivos. Trabalho de conclusão de curso, Universidade Tecnológica Federal do Paraná, Ponta Grossa.