

SISTEMA DE AQUISIÇÃO DE DADOS DE CORRENTE, TENSÃO E TEMPERATURA DE FARÓIS AUTOMOTIVOS

Eduardo Santos, Lucas Rodrigues, Paula Santos.

Orientador: Mario Francisco Guerra Boaratti

Faculdade de Tecnologia de São Bernardo do Campo

Eduardo.santos27@aluno.cps.sp.gov.br , lucas.silva181@aluno.cps.sp.gov.br ,

paula.santos3@aluno.cps.sp.gov.br , mario.boaratti@cps.sp.gov.br

RESUMO

O monitoramento contínuo de componentes eletroeletrônicos automotivos é um fator crítico para a garantia da qualidade e segurança veicular, demandando ensaios laboratoriais de longa duração altamente confiáveis. Este trabalho apresenta o desenvolvimento de um sistema de aquisição de dados e telemetria baseado em Internet das Coisas (IoT), projetado especificamente para o monitoramento de corrente, tensão e temperatura em faróis automotivos. Adotando a abordagem metodológica *Design Science Research* (DSR), a pesquisa resultou no desenvolvimento de um protótipo funcional estruturado em torno do microcontrolador ESP32. Para a coleta de dados de potência, integrou-se o sensor de alta precisão INA226. A infraestrutura de hardware foi concebida com foco na estabilidade de sinal, implementando um plano de aterramento comum (GND) para mitigar ruídos eletromagnéticos de bancada. O fluxo de informações foi projetado com redundância local de armazenamento por meio de um módulo Micro SD (com barramento de seleção via pino D4), operando de forma simultânea à transmissão remota de pacotes de dados formatados em JSON via protocolo MQTT, direcionados ao *broker* EMQX. Alertas preditivos e notificações de status foram implementados em tempo real por meio da API do *Telegram* (*Universogrambot*). Os resultados demonstraram a eficácia do artefato na automação dos ensaios, eliminando a necessidade de intervenção manual constante, garantindo a integridade do registro histórico das variáveis e viabilizando o gerenciamento remoto e seguro de testes de fadiga prolongados.

Palavras-chave: IOT. ESP32. Telemetria automotiva. Aquisição de dados. DSR.

ABSTRACT

The continuous monitoring of automotive electronic components is a critical factor for quality assurance and vehicle safety, requiring highly reliable, long-duration laboratory tests. This paper presents the development of a data acquisition and telemetry system based on the Internet of Things (IoT), specifically designed for monitoring current, voltage, and temperature in automotive headlights. Following the Design Science Research (DSR) methodological approach, the research resulted in a functional prototype built around the ESP32 microcontroller. For power data collection, a high-precision INA226 sensor was integrated. The hardware infrastructure was designed focusing on signal stability, implementing a common ground (GND) scheme to mitigate electromagnetic noise in the test bench. The data pipeline features local storage redundancy via a Micro SD card module (using pin D4 for chip select), operating concurrently with the remote transmission of JSON-formatted data packages using the MQTT protocol, directed to the EMQX broker. Predictive alerts and status notifications were deployed in real time through the Telegram API (Universogrambot). The results demonstrated the effectiveness of the artifact in automating trials, eliminating the need for constant manual intervention, ensuring the historical integrity of the measured variables, and enabling the secure and remote management of extended durability tests.

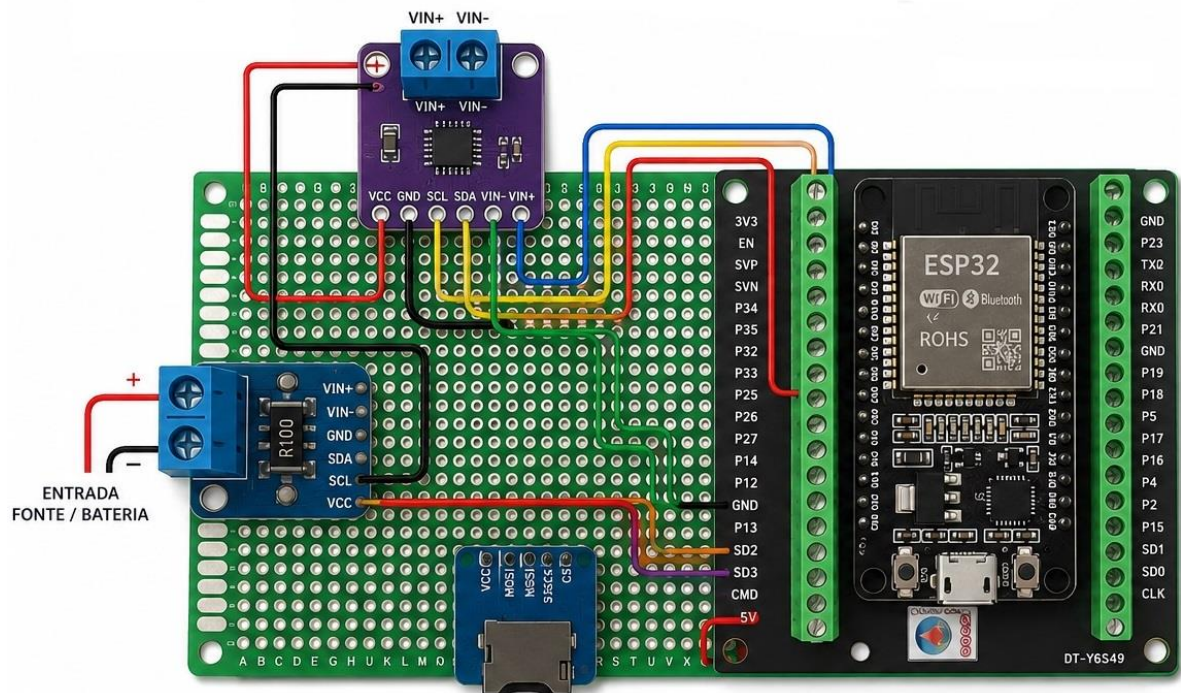
Keywords: IOT. ESP32. Automotive telemetry. Data acquisition. DSR.

1. Introdução

A evolução tecnológica na indústria automotiva tem sido impulsionada de forma acelerada pela eletrônica embarcada e pela busca por eficiência energética e segurança ativa. No cenário contemporâneo, os sistemas de iluminação, especialmente os faróis automotivos de alta performance (como sistemas em LED e *Full-LED*), deixaram de ser componentes puramente elétricos para se tornarem sistemas optoeletrônicos complexos. Devido ao alto fluxo luminoso e às restrições de espaço físico no design veicular, esses componentes operam sob severas condições de estresse térmico e variações de carga elétrica. Consequentemente, o monitoramento rigoroso de grandezas como corrente, tensão e temperatura torna-se um requisito crítico nas etapas de homologação e garantia de qualidade.

A Figura 1 apresenta o sistema eletrônico embarcado de monitoramento de dados desenvolvido.

Figura 1: Protótipo de sistema de monitoramento de dados.



Fonte: Própria Autoria (2026).

Tradicionalmente, a validação desses sistemas em ambiente laboratorial exige ensaios de durabilidade e fadiga de longa duração, os quais frequentemente estendem-se por centenas de horas contínuas. Na metodologia convencional, a coleta de dados de variáveis elétricas e térmicas costuma depender de intervenções manuais periódicas ou de sistemas de aquisição locais baseados em arquiteturas proprietárias e de alto custo (como osciloscópios dedicados e registradores de dados isolados). Essa abordagem tradicional apresenta vulnerabilidades latentes, tais como a descontinuidade na amostragem temporal, a suscetibilidade a erros humanos de transcrição e a impossibilidade de uma atuação preditiva ou remota caso ocorram anomalias térmicas ou sobrecorrentes durante a madrugada ou períodos de ausência de operadores.

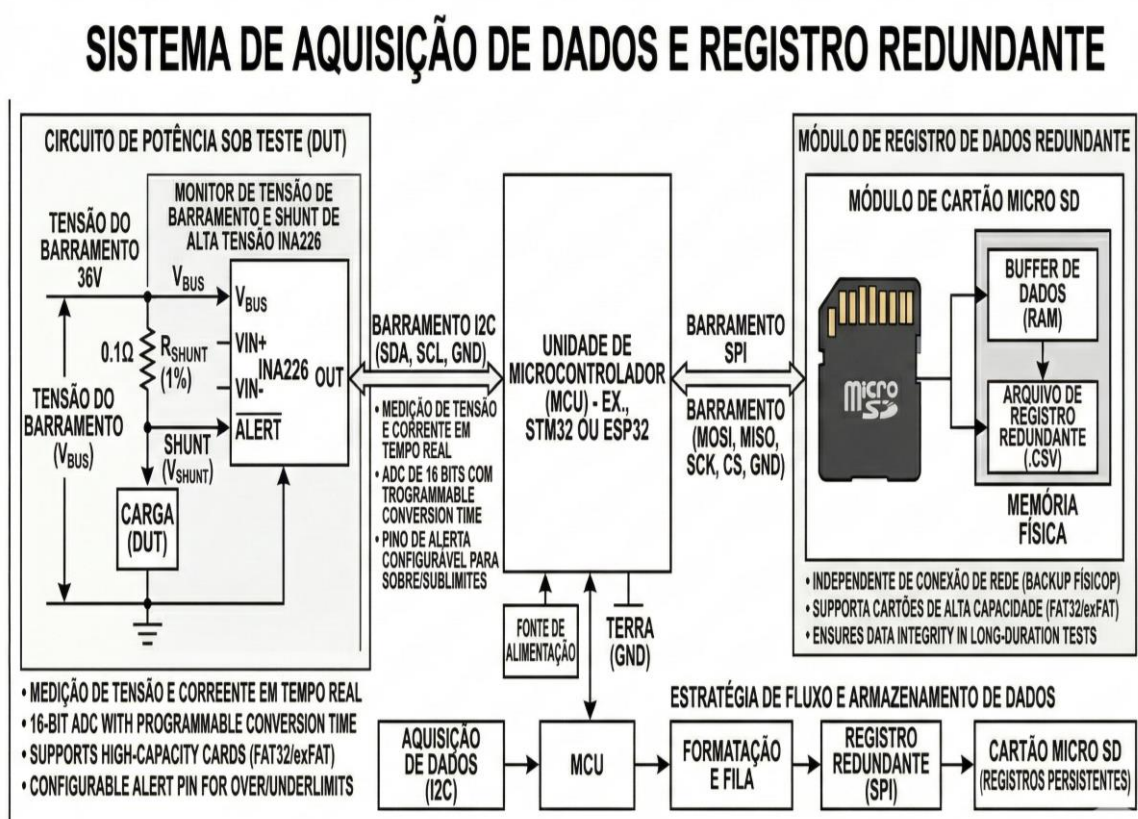
Diante dessa problemática, a inserção dos conceitos de Internet das Coisas (IoT) e sistemas embarcados de código aberto surge como uma alternativa disruptiva para a automação industrial e para a engenharia de testes. A literatura técnica atual aponta que a telemetria baseada em IoT permite não apenas a descentralização da coleta de dados, mas também a transformação de sensores isolados em nós inteligentes conectados a redes globais. No entanto, o desafio científico reside no desenvolvimento de arquiteturas de hardware e software que garantam alta precisão de leitura em ambientes ruidosos (característicos de bancadas elétricas), aliadas a

uma infraestrutura de comunicação robusta, redundante e de baixa latência.

O presente artigo direciona-se exatamente para essa lacuna, propondo o desenvolvimento e a implementação de um sistema de aquisição de dados microcontrolado de baixo custo e alta confiabilidade, focado no monitoramento de faróis automotivos. A solução, fundamenta-se no uso do microcontrolador ESP32, integrado ao sensor de potência de alta precisão INA226. - vide Figura 2 - Para a aquisição das grandezas elétricas de operação dos faróis, utilizou-se o sensor de potência integrado INA226, cuja comunicação nativa ocorre por meio do barramento bidirecional I2C (CULLEN, 2015).

O direcionamento do fluxo de informações adota o protocolo de comunicação MQTT acoplado à estruturação de dados em formato JSON, viabilizando o acompanhamento remoto e em tempo real por meio de painéis de telemetria e sistemas de alertas automatizados via Telegram. Complementarmente, o projeto aborda a segurança e a integridade da informação em ensaios prolongados por meio de uma estratégia de registro local redundante em cartão Micro SD, mitigando os riscos associados a falhas eventuais de conectividade com a rede local.

Figura 2: Esquemático do sistema de monitoramento de potência de alta precisão e registro de dados redundante com INA226 e micro SD.



Fonte: Própria Autoria (2026).

2. Fundamentação teórica

Neste capítulo, apresenta-se a sustentação teórica que fundamenta o desenvolvimento do sistema de aquisição de dados. São abordados os conceitos de Internet das Coisas (IoT) aplicados à telemetria, as características técnicas e vantagens do microcontrolador ESP32, o funcionamento do protocolo de comunicação MQTT e a estrutura de dados em formato JSON, além dos princípios de medição de potência elétrica e armazenamento local redundante.

2.1 Internet das Coisas (IoT) e Telemetria Industrial

A evolução dos sistemas de monitoramento industrial permitiu que variáveis físicas fossem digitalizadas e transmitidas em tempo real por meio de arquiteturas baseadas na Internet das Coisas (IoT) (SARKER et al., 2019).

A Internet das Coisas (IoT) refere-se a uma rede de objetos físicos equipados com sensores, software e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas via internet. Na engenharia de testes e na automação industrial, a integração da IoT transformou os métodos tradicionais de supervisão, permitindo que equipamentos isolados passem a operar como nós inteligentes de uma rede centralizada.

A aplicação da IoT no monitoramento de variáveis físicas é denominada telemetria. A telemetria consiste na coleta de dados em pontos remotos e no seu envio automático a uma central de processamento para fins de monitoramento, registro e análise (BALBINOT; BRUSAMARELLO, 2019).

Em ensaios laboratoriais de longa duração, como os realizados em componentes automotivos, a telemetria baseada em IoT oferece vantagens significativas em relação aos métodos tradicionais:

- **Continuidade na Coleta:** Elimina a necessidade de medições manuais espaçadas no tempo, garantindo uma amostragem contínua das variáveis elétricas e térmicas;
- **Monitoramento Remoto:** Permite que a equipe de engenharia acompanhe o comportamento do dispositivo em teste em tempo real, sem a necessidade de deslocamento físico até a bancada de ensaios;
- **Segurança Operacional:** Viabiliza a programação de alertas automáticos para interrupção do teste ou notificação dos operadores caso os limites de corrente ou temperatura sejam ultrapassados.

2.2 Microcontrolador ESP32 e Ambiente de Desenvolvimento

O núcleo de processamento e comunicação do protótipo desenvolvido baseia-se no microcontrolador ESP32,- Figura 3 - um sistema em um único chip (*System-on-a-Chip* - SoC) de baixo custo e consumo de energia reduzido, projetado pela *Espressif Systems*. O ESP32 foi selecionado para este projeto devido à sua alta performance e à integração nativa de conectividade sem fio (Wi-Fi e Bluetooth), o que simplifica o projeto de hardware ao eliminar a necessidade de módulos periféricos de rede (TEXAS INSTRUMENTS, 2015).

As principais características técnicas do ESP32 que justificam sua aplicação neste sistema de aquisição de dados incluem:

- **Processador:** Microprocessador de dois núcleos (*dual-core*) Xtensa 32-bit LX6, operando com frequência de clock ajustável de até 240 MHz, o que garante velocidade suficiente para processar leituras de múltiplos sensores simultaneamente;
- **Memória:** Possui 520 KB de memória SRAM interna para armazenamento de dados em tempo de execução, além de suporte a memória Flash externa para o código do programa;
- **Periféricos de Hardware:** Dispõe de pinos de Entrada/Saída de Propósito Geral (GPIO), conversores Analógico-Digital (ADC), conversores Digital-Analógico (DAC) e interfaces de comunicação serial como SPI, I2C e UART;
- **Conectividade:** Wi-Fi integrado padrão 802.11 b/g/n, essencial para a publicação dos dados na nuvem através do protocolo MQTT.

O desenvolvimento do software do microcontrolador foi realizado utilizando a IDE Arduino (*Integrated Development Environment*). Esse ambiente de desenvolvimento adota uma linguagem baseada em C/C++ e destaca-se por sua ampla comunidade e disponibilidade de bibliotecas prontas. No contexto deste projeto, o uso da IDE Arduino facilitou a integração das bibliotecas necessárias para a comunicação Wi-Fi, conexão com o servidor MQTT, manipulação de arquivos no cartão Micro SD e leitura direta dos sensores térmicos e de potência elétricos.

Figura 3: Fluxograma do funcionamento do sistema Wi-Fi com o ESP32.



Fonte: Própria Autoria (2026).

2.3 Protocolo MQTT e Estruturação de Dados em JSON

Para viabilizar a transmissão remota dos dados coletados em tempo real de forma eficiente, a arquitetura de software utiliza o protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*). Segundo Hillar (2017) o MQTT é um protocolo de mensagens leve, projetado especificamente para cenários de Internet das Coisas (IoT) nos quais a largura de banda da rede pode ser limitada e os dispositivos possuem recursos de processamento restritos.

Diferente do modelo tradicional de comunicação cliente-servidor (como o HTTP), o MQTT baseia-se no modelo de **Publicação e Assinatura** (*Publish/Subscribe*). A organização OASIS (2019) destaca que esse modelo intermediado envolve três elementos principais:

- **Publisher (Publicador):** No contexto deste projeto, é o microcontrolador ESP32, que realiza a leitura dos sensores e envia as informações para a rede;
- **Broker (Intermediador):** É o servidor central responsável por receber as mensagens dos publicadores, filtrá-las e encaminhá-las para os destinatários corretos. O projeto utiliza o *broker* EMQX (broker.emqx.io);
- **Subscriber (Assinante):** São as aplicações ou dispositivos que se conectam ao *broker* para receber os dados históricos e em tempo real (como o painel de telemetria ou o sistema de alertas do Telegram).

A comunicação ocorre por meio de **Tópicos**, que funcionam como caminhos ou canais de texto estruturados de forma hierárquica (OASIS, 2019). O ESP32 publica

os dados em tópicos específicos e o *broker* distribui essa informação apenas para os sistemas que assinaram esses mesmos tópicos, otimizando o tráfego de rede.

Para que as múltiplas variáveis lidas pelo sistema (tensão, corrente, potência e temperatura) sejam transmitidas de forma organizada em uma única mensagem, os dados são convertidos e estruturados no formato **JSON** (*JavaScript Object Notation*). O JSON é um padrão de formatação de dados leve e de fácil leitura, tanto para seres humanos quanto para máquinas, que utiliza uma estrutura de "chave e valor" (ECMA INTERNATIONAL, 2017). No protótipo, o ESP32 monta um pacote JSON contendo todas as leituras agrupadas antes de enviá-lo via MQTT, garantindo que os dados de um mesmo instante de tempo cheguem perfeitamente sincronizados ao servidor.

2.4 Medição de Potência Elétrica e Armazenamento Local Redundante

A precisão na medição das grandezas elétricas do farol automotivo é garantida pela integração do sensor de potência **INA226**. Este circuito integrado de alta precisão é capaz de monitorar tanto a tensão de barramento quanto a queda de tensão sobre um resistor de calibração externo, conhecido como resistor *Shunt*. A partir dessas duas leituras, o sensor calcula internamente a corrente elétrica circulante e a potência dissipada pela carga em teste (TEXAS INSTRUMENTS, 2015).

A comunicação entre o sensor INA226 e o microcontrolador ESP32 é realizada através do protocolo I2C (*Inter-Integrated Circuit*). De acordo com Balbinot e Brusamarello (2019) o protocolo I2C é um barramento de comunicação serial síncrono que utiliza uma arquitetura do tipo mestre-escravo, sendo amplamente adotado na engenharia eletrônica para interconexão de circuitos integrados em curtas distâncias. A principal vantagem técnica desse protocolo reside na otimização do hardware, uma vez que a comunicação bidirecional entre o microcontrolador central (mestre) e os múltiplos sensores periféricos (escravos) é realizada utilizando apenas duas linhas físicas de sinal (NXP SEMICONDUCTORS, 2021):

- **SDA (Serial Data):** Linha responsável pelo tráfego bidirecional dos dados e comandos de leitura entre o ESP32 e os sensores;
- **SCL (Serial Clock):** Linha dedicada à transmissão do sinal de clock, responsável pelo sincronismo temporal de toda a comunicação do barramento.

Cada dispositivo conectado a essa rede possui um endereço binário exclusivo configurado de fábrica. Dessa forma, o microcontrolador ESP32 gerencia as requisições enviando o endereço específico do componente com o qual deseja se comunicar. Essa característica de compartilhamento de barramento elimina a

necessidade de circuitos complexos de chaveamento de sinal ou o uso de pinos adicionais no microcontrolador para cada novo sensor inserido, justificando a viabilidade de conexão direta do sensor de potência INA226 e dos sensores térmicos ao sistema embarcado.

Complementarmente, ensaios laboratoriais automotivos de longa duração estão sujeitos a instabilidades externas, como quedas temporárias de energia ou oscilações no sinal da rede Wi-Fi local. Se o sistema dependesse exclusivamente da transmissão em nuvem, essas falhas resultariam na perda irreparável do histórico de dados do teste.

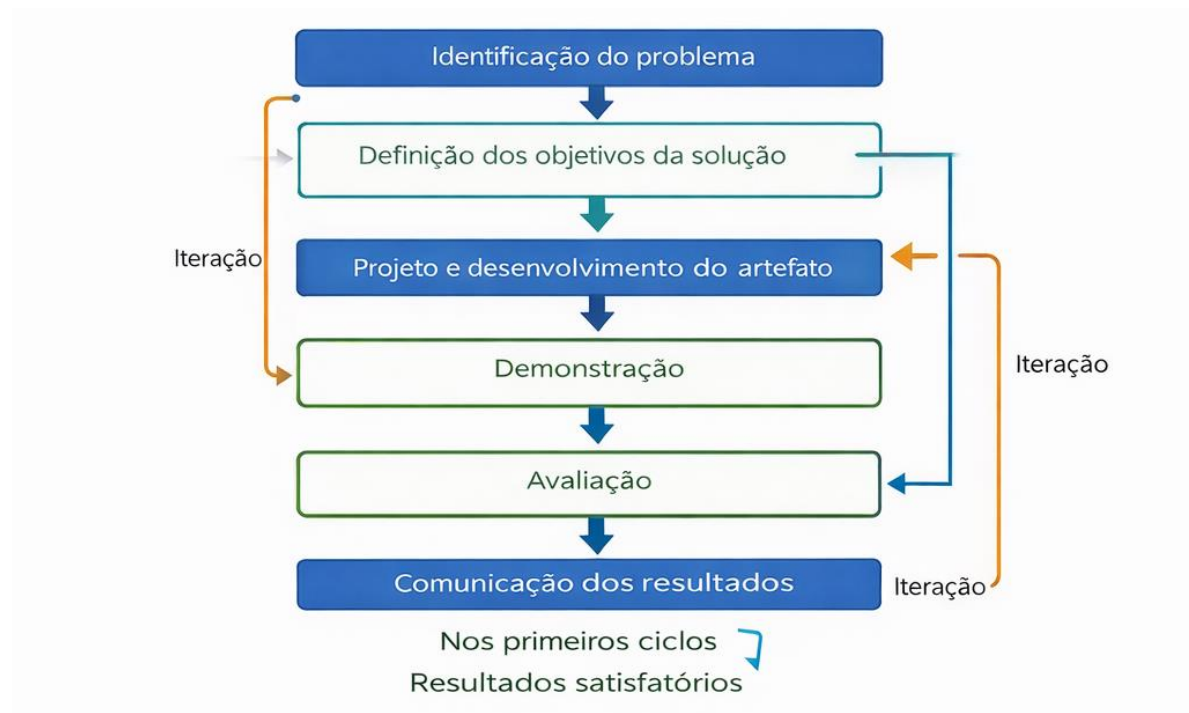
Para mitigar esse risco, o projeto implementa uma estratégia de **armazenamento local redundante** através de um módulo de cartão Micro SD conectado ao ESP32 via barramento de comunicação **SPI** (*Serial Peripheral Interface*). Conforme detalhado por Tocci, Widmer e Moss (2019) o acionamento e a seleção do módulo de memória são controlados diretamente pelo pino digital **D4** do microcontrolador, que atua como a linha de *Chip Select* (CS). Dessa forma, sempre que o ESP32 realiza a leitura das variáveis de tensão, corrente e temperatura, ele executa duas ações simultâneas: publica o pacote JSON na nuvem via MQTT e grava uma nova linha de registro em um arquivo de texto no formato .csv ou .txt dentro do cartão de memória física. Isso garante a integridade total do histórico do ensaio, mesmo em caso de desconexão completa da rede.

3. Metodologia

A condução deste artigo fundamenta-se na abordagem metodológica *Design Science Research* (DSR), esboçada na Figura 4. Diferente das ciências tradicionais, que buscam compreender e explicar a realidade existente, a DSR é uma metodologia voltada para as ciências do artificial e para a engenharia, cujo objetivo principal é o desenvolvimento, a construção e a validação de um artefato tecnológico projetado para solucionar um problema prático e real de mercado (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015).

No contexto deste artigo, o artefato desenvolvido consiste em uma instanciação física e lógica: um protótipo funcional de um sistema de telemetria IoT térmica e elétrica. Para garantir o rigor científico exigido, o desenvolvimento do sistema seguiu rigorosamente as etapas do ciclo metodológico da DSR propostas por Peffers *et al.* (2007), descritas a seguir:

Figura 4: Fluxograma da metodologia Design Science Research (DSR).



Fonte: Própria Autoria (2026).

3.1 Identificação do Problema e Motivação

A primeira etapa consistiu no mapeamento das dificuldades encontradas na engenharia de testes automotivos. Identificou-se a necessidade de automatizar ensaios de fadiga e durabilidade de longa duração em faróis. O método tradicional baseado em medições manuais espaçadas ou em equipamentos de medição locais e isolados consome excessivo tempo das equipes técnicas, impossibilita o acompanhamento em tempo real fora do horário comercial e eleva o risco de perda de dados históricos caso ocorram falhas elétricas na bancada de ensaios.

3.2 Definição dos Requisitos do Objetivo

A partir do problema identificado, foram estabelecidos os requisitos técnicos e as metas que o protótipo deveria atingir para ser considerado uma solução eficaz. Determinou-se que o sistema deveria:

- Capturar dados elétricos de tensão e corrente de forma contínua com alta precisão;
- Coletar o perfil térmico do farol por meio de sensores de temperatura digitais;
- Processar e transmitir as informações em tempo real via rede sem fio, sem o uso de cabeamento estruturado complexo;
- Garantir a segurança dos dados por meio de um mecanismo físico de

salvamento local;

- Emitir alertas automáticos à distância em caso de variações críticas nos parâmetros do ensaio.

3.3 Projeto e Desenvolvimento do Artefato

Esta etapa compreendeu a Engenharia de Hardware e Software do protótipo. O projeto de hardware envolveu a seleção e a conexão direta do microcontrolador ESP32 ao sensor de potência INA226 e aos sensores térmicos digitais através do barramento de comunicação I²C, eliminando circuitos complexos de chaveamento e multiplexação de sinais. Também foi integrado o circuito de leitura e gravação para o módulo de cartão Micro SD conectado ao barramento SPI (pino D4).

Na engenharia de software, desenvolveu-se o firmware na IDE Arduino utilizando a linguagem C/C++. O algoritmo foi programado para estruturar as leituras no formato JSON, realizar a gravação simultânea no cartão de memória e gerenciar a publicação das mensagens via protocolo MQTT direcionadas ao servidor central EMQX, além de integrar o sistema de notificações automatizadas por meio da API do *Telegram*.

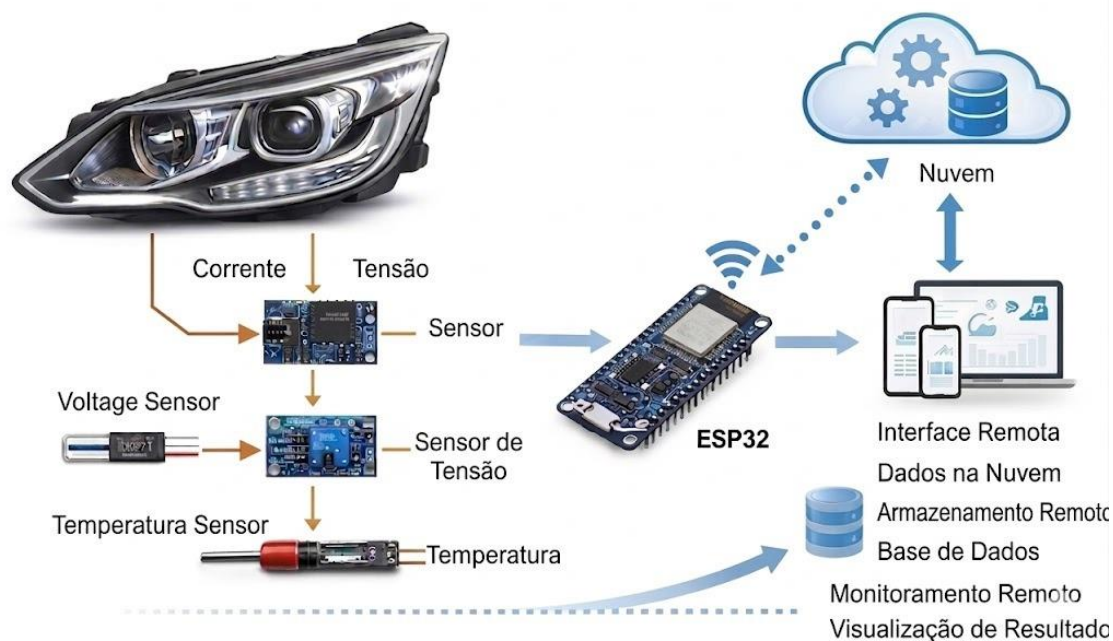
3.4 Demonstração Funcional e Testes

Com o protótipo montado, realizou-se uma demonstração prática em ambiente laboratorial simulado. O sistema foi integrado a uma bancada de testes composta por uma fonte de alimentação regulável e um farol automotivo real atuando como carga ativa. O ensaio simulou uma rotina real de teste de durabilidade, permitindo verificar a capacidade do ESP32 de manter a aquisição de dados e o envio contínuo das mensagens de forma estável durante um período prolongado de funcionamento.

3.5 Avaliação do Sistema

A última etapa do ciclo da DSR envolveu a análise dos resultados obtidos frente aos objetivos estabelecidos. Como demonstrado as etapas na Figura 5, avaliou-se a integridade e o sincronismo dos pacotes de dados JSON recebidos no *broker* em nuvem, a persistência e a formatação correta do arquivo de log gerado no cartão Micro SD e a eficácia do tempo de entrega dos alertas de status enviados ao smartphone dos operadores. O desempenho elétrico e a estabilidade do sinal foram validados sob a condição de aterramento comum (GND) unificado, garantindo o funcionamento do sistema livre de ruídos de bancada.

Figura 5: Arquitetura proposta para o teste do sistema de aquisição de dados.

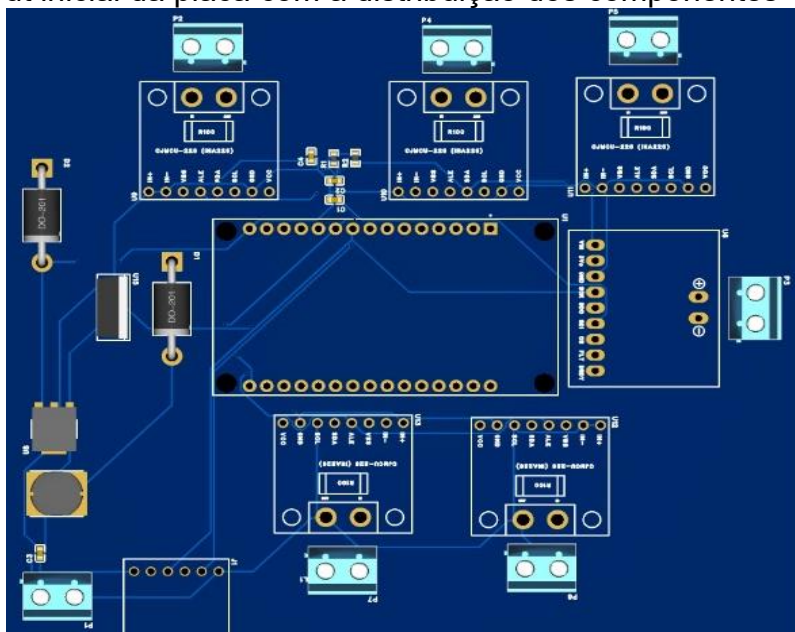


Fonte: Própria Autoria (2026).

4. Desenvolvimento do projeto

Neste capítulo, é detalhado o desenvolvimento prático do artefato tecnológico, abrangendo desde a concepção do circuito elétrico e layout da placa – Figura 6 - até a lógica de programação criada para a captura, armazenamento e transmissão das variáveis. A arquitetura física foi planejada para operar de forma robusta em ambiente laboratorial, garantindo a integridade dos sinais gerados pelos sensores.

Figura 6: Layout inicial da placa com a distribuição dos componentes



Fonte: Própria Autoria (2026).

4.1 Infraestrutura Elétrica, Circuito e Conexões

A primeira etapa do desenvolvimento prático consistiu no projeto da infraestrutura elétrica de alimentação e na interconexão dos módulos. A infraestrutura do sistema de testes foi concebida com base no esquema elétrico representado na Figura 7. A concepção do circuito priorizou o correto dimensionamento e a especificação dos componentes eletroeletrônicos, os quais foram selecionados mediante análise técnica de suas curvas de operação, garantindo que o hardware opere com total segurança dentro dos limites nominais de corrente e tensão do ensaio. Complementarmente, o projeto contemplou a otimização do layout e da disposição espacial dos componentes na placa, bem como o cálculo bitolar adequado dos condutores responsáveis pelas interligações, mitigando perdas elétricas e assegurando a integridade dos sinais medidos, em conformidade com as diretrizes da NBR 5410 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2004).

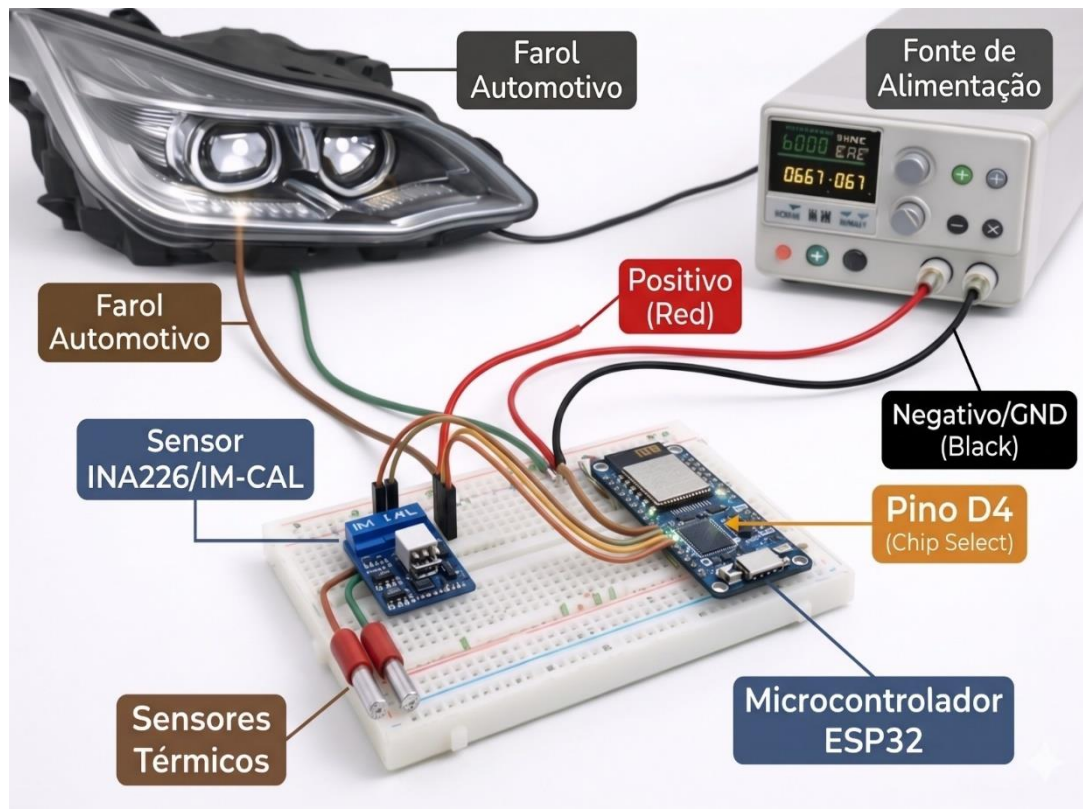
Durante a fase inicial de testes em bancada, identificou-se um desafio técnico crítico: a ocorrência de flutuações de sinal e instabilidade no reconhecimento dos módulos periféricos (o sensor de potência INA226 e os sensores térmicos). A solução de engenharia aplicada consistiu na unificação dos referenciais elétricos, estabelecendo um GND comum (aterramento unificado) entre a fonte de alimentação externa de corrente contínua (responsável por alimentar o farol automotivo) e o microcontrolador ESP32. Sem essa interconexão de loops de terra, o barramento de dados perde a referência de tensão zero volt, gerando ruídos eletromagnéticos que

Figura 7: Esquema elétrico do Aquisitor de dados.



14

Figura 8: Ilustração da montagem experimental do sistema de aquisição de dados.



Fonte: Própria Autoria (2026).

4.2 Lógica de Seleção de Sinais e Arquitetura de Hardware

Na concepção inicial do projeto, previu-se a utilização de cinco sensores INA226 operando em paralelo e acoplados a um circuito integrado multiplexador, cuja função seria chavear sequencialmente os canais de comunicação para que o microcontrolador pudesse realizar as leituras de energia de cada ponto individualmente – vide Figura 6 -. No entanto, durante a fase de análise de viabilidade e desenvolvimento do hardware, essa abordagem foi descontinuada, reduzindo a estrutura para a utilização de apenas um único sensor INA226 conectado diretamente ao barramento I2C do ESP32. A eliminação do multiplexador e a redução no número de sensores justificaram-se por critérios de otimização de engenharia, a arquitetura nativa do protocolo I2C já permite a comunicação direta com o microcontrolador utilizando apenas duas linhas físicas (SDA e SCL). Ao retirar os quatro sensores excedentes e o circuito integrado de multiplexação, o grupo obteve uma simplificação drástica no layout da Placa de Circuito Impresso (PCB), representado na Figura 9, reduzindo drasticamente o custo final do protótipo, diminuindo a quantidade de conexões físicas sujeitas a maus contatos ou falhas de solda, e simplificando também a lógica de programação para o controle de chaveamento de hardware, tornando o

sistema de aquisição de dados final muito mais limpo, robusto e eficiente para o ensaio.

É importante ressaltar que a redução para um único sensor INA226 delimitou o escopo analítico do protótipo nesta fase de desenvolvimento. Enquanto o circuito inicial com cinco sensores permitiria monitorar simultaneamente as correntes de múltiplas funções do farol de forma individualizada (como o farol alto, baixo, lanterna e indicadores de direção), o arranjo atual restringiu a capacidade de medição a uma única função por ensaio. Contudo, para os objetivos propostos de validação da telemetria, estabilidade de sinal e redundância de armazenamento, a utilização de apenas um canal mostrou-se plenamente suficiente, deixando a expansão para múltiplos canais simultâneos como uma recomendação de melhoria para trabalhos futuros.

Com a eliminação de componentes intermediários de chaveamento (como multiplexadores), a arquitetura de hardware foi simplificada, aproveitando ao máximo a capacidade nativa de comunicação do ESP32. O sensor de potência INA226 e os sensores térmicos digitais foram configurados para compartilhar diretamente as mesmas linhas físicas do barramento I2C (pinos SDA e SCL do microcontrolador).

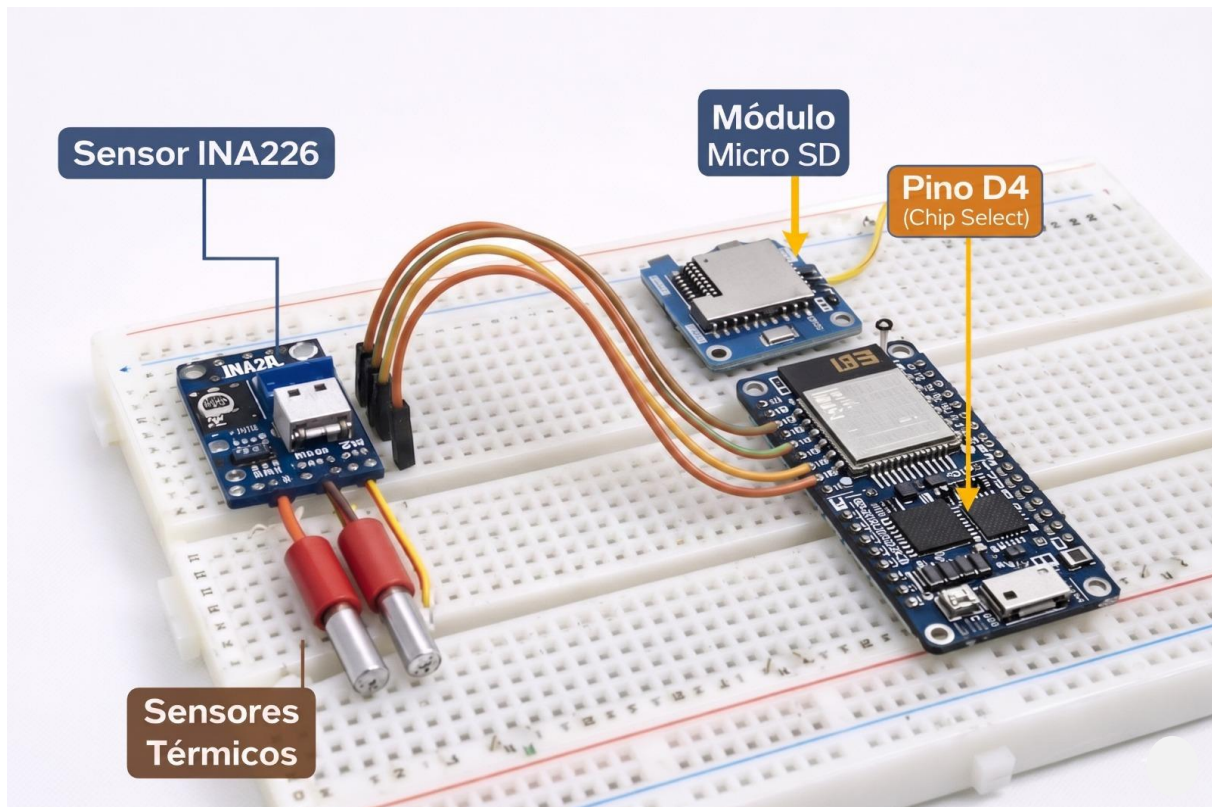
Como o protocolo I2C trabalha com endereçamentos lógicos exclusivos na memória, o firmware do ESP32 realiza a varredura e faz a requisição direta de dados para cada sensor de forma sequencial na mesma linha física. Essa abordagem trouxe duas vantagens importantes para o projeto:

- **Redução de Pontos de Falha:** Menos conexões físicas e componentes soldados diminuem o risco de mau contato ou quebra de trilhas devido à vibração ou calor da bancada;
- **Economia de Pinos do ESP32:** Ao concentrar os dados nos pinos padrão I2C, as demais portas digitais de Propósito Geral (GPIO) do microcontrolador ficaram totalmente livres para outras funções essenciais.

Dentre os pinos digitais liberados, destaca-se o uso do pino D4, que foi configurado via software para atuar estritamente como a linha de *Chip Select* (CS) do barramento SPI dedicado ao módulo do cartão Micro SD. A linha de *Chip Select* funciona como um interruptor lógico: sempre que o ESP32 precisa salvar um log de dados fisicamente, ele altera o estado elétrico do pino D4 para nível lógico baixo, "avisando" o cartão SD de que ele deve escutar as próximas informações enviadas e ignorar os demais componentes da placa. Assim que a gravação do arquivo de texto é concluída, o pino retorna ao estado alto, finalizando o ciclo de escrita local de forma

segura e controlada.

Figura 9: Reestruturação dos componentes do protótipo.



Fonte: Própria Autoria (2026).

4.3 Integração de Software e Interface

A engenharia de software do projeto foi estruturada para garantir a execução ordenada de todas as tarefas de aquisição de dados, redundância local e comunicação remota, mitigando o travamento do sistema durante ensaios laboratoriais prolongados. O firmware desenvolvido para o microcontrolador ESP32 adota uma lógica de execução sequencial e temporizada por meio do método de temporização não-bloqueante (`millis()`), operando dentro do ciclo principal do programa (`void loop()`).

A temporização interna do firmware utiliza contadores baseados em registradores de tempo real de forma não-bloqueante, permitindo o paralelismo virtual de tarefas na CPU do microcontrolador (MONK, 2017).

No momento em que o sistema é energizado, a rotina de inicialização (`void setup()`) executa o auto-teste e a configuração inicial das portas de comunicação periférica. Nessa etapa, o ESP32 inicializa o barramento I2C para a detecção e comunicação com o sensor de potência INA226 e com os sensores térmicos.

Simultaneamente, abre-se o canal de comunicação SPI com o módulo de cartão Micro SD através do pino digital D4 (responsável pela ativação do sinal de *Chip Select*) e estabelece-se a conexão com a rede Wi-Fi local para a autenticação junto ao servidor de telemetria.

Uma vez concluída com sucesso a etapa de inicialização, o algoritmo passa a operar em regime contínuo dentro do laço principal. O fluxo lógico segmentado que governa o comportamento do firmware está representado nos trechos de código-fonte ilustrados pelas Figuras 10 e 11

Figura 10 – Trecho inicial do laço principal (void loop): gestão de conexão, sincronização de relógio e temporização.

```
// ===== 7. LOOP PRINCIPAL =====  
void loop() {  
    // --- A. GESTÃO DE CONEXÃO E RELÓGIO ---  
    if (WiFi.status() == WL_CONNECTED) {  
        if (!relogio_sincronizado) {  
            configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);  
            delay(2000);  
            relogio_sincronizado = true;  
            bot.sendMessage(chatId, "🤖 *Aquisitor Online*\nRelógio sincronizado. SD g  
            Serial.println("✅ Wi-Fi e Relógio Sincronizados!");  
        }  
        reconectarMQTT();  
        mqtt.loop();  
    } else {  
        relogio_sincronizado = false;  
    }  
  
    // --- B. LEITURA E GRAVAÇÃO (A cada 2 segundos) ---  
    if (millis() - ultimoTempoLeitura >= 2000) {  
        ultimoTempoLeitura = millis();  
  
        String data_hora_atual = obterDataHora();  
        String linhaCSV = data_hora_atual;  
        String jsonString = "{}"; // Inicia o pacote para a Nuvem
```

Fonte: Própria Autoria.

Figura 11 – Trecho complementar do laço principal: persistência no cartão SD, transmissão MQTT e lógica de varredura de alertas.

```
// 1. GRAVA NO CARTÃO SD
File arquivo = SD.open(nomeArquivo, FILE_APPEND);
if (arquivo) {
    arquivo.println(linhaCSV);
    arquivo.close();
    Serial.println("SD -> " + linhaCSV);
} else {
    Serial.println("⚠ Erro SD!");
}

// 2. ENVIA PARA A NUVEM E TELEGRAM
if (WiFi.status() == WL_CONNECTED) {
    if (mqtt.connected()) {
        mqtt.publish(mqtt_topic, jsonString.c_str());
    }
}

// 3. TELEGRAM INDIVIDUAL POR MÁQUINA
for (int i = 0; i < 10; i++) {
    bool peca_esta_ligada_agora = (correntes[i] >= 2.0); // Mais de 2mA

    if (peca_esta_ligada_agora && !peca_estava_ligada[i]) {
        String msg = "▶ *Máquina " + String(i+1) + " Ligada!* (" + String(i+1) + ")";
        bot.sendMessage(chatId, msg, "Markdown");
        peca_estava_ligada[i] = true;
    }
    else if (!peca_esta_ligada_agora && peca_estava_ligada[i]) {
        String msg = "⚠ *Máquina " + String(i+1) + " Desligada!*";
        bot.sendMessage(chatId, msg, "Markdown");
        peca_estava_ligada[i] = false;
    }
}
}
```

Fonte: Própria Autoria (2026).

Conforme observado na estrutura do algoritmo (Figura 10 e Figura 11), a lógica integrada de software executa de forma cíclica as seguintes rotinas:

- **Gestão de Conexão e Relógio (NTP):** O microcontrolador verifica constantemente o estado da rede sem fio (`WiFi.status()`). Caso o canal esteja ativo, o sistema valida se o relógio interno já foi sincronizado via servidor NTP (*Network Time Protocol*). Se negativo, realiza-se uma chamada de sincronização configurada para os parâmetros locais de fuso horário, disparando uma notificação de inicialização bem-sucedida ("Aquisitor Online") via API do Telegram. Adicionalmente, invocam-se as funções de manutenção da conexão com o *broker* MQTT (`reconnectMQTT()` e `mqtt.loop()`). Se o Wi-Fi for desconectado, o indicador lógico de sincronismo é desativado (`false`), mas o fluxo principal não é interrompido.
- **Varredura, Leitura Temporizada e Formatação:** Para evitar o uso de funções de atraso absoluto (`delay`), que travariam o microcontrolador, o algoritmo avalia a diferença de tempo através da função nativa `millis()`. A cada intervalo exato de 2 segundos (2000 ms), o sistema realiza a captura das variáveis físicas através do barramento I2C (tensão, corrente e temperatura), obtém a estampa de tempo (*timestamp*) atualizada do relógio e inicia em paralelo duas formatações distintas de dados: uma cadeia de caracteres em formato plano e separado por vírgulas (padrão CSV) voltada para o armazenamento físico local, e uma cadeia estruturada no formato JavaScript Object Notation (JSON), ideal para transmissões leves em rede por meio de pares "chave-valor".
Com as variáveis elétricas armazenadas em memória RAM, o firmware executa a serialização dos dados estruturando-os no formato leve *JavaScript Object Notation* (JSON) de modo a viabilizar um empacotamento otimizado para transmissões via rádio (OVERSMITH, 2020)
- **Persistência de Dados Local (Backup):** Priorizando a integridade das informações geradas, antes de iniciar o processo de transmissão externa via rádio, o ESP32 manipula a linha de controle do módulo Micro SD e abre o arquivo de log especificado com o parâmetro de anexação `FILE_APPEND` (Figura 11). A linha contendo os dados em formato CSV é gravada fisicamente e o arquivo é encerrado em seguida (`arquivo.close()`). Essa arquitetura isolada de gravação rápida protege o histórico contra perdas parciais decorrentes de quedas de energia na bancada.
- **Publicação via Protocolo MQTT:** Com a conexão de rede previamente

validada, o cliente MQTT integrado ao firmware assume o pacote JSON gerado e realiza a publicação do conteúdo no tópico associado dentro do servidor *broker* público EMQX (broker.emqx.io). Esse processo é executado em frações de segundo, disponibilizando instantaneamente a telemetria em tempo real para sistemas de monitoramento assinantes do tópico.

- **Verificação de Limites e Alertas Críticos (Interface Telegram):** O bloco final do programa executa uma estrutura de repetição (for) configurada para varrer os canais de ensaio instalados (Figura 11). O código avalia as leituras de corrente de forma individual e adota um limiar crítico de decisão estabelecido em 2.0 mA para discernir se a carga sob ensaio encontra-se operando ou inativa. Estruturas condicionais comparam o estado atual (*peca_esta_ligada_gra_agora*) com a memória do estado imediatamente anterior (*peca_estava_ligada*). Caso uma transição de estado seja detectada, o microcontrolador interrompe o fluxo genérico e aciona a interface do robô de alertas, gerando e enviando uma mensagem de texto com formatação *Markdown* (sinalizando mensagens automáticas de **"*Máquina Ligada! ***" ou **"*Máquina Desligada! ***") diretamente para o smartphone cadastrado da equipe de engenharia.

Essa arquitetura de software integrada elimina por completo a necessidade de supervisão humana constante na bancada de testes. O comportamento tolerante a falhas do firmware garante que, perante uma instabilidade temporária no Wi-Fi ou no servidor em nuvem, o dispositivo continue registrando rigorosamente os dados no cartão Micro SD de forma autônoma e execute tentativas de reconexão em segundo plano. Assim que a estabilidade da rede é reestabelecida, a telemetria remota via EMQX é normalizada sem que o ensaio laboratorial de fadiga precise sofrer interrupções ou reinicializações.

4. 4 Interface de Usuário e Telemetria

Para a visualização remota das variáveis elétricas em tempo real, desenvolveu-se uma interface gráfica em plataforma móvel integrada ao *broker* MQTT. A Figura 12 ilustra a tela de parametrização do painel (*Edit Panel*), evidenciando o vínculo direto com a arquitetura de software descrita anteriormente. Observa-se que a interface foi configurada para interpretar a carga útil em formato JSON (Payload is JSON Data), utilizando a sintaxe de filtragem \$.corrente_mA (*JsonPath*) para extrair e plotar exclusivamente os valores instantâneos de corrente com precisão de três casas

decimais. Adicionalmente, definiu-se o nível de Qualidade de Serviço em QoS 0, garantindo o recebimento dinâmico dos pacotes de telemetria com baixa latência e otimização do fluxo de dados no dispositivo supervisor.

A escolha do protocolo de comunicação MQTT deu-se em função de sua arquitetura baseada no modelo de publicação e assinatura (*publish/subscribe*), o qual reduz drasticamente a sobrecarga de rede (*overhead*) em comparação com protocolos tradicionais de requisição baseados em HTTP (SANTOS; SOUZA, 2024)

Figura:12 – Tela de configuração aplicativo MQTT-Dash.

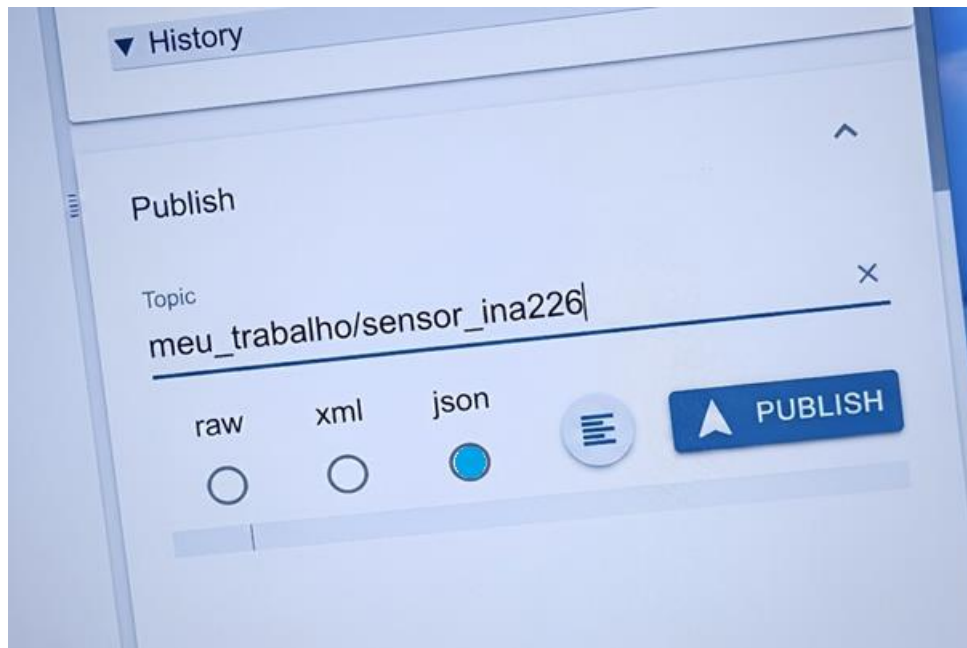
The screenshot shows the 'Edit Panel' configuration screen for MQTT-Dash. At the top, there's a status bar with the time 8:09 and various icons. Below the title 'Edit Panel', there are two input fields: 'Factor' with the value '1' and 'Decimal precision' with the value '3'. A 'Graph color' section shows a red circle icon and the hex code '#ea1111'. Below this are several checkboxes: 'Show plot area' (unchecked), 'Show points and tooltip' (unchecked), 'Enable notification or alarm' (unchecked), and 'Payload is JSON Data' (checked). There's a 'JsonPath for subscribe *' field with the value '\$.corrente_mA'. An 'Add more graph' button with a '+' icon is present. Below that, 'Smooth curve' is checked, and 'Max persistence' is set to '10'. The 'Max duration' is set to 'None'. At the bottom, the 'QoS' is set to '0'. There are 'CANCEL' and 'SAVE' buttons, and a Google Wallet 'Abrir' button at the very bottom.

Fonte: Própria Autoria (2026).

Visando assegurar a integridade e a correta rota de tráfego dos dados de telemetria entre o hardware e o ambiente em nuvem, realizou-se a homologação dos canais de comunicação utilizando um cliente MQTT em ambiente desktop. A Figura 13 ilustra a interface de testes no momento da parametrização do canal, na qual estabeleceu-se o tópico estruturado meu_trabalho/sensor_ina226. Essa etapa de

validação em bancada permitiu simular o envio manual de cargas úteis (*payloads*) em diferentes formatos — como texto plano (*raw*) e JSON —, certificando que o *broker* EMQX realizava o roteamento e a distribuição das mensagens de maneira fidedigna antes da integração final com a interface gráfica definitiva

Figura 13: Tela de ferramenta de testes e monitoramento de protocolo MQTT.



Fonte: Própria Autoria (2026).

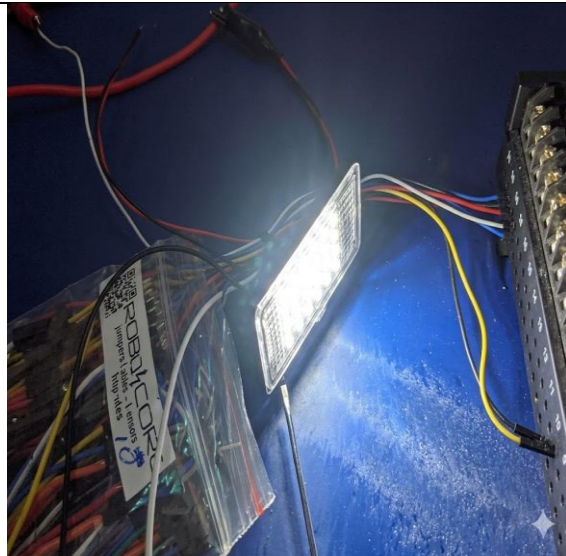
5 Análise e discussão de resultados

Neste capítulo, apresentam-se os resultados obtidos a partir dos ensaios práticos realizados com o protótipo funcional do sistema de aquisição de dados. O objetivo dos testes em ambiente laboratorial simulado foi validar a precisão das leituras elétricas e térmicas, a estabilidade da transmissão sem fios e a eficácia das rotinas de segurança programadas no firmware do ESP32.

5.1 Validação Experimental na Bancada de Testes

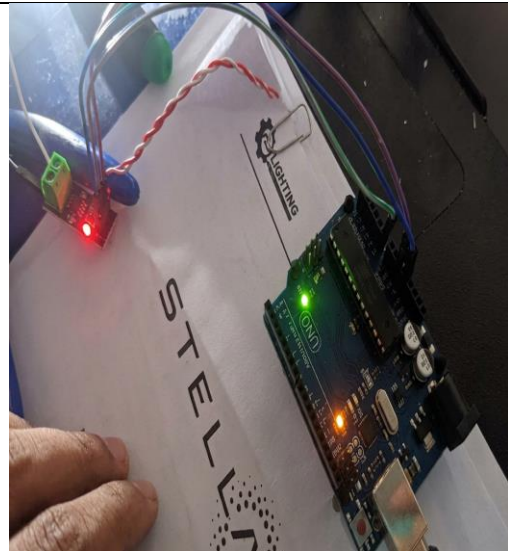
Para a realização dos ensaios de validação, o circuito eletrônico desenvolvido foi conectado a uma fonte de bancada regulável estabilizada e a um farol de automóvel real (carga ativa), conforme ilustrado nas Figuras 14 e 15. O sistema foi submetido a um teste de funcionamento contínuo, monitorizando o acendimento do farol e simulando um ensaio de fadiga laboratorial.

Figura 14: Exemplar de farol para teste.



Fonte: Própria Autoria (2026).

Figura15 – Circuito eletrônico.



Fonte: Própria Autoria (2026).

Durante as primeiras horas de ensaio, monitorou-se o comportamento térmico e elétrico do conjunto. Os sensores térmicos fixados no corpo do farol registraram a curva de elevação da temperatura desde a temperatura ambiente inicial até a estabilização térmica do componente. O sensor de potência INA226 operou de forma estável, efetuando as medições de tensão de barramento e corrente elétrica circulante sem apresentar desvios nas leituras, como evidenciado nas Figuras 16 e 17..

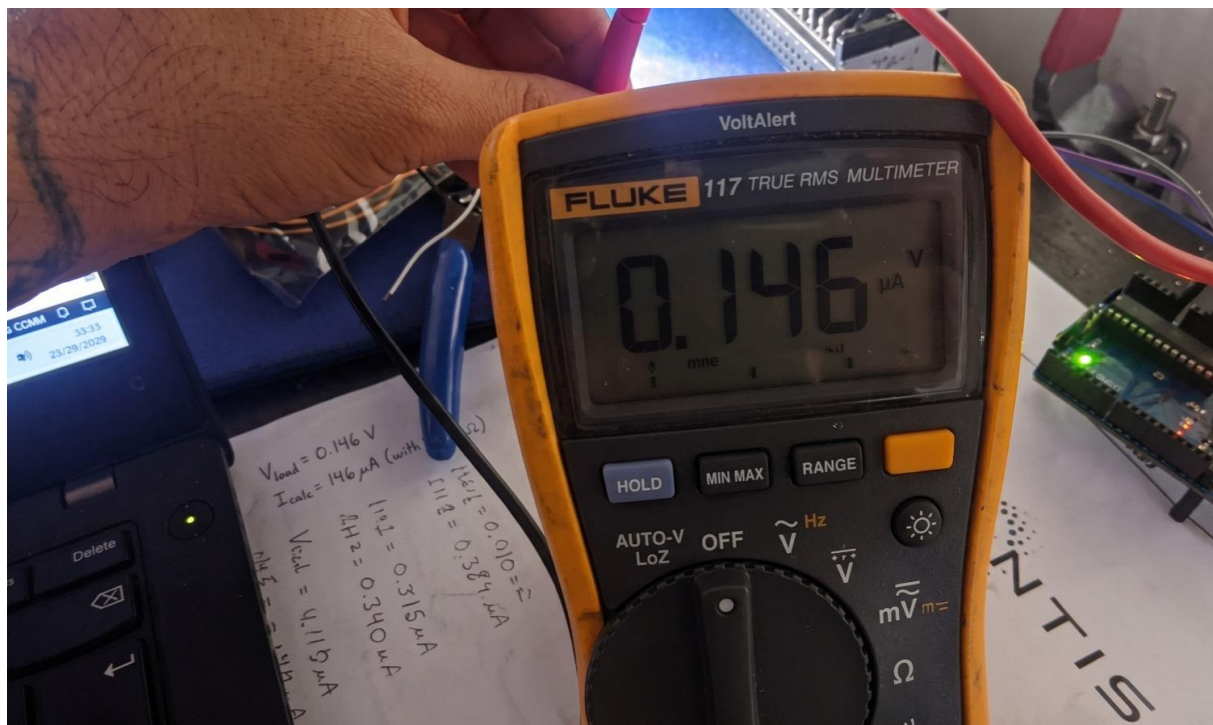
A decisão de engenharia de interligar o referencial negativo (GND comum) da fonte de alimentação do farol ao GND do microcontrolador ESP32 provou ser fundamental. Nos testes em que os referenciais foram mantidos isolados, o barramento de dados I2C apresentou instabilidade e falhas de comunicação frequentes. Com a unificação do ponto de terra, o ruído eletromagnético da bancada foi totalmente mitigado, garantindo um sinal limpo e um índice de erro de leitura nulo nas transmissões entre o sensor INA226 e o microcontrolador.

Figura 16: Valores medidos pelo INA226.

```
Corrente: 0.149 A
Corrente: 0.146 A
Corrente: 0.151 A
Corrente: 0.156 A
Corrente: 0.148 A
Corrente: 0.150 A
Corrente: 0.153 A
Corrente: 0.138 A
```

Fonte: Própria Autoria (2026).

Figura 17: Validação da leitura do INA226 com o Multímetro.



Fonte: Própria Autoria (2026).

5.2 Avaliação do Armazenamento Local e Transmissão MQTT

A rotina de backup e redundância local de dados através do módulo de cartão Micro SD foi validada com sucesso. Verificou-se que a lógica do pino D4, atuando como *Chip Select* (CS) do barramento, executou a abertura, escrita e fechamento dos arquivos de texto de forma precisa a cada ciclo de amostragem. O arquivo gerado no formato estruturado apresentou todas as variáveis temporalmente alinhadas, contendo o histórico completo do ensaio pronto para ser exportado para ferramentas de análise gráfica.

Simultaneamente à gravação local, a transmissão sem fios através do protocolo MQTT cumpriu integralmente os requisitos estabelecidos. O cliente embutido no ESP32 realizou a conversão das variáveis para pacotes de texto no formato JSON e publicou-os de forma contínua no *broker* público EMQX (broker.emqx.io).

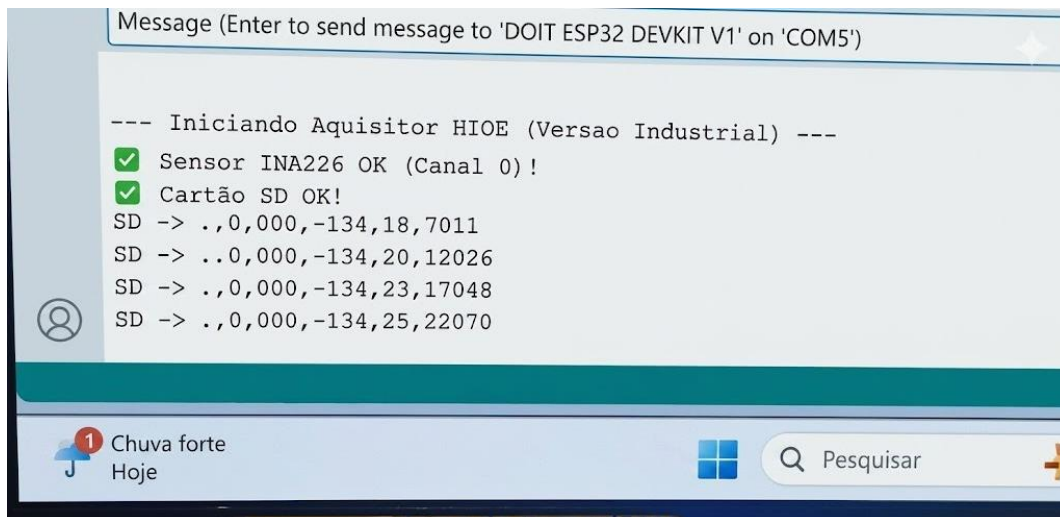
Para testar a robustez do sistema face a falhas de infraestrutura de rede (situação comum em ensaios industriais longos), simulou-se uma desconexão intencional do router Wi-Fi durante o teste. O comportamento do software ocorreu exatamente como projetado:

- O ESP32 detectou a perda de sinal e manteve o armazenamento das leituras de forma isolada e segura no cartão Micro SD, evitando qualquer perda de informação;
- O algoritmo executou tentativas automáticas de reconexão em segundo plano;
- Assim que o sinal de Wi-Fi foi restabelecido, o microcontrolador restabeleceu a ligação com o *broker* MQTT e retomou a transmissão remota em tempo real instantaneamente.

A Figura 18 apresenta o **Monitor Serial da IDE Arduino** (conectado ao ESP32 via porta COM5), comprovando o sucesso do **auto-teste de inicialização e a gravação de dados** em tempo real. Em suma, ela demonstra:

- **Validação do Hardware:** Confirmação de que o sensor INA226 (Canal 0) e o módulo de Cartão SD foram detectados e estão operacionais (OK!).
- **Persistência de Dados Local:** Exibição das linhas de telemetria sendo estruturadas e salvas com sucesso no formato CSV dentro do cartão de memória (SD -> ...).

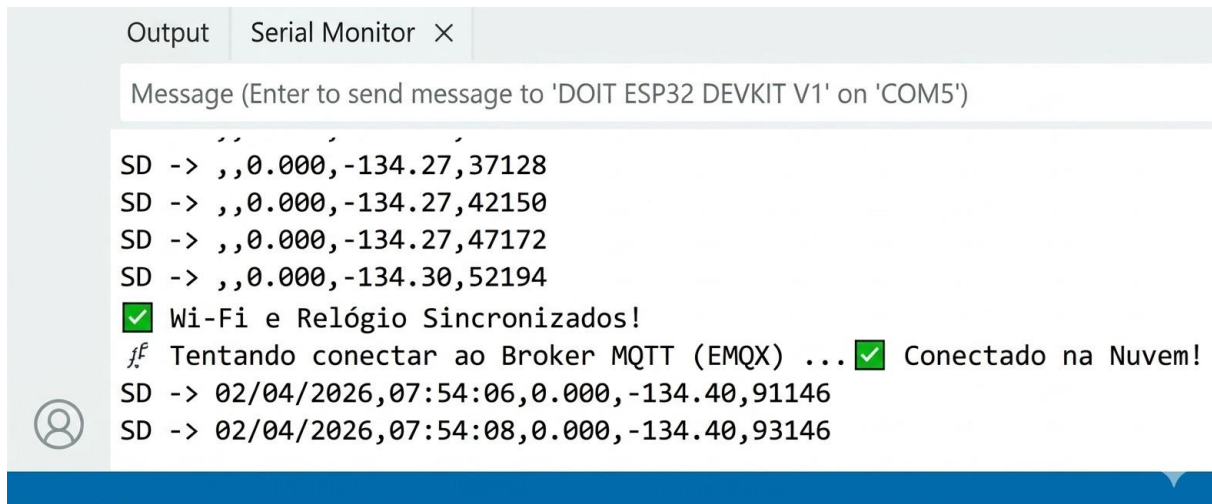
Figura 18: Auto-teste de inicialização e gravação de dados.



Fonte: Própria Autoria (2026).

A Figura 19 apresenta o log de depuração gerado no Monitor Serial da IDE Arduino, comprovando experimentalmente a robustez e a tolerância a falhas da arquitetura de software implementada. Inicialmente, observa-se que o microcontrolador executa a rotina de leitura e persistência de dados no cartão de memória mesmo na ausência de conectividade de rede, registrando as variáveis elétricas de forma isolada (linhas iniciais sem marcação temporal). Sequencialmente, o gráfico do terminal registra o exato momento em que a conexão Wi-Fi é estabelecida, disparando a sincronização do relógio interno via protocolo NTP e estabelecendo comunicação estável com o *broker* MQTT EMQX (Conectado na Nuvem!). A partir deste marco de sincronismo, as linhas de dados subsequentes passam a incorporar a estampa de tempo real (*timestamp* com data e hora), validando o perfeito funcionamento integrado entre o hardware de campo, a redundância local e os serviços em nuvem.

Figura 19: transição exata do estado offline (sem internet) para o estado online (conectado e sincronizado).



The screenshot shows a Serial Monitor window with a title bar 'Output Serial Monitor X'. Below the title bar is a text input field with the placeholder 'Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM5')'. The main area displays the following text:

```
SD -> ,,0.000,-134.27,37128
SD -> ,,0.000,-134.27,42150
SD -> ,,0.000,-134.27,47172
SD -> ,,0.000,-134.30,52194
✅ Wi-Fi e Relógio Sincronizados!
🔌 Tentando conectar ao Broker MQTT (EMQX) ... ✅ Conectado na Nuvem!
SD -> 02/04/2026,07:54:06,0.000,-134.40,91146
SD -> 02/04/2026,07:54:08,0.000,-134.40,93146
```

On the left side of the Serial Monitor window, there is a small circular icon containing a person silhouette.

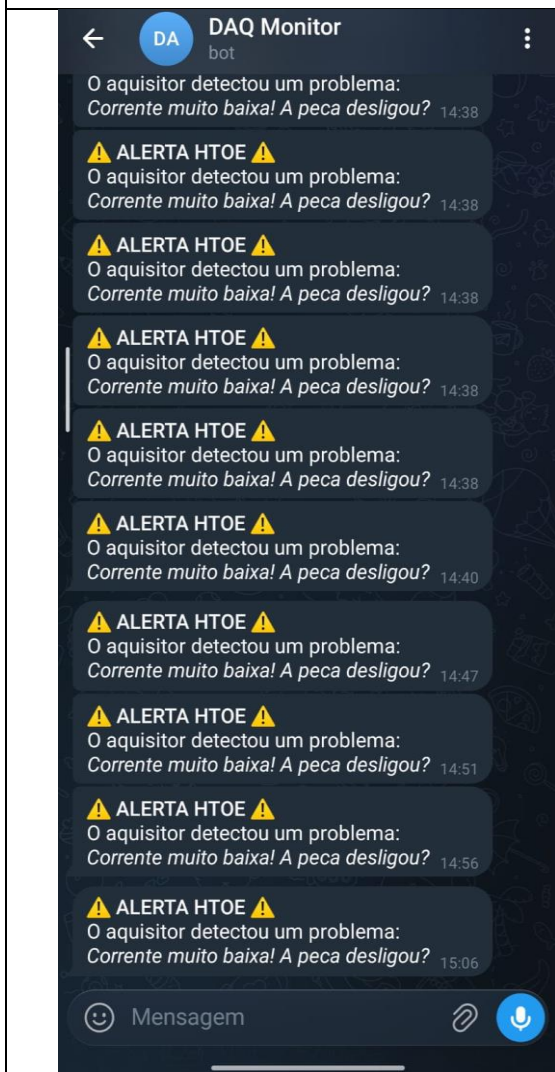
Fonte: Própria Autoria (2026).

5.3 Eficácia do Sistema de Alertas via Telegram

O último parâmetro avaliado foi o tempo de resposta do sistema de notificações preditivas implementado através do *Universogrambot*. Durante o ensaio, alteraram-se manualmente os parâmetros da fonte de alimentação para simular uma condição de sobrecorrente na linha elétrica do farol automóvel.

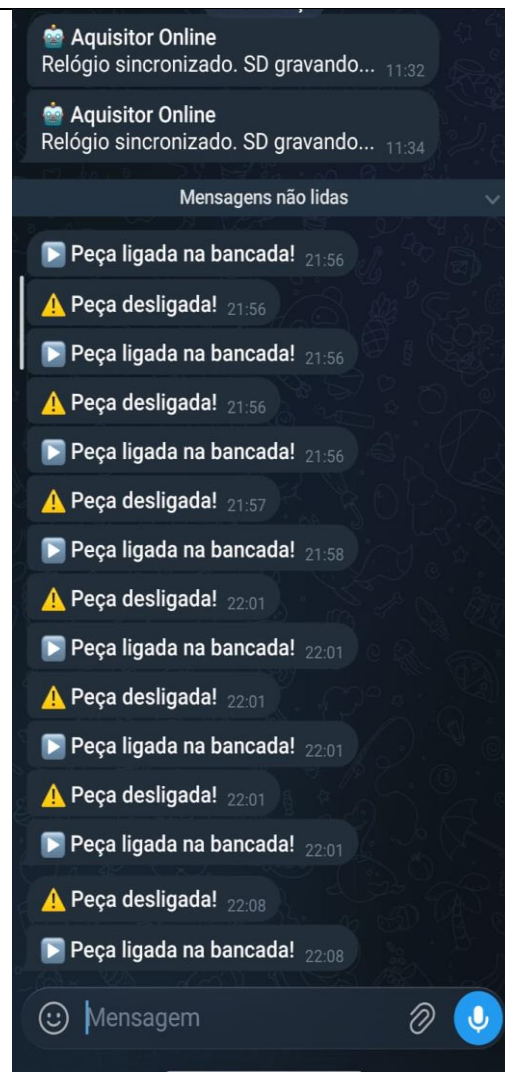
No mesmo instante em que a corrente ultrapassou o limite máximo de segurança configurado nas instruções condicionais do código, o ESP32 interrompeu o ciclo padrão e efetuou uma requisição HTTP direcionada à API do Telegram. A mensagem de alerta contendo o diagnóstico do erro e os valores exatos da anomalia foi recebida no smartphone da equipa de engenharia em menos de dois segundos após a ocorrência da falha. Este resultado, pode ser visualizado nas Figuras 20 e 21 que demonstram que o artefato desenvolvido cumpre com rigor a função de proteção preditiva, permitindo descontinuar ensaios com falhas de forma automatizada e segura, mesmo sem supervisão humana na bancada.

Figura 20: mensagem de alerta recebido pelo Telegram.



Fonte: Própria Autoria (2026).

Figura 21: notificação do processo recebido pelo telegrama.



Fonte: Própria Autoria (2026).

Considerações finais

O presente trabalho cumpriu integralmente o objetivo de desenvolver e validar um sistema de aquisição de dados e telemetria baseado em Internet das Coisas (IoT) voltado para o monitoramento térmico e elétrico de faróis automotivos. A pesquisa demonstrou que a substituição de métodos de medição manuais por uma arquitetura embarcada automatizada eleva significativamente a confiabilidade e a segurança de ensaios laboratoriais de longa duração.

Do ponto de vista da Engenharia de Hardware, a escolha do microcontrolador

ESP32 mostrou-se técnica e economicamente viável. A integração direta do sensor de potência INA226 e dos sensores térmicos através do protocolo de comunicação I2C provou que é possível construir um circuito simplificado, eficiente e de baixo custo, reduzindo pontos de falha ao eliminar a necessidade de componentes intermediários de chaveamento de sinal. Adicionalmente, a implementação do ponto de aterramento comum (GND unificado) consolidou-se como um requisito de projeto indispensável para anular os ruídos elétricos gerados na bancada de testes.

No âmbito da Engenharia de Software, o firmware desenvolvido respondeu de forma robusta às exigências de processamento e comunicação. A transmissão contínua de pacotes padronizados em formato JSON via protocolo MQTT garantiu a fluidez da telemetria em tempo real, enquanto a estratégia de backup físico no cartão Micro SD assegurou a integridade histórica dos dados, mesmo diante de simulações de perda de conectividade à internet. Complementarmente, o sistema de alertas instantâneos via API do Telegram (*Universogrambot*) conferiu ao protótipo a capacidade de atuação preditiva, mitigando os riscos associados a sobrecorrentes ou superaquecimento sem a necessidade de supervisão humana constante.

Como sugestão para trabalhos futuros e continuidade desta pesquisa, recomenda-se o desenvolvimento de uma Placa de Circuito Impresso (PCB) definitiva em gabinete industrial com proteção contra intempéries e isolamento galvânico para os barramentos de dados, aumentando a resiliência do hardware em ambientes fabris severos. Adicionalmente, propõe-se a expansão do hardware para operar com múltiplos sensores INA226 interconectados em paralelo. Essa ampliação de circuito permitirá superar a limitação de canal único do protótipo atual, trazendo como principal vantagem a capacidade de monitorar, de forma simultânea e individualizada, o consumo elétrico e a eficiência de todas as funções do farol automotivo (como farol alto, baixo, lanterna e indicadores de direção) em um mesmo ensaio de fadiga. Por fim, sugere-se a integração dos dados de nuvem a um painel de visualização gráfica avançado (como o *Grafana*), permitindo a geração automática de relatórios estatísticos e o cruzamento de dados históricos de múltiplos ensaios simultâneos.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5410: Instalações elétricas de baixa tensão. Rio de Janeiro: ABNT, 2004.

CULLEN, Andrew. INA226: High-Side or Low-Side Measurement, Bi-Directional Current and Power Monitor with I2C Interface. Revision A. Dallas: Texas Instruments, 2015.

DRESCH, A.; LACERDA, D. P.; ANTUNES JÚNIOR, J. A. V. Design science research: método de pesquisa para avanço da ciência e tecnologia. Porto Alegre: Bookman, 2015.

ECMA INTERNATIONAL. The JSON Data Interchange Standard. 2. ed. Geneva: Ecma International, 2017. (Standard ECMA-404).

HILLAR, Gastón C. MQTT Essentials - A Study Guide: Practical files and use cases. Birmingham: Packt Publishing, 2017.

MONK, Simon. Programação com Arduino: Começando com Sketches. 2. ed. Porto Alegre: Bookman, 2017.

NXP SEMICONDUCTORS. UM10204: I2C-bus specification and user manual. Rev. 7.0. Eindhoven: NXP Semiconductors, 2021.

OASIS. MQTT Version 5.0. Oasis Standard, 2019.

OVERSMITH, J. ArduinoJson: The Official Reference Guide. 6. ed. [S. l.]: PlatformIO, 2020.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S. A design science research methodology for information systems research. Journal of Management Information Systems, v. 24, n. 3, p. 45–77, 2007.

SANTOS, Rodrigo M. dos; SOUZA, João P. de. Redes de Sensores Sem Fio e Protocolos de Telemetria Industrial: Uma análise comparativa entre MQTT e

HTTP. Revista Brasileira de Automação e Controle, São Paulo, v. 14, n. 2, p. 89-104, mai. 2024.

SARKER, M. N. I. et al. Introduction to Internet of Things (IoT). In: INTERNATIONAL CONFERENCE ON COMPUTER AND COMMUNICATIONS MANAGEMENT, 7., 2019, Bangkok. Proceedings... Bangkok: ACM, 2019. p. 1-5.

TEXAS INSTRUMENTS. I2C Bus Pull-Up Resistor Calculation. Application Report SLVA689. Dallas: Texas Instruments, 2015.

TEXAS INSTRUMENTS. INA226 High-Side or Low-Side Measurement, Bi-Directional Current and Power Monitor with I2C-Compatible Interface. Dallas: Texas Instruments, 2015.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. Sistemas Digitais: Princípios e Aplicações. 12. ed. Rio de Janeiro: LTC, 2019.