

Sistema portátil de inventário e rastreamento de ativos na construção civil baseado em RFID UHF e IoT

Portable asset inventory and tracking system for construction using RFID UHF and IoT

João Vitor Gomes Vieira ¹

Yuri Cavalcanti Aquino ²

Ubiracy Dantas da Nóbrega ³

Lucas de Almeida Santos ⁴

Jan Erik Mont Gomery Pinto ⁵

Leonardo Gomes de Paiva Amorim ⁶

João Moreno Vilas Boas de Souza Silva ⁷

Resumo

A construção civil apresenta desafios persistentes relacionados ao controle de materiais e à rastreabilidade de ativos, especialmente em ambientes remotos ou sem conectividade contínua. Esses fatores contribuem para perdas, falhas de inventário e baixa eficiência operacional. Diante desse cenário, este trabalho teve como objetivo desenvolver um sistema portátil de rastreamento baseado em RFID UHF, integrado a funcionalidades de IoT, capaz de operar tanto *online* quanto *offline*. A metodologia adotada envolveu o desenvolvimento de um dispositivo embarcado com armazenamento local em cartão de memória, comunicação via Wi-Fi para sincronização posterior e integração com uma API responsável pela persistência dos dados em servidor e banco PostgreSQL. O sistema foi avaliado por meio de testes de bancada e testes em campo, analisando desempenho de leitura, estabilidade da comunicação e integridade dos dados coletados. Os resultados demonstraram que a solução apresenta funcionamento estável, leitura consistente das etiquetas e sincronização confiável após reconexão. Conclui-se que o sistema proposto representa uma alternativa viável e eficiente

¹ Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: joaovictor73307@gmail.com

² Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: yuricavalcantia@hotmail.com

³ Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: dantasnobrega@gmail.com

⁴ Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: luksalmeida123@gmail.com

⁵ Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: jan.pinto@ufersa.edu.br

⁶ Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: leonardo.amorim@escolar.ifrn.edu.br

⁷ Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: joao.vilasboas@escolar.ifrn.edu.br

para aprimorar processos de inventário e rastreamento na construção civil, oferecendo maior controle, auditabilidade e resiliência operacional.

Palavras-chave: RFID; IoT; Rastreamento de ativos; Inventário automatizado; Construção civil.

Abstract

The construction industry faces persistent challenges related to material control and asset traceability, particularly in remote environments or locations without continuous connectivity. These factors contribute to losses, inventory failures, and low operational efficiency. In light of this scenario, this study aimed to develop a portable RFID UHF-based tracking system integrated with IoT functionalities, capable of operating in both online and offline modes. The adopted methodology involved the development of an embedded device with local storage on a memory card, Wi-Fi communication for subsequent synchronization, and integration with an API responsible for data persistence on a server and PostgreSQL database. The system was evaluated through bench and field tests, analyzing reading performance, communication stability, and the integrity of the collected data. The results demonstrated that the solution provides stable operation, consistent tag reading, and reliable synchronization after reconnection. It is concluded that the proposed system represents a viable and efficient alternative for improving inventory and tracking processes in the construction industry, offering greater control, auditability, and operational resilience.

Keywords: RFID; IoT; Asset Tracking; Automated Inventory; Construction Industry.

1 INTRODUÇÃO

A *Radio-Frequency Identification* (RFID) vem se consolidando como uma tecnologia essencial para sistemas modernos de automação. Segundo Landt (2005), sua crescente presença no cotidiano decorre da capacidade de tornar processos mais práticos, produtivos e rastreáveis. Atualmente, a RFID é aplicada em diversos contextos, como prevenção de furtos, controle de mercadorias, cobrança automática de pedágios, gerenciamento de tráfego, controle de acesso e operações logísticas.

Além de seu uso consolidado em setores industriais, a tecnologia RFID apresenta grande potencial para a construção civil. Em muitos projetos, o acompanhamento de mão de obra, materiais e equipamentos ainda depende exclusivamente de supervisores ou engenheiros de campo, o que se torna inviável à medida que a escala cresce. A falta de controle adequado resulta em atrasos, desperdícios e aumento de custos. Nesse cenário, o RFID surge como uma solução capaz de ampliar o controle sobre estoques, movimentações de equipamentos e alocação da força de trabalho, contribuindo para operações mais seguras e eficientes (Khan, 2024).

As perdas de materiais e ferramentas continuam sendo um desafio significativo no setor da construção civil, afetando diretamente custos e produtividade. A tecnologia RFID, ao permitir a identificação de objetos sem linha de visão direta, supera limitações de códigos de barras e reduz o tempo necessário para inventários (Finkenzeller, 2010). Quando integrada à *Internet das Coisas* (IoT), possibilita monitoramento e correlação de dados em tempo real, alinhando-se à visão de Ashton (2009) sobre a interação entre objetos físicos e sistemas digitais.

Diante desse contexto, o presente projeto propõe o desenvolvimento de um sistema portátil voltado ao rastreamento e registro de itens em canteiros de obras, permitindo acompanhar não apenas a identificação, mas também a movimentação física dos objetos. Combinada ao georreferenciamento, a solução possibilita registrar deslocamentos com precisão temporal e espacial, oferecendo maior transparência, segurança e auditabilidade. Em ambientes com conectividade limitada, o armazenamento local no dispositivo portátil assegura a continuidade das operações e a confiabilidade dos dados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos e tecnologias fundamentais que embasam o desenvolvimento deste trabalho. A seção aborda a Internet das Coisas, a tecnologia RFID em *Ultra High Frequency* (UHF), padrões de comunicação e identificação, além dos principais protocolos e plataformas utilizadas na implementação do sistema.

2.1 Internet das Coisas (IoT)

Refere-se à interconexão de dispositivos físicos capazes de coletar, transmitir e, em alguns casos, processar dados de forma autônoma. Esses dispositivos formam redes distribuídas compostas por sensores, atuadores e sistemas de comunicação, permitindo a integração entre o mundo físico e plataformas digitais (Gubbi *et al.*, 2013).

2.2 RS-232

Estabelecido pela *Electronic Industries Association* (EIA), define um método de comunicação serial assíncrona ponto a ponto amplamente utilizado em sistemas embarcados e equipamentos industriais (EIA, 1997). Ele especifica níveis de tensão, sinalização e características elétricas, empregando conectores DB-9 ou DB-25 para estabelecer o enlace físico.

2.3 Electronic Product Code (EPC)

Identificador padronizado para objetos físicos no contexto de sistemas RFID, desenvolvido pelo Auto-ID Center e normatizado pelo EPCGLOBAL. A versão mais utilizada em aplicações UHF é o EPC de 96 bits, estruturado para fornecer identificação única e escalável de itens individuais (EPCGLOBAL, 2008). Sua codificação segmenta os 96 bits em campos como *Header*, *Filter*, *Partition*, *Company Prefix*, *Item Reference* e *Serial Number*, permitindo grande capacidade de endereçamento e interoperabilidade entre sistemas.

2.4 ESP32

Microcontrolador de alto desempenho desenvolvido pela Espressif, integrando CPU *dual-core*, conectividade Wi-Fi e Bluetooth em um único chip de baixo consumo energético. Ele inclui conversores ADC e DAC, interfaces como SPI, I²C, UART, PWM e capacidade para operar em modos de economia de energia, o que o torna adequado para aplicações móveis e IoT (Espressif, 2016).

2.5 Hypertext Transfer Protocol (HTTP)

Protocolo de comunicação utilizado para transferência de dados na Web, operando no modelo cliente-servidor. Ele define como requisições e respostas são estruturadas. Sua natureza sem estado (*stateless*) simplifica a comunicação, enquanto métodos como *GET*, *POST*, *PUT* e *DELETE* oferecem suporte às operações fundamentais sobre recursos. As versões recentes, como o *Protocolo de Transferência de Hipertexto* (HTTP/1.1 e HTTP/2),

introduzem otimizações de persistência, compressão e multiplexação, tornando o protocolo eficiente para sistemas distribuídos e aplicações IoT conectadas a serviços remotos (Fielding; Reschke, 2014).

2.6 Application Programming Interface (API)

Conjunto de regras e padrões que define como sistemas distintos podem se comunicar entre si. Em arquiteturas modernas, especialmente na Rede Internet (Web), APIs fornecem *endpoints* que permitem acesso controlado a dados e funcionalidades por meio de protocolos como HTTP. Elas promovem interoperabilidade, abstração e modularidade, permitindo que aplicações clientes, incluindo dispositivos IoT, enviem requisições e recebam respostas em formatos padronizados, como JSON ou XML. Segundo Fielding (2000), APIs baseadas nos princípios *REST* tornam os serviços escaláveis, simples de integrar e adequados para sistemas distribuídos.

3 METODOLOGIA

Neste capítulo são descritas as etapas necessárias para o desenvolvimento, implementação e validação de um coletor RFID portátil voltado ao rastreamento *offline* de itens em canteiros de obras remotos.

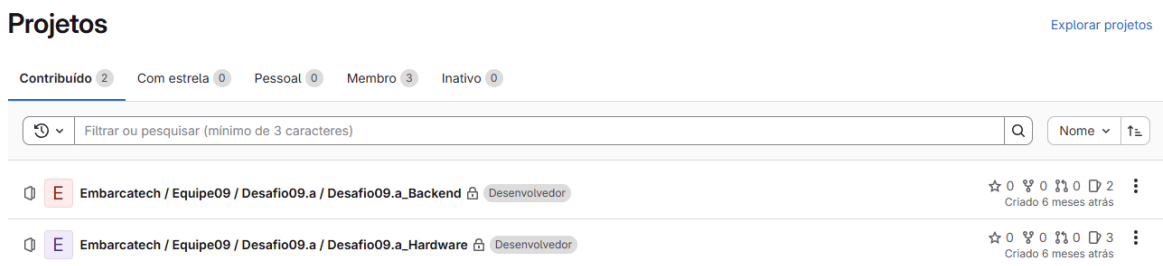
3.1 Scrum

A equipe adotou uma abordagem ágil baseada no *framework* Scrum, garantindo organização, comunicação contínua e evolução incremental do sistema. As reuniões presenciais ocorrem semanalmente, às quintas-feiras à tarde, e são conduzidas pelo Scrum Master, que orienta o grupo quanto às próximas etapas, esclarece impedimentos e avalia as tarefas concluídas no ciclo anterior. Durante esses encontros, o time revisa o *backlog* técnico, distribui responsabilidades e define prioridades para o *sprint* seguinte, assegurando alinhamento contínuo com os objetivos do projeto.

3.2 Organização do Projeto

O grupo mantém uma carga horária média de 12 horas semanais de atividades presenciais no Instituto Federal de Educação, Ciência e Tecnologia do Estado do Rio Grande do Norte (IFRN), distribuídas entre planejamento, desenvolvimento, testes e documentação. As tarefas são organizadas e monitoradas por meio da plataforma GitLab, na qual são gerenciados os repositórios de *backend* e *hardware*. O uso de *issues*, *branches* e *merge requests* permite rastreabilidade completa da evolução do código e das decisões técnicas tomadas durante o desenvolvimento. A Figura 1 apresenta a organização das atividades e o gerenciamento dos repositórios na plataforma GitLab.

Figura 1 – Projetos no GitLab



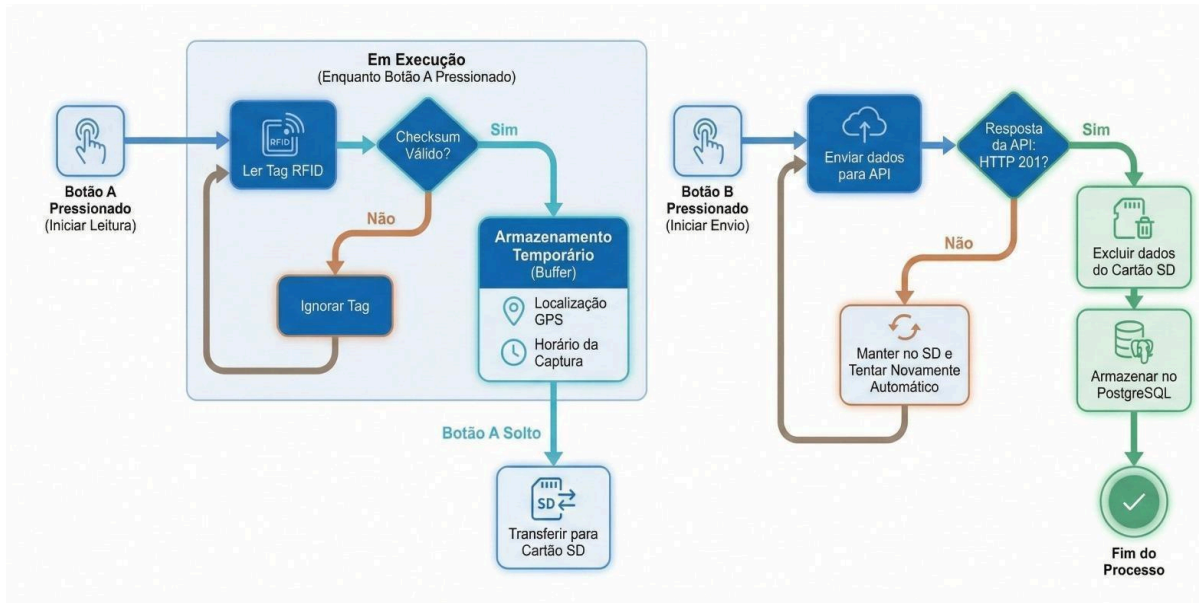
Fonte: Autores, 2026

3.3 Fluxo do sistema

O fluxo do sistema inicia-se quando o botão A é pressionado, ativando o modo de leitura de *tags*. A cada *tag* capturada, realiza-se a validação do *checksum*: se o valor for inválido, a leitura é descartada e o equipamento permanece no modo de captura enquanto o botão A continuar pressionado; se o *checksum* for válido, o identificador da *tag* é registrado em um *buffer* juntamente com os dados de GPS, como localização e horário. Quando o botão A é liberado, todo o conteúdo armazenado no *buffer* é transferido para o cartão de memória MicroSDHC. Caso o botão B seja acionado, inicia-se o modo de envio e os dados presentes no cartão de memória são transmitidos para a API. Se o envio ocorrer com sucesso, com retorno HTTP 201, os registros são removidos do cartão de memória; se a resposta for diferente de 201, os dados são mantidos e o sistema realiza novas tentativas automaticamente.

Por fim, os dados aceitos pela API são encaminhados a um banco de dados PostgreSQL, onde são armazenados de forma permanente. A Figura 2 apresenta o fluxo de funcionamento do sistema, desde a leitura das *tags* até o armazenamento dos dados no banco PostgreSQL.

Figura 2 - Fluxo de Coleta, Armazenamento e Envio de Dados da Tag RFID



Fonte: Autores, 2026

3.4 Hardware

A configuração de *hardware* do sistema encontra-se em fase de desenvolvimento e consolidação, com os componentes sendo selecionados de acordo com os requisitos funcionais definidos para o protótipo. O projeto prevê a utilização de uma placa ESP32 como unidade central de processamento, responsável pelo controle dos periféricos, integração dos sensores e gerenciamento da comunicação de rede. Para a leitura das etiquetas, está prevista a incorporação de um módulo RFID UHF compatível com o padrão EPC Gen2, associado a uma antena projetada para a faixa UHF, de modo a garantir alcance adequado e estabilidade nas leituras.

O sistema também deverá incluir um módulo GPS para obtenção de dados de localização e tempo, juntamente com um cartão de memória MicroSDHC para armazenamento local dos registros no formato TXT. Os testes preliminares indicam a necessidade de uma bateria recarregável com circuito de regulação apropriado, além da

inclusão de botões físicos destinados às funções de acionamento de leitura e *upload* manual dos dados.

3.5 Software e tecnologias

A arquitetura de *software* do sistema contempla tanto o desenvolvimento embarcado quanto os serviços de *backend* e os formatos de comunicação utilizados. No microcontrolador, a implementação foi realizada em C/C++, responsável pelo controle do leitor RFID via UART, gerenciamento do cartão de memória utilizando sistema de arquivos FATFS, processamento dos dados do módulo GPS e envio de requisições HTTP/HTTPS ao servidor. O desenvolvimento foi conduzido no PlatformIO, integrado ao VSCode, garantindo automação de compilação, padronização do ambiente e organização das configurações do *firmware*. No *backend*, a API REST foi implementada em Node.js. Os dados enviados pelo coletor são armazenados em um banco PostgreSQL, escolhido pela robustez, confiabilidade transacional e boa performance no tratamento de grandes volumes de registros temporais.

A comunicação entre o dispositivo e o servidor ocorre utilizando mensagens estruturadas em JSON, assegurando padronização e interoperabilidade. Para funcionamento *offline*, o armazenamento local dos dados no cartão de memória é feito em arquivos no formato TXT separados em linha, facilitando leitura sequencial e recuperação dos dados coletados.

3.6 Desenvolvimento

O desenvolvimento da solução ocorreu de forma gradual, conforme surgiam as demandas e conforme os equipamentos se tornavam disponíveis para testes. Inicialmente, a empresa Dois A Engenharia LTDA disponibilizou duas antenas RFID M-ID12L e tags da Siamac. A Figura 3 apresenta a antena RFID disponibilizada.

Figura 3 - Antenas RFID M-ID12L

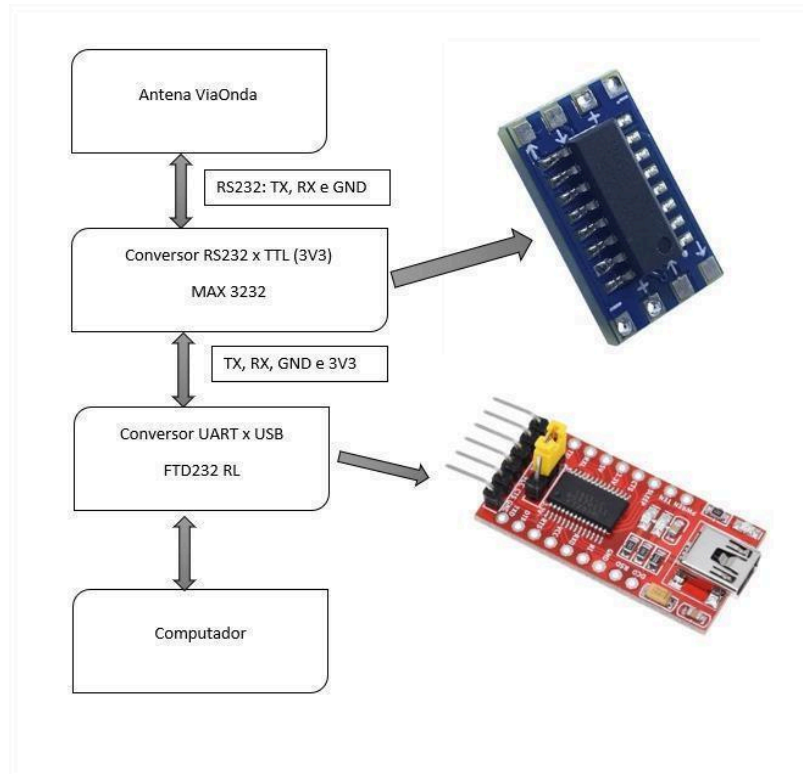


Fontes: Autores, 2026

O primeiro protótipo, desenvolvido para testes preliminares de comunicação, revelou a incompatibilidade do protocolo Wiegand de 34 bits para esta aplicação. A falha de integração decorreu da disparidade entre a capacidade do protocolo e o volume de dados das tags, que transmitem um EPC de 96 bits somado a informações adicionais. Esse excedente resultava no truncamento e na conseqüente perda de dados durante a recepção, tornando o padrão Wiegand insuficiente para o tráfego de informações exigido.

Em seguida, buscou-se estabelecer comunicação via Ethernet, porém a antena encontrava-se configurada com um endereço IP desconhecido, previamente alterado. Durante o processo de busca pelo IP, mais de cem mil endereços foram testados sem sucesso, até que se decidiu realizar o reset da configuração de rede. Com o IP restaurado, tornou-se viável acessar os dados lidos pela antena diretamente no computador, através do protocolo Ethernet e do software UHFReader18, como apresentado na Figura 4.

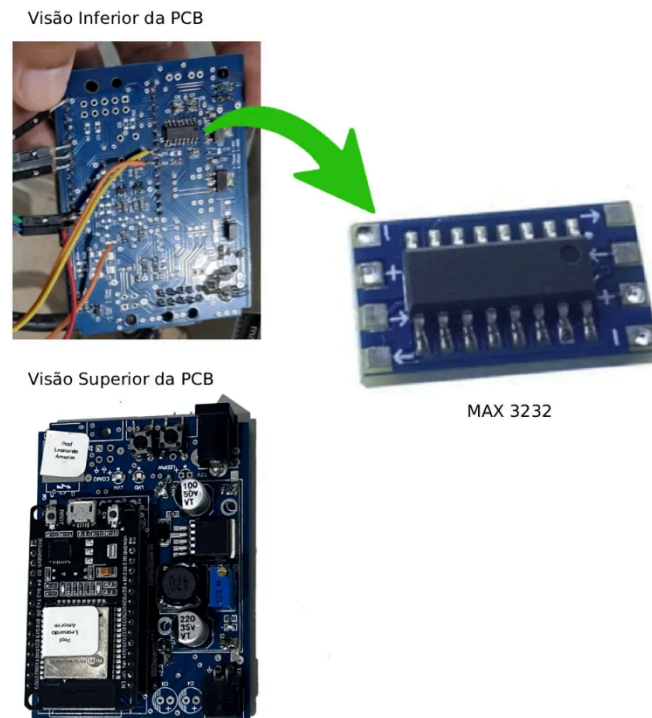
Figura 5 - Fluxograma de conversão RS-232 para USB



Fonte: Autores, 2026

Com a comunicação via Wiegand apresentando perda de informação e a comunicação serial via RS-232 funcionando corretamente no desktop, optou-se por migrar a integração do dispositivo de leitura RFID com o ESP32 para esse padrão. Entretanto, tal abordagem introduziu novos requisitos, como converter os dados em RS-232 para o padrão TTL, permitindo que os dados fossem interpretados pela UART do microcontrolador. A necessária conversão dos dados foi feita utilizando uma Placa de Circuito Impresso (PCB) que contém em seu circuito um Módulo Conversor MAX3232 e pinos dedicados à ancoragem do ESP32, como mostra a Figura 6.

Figura 6 - PCB que converte RS-232 para TTL integrada ao ESP32



Fonte: Autores, 2026

3.7 Testes e validação

A validação do sistema foi estruturada em múltiplas etapas complementares, combinando testes de bancada, testes unitários e avaliações em campo para garantir robustez, desempenho e confiabilidade operacional. Inicialmente, foram realizados testes de bancada no laboratório do IFRN, em ambiente controlado, utilizando antenas RFID e cenários simulados. Nessa fase, foram avaliados parâmetros como alcance efetivo, integridade do sinal, confiabilidade das leituras, estabilidade da comunicação entre o microcontrolador e os módulos, além de verificação da camada lógica, por meio de testes unitários aplicados à verificação de *checksum*, manipulação de arquivos e escrita em cartão de memória. Em seguida, foram conduzidos testes de comunicação, consistindo no envio de lotes de registros para a API, de modo a analisar o comportamento do sistema diante de cenários de falha, como timeouts, perda de conexão e respostas de erro. Também foram avaliados mecanismos de retransmissão, *buffers* temporários e estratégias de recuperação.

A etapa final envolve testes em campo, realizados no campo de futebol do IFRN, em ambiente aberto e com menor reflexão de sinal, permitindo observar o desempenho do sistema em condições próximas à operação real. Nessa fase, foram analisados a taxa de leitura (tags/s), o tempo de registro, a confiabilidade das leituras em diferentes distâncias e orientações, bem como os efeitos de interferências ambientais. Esses testes complementares possibilitam validar a robustez do sistema e identificar ajustes necessários para sua utilização em cenários práticos

4 RESULTADOS E DISCUSSÕES

A primeira versão do protótipo, a interface física, ainda não incluía botões de interação, de modo que o funcionamento dependia integralmente de rotinas temporizadas. Após um intervalo pré-definido, em torno de um minuto, o dispositivo realiza a leitura dos dados armazenados no cartão de memória e envia automaticamente esses registros para a API, permitindo que fossem persistidos no banco de dados.

Essa estratégia foi adotada como solução provisória até a implementação completa dos controles físicos, garantindo que o fluxo de coleta e transmissão de informações permanecesse funcional durante os testes iniciais. Os primeiros ensaios de desempenho foram realizados utilizando as tags da Siamac, testadas em campo aberto no gramado do estádio do IFRN, como mostra a Figura 7. Nessa etapa, a distância máxima de leitura obtida foi de aproximadamente quatro metros, valor significativamente inferior ao esperado para o cenário de uso real.

Figura 7 - Teste em campo das tags Siamac



Fonte: Autores, 2026

Diante dessa limitação, iniciou-se uma busca por fornecedores nacionais de tags RFID que pudessem oferecer melhor sensibilidade e maior ganho de leitura. Como parte desse processo, foram obtidas amostras gratuitas da empresa Haco RFID, permitindo a realização de novos testes comparativos. Com as tags fornecidas pela Haco, foram realizados testes em campo, como mostrado na Figura 8, com isso foi possível alcançar um desempenho expressivamente superior, atingindo cerca de doze metros de distância de leitura, o que representou um avanço importante para a viabilidade operacional do sistema.

Figura 8 - Testes em campo das tags da Haco

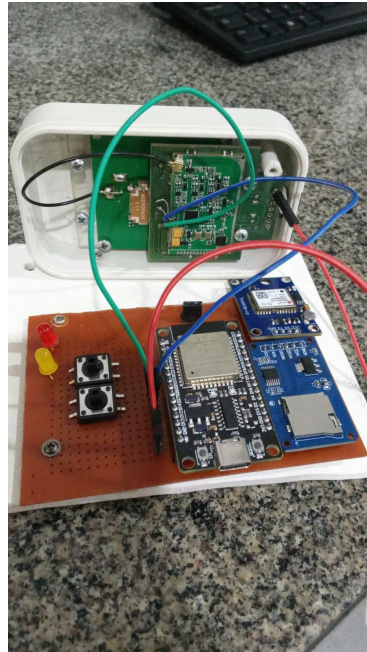


Fonte: Autores, 2026

Na etapa seguinte do desenvolvimento, foram disponibilizados dois leitores RFID USB de mesa, modelo M-ID10S, compostos por um módulo leitor integrado à antena. A adoção desses dispositivos representou um avanço significativo, pois possibilitou reduzir o tamanho do conjunto experimental e aproximar o protótipo do formato previsto para o hardware final, tanto em termos de dimensões quanto de componentes necessários para o projeto da placa e do invólucro físico.

Após a identificação dos pinos de transmissão e recepção do módulo, estabeleceu-se a comunicação com o ESP32 via UART, enfrentando novamente desafios relacionados à definição do *baud rate* correto e à interpretação dos dados em formato hexadecimal. Superadas essas etapas, as leituras passaram a ser processadas de maneira íntegra e estável. Um avanço importante dessa versão foi a integração do módulo GPS, permitindo que cada coleta fosse registrada com coordenadas geográficas e marca temporal precisa, conforme mostrado na Figura 9.

Figura 9 - Protótipo com GPS



Fonte: Autores, 2026

Durante o processo de desenvolvimento e validação dos primeiros protótipos, diversos desafios técnicos foram identificados, contribuindo para refinar a compreensão sobre os limites e potencialidades do sistema proposto. Inicialmente, os testes realizados com a antena M-ID12L levantaram a hipótese de que seu alcance poderia ter sido comprometido pelo tempo de uso. Entretanto, a chegada de novas TAGs permitiu confirmar a eficiência da antena, demonstrando que o desempenho inferior observado estava relacionado às TAGs previamente utilizadas e não ao hardware principal. Mesmo com essa validação, destaca-se que a antena definitiva, de menor tamanho e, conseqüentemente, menor potência, ainda não pôde ser testada, o que limita, por ora, conclusões mais precisas sobre o comportamento esperado na configuração final.

Além disso, a segunda etapa de experimentação foi realizada com o módulo leitor presente no dispositivo M-ID10S, apresentado na Figura 10, que serviu como plataforma intermediária enquanto o circuito definitivo continua em desenvolvimento. Apesar de permitir avanços importantes, o módulo apresentou indícios de falha elétrica, como aquecimento anormal do hardware, acionamento aleatório de LEDs e comportamentos inconsistentes do buzzer. Esses sintomas impediram a realização de testes de campo mais extensivos, especialmente porque o alcance do módulo é reduzido e seu desempenho não representa o previsto para a versão final do equipamento. Como o projeto prevê a integração de um leitor

mais potente para atender às demandas da Dois A Engenharia LTDA, os resultados desta fase devem ser interpretados como avaliações preliminares.

Figura 10 - Gravador M-ID10S



Fonte: Autores, 2026

A infraestrutura do sistema, na etapa atual de desenvolvimento, opera integralmente em serviços gratuitos com o objetivo de reduzir custos durante a fase experimental. A API está hospedada na plataforma Vercel, que oferece *deploy* contínuo, gerenciamento simplificado e boa disponibilidade para aplicações em estágio inicial. O banco de dados, por sua vez, utiliza o serviço Neon DB, escolhido pela compatibilidade com PostgreSQL, escalabilidade automática e suporte ao modelo serverless. Essa combinação permitiu persistir os dados enviados pelo coletor de forma confiável e com baixa complexidade operacional, assegurando um ambiente adequado para testes funcionais e validação da arquitetura proposta. A Figura 11 apresenta dados salvos no Neon DB.

Figura 11 - Dados armazenados no Neon DB

id	idTag	idDispositivo	dataHoraLeitura	latitude	longitude	altitude	criadoEm
22729	4341baf61c22419b09a24d...	esp32_490	2025-11-19 17:15:36.32...	-5.8119549	-35.1685198	11.336	2025-11-26 10:36:58.38...
22728	5e5449cb818d4c42b559f2...	esp32_489	2025-11-20 08:04:06.32...	-5.8058146	-35.1748003	14.342	2025-11-26 10:36:58.38...
22727	357c9ea15e524564bf2b4...	esp32_488	2025-11-23 16:24:38.32...	-5.8081430	-35.1682188	7.310	2025-11-26 10:36:58.38...
22726	89ef177e1a91438c848be2...	esp32_487	2025-11-25 21:33:27.32...	-5.7954673	-35.1747576	6.698	2025-11-26 10:36:58.38...
22725	dcaeed0ba657401eb0978d...	esp32_486	2025-11-21 02:11:37.32...	-5.8029231	-35.1748646	12.208	2025-11-26 10:36:58.38...
22724	dce9f250842a40a2ae1d25...	esp32_485	2025-11-20 17:13:25.32...	-5.8060933	-35.1731756	13.501	2025-11-26 10:36:58.38...
22723	b67961777e6f4758aa496d...	esp32_484	2025-11-21 18:50:23.32...	-5.8060765	-35.1696335	10.737	2025-11-26 10:36:58.38...
22722	b3de1be12cf4d22a987dd...	esp32_483	2025-11-24 18:29:16.32...	-5.8040345	-35.1648572	10.895	2025-11-26 10:36:58.38...
22721	8be9432c66b64b5aa83d3d...	esp32_482	2025-11-25 12:56:08.32...	-5.7992301	-35.1741424	10.819	2025-11-26 10:36:58.38...
22720	401c33e7725845219007df...	esp32_481	2025-11-22 07:12:54.32...	-5.7957321	-35.1635152	9.674	2025-11-26 10:36:58.38...
22719	1d2de4a1d6cf4d04938c5c...	esp32_480	2025-11-20 15:01:00.32...	-5.8056815	-35.1648507	13.826	2025-11-26 10:36:58.38...
22718	965af1a862cd43cae8ebd...	esp32_479	2025-11-21 03:31:06.32...	-5.7988112	-35.1672893	11.296	2025-11-26 10:36:58.38...
22717	22d961a199b74b0b2d1ed...	esp32_478	2025-11-20 00:41:34.32...	-5.7977271	-35.1669686	14.326	2025-11-26 10:36:58.38...
22716	97afba5796542aba2f542...	esp32_477	2025-11-22 06:53:45.32...	-5.7961644	-35.1731739	12.139	2025-11-26 10:36:58.38...
22715	bcb1a9aa2a2e24a447cd...	esp32_476	2025-11-24 12:22:36.32...	-5.8061429	-35.1684678	9.277	2025-11-26 10:36:58.38...
22714	5b54bb38d9af415990801...	esp32_475	2025-11-21 16:32:29.32...	-5.8037389	-35.1588856	10.649	2025-11-26 10:36:58.38...
22713	d161003514bb453bea5cd...	esp32_474	2025-11-21 20:24:39.32...	-5.8049353	-35.1576425	12.459	2025-11-26 10:36:58.38...
22712	f2979bf78d0448a854d1...	esp32_473	2025-11-21 22:32:57.32...	-5.8120159	-35.1593030	15.433	2025-11-26 10:36:58.38...
22711	1995b3acbd5a4f11974096...	esp32_472	2025-11-24 00:05:31.32...	-5.8060334	-35.1743632	7.508	2025-11-26 10:36:58.38...
22710	94828ee386964bb181ed6...	esp32_471	2025-11-19 23:17:41.32...	-5.8064077	-35.1686701	11.333	2025-11-26 10:36:58.38...
22709	68ce21b1c04745cfadb2...	esp32_470	2025-11-20 16:47:22.32...	-5.8016801	-35.1561048	8.587	2025-11-26 10:36:58.38...
22708	946c2aee31748c8e2fce7...	esp32_469	2025-11-24 11:54:28.32...	-5.8101814	-35.1699368	7.087	2025-11-26 10:36:58.38...
22707	9fcd7bb5d5d55e39d9354...	esp32_468	2025-11-20 17:05:31.32...	-5.8029285	-35.1684366	11.428	2025-11-26 10:36:58.38...
22706	ac62b00215114bc6899a31...	esp32_467	2025-11-20 12:40:36.32...	-5.8041715	-35.1658903	14.613	2025-11-26 10:36:58.38...
22705	dfe556180a274b6793fe1e...	esp32_466	2025-11-22 20:53:32.32...	-5.8056255	-35.1685091	15.414	2025-11-26 10:36:58.38...
22704	45415b2d15a3417bac2ee...	esp32_465	2025-11-23 02:19:53.32...	-5.8121206	-35.1658734	13.708	2025-11-26 10:36:58.38...
22703	2782e6e3bf354586a6d2f0...	esp32_464	2025-11-20 20:52:26.32...	-5.8089266	-35.1616829	9.356	2025-11-26 10:36:58.38...

Fonte: Autores, 2026

Outro ponto discutido diz respeito ao processamento paralelo e à arquitetura de *software* embarcado. Observou-se que a leitura das TAGs é interrompida quando o ESP32 grava dados no cartão de memória ou tenta enviá-los à API, o que levanta o questionamento sobre a necessidade de empregar um sistema operacional embarcado (como FreeRTOS) para permitir tarefas concorrentes com maior eficiência. Essa decisão impacta diretamente o desempenho em cenários reais de uso, nos quais a leitura contínua não pode ser comprometida por operações de escrita ou comunicação.

A utilização ou não de um display no coletor também foi objeto de análise. Embora a tela pudesse fornecer *feedback* visual ao operador, a empresa optou por desenvolver uma rede própria dedicada aos coletores, reduzindo a necessidade de interfaces locais. Essa escolha simplifica o hardware, diminui consumo energético e reduz possíveis pontos de falha, alinhando-se ao objetivo de obter um dispositivo robusto, leve e de uso intuitivo.

5 CONCLUSÃO

O desenvolvimento do coletor RFID portátil para rastreamento de ativos em canteiros de obras permitiu evidenciar o potencial da integração entre RFID UHF, IoT e

georreferenciamento como solução para ambientes remotos e com infraestrutura limitada. A proposta demonstrou viabilidade técnica ao unir um dispositivo de leitura de longo alcance, armazenamento offline em cartão de memória e sincronização automática com servidor, garantindo continuidade operacional mesmo na ausência de conectividade.

Ao longo do processo de prototipação, diferentes desafios foram enfrentados, especialmente relacionados à comunicação serial com os módulos RFID, à necessidade de decodificação adequada dos dados e à identificação de parâmetros como baud rate e checksum. As dificuldades relativas ao alcance inicial foram superadas com a validação por meio de novas TAGs, reforçando a importância da escolha correta de insumos para garantir a eficiência do sistema. A experimentação com o módulo M-ID10S permitiu avançar no refinamento do protótipo, embora limitações de desempenho e sinais de falha elétrica tenham restringido testes em campo mais abrangentes.

A arquitetura de software mostrou-se adequada às necessidades do projeto, oferecendo comunicação estruturada via JSON, sincronização eficiente com uma API hospedada em ambiente gratuito e registro local resiliente. A análise do comportamento multitarefa do ESP32 indicou a necessidade futura de aprimoramentos, possivelmente com adoção de um sistema operacional embarcado para garantir paralelismo entre leitura, escrita e transmissão de dados.

O conjunto dos resultados preliminares reforça que a solução tem potencial de impactar de forma significativa a gestão de materiais na construção civil, oferecendo rastreamento contínuo, auditável e de baixo custo. Além disso, o processo iterativo empregado possibilitou identificar pontos críticos para aprimoramento nas próximas etapas, como o desenvolvimento da PCB final, o uso de um leitor RFID mais potente, a otimização da ergonomia e a realização de testes de campo com o hardware definitivo.

Conclui-se que o trabalho avança no sentido de suprir limitações recorrentes no controle de ativos em obras, apresentando uma solução tecnicamente sólida e alinhada às demandas reais do setor. Com a finalização do protótipo e a consolidação dos testes em campo, espera-se que o sistema atinja plena maturidade operacional, oferecendo maior precisão, confiabilidade e eficiência às rotinas de inventário e movimentação de materiais.

REFERÊNCIAS

ASHTON, Kevin. That ‘Internet of Things’ thing – in the real world things matter more than ideas. **RFID Journal**, New York, 22 jun. 2009. Disponível em: <https://www.rfidjournal.com/articles/view?4986>. Acesso em: 7 dez. 2025.

ELECTRONIC INDUSTRIES ASSOCIATION. **RS-232-C**: interface between data terminal equipment and data circuit-terminating equipment employing serial binary data interchange. EIA/TIA-232-C. Washington, DC: EIA, 1997.

ESPRESSIF SYSTEMS. **ESP32 series datasheet**. [S. l.]: Espressif Systems, 2016. Disponível em: <https://www.espressif.com>. Acesso em: 7 dez. 2025.

FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado em Ciência da Computação) — University of California, Irvine, 2000.

FIELDING, Roy; RESCHKE, Julian. **Hypertext transfer protocol (HTTP/1.1)**: message syntax and routing. versão RFC 7230. [S. l.]: IETF, 2014.

FINKENZELLER, Klaus. **RFID handbook**: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication. 3. ed. Chichester: Wiley, 2010.

GS1. **EPC Tag Data Standard (TDS)**: versão 2.1. [S. l.]: GS1, 2024. Disponível em: <https://ref.gs1.org/standards/tds/>. Acesso em: 28 jan. 2026.

GUBBI, Jayavardhana et al. Internet of things (IoT): a vision, architectural elements, and future directions. **Future generation computer systems**, v. 29, n. 7, p. 1645–1660, 2013.

HOFMANN-WELLENHOF, Bernhard; LICHTENEGGER, Herbert; COLLINS, James. **GPS**: theory and practice. 5. ed. Wien: Springer-Verlag, 2001.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 18000-6**: Information technology: radio frequency identification for item management: part 6: parameters for air interface communications at 860 MHz to 960 MHz. 3rd ed. Geneva: ISO: IEC, 2013. 1 pdf.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; INTERNATIONAL ELECTROTECHNICAL COMMISSION. **Information technology**: radio frequency identification for item management (ISO/IEC 18000 series). Geneva: ISO: IEC, 2004–2024.

KHAN, Shahed I.; RAY, Biplob R.; KARMAKAR, Nemai C. RFID localization in construction with IoT and security integration. **Automation in Construction**, [S. l.], v. 159, p. 105249, 2024. DOI: <https://doi.org/10.1016/j.autcon.2023.105249>. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0926580523005095>. Acesso em: 28 jan. 2026.

LANDT, Jeremy. A história do RFID. **IEEE Potentials**, v. 24, n. 4, p. 8–11, out./nov. 2005. DOI: 10.1109/MP.2005.1549751.

STALLINGS, William. **Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud**. Boston: Addison-Wesley, 2017.

STONEBRAKER, Michael; ROWE, Lawrence A. The design of Postgres. **ACM SIGMOD Record**, v. 15, n. 2, p. 340–355, 1986.

TANENBAUM, Andrew S.; WETHERALL, David J. **Computer networks**. 5. ed. Boston: Pearson, 2011.