

DESENVOLVIMENTO DE UM SISTEMA MICROCONTROLADO PARA GERENCIAMENTO DE BICICLETAS COMPARTILHADAS NA UNIVERSIDADE DO EXTREMO SUL CATARINENSE

Miguel Sombrio¹, Sergio Coral²

Resumo: Este trabalho apresenta o desenvolvimento e a validação do BikelOT, com o objetivo de propor uma solução baseada em Internet das Coisas para o gerenciamento de bicicletas compartilhadas em ambiente universitário. A solução utiliza uma arquitetura distribuída composta por *firmware* embarcado, API *backend*, aplicativo móvel e painel administrativo, integrando Bluetooth Low Energy, protocolo de mensageria sobre rede móvel 4G, GPS e *geofencing*. O *firmware* embarcado em ESP32 foi responsável pelo controle da trava, validação criptográfica de comandos e envio de telemetria. A API realizou autenticação, assinatura digital temporária e integração com o sistema de mensageria, enquanto o aplicativo móvel permitiu a autenticação do usuário e a comunicação com a bicicleta. Os resultados demonstraram a viabilidade técnica da arquitetura proposta, validando a integração entre comunicação local e remota, controle de acesso, telemetria e restrição espacial. Também foram identificadas limitações relacionadas à conectividade móvel, à precisão do GPS e à validação em ambiente controlado.

Palavras-chave: internet das coisas; bicicletas compartilhadas; ESP32; MQTT; Bluetooth Low Energy; *geofencing*.

Abstract: This work presents the development and validation of BikelOT, aiming to propose an Internet of Things-based solution for shared bicycle management in a university environment. The solution employs a distributed architecture composed of embedded *firmware*, a *backend* API, a mobile application, and an administrative dashboard, integrating Bluetooth Low Energy, a messaging protocol over a 4G mobile network, GPS, and *geofencing*. The embedded *firmware* running on an ESP32 microcontroller was responsible for lock control, cryptographic validation of commands, and te-

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. miguelsoeth@unesc.net

²Orientador, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. sergiocoral@unesc.net

lemetry transmission. The API handled authentication, temporary digital signature generation, and integration with the messaging system, while the mobile application enabled user authentication and communication with the bicycle. The results demonstrated the technical feasibility of the proposed architecture, validating the integration of local and remote communication, access control, telemetry, and spatial restriction mechanisms. Limitations related to mobile network connectivity, GPS accuracy, and validation in a controlled environment were also identified.

Keywords: internet of things; bike sharing; ESP32; MQTT; Bluetooth Low Energy; *geofencing*.

1 INTRODUÇÃO

Com o crescimento populacional nas grandes cidades, muitas vezes acima da capacidade de suporte da infraestrutura urbana, intensificam-se problemas sociais ligados à mobilidade urbana. Segundo Tavares et al. (2025), o aumento da frota de veículos contribui diretamente para a deterioração da qualidade de vida, sendo responsável por grande parte das emissões urbanas, como no caso de São Paulo, onde cerca de 40% do material particulado fino (PM_{2.5}) tem origem veicular. Esse cenário impacta não apenas a saúde, mas também a dinâmica social e econômica: longos períodos no trânsito resultam em perda de tempo produtivo, aumento do estresse diário e redução do bem-estar da população. Além disso, os congestionamentos geram custos indiretos significativos, como queda de produtividade e maiores gastos com transporte e energia, afetando tanto indivíduos quanto a economia urbana.

Nesse contexto, sistemas de compartilhamento e aluguel de meios de locomoção, como bicicletas compartilhadas (*bike-sharing*), têm sido implementados ao redor do mundo com o propósito de promover alternativas mais sustentáveis e eficientes a esse cenário. Esses sistemas apresentam diversas vantagens, atuando na diminuição de poluentes por carros e no congestionamento urbano, funcionando como solução eficiente para os deslocamentos de primeira e última milha (Cui; Zhang, 2024). Ao mesmo tempo, o compartilhamento desses modais constitui uma alternativa com flexibilidade para os usuários, promovendo benefícios sociais e de saúde ao incentivar a atividade física e ampliar o acesso à mobilidade urbana (Zimmermann; Palgan, 2024).

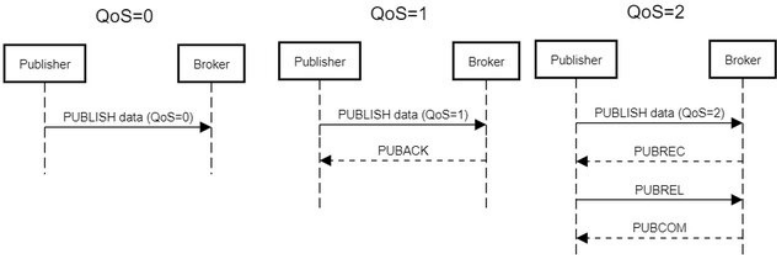
Considerando o campus universitário como um microcosmo ur-

bano, em que longas distâncias entre prédios e intervalos reduzidos entre aulas são comuns, a bicicleta surge como uma alternativa de transporte eficiente, como demonstrado pelo caso real da Tembici, cujo projeto inicial foi desenvolvido dentro da Universidade de São Paulo (Barberio; Manocchio; Gallo, 2023). Segundo Chicco e Diana (2022), viagens com bicicletas alugáveis apresentam uma distância média de cerca de 1,35 km, faixa ideal para atender a locomoção interna em um ambiente universitário.

Para viabilizar soluções mais eficientes nesse contexto, torna-se necessário compreender os fundamentos tecnológicos que sustentam sistemas modernos de mobilidade inteligente. A Internet das Coisas (IoT) surge como um padrão que conecta objetos físicos à internet, permitindo comunicação e automação entre dispositivos. Segundo Ashton (2009), sensores e atuadores podem coletar e compartilhar dados de forma autônoma, criando sistemas inteligentes aplicáveis à mobilidade urbana. Nesse cenário, o microcontrolador ESP32 destaca-se como uma plataforma amplamente utilizada, por integrar conectividade Wi-Fi e Bluetooth, baixo consumo de energia e capacidade de processamento embarcado (Espressif Systems, 2025), sendo adequado para aplicações portáteis e autônomas.

Além do hardware, a estrutura de comunicação desses sistemas é fundamental. A arquitetura orientada a eventos (EDA) baseia-se na emissão e consumo assíncrono de eventos entre serviços desacoplados, o que favorece escalabilidade e resiliência (Amazon Web Services, Inc., 2024). Nesse contexto, o protocolo MQTT se destaca por sua leveza e eficiência, sendo projetado para dispositivos com recursos limitados e redes instáveis (MQTT.org, 2025). Seu modelo *publish/subscribe* permite ajustar o equilíbrio entre desempenho e confiabilidade na entrega de mensagens. Como ilustrado na Figura 1, o MQTT define três níveis de QoS: QoS 0 (entrega no máximo uma vez), QoS 1 (entrega pelo menos uma vez) e QoS 2 (entrega exatamente uma vez), cada um com diferentes garantias e custos de comunicação (HiveMQ, 2015; Laghari et al., 2024).

Figura 1 - Níveis de QoS do MQTT



Fonte: Hmissi e Ouni (2021)

Complementando essa infraestrutura, tecnologias de comunicação sem fio como o Bluetooth Low Energy (BLE) possibilitam interação direta e de baixo consumo energético entre dispositivos próximos, sendo amplamente utilizadas em aplicações IoT. Segundo Al-Shareeda et al. (2023), o BLE oferece confiabilidade e eficiência energética, sendo adequado para operações críticas como travamento e destravamento local, mesmo na ausência de conectividade com a internet.

Na prática, diferentes abordagens têm explorado essas tecnologias em sistemas de compartilhamento de bicicletas. Hameed et al. (2021) propuseram o Rent-A-Cycle, um sistema *dockless* controlado por aplicativo, com integração de GPS, QR Code e travamento automático. Já Leme et al. (2021) desenvolveram o Smart Bike, que utiliza ESP8266, comunicação MQTT e infraestrutura com pontos físicos de ancoragem e autenticação via RFID. De forma semelhante, Carvalho et al. (2023) apresentaram o Vá de Bike, integrando hardware embarcado, broker MQTT e aplicativo móvel, incluindo mecanismos adicionais como *cashback* e autenticação em dois fatores.

Complementarmente, Kurniasyah e Rakhmawati (2025) analisaram um sistema de trava inteligente com ESP32, BLE e MQTT, avaliando aspectos como alcance do sinal, estabilidade da comunicação e impacto dos níveis de QoS. Os resultados reforçam a viabilidade técnica dessas tecnologias, mas também evidenciam limitações práticas relacionadas ao alcance, confiabilidade e escolha adequada dos parâmetros de comunicação.

No entanto, o sistema atualmente disponível no campus analisado já apresenta limitações estruturais. O uso de travas com chaves físicas gera deslocamentos adicionais até pontos específicos de retirada e devolução, o que compromete a praticidade do serviço e expõe o processo a falhas humanas, como perda, retenção ou cópia indevida das chaves. Conforme Cohen et al. (2024), a eficiência e a adoção desse modal estão diretamente ligadas à sua conveniência e segurança. Observa-se que, embora existam soluções já consolidadas no mercado para sistemas de bicicletas compartilhadas mais práticos e tecnologicamente avançados, sua adoção pode ser limitada em função da necessidade de infraestrutura específica, além das despesas contínuas de operação e manutenção. (Diogo; Pinto; Ribeiro, 2025).

Desse modo, o presente trabalho tem como objetivo validar uma solução baseada em IoT para aprimorar o controle, a utilização e a segu-

rança de bicicletas compartilhadas em ambiente universitário. Especificamente, buscou-se desenvolver o firmware embarcado, implementar a API *backend*, o aplicativo móvel e o painel administrativo, integrar esses componentes por meio de BLE e MQTT e validar experimentalmente a solução em ambiente controlado. A proposta consiste no desenvolvimento de um sistema *dockless* com travas inteligentes, operadas e gerenciadas de forma automatizada, dispensando estações físicas fixas e ampliando a acessibilidade a esse modal no campus.

Em ambientes controlados, como campus universitários, a análise de soluções de micromobilidade permite testar alternativas de baixo impacto operacional e de rápida implementação, com potencial de reduzir o uso de automóveis e melhorar o fluxo de pessoas em percursos curtos. A abordagem se alinha ao conceito de mobilidade inteligente discutido por Martins e Gonzales-Taco (2020), que ressalta a IoT como núcleo tecnológico em soluções sustentáveis e orientadas por dados.

Dessa forma, o projeto não se limita à disponibilização de um modal alternativo, mas estabelece uma infraestrutura digital completa para gestão da micromobilidade no campus. A solução possibilita auditoria de uso, monitoramento contínuo, tomada de decisão baseada em indicadores operacionais e redução de intervenções manuais. Com isso, espera-se uma operação mais eficiente, menor impacto ambiental e um modelo replicável para demais ambientes educacionais ou corporativos que buscam implementar sistemas de mobilidade inteligente.

O trabalho está estruturado em quatro capítulos. O primeiro apresenta a introdução, contextualizando o tema, explicitando o problema de pesquisa, bem como os objetivos e a justificativa, delimitando o escopo geral do estudo. O segundo descreve os materiais e métodos utilizados, detalhando os recursos empregados e as decisões metodológicas adotadas para desenvolvimento de um protótipo de *bike-sharing*. O terceiro apresenta os resultados obtidos e sua respectiva discussão, incluindo a análise dos dados, métricas avaliadas, dificuldades encontradas e a comparação com trabalhos correlatos e abordagens existentes de sistemas baseados em IoT. Por fim, o quarto capítulo expõe as conclusões do estudo, reunindo as considerações finais, destacando vantagens e desafios, além de apontar possíveis trabalhos futuros e questões em aberto.

2 MATERIAIS E MÉTODOS

O objeto de estudo deste trabalho foi um protótipo microcontrolado para gerenciamento de bicicletas compartilhadas, desenvolvido para validar uma solução baseada em Internet das Coisas aplicada ao controle, utilização e segurança de bicicletas em modelo *dockless*.³ O sistema foi projetado para aplicação em ambiente universitário, tendo como referência o campus da Universidade do Extremo Sul Catarinense, considerando características de mobilidade interna e circulação de usuários. A validação do protótipo, entretanto, foi realizada em ambiente controlado de bancada, permitindo a execução controlada dos testes e a observação detalhada do comportamento do sistema.

A solução foi composta por *firmware* embarcado, *backend* em C, aplicativo móvel e painel administrativo. O protótipo foi utilizado como unidade física de validação da arquitetura IoT, incorporando comunicação sem fio, localização, controle de estados e acionamento da trava. Além disso, foram avaliadas a integração entre os componentes e a troca contínua de informações entre os diferentes módulos da solução proposta.

Foram considerados como critérios a comunicação entre os componentes, a consistência dos estados da bicicleta, a execução dos comandos de bloqueio e desbloqueio, a validação criptográfica e temporal das permissões, a restrição espacial por *geofencing* e a transmissão de telemetria. Também foram observados aspectos relacionados à sincronização de dados, confiabilidade das mensagens e resposta do sistema durante os testes realizados. Não foram avaliados aspectos comerciais, aceitação de usuários em larga escala ou operação pública contínua.

2.1 DELINEAMENTO DA PESQUISA

A pesquisa caracterizou-se como aplicada, com abordagem baseada em prototipagem, visando validar uma solução IoT para mobilidade por meio de bicicletas compartilhadas. A metodologia adotou arquitetura distribuída composta por dispositivo embarcado, serviços de *backend* e aplicações cliente. O trabalho envolveu o desenvolvimento do *firmware*, integração de módulos de conectividade, localização e segurança, implementação da API para autenticação e comunicação MQTT, além da criação dos componentes de interação e administração. A validação experimental foi realizada em ambiente controlado de bancada, utilizando uma estrutura demonstrativa para simular os fluxos do sistema.

³Vídeo de demonstração do protótipo BikeIoT: <<https://youtu.be/Cd4i5ojTPhs>>

Foram utilizados os componentes de hardware apresentados na Tabela 1, incluindo microcontrolador ESP32, modem GSM/4G, módulo GPS e demais dispositivos necessários ao funcionamento do protótipo. O registro fotográfico dos principais componentes físicos utilizados na montagem do protótipo, incluindo ESP32, protoboard, mecanismo de trava e integração elétrica, é apresentado no Apêndice A.

Tabela 1 - Componentes de hardware utilizados no protótipo

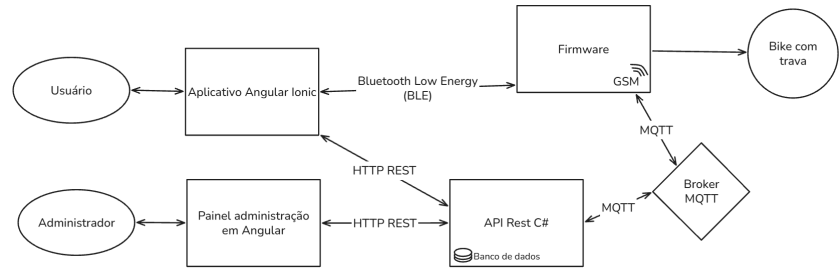
Componente	Função
ESP32-WROVER-E	Controle principal e comunicação BLE
Modem A7670SA	Comunicação remota
Antena GPS	Obtenção de localização
Servo MG996R	Movimento da trava
Bateria 18650	Alimentação do ESP32
Antena Full-Band LTE	Conexão GSM
Step-up MT3608	Conversão de tensão para o servo motor
LED Vermelho	Indicador de estado da bicicleta
Resistor 220 Ohms	Limitação de corrente do LED
Botão-chave	Trava emergencial da bicicleta
Buzzer ativo	Alerta ao sair da zona de uso

Fonte: Elaborado pelo autor.

No *backend*, foi implementada uma API em .NET organizada, contendo autenticação JWT, persistência em banco de dados e integração MQTT. O aplicativo móvel foi desenvolvido com Ionic, Angular e Capacitor, utilizando BLE para comunicação com a bicicleta. O painel administrativo foi implementado em Angular para configuração e monitoramento do sistema. A unidade de análise foi o protótipo da bicicleta e sua integração com os demais componentes. Os testes contemplaram autenticação, assinatura digital, bloqueio e desbloqueio, publicação de telemetria, validação de *geofencing*, reconexão MQTT e comportamento diante de falhas.

2.2 VISÃO GERAL E ARQUITETURA DO SISTEMA

Figura 2 - Arquitetura geral do sistema BikeIoT



Fonte: Autor (2026)

O sistema BikeIoT foi estruturado em *firmware*, API, aplicativo móvel e painel administrativo, conforme apresentado na Figura 2. Essa divisão permitiu separar responsabilidades e facilitar o desenvolvimento e validação dos componentes.

O *firmware* embarcado foi responsável pelo controle da trava, leitura de localização, comunicação BLE via NimBLE, envio de telemetria por MQTT e acionamento físico do mecanismo de bloqueio. A API centralizou autenticação, gerenciamento de usuários, geração de assinaturas temporárias, persistência de dados e recebimento de telemetria. O aplicativo móvel permitiu autenticação e envio de comandos de bloqueio e desbloqueio, enquanto o painel administrativo foi utilizado para gerenciamento operacional e configuração do *geofencing*.

A comunicação entre os componentes ocorreu por BLE, MQTT e HTTP. O BLE foi utilizado na comunicação local entre aplicativo e bicicleta devido ao baixo consumo energético. O MQTT foi empregado na troca assíncrona de eventos, publicação de telemetria e distribuição de configurações. Já o HTTP foi utilizado nas interações entre aplicativo, painel administrativo e API.

2.3 DESENVOLVIMENTO DO *FIRMWARE* EMBARCADO

O *firmware* iniciou com uma estrutura básica de comunicação BLE para execução de comandos de bloqueio e desbloqueio. Inicialmente, o estado da bicicleta era controlado logicamente e sinalizado por LED e saídas seriais de depuração. Posteriormente, o *firmware* foi reorganizado em módulos responsáveis por BLE, modem, GPS, MQTT, estado da bicicleta, botão de emergência, alarme e controle da trava.

Na etapa de conectividade, foram avaliadas alternativas utilizando Wi-Fi e armazenamento de credenciais em memória flash. Entretanto, os testes demonstraram aumento de complexidade no gerenciamento simultâneo de Wi-Fi, *fallback* de rede e modem, levando à simplificação da arquitetura com priorização da rede móvel 4G para comunicação MQTT e obtenção de dados de localização.

A comunicação MQTT passou a ser utilizada para o envio periódico de telemetria e o recebimento das configurações de *geofencing*, incluindo estado da bicicleta, usuário atual, coordenadas GPS, última posição válida e vértices do polígono da área permitida.

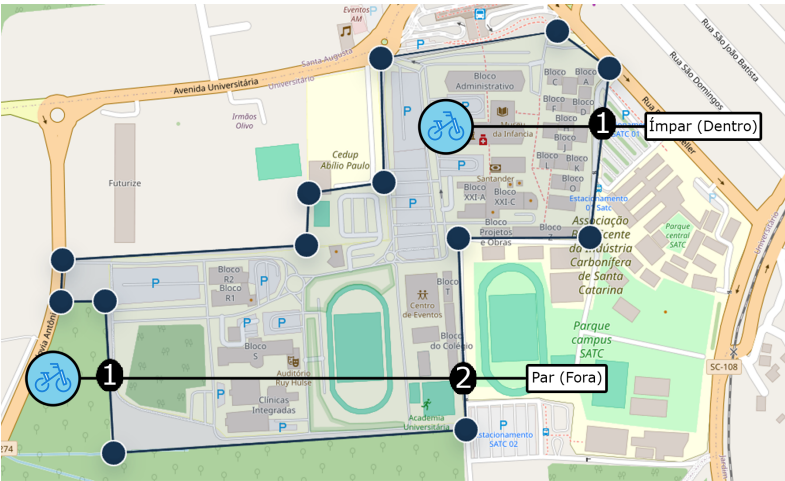
A geolocalização foi incorporada utilizando GPS e recursos auxiliares de A-GPS, sendo observadas dificuldades relacionadas ao tempo

de obtenção de *fix* e à precisão da posição. Como resultado, o *firmware* passou a exigir GPS válido, geofence configurada e posição dentro da área autorizada como condições para desbloqueio.

Para evitar bloqueios na comunicação BLE enquanto aguardava a disponibilidade da localização, o fluxo de desbloqueio foi reorganizado de forma assíncrona. Após a validação inicial do *payload* recebido, a solicitação permanecia pendente até a obtenção de um *fix* válido de GPS, com tentativas periódicas de leitura, limite global de tempo de espera e cancelamento seguro em casos de desconexão do cliente BLE, mudança de estado da bicicleta ou conflito com o processo de pareamento.

Para verificar se a bicicleta estava dentro da área permitida, foi utilizado o algoritmo Ray Casting, também conhecido como regra par-ímpar (*Even-Odd Rule*). O método contabiliza as interseções entre um raio projetado a partir da posição da bicicleta e as arestas do polígono: se o número for ímpar, o ponto é considerado dentro da área; caso contrário, está fora, conforme ilustrado na Figura 3. Também foi realizado tratamento para pontos sobre a borda da geofence por meio da verificação de colinearidade utilizando produto vetorial.

Figura 3 - Exemplificação de regra par-ímpar para geofencing



Fonte: Autor (2026)

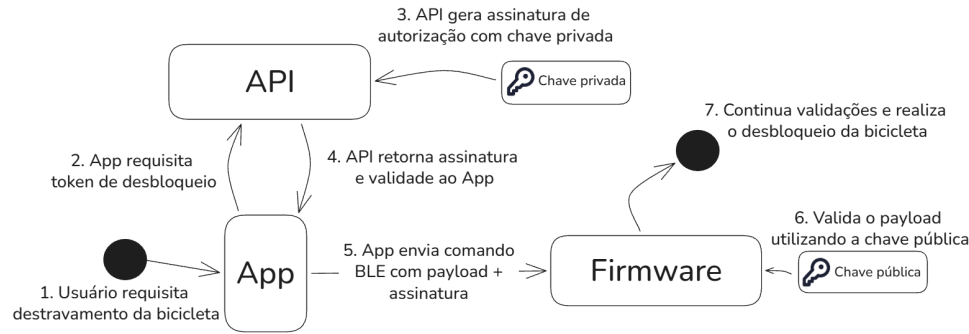
Durante o desenvolvimento, identificou-se que a utilização de comandos BLE sem autenticação adequada representava risco de desbloqueio não autorizado. Para mitigar esse problema, foi implementado um mecanismo de autorização baseado em assinaturas digitais, ilustrado na Figura 4.

Quando o usuário solicita o desbloqueio pelo aplicativo, a API

gera uma autorização assinada utilizando uma chave privada e retorna ao aplicativo um *payload* contendo os dados da solicitação, a assinatura digital e seu período de validade. Em seguida, o aplicativo transmite essas informações ao *firmware* por meio do BLE.

Ao receber a requisição, o *firmware* utiliza a chave pública embarcada para validar a assinatura ECDSA com SHA-256, verificar a validade temporal da autorização e confirmar a integridade dos dados recebidos. Somente após a aprovação dessas verificações o processo de desbloqueio prossegue; caso contrário, a bicicleta permanece bloqueada.

Figura 4 - Fluxo de desbloqueio com validação criptográfica



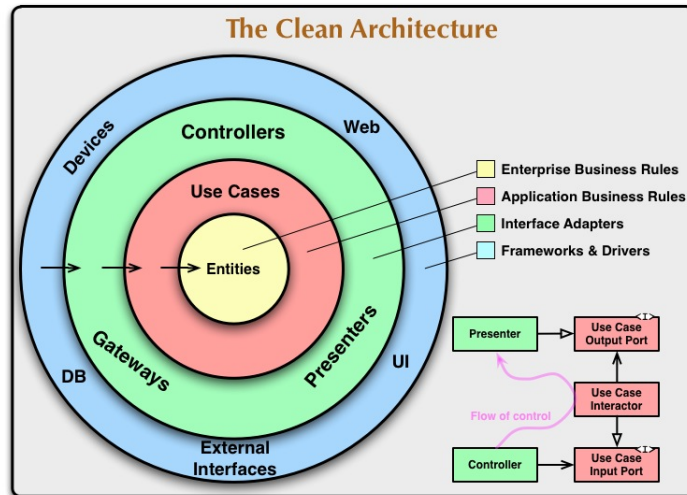
Fonte: Autor (2026)

Paralelamente, o controle da trava evoluiu para atuação mecânica por servo integrado ao mecanismo impresso em 3D. O acionamento foi implementado de forma assíncrona para evitar bloqueios no processamento das tarefas de BLE, MQTT e GPS.

2.4 DESENVOLVIMENTO DA API

A API foi desenvolvida em .NET utilizando ASP.NET Core e organizada em arquitetura em camadas, com separação entre apresentação, regras de negócio, infraestrutura e componentes compartilhados. A estrutura adotou Service Layer em conjunto com Repository Pattern, centralizando a lógica da aplicação e o acesso a dados de forma desacoplada. Essa organização permitiu maior modularização dos componentes, reduzindo o acoplamento entre as camadas e favorecendo a reutilização de código. A comunicação entre os módulos foi realizada por meio de interfaces e injeção de dependência, contribuindo para a testabilidade e manutenção do sistema. A organização foi inspirada nos princípios da Clean Architecture (Figura 5), visando facilitar manutenção, escalabilidade e separação de responsabilidades.

Figura 5 - Conceito arquitetural de Clean Architecture



Fonte: Martin (2012)

A autenticação foi implementada com JWT, permitindo acesso aos recursos de acordo com o perfil do usuário. Foram definidos papéis distintos para usuário comum e administrador.

A assinatura temporária foi implementada para impedir que o aplicativo pudesse liberar a bicicleta de forma autônoma. Nesse fluxo, o aplicativo solicitava à API uma autorização temporária de desbloqueio. A API gerava uma mensagem contendo identificação do usuário e validade temporal, assinava o conteúdo utilizando ECDSA com SHA-256 e retornava a assinatura ao aplicativo. O *firmware* validava a assinatura utilizando a chave pública correspondente antes de aceitar o comando BLE.

A API também foi integrada ao broker MQTT. Os pontos de *geo-fencing* eram publicados em tópico específico utilizando mensagens retidas, mantendo a última configuração disponível ao *firmware* após reconexões. Paralelamente, a API assinava tópicos de telemetria, recebia os dados enviados pelas bicicletas e atualizava os registros persistidos.

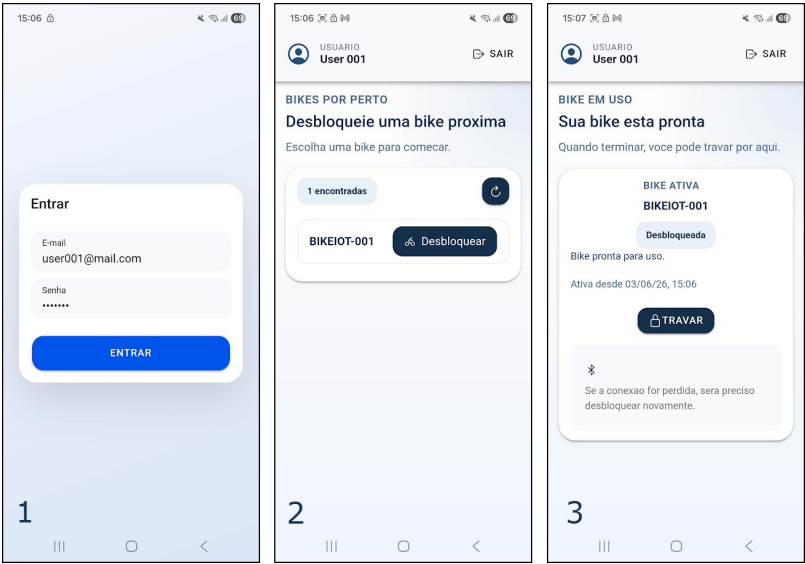
2.5 DESENVOLVIMENTO DO APLICATIVO MÓVEL

O aplicativo móvel foi desenvolvido com Ionic, Angular e Capacitor, permitindo acesso ao BLE do dispositivo. Ele foi responsável por autenticar o usuário, solicitar autorização temporária à API, localizar bicicletas próximas e enviar comandos assinados ao *firmware*.

Após a autenticação, a sessão JWT era armazenada para acesso aos endpoints protegidos. O serviço BLE realizava a inicialização do Bluetooth, descoberta e conexão com dispositivos, além do envio dos comandos.

O fluxo de desbloqueio, ilustrado na Figura 6, consistia na autenticação do usuário, seleção da bicicleta, obtenção ou reutilização de uma autorização temporária válida e envio do comando assinado ao *firmware*. A resposta era interpretada para identificar sucesso, falha de validação, *geofencing* não autorizado ou falha mecânica.

Figura 6 - Fluxo de telas para desbloqueio via aplicativo móvel



Fonte: Autor (2026)

Após o desbloqueio, o aplicativo mantinha uma sessão local associada à bicicleta utilizada, permitindo o envio posterior do comando de bloqueio. Em caso de falha de conexão ou interrupção inesperada, o serviço BLE encerrava a sessão local e limpava os estados pendentes.

2.6 DESENVOLVIMENTO DO PAINEL ADMINISTRATIVO

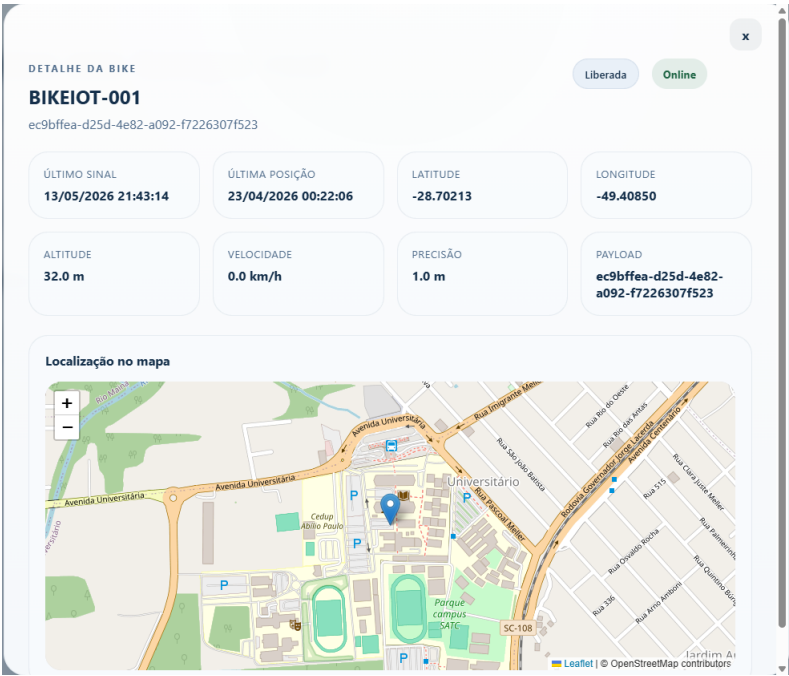
O painel administrativo foi desenvolvido em Angular e integrado à API por meio de requisições HTTP autenticadas com JWT. Sua finalidade metodológica foi permitir configuração e observação do sistema durante os testes. O painel foi utilizado para gerenciamento de usuários, definição da área de *geofencing* e acompanhamento das informações publicadas pelas bicicletas.

A configuração de *geofencing* foi implementada com mapa interativo baseado na biblioteca Leaflet, permitindo definir a área autorizada por meio de cliques e geração de vértices geográficos. Antes do envio à API, a configuração era validada para garantir a existência mínima de três pontos.

O painel também incluiu um menu de monitoramento para exibi-

ção do estado das bicicletas, telemetria, localização e condição de bloqueio (Figura 7), permitindo verificar o correto recebimento e persistência das informações enviadas pelo *firmware*.

Figura 7 - Telemetrias da bicicleta



Fonte: Autor (2026)

2.7 ANÁLISE DOS DADOS

A análise dos dados foi realizada por meio de avaliação técnica dos registros produzidos pelos componentes do sistema. Foram utilizados logs seriais do *firmware*, mensagens MQTT, registros da API, dados persistidos em banco e respostas obtidas durante os testes BLE.

Os dados foram analisados de forma funcional, verificando se a entrada fornecida produzia a saída esperada. No desbloqueio, observou-se se o *firmware* aceitava apenas mensagens assinadas, dentro do prazo de validade, com GPS válido e posição autorizada. No bloqueio, verificou-se se o estado final da bicicleta era atualizado corretamente e se a telemetria refletia o estado real do dispositivo.

As principais métricas avaliadas foram sucesso ou falha na autenticação, validade da assinatura, aceitação ou rejeição do comando BLE, consistência do estado bloqueado/desbloqueado, recebimento de telemetria pela API, validade da configuração de *geofencing* e comportamento diante de desconexões.

A comunicação MQTT foi avaliada pela publicação e retenção

das mensagens de configuração e recebimento das telemetrias pela API. A comunicação BLE foi avaliada pela descoberta do dispositivo, estabelecimento de conexão, envio do comando assinado e recebimento da resposta. A atuação física foi analisada pela correspondência entre comando lógico e movimentação da trava.

A análise final buscou verificar se a arquitetura proposta atendia ao objetivo do trabalho: validar uma solução IoT para aprimorar o controle, utilização e segurança de bicicletas compartilhadas em ambiente universitário. O foco da validação concentrou-se na integração funcional entre autenticação, autorização criptográfica, comunicação local, comunicação remota, geolocalização, *geofencing*, telemetria e atuação física.

3 RESULTADOS E DISCUSSÃO

3.1 RESULTADOS OBTIDOS

O desenvolvimento do BikelIoT resultou em um protótipo funcional de bicicletas compartilhadas baseado em Internet das Coisas, composto por código embarcado, *backend*, aplicativo e página web administrativa. A validação ocorreu em bancada, utilizando uma trava demonstrativa acionada por servo motor, permitindo verificar a integração entre BLE, MQTT sobre rede móvel 4G, autenticação via API HTTP, geolocalização, *geofencing* e controle da trava.

Foram realizados 30 testes distribuídos em sete cenários operacionais, abrangendo autenticação, autorização temporária, desbloqueio, bloqueio remoto, bloqueio por botão físico, sincronização de estados e tratamento de falhas de conectividade e localização. Em todos os cenários, o sistema apresentou comportamento consistente com os requisitos definidos, alcançando taxa de conformidade funcional em todos os testes realizados no ambiente controlado. A descrição individual dos 30 testes, com suas condições de entrada e resultados observados, é apresentada no Apêndice B.

Os mecanismos de autenticação baseados em JWT foram validados entre aplicativo, painel web e API, para usuários comuns e administradores. O controle de acesso operou corretamente, restringindo cada perfil aos recursos autorizados. Nos testes de autorização e desbloqueio, o *firmware* aceitou comandos apenas quando a mensagem possuía assinatura digital válida, janela temporal permitida, GPS válido e posição compatível com a geofence configurada. Em casos de assinatura inválida, mensagem expirada, ausência de GPS válido ou posição não autorizada, o des-

bloqueio foi negado e a bicicleta permaneceu em estado seguro.

Também foram observadas métricas técnicas relacionadas à operação do protótipo. A Tabela 2 apresenta as principais medições utilizadas na análise do sistema.

Tabela 2 - Métricas técnicas observadas no protótipo BikeIOT

Métrica	Valor	Condição de avaliação
Inicialização do modem e registro na rede celular	12,63s	Média das inicializações com registro bem-sucedido na rede
Tempo médio de desbloqueio	13,95s	Intervalo entre envio do comando válido e acionamento completo da trava
Tempo médio de bloqueio pelo aplicativo	12,69s	Intervalo entre comando de bloqueio e retorno ao estado travado
Tempo de autorização BLE	3,37s	Tempo entre autorização e liberação do desbloqueio
Tempo para obter posição do GPS em ambiente aberto	12,91s	Média do primeiro <i>fix</i> válido em ambiente aberto
Precisão média do GPS via GNSS	0,73 m	Média do campo <i>acc</i> reportado pelo módulo GNSS em ensaios estáveis a céu aberto

Fonte: Elaborado pelo autor.

Durante os testes, observou-se que o tempo entre o envio do comando válido e o acionamento completo da trava permaneceu na ordem de 12,86 a 14,8 segundos. Já o tempo inicial para obtenção de GPS válido, em ambiente aberto, variou entre aproximadamente 9,45 e 16,76 segundos, evidenciando a influência direta do contexto físico e da disponibilidade de sinal nas funcionalidades dependentes de localização.

Embora os tempos observados sejam superiores aos esperados, observou-se que a maior parcela da latência está relacionada à obtenção da posição GPS e ao registro na rede móvel inicial, enquanto a comunicação BLE apresentou baixa influência no tempo total de desbloqueio, o qual seria o perceptível pelo usuário.

Nos testes de perda de conectividade MQTT e internet móvel após a inicialização do sistema, BLE, lógica local e controle da trava continuaram operando normalmente. Como a configuração da geofence permanece armazenada localmente em memória, a bicicleta conseguiu manter operação parcial mesmo sem comunicação contínua com a API ou com o *broker* MQTT. Após o restabelecimento da conectividade, o *firmware* rea-

lizou nova sincronização automática de telemetria e estados operacionais em aproximadamente 0,18 segundos.

A lógica de segurança associada à sessão BLE também foi validada. Quando ocorre interrupção inesperada da conexão entre aplicativo e *firmware*, a bicicleta executa bloqueio automático, encerra a sessão ativa e limpa os estados temporários armazenados, impedindo a retomada automática da sessão anterior. Esse comportamento foi reproduzido nos 3 testes destinados a esse cenário, todos concluídos com tratamento seguro da falha.

Nos testes relacionados ao *geofencing*, verificou-se que, ao perder GPS válido durante o uso ou ao ultrapassar a área autorizada, o sistema acionava alerta sonoro por *buzzer* e realizava bloqueio automático após o intervalo configurado de aproximadamente 30 segundos. Esse resultado confirma que a restrição espacial não se limita ao momento do desbloqueio, mas também atua como mecanismo de controle durante a sessão de uso.

De forma geral, os resultados demonstram que o BikelOT foi capaz de integrar autenticação, autorização temporária, comunicação BLE, comunicação MQTT, *geofencing*, telemetria e atuação física da trava em um único protótipo experimental. Ainda que a validação tenha ocorrido em ambiente controlado, o conjunto de testes executados foi suficiente para evidenciar o funcionamento coerente da arquitetura proposta e para identificar, de forma objetiva, os pontos que ainda exigem medições complementares em etapas futuras.

3.2 DISCUSSÃO DOS RESULTADOS

Os resultados obtidos indicam que a arquitetura proposta é tecnicamente viável para aplicações experimentais de bicicletas compartilhadas em ambiente universitário. A principal evidência dessa viabilidade está no fato de que o protótipo apresentou comportamento aderente ao esperado em todos os 30 testes realizados, cobrindo desde cenários nominais de desbloqueio e bloqueio até situações de erro, perda de conectividade e violação espacial. Assim, o estudo demonstra que o sistema responde de maneira previsível e segura diante das condições testadas.

Quando comparado aos trabalhos correlatos centrados em bicicletas compartilhadas, o BikelOT se aproxima de Leme et al. (2021) e Carvalho et al. (2023) no sentido de validar uma arquitetura composta por aplicação cliente, elementos embarcados e comunicação mediada por MQTT. Entretanto, tanto Smart Bike quanto Vá de Bike concentram sua contribui-

ção principalmente na prototipagem e na demonstração de viabilidade da solução, com menor detalhamento experimental de métricas e cenários de falha. Nesse contexto, o BikelOT avança ao explicitar os cenários testados, registrar quantidade de execuções por caso e apresentar evidências funcionais mais sistematizadas, especialmente no que se refere a rejeição de comandos inválidos, consistência de estados e comportamento seguro sob desconexão.

Em relação ao trabalho de Hameed et al. (2021), a aproximação ocorre principalmente no modelo *dockless*, no uso de GPS e na proposta de travamento automatizado controlado por aplicação. Contudo, o Rent-A-Cycle adota uma avaliação predominantemente descritiva, baseada em levantamento de funcionalidades, relato de tentativas de implementação e análise econômica do sistema. O BikelOT, por sua vez, contribui com uma validação técnica mais estruturada sobre autorização, *geofencing*, sincronização entre componentes e tratamento de falhas, o que fortalece a argumentação experimental do protótipo dentro do contexto acadêmico.

O correlato que mais se aproxima metodologicamente da presente pesquisa é o sistema de trava inteligente estudado por Kurniasyah e Rakhmawati (2025), uma vez que esse trabalho também analisa a integração entre ESP32, BLE e MQTT a partir de métricas observáveis e ensaios experimentais. Os autores avaliam, por exemplo, alcance de sinal, impacto de obstruções, comportamento de diferentes níveis de QoS do MQTT e estabilidade de comunicação. Embora o BikelOT ainda não apresente, nesta etapa, um conjunto quantitativo equivalente para parâmetros como *throughput*, atraso, *jitter*, perda de pacotes ou variação de RSSI, a presente seção já estabelece uma base clara para incorporação dessas medições em trabalhos futuros. Assim, o estudo mantém coerência metodológica com esse correlato mais robusto e estabelece uma base clara para aprofundamento experimental posterior. Em comparação aos trabalhos de Leme et al. (2021) e Hameed et al. (2021), o BikelOT diferencia-se principalmente pela integração entre autenticação criptográfica, comunicação BLE, MQTT e controle *dockless*, ainda que sua validação tenha sido restrita ao ambiente controlado.

Do ponto de vista funcional, a utilização de BLE mostrou-se adequada para a comunicação local entre aplicativo e bicicleta, principalmente devido ao baixo consumo energético e à possibilidade de limitar a exposição do dispositivo durante a sessão de uso. Contudo, os testes também evidenciaram que o emprego de comandos BLE simples, sem validação

adicional, representaria risco operacional. Por esse motivo, a adoção de assinaturas temporárias validadas localmente pelo *firmware* constituiu um diferencial importante da proposta, reduzindo a confiança depositada exclusivamente no aplicativo móvel e reforçando o controle embarcado sobre o desbloqueio.

O uso de MQTT também se mostrou adequado para a arquitetura distribuída do sistema, pois permitiu sincronização assíncrona de telemetria e configurações sem exigir comunicação direta contínua entre *firmware* e API. Essa decisão arquitetural revelou-se particularmente relevante nos testes de desconexão, nos quais a operação local permaneceu disponível mesmo sem acesso imediato à infraestrutura remota. Em termos práticos, isso indica que a separação entre lógica crítica embarcada e sincronização remota aumenta a tolerância a falhas temporárias de rede, aspecto importante para soluções *dockless* dependentes de conectividade móvel.

Os testes com GPS e *geofencing* evidenciaram simultaneamente potencial e limitação. Por um lado, a restrição espacial permitiu controlar a operação da bicicleta dentro da área autorizada e reagir com bloqueio automático quando a condição esperada deixava de ser satisfeita. Por outro, a dependência de GPS introduziu variabilidade associada ao tempo de obtenção do primeiro *fix* e à precisão da posição, especialmente em ambientes fechados ou com obstrução parcial do céu. Esses fatores impactam diretamente a experiência de uso e indicam que a validação em campo deverá considerar cenários mais diversos de cobertura e visada.

Outro resultado relevante diz respeito à concorrência entre BLE, MQTT, GPS e atuação mecânica da trava no *firmware* embarcado. Durante o desenvolvimento, observou-se que abordagens baseadas em operações bloqueantes comprometiam a responsividade do sistema e podiam afetar negativamente a comunicação BLE. A reorganização modular do *firmware* e o uso de acionamentos assíncronos reduziram esses efeitos, contribuindo para maior estabilidade operacional. Embora esse aspecto não tenha sido expresso por meio de uma métrica numérica específica nesta etapa, ele representa um achado técnico importante do estudo, pois evidencia a necessidade de coordenação cuidadosa entre múltiplos subsistemas embarcados em aplicações IoT móveis.

Por fim, os resultados precisam ser interpretados à luz das limitações do estudo. A validação ocorreu em ambiente controlado, com estrutura demonstrativa e número restrito de execuções por cenário, o que

impede generalizações diretas sobre comportamento em uso contínuo, com múltiplos usuários ou em operação pública real. Também não foram realizadas, nesta fase, avaliações de escalabilidade, utilização simultânea por múltiplos usuários, operação prolongada em ambiente real, medições quantitativas aprofundadas de desempenho de rede, qualidade de sinal BLE, consumo energético, usabilidade ou confiabilidade de longo prazo.

4 CONCLUSÃO

O presente trabalho teve como objetivo validar uma solução baseada em Internet das Coisas para aprimorar o controle, utilização e segurança de bicicletas compartilhadas em ambiente universitário. Para isso, foi desenvolvido um sistema *dockless* composto por *firmware*, *backend*, aplicativo móvel e painel administrativo, substituindo o uso de chaves físicas por uma arquitetura digital integrada baseada em BLE, MQTT, GPS, *geofencing* e autenticação criptográfica.

Os resultados demonstraram a viabilidade técnica da arquitetura proposta. O protótipo validou a integração entre comunicação local via BLE, comunicação remota por MQTT sobre rede móvel 4G, autenticação via API HTTP, controle físico da trava e monitoramento remoto por telemetria. Também foi validado o uso de autorizações temporárias assinadas digitalmente, permitindo que apenas comandos autenticados e válidos fossem aceitos pelo *firmware*.

A integração entre GPS e *geofencing* permitiu restringir o uso da bicicleta a áreas autorizadas, enquanto as mensagens MQTT retidas possibilitaram a recuperação automática das configurações após reconexões, contribuindo para a continuidade operacional do sistema.

Como principal contribuição, o trabalho apresentou uma arquitetura experimental integrada para gerenciamento de bicicletas compartilhadas baseada em IoT, reunindo comunicação BLE, telemetria MQTT, autenticação JWT, validação criptográfica embarcada e *geofencing* em um único protótipo funcional, organizado de forma modular para facilitar sua evolução.

O estudo apresentou limitações. A validação foi realizada em ambiente controlado, utilizando uma trava demonstrativa impressa em 3D, sem exposição prolongada a condições reais de uso, vibração, clima ou tentativas de violação. Também não foram avaliados quantitativamente o consumo energético do sistema, os impactos da dependência da conectividade 4G e do tempo de obtenção de *fix* do GPS. Além disso, a trava utilizou

um servo motor sem realimentação de posição, podendo ocorrer dessincronização entre a posição física e o estado lógico após uso prolongado. Essa limitação, entretanto, não comprometeu os objetivos de validação deste trabalho.

Como trabalhos futuros, recomenda-se validar o sistema em bicicletas reais e em cenários operacionais contínuos, incluindo análises de consumo energético, tolerância a falhas, auditoria, proteção contra reuso de comandos e testes de usabilidade com usuários reais. Também são sugeridos mecanismos de trava com realimentação de posição, rotação de chaves criptográficas, monitoramento avançado, histórico operacional e integração com soluções antifurto mais robustas.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que ferramentas de Inteligência Artificial generativa foram utilizadas exclusivamente como apoio auxiliar à elaboração deste trabalho, incluindo assistência na revisão linguística, estruturação textual e organização de ideias. Nenhuma ferramenta de IA foi utilizada para substituição da autoria intelectual, análise científica, interpretação dos resultados ou tomada de decisões acadêmicas. A responsabilidade integral pelo conteúdo, originalidade, veracidade das informações e conformidade com os princípios de integridade científica permanece integralmente atribuída aos autores.

REFERÊNCIAS

AL-SHAREEDA, M. A. et al. **Bluetooth low energy for internet of things: Review, challenges, and open issues**. Indonesian Journal of Electrical Engineering and Computer Science, v. 31, n. 2, p. 1182–1189. 2023.

Amazon Web Services, Inc. **What is EDA (Event-Driven Architecture)?** 2024. Disponível em: <<https://aws.amazon.com/pt/what-is/eda/>>. Acesso em: 17 nov. 2025.

ASHTON, K. **That ‘Internet of Things’ thing**. RFID Journal. 2009.

BARBERIO, L. D. G.; MANOCCHIO, F. G. S.; GALLO, F. **Micromobilidade urbana e atuação de agentes financeiros: o caso da empresa tembici (banco itaú) na cidade de são paulo, brasil**. Geografares, Universidade Federal do Espírito Santo, n. 37. 2023, Disponível em: <<https://journals.openedition.org/geografares/10639>>. Acesso em: 15 out. 2025.

CARVALHO, J. G. A. et al. **Vá de bike: Plataforma iot para aluguel de bicicletas.** Anais do XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais. 2023.

CHICCO, A.; DIANA, M. **Understanding micro-mobility usage patterns: a preliminary comparison between dockless bike sharing and e-scooters in the city of turin (italy).** Transportation Research Procedia, v. 62, p. 459–466. ISSN 2352-1465. 2022, Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352146522001843>. Acesso em: 6 jun. 2026.

COHEN, A. et al. **The impact of bicycle theft on ridership behavior.** International Journal of Sustainable Transportation, Taylor & Francis, v. 18, n. 5, p. 453–463. 2024.

CUI, C.; ZHANG, Y. **Integration of shared micromobility into public transit: A systematic literature review with grey literature.** Sustainability, v. 16, n. 9, p. 3557. ISSN 2071-1050. 2024, Disponível em: <https://www.mdpi.com/2071-1050/16/9/3557>. Acesso em: 6 jun. 2026.

DIOGO, T. S.; PINTO, L. F. R.; RIBEIRO, A. P. **Desafios do sistema de bicicletas compartilhadas.** Revista Políticas Públicas & Cidades, v. 14, n. 2, p. e1834. 2025, Disponível em: <https://journalppc.com/RPPC/article/view/1834>. Acesso em: 6 jun. 2026.

Espressif Systems. **ESP32 Hardware Design Guidelines.** [S.l.], 2025. Disponível em: <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/index.html>. Acesso em: 17 nov. 2025.

HAMEED, S. et al. **Rent-a-cycle (smart bicycle sharing service-iot based).** Journal of Robotics and Mechanical Engineering, Coalesce Research Group, v. 1, n. 1, p. 1–10. 2021.

HiveMQ. **What is MQTT Quality of Service (QoS) 0,1, & 2? – MQTT Essentials: Part 6.** 2015. Disponível em: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>. Acesso em: 17 nov. 2025.

HMISSI, F.; OUNI, S. **An MQTT Brokers Distribution Based on Mist Computing for Real-Time IoT Communications.** 2021. Disponível em: https://www.researchgate.net/publication/353232159_An_MQTT_Brokers_Distribution_Based_on_Mist_Computing_for_Real-Time_IoT_Communications/. Acesso em: 17 nov. 2025.

KURNIASYAH, I. M.; RAKHMAWATI, L. **Smart door lock innovation using integration of bluetooth low energy and mqtt iot protocol.** Indonesian Journal of Electrical and Electronics Engineering (INAJEEE), v. 8, n. 1, p. 41–46. 2025.

LAGHARI, S. U. A. et al. **Securing mqtt ecosystem: Exploring vulnerabilities, mitigations, and future trajectories**. IEEE Access. 2024, Disponível em: <<https://ieeexplore.ieee.org/document/10552255/>>. Acesso em: 17 nov. 2025.

LEME, T. R. et al. **Smart bike: Sistema de aluguel de bicicletas**. In: SBRT. **Anais do XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)**. [S.l.], 2021. 2021.

MARTIN, R. C. **The Clean Architecture**. 2012. Disponível em: <<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>>. Acesso em: 4 maio 2026.

MARTINS, J. V.; GONZALES-TACO, P. **Mobilidade urbana no contexto de cidades inteligentes: uma análise bibliométrica e de conteúdo**. Procesos Urbanos, v. 7, p. e497. 2020, Disponível em: <https://www.researchgate.net/publication/347100883_Mobilidade_urbana_no_contexto_de_cidades_inteligentes_uma_analise_bibliometrica_e_de_conteudo>. Acesso em: 17 nov. 2025.

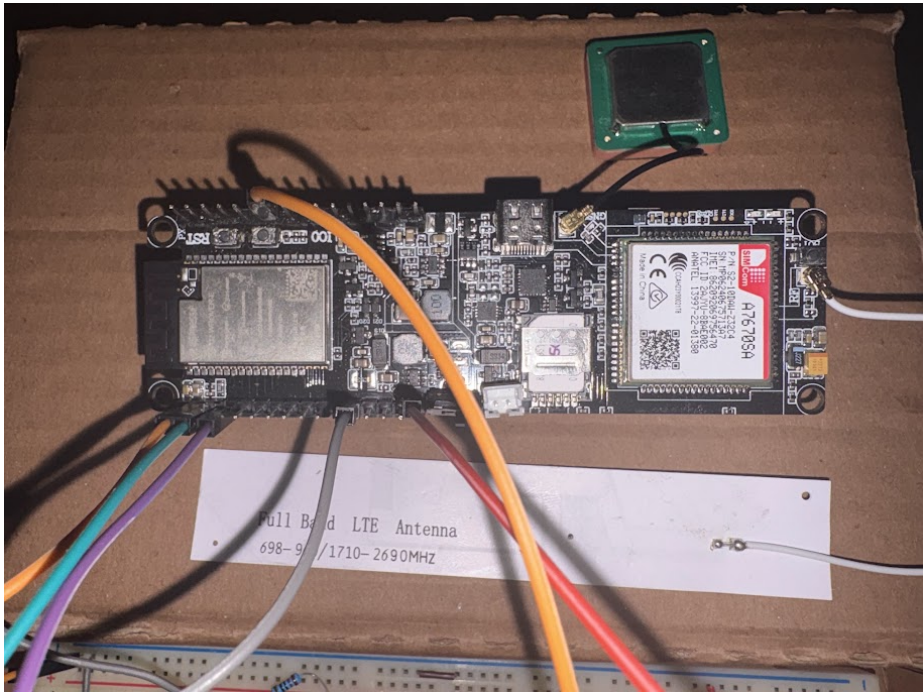
MQTT.org. **MQTT FAQ**. [S.l.], 2025. Disponível em: <<https://mqtt.org/faq/>>. Acesso em: 17 nov. 2025.

TAVARES, M. S. et al. **Health at risk: Air pollution and urban vulnerability—perspectives in light of the 2030 agenda**. Green Health, v. 1, n. 3, p. 21. ISSN 3042-5832. 2025, Disponível em: <<https://www.mdpi.com/3042-5832/1/3/21>>. Acesso em: 6 jun. 2026.

ZIMMERMANN, K.; PALGAN, Y. V. **Upscaling cargo bike sharing in cities: A comparative case study**. Journal of Cleaner Production, v. 477, p. 143774. ISSN 0959-6526. 2024, Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0959652624032232>>. Acesso em: 6 jun. 2026.

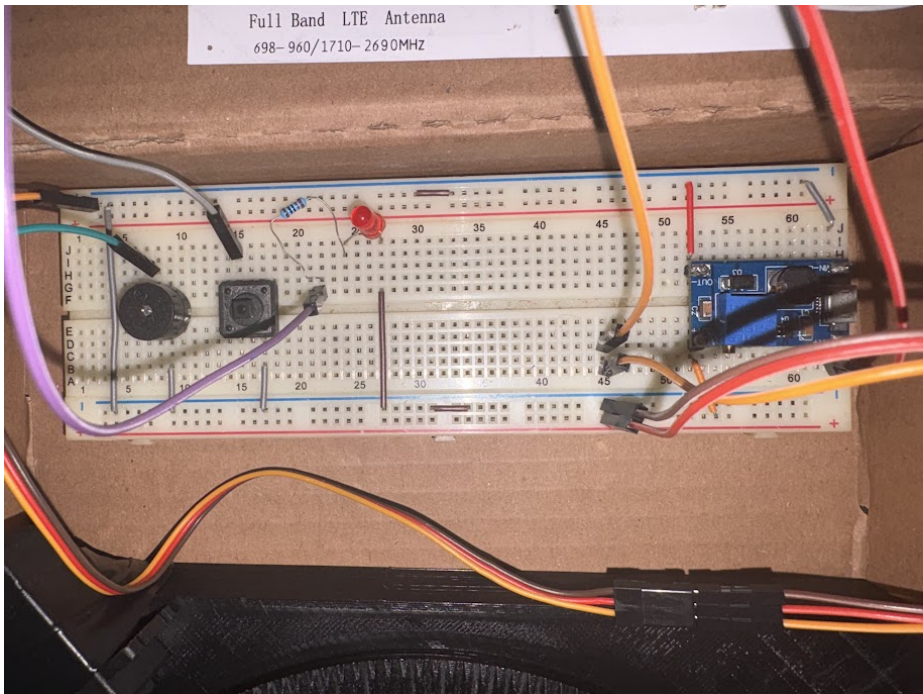
APÊNDICE A - FOTOS DO HARDWARE DO PROJETO

Figura A.1 - Protótipo físico desenvolvido para validação do sistema BikeIoT



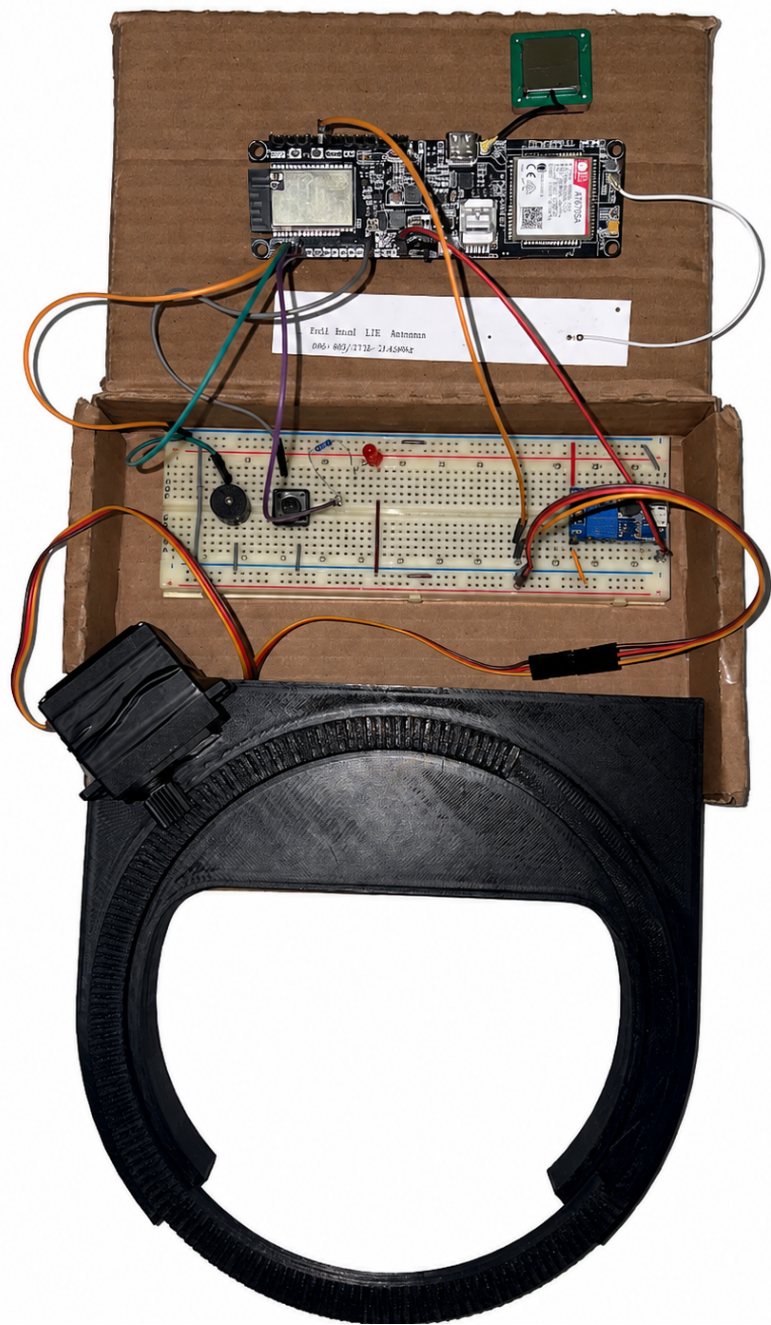
Fonte: Autor (2026).

Figura A.2 - Detalhamento dos componentes eletrônicos utilizados no protótipo



Fonte: Autor (2026).

Figura A.3 - Visão geral do protótipo completo desenvolvido para validação funcional do projeto



Fonte: Autor (2026).

APÊNDICE B - DETALHAMENTO DOS 30 TESTES REALIZADOS

Tabela B.1 - Resultados dos testes realizados no protótipo

Teste	Cenário	Condição avaliada	Valor observado	Resultado observado
1	Sincronização de estados	Inicialização do modem e registro na rede celular	12,63 s em média	Comunicação remota estabelecida com sucesso
2	Autenticação	Login de administrador no painel com credenciais válidas	JWT emitido	Acesso autorizado aos recursos administrativos
3	Sincronização de estados	Definição e envio inicial da geofence pelo painel administrativo	Geofence válida publicada	Configuração operacional disponibilizada ao firmware
4	Sincronização de estados	Recebimento da configuração de geofencing via MQTT retido	Última configuração disponível após reconexão	Firmware recuperou a configuração sem re-envio manual
5	Autenticação	Login de usuário comum no aplicativo com credenciais válidas	JWT emitido	Acesso autorizado aos recursos do perfil de usuário comum
6	Autenticação	Acesso do usuário comum a recurso restrito de administrador	JWT sem privilégio administrativo	Requisição rejeitada e acesso restrito corretamente
7	Autenticação	Uso de token inválido ou ausente em endpoint protegido	JWT inválido ou ausente	Requisição negada pela API
8	Autorização temporária	Solicitação de autorização temporária de desbloqueio para bicicleta válida	Assinatura ECDSA emitida	Payload assinado retornado ao aplicativo com validade temporal
9	Autorização temporária	Validação local da autorização com chave pública embarcada	SHA-256 + ECDSA válidos	Comando liberado apenas após verificação criptográfica completa

Continua na próxima página

Teste	Cenário	Condição avaliada	Valor observado	Resultado observado
10	Autorização temporária	Reutilização de autorização temporária ainda dentro da janela válida	Janela temporal permitida	Assinatura aceita para novo envio ao firmware
11	Autorização temporária	Envio de payload com integridade alterada após assinatura	Conteúdo inconsistente	Validação rejeitada pelo firmware
12	Autorização temporária	Envio de mensagem com assinatura digital inválida	Assinatura inválida	Desbloqueio negado e bicicleta mantida em estado seguro
13	Autorização temporária	Envio de mensagem com autorização expirada	Mensagem expirada	Desbloqueio negado por validade temporal vencida
14	Desbloqueio	Obtenção do primeiro <i>fix</i> em ambiente aberto durante desbloqueio	12,91 s em média; faixa de 9,45 a 16,76 s	Fluxo prosseguiu somente após validação da localização
15	Desbloqueio	Desbloqueio com assinatura válida, GPS válido e posição autorizada	13,95 s em média	Trava liberada com sucesso e estado atualizado
16	Desbloqueio	Desbloqueio nominal com menor tempo observado em ensaio válido	12,86 s	Acionamento concluído dentro da faixa observada
17	Desbloqueio	Desbloqueio nominal com maior tempo observado em ensaio válido	14,8 s	Acionamento concluído sem inconsistência de estado
18	Desbloqueio	Solicitação de desbloqueio sem GPS válido disponível	Sem <i>fix</i> válido	Desbloqueio negado até a obtenção de posição válida
19	Desbloqueio	Solicitação de desbloqueio com posição fora da geofence configurada	Fora da área autorizada	Desbloqueio negado e bicicleta mantida bloqueada
20	Sincronização de estados	Publicação de telemetria após operação de desbloqueio	Estado, usuário e coordenadas enviados	API recebeu e persistiu os dados corretamente

Continua na próxima página

Teste	Cenário	Condição avaliada	Valor observado	Resultado observado
21	Falhas de conectividade e localização	Perda de conectividade MQTT após inicialização do sistema	Operação local mantida	BLE, lógica local e controle da trava continuaram funcionais
22	Sincronização de estados	Restabelecimento de conectividade com sincronização automática	0,18 s	Telemetria e estados operacionais foram resincronizados
23	Falhas de conectividade e localização	Interrupção inesperada da conexão BLE durante sessão ativa	Desconexão de cliente BLE	Bloqueio automático e limpeza dos estados temporários
24	Falhas de conectividade e localização	Perda de GPS válido durante o uso da bicicleta	GPS inválido em sessão ativa	Alerta sonoro por <i>buzzer</i> e bloqueio automático após cerca de 30 s
25	Falhas de conectividade e localização	Ultrapassagem da área autorizada pela geofence durante o uso	Saída da geofence + precisão média de 0,73 m	Alerta sonoro e bloqueio automático após cerca de 30 s
26	Bloqueio remoto	Comando de bloqueio enviado pelo aplicativo após sessão ativa	12,69 s em média	Bicicleta retornou ao estado travado corretamente
27	Bloqueio remoto	Verificação de atualização de estado após bloqueio remoto	Estado bloqueado + telemetria publicada	Consistência entre comando lógico e estado persistido
28	Bloqueio remoto	Verificação de persistência após bloqueio por aplicativo	Telemetria sincronizada	API recebeu e registrou o novo estado da bicicleta
29	Bloqueio por botão físico	Acionamento da trava emergencial por botão físico local	Evento local de bloqueio	Bicicleta bloqueada com atualização de estado seguro
30	Bloqueio por botão físico	Validação de prioridade do bloqueio físico sobre uso em andamento	Comando local imediato	Sistema priorizou o bloqueio e encerrou a sessão ativa

Fonte: Autor (2026).