



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE  
CONTROLE E AUTOMAÇÃO

HENRIQUE CAMPOS DE LACERDA

**SISTEMA DE RASTREAMENTO DE CARGAS TRANSPORTADAS POR VIA  
TERRESTRE**

Recife  
2024

HENRIQUE CAMPOS DE LACERDA

**SISTEMA DE RASTREAMENTO DE CARGAS TRANSPORTADAS POR VIA  
TERRESTRE**

Trabalho de Conclusão de Curso apresentado  
ao Departamento de Engenharia Elétrica da  
Universidade Federal de Pernambuco, como  
requisito parcial para obtenção do grau de  
Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Douglas Contente Pimentel Barbosa

Recife  
2024

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Lacerda, Henrique Campos de.

Sistema de rastreamento de cargas transportadas por via terrestre / Henrique Campos de Lacerda. - Recife, 2024.

43 : il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2024.

Inclui referências, apêndices.

1. NPS. 2. Arduino UNO. 3. NEO-6M. 4. SIM900. 5. GSM. 6. ThinkSpeak. 7. HTTP. I. Pimentel Barbosa, Douglas Contente. (Orientação). II. Título.

620 CDD (22.ed.)

HENRIQUE CAMPOS DE LACERDA

**SISTEMA DE RASTREIO DE CARGAS TRANSPORTADAS POR VIA  
TERRESTRE**

Trabalho de Conclusão de Curso apresentado  
ao Departamento de Engenharia Elétrica da  
Universidade Federal de Pernambuco, como  
requisito parcial para obtenção do grau de  
Engenheiro de Controle e Automação.

Aprovado em: 20/03/2024

**BANCA EXAMINADORA**

---

Prof. Dr. Douglas Contente Pimentel Barbosa (Orientador)  
Universidade Federal de Pernambuco

---

Prof. Artur Muniz Szpak Furtado (Examinador Interno)  
Universidade Federal de Pernambuco

---

MSc. Eduardo Augusto Oliveira Barbosa (Examinador Interno)  
Universidade Federal de Pernambuco

Dedico este trabalho a todas as pessoas que construíram quem sou hoje.

Em especial, minha mãe que sempre batalhou muito para que eu tivesse a oportunidade de desfrutar das melhores escolas, que possibilitou estudar e concluir um curso superior na UFPE.

## AGRADECIMENTOS

Agradeço a minha mãe, Cristiane, por todo incentivo dado para os estudos durante toda a minha vida, foi a educação nos trouxe aqui, mãe. Agradeço também a minha irmã, Camile, pessoa que sempre me orientou a respeito da vida acadêmica e profissional e me incentivou durante toda a jornada na UFPE. A minha namorada, Vitória, por ter sido um suporte emocional indispensável durante todos os momentos de inquietação durante a graduação que possibilitou que esse ciclo se encerrasse. Vocês são as pessoas que me deram o incentivo para que essa etapa fosse concluída. Por fim, do núcleo familiar, agradeço ao meu pai, Henrique e ao meu tio, Wirandé, se não fosse por eles seria muito mais difícil estar aqui.

Dos meus amigos de curso, a quem levarei para o resto da vida, agradeço por todos os momentos de estudo em conjunto, dominós jogados e encontros em bares realizados. Vocês tornaram essa jornada mais leve e divertida, agradeço a todos: Caio, Lucas, André, Heitor, João, Leonardo, Mauro, Gabriel, Paulo, Rodrigo, Victor, Augusto, Denis, Arthur e Filipe.

Daqueles que levo no coração desde o colégio que foram indispensáveis na construção de quem sou, agradeço a vocês: Felipe, Leandro, Adriel, Daniel, Otávio, Guilherme, Marcelo, Caio, Paulo e Claro.

Aos meus amigos de trabalho, aos quais tenho o prazer de dividir a maior parte do dia e que sempre incentivaram os meus estudos, agradeço demais a: João, Erick, Neubert, Daniel, Cynthia, Danielle, Priscilla, Hélio e Saulo.

Por fim, obrigado a todos os professores aos quais tive a honra de ser aluno, o conhecimento fornecido por cada um de vocês foi imprescindível. Em especial, agradeço ao professor Douglas por ter aceitado me orientar na escrita desse TCC e pelas aulas lecionadas durante o período da pandemia da disciplina de Sistemas supervisórios 2, que me deram motivação para seguir em frente. Ao professor Geraldo Maia, que no final do ano de 2019 me aconselhou com as seguintes palavras: “Caso seja impossível ficar claro o que você quer para sua vida, avance na estrada que você está até ficar claro! Será uma questão de tempo! Há tempo para tudo!” e hoje, professor, o caminho está bem mais claro do que antes.

Deus é todo mundo sorrindo ao mesmo tempo (Parteum)

## RESUMO

Através da pesquisa de NPS (*Net Promoter Score*) realizada mensalmente numa empresa foi verificado que uma das maiores insatisfações dos clientes era a falta de previsibilidade quanto a chegada da carga em suas instalações. Com isso, surgiu a oportunidade do desenvolvimento de um dispositivo que realizasse o rastreamento em tempo real do veículo que está transportando a carga e enviasse essas informações para um servidor na nuvem. Para isso, foi utilizada uma placa de Arduino UNO com dois módulos acessórios, o NEO-6M que é responsável por obter através a geolocalização do dispositivo e o SIM900, módulo de GSM (*Global System for Mobile Communications*) que será responsável por transmitir essas informações para um banco de dados na nuvem. O local escolhido para realizar o armazenamento dos dados foi a plataforma ThingSpeak, que permite a inserção e aquisição de dados através do protocolo HTTP (*Hypertext Transfer Protocol*).

**Palavras-chave:** NPS; Arduino UNO; NEO-6M; SIM900; GSM; *ThingSpeak*; HTTP.



## ABSTRACT

A Monthly NPS (*Net Promoter Score*) survey, conducted within a company, revealed a significant source of customer dissatisfaction: unpredictable cargo arrival times at their facilities. This discovery presented an opportunity to address this issue by developing a real-time tracking solution for transporting goods vehicles. To achieve this, an Arduino UNO board was employed in conjunction with two accessory modules: the NEO-6M for geolocation and the SIM900 GSM (*Global System for Mobile Communications*) module for transmitting data to a cloud server. The chosen platform for data storage was ThingSpeak, facilitating data insertion and retrieval via the HTTP (*Hypertext Transfer Protocol*) protocol.

**Keywords:** NPS; Arduino UNO; NEO-6M; SIM900; GSM; ThingSpeak; HTTP.

## LISTA DE FIGURAS

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| Figura 1 - Arduino UNO.....                                                                 | 17 |
| Figura 2 - Arduino IDE.....                                                                 | 18 |
| Figura 3 - Arquitetura da solução .....                                                     | 24 |
| Figura 4 - Arduino UNO utilizado.....                                                       | 25 |
| Figura 5 - Módulo NEO-6M .....                                                              | 26 |
| Figura 6 - Módulo SIM900 .....                                                              | 27 |
| Figura 7 - Arquitetura final da solução.....                                                | 29 |
| Figura 8 - Esquemático ligação física.....                                                  | 29 |
| Figura 9 - Ligação física .....                                                             | 30 |
| Figura 10 - Cabo USB A/B .....                                                              | 31 |
| Figura 11 - Fonte de alimentação 9V-1A.....                                                 | 31 |
| Figura 12 - Sistema completo com alimentação a rede elétrica.....                           | 32 |
| Figura 13 - Configuração bateria adaptador para alimentar o sistema.....                    | 33 |
| Figura 14 - Alimentação do dispositivo em cenário real de uso .....                         | 33 |
| Figura 15 - Informações registradas no banco de dados no teste residencial .....            | 35 |
| Figura 16 - Informações registradas no banco de dados no teste em cenário real de uso ..... | 36 |

## LISTA DE TABELAS

|                                                   |    |
|---------------------------------------------------|----|
| Tabela 1 - <i>Datasheet</i> Arduino.....          | 18 |
| Tabela 2 - Legenda campos do banco de dados ..... | 34 |

## LISTA DE ABREVIATURAS E SIGLAS

|      |                                                      |
|------|------------------------------------------------------|
| AT   | <i>Attention</i>                                     |
| GPRS | <i>General Packet Radio Service</i>                  |
| GPS  | <i>Global Positioning System</i>                     |
| GSM  | <i>Global System for Mobile Communications</i>       |
| HTTP | <i>Hypertext Transfer Protocol</i>                   |
| IP   | <i>Internet Protocol</i>                             |
| NPS  | <i>Net Promoter Score</i>                            |
| PPS  | <i>Precise Positioning Service</i>                   |
| SMS  | <i>Short Message Service</i>                         |
| SPS  | <i>Standard Positioning Service</i>                  |
| TCP  | <i>Transmission Control Protocol</i>                 |
| UART | <i>Universal Asynchronous Receiver / Transmitter</i> |
| UDP  | <i>User Datagram Protocol</i>                        |
| USB  | <i>Universal Serial Bus</i>                          |

## LISTA DE SÍMBOLOS

|   |        |
|---|--------|
| A | Ampere |
| V | Volt   |

## SUMÁRIO

|          |                                                     |           |
|----------|-----------------------------------------------------|-----------|
| <b>1</b> | <b>INTRODUÇÃO .....</b>                             | <b>14</b> |
| 1.1      | OBJETIVOS .....                                     | 15        |
| 1.1.1    | Geral.....                                          | 15        |
| 1.1.2    | Específicos .....                                   | 15        |
| 1.2      | ORGANIZAÇÃO DO TRABALHO.....                        | 15        |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA .....</b>                  | <b>16</b> |
| 2.1      | MICROCONTROLADORES .....                            | 16        |
| 2.1.1    | Comunicação serial .....                            | 19        |
| 2.2      | GPS.....                                            | 19        |
| 2.2.1    | Funcionamento do GPS .....                          | 20        |
| 2.3      | REDES MÓVEIS .....                                  | 20        |
| 2.3.1    | GPRS .....                                          | 20        |
| 2.4      | COMANDOS AT .....                                   | 21        |
| 2.5      | BANCO DE DADOS NA NUVEM .....                       | 22        |
| 2.6      | PROTOCOLO HTTP .....                                | 22        |
| <b>3</b> | <b>DESENVOLVIMENTO DO TRABALHO .....</b>            | <b>24</b> |
| 3.1      | DEFINIÇÃO DA ARQUITETURA DA SOLUÇÃO .....           | 24        |
| 3.2      | ESCOLHA DO HARDWARE E BANCO DE DADOS.....           | 24        |
| 3.2.1    | Arduino UNO .....                                   | 25        |
| 3.2.2    | Módulo GPS .....                                    | 25        |
| 3.2.3    | Módulo GPRS.....                                    | 27        |
| 3.2.4    | Banco de dados.....                                 | 28        |
| 3.2.5    | Arquitetura final da solução .....                  | 28        |
| 3.3      | LIGAÇÃO FÍSICA .....                                | 29        |
| 3.4      | SISTEMA DE ALIMENTAÇÃO .....                        | 30        |
| 3.4.1    | Alimentação para testes residenciais .....          | 30        |
| 3.4.2    | Alimentação no cenário real de uso.....             | 32        |
| 3.5      | TESTES.....                                         | 34        |
| 3.5.1    | Ambiente Residencial .....                          | 34        |
| 3.5.2    | Dentro de um veículo em locomoção .....             | 36        |
| <b>4</b> | <b>CONCLUSÕES E PROPOSTAS DE CONTINUIDADE .....</b> | <b>38</b> |
|          | <b>REFERÊNCIAS.....</b>                             | <b>40</b> |
|          | <b>APÊNDICES.....</b>                               | <b>41</b> |

## 1 INTRODUÇÃO

A satisfação do cliente é um dos fatores cruciais para o sucesso de qualquer empresa. A capacidade de atender e superar as expectativas dos clientes é um diferencial competitivo para sobrevivência no mercado. Um dos métodos para perceber como está o relacionamento do cliente com uma marca é através da pesquisa de NPS, que é um indicador-chave essencial para empresas que desejam entender melhor o feedback dos clientes e melhorar continuamente seus serviços (REICHHELD, 2003). A pesquisa NPS envolve, pelo menos, uma pergunta central que indaga a pessoa entrevistada sobre a probabilidade de recomendar a empresa em questão a outras pessoas, além disso podem ser realizadas outras perguntas acessórias de determinado serviço prestado pela empresa para captar mais informações do entrevistado.

Em posse desses dados, é possível analisar a percepção que os clientes têm sobre a empresa possibilitando a tomada de decisão de modo a reduzir ou eliminar as dores e potencializar as qualidades. Se sabe que o NPS é um indicador que traz certa previsibilidade a respeito do aumento das vendas de uma empresa (CZARNECKI, 2021).

Com o mundo cada vez mais conectado, a busca de informações em tempo real se tornou parte de nossas vidas, fato que também se expande para o contexto empresarial, onde os clientes buscam informações sobre o serviço prestado de forma constante. Com base na pesquisa de NPS realizada em uma empresa de Pernambuco que vende e entrega vidros planos para o mercado de construção civil, os clientes relataram que a falta de rastreio do veículo que faz a entrega do produto é um ponto muito negativo. Com isso, surge a oportunidade de melhoria do serviço prestado com uso de tecnologia e engenharia.

Nesse contexto, este trabalho tem como objetivo aliviar as preocupações do cliente, utilizando tecnologia para aprimorar sua satisfação.

## **1.1 Objetivos**

### **1.1.1 Geral**

Desenvolver um dispositivo que seja capaz de obter a geolocalização de um veículo e enviar as informações de latitude, longitude, horário e *link* do *google maps* indicando a posição do dispositivo para um servidor na nuvem.

### **1.1.2 Específicos**

- Identificar módulo de geolocalização compatível com arduino;
- Localizar hardware para envio de informações para um banco de dados na nuvem que se conecte com um arduino;
- Encontrar banco de dados na nuvem de baixo custo e fácil utilização.

## **1.2 Organização do Trabalho**

Este trabalho está dividido em quatro capítulos. Neste capítulo, é realizada uma breve introdução sobre o trabalho e apresentados os objetivos.

No Capítulo 2 será realizada a fundamentação teórica, onde serão descritos os temas estudados para construção da solução.

Já no Capítulo 3 será apresentado o desenvolvimento da solução. Onde será descrito quais hardwares foram utilizados e a arquitetura da solução.

Por fim, no Capítulo 4, estarão a conclusão e a proposta de trabalhos futuros para melhoria da solução.



## 2 FUNDAMENTAÇÃO TEÓRICA

Nas sessões abaixo, estão exibidos em tópicos uma explicação teórica a respeito dos temas que sustentam esse trabalho, de forma a permitir a total compreensão do sistema que foi projetado. Os tópicos explicaram conceitos básicos sobre o hardware escolhido para a solução, em seguida uma breve explicação sobre o GPS (*Global Positioning System*), redes móveis, comandos AT (*Attention*), alguns conceitos do protocolo HTTP e banco de dados.

### 2.1 Microcontroladores

Para construir a solução proposta é necessário a utilização de um microcontrolador. Microcontroladores são dispositivos eletrônicos integrados que contêm um processador, memória e periféricos de entrada/saída em um único chip. São projetados para executar tarefas específicas que são desenvolvidas através de linguagens de programação. A eficiência de espaço, consumo de energia e custo são importantes na escolha do microcontrolador adequado para um projeto. Atualmente existem algumas opções de microcontroladores que são utilizados amplamente, sendo eles:

- Arduino;
- Raspberry Pi;
- ESP8266;
- STM32.

Será avaliado quatro quesitos cruciais para a escolha do microcontrolador para este sistema, sendo eles: Facilidade de uso, consumo de energia, custo e potência de processamento.

- Facilidade no uso: Dentre as opções acima o que possui a maior facilidade no uso é a plataforma do Arduino, que é altamente indicado para pessoas que desejam desenvolver seus primeiros projetos com microcontroladores;

- Consumo de energia: Nesse quesito, o ESP8266 é superior devido ao modo de baixa potência que permite uma alta economia de bateria;
- Custo: Para esse ponto, o Arduino e o ESP8266 são os mais recomendados visto que possuem o mesmo custo e estão na menor faixa de preço dentre os quatro citados;
- Potência de processamento: O Raspberry PI dispõe de maior poder de processamento que os demais.

Para o contexto deste trabalho o mais recomendado é a plataforma do Arduino, pois além de ter a maior facilidade de uso e um bom custo em comparação com os demais microcontroladores, possui a maior quantidade de conteúdo disponível para estudo e conta com uma amplo arsenal de módulos acessórios.

O Arduino possui algumas possibilidades de modelos, e optou-se por utilizar o Arduino UNO pelo fato de o autor deste trabalho já ter trabalhado com esse modelo em outros projetos. O Arduino UNO, exibido na Figura 1, é um dispositivo de desenvolvimento que permite a criação de protótipos de sistemas que podem ser escalados para arquiteturas mais robustas. Ele se baseia no microcontrolador ATmega328P e oferece uma boa diversidade de pinos de entradas e saídas, que estarão descritos na Tabela 1. O Arduino UNO é programado usando o ambiente de desenvolvimento chamado de Arduino IDE, que permite a gravação de códigos em C++ no ATmega328P (ARDUINO.CC, 2023). A Figura 2 exibe de forma breve o ambiente de desenvolvimento do Arduino IDE.

Figura 1 - Arduino UNO



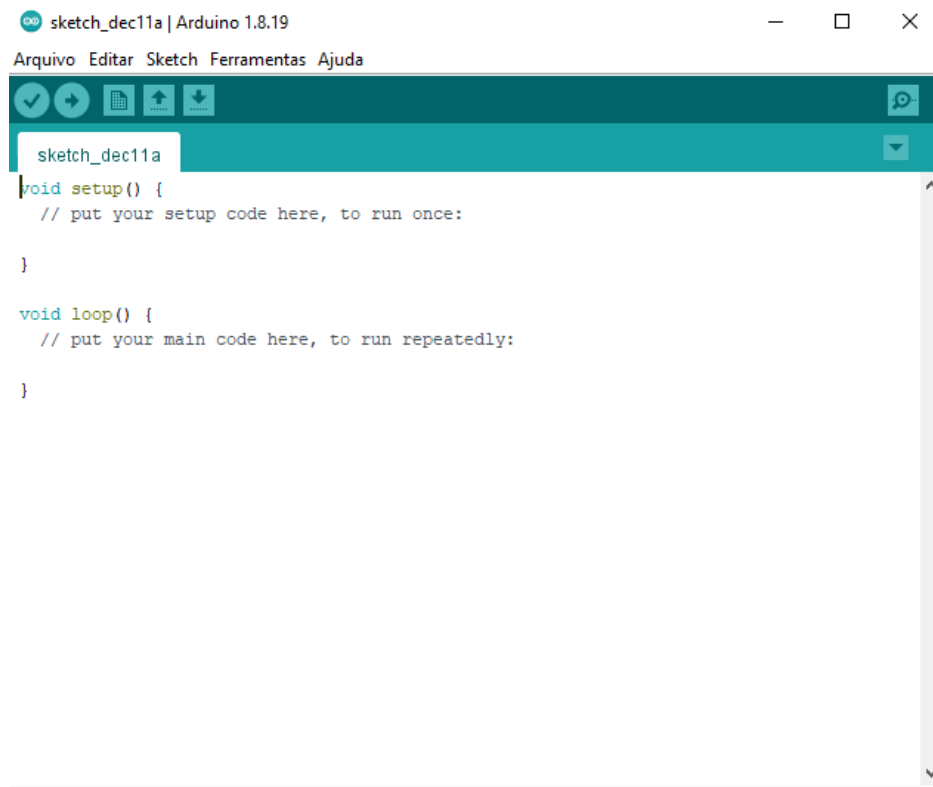
Fonte: Própria

Tabela 1 - Datasheet Arduino

| Especificações técnicas Arduino |                                   |
|---------------------------------|-----------------------------------|
| Microcontrolador                | ATmega328P                        |
| Tensão de operação              | 5V                                |
| Tensão de entrada (recomendada) | 7-12V                             |
| Tensão de entrada (limite)      | 6-20V                             |
| Pinos digitais                  | 14 (dessas, 6 fornecem saída PWM) |
| Pinos PWM                       | 6                                 |
| Corrente CC por pino digital    | 20mA                              |
| Corrente CC por saída de 3,3V   | 50mA                              |
| Memória Flash                   | 32KB                              |
| SRAM                            | 2KB                               |
| EEPROM                          | 1KB                               |
| Velocidade do clock             | 16MHz                             |
| Peso                            | 25g                               |

Fonte: Data Sheet Arduino

Figura 2 - Arduino IDE



Fonte: Própria

### 2.1.1 Comunicação serial

Neste trabalho foram utilizados dois módulos acessórios ao Arduino UNO que estarão descritos na sessão 3.2, tais módulos realizam comunicação com o Arduino UNO de forma serial assíncrona. O protocolo de comunicação serial que o Arduino utiliza é o UART (*Universal Asynchronous Receiver / Transmitter*), que é um dos mais utilizados para comunicação entre dispositivos (ARDUINO, 2024). A comunicação UART é assíncrona, ou seja, não há um *clock* compartilhado entre os dispositivos. Ao invés disso, os dispositivos precisam estabelecer uma taxa de transmissão para garantir uma comunicação eficaz, no caso deste projeto foi definido por meio de linha de código a taxa de transmissão de cada um desses dispositivos. Uma taxa de transmissão mais alta possibilita uma troca de dados mais rápida, mas exige uma sincronização de tempo mais precisa entre os dispositivos. Por outro lado, uma taxa de transmissão mais baixa pode ser adequada para aplicações onde a precisão do tempo é menos crítica, mas resulta em uma transferência de dados mais lenta.

Uma limitação do Arduino é que ele só consegue realizar a comunicação de forma serial com um dispositivo de cada vez (ARDUINO, 2024), dessa forma, é necessário definir uma estrutura para que o Arduino se comunique com o módulo adequado no momento desejado. Uma alternativa é utilizar a biblioteca *Arduino SoftwareSerial* que permite programar o Arduino para que no momento desejado ele troque informações com um módulo, e na sequência, se comunique com outro módulo, possibilitando assim a utilização de vários módulos com um único Arduino.

## 2.2 GPS

O GPS, é um sistema que permite determinar a localização de um dispositivo e a definição do horário em qualquer local da terra em tempo real. Ele foi desenvolvido no final do século XX pelo Departamento de Defesa dos Estados Unidos que mantém sua sustentação até os dias de hoje. O GPS possuía utilização unicamente para o meio militar quando foi criado, mas na virada do milênio foi aberto para a população geral, entretanto, ainda hoje há dois níveis diferentes de serviço, o SPS (*Standard Positioning Service*) e o PPS (*Precise Positioning Service*), sendo esse último de uso restrito das forças armadas americanas. (MANNING, 2023)

### **2.2.1 Funcionamento do GPS**

Atualmente existem 24 satélites em operação ao redor do globo que orbitam numa altitude de 20.200 quilômetros da terra, satélites esses que são estrategicamente configurados para garantir que, a partir de qualquer ponto na superfície terrestre, um dispositivo receptor possa captar o sinal de pelo menos três desses satélites simultaneamente. Cada satélite GPS transmite de forma constante uma mensagem para o receptor sinalizando a posição do satélite no momento da transmissão, bem como um registro de data e hora extremamente precisos obtidos por relógios atômicos existentes no satélites. O funcionamento do sistema se baseia na triangulação, onde o receptor capta os sinais transmitidos e calcula a diferença de tempo que cada sinal levou para percorrer a distância até o dispositivo receptor, essa informação é então utilizada para determinar a posição exata do receptor. (EMBRAPA, 2020)

## **2.3 Redes móveis**

As redes móveis são parte fundamental na estrutura de comunicação existente atualmente no mundo, através delas é possível existir a conexão sem fio entre pontos distintos em todo o mundo. Para este trabalho foi necessário o uso dessa tecnologia, mais especificamente a tecnologia GPRS (*General Packet Radio Service*), para realizar a comunicação do dispositivo de GPS com um banco de dados na nuvem que possibilite o armazenamento das informações.

### **2.3.1 GPRS**

O GPRS, também conhecido como 2.5G, foi uma tecnologia criada para possibilitar a comunicação de dados entre redes móveis, que pertence ao grupo de tecnologias GSM. Essa tecnologia abriu um novo leque de possibilidades na troca de dados entre dispositivos, com isso, diversas aplicações foram surgindo, como por exemplo, a possibilidade de realizar o envio de um simples e-mail por meio de um aparelho celular.

A rede GPRS se baseia na comutação de pacotes, ou seja, o emissor da informação dispara para a rede um pacote que tem em seus metadados o destinatário da mensagem. Esse pacote é transmitido pela rede que fica como responsável entregar no melhor caminho para o destinatário (HALONEN, ROMERO e MELERO, 2003).

A tecnologia GPRS possibilitará que o dispositivo que será resultado deste trabalho envie as informações obtidas por meio de um módulo GPS para um banco de dados na nuvem com segurança e robustez.

## 2.4 Comandos AT

Os comandos AT (abreviação da palavra *attention*), conhecidos também como comandos *Hayes*, foram desenvolvidos na década de 1970 por *Dale Heatherington* e *Dennis Hayes*, são uma forma de linguagem desenhada e pensada para permitir a comunicação entre modems, através de comandos curtos de texto que combinados são capazes de realizar algumas ações como ligações, envio de SMS (*Short Message Service*), transmissão de dados, navegação na internet, entre outros.

Com o avanço das redes móveis somada a tecnologia GPRS, esses comandos foram absorvidos pelos padrões de comunicação, possibilitando que os comandos AT fossem utilizados por dispositivos que utilizem a tecnologia GPRS.

Para este trabalho foram utilizados os comandos AT abaixo para possibilitar o envio de informações obtidas do GPS para um banco de dados através da tecnologia GPRS (SIMCOM, 2015).

- AT - Checa a conexão entre o módulo e o controlador;
- AT + CREG? – Verifica o status da rede;
- AT + CGATT? – Status das configurações;
- AT + CIPSHUT – Finaliza a conexão GPRS;
- AT + CIPSTATUS – Verifica o status da conexão;
- AT + CIPMUX = 0 – Configura o dispositivo como IP único;
- AT + CSTT – Se conecta com a operadora do chip;

- AT + CIIER – Gera conexão com a rede GPRS;
- AT + CIIER - Adquire o endereço de IP;
- AT + CIIER=0 – Utilizado após o comando de envio de dados;
- AT + CIIER – Inicia conexão TCP ou UDP;
- AT + CIIER – Envia dados para um servidor.

## 2.5 Banco de dados na nuvem

Para armazenar dados na nuvem de maneira estruturada, segura e robusta, a escolha de um banco de dados adequado é essencial. O banco de dados pode ser brevemente definido por um sistema desenhado para coletar, armazenar, gerenciar e recuperar informações de maneira eficiente. Essa infraestrutura é projetada para permitir a usuários e aplicativos a capacidade de acessar, atualizar e processar dados de forma rápida, segura e eficiente (ALVES, 2020). Ao se escolher um banco de dados na nuvem, é possível realizar o acesso remoto a partir de diferentes localizações, o que vai ser de extrema importância para este projeto pois o dispositivo estará em constante deslocamento.

Bancos de dados são importantes para muitas aplicações, e o utilizaremos neste trabalho para armazenar as informações obtidas do GPS que serão transmitidas através da tecnologia GPRS.

## 2.6 Protocolo HTTP

O protocolo HTTP (*Hypertext Transfer Protocol*) é um protocolo de comunicação utilizado para a transmissão de informações na World Wide Web (WWW). Ele é a base para a comunicação entre aplicações, como por exemplo, um navegador web e um servidor. O HTTP permite que essas duas instâncias troquem informações à medida que requisições são feitas entre os sistemas.

As operações que realizam requisições entre as aplicações são chamados de métodos HTTP (GOURLEY, TOTTY, *et al.*, 2002). Tais métodos sinalizam para o servidor qual ação deve ser realizada, sendo os principais métodos descritos abaixo:

- *GET* – Solicita informações do servidor;
- *PUT* – Atualiza uma informações do servidor;
- *DELETE* – Remove informações do servidor;
- *POST* – Envia informações para o servidor.

No contexto deste trabalho, o método a ser utilizado pelo dispositivo é o POST, visto que o rastreador irá enviar ativamente e frequentemente informações para o banco de dados armazenar.



### 3 DESENVOLVIMENTO DO TRABALHO

Nessa sessão será apresentado o caminho traçado para desenvolvimento do dispositivo, sendo dividido em 5 etapas: definição da arquitetura da solução, escolha do hardware e banco de dados, ligação física, alimentação do sistema e os testes.

#### 3.1 Definição da arquitetura da solução

Esse projeto tem como objetivo desenvolver um dispositivo que obtenha sua geolocalização através de sinal GPS e faça o envio dessa informação através de redes GPRS para um banco de dados na nuvem. Esse fluxo será gerenciado por um microcontrolador que será responsável por orquestrar a comunicação entre sistemas e módulos. Um resumo dessa arquitetura encontra-se na Figura 3.

Figura 3 - Arquitetura da solução



Fonte: Própria

#### 3.2 Escolha do hardware e banco de dados

A seguir serão descritos quais os hardwares utilizados e a escolha do banco de dados, bem como o motivo da escolha de cada um desses. Com esses alicerces bem definidos, será possível implementar de maneira robusta e segura a arquitetura proposta na Figura 3.

### 3.2.1 Arduino UNO

O dispositivo escolhido para orquestrar todo o funcionamento do sistema foi o Arduino UNO, exibido na Figura 4. Pois além das vantagens citadas na sessão **Erro! Fonte de referência não encontrada.** desse trabalho, ele possui um baixo custo de aquisição e uma ampla quantidade de conteúdos disponíveis para pesquisa. Para este trabalho, o Arduino isoladamente não consegue desempenhar todos os papéis necessários, mas por ser um dispositivo amplamente utilizado para a construção de protótipos de dispositivos mais robustos, diversos outros módulos acessórios são desenvolvidos por outras empresas e que possuem plena compatibilidade com a arquitetura do Arduino UNO.

Foi realizada uma pesquisa para compreender quais módulos GSM e GPRS possibilitariam uma comunicação adequada com o Arduino para possibilitar o êxito do projeto.

Figura 4 - Arduino UNO utilizado



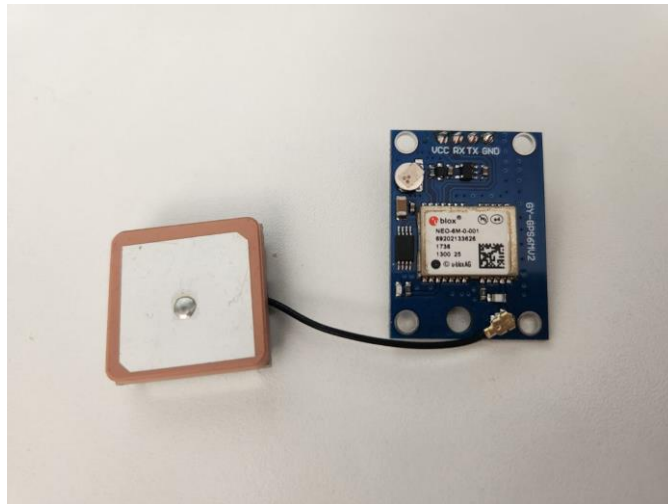
Fonte: Própria

### 3.2.2 Módulo GPS

Para obter a geolocalização em tempo real do dispositivo é necessário agregar ao Arduino um módulo GPS. No mercado atualmente não existe uma grande variedade de módulos que desempenham esse papel. A opção dita comercial, ou seja,

que é de fácil acesso para compra e baixo custo é o NEO-6M, exibido na Figura 5. Esse módulo é fabricado pela empresa suíça U-blox que é especializada na criação de dispositivos de geolocalização.

Figura 5 - Módulo NEO-6M



Fonte: Própria

Tal módulo se mostrou uma solução adequada para o projeto por quatro principais motivos:

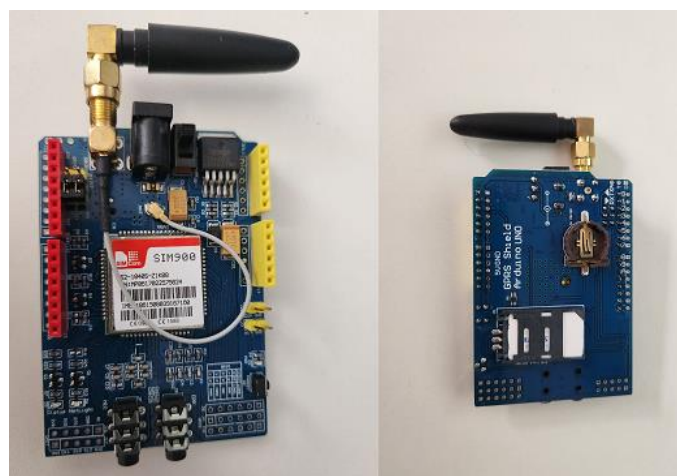
- Alta compatibilidade com o Arduino UNO;
- Projetado para um baixo consumo de energia, podendo ser alimentado diretamente por pinos do Arduino UNO, sem a necessidade de uma outra fonte externa de alimentação;
- Alta sensibilidade, ou seja, o módulo consegue obter sinais de satélite mesmo em situações em que o sinal está fraco;
- Alta precisão na obtenção dos dados advindos dos satélites. Isso significa que o módulo consegue obter a localização exata do dispositivo, permitindo assim um acompanhamento preciso do sistema (U-BLOX).

### 3.2.3 Módulo GPRS

A premissa do projeto em questão é que a geolocalização obtida através do módulo GPS seja transmitida para um banco de dados na nuvem, com isso, é necessário que de alguma forma a geolocalização seja transmitida para esse banco de dados.

Nesse contexto, optou-se por fazer a transmissão de dados através da internet. Para isso, é necessário agregar ao Arduino UNO um módulo que é capaz de transmitir dados pela internet através da tecnologia GPRS. Atualmente existem algumas opções de módulos que desempenham esse papel, entretanto no Brasil apenas os módulos fabricados pela SIMCom, empresa chinesa especializada na criação de soluções de comunicação sem fio, são de fácil aquisição no mercado. Dessa forma, foi escolhido o módulo SIM900, exibido na Figura 6.

Figura 6 - Módulo SIM900



Fonte: Própria

O módulo consegue realizar diversas funções através dos comandos AT, que foram descritos na sessão 2.4 deste trabalho. Para esse projeto, a função AT mais relevante é a de transmissão de informações através do protocolo HTTP, com isso, conseguiremos transmitir as informações de geolocalização para um banco de dados na nuvem.

### **3.2.4 Banco de dados**

Com toda a arquitetura de obtenção e transmissão de informação resolvida, partiu-se para a busca de um banco de dados que fosse capaz de receber os dados transmitidos.

Os critérios de busca para encontrar o banco de dados mais adequado para a solução foram:

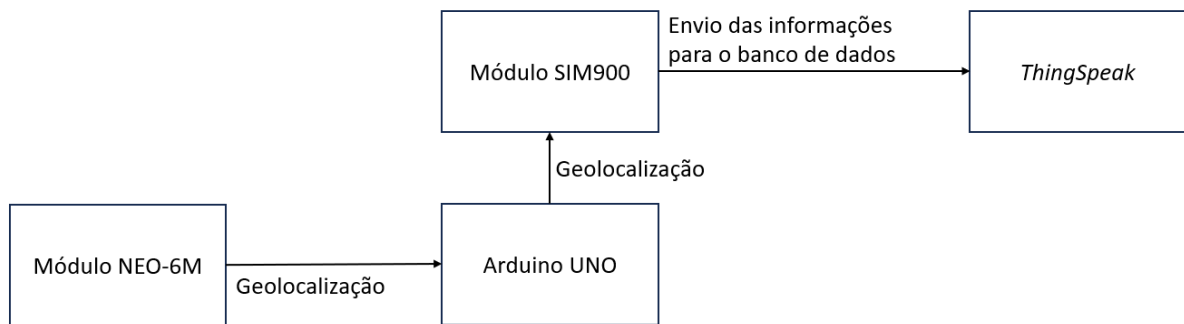
- Nenhum custo inicial;
- Facilidade para inserir os dados;
- Pertencente ao contexto de *IoT*.

Com essas premissas, encontrou-se a solução da *ThingSpeak*, uma plataforma cuja arquitetura está totalmente preparada para usuários que possuam dispositivos *IoT* consigam armazenar e gerenciar os dados gerados (THINGSPEAK). Para inserir ou adquirir dados do banco de dados basta que sejam utilizados os métodos do protocolo HTTP que forma descritos na sessão 2.6 deste documento. Além disso, não é necessário realizar a aquisição de nenhum plano para uma pequena quantidade de inserção de dados no banco, com isso, não existiu nenhum custo de utilização da plataforma.

### **3.2.5 Arquitetura final da solução**

Na Figura 3 foi exibido a arquitetura da solução proposta sem mencionar nominalmente os módulos e dispositivos utilizados nessa construção. Abaixo na Figura 7 se encontra a arquitetura final da solução nomeando cada hardware e software utilizado.

Figura 7 - Arquitetura final da solução

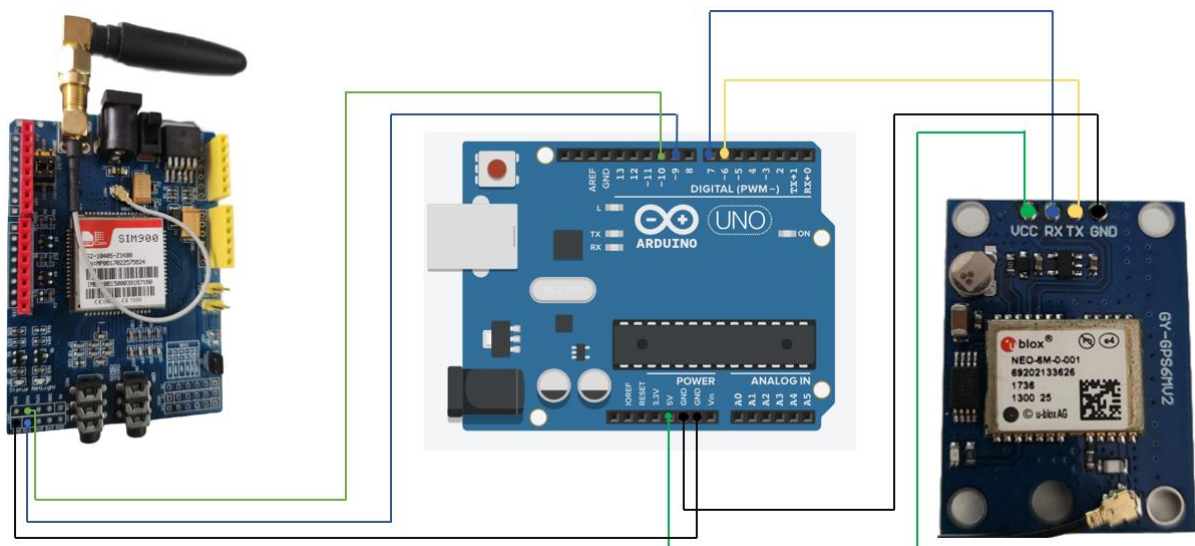


Fonte: Própria

### 3.3 Ligação física

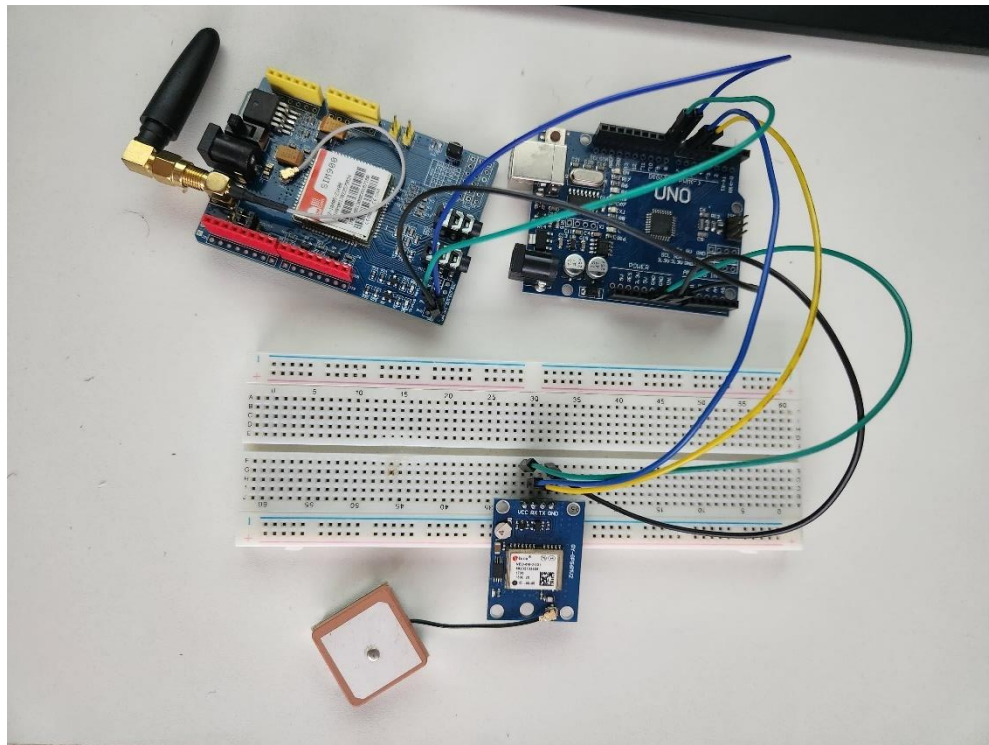
Uma vez resolvida a arquitetura da solução e quais módulos serão utilizados, foi necessário realizar a ligação física entre todas as partes do sistema. Para fornecer melhor clareza, na Figura 8 está um esquemático da ligação entre os módulos e o Arduino, e na Figura 9 a ligação física real realizada.

Figura 8 - Esquemático ligação física



Fonte: Própria

Figura 9 - Ligação física



Fonte: Própria

### 3.4 Sistema de alimentação

Nesta sessão será apresentada a forma que será fornecido energia para o funcionamento do dispositivo, dividido em duas etapas: A alimentação para testes residenciais e alimentação para o cenário real de uso.

#### 3.4.1 Alimentação para testes residenciais

Para teste do sistema em ambiente residencial a alimentação do sistema foi feita através de fontes de energias ligadas a rede elétrica.

O Arduino UNO foi alimentado através de um cabo USB A/B, exibido na Figura 10, onde a outra extremidade do cabo que não está conectada ao Arduino UNO está ligada a um computador de mesa padrão.

O módulo NEO-6M necessita de alimentação uma fonte CC de 5V, essa que pode ser fornecida através da própria placa do Arduino UNO, conforme exposto na Figura 9 que exhibe a ligação física do sistema.

Já no caso do módulo SIM900 tem-se uma particularidade, ele pode requisitar uma corrente contínua de até 1A, e suporta tensões de até 9V. Dessa forma, o mesmo não pode ser alimentado pelo Arduino UNO como o módulo NEO-6M. Para isso, foi adquirido uma fonte CC, exibida na Figura 11, que converte a tensão da rede de 220V para 9V, com capacidade máxima de corrente de 1A. Dessa forma, é possível fornecer a energia necessária para o funcionamento do módulo. A ligação completa do sistema encontra-se na Figura 12.

Figura 10 - Cabo USB A/B



Fonte: Própria

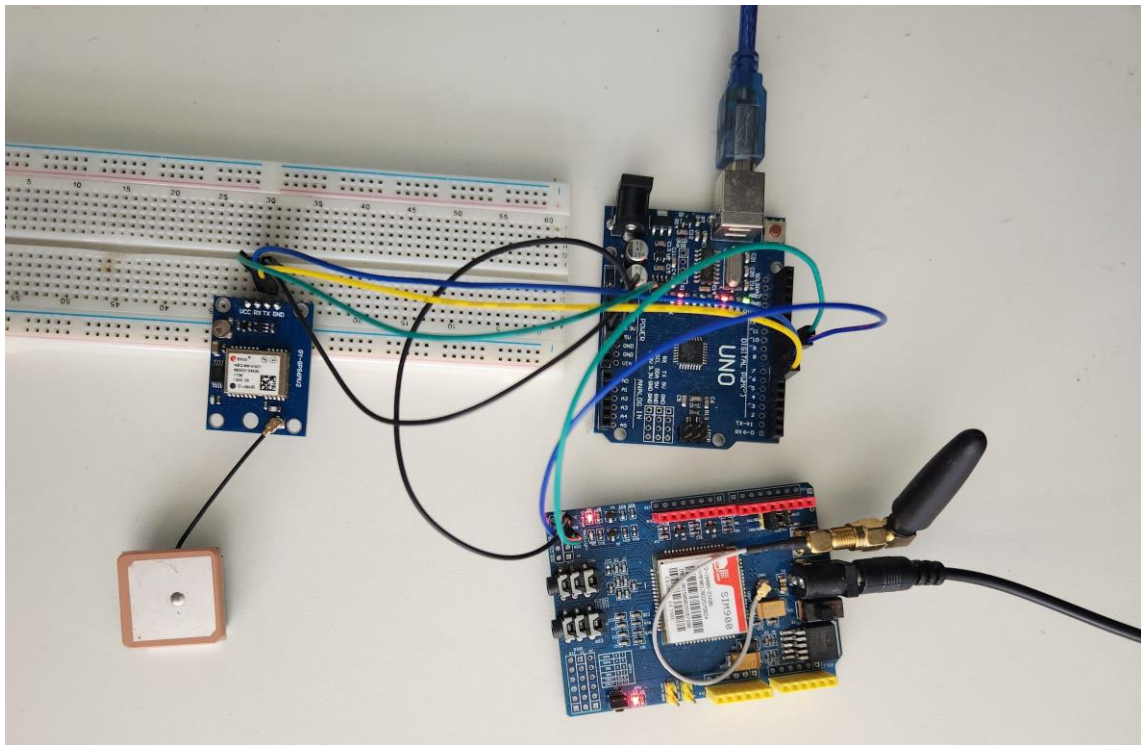
Figura 11 - Fonte de alimentação 9V-1A



Fonte: Própria



Figura 12 - Sistema completo com alimentação a rede elétrica



Fonte: Própria

### 3.4.2 Alimentação no cenário real de uso

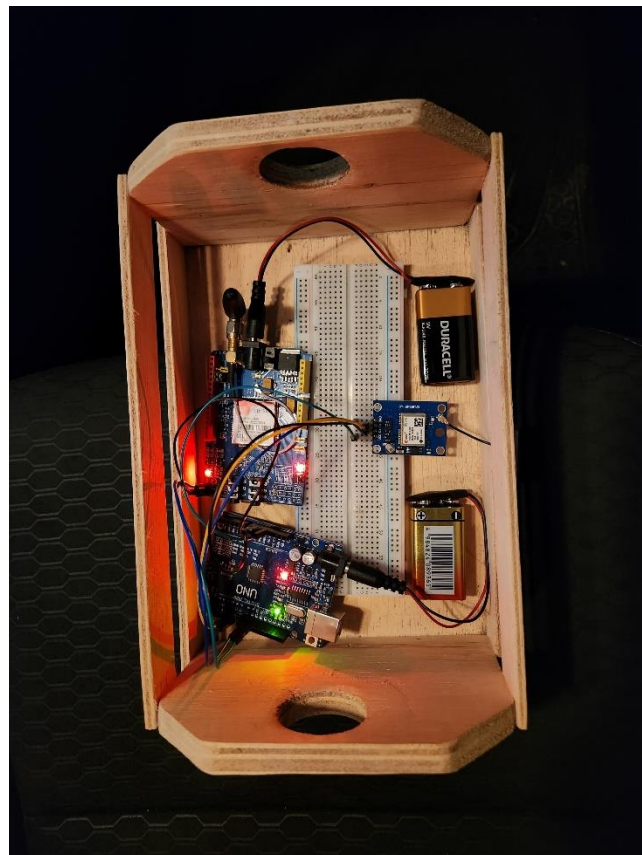
No cenário real de uso não é possível utilizar a estratégia sinalizada no tópico anterior, visto que se deseja que o dispositivo funcione normalmente nas rodovias, logo, sem ligação direta com a rede elétrica. Para viabilizar o fornecimento de energia no sistema sem dependência de fontes ligadas a rede elétrica, as fontes serão substituídas por baterias de 9V em conjunto com um adaptador que possibilitará a ligação com os dispositivos, tal adaptador está exibido na Figura 13. Para alimentar todo o sistema serão necessárias duas baterias, uma exclusivamente para o módulo SIM900 e outra para a alimentação do Arduino UNO, e por consequência, o módulo NEO-6M. A ligação completa do sistema em um cenário real de uso está exibida na Figura 14, que exhibe o dispositivo dentro de um veículo.

Figura 13 - Configuração bateria adaptador para alimentar o sistema



Fonte: Própria

Figura 14 - Alimentação do dispositivo em cenário real de uso



Fonte: Própria

### 3.5 Testes

Será apresentado nesse tópico o resultados dos testes do dispositivo em dois cenários diferentes, inicialmente em ambiente controlado no interior de uma residência para validar o funcionamento da solução. Na sequência será exibido o resultado do teste realizado dentro de um veículo em movimento percorrendo a cidade do Recife.

Para ambos os cenários o teste será considerado um sucesso caso o dispositivo envie corretamente para o banco de dados na nuvem as seguintes informações: Latitude, longitude, horário e *link* do *google maps* indicando a posição do dispositivo.

Será exibido nas sessões abaixo o retorno do banco de dados em formato *json*, nele os campos não estão sendo exibidos com os devidos nomes e sim com as nomenclaturas genéricas do banco de dados. Na Tabela 2 está exibida a legenda de nomenclatura real do dado que é exibido no banco de dados, facilitando o entendimento das informações.

Tabela 2 - Legenda campos do banco de dados

| Nome no banco de dados | Nome adequado                     |
|------------------------|-----------------------------------|
| Field 1                | Campo de teste genérico           |
| Field 2                | Latitude                          |
| Field 3                | Longitude                         |
| Field 4                | Horário da informação             |
| Field 5                | <i>Link</i> do <i>google maps</i> |

Fonte: Própria

#### 3.5.1 Ambiente Residencial

Para o teste em ambiente residencial o sistema foi energizado conforme exibido na sessão 3.4.1 desse trabalho. Será verificado através do banco de dados se as informações desejadas estão sendo enviadas corretamente.

Como neste teste o dispositivo estará parado é inevitável que a latitude, longitude e o *link* do *google maps* se mantenham aproximadamente constantes. Se diz aproximadamente pois os dados de latitude e longitude estão sendo considerados

com uma precisão de 5 casas decimais, então, naturalmente haverá uma leve variação da posição devido a esse grau precisão.

Na Figura 15 estão as informações que o dispositivo enviou para o banco de dados *ThingSpeak* em formato *json* em 4 momentos diferentes. Para preservar a localização do autor deste trabalho as informações de latitude, longitude e o link do *google maps* estarão com parte da informação censurada.

Figura 15 - Informações registradas no banco de dados no teste residencial

```

{
  "created_at": "2023-11-25T15:31:37Z",
  "entry_id": 120,
  "field1": null,
  "field2": "-8.0[REDACTED]",
  "field3": "-34.9[REDACTED]",
  "field4": "25112023151526",
  "field5": "https://maps.google.com/?q=-8.0[REDACTED],-34.9[REDACTED]"
},
{
  "created_at": "2023-11-25T15:34:38Z",
  "entry_id": 121,
  "field1": null,
  "field2": "-8.03[REDACTED]",
  "field3": "-34.9[REDACTED]",
  "field4": "25112023153356",
  "field5": "https://maps.google.com/?q=-8.0[REDACTED],-34.9[REDACTED]"
},
{
  "created_at": "2023-11-25T15:37:18Z",
  "entry_id": 122,
  "field1": null,
  "field2": "-8.0[REDACTED]",
  "field3": "-34.9[REDACTED]",
  "field4": "25112023153356",
  "field5": "https://maps.google.com/?q=-8.0[REDACTED],-34.9[REDACTED]"
},
{
  "created_at": "2023-11-25T15:38:35Z",
  "entry_id": 123,
  "field1": null,
  "field2": "-8.0[REDACTED]",
  "field3": "-34.9[REDACTED]",
  "field4": "25112023153753",
  "field5": "https://maps.google.com/?q=-8.0[REDACTED],-34.9[REDACTED]"
}

```

Fonte: Própria

Como é possível visualizar na Figura 15, o banco de dados recebeu e armazenou corretamente as informações enviadas pelo dispositivo, desta forma, é possível concluir que o teste em ambiente residencial foi um sucesso.

### 3.5.2 Dentro de um veículo em locomoção

No cenário real de uso, ou seja, em um veículo em locomoção por via terrestre, o sistema foi alimentado da forma que foi exibido na sessão 3.4.2 deste trabalho. O teste em questão será um sucesso caso as informações enviadas pelo dispositivo cheguem de maneira correta no banco de dados. Para isso, será verificado se os dados presentes nos campos exibidos na Figura 16 estão devidamente preenchidos e com valores coerentes de latitude, longitude e o link do *google maps*, de forma análoga ao exibido na sessão 3.5.1.

Como o teste é em um veículo em locomoção, é esperado que as informações de latitude e longitude se modifiquem ao decorrer do tempo, indicando que o dispositivo está representando o deslocamento do veículo que ele se encontra.

Na Figura 16 se encontra o resultado deste teste. Novamente, para preservar a localização do autor deste trabalho a informação em questão será censurada.

Figura 16 - Informações registradas no banco de dados no teste em cenário real de uso

```

{
  "created_at": "2023-10-28T17:31:53Z",
  "entry_id": 103,
  "field1": null,
  "field2": "-8.058[REDACTED]",
  "field3": "-34.885[REDACTED]",
  "field4": "2810202317015",
  "field5": "https://maps.google.com/?q=-8.058[REDACTED],-34.885[REDACTED]",
},
{
  "created_at": "2023-10-29T07:40:33Z",
  "entry_id": 104,
  "field1": null,
  "field2": "-8.044[REDACTED]",
  "field3": "-34.902[REDACTED]",
  "field4": "2910202373951",
  "field5": "https://maps.google.com/?q=-8.044[REDACTED],-34.902[REDACTED]",
},
{
  "created_at": "2023-10-29T07:44:28Z",
  "entry_id": 105,
  "field1": null,
  "field2": "-8.061[REDACTED]",
  "field3": "-34.897[REDACTED]",
  "field4": "2910202374343",
  "field5": "https://maps.google.com/?q=-8.061[REDACTED],-34.897[REDACTED]",
},
]
{ 7 items }
{
  "created_at": "2023-10-29T07:52:09Z",
  "entry_id": 107,
  "field1": null,
  "field2": "-8.090[REDACTED]",
  "field3": "-34.884[REDACTED]",
  "field4": "2910202374735",
  "field5": "https://maps.google.com/?q=-8.090[REDACTED],-34.884[REDACTED]",
}

```

Fonte: Própria

Como exibido na Figura 16, o teste no cenário real de uso da solução pode ser considerado um sucesso pois as informações foram gravadas corretamente no banco de dados e as variações de latitude e longitude ocorreram conforme esperado.

## 4 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Neste trabalho foi apresentado o desenvolvimento e os resultados de um protótipo de um dispositivo que realiza o envio de sua geolocalização para um banco de dados, aplicado a um contexto de rastreamento de cargas. O estudo teve como motivação o resultado da pesquisa de NPS realizado numa empresa de Pernambuco que demonstrou a insatisfação dos clientes em não terem ciência de quando a carga adquirida chegaria em suas instalações.

Conforme os resultados apresentados nos tópicos 3.5 é possível concluir que o protótipo desempenhou bem sua função, pois ele conseguiu transmitir as informações desejadas de forma correta para o banco de dados.

Entretanto, durante a fase de testes foram percebidos alguns pontos que tornariam o dispositivo mais robusto, mas que não trouxeram malefícios para o trabalho. Dentre eles podemos citar:

- i. Autonomia da bateria;
- ii. Área de sombra de sinal de internet;
- iii. Conexões não soldadas.

Dos pontos citados acima o mais crítico é com relação a autonomia da bateria, visto que para o cenário de teste exibido no tópico 3.5.2 as baterias tiveram sua carga totalmente drenada em aproximadamente 12 horas de utilização, considerando o envio de dados a cada 5 minutos. Com isso, para deslocamento de cargas que levam tempo superior a esse seria necessário a substituição da bateria.

Durante alguns trechos utilizados no teste a operadora de celular utilizada não possuía cobertura móvel na área, como o dispositivo se baseia no envio de dados via rede de telefonia móvel a informação não pôde ser transmitida para o banco de dados.

Por fim, como exibido na Figura 9, por se tratar de um protótipo os pinos não foram soldados entre os dispositivos, com isso, em trechos de rodovia mais esburacados ou desnivelados uma trepidação mais abrupta do veículo fez com que as conexões fossem desfeitas.

Portanto, há espaço para desenvolvimento de melhorias desse sistema de forma a permiti-lo ter mais robustez com base no que foi identificado nos testes, sendo elas:

- i. Adaptar a forma de energização do sistema para permitir ser alimentado pela energia elétrica dos veículos;
- ii. Estudo de otimização de gasto de energia do sistema e busca por bateria mais duradoura que as utilizadas para um possível *backup*;
- iii. Substituir o chip de uma única operadora por uma solução multioperadora existente no mercado;
- iv. Integrar os dispositivos em uma placa de circuito impresso.

Além dos pontos de melhoria identificados durante os testes realizados, existe mais uma possibilidade de melhoria desse sistema, que consiste em expandir suas funcionalidades para permitir o envio direto das informações armazenadas no banco de dados ao cliente, utilizando mensagens via WhatsApp. Este incremento tem o potencial de melhorar significativamente a dor relatada pelo cliente na pesquisa NPS, ao fornecer informações de forma mais rápida e acessível.



## REFERÊNCIAS

ALVES, Willian P. **Banco de dados: Teoria e Desenvolvimento**. 2. ed.

ARDUINO. Arduino Software Serial. **Arduino**, 2024. Disponível em: <<https://docs.arduino.cc/learn/built-in-libraries/software-serial>>. Acesso em: 10 jan. 2024.

ARDUINO. ARDUINO.CC. **ARDUINO.CC**, 2024. Disponível em: <<https://docs.arduino.cc/learn/communication/uart/>>. Acesso em: 17 fev. 2024.

ARDUINO.CC. Arduino. **store.arduino.cc**, 2023. Disponível em: <<https://store.arduino.cc/products/arduino-uno-rev3>>. Acesso em: 20 out. 2023.

CZARNECKI, Harley M. A. D. Customer Experience Drives Revenue Growth. **Forrester**, 2021. Disponível em: <[https://www.forrester.com/blogs/16-06-21-customer\\_experience\\_drives\\_revenue\\_growth\\_2016/](https://www.forrester.com/blogs/16-06-21-customer_experience_drives_revenue_growth_2016/)>. Acesso em: 20 nov. 2023.

EMBRAPA. Embrapa. **Satelites de monitoramento**, 2020. Disponível em: <<https://www.embrapa.br/satelites-de-monitoramento/missoes/gps>>. Acesso em: 25 nov. 2023.

GOURLEY, David et al. **HTTP: The Definitive Guide**. 1. ed.

HALONEN, Timo; ROMERO, Javier; MELERO, Juan. **GSM, GPRS AND EDGE PERFORMANCE**. 2. ed.

MANNING, Catherine G. nasa.gov. **Nasa**, 1 set. 2023. Disponível em: <<https://www.nasa.gov/directorates/somd/space-communications-navigation-program/gps/>>. Acesso em: 15 dez. 2023.

REICHHELD, Frederick F. The One Number You Need to Grow. **Harvard Business Review OnPoint**, p. 12, dez. 2003.

SIMCOM. SIMCom. **SIMCom**, 01 dez. 2015. Disponível em: <[https://simcom.ee/documents/SIM900/SIM900\\_AT%20Command%20Manual\\_V1.11.pdf](https://simcom.ee/documents/SIM900/SIM900_AT%20Command%20Manual_V1.11.pdf)>. Acesso em: 03 out. 2023.

THINGSPEAK. Learn more. **Thingspeak**. Disponível em: <[https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more)>. Acesso em: 10 nov. 2023.

U-BLOX. Datasheet NEO-6M. **U-blox**. Disponível em: <[https://content.u-blox.com/sites/default/files/products/documents/NEO-6\\_DataSheet\\_%28GPS.G6-HW-09005%29.pdf](https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf)>. Acesso em: 20 nov. 2023.

VIANNA, Bernardo. Insper. **Insper**, 2022. Disponível em: <<https://www.insper.edu.br/noticias/cada-vez-mais-pessoas-passam-conectadas-por-mais-tempo/#:~:text=Popula%C3%A7%C3%A3o%20mundial%20conectada%20em%20janeiro%20de%202022&text=A%20an%C3%A1lise%20dos%20dados%20hist%C3%B3ricos,4%2C95%20bilh%C3%B5es%20em%202022>>. Acesso em: 20 out. 2023.

## APÊNDICES

### APÊNDICE A – CÓDIGO IMPLEMENTADO NO ARDUINO

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
TinyGPSPlus gps;
SoftwareSerial serial_neo(6, 7); // RX, TX
SoftwareSerial mySerial(9, 10); // RX, TX
String Write_API_key = "OMITIDO_PARA_APENDICE";
String lat_long;
float latitude = 99;
float longitude;
String longitude_maior;
String latitude_maior;
String date_time;

void setup() {
  //Baud rate do objeto serial
  Serial.begin(9600);
  // Definindo baud rate dos objetos da software serial
  serial_neo.begin(9600);
  mySerial.begin(9600);
  Serial.println("Iniciando GPS");
}

void loop() { //Loop principal do programa
  serial_neo.listen();
  //Serial.print(latitude);
  //Serial.print(longitude);
  Serial.print(latitude_maior);
  Serial.print(longitude_maior);
  if(latitude == 99){
    while (serial_neo.available() > 0){ // While de aquisição da posição
      if (gps.encode(serial_neo.read())){
        sinalGPS();
      }
    }
    if (latitude != 99){ //Quando a latitude é populada pelo valor correto rodamos a
    rotina de enviar o dado para o servidor
      mySerial.listen();
      if (Serial.available()>0)
        delay(1000);
      SetupModule();
      mySerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\");
      delay(3000);
      MostraSerial();
      mySerial.println("AT+CIPSEND");
    }
  }
}
```

```

        delay(2000);
        Serial.println();
        MostraSerial();
        String str="GET
https://api.thingspeak.com/update?api_key="+Write_API_key+"&field2="+latitude_ma
ior+"&field3="+longitude_maior+"&field4="+date_time+"&field5=https://maps.google.c
om/?q="+latitude_maior+","+longitude_maior;
        Serial.println(str); delay(2000);
        mySerial.println(str); delay(4000);
        MostraSerial();
        mySerial.println((char)26); delay(4000);
        mySerial.println();
        MostraSerial();
        mySerial.println("AT+CIPSHUT");//close the connection
        delay(500);
        MostraSerial();
        str="";
        Serial.print("Vou entrar no delay");
        delay(180000); // Rodando o loop a cada 3 minutos
        Serial.print("Passei delay");
        latitude = 99;
        break;
    }
}
}
Serial.println();

} // Fim do loop principal do programa

void sinalGPS(){
    if (gps.location.isValid()){
        latitude = gps.location.lat();
        longitude= gps.location.lng();
        String lat_long = "";
        latitude_maior = String(latitude,6);
        longitude_maior = String(longitude,6);
        lat_long += latitude_maior;
        lat_long += longitude_maior;
        Serial.print(gps.location.lat(), 6);
    }else{
        Serial.print(F("INVALID"));
    }

    if (gps.date.isValid()){
        date_time = "";
        date_time += gps.date.day();
        //date_time += "/";
        date_time += gps.date.month();
        //date_time += "/";
        date_time += gps.date.year();
    }
}

```

```

    //date_time += " ";
}else{
    Serial.print(F("INVALID"));
}

Serial.print(F(" "));
if (gps.time.isValid()){
    date_time += gps.time.hour();
    //date_time += ":";
    date_time += gps.time.minute();
    //date_time += ":";
    date_time += gps.time.second();
    Serial.print(date_time);
}else{
    Serial.print(F("INVALID"));
}
Serial.println();
}

void SetupModule(){
    if (mySerial.available())Serial.write(mySerial.read());
    mySerial.println("AT"); delay(1000);
    mySerial.println("AT+CPIN?"); delay(1000);
    mySerial.println("AT+CREG?"); delay(1000);
    mySerial.println("AT+CGATT?"); delay(1000);
    mySerial.println("AT+CIPSHUT");delay(1000);
    mySerial.println("AT+CIPSTATUS"); delay(2000);
    mySerial.println("AT+CIPMUX=0"); delay(2000);

    //setting the APN,
    mySerial.println("AT+CSTT=\"timbrasil.br\", \"tim\", \"tim\""); delay(1000);
    MostraSerial();
    mySerial.println("AT+CIICR"); delay(2000);
    MostraSerial();
    //get local IP adress
    mySerial.println("AT+CIFSR"); delay(2000);
    MostraSerial();

    mySerial.println("AT+CIPSPRT=0");delay(2000);
    MostraSerial();
}

void MostraSerial(){
    while(mySerial.available() != 0)
        Serial.write(mySerial.read());
    delay(2000);
}

```