



UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS DE PALMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

BRUNO SOUSA DIAS

**PLATAFORMA DE RASTREAMENTO VEICULAR
UTILIZANDO MICROCONTROLADOR ESP32**

Palmas/TO
2021

BRUNO SOUSA DIAS

**PLATAFORMA DE RASTREAMENTO VEICULAR
UTILIZANDO MICROCONTROLADOR ESP32**

Monografia foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, Curso de Engenharia Elétrica para obtenção do título de bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: Prof. Dr. Humberto Xavier de Araújo

Palmas/TO
2021

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

D541p Dias, Bruno Sousa.
PLATAFORMA DE RASTREAMENTO VEICULAR UTILIZANDO
MICROCONTROLADOR ESP32. / Bruno Sousa Dias. – Palmas, TO, 2021.
57 f.

Monografia Graduação - Universidade Federal do Tocantins – Câmpus
Universitário de Palmas - Curso de Engenharia Elétrica, 2021.
Orientador: Humberto Xavier de Araujo

1. GPS. 2. IoT. 3. Rastreamento. 4. ESP32. I. Título

CDD 621.3

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).

FOLHA DE APROVAÇÃO

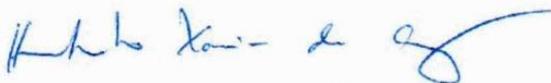
BRUNO SOUSA DIAS

PLATAFORMA DE RASTREAMENTO VEICULAR UTILIZANDO MICROCONTROLADOR ESP32

Monografia foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas - TO, Curso de engenharia elétrica para obtenção do título de bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de aprovação: 16/04/2021

Banca Examinadora



Prof. Dr. Humberto Xavier de Araujo, UFT



Prof. Dr. Adélcio Maximiano Sobrinho, UFT



Prof. Me. Nailson Martins Dantas Landim, UFT

Palmas/TO, 2021

AGRADECIMENTOS

Aos meus pais Cleidon Dias Nascimento e Ana Lúcia Ribeiro de Sousa, que sempre me apoiaram, e acreditaram em mim, sem eles minha graduação não seria possível.

A minha namorada Patrícia Gontijo, por todo apoio e incentivo, desde 2013.

Ao meu orientador Dr. Humberto Xavier de Araujo, por todos os conselhos e suporte, mesmo com seu tempo apertado, sempre esteve disponível para contribuir com este trabalho.

A todos os professores e técnicos do curso de engenharia elétrica da Universidade Federal do Tocantins, que contribuíram para minha formação.

A toda equipe da empresa júnior do curso de engenharia elétrica, Tesla Engenharia, pelo companheirismo e confiança.

A todos os amigos que fiz durante a graduação.

RESUMO

Neste trabalho, será desenvolvido uma plataforma de rastreamento em tempo real com o uso do microcontrolador ESP32 em conjunto com a tecnologia GPS (*Global Positioning System*). Visando obter um sistema capaz de ser usado para rastreamento veicular e em projetos de cidades inteligentes, onde a finalidade vai além de se obter uma localização fixa, de forma que os dados do trajeto serão armazenados e o usuário terá a oportunidade de verificar as rotas já traçadas, contando com redundância para evitar a perda de dados.

Será necessário integrar o sistema desenvolvido com um servidor web, para o armazenamento e processamento de dados. O módulo SIM800L, será o responsável por estabelecer a conexão via GRPS entre o servidor e os dispositivos, esta conexão também poderá ocorrer com redes *Wi-Fi* via ESP32.

Dessa forma obteremos uma opção de sistema de rastreamento flexível e confiável, para aplicações que necessitam armazenar seus históricos de rota.

Palavras-chaves: ESP32. GPS. Rastreamento GPS.

ABSTRACT

In this work, a real-time tracking platform will be developed using the ESP32 microcontroller with GPS (Global Positioning System) technology. Aiming to obtain a system for vehicle tracking and smart city projects, where the purpose goes beyond obtaining a fixed location, so that the route data will be stored and the user will have the opportunity to check the routes, with redundancy to avoid data loss.

It will be necessary to integrate the system developed with a web server, for the storage and data processing. The SIM800L module, will be responsible for establishing the connection using GRPS between the server and the devices, this connection may also occur with Wi-Fi networks via ESP32.

In this way, we will obtain an option of a flexible and reliable tracking system, for applications that need to store their route in database.

Palavras-chaves: ESP32. GPS. tracking GPS.

LISTA DE ILUSTRAÇÕES

Figura 1 – Montagem Preliminar	12
Figura 2 – Localização, Velocidade, Data e Hora	13
Figura 3 – Sistema Com Display e Bateria.....	14
Figura 4 – Ponto Obtido No Google Maps.....	14
Figura 5 – Sistema Montado em Protoboard	15
Figura 6 – Cabos Soldados ao Cartão de Memória	16
Figura 7 – Arquivos de Backup.....	16
Figura 8 – Bateria	17
Figura 9 – Módulo Fotovoltaico de 1,5W	18
Figura 10 – Controle de Carga	19
Figura 11 – Seleção de Método de Carregamento.....	19
Figura 12 – Controle de Carga	20
Figura 13 – Protótipo em Protoboard	20
Figura 14 – Esquemático	21
Figura 15 – Projeto da Placa de Circuito Impresso	22
Figura 16 – Polimento da Placa.....	22
Figura 17 – Aplicação da Tinta Fotossensível	23
Figura 18 – Molde o Circuito Impresso em Fitolito	23
Figura 19 – Molde o Circuito Impresso em Fitolito	24
Figura 20 – Circuito Revelado	24
Figura 21 – Camada Inferior	25
Figura 22 – Camada Superior.....	25
Figura 23 – Verificação das Trilhas e Furos.....	26
Figura 24 – Placa Com Componentes e Barras de Pino.....	26
Figura 25 – Placa Todos os Componentes e Dispositivos.....	27
Figura 26 – Módulo Solar Acoplado à Caixa Hermética	27
Figura 27 – Chave Acoplada à Caixa	28
Figura 28 – Encapsulado	28
Figura 29 – Teste Com Banco de Dados	29
Figura 30 – Página Inicial.....	30
Figura 31 – Página de Buscas Integrada ao Google Maps	31
Figura 32 – Aguardando Localização GPS	32
Figura 33 – Aguardando Localização GPS	33
Figura 34 – Banco de Dados Atualizado	33
Figura 35 – Exibição na Página Web	34
Figura 36 – Exibição da Localização no Mapa	34
Figura 37 – Exibição da Localização Com Imagens de Satélite	35
Figura 38 – Exibição da Rota no Mapa – <i>Wi-fi</i>	35
Figura 39 – Exibição da Rota Com Imagens de Satélite – <i>Wi-fi</i>	36
Figura 40 – Exibição da Rota no Mapa – GPRS	36
Figura 41 – Exibição da Rota Com Imagens de Satélite – GPRS.....	37
Figura 42 – Pontos de Localização na Mesma Avenida - GPRS	37
Figura 43 – Dados Coletados Durante o Teste de <i>Backup</i>	38
Figura 44 – Envio dos Dados de Backup ao Servidor.....	39
Figura 45 – Exibição da Rota no Mapa – <i>Backup</i>	39
Figura 46 – Exibição da Rota Com Imagens de Satélite– <i>Backup</i>	40
Figura 47 – Pontos de Localização na Mesma Avenida – <i>Backup</i>	40

LISTA DE TABELAS

Tabela 1 : Relação das Cores a Conexão Realizada.....	16
---	----

LISTA DE ABREVIATURAS E SIGLAS

GPRS	General Packet Radio Service
GPS	Global Positioning System
IOT	Internet Of Things
NAVSTAR GPS	Sistema Navigation Satellite Timing And Ranging Global Positioning System
SPI	Serial Peripheral Interface
SQL	Structured Query Language
TEC	Total Electron Content

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Contextualização	5
1.2 Justificativa.....	7
1.3 Objetivos.....	7
1.3.1 Objetivo Geral	7
1.3.2 Objetivos Específicos	7
1.4 Metodologia.....	7
1.5 Estrutura do Projeto de Graduação	8
2 REFERENCIAL TEÓRICO	9
2.1 Tecnologia GPS	9
2.1.1 A Ionosfera e Sua Interferência Em Sistemas De GPS	9
2.2 Dispositivos e Tecnologias	10
2.2.1 ESP 32	10
2.2.2 NEO-6M	11
2.2.3 SIM 800L.....	11
2.2.4 <i>Web Service</i>	11
2.3 Considerações Preliminares	12
3 DESENVOLVIMENTO DO <i>HARDWARE</i>.....	15
3.1 Sistema de <i>Backup</i>	15
3.2 Alimentação	17
3.2.1 Controle de Carga.....	18
3.3 Protótipo	20

3.4 Desenvolvimento da Placa de Circuito Impresso.....	21
3.4.1 Projeto.....	21
3.4.2 Confeção.....	22
3.5 Montagem.....	26
4 DESENVOLVIMENTO DO SISTEMA WEB.....	29
4.1 Banco de Dados MySQL	29
4.2 Página Web	30
4.3 Google Maps API.....	30
5 RESULTADOS	32
5.1 Teste Com Localização Fixa	32
5.2 Teste Com Deslocamento – Conexão <i>Wi-Fi</i>	35
5.3 Teste Com Deslocamento – Conexão GPRS.....	36
5.4 Teste Com Deslocamento – <i>Backup</i>	38
6 CONCLUSÃO.....	41
6.1 Trabalhos Futuros	41
REFERÊNCIAS	42
APÊNDICE A – Código Sistema De Rastreamento	44
APÊNDICE B – Código em PHP Inserção no Banco de Dados.	53
APÊNDICE C – Código da Página Inicial.	54
APÊNDICE D – Criação de Arquivo XML a Partir do Banco de Dados.	56
APÊNDICE E – Código Para Página com Integração com o <i>Google Maps</i>.	57

1 INTRODUÇÃO

1.1 Contextualização

Sistemas baseados em localização existem mesmo antes da popularização do GPS (*Global Positioning System*). Como o caso da cidade de Londres - Inglaterra, que em 1992 já havia um sistema em fase em teste, o *Countdown*, onde os ônibus da linha 18 possuíam um sistema embarcado, que se comunicavam com *transponders* posicionados em locais específicos, cada ônibus possuía uma identificação própria e com isso era possível obter a localização de cada veículo, nesse cenário os dados eram transmitidos via rádio para uma central. As informações eram repassadas para o usuário através de painéis instalados em pontos de parada de ônibus, informando o tempo de chegada dos veículos equipados com o sistema (MAGALHÃES, 2008).

Atualmente já é possível desenvolver plataformas robustas e eficientes de rastreamento. O sistema desenvolvido na Inglaterra apesar de sofisticado para época, era bastante limitado, pois não só era necessário equipar os ônibus, mas também espalhar *transponders* pela rota. Além disso a comunicação também era limitada, pois a localização só seria obtida no momento em que o *transponder* e o dispositivo embarcado estivessem próximos. Todos esses problemas são resolvidos pelo uso do GPS, pois a localização pode ser obtida a todo instante e em qualquer lugar, contando ainda com a flexibilidade, pois o mesmo dispositivo pode ser utilizado para o caso citado ou em outras aplicações.

Com o avanço da tecnologia e barateamento do acesso á internet, a integração de sistemas e dispositivos se tornou mais dinâmica. Em consonância, inovações ligadas a rede estão surgindo constantemente, abrindo espaço ao acesso de informações que outrora nem se quer existiam, principalmente no que tange cidades inteligentes, no desenvolvimento de soluções que tragam melhor qualidade de vida a seus moradores.

Algumas cidades já contam com sistemas que monitoram o deslocamento dos ônibus de transporte urbano, e repassa ao usuário uma estimativa de tempo de sua chegada até o ponto desejado, o que auxilia o controle de tempo e organização dos usuários. Entretanto, muito ainda pode ser feito, um dos problemas recorrentes nas zonas urbanas é o controle do descarte do lixo, levando em consideração que é comum que animais rasguem os sacos de lixo e espalhem todo esse material indesejado nas ruas e calçadas, como os moradores possuem no máximo os dias de coleta, o controle do descarte é deficitário.

Uma solução é a implantação de um sistema de rastreamento, não só nos caminhões que realizam a coleta como também nos carrinhos dos varredores de rua, onde a localização dos mesmos pode ser enviada para um servidor web, e repassada para os habitantes através de um aplicativo móvel, de forma que o usuário possa obter um horário estimado para a coleta em sua rua, bem como acompanhar a limpeza de sua cidade. O uso de IOT (*Internet of Things*) em cidades, se bem empregada, tem um potencial enorme de melhorar a vida de seus habitantes.

Com a implantação de sistemas embarcados nos serviços de ônibus e limpeza urbana, a cidade passa a “conversar” com seus moradores. Entretanto, ainda é necessário desenvolver os sistemas que irão compor estas cidades inteligentes, pois eles deverão ser capazes de coletar e enviar os dados ao usuário final. Para que isso seja viável, é necessário desenvolver plataformas focadas em IOT e possíveis de implementar no desenvolvimento urbano.

Com um bom planejamento, é possível desenvolver plataformas capazes de suprir diversas demandas. Neste trabalho será projetado um sistema com o uso da tecnologia de monitoramento de localização GPS. Onde o mesmo deve ser versátil, podendo ser aplicado ao rastreamento veicular e projetos de cidades inteligentes.

Levando em consideração as características desejadas no escopo do projeto, o microcontrolador utilizado será o ESP32. Este microcontrolador tem se destacado em projetos que envolvam IOT, pois, seu chip já conta com *Wi-Fi* integrado, possui dois núcleos de processamento e opera em baixa potência, o que implica baixo consumo de energia em sua operação.

Como a conexão *Wi-Fi*, possui limitação no que se refere a distância, a comunicação com o servidor web será realizada principalmente via GPRS (*General Packet Radio Service*). Para isso é necessário o uso um módulo GSM, que se comunique com o microcontrolador através das portas Rx/Tx. Por mais que a conexão principal seja feita por GPRS, caso haja falhas na comunicação, os dados gravados poderão ser enviados assim que uma rede *Wi-Fi* conhecida for detectada, esta redundância mitiga a possibilidade de perda de dados do trajeto traçado.

1.2 Justificativa

Os sistemas de rastreamento podem possuir diversas aplicações. Serviços de carona e de entrega de comida, são exemplos claros, onde geralmente o rastreamento se dá através do próprio smartphone de um colaborador.

Entretanto, há casos em que são necessários sistemas dedicados, como serviços baseados em localização aplicados a cidades inteligentes. No que tange a gestão de lixos e resíduos, o registro dos trajetos realizadas pelos caminhões de lixo e varredores, pode contribuir com a administração pública no controle e planejamento das rotas realizadas, para de garantir o atendimento á toda população.

A fim de atender esta demanda, o sistema proposto será desenvolvido. Podendo ser aplicado em outras soluções que necessitem de um sistema de rastreamento dedicado.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um sistema de rastreamento confiável, utilizando o microcontrolador ESP32 em seu desenvolvimento.

1.3.2 Objetivos Específicos

1. Gravar os dados lidos em um cartão micro SD, e conectar o dispositivo ao servidor utilizando conexão GPRS, provida de um módulo GSM.
2. Comparar e testar as conexões *Wi-Fi* e GPRS, a fim de determinar o melhor momento para o uso de cada tecnologia.
3. Testar o dispositivo desenvolvido, analisando seu funcionamento e rotas obtidas.

1.4 Metodologia

A primeira etapa deste projeto consiste em um levantamento de todos os itens que serão necessários para sua construção. O microcontrolador será o ESP32, que será programado para gerir o sistema, para se obter a localização será utilizado o módulo GPS NEO-6M em conjunto com sua antena, este módulo será responsável por obter os dados de longitude, latitude, velocidade e hora; os dados obtidos serão enviados ao servidor, caso haja falhas de conexão

com a internet, serão gravados localmente em um cartão micro SD, de modo que o trajeto percorrido não seja perdido.

A segunda etapa será o desenvolvimento do *software* do sistema proposto, e também do circuito de controle de carga, que será simulado no Proteus, antes de sua montagem.

Então a terceira etapa será iniciada, onde um protótipo será montado em protoboard. Aqui os primeiros testes serão realizados, onde serão verificados os seguintes itens:

- Coerência da longitude e latitude obtida;
- Armazenamento local;
- Conexão *wi-fi*;
- Conexão GPRS;

Com um protótipo em pleno funcionamento em protoboard, uma placa de circuito impresso será projetada e confeccionada.

1.5 Estrutura do Projeto de Graduação

Este trabalho possui 6 capítulos. No primeiro capítulo tem-se o texto introdutório, contextualização, justificativa, objetivos do trabalho e a metodologia de seu desenvolvimento.

O Capítulo 2 aborda a fundamentação teórica, onde as principais tecnologias e dispositivos são apresentados.

No Capítulo 3, todas as etapas da construção do *hardware* são apresentadas, contando com o sistema de *backup*, alimentação, circuito para controle de carga e também com protótipo do sistema em *protoboard*.

No Capítulo 4 são apresentadas as etapas e tecnologias para o desenvolvimento do sistema *web*, abordando as configurações do banco de dados, página inicial do sistema e a integração com a API do *Google Maps*.

Os resultados dos testes realizados são apresentados no Capítulo 5, com os resultados obtidos em 4 testes distintos, sendo um teste com localização fixa e 3 com deslocamento, variando a maneira como os dados são coletados e enviados.

No Capítulo 6 são apresentadas as conclusões obtidas com o desenvolvimento deste trabalho. Além disso, todo o código desenvolvido está disponível nos Apêndices A, B, C, D e E.

2 REFERENCIAL TEÓRICO

2.1 Tecnologia GPS

O sistema Navstar GPS (*Navigation Satellite Timing and Ranging Global Positioning System*), popularmente conhecido como GPS teve seu desenvolvimento iniciado em 1973, com fundos do Departamento de Defesa dos Estados Unidos. Desde 1995, o GPS tem sido considerado em pleno funcionamento, sua concepção se deu para suprir a necessidade de se obter a posição instantânea e a velocidade de um ponto em qualquer lugar do planeta terra, ainda hoje é utilizado em diversas aplicações (ASSIS, 2010).

Para seu funcionamento, o sistema GPS opera com 24 satélites divididos em 6 planos de órbita com inclinação de 55°, onde são alocados 4 satélites em cada plano, com separação em arcos de 60°. Também existem de 2 a 5 satélites reservas que orbitam a terra, prontos para serem utilizados imediatamente, se necessário (GONÇALVES E TAVARAYAMA, 2011).

Desde sua concepção, o GPS vem revolucionando a interação do homem com seu meio. Através da latitude, longitude e altitude é possível determinar uma posição em qualquer ponto no globo terrestre. Apesar de seu desenvolvimento ter se iniciado com fins bélicos, hoje seu uso está nas mãos de civis e empresas, com aplicações que atendem desde problemas do dia a dia, como encontrar um determinado endereço, a controle de tráfego de aviões.

Há pouco mais 20 anos, Paz, Ferreira e Cugnasca (1998) já consideravam o GPS a mais sofisticada tecnologia já criada para se obter uma localização na terra, onde já era possível se obter uma precisão na casa de centímetros. Com o passar o tempo esta tecnologia se popularizou, e hoje, é a tecnologia de localização mais utilizada no mundo.

2.1.1 A Ionosfera e Sua Interferência Em Sistemas De GPS

A formação da ionosfera se dá pela ação da radiação solar magnética na atmosfera da terra. A radiação cósmica também tem seu papel, apesar de pequeno, contribui com a manutenção da ionosfera noturna (SALOMONI, 2008 apud. KIRCHHOFF, 1991). Possui características dispersivas, deste modo a propagação de uma onda eletromagnética depende diretamente de sua frequência. Os sinais de GPS são afetados pela ionosfera, podendo haver atrasos na modulação do sinal, e o avanço na fase da onda portadora. Irregularidades na

Ionosferas afetam a precisão do GPS, reduzindo sua confiabilidade (SALOMONI, 2008 apud. SKONE, 2000).

Segundo Salomoni, Matsouka e Souza (2011), eventos solares (explosões solares e ejeções coronais de massa) podem provocar tempestades geomagnéticas, que alteram o TEC (*Total Electron Content*), gerando mudanças repentinas na ionosfera, influenciando nos resultados de sistemas de localização por satélite.

Devido as possíveis interferências ocasionadas pelos distúrbios na ionosfera, pode-se haver erros na localização obtida. Por isso, caso haja variações abruptas nos dados recebidos via GPS, pode ser um indicativo que há uma anormalidade naquele momento.

2.2 Dispositivos e Tecnologias

2.2.1 ESP 32

Quando se trata de desenvolvimento de sistemas inteligentes, os microcontroladores são indispensáveis, pois são eles os responsáveis por integrar e gerenciar todos os componentes e dispositivos envolvidos. O microcontrolador ESP32 foi desenvolvido pela *Espressif Systems*, empresa com sede em Xangai. Este microcontrolador possui duas espécies de comunicação sem fio, *Wi-Fi* e *bluetooth*; foi desenvolvido para aplicações móveis, eletrônicos portáteis e IOT, além disso é otimizado para consumir pouca energia durante sua operação (ESPRESSIF SYSTEMS, 2019).

Principais características:

- *CPU: Xtensa® dual-core 32-bit LX6*
- *Memória RAM: 520 KB*
- *Memória ROM: 448 KB*
- *Memória Flash: 4MB*
- *Wi-Fi 2,4 GHz*
- *Bluetooth BLE 4.2*

2.2.2 NEO-6M

Uma parte fundamental de um sistema de rastreamento é o receptor de sinal GPS, pois este dispositivo irá receber as informações dos satélites e enviar para o microcontrolador. O módulo GPS NEO-6M é uma opção de baixo custo e fácil utilização. É um receptor GPS independente e de alto desempenho, capaz de realizar buscas paralelas, permitindo encontrar satélites instantaneamente. A localização fornecida possui precisão de 2,5 metros; sua operação é limitada a velocidades de até 500 m/s e a altitude máxima de 50.000 metros (U-BLOX, 2011).

Visto que a localização recebida pelo módulo NEO-6M é obtida diretamente pelos satélites GPS, os dados de localização não dependem do sinal de redes de telefonia, como ocorre com a tecnologia A-GPS. Com isso, é possível realizar a coleta de dados constantemente.

2.2.3 SIM800L

Com os dados de localização recebidos e lidos pelo microcontrolador, os mesmos devem ser transmitidos para um servidor *web*, uma opção viável e de ampla cobertura de sinal é o uso de um transmissor GPRS. O SIM800L é um módulo GSM produzido pela SIMCOM. Permite efetuar ligações e envio de SMS através da conexão GSM, também é possível se conectar com a internet através da conexão GPRS, sendo compatível com diversas plataformas, como: Arduino, Pi, PIC, ESP entre outras (USINAINFO, 2019).

Características principais:

- Tensão de operação: 3,7V a 4.2V
- Corrente de pico: 2A
- Quad-Band: 850/900/1800/1900MHz
- Temperatura: -40 °C a +85 °C

2.2.4 Web Service

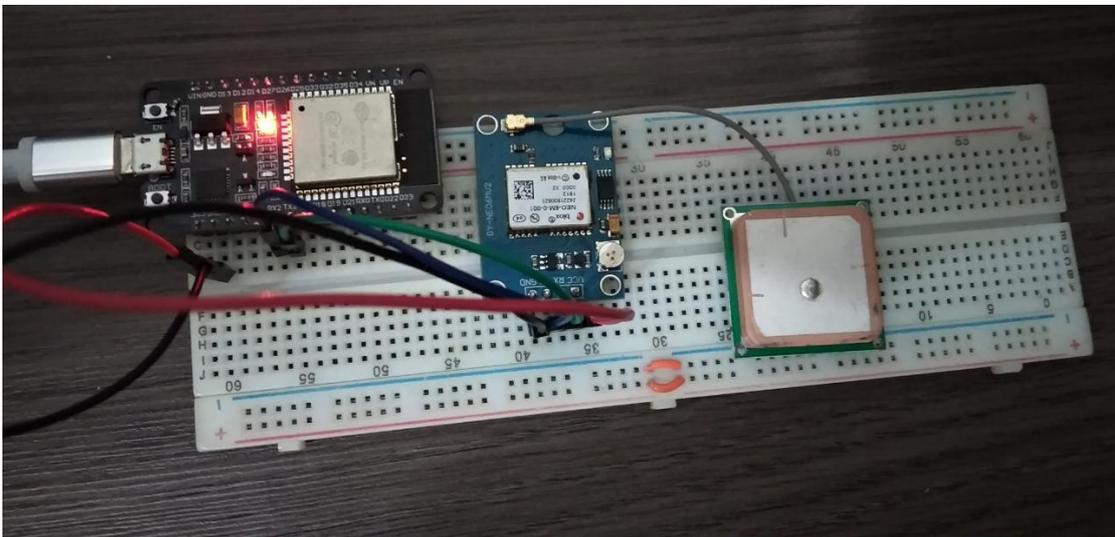
Os serviços ligados a internet (*Web Services*) de maneira geral, necessitam de um servidor web, para armazenamento e processamento de dados. Em sistemas de rastreamento

seu uso se torna necessário, visto que os dados colhidos pelo rastreador devem ser armazenados e transmitidos ao usuário. O rastreador GPS informa um conjunto de pontos, que são enviados ao servidor, onde são armazenados e processados; haja vista que para formar uma rota, os pontos obtidos devem ser agrupados, existindo a possibilidade de criar um histórico de rotas, também é possível solicitar um histórico de posições específicas, variando com a necessidade do usuário.

2.3 Considerações Preliminares

Alguns testes iniciais foram realizados a fim de verificar o funcionamento do microcontrolador e do módulo GPS. Para o desenvolvimento do código foi utilizada a IDE Arduino, pois o microcontrolador ESP32 é multiplataforma, e esta é uma das plataformas suportadas. O primeiro teste consiste em ler os dados recebidos pelo módulo GPS e exibi-los no monitor serial da IDE Arduino. O circuito montado e os resultados obtidos podem ser vistos nas Figuras 1 e 2, respectivamente.

Figura 1 – Montagem Preliminar



Fonte: Autor

Figura 2 – Localização, Velocidade, Data e Hora

```

COM3
01:08:03.895 -> Localização: -10.198728,-48.900629 Velocidade:,0.09 Data/Hora: 11/21/2019 04:08:03.00
01:08:03.977 -> Localização: -10.198728,-48.900629 Velocidade:,0.09 Data/Hora: 11/21/2019 04:08:03.00
01:08:04.215 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.337 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.414 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.537 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.617 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.694 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.816 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.895 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:04.976 -> Localização: -10.198730,-48.900628 Velocidade:,0.07 Data/Hora: 11/21/2019 04:08:04.00
01:08:05.216 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.339 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.421 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.497 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.616 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.696 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.816 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.896 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:05.976 -> Localização: -10.198732,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:05.00
01:08:06.216 -> Localização: -10.198734,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:06.00
01:08:06.336 -> Localização: -10.198734,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:06.00
01:08:06.416 -> Localização: -10.198734,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:06.00
01:08:06.509 -> Localização: -10.198734,-48.900628 Velocidade:,0.04 Data/Hora: 11/21/2019 04:08:06.00

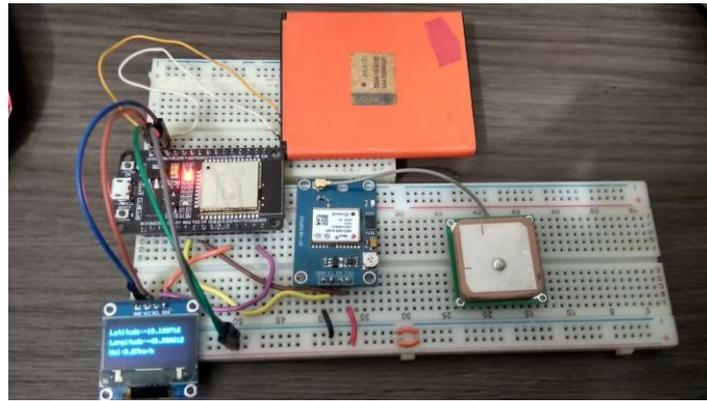
```

Fonte: Autor

Os dados exibidos no monitor serial foram: latitude, longitude, velocidade, data e hora. Também é possível informar a altitude e o número de satélites conectados. É importante observar que o horário recebido pelo módulo GPS corresponde ao GMT-0, com isso, está 3 com horas adiantadas em relação ao horário de Brasília, para ajustar ao fuso-horário desejado é necessário corrigir a hora via programação no microcontrolador.

Para realizar testes sem que o microcontrolador esteja conectado ao computador, foi adicionado um display OLED de 0,96'' e uma bateria, tal como visto na Figura 3. o sistema foi programado para exibir a latitude, longitude e velocidade, para fins de comparação, estes dados foram inseridos no *Google Maps* (Figura 4), e a localização indicada correspondeu com a localização do sistema.

Figura 3 – Sistema Com Display e Bateria



Fonte: Autor

Figura 4 - Ponto Obtido No Google Maps



Fonte: Autor

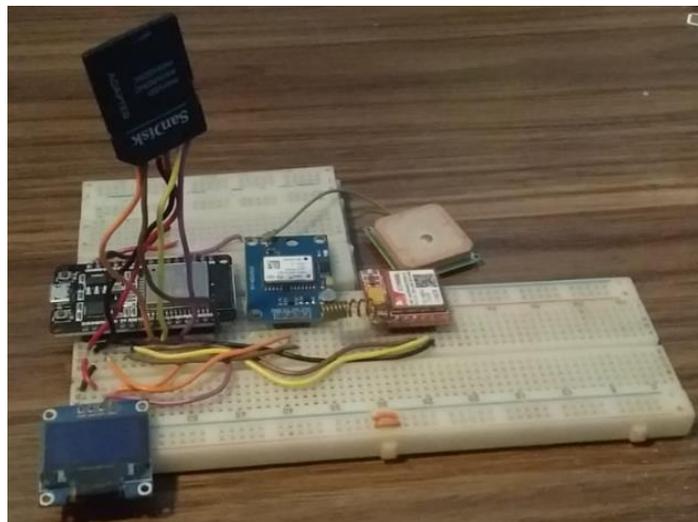
Os testes preliminares indicaram confiabilidade da localização obtida. Entretanto foi detectado um atraso de aproximadamente 60 segundos, em referência ao momento que o módulo GPS é ligado até que inicie a recepção dos dados, porém após a conexão ser estabelecida, permanece constante.

3 DESENVOLVIMENTO DO *HARDWARE*

Este capítulo é dedicado aos aspectos construtivos e de funcionamento do sistema de rastreamento. Todo o código desenvolvido para sua operação está disponível no Apêndice A.

A partir do sistema já implementado com o microcontrolador ESP32 e o módulo GPS NEO-6M, foi adicionado um módulo SIM800L para possibilitar o envio dos dados via GPRS. Como a memória flash disponível no ESP32 é de apenas 4MB, um cartão de memória de 4GB também foi integrado (Figura 5), para armazenar a localização quando não houver conexão com a internet.

Figura 5 – Sistema Montado em Protoboard

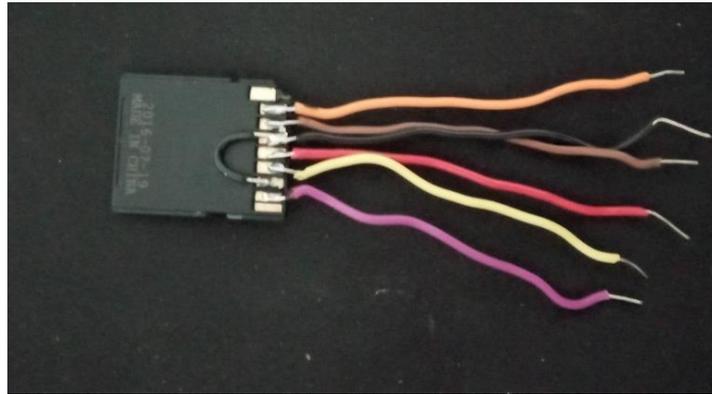


Fonte: Autor

3.1 Sistema de *Backup*

Os pinos SPI (*Serial Peripheral Interface*) do microcontrolador foram conectados ao cartão de memória por cabos de cores variadas, deste modo, fica mais fácil identificar os pontos de conexão. A Figura 6 mostra o cartão de memória com os cabos já soldados, e a tabela 1 apresenta relação das cores com a conexão realizada.

Figura 6 - Cabos Soldados ao Cartão de Memória



Fonte: Autor

Tabela 1 : Relação das Cores a Conexão Realizada

Cor	Cartão de Memória	ESP32
Roxo	DAT0	MISO (Pin. 19)
Amarelo	CLK	SCK (Pin. 18)
Vermelho	VDD	3V3
Preto	GND	GND
Marrom	CMD	MOSI (Pin. D23)
Laranja	DAT3	CS (Pin. D05)

Fonte: Autor

Os dados coletados pelo sistema são latitude, longitude, velocidade, data e hora. Que são enviados para o servidor a cada *loop* do sistema. Caso não exista conexão com a internet disponível, um *backup* é criado no cartão de memória, sendo um arquivo de texto para cada ponto, como pode ser visto na Figura 7.

Figura 7 - Arquivos de Backup

Name	Date modified	Type	Size
4.6.2021_16.25.14.0	01/01/1980 00:02	Text Document	1 KB
4.6.2021_16.25.22.50	01/01/1980 00:02	Text Document	1 KB
4.6.2021_16.25.26.0	01/01/1980 00:02	Text Document	1 KB
4.6.2021_16.25.30.0	01/01/1980 00:02	Text Document	1 KB
4.6.2021_16.25.34.0	01/01/1980 00:02	Text Document	1 KB
4.6.2021_16.25.57.0	01/01/1980 00:03	Text Document	1 KB
4.6.2021_16.26.5.0	01/01/1980 00:03	Text Document	1 KB
4.6.2021_16.26.9.0	01/01/1980 00:03	Text Document	1 KB
4.6.2021_16.26.27.73	01/01/1980 00:03	Text Document	1 KB
4.6.2021_16.26.36.0	01/01/1980 00:03	Text Document	1 KB
4.6.2021_16.26.55.0	01/01/1980 00:04	Text Document	1 KB
4.6.2021_16.26.59.0	01/01/1980 00:04	Text Document	1 KB
4.6.2021_16.27.10.0	01/01/1980 00:04	Text Document	1 KB
4.6.2021_16.27.14.0	01/01/1980 00:04	Text Document	1 KB

Fonte: Autor

A cada ciclo, o sistema verifica se a conexão foi reestabelecida, assim que o GPRS voltar a funcionar, ou uma rede wifi conhecida foi encontrada, os arquivos de *backup* são enviados para o servidor.

3.2 Alimentação

Para suprir a necessidade energética do sistema, foi utilizada uma bateria de 8v e 4400mah, apresentada na Figura 8.

Figura 8 – Bateria



Fonte: Autor

Com intuito de aumentar a autonomia e promover economia de energia, um pequeno módulo solar foi incluído, com tensão e potência nominais de 12V e 1.5W respectivamente (Figura 9). Logo, sempre que exposto a luz solar, o sistema gerará sua própria energia, e conseqüentemente reduzirá a necessidade de intervenção humana no carregamento de sua bateria.

Figura 9 – Módulo Fotovoltaico de 1,5W



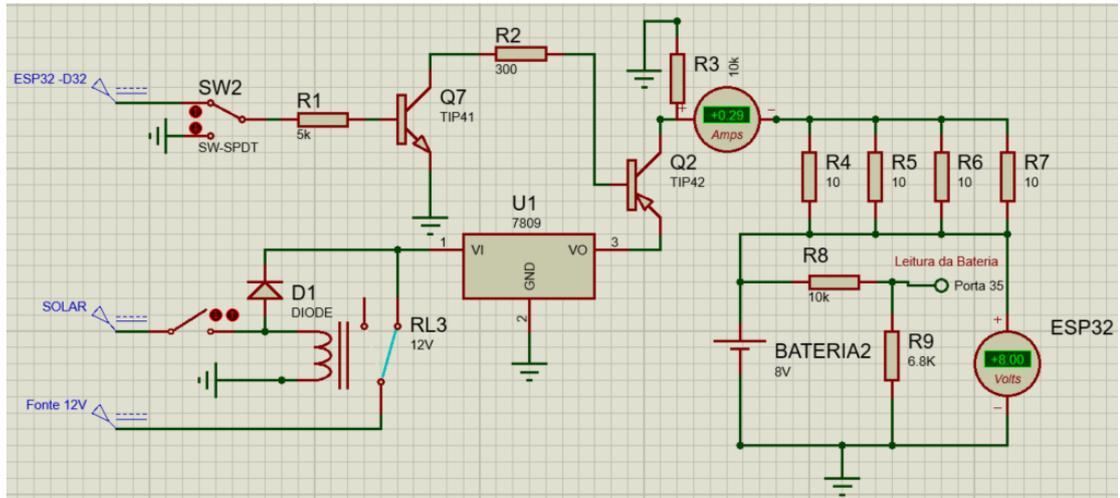
Fonte: Autor

3.2.1 Controle de Carga

O gerenciamento da alimentação do sistema é realizado pelo microcontrolador em conjunto com o circuito apresentado na Figura 10. Onde o controle de passagem de corrente e a leitura do nível de carga da bateria, são realizados pelo ESP32 o que torna o sistema mais barato e permite melhorias e otimizações através de atualizações de *software*.

Contudo, a seleção do método de carregamento, que pode ocorrer pelo módulo solar, ou através de uma fonte de 12V, é realizada pelo circuito de maneira independente. Ambas as possibilidades de carga são conectadas ao regulador de tensão LM7809, que mantém a tensão estabilizada em 9V.

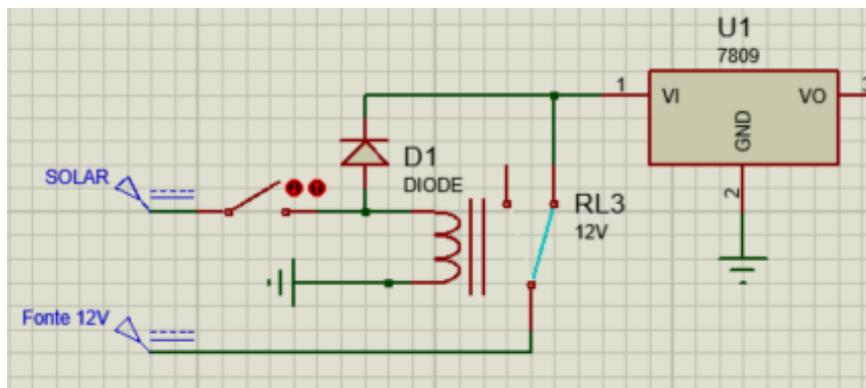
Figura 10 – Controle de Carga



Fonte: Autor

Sempre que houver tensão nos terminais do módulo solar, o relé estará polarizado, com a mudança de posição da agulha, a fonte será desconectada do regulador de tensão (Figura 11), portanto, a bateria será carregada prioritariamente a partir de uma geração própria e sustentável.

Figura 11 – Seleção de Método de Carregamento



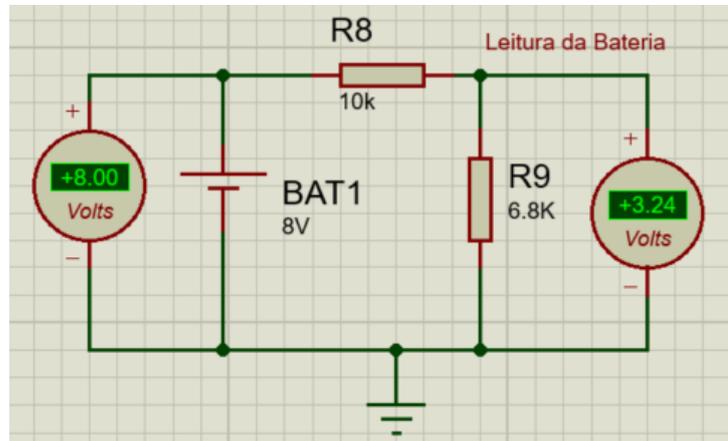
Fonte: Autor

para evitar sobrecarga na bateria. O monitoramento é realizado pelo pino 35, por leitura de tensão, e o controle se dá pelo pino 32 através da polarização de transistores.

O ESP32 possui tensão nominal de 3,3V podendo operar com tensão máxima de até 3,6V em seus terminais (ESPRESSIF SYSTEMS, 2019). Por esse motivo, foi feita uma divisão de tensão com um resistor de 10k Ω e outro de 6,8k Ω . No voltímetro a esquerda da Figura 12, tem-se a tensão nominal da bateria, já no voltímetro disposto a direita, tem-se a tensão dividida,

3,24V, deste modo a leitura pode ser realizada pelo microcontrolado, sem que ele seja danificado.

Figura 12 – Controle de Carga

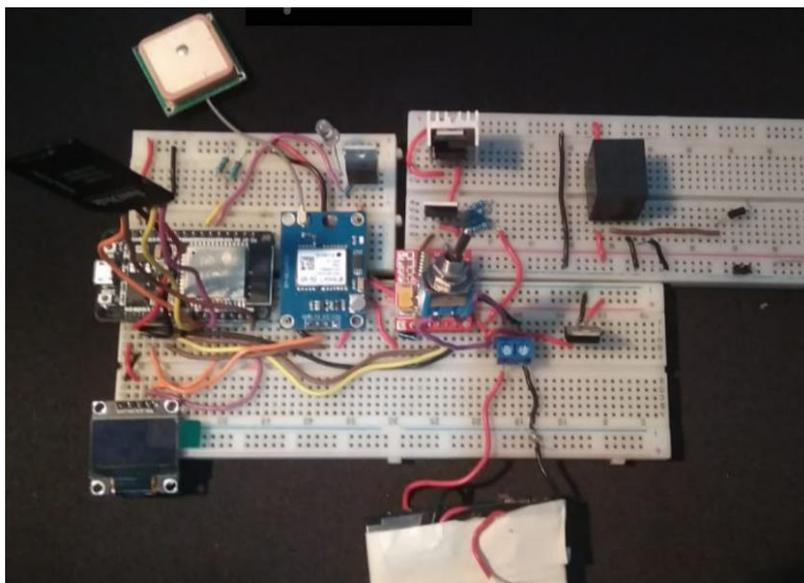


Fonte: Autor

3.3 Protótipo

Com o circuito de controle de carga já simulado, o mesmo foi integrado ao sistema na *protoboard*, vide Figura 13. Além disso, foi adicionado um dissipador de calor no regulador de tensão LM7809, para reduzir sua temperatura de operação e um regulador de tensão LM7805 de 5V, que alimenta os módulos GPS e GPRS.

Figura 13 – Protótipo em Protoboard



Fonte: Autor

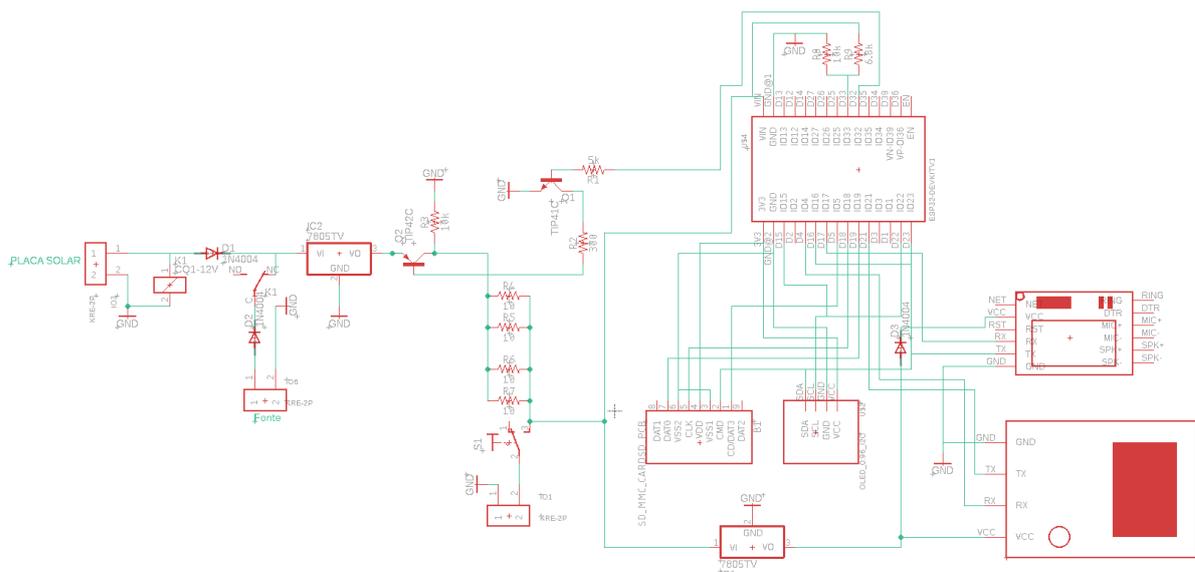
3.4 Desenvolvimento da Placa de Circuito Impresso

3.4.1 Projeto

A placa de circuito impresso foi desenvolvida no *Eagle*, na versão educacional. Para a conexão da bateria, módulo solar e fonte externa foram utilizados *bornes* de duas vias para cada um dos casos.

O primeiro passo é a montagem do esquemático, que contém todos os dispositivos, componentes e conexões necessárias para o funcionamento do sistema, como pode ser visto na Figura 14.

Figura 14 – Esquemático

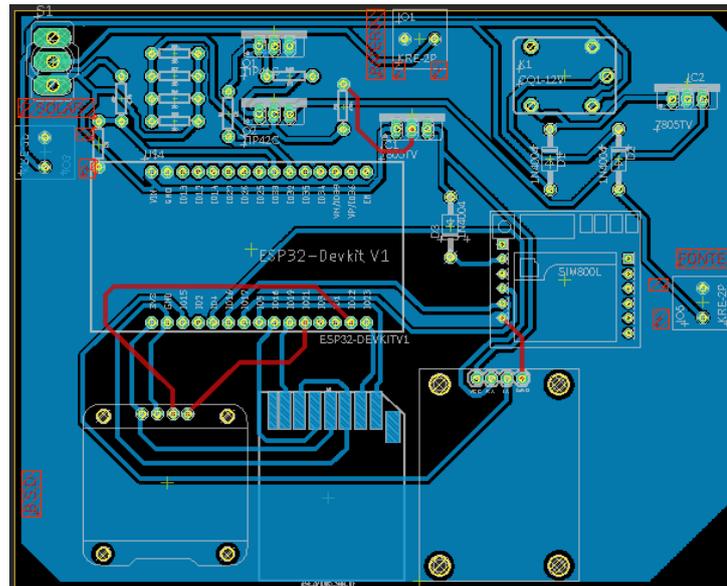


Fonte: Autor

Com a montagem do esquemático concluída, a placa foi projetada, onde todos os componentes foram dispostos de modo que se obtenha um circuito organizado e compacto. Como a confecção será feita manualmente, o projeto deve possuir no máximo duas camadas, para aplicação em uma placa de fenolite com as duas faces cobreadas.

Neste caso, tem-se a camada superior em vermelho e a inferior em azul, no circuito apresentado na Figura 15. Apenas quatro trilhas estão localizadas na primeira camada, pois condensar o máximo de trilhas na camada inferior, facilita a confecção manual.

Figura 15 – Projeto da Placa de Circuito Impresso



Fonte: Autor

3.4.2 Confeção

A confecção da placa foi realizada com uso de tinta fotossensível e corrosão. A placa foi polida para garantir maior adesão da tinta e eliminar quaisquer resíduos indesejados (Figura 16).

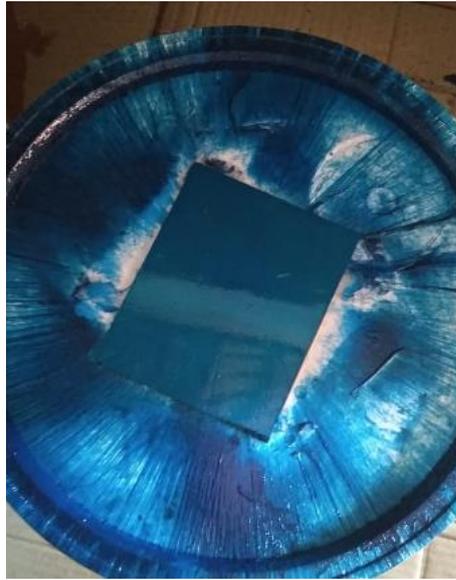
Figura 16 – Polimento da Placa



Fonte: Autor

Então a tinta foi aplicada suavemente sobre o cobre, e a placa centrifugada, de modo que houvesse um espalhamento uniforme por toda sua superfície (Figura 17), antes de prosseguir, a placa permaneceu 20 minutos em um forno para a cura da tinta.

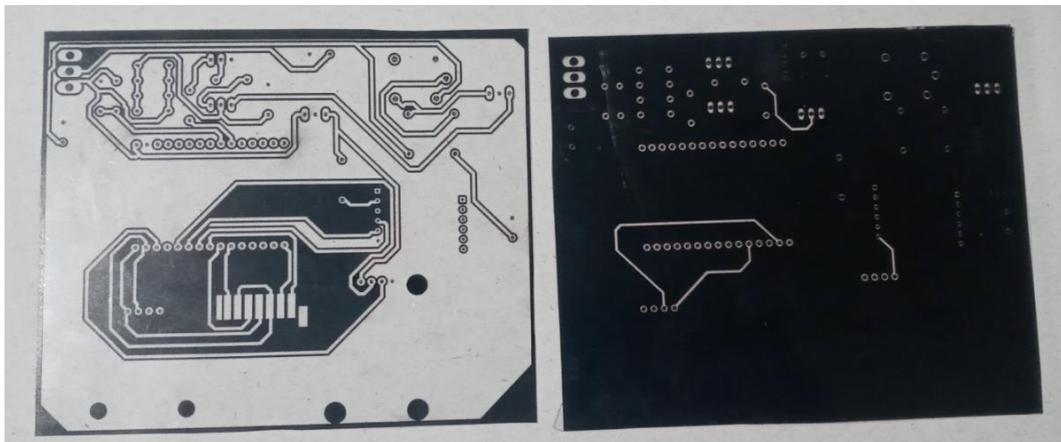
Figura 17 Aplicação da Tinta Fotossensível



Fonte: Autor

A transferência do circuito para a placa, ocorre pela exposição a radiação ultravioleta. Sendo assim, o circuito foi impresso em um fotolito com as cores invertidas (Figura 18).

Figura 18 - Molde o Circuito Impresso em Fitolito



Fonte: Autor

Em seguida, o fotolito foi posicionado sobre a placa cobreada, e uma placa de vidro foi posicionada sobre o fotolito, para garantir o contato e evitar que haja mudança de posição durante a exposição (Figura 19). Neste caso, a radiação é emitida por uma lâmpada UV de 28W por 3 minutos.

Figura 19 - Molde o Circuito Impresso em Fotolito



Fonte: Autor

O circuito foi revelado sendo submerso em uma solução com carbonato de sódio, onde toda a área que não sobreu exposição foi removida, vide Figura 20.

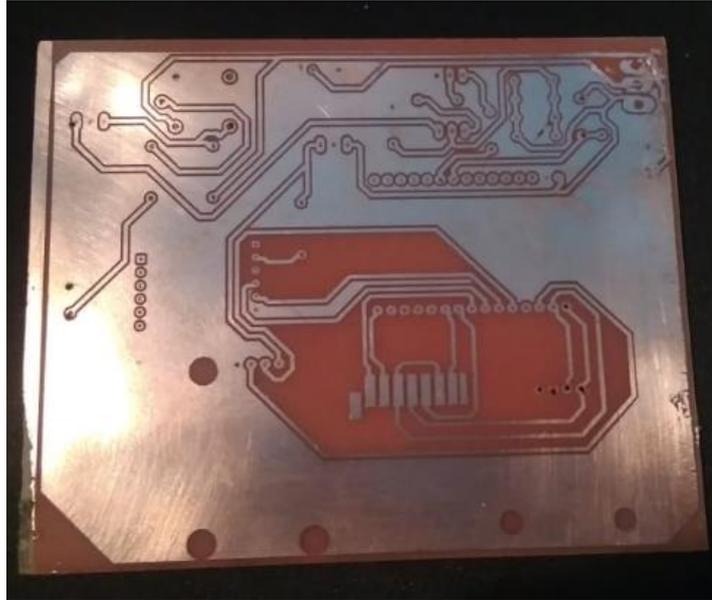
Figura 20 – Circuito Revelado



Fonte: Autor

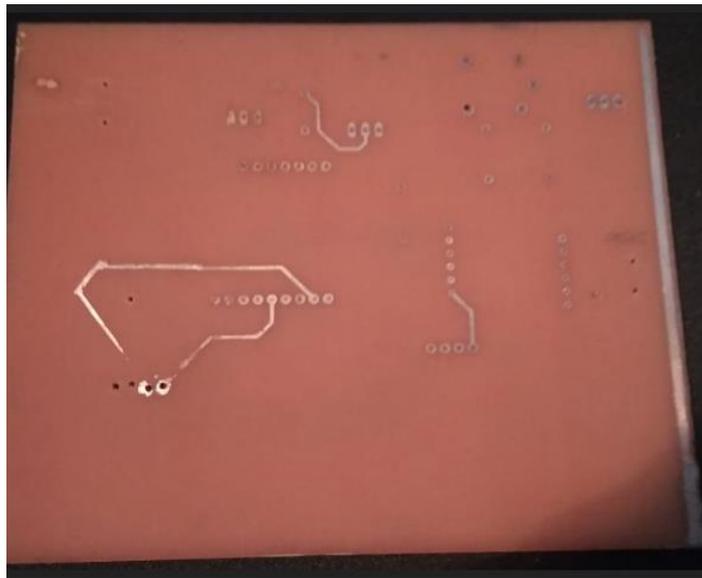
Então a placa foi submersa em uma solução com percoleto de ferro, para remover todo o cobre que está exposto; após a corrosão do cobre, o restante da tina foi removida com hidróxido de sódio. As Figuras 21 e 22 apresentam as camadas da placa de circuito impresso, após a finalização do processo.

Figura 21– Camada Inferior



Fonte: Autor

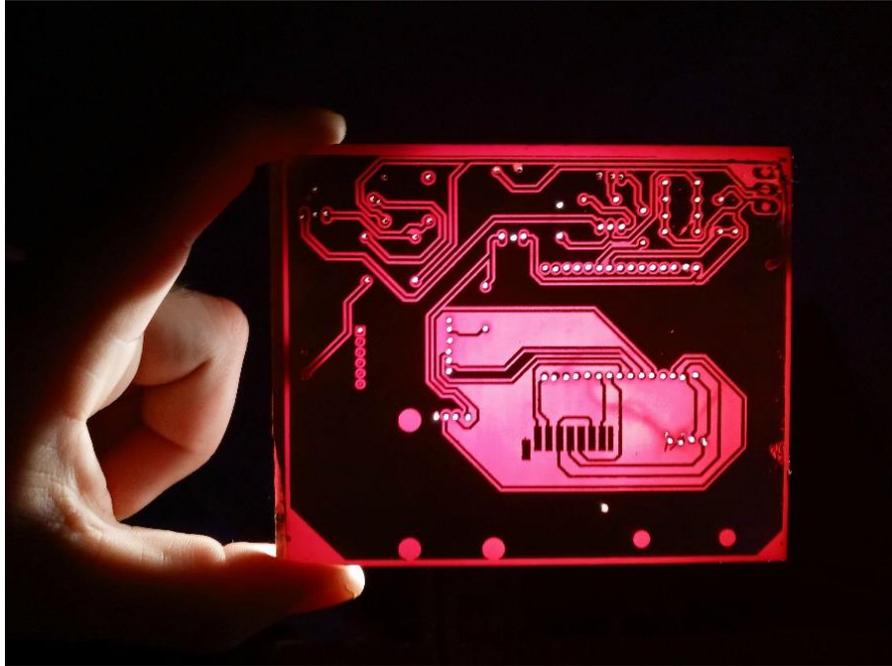
Figura 22 – Camada Superior



Fonte: Autor

A placa obtida foi posicionada contra a luz, para verificação das trilhas e dos furos (Figura 23).

Figura 23 – Verificação das Trilhas e Furos



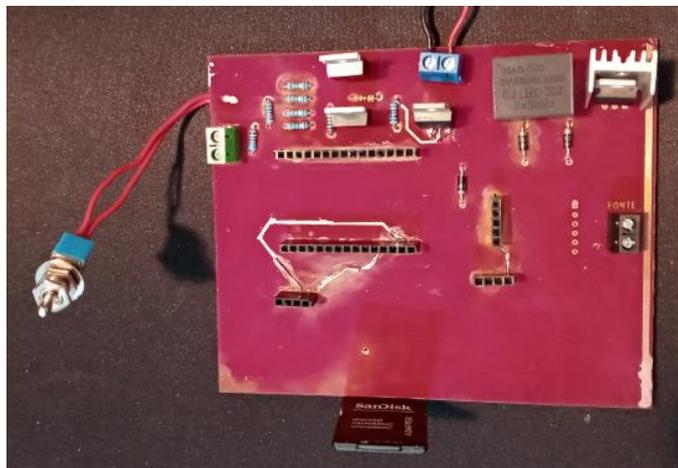
Fonte: Autor

3.5 Montagem

Os componentes foram acoplados e soldados a placa, seguindo as conexões já apresentadas no esquemático. Entretanto, o microcontrolador, módulo GPS, módulo GPRS e display não foram diretamente soldados, mas sim encaixados em barras de pinos fêmea.

Para as conexões da bateria, módulo solar e fonte de energia, foram utilizados *bornes* de duas vias. Cada *borne* possui uma cor diferente, o que facilitar sua identificação, sendo azul correspondente a bateria, o verde ao módulo solar e o preto a fonte (Figura 24).

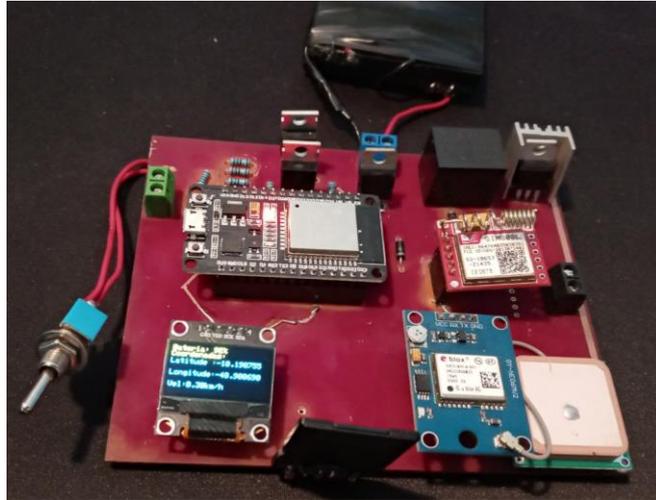
Figura 24 – Placa Com Componentes e Barras de Pino



Fonte: Autor

Em sequência, os dispositivos foram encaixados, e o sistema ligado. Com isso, têm-se um sistema funcional em operação, onde é possível visualizar no display, o nível de carga da bateria, latitude, longitude e velocidade, conforme a Figura 25.

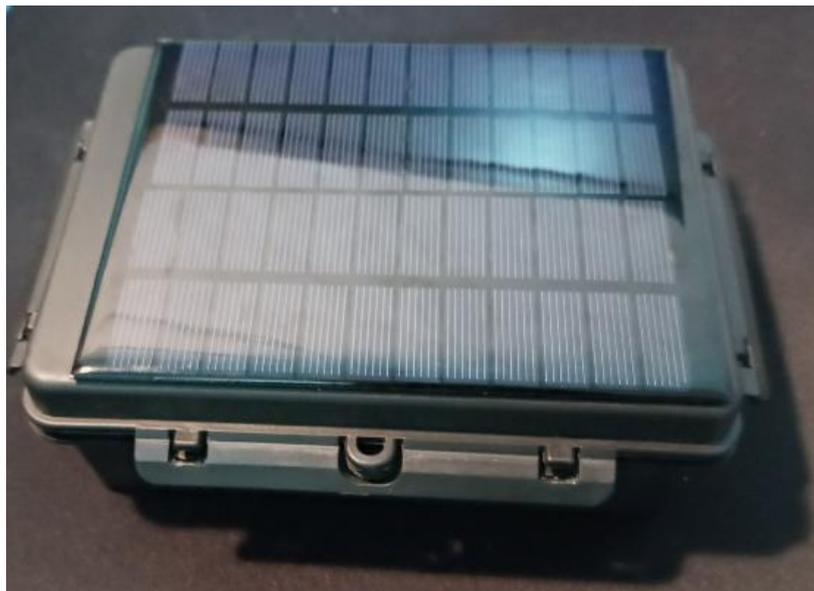
Figura 25 – Placa Todos os Componentes e Dispositivos



Fonte: Autor

Uma caixa hermética de plástico foi utilizada para o encapsulamento. Neste caso, o módulo solar foi colado à tampa com fita dupla face (Figura 26).

Figura 26 – Módulo Solar Acoplado à Caixa Hermética



Fonte: Autor

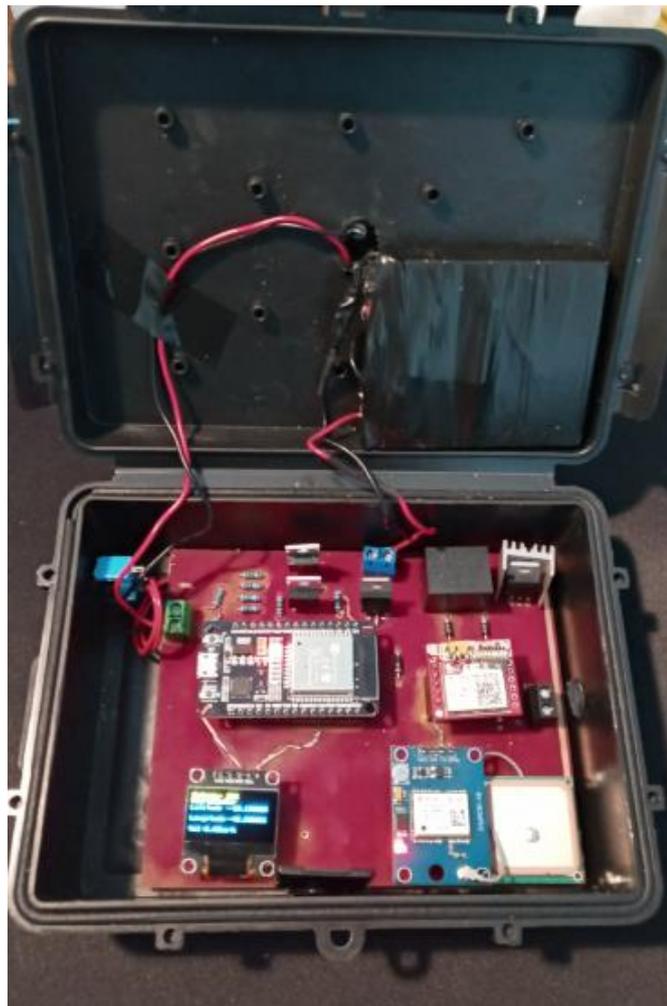
O sistema é ligado e desligado através de uma chave gangorra, conforme a Figura 27. Logo, foi necessário perfurar a tampa e duas paredes da caixa, para passagem dos cabos do módulo solar, encaixe da chave, e para conexão da fonte quando necessário (Figura 28).

Figura 27 – Chave Acoplada à Caixa



Fonte: Autor

Figura 28 – Encapsulado



Fonte: Autor

4 DESENVOLVIMENTO DO SISTEMA WEB

Uma página web foi desenvolvida para facilitar a visualização dos dados. Portanto, o usuário poderá acessar a localização atual ou obter um histórico de uma maneira simples e direta, necessitando apenas de um dispositivo conectado à internet.

Em consonância, o sistema desenvolvido foi integrado ao *Google Maps*. Sendo assim, é possível verificar a localização e a rota diretamente no mapa, ou mesmo em imagens de satélite.

4.1 Banco de Dados MySQL

Um banco de dados é uma coleção de dados estruturada, desde uma lista a uma coleção de fotografias. Os computadores são excelentes para lidar com uma grande quantidade de dados, por isso, um sistema de gerenciamento se torna necessário, o MySQL é um sistema *open source* de gerenciamento de banco de dados SQL (*Structured Query Language*), desenvolvido pela Oracle (ORACLE, 2020).

Neste caso, os dados são organizados por colunas, onde uma linha é criada para cada nova atualização. A inserção é feita por PHP e MySQL, com os dados recebidos via HTTP POST, o código responsável está disponível no Apêndice B; a tabela obtida pode ser vista na Figura 29. Por se tratar dos testes iniciais, os valores estão nulos pois ainda não correspondem aos dados coletados pelo sistema.

Figura 29 – Teste Com Banco de Dados

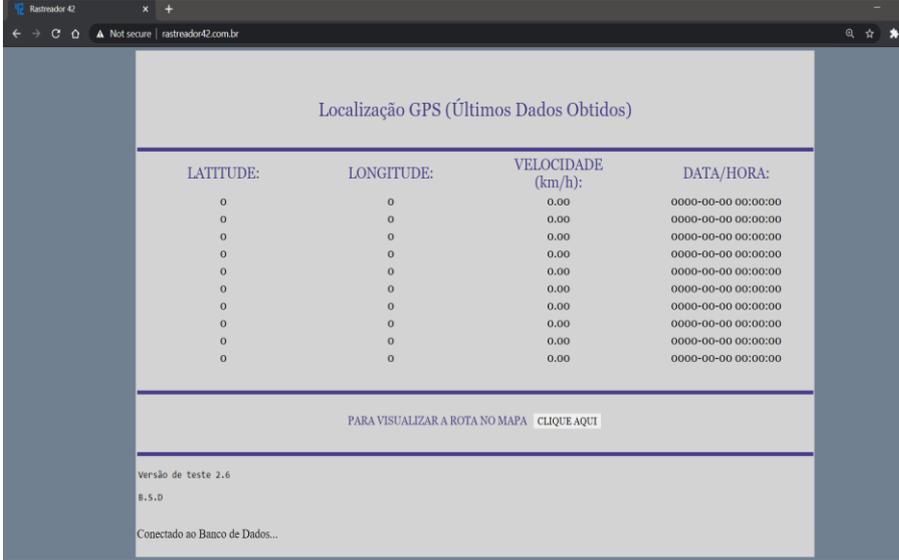
	Id	Latitude	Longitude	Velocidade	Data_Hora
<input type="checkbox"/>  Editar  Copiar  Apagar	1	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	2	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	3	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	4	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	5	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	6	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	7	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	8	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	9	0	0	0.00	0000-00-00 00:00:00
<input type="checkbox"/>  Editar  Copiar  Apagar	10	0	0	0.00	0000-00-00 00:00:00

Fonte: Autor

4.2 Página Web

A interface *web* foi construída em HTML, CSS e a integração com o banco de dados em PHP. A página inicial exibe as 10 (dez) últimas atualizações que estão contidas no banco de dados, contendo a latitude, longitude, velocidade, data e hora. Na Figura 30 tem-se a página desenvolvida exibindo o bando de dados de teste apresentado na Figura 29, o código está disponível no Apêndice C.

Figura 30 – Página Inicial



The screenshot shows a web browser window with the URL 'rastreador42.com.br'. The page title is 'Localização GPS (Últimos Dados Obtidos)'. It features a table with four columns: LATITUDE, LONGITUDE, VELOCIDADE (km/h), and DATA/HORA. The table contains 10 rows of test data, all with zero values for latitude, longitude, and velocity, and a timestamp of '0000-00-00 00:00:00'. Below the table is a button labeled 'PARA VISUALIZAR A ROTA NO MAPA CLIQUE AQUI'. At the bottom, there is a footer with the text 'Versão de teste 2.6', 'B.S.D', and 'Conectado ao Banco de Dados...'.

LATITUDE:	LONGITUDE:	VELOCIDADE (km/h):	DATA/HORA:
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00
0	0	0.00	0000-00-00 00:00:00

PARA VISUALIZAR A ROTA NO MAPA CLIQUE AQUI

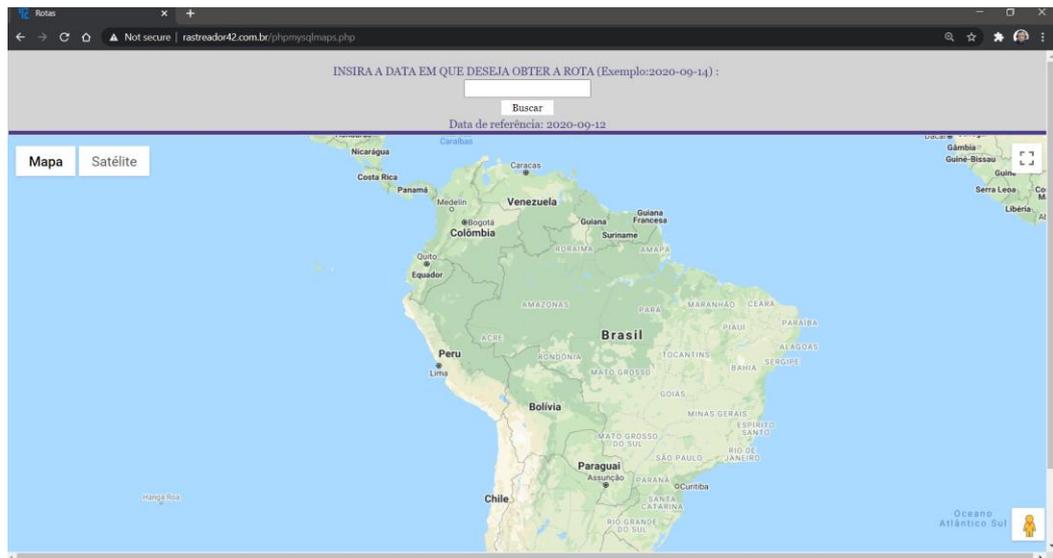
Versão de teste 2.6
B.S.D
Conectado ao Banco de Dados...

Fonte: Autor

4.3 Google Maps API

O *Google Maps* é um serviço de mapas que abrange o mundo todo. Onde é possível visualizar mapas, imagens de satélites e até mesmo condições de tráfego, com a API do Google Maps, pode-se utilizar estes recursos em páginas de terceiros (LUTHFI; KARNA; MAYASARI, 2019).

Pontando, com o uso da API do *Google Maps* é possível exibir os pontos de localização diretamente no mapa, permitindo uma visualização clara do trajeto percorrido. Ao clicar no botão “Clique Aqui” da página inicial, o usuário será redirecionado para a página mostrada na Figura 31, onde poderá buscar as rotas por data.

Figura 31 – Página de Buscas Integrada ao *Google Maps*

Fonte: Autor

Com a data inserida no campo de busca, será gerado um arquivo XML com os pontos de localização correspondentes encontrados no banco de dados (Apêndice D). A partir deste arquivo os pontos serão apresentados no mapa, esta integração com o Google Maps está disponível no Apêndice E.

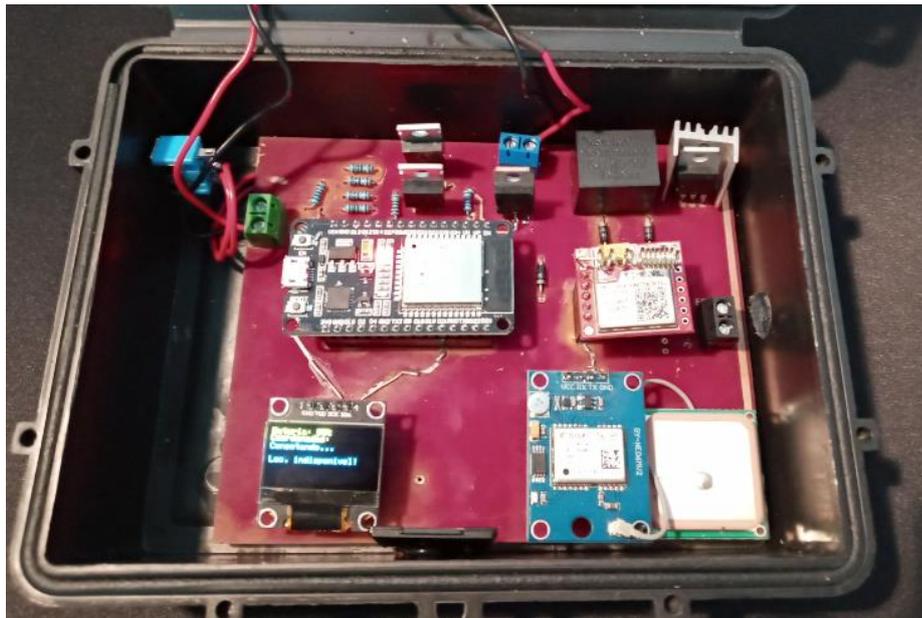
5 RESULTADOS

Todos os testes foram realizados em Paraíso do Tocantins – TO, Brasil, a fim de verificar o funcionamento do sistema desenvolvido. Efetuando análises do envio de dados para o servidor, gravação no banco de dados, *backup*, operação da página web e sua integração com o *Google Maps*.

5.1 Teste Com Localização Fixa

O primeiro teste foi realizado sem deslocamento. O sistema de rastreamento foi ligado, e após alguns segundos, passou a exibir os dados de localização no *display*, este atraso entre a inicialização até a localização ser exibida é o tempo que o módulo GPS necessita para estabelecer sua conexão (Figura 32).

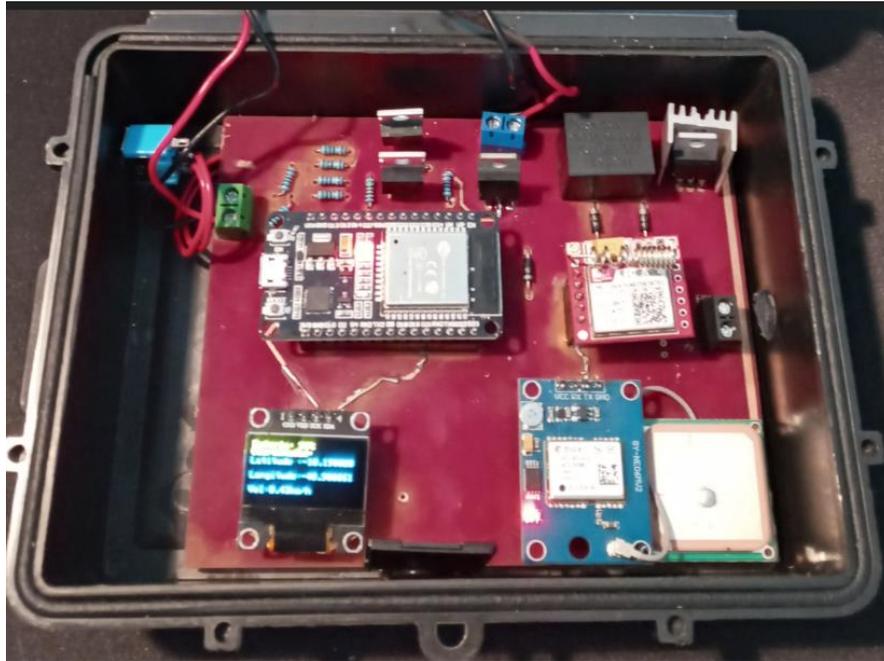
Figura 32 Aguardando Localização GPS



Fonte: Autor

Com a conexão estabelecida, os dados de localização e velocidade são exibidos no display (Figura 33).

Figura 33 - Aguardando Localização GPS



Fonte: Autor

Com o sistema de rastreamento em funcionamento, os dados coletados são enviados ao servidor. Na Figura 34 é possível visualizar a tabela do banco de dados atualizada.

Figura 34 – Banco de Dados Atualizado

				Id	Latitude	Longitude	Velocidade	Data_Hora
<input type="checkbox"/>				9039	-10.198725	-48.900667	0.30	2020-08-15 09:55:57
<input type="checkbox"/>				9038	-10.198693	-48.900687	0.24	2020-08-15 09:54:33
<input type="checkbox"/>				9037	-10.198726	-48.900717	0.33	2020-08-15 09:50:55
<input type="checkbox"/>				9036	-10.198726	-48.900717	0.33	2020-08-15 09:50:55
<input type="checkbox"/>				9035	-10.198758	-48.900733	0.67	2020-08-15 09:49:57
<input type="checkbox"/>				9034	-10.198736	-48.900711	0.07	2020-08-15 09:47:14
<input type="checkbox"/>				9033	-10.198737	-48.900708	0.44	2020-08-15 09:46:44
<input type="checkbox"/>				9032	-10.198766	-48.900661	0.07	2020-08-15 09:38:35
<input type="checkbox"/>				9031	-10.198761	-48.900689	0.72	2020-08-15 09:38:09
<input type="checkbox"/>				9030	-10.198697	-48.900698	0.56	2020-08-15 09:36:22
<input type="checkbox"/>				9029	-10.198713	-48.900725	1.30	2020-08-15 09:35:31
<input type="checkbox"/>				9028	-10.198735	-48.900685	0.20	2020-08-15 09:30:46
<input type="checkbox"/>				9027	-10.198702	-48.900717	0.61	2020-08-15 09:29:41
<input type="checkbox"/>				9026	-10.198689	-48.900737	1.30	2020-08-15 09:28:34
<input type="checkbox"/>				9025	-10.198721	-48.900711	0.00	2000-00-00 09:25:18
<input type="checkbox"/>				9024	-10.198753	-48.900741	0.07	2020-08-15 09:24:07
<input type="checkbox"/>				9023	-10.198763	-48.900758	0.09	2020-08-15 09:23:17
<input type="checkbox"/>				9022	-10.198764	-48.900758	0.04	2020-08-15 09:22:50
<input type="checkbox"/>				9021	-10.198755	-48.900751	0.06	2020-08-15 09:22:40
<input type="checkbox"/>				9020	-10.198740	-48.900740	0.09	2020-08-15 09:22:30
<input type="checkbox"/>				9019	-10.198739	-48.900666	0.02	2020-08-15 00:37:34

Fonte: Autor

O resultado obtido na página inicial do sistema web corresponde ao que foi gravado no banco de dados, como pode ser visto na Figura 35.

Figura 35 – Exibição na Página Web

LATITUDE:	LONGITUDE:	VELOCIDADE (km/h):	DATA/HORA:
-10.198725	-48.900667	0.30	2020-08-15 09:55:57
-10.198693	-48.900687	0.24	2020-08-15 09:54:33
-10.198726	-48.900717	0.33	2020-08-15 09:50:55
-10.198726	-48.900717	0.33	2020-08-15 09:50:55
-10.198758	-48.900733	0.67	2020-08-15 09:49:57
-10.198736	-48.900711	0.07	2020-08-15 09:47:14
-10.198737	-48.900708	0.44	2020-08-15 09:46:44
-10.198766	-48.900661	0.07	2020-08-15 09:38:35
-10.198761	-48.900689	0.72	2020-08-15 09:38:09
-10.198697	-48.900698	0.56	2020-08-15 09:36:22

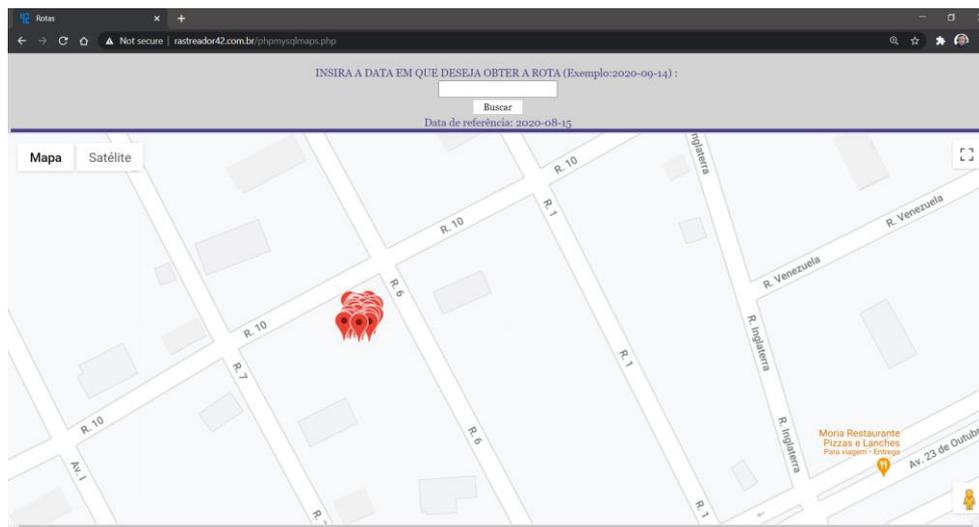
PARA VISUALIZAR A ROTA NO MAPA [CLIQUE AQUI](#)

Versão de teste 2.6
B.S.D
Conectado ao Banco de Dados...

Fonte: Autor

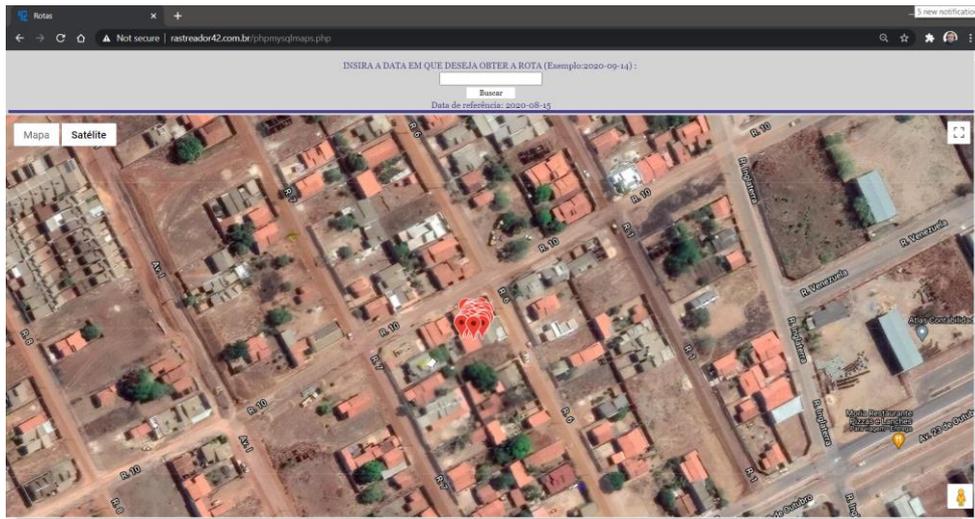
Então, ao clicar em “clique aqui” e inserir data em que o teste foi realizado no campo de busca, todos os pontos correspondentes foram exibidos no mapa, vide Figura 36; os pontos também podem ser visualizados com imagens de satélite (Figura 37). Visto que não houve deslocamento, é possível observar que apesar dos pontos obtidos não serem completamente precisos, a variação é pequena.

Figura 36 – Exibição da Localização no Mapa



Fonte: Autor

Figura 37 – Exibição da Localização Com Imagens de Satélite



Fonte: Autor

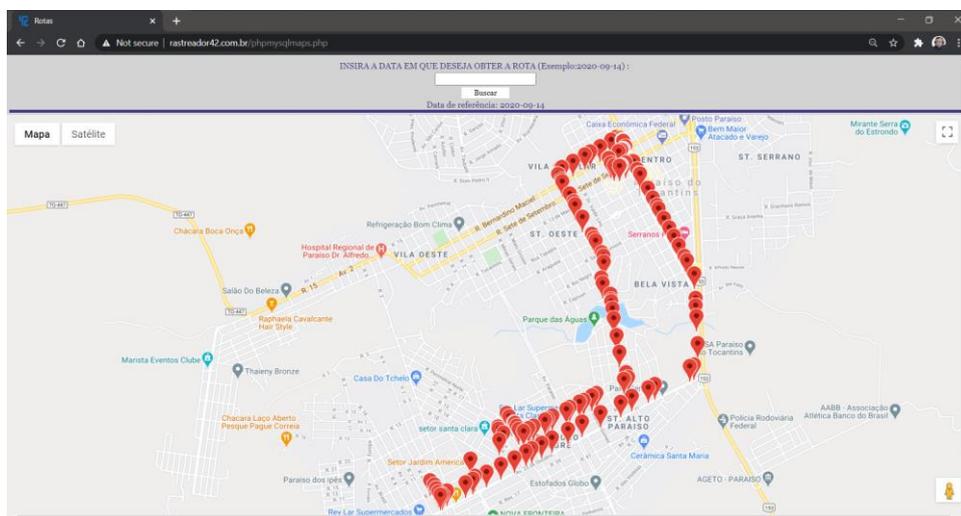
5.2 Teste Com Deslocamento – Conexão Wi-Fi

O teste realizado ocorreu na cidade de Paraíso do Tocantins - TO, Brasil. O protótipo foi inserido em um carro, a fim de registrar o todo trajeto percorrido, com saída do setor Jardim América, percorrendo algumas áreas da cidade, até retornar ao ponto inicial.

O sistema registrou diversos pontos no mapa, de maneira que é possível visualizar todo o caminho percorrido durante o teste, que ocorreu em 14 de setembro de 2020. Deste modo, ao realizar a busca pela data a rota é exibida no mapa (Figura 38).

Neste teste, o sistema foi conectado a uma rede *Wi-Fi* provida de um *smartphone* conectado à rede 4G.

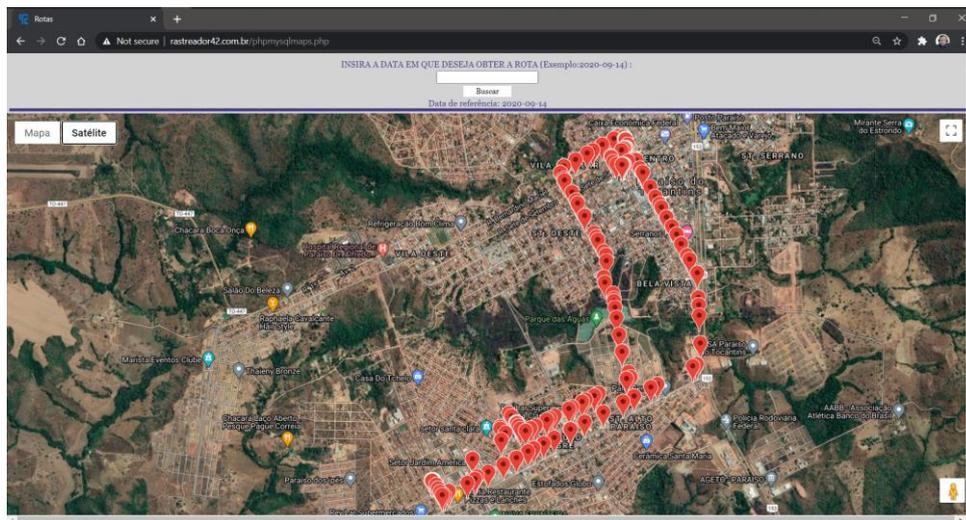
Figura 38 – Exibição da Rota no Mapa – Wi-fi



Fonte: Autor

Na Figura 39 tem-se a rota exibida em imagens de satélite.

Figura 39 – Exibição da Rota Com Imagens de Satélite – *Wi-Fi*

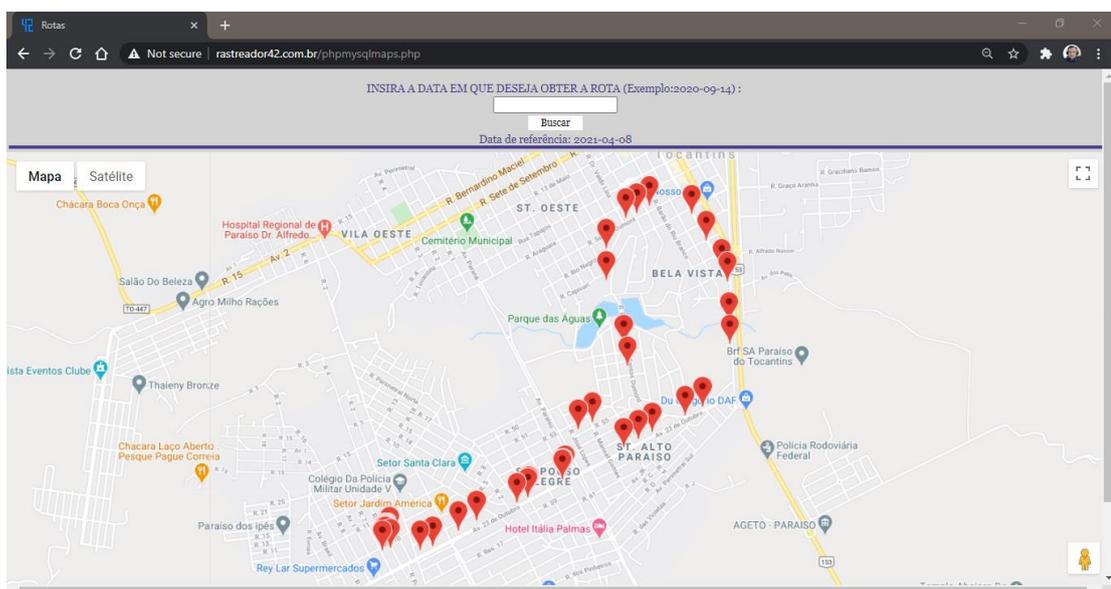


Fonte: Autor

5.3 Teste Com Deslocamento – Conexão GPRS

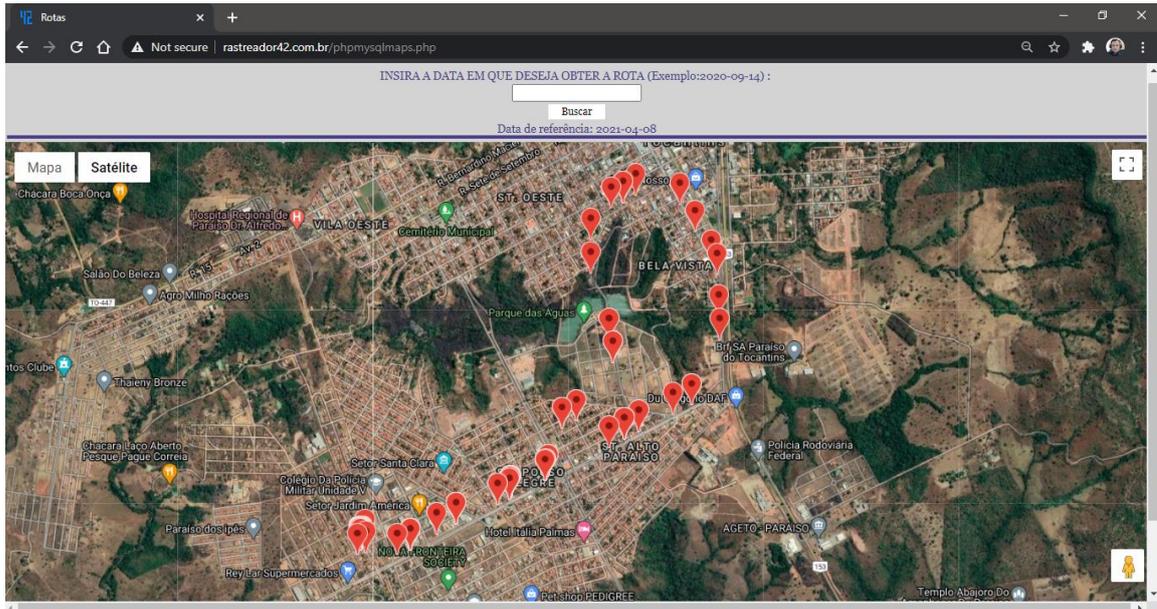
Neste caso, a conexão foi fornecida por um chip de celular conectado ao módulo GPRS. Como o teste foi realizado no dia 8 de abril de 2021, a rota é exibida ao buscar pela data de referência, os resultados podem ser visualizados nas Figuras 40 e 41.

Figura 40 – Exibição da Rota no Mapa – GPRS



Fonte: Autor

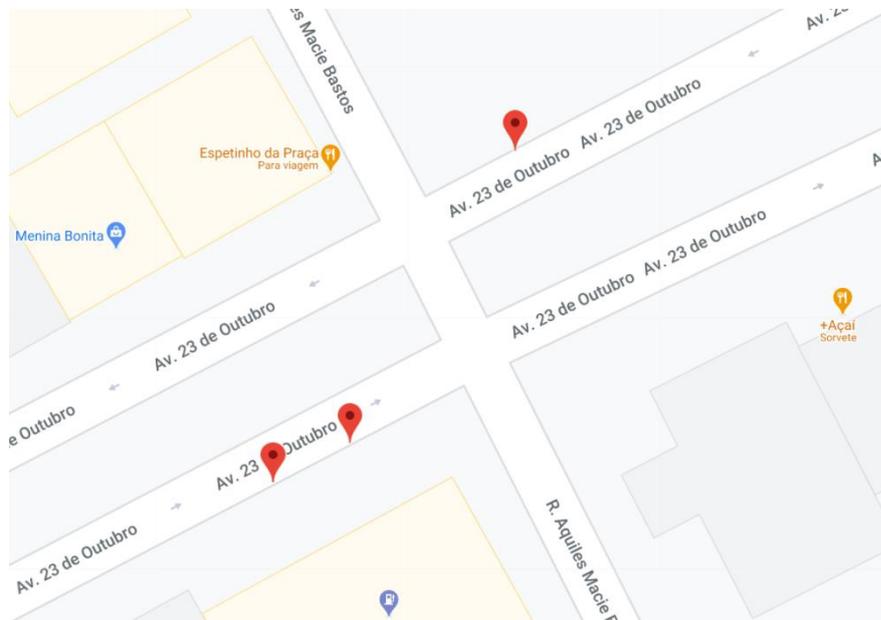
Figura 41 – Exibição da Rota Com Imagens de Satélite – GPRS



Fonte: Autor

Durante o retorno ao ponto de origem, entre os setores Pouso Alegre e Jardim América, foi percorrida a mesma avenida do trajeto de partida. Na Figura 42, tem-se pontos marcados em ambos os sentidos da avenida 23 de Outubro.

Figura 42 – Pontos de Localização na Mesma Avenida - GPRS



Fonte: Autor

5.4 Teste Com Deslocamento – Backup

Para testar o sistema de *backup*, a rota foi percorrida com o sistema operando sem conexão com a internet, deste modo, todos os pontos obtidos foram armazenados no cartão de memória (Figura 43).

Figura 43 – Dados Coletados Durante o Teste de Backup

4.6.2021_22.49.37.0	4.6.2021_22.44.14.0	4.6.2021_22.39.46.0	4.6.2021_22.35.37.0	4.6.2021_22.31.2.0	4.6.2021_22.27.21.0
4.6.2021_22.49.34.0	4.6.2021_22.43.29.0	4.6.2021_22.39.37.0	4.6.2021_22.35.25.0	4.6.2021_22.30.51.0	4.6.2021_22.27.6.0
4.6.2021_22.49.22.0	4.6.2021_22.43.26.0	4.6.2021_22.39.29.0	4.6.2021_22.35.19.0	4.6.2021_22.30.45.0	4.6.2021_22.27.4.0
4.6.2021_22.48.52.0	4.6.2021_22.43.16.0	4.6.2021_22.39.23.0	4.6.2021_22.35.10.0	4.6.2021_22.30.42.0	4.6.2021_22.26.58.0
4.6.2021_22.48.49.0	4.6.2021_22.43.10.0	4.6.2021_22.39.9.0	4.6.2021_22.35.7.0	4.6.2021_22.30.31.0	4.6.2021_22.26.49.0
4.6.2021_22.48.37.0	4.6.2021_22.43.7.0	4.6.2021_22.38.52.0	4.6.2021_22.35.5.0	4.6.2021_22.30.28.0	4.6.2021_22.26.46.0
4.6.2021_22.48.22.0	4.6.2021_22.43.4.0	4.6.2021_22.38.41.0	4.6.2021_22.34.59.0	4.6.2021_22.30.17.0	4.6.2021_22.26.43.0
4.6.2021_22.48.7.0	4.6.2021_22.42.51.0	4.6.2021_22.38.24.0	4.6.2021_22.34.45.0	4.6.2021_22.30.6.0	4.6.2021_22.26.37.0
4.6.2021_22.47.52.0	4.6.2021_22.42.48.0	4.6.2021_22.38.18.0	4.6.2021_22.34.39.0	4.6.2021_22.30.0.0	4.6.2021_22.26.31.0
4.6.2021_22.47.37.0	4.6.2021_22.42.45.0	4.6.2021_22.38.12.0	4.6.2021_22.34.34.0	4.6.2021_22.29.57.0	4.6.2021_22.26.28.0
4.6.2021_22.47.22.0	4.6.2021_22.42.38.0	4.6.2021_22.38.6.0	4.6.2021_22.34.22.0	4.6.2021_22.29.46.0	4.6.2021_22.26.25.0
4.6.2021_22.47.7.0	4.6.2021_22.42.35.0	4.6.2021_22.38.0.0	4.6.2021_22.34.8.0	4.6.2021_22.29.43.0	4.6.2021_22.26.10.0
4.6.2021_22.46.52.0	4.6.2021_22.42.29.0	4.6.2021_22.37.54.0	4.6.2021_22.34.2.0	4.6.2021_22.29.32.0	4.6.2021_22.26.8.0
4.6.2021_22.46.22.0	4.6.2021_22.42.22.0	4.6.2021_22.37.48.0	4.6.2021_22.33.56.0	4.6.2021_22.29.26.0	4.6.2021_22.26.2.0
4.6.2021_22.46.13.0	4.6.2021_22.42.19.0	4.6.2021_22.37.42.0	4.6.2021_22.33.50.0	4.6.2021_22.29.20.0	4.6.2021_22.25.55.0
4.6.2021_22.46.7.0	4.6.2021_22.42.13.0	4.6.2021_22.37.39.0	4.6.2021_22.33.44.0	4.6.2021_22.29.11.0	4.6.2021_22.25.53.0
4.6.2021_22.46.3.0	4.6.2021_22.42.8.0	4.6.2021_22.37.33.0	4.6.2021_22.33.37.0	4.6.2021_22.29.8.0	4.6.2021_22.25.47.0
4.6.2021_22.45.54.0	4.6.2021_22.41.56.0	4.6.2021_22.37.28.0	4.6.2021_22.33.34.0	4.6.2021_22.29.5.0	4.6.2021_22.25.41.0
4.6.2021_22.45.51.0	4.6.2021_22.41.50.0	4.6.2021_22.37.16.0	4.6.2021_22.33.11.0	4.6.2021_22.29.2.0	4.6.2021_22.25.38.0
4.6.2021_22.45.48.0	4.6.2021_22.41.47.0	4.6.2021_22.37.13.0	4.6.2021_22.33.5.0	4.6.2021_22.28.56.0	4.6.2021_22.25.30.0
4.6.2021_22.45.38.0	4.6.2021_22.41.44.0	4.6.2021_22.36.59.0	4.6.2021_22.32.59.0	4.6.2021_22.28.53.0	4.6.2021_22.25.24.0
4.6.2021_22.45.35.0	4.6.2021_22.41.33.0	4.6.2021_22.36.56.0	4.6.2021_22.32.55.0	4.6.2021_22.28.50.0	4.6.2021_22.25.9.0
4.6.2021_22.45.25.0	4.6.2021_22.41.30.0	4.6.2021_22.36.50.0	4.6.2021_22.32.48.0	4.6.2021_22.28.41.0	4.6.2021_22.25.6.0
4.6.2021_22.45.22.0	4.6.2021_22.41.13.0	4.6.2021_22.36.44.0	4.6.2021_22.32.42.0	4.6.2021_22.28.35.0	4.6.2021_22.25.3.0
4.6.2021_22.45.19.0	4.6.2021_22.41.7.0	4.6.2021_22.36.41.0	4.6.2021_22.32.40.0	4.6.2021_22.28.32.0	4.6.2021_22.25.1.0
4.6.2021_22.45.10.0	4.6.2021_22.41.1.0	4.6.2021_22.36.35.0	4.6.2021_22.32.34.0	4.6.2021_22.28.22.0	4.6.2021_22.24.52.0
4.6.2021_22.45.7.0	4.6.2021_22.40.58.0	4.6.2021_22.36.26.0	4.6.2021_22.32.31.0	4.6.2021_22.28.15.0	4.6.2021_22.24.8.0
4.6.2021_22.44.58.0	4.6.2021_22.40.52.0	4.6.2021_22.36.17.0	4.6.2021_22.32.12.0	4.6.2021_22.28.2.0	
4.6.2021_22.44.45.0	4.6.2021_22.40.46.0	4.6.2021_22.36.12.0	4.6.2021_22.31.55.0	4.6.2021_22.27.58.0	
4.6.2021_22.44.39.0	4.6.2021_22.40.43.0	4.6.2021_22.36.9.0	4.6.2021_22.31.47.0	4.6.2021_22.27.50.0	
4.6.2021_22.44.33.0	4.6.2021_22.40.37.0	4.6.2021_22.36.5.0	4.6.2021_22.31.41.0	4.6.2021_22.27.44.0	
4.6.2021_22.44.29.0	4.6.2021_22.40.26.0	4.6.2021_22.35.57.0	4.6.2021_22.31.22.0	4.6.2021_22.27.38.0	
4.6.2021_22.44.23.0	4.6.2021_22.40.8.0	4.6.2021_22.35.43.0	4.6.2021_22.31.16.0	4.6.2021_22.27.32.0	
4.6.2021_22.44.17.0	4.6.2021_22.40.0.0	4.6.2021_22.35.40.0	4.6.2021_22.31.13.0	4.6.2021_22.27.24.0	

Fonte: Autor

Então, uma conexão *Wi-Fi* foi estabelecida, com isso foi possível enviar todos os registros ao servidor. Para acompanhar o envio pelo monitor serial da IDE Arduino, o sistema foi conectado a um computador. É importante observar que o computador só é necessário para visualizar o processo de atualização (Figura 44), não sendo necessário para a operação em si.

Figura 44 – Envio dos Dados de *Backup* ao Servidor

```

COMS

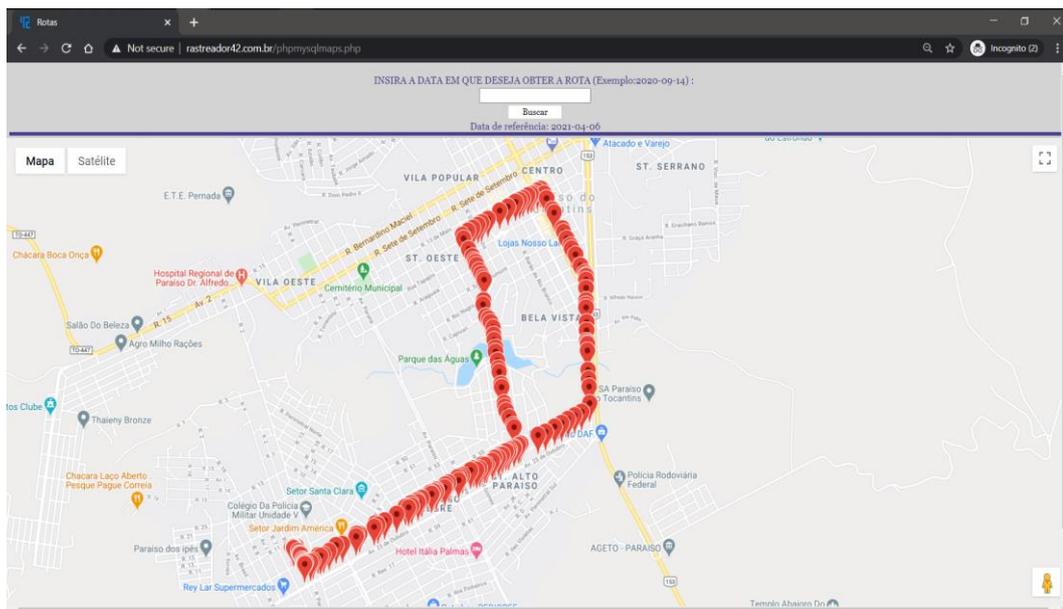
Conectado a uma rede WiFi
httpRequestData: api_key=YjhoOp42L&latitude=-10.198666&longitude=-48.900609&velocidade=0.56&datahora=2021-4-6 19:25:47
HTTP Response code: 200
Apagando arquivo: 05?0
Arquivo deletado
Lendo arquivo: /4.6.2021_22.25.53.0.txt
Conteudo:
api_key=YjhoOp42L&latitude=-10.198676&longitude=-48.900615&velocidade=0.48&datahora=2021-4-6 19:25:53
Arquivo: /4.6.2021_22.25.53.0.txt
Conectado a uma rede WiFi
httpRequestData: api_key=YjhoOp42L&latitude=-10.198676&longitude=-48.900615&velocidade=0.48&datahora=2021-4-6 19:25:53
HTTP Response code: 200
Apagando arquivo: 0?0?0
Arquivo deletado
Lendo arquivo: /4.6.2021_22.25.55.0.txt
Conteudo:
api_key=YjhoOp42L&latitude=-10.198683&longitude=-48.900623&velocidade=2.19&datahora=2021-4-6 19:25:55
Arquivo: /4.6.2021_22.25.55.0.txt
Conectado a uma rede WiFi
httpRequestData: api_key=YjhoOp42L&latitude=-10.198683&longitude=-48.900623&velocidade=2.19&datahora=2021-4-6 19:25:55
HTTP Response code: 200
Apagando arquivo: 0R?0
Arquivo deletado
Lendo arquivo: /4.6.2021_22.26.2.0.txt
Conteudo:
api_key=YjhoOp42L&latitude=-10.198694&longitude=-48.900629&velocidade=0.69&datahora=2021-4-6 19:26:2
Arquivo: /4.6.2021_22.26.2.0.txt
Conectado a uma rede WiFi
httpRequestData: api_key=YjhoOp42L&latitude=-10.198694&longitude=-48.900629&velocidade=0.69&datahora=2021-4-6 19:26:2
HTTP Response code: 200
Apagando arquivo: 0;?0
Arquivo deletado
Lendo arquivo: /4.6.2021_22.26.8.0.txt
Conteudo:
api_key=YjhoOp42L&latitude=-10.198694&longitude=-48.900629&velocidade=0.69&datahora=2021-4-6 19:26:8
Arquivo: /4.6.2021_22.26.8.0.txt
Conectado a uma rede WiFi
httpRequestData: api_key=YjhoOp42L&latitude=-10.198694&longitude=-48.900629&velocidade=0.69&datahora=2021-4-6 19:26:8
HTTP Response code: 200
Apagando arquivo: 0;?0
Arquivo deletado
Lendo arquivo: /4.6.2021_22.26.10.0.txt
Conteudo:
api_key=YjhoOp42L&latitude=-10.198711&longitude=-48.900621&velocidade=0.56&datahora=2021-4-6 19:26:10
Arquivo: /4.6.2021_22.26.10.0.txt
 Autoscroll  Show timestamp

```

Fonte: Autor

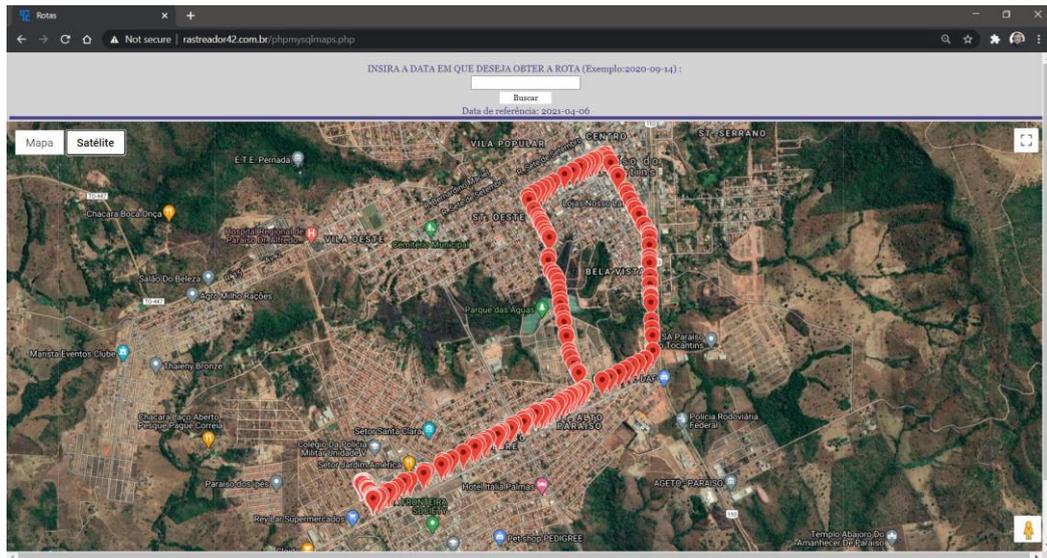
Com o banco de dados atualizado, a rota pode ser visualizada nas Figuras 45 e 46, de maneira análoga aos testes anteriores.

Figura 45 – Exibição da Rota no Mapa – *Backup*



Fonte: Autor

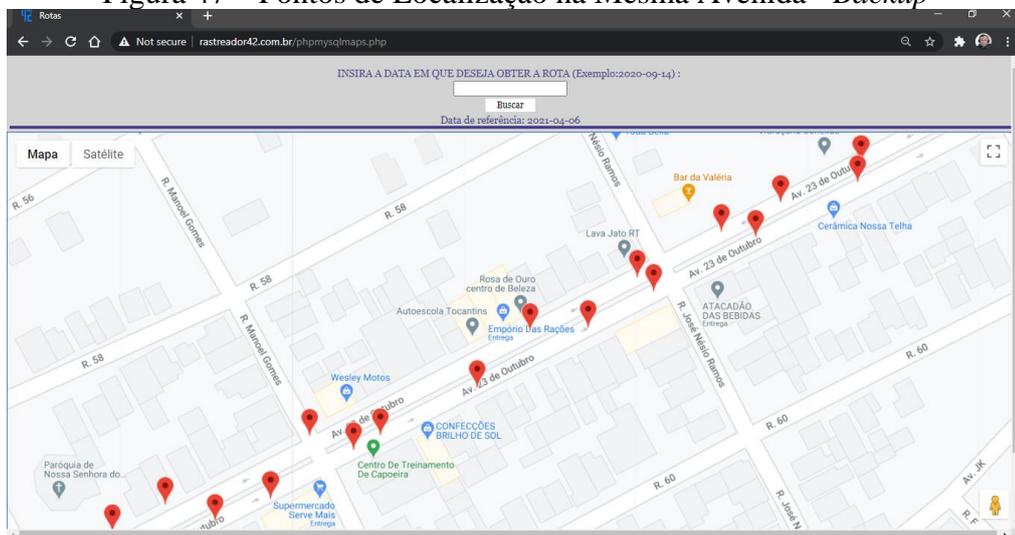
Figura 46 – Exibição da Rota Com Imagens de Satélite – Backup



Fonte: Autor

Assim como ocorreu na rota de teste com o GPRS, neste teste, a avenida 23 de Outubro também foi percorrida na partida e no retorno, os pontos obtidos em ambos os sentidos estão apresentados na Figura 47.

Figura 47 – Pontos de Localização na Mesma Avenida - Backup



Fonte: Autor

6 CONCLUSÃO

O sistema de rastreamento desenvolvido neste trabalho possui três modos de coleta e envio de dados. Sendo possível enviar os pontos de localização diretamente para o servidor por *Wi-Fi* e GPRS, também conta com armazenamento em um cartão de memória. Como o gerenciamento das conexões e *backup* é automático, as chances de perda de dados são reduzidas. Além disso, esta solução é *open-source* e *open-hardware*.

Em todos os testes foi possível registrar a rota percorrida. Entretanto, com a conexão GPRS, a quantidade de pontos coletados foi inferior aos demais. Isto ocorre devido ao tempo necessário para que a conexão seja estabelecida e os dados enviados para o servidor através do módulo SIM800L. Nos testes com conexão *Wi-Fi* e armazenamento no cartão de memória, a densidades de pontos de localização foi similar; em ambos os casos, a rota ficou clara, com uma grande quantidade de marcações no mapa.

Como a diferença de densidade dos pontos se dá devido ao atraso da conexão GPRS e os testes foram realizados em um carro. Para aplicações em baixas velocidades, poderá se obter uma boa densidade de pontos com este método.

O sistema web também se mostrou funcionou, considerando que foi possível obter todo o trajeto realizado diretamente no mapa e em imagens de satélite em todos os testes.

6.1 Trabalhos Futuros

A partir dos resultados obtidos neste trabalho pode-se desenvolver alguns trabalhos futuros, como:

- Otimização da comunicação com o servidor através da conexão GPRS, para que se obtenha uma maior densidade de pontos de localização com esta tecnologia.
- Desenvolvimento de mais opções de filtragem de rotas exibidas no mapa, incluindo buscas por horário, semana, mês e períodos específicos.
- Substituição da integração com o *Google Maps* por um sistema de mapas gratuito.

REFERÊNCIAS

- ASSIS, PAULO UEINER MOREIRA DE. **SISTEMA DE RASTREAMENTO DE VEÍCULOS PARA EMPRESAS DE TRANSPORTE UTILIZANDO NAVEGAÇÃO POR SATÉLITE**. 2010. Monografia (Graduação em Engenharia de Computação.) - Centro Universitário de Brasília, [S. l.], 2010
- ESPRESSIF SYSTEMS (Xangai). **ESP32 Series Datasheet**. [S. l.: s. n.], 2019
- GONÇALVES, Gabriel Ananias; TAVARAYAMA, Rodrigo. ESTUDO DA VIABILIDADE DO USO DE GPS PARA O MONITORAMENTO DE VEICULOS DE PASSEIO NA REGIÃO DE ORLÂNDIA-SP. **Nucleus**, [s. l.], p. 297-316, 15 set. 2011.
- LUTHFi, Achmad M.; KARNA, Nyoman; MAYASARI, Ratna. **Google Maps API Implementation On IOT Platform For Tracking an Object Using GPS**. *2019 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, BALI, Indonesia, 2019, pp. 126-131, doi: 10.1109/APWiMob48441.2019.896413
- MAGALHÃES, Caroline Tristão de Alencar. EXPERIÊNCIAS INTERNACIONAIS E NACIONAIS DE MONITORAMENTO DO TRANSPORTE PÚBLICO URBANO POR GPS. *In: MAGALHÃES, Caroline Tristão de Alencar. AVALIAÇÃO DE TECNOLOGIAS DE RASTREAMENTO POR GPS PARA MONITORAMENTO DO TRANSPORTE PÚBLICO POR ÔNIBUS*. 2008. DISSERTAÇÃO (Mestrado em Engenharia de Transportes) - UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, [S. l.], 2008
- ORACLE. MySQL. What is MySQL?. *In: MySQL 8.0 Reference Manual*. [S. l.], 29 set. 2020. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Acesso em: 29 set. 2020.
- PAZ, Sérgio M.; FERREIRA, Waldir; CUGNASCA, Carlos E. Sistema de Posicionamento Global (GPS) e o Turismo. **Turismo em Análise**, [s. l.], p. 25-46, 9 set. 1998.
- SALOMONI, Christiane Silva. **GPS e Ionosfera: Estudo do comportamento do TEC e de sua influência no posicionamento com GPS na região brasileira em períodos de alta e baixa atividade solar**. 2008. DISSERTAÇÃO (Mestrado em Sensoriamento Remoto) - UFRGS, [S. l.], 2008.

SALOMONI, Christiane Silva; MATSUOKA, Marcelo Tomio; SOUZA, Sérgio Florêncio de. EFEITOS DE TEMPESTADES GEOMAGNÉTICAS NO POSICIONAMENTO RELATIVO COM GPS NA REGIÃO BRASILEIRA: ESTUDO DE CASO PARA 29 DE OUTUBRO DE 2003. **Revista Brasileira de Cartografia**, [s. l.], 16 out. 2011.

U-BLOX (Thalwil). **NEO-6 u-blox 6 GPS Modules**: Data Sheet. [S. l.: s. n.], 2011

APÊNDICE A – Código Sistema De Rastreamento

Neste apêndice está o código desenvolvido para o sistema de rastreamento. Onde ocorre a coleta de dados GPS, gerenciamento das conexões de internet, envio dos dados para o servidor, leitura e controle de carga da bateria e também o gerenciamento dos arquivos de *backup*.

```

1 #include <TinyGPS++.h>
2 #include <Wire.h>
3 #include <WiFi.h>
4 #include <HTTPClient.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_SSD1306.h>
7 #include "FS.h"
8 #include "SD.h"
9 #include "SPI.h"
10 #define SCREEN_WIDTH 128
11 #define SCREEN_HEIGHT 64
12 #define SD_CS 5
13
14 const char* ssid      = "";
15 const char* senha     = "*****";
16 String servidor      = "http://rastreador42.com.br/paginawebenvia.php";
17 String apiKeyValor   = "APIsenha";
18 const char CONTENT_TYPE[] = "application/x-www-form-urlencoded";
19 String apn           = "claro.com.br";
20 String apn_u         = "claro";
21 String apn_p         = "claro";
22 String initsim       = "";
23 String loc           = "";
24 String dathora       = "";
25 String httpRequestData; // Armazena os dados para enviar ao servidor
26 uint8_t i;
27 uint8_t redeAtiva; // Esta variável indica o estado da conexão
28 uint8_t bat;
29 int controlebat = 32;
30
31 HardwareSerial gps_serial(1);
32 HardwareSerial sim_serial(2);
33
34 TinyGPSPlus gps;
35
36 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
37
38 void setup(){
39   gps_serial.begin(9600, SERIAL_8N1, 15, 4);
40   sim_serial.begin(115200, SERIAL_8N1, 16, 17);
41   Serial.begin(9600);
42   SD.begin(SD_CS);
43   if(!SD.begin(SD_CS)){
44     Serial.println("Falha na leitura do cartão");
45     return;
46   }else{
47     Serial.println("Iniciando o cartão de memória");
48   }
49   if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
50     Serial.println(F("falha no diplay"));
51     for(;;);
52   }
53
54   pinMode(32,OUTPUT);

```

```

55  pinMode(33, INPUT);
56  wifiinit();
57  }
58  void loop () {
59
60  DADOS_GPS();
61
62
63  }
64  void DADOS_GPS      () {
65  while (gps_serial.available() > 0){
66    if (gps.encode(gps_serial.read())){
67      Serial.print(F("Localizacao: "));
68  if (gps.location.isValid())
69    {
70
71    Serial.print(gps.location.lat(), 6);
72    Serial.print(F(", "));
73    Serial.print(gps.location.lng(), 6);
74  config_display();
75  display.print("Latitude :");
76  display.println(gps.location.lat(), 6);
77  display.println(" ");
78  display.print("Longitude:");
79  display.println(gps.location.lng(), 6);
80  display.println(" ");
81  display.display();
82  Serial.print(F(" "));
83
84  }
85  else
86  {
87  Serial.print(F("invalido"));
88  config_display();
89  display.println("Conectando...");
90  display.println("");
91  display.println("Loc. indisponivel!");
92  display.display();
93  }
94
95  Serial.print("Velocidade:");
96  if(gps.speed.isValid()){
97  Serial.print(F(", "));
98  Serial.print(gps.speed.kmph());
99  display.print("Vel:");
100 display.print(gps.speed.kmph());
101 display.print("km/h");
102 display.display();
103
104 }
105
106 Serial.print(F(" Data/Hora: "));
107 if (gps.date.isValid())
108 {
109   Serial.print(gps.date.month());
110   Serial.print(F("/"));
111   Serial.print(gps.date.day());
112   Serial.print(F("/"));
113   Serial.print(gps.date.year());
114 }
115 }
116 else

```

```

117 {
118     Serial.print(F("invalido"));
119 }
120
121     Serial.print(F(" "));
122     if (gps.time.isValid())
123     {
124         if (gps.time.hour() < 10) Serial.print(F("0"));
125         Serial.print(gps.time.hour());
126         Serial.print(F(":"));
127         if (gps.time.minute() < 10) Serial.print(F("0"));
128         Serial.print(gps.time.minute());
129         Serial.print(F(":"));
130         if (gps.time.second() < 10) Serial.print(F("0"));
131         Serial.print(gps.time.second());
132         Serial.print(F("."));
133         if (gps.time.centisecond() < 10) Serial.print(F("0"));
134         Serial.print(gps.time.centisecond());
135     }
136     else
137     {
138         Serial.print(F("Invalido"));
139     }
140 }
141 Serial.println();
142     datahora();
143     if (gps.location.isValid())
144     {
145
146
147         datahora();
148         delay(200);
149         httpRequestData = "api_key=" + apiKeyValor + "&latitude=" + String(gps.location.lat(),
150 6)
151         + "&longitude=" + String(gps.location.lng(), 6) + "&velocidade=" +
152         String(gps.speed.kmph()) + "&datahora="+ dathora+"";
153     verificaBackup(SD, "/"); // Verifica se há arquivos de backup antes de enviar os dados mais recentes
154     //sdcard();// Descomente para gravar os dados no cartão cd de maneira direta
155     //enviarDados(); // Descomente para enviar os dados diretamente (sem verificar backup
156     }
157     else
158     {
159         Serial.println(F("Ainda não conectado ao GPS:DADOS INVÁLIDOS!"));
160         delay(10000);
161     }
162 }
163     bateria();
164 }
165     if (millis() > 5000 && gps.charsProcessed() < 10)
166     {
167
168         Serial.println(F("GPS desconectado."));
169         bateria();
170         config_display();
171         display.println("GPS Desativado!");
172         display.display();
173         while(true);
174     }
175 }
176     void resposta      () {
177         while (sim_serial.available()) {

```

```

178     Serial.write(sim_serial.read());
179 }
180 }
181 void datahora      () {
182     /* Como o horário do GPS é GMT 0
183     Esta função tem como objetivo converter para o
184     horário de Brasília
185     */
186     int hora;
187     hora = int(gps.time.hour()) - 3;
188     if(int(gps.time.hour()) == 02){
189         hora = 23;
190     }
191     if(int(gps.time.hour()) == 01){
192         hora = 22;
193     }
194     if(int(gps.time.hour()) == 00){
195         hora = 21;
196     }
197     dathora = String(gps.date.year()+"-"+String(gps.date.month())
198     +"-"+String(gps.date.day())+" "+String(hora)
199     +":"+String(gps.time.minute())+":"+String(gps.time.second())+"";
200     // Serial.println("Data e Hora: "+dathora);
201 }
202 void enviarDados   () {
203     redeAtiva = 0;
204     if(WiFi.status() == WL_CONNECTED){
205         COM_WIFI();
206     }else{
207         COM_GPRS();
208         wifininit();
209     }
210 }
211 void COM_GPRS      () {
212     sim_serial.println("AT"); // TESTE (RESPOSTA DEVE SER "OK")
213     delay(1000);
214     VerificaModulo();
215     sim_serial.println("AT+SAPBR=3,1,Contype,GPRS");// iniciando modo GPRS
216     resposta();
217     delay(100);
218     sim_serial.println("AT+SAPBR=3,1,APN," + apn); //apn
219     resposta();
220     delay(100);
221     sim_serial.println("AT+SAPBR=3,1,USER," + apn_u); //usuário
222     resposta();
223     delay(100);
224     sim_serial.println("AT+SAPBR=3,1,PWD," + apn_p); //senha
225     resposta();
226     delay(1000);
227     sim_serial.println("AT+SAPBR =1,1"); // habilitar GPRS
228     delay(2000);
229     VerificaRede();
230
231     if (redeAtiva == 1 ){
232         sim_serial.println("AT+SAPBR=2,1"); // Exibe o endereço IP do módulo
233         resposta();
234         delay(2000);
235         sim_serial.println("AT+HTTPIPINIT"); // Iniciando serviço HTTP
236         resposta();
237         delay(1000);
238         sim_serial.println("AT+HTTTPARA=CID,1");
239         resposta();

```

```

240 delay(100);
241 sim_serial.println("AT+HTTTPARA=URL," + servidor);
242 resposta();
243 delay(100);
244 sim_serial.println("AT+HTTTPARA=CONTENT,application/x-www-form-urlencoded");
245 resposta();
246 delay(100);
247 sim_serial.println("AT+HTTTPDATA=192,1000");
248 resposta();
249 delay(100);
250 sim_serial.println(httpRequestData);
251 resposta();
252 Serial.println ( httpRequestData );
253 delay(1000);
254 sim_serial.println("AT+HTTTPACTION=1");
255 delay(2000);
256 resposta();
257 sim_serial.println("AT+HTTTPREAD");
258 resposta();
259 delay(100);
260 sim_serial.println("AT+HTTPTERM");
261 delay(1000);
262 resposta();
263
264
265 sim_serial.println("AT+SAPBR =0,1"); // desabilitar GPRS
266 delay(1000);
267 resposta();
268 }
269 //Fim da Configuração da rede Conexão GPRS
270 }
271 void COM_WIFI      () {
272   if(WiFi.status()== WL_CONNECTED){
273     redeAtiva = 1;
274     Serial.println("Conectado a uma rede WiFi");
275     HTTPClient http;
276
277     http.begin(servidor);
278
279
280     http.addHeader("Content-Type", "application/x-www-form-urlencoded", "Method","POST");
281
282     Serial.print("httpRequestData: ");
283     Serial.println(httpRequestData);
284
285     int httpResponseCode = http.POST(httpRequestData);
286
287     if (httpResponseCode>0) {
288       Serial.print("HTTP Response code: ");
289       Serial.println(httpResponseCode);
290     }
291     else {
292       Serial.print("Error code: ");
293       Serial.println(httpResponseCode);
294     }
295
296     http.end();
297     delay(5000);
298   }
299   else {
300     Serial.println("WiFi Desconectado");
301   }

```

```

302
303
304 // Fim da configuração WIFI
305
306 }
307 void wifiinit      () {
308
309   WiFi.begin(ssid, senha);
310   Serial.println("Conectando...");
311   delay(800);
312   if(WiFi.status() != WL_CONNECTED) {
313     Serial.println("WiFi não conectado");
314   }else{
315     Serial.println("");
316     Serial.println("Conectado a rede WiFi, com o seguinte endereço IP: ");
317     Serial.println(WiFi.localIP());
318   }
319 }
320 void VerificaModulo() {
321   initsim = "";
322   if(sim_serial.available()){
323     while (sim_serial.available()) {
324       initsim = sim_serial.readStringUntil('\n');
325       Serial.println("Inicio: "+initsim);
326     }
327
328   }else{
329     Serial.println("Falha ao iniciar o módulo, gravando no Cartão de Memória");
330     initsim = "falha";
331     // sdcard();
332   }
333 }
334 void VerificaRede  () {
335   String res = "";
336   while (sim_serial.available()) {
337     res = sim_serial.readStringUntil('\n');
338     // Serial.println("Resposta por linha: "+res);
339   }
340   if (res == initsim){
341     Serial.println("Conectado a uma rede: "+res);
342     redeAtiva = 1;
343   }else{
344     Serial.println("Não conectado a uma rede, gravando no cartão de memória.");
345     sdcard();
346   }
347   Serial.println("Resposta Final: "+res);
348 }
349 void sdcard        () {
350
351   if((String(gps.location.lat(), 6)) != "0.000000" ){
352     if (gps.date.isValid()){
353       if (gps.time.isValid()){
354         loc = String(gps.date.month())+"."+String(gps.date.day())+"."
355         +String(gps.date.year())+"_"+String(gps.time.hour())
356         +String(gps.time.minute())+"."+String(gps.time.second())+"."
357         +String(gps.time.centisecond())+"";
358         Serial.print("Novo arquivo:");
359         Serial.println(loc);
360       }
361     }else{
362       Serial.print("ERRO ao criar nome para novo arquivo");
363     }

```

```

364 File file = SD.open("/"+String(loc)+".txt");
365 if(!file){
366
367     Serial.println("Arquivo inexistente");
368     Serial.println("Criando Arquivo");
369     datahora();
370     httpRequestData = "api_key=" + apiKeyValor +
371     "&latitude=" + String(gps.location.lat(), 6) +
372     "&longitude=" + String(gps.location.lng(), 6) +
373     "&velocidade=" + String(gps.speed.kmph()) +
374     "&datahora="+ dathora+"";
375
376     writeFile(SD,"/"+String(loc)+".txt",httpRequestData);
377
378     }else{
379         Serial.println("O arquivo já existe");
380     }
381     file.close();
382
383 delay(500);
384 }else{
385     Serial.println("Dados Inválidos, NÃO GRAVADO.");
386 }
387 }
388 void bateria      () {
389     int nivelbat;
390     bat = map(analogRead(33), 0, 4095, 0,100);
391 // Os níveis de bateria e seu controle de carga.
392
393 if((bat>=0) && (bat<55)){ //leitura de 55 = bateria com aproximadamente 5V
394     nivelbat = 0;
395     digitalWrite(controlebat,HIGH);
396 }
397 if((bat>54) && (bat<65)){ //leitura de 65 = bateria com aproximadamente 6V
398     nivelbat = 1;
399     digitalWrite(controlebat,HIGH);
400 }
401 if((bat>64) && (bat<81)){ //leitura de 81 = bateria com aproximadamente 7V
402     nivelbat = 2;
403     digitalWrite(controlebat,HIGH);
404 }
405 if((bat>80)&& ((bat<97))){ //leitura de 100 = bateria com aproximadamente 8V
406     nivelbat = 3;
407     digitalWrite(controlebat,HIGH);
408 }
409 if (bat>96){
410     digitalWrite(controlebat,LOW);
411 }
412 }}
413 void config_display() {
414
415     display.clearDisplay();
416     display.setTextSize(0.8);
417     display.setTextColor(WHITE);
418     display.setCursor(1, 0);
419
420 // Monitoramento da bateria-----
421 // O monitoramento da bateria foi colocado dentro da função do display
422     bat = map(analogRead(33), 0, 4095, 0,100);
423     display.print("Bateria: ");
424     display.println(String(bat)+"%");
425     display.println("Coordenadas:");

```

```

426   display.display();
427   display.setTextSize(1);
428   // -----
429   }
430   void lerarquivo      (fs::FS &fs, const char * path)          {
431       Serial.printf("Lendo arquivo: %s\n", path);
432
433       File file = fs.open(path);
434       if(!file){
435           Serial.println("Falha na abertura para leitura");
436           return;
437       }
438
439       String c = "";
440       while (file.available())
441       {
442           c += (char)file.read();
443       }
444       httpRequestData = c;
445
446       Serial.println("Conteudo:");
447       Serial.println(c);
448       file.close();
449   }
450   void lersd           (fs::FS &fs, const char*diretorio)      {
451       File root = fs.open(diretorio);
452       if(!root){
453           Serial.println("Falha ao abrir o diretório");
454           return;
455       }
456       if(!root.isDirectory()){
457           Serial.println("Não e um diretório");
458           return;
459       }
460       File file = root.openNextFile();
461       while(file){
462           lerarquivo(SD, file.name());
463           if(file.isDirectory()){
464               Serial.print("  DIR : ");
465               Serial.println(file.name());
466           } else {
467               Serial.print("Arquivo: ");
468               Serial.println(file.name());
469           }
470           enviarDados();
471           if(redeAtiva == 1){
472               // se a rede estiver ativa, deletar arquivo já enviado ao servidor
473               deleteFile(SD, file.name()); // IMPORTANTE: CUIDADO AQUI, APAGAR ARQUIVOS
474               file = root.openNextFile();
475           }else{
476               Serial.println("Falha ao deletar; rede indisponível");
477               break;
478           }
479       }
480       file.close();
481   }
482   void writeFile       (fs::FS &fs, String path, String message) {
483       //Escrevendo/gravando no cartão SD
484       Serial.printf("Escrevendo no arquivo: %s\n", path);
485       File file = fs.open(path, FILE_WRITE);
486       if(!file) {
487           Serial.println("Falha ao abrir o arquivo para escrita");

```

```
488     return;
489 }
490 if(file.print(message)) {
491     Serial.println("Arquivo atualizado");
492 } else {
493     Serial.println("Erro na escrita.");
494 }
495 file.close();
496 }
497 void deleteFile    (fs::FS &fs, String path)           {
498     Serial.printf("Deleting file: %s\n", path);
499     if(fs.remove(path)){
500         Serial.println("File deleted");
501     } else {
502         Serial.println("Delete failed");
503     }
504 }
505 void verificaBackup(fs::FS &fs, const char*diretorio) {
506     File root = fs.open(diretorio);
507     File file = root.openNextFile();
508     int i = 0;
509     while(file){
510         i++;
511         file = root.openNextFile();
512         if(i>3){
513             break;
514         }
515     }
516     if(i>1){
517         Serial.println("Possui Backup!");
518         lersd(SD, "/");
519     }else{
520         Serial.println("Não Possui Backup!");
521         enviarDados();
522     }
523     file.close();
```

APÊNDICE B – Código em PHP Inserção no Banco de Dados.

Antes dos dados serem gravados no servidor, são recebidos por uma página PHP, que os separa em variáveis e os envia para o banco de dados; o código desenvolvido para este caso está disponível neste apêndice.

```

1 <?php
2 require("conexBD.php");
3 $conn = mysqli_connect($servername, $username, $password, $database);
4 if (!$conn) {
5     die("Falha na conexão: " . mysqli_connect_error());
6 }
7 echo "Banco de Dados Conectado<br><br>";
8
9
10     $api_key_valor = "APIsenha";
11
12 $api_key = $latitude = $longitude = $velocidade = $datahora = "";
13
14
15 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
16     $api_key = test_input($_POST["api_key"]);
17     if($api_key == $api_key_valor) {
18
19         $latitude = test_input($_POST["latitude" ]);
20         $longitude = test_input($_POST["longitude" ]);
21         $velocidade = test_input($_POST["velocidade"]);
22         $datahora = test_input($_POST["datahora" ]);
23
24         if ($conn->connect_error) {
25             die("Connection failed: " . $conn->connect_error);
26         }
27 $sql = "INSERT INTO `dados` (`Latitude`, `Longitude`, `Velocidade`, `Data_Hora`)
28         VALUES ('$latitude', '$longitude', '$velocidade', '$datahora')";
29
30         if ($conn->query($sql) === TRUE) {
31             echo "Gravado com sucesso";
32         }
33         else {
34             echo "Error: " . $sql . "<br>" . $conn->error;
35         }
36         $conn->close();
37     }
38     else {
39         echo "Key de verificação incorreta... ";
40     } }
41 else {
42     echo "Não houve inserção de dados via HTTP POST.";}
43
44 function test_input($data) {
45     $data = trim($data);
46     $data = stripslashes($data);
47     $data = htmlspecialchars($data);
48     return $data; }
49 ?>

```

APÊNDICE C – Código da Página Inicial.

A página inicial do sistema *web* exibe os dez últimos registros. Este código é responsável por estruturar a página e realizar a busca no banco de dados.

```

1 <!DOCTYPE html>
2 <html>
3 <?php
4 require("conexBD.php");
5 $conn = mysqli_connect($servername, $username, $password, $database);
6 $mysqli = new mysqli($servername, $username, $password, $database);
7 $consulta = "SELECT * FROM dados order by id desc limit 10 ";
8 $cons = $mysqli->query($consulta) or die ($mysqli->error);
9 ?>
10 <head>
11 <title>Rastreador 42</title>
12 <link rel="stylesheet" type="text/css" href="estilo.css"/>
13 <meta http-equiv="content-Type" content="text/html; charset=UTF-8">
14 </head>
15 <body style="background-color:#708090;">
16 <body>
17 <div bgcolor="black" id="geral">
18 <div id="cab">
19 <br/>
20 <br/>
21 Localização GPS (Últimos Dados Obtidos)
22 </div>
23 <hr color="#483D8B" size="1">
24 <div id="centro">
25 <table id="tabela">
26 <tr id="linha1">
27 <td>LATITUDE:</td>
28 <td>LONGITUDE:</td>
29 <td>VELOCIDADE<br/>(km/h) :</td>
30 <td>DATA/HORA:</td>
31 <td>LINK:</td>
32 </tr>
33 <?php
34 while($dadosgps = $cons->fetch_array()){
35 ?>
36 <tr>
37 <td>
38 <?php
39 echo $dadosgps["Latitude"];
40 ?></td>
41 <td>
42 <?php
43 echo $dadosgps["Longitude"];
44 ?></td>
45 <td>
46 <?php
47 echo $dadosgps["Velocidade"];
48 ?></td>
49 <td>
50 <?php
51 echo $dadosgps["Data_Hora"];
52 ?>
53 </td>
54 </tr>
55
56 <?php

```

```
57 }
58 ?>
59 <br/>
60 </table>
61
62 </div>
63 <br/>
64 <hr/ color="#483D8B" size="1">
65     <br/>
66     <form id="formulario" action="phpmysqlmaps.php" method="post">
67     PARA VISUALIZAR A ROTA NO MAPA
68     <input id="Button2" type="submit" value="CLIQUE AQUI"> <br/>
69
70     <br/>
71     </form>
72     <hr/ color="#483D8B" size="1">
73         <pre>
74 Versão de teste 2.6
75 B.S.D
76                                     </pre>
77     </body>
78     <?php
79 header("Refresh: 3");
80 if (!$conn) {
81     die("Falha na Conexão.: " . mysqli_connect_error());
82 }
83 echo "Conectado ao Banco de Dados...";
84 ?>
85 <br/>
86 <br/>
87
88 </html>
```

APÊNDICE D – Criação de Arquivo XML a Partir do Banco de Dados.

A API do *Google maps* exibe os pontos no mapa a partir de um arquivo XML, o código disponível neste apêndice é o responsável por gerar este arquivo com os dados de localização correspondentes a busca realizada.

```

1 <?php
2 require("conexBD.php");
3 function parseToXML($htmlStr)
4 {
5 $xmlStr=str_replace('<','&lt;',$htmlStr);
6 $xmlStr=str_replace('>','&gt;',$xmlStr);
7 $xmlStr=str_replace('"','&quot;',$xmlStr);
8 $xmlStr=str_replace("'",'&#39;',$xmlStr);
9 $xmlStr=str_replace("&","&amp;",$xmlStr);
10 return $xmlStr;
11 $connection=mysqli_connect ($servername, $username, $password);
12 if (!$connection) {
13 die('Não conectado: ' . mysqli_error());}
14 $db_selected = mysqli_select_db($connection, $database);
15 if (!$db_selected) {
16 die ('Can\'t use db : ' . mysqli_error());
17 }
18
19 $mysqli = new mysqli($servername, $username, $password, $database);
20 $dat1 = "SELECT * FROM `buscas` order by id desc limit 1";
21 $cons = $mysqli->query($dat1) or die ($mysqli->error);
22 while($date = $cons->fetch_array()){
23 $dat = $date["dat"];
24
25 }
26 $query ="SELECT * FROM `dados` WHERE Data_Hora >='$dat 00:00:00' AND Data_Hora <= '$dat 23:59:59'";
27 $result = mysqli_query($connection, $query);
28 if (!$result) {
29 die('Invalid query: ' . mysqli_error());
30 }
31
32 header("Content-type: text/xml");
33 echo "<?xml version='1.0' ?>";
34 echo '<markers>';
35 $ind=0;
36 while ($row = @mysqli_fetch_assoc($result)){
37 echo '<marker ' ;
38 echo 'lat="' . $row['Latitude'] . ' ' ;
39 echo 'lng="' . $row['Longitude'] . ' ' ;
40 echo 'data="' . $row['Data_Hora'] . ' ' ;
41 echo '>';
42 $ind = $ind + 1;
43 }
44
45 echo '</markers>';
46
47 ?>

```

APÊNDICE E – Código Para Página com Integração com o *Google Maps*.

Este apêndice contém o código utilizado no desenvolvimento da página *web* com integração com o *Google Maps*, onde é possível exibir as rotas por data.

```

1 <!DOCTYPE html >
2 <head>
3 <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
4 <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
5 <title>Rotas</title>
6 <link rel="stylesheet" type="text/css" href="estilo.css"/>
7 <style>
8 #map {
9 height: 100%;
10 }
11 html, body {
12 height: 100%;
13 margin: 0;
14 padding: 0;
15 }
16 </style>
17 </head>
18 <html>
19 <body>
20 <div id="form">
21 <br/>
22 INSIRA A DATA EM QUE DESEJA OBTER A ROTA (Exemplo:2020-09-14) :
23 <form id="formulario" action="phpmysqlmaps.php" method="post">
24 <input type="text" name="dat"/><br/>
25 <input id="Button" type="submit" value="Buscar"> <br/>
26 <?php
27 require("conexBD.php");
28 $conn = mysqli_connect($servername, $username, $password, $database);
29
30 if(isset($_POST['dat'])) {
31 $dat = $_POST['dat'];
32 echo "Data de referência: ";
33 echo $dat;
34 $sql ="INSERT INTO `buscas` (`dat`) VALUES ('$dat')";
35 if (mysqli_query($conn, $sql)) {
36
37 }
38 else {
39 echo "Erro:<br> " . $sql . "<br>" . mysqli_error($conn);
40 }
41 }else{
42 echo "Insira a data";
43 }
44 ?>
45 <hr/ color="#483D8B" size="1">
46 </form>
47
48 </div>
49 <div id="map"></div>
50
51 <script>
52 var customLabel = {
53 }
54
55 function initMap() {
56 var map = new google.maps.Map(document.getElementById('map'), {

```

```

57         center: new google.maps.LatLng(-10.198828 , -48.900661 ),
58         zoom: 13
59     });
60     var infoWindow = new google.maps.InfoWindow;
61
62     downloadUrl('result.php', function(data) {
63         var xml = data.responseXML;
64         var markers = xml.documentElement.getElementsByTagName('marker');
65         Array.prototype.forEach.call(markers, function(markerElem) {
66             var id = markerElem.getAttribute('id');
67             var address = markerElem.getAttribute('address');
68             var type = markerElem.getAttribute('type');
69             var point = new google.maps.LatLng(
70                 parseFloat(markerElem.getAttribute('lat')),
71                 parseFloat(markerElem.getAttribute('lng')));
72
73             var infowincontent = document.createElement('div');
74             var strong = document.createElement('strong');
75             strong.textContent = name
76             infowincontent.appendChild(strong);
77             infowincontent.appendChild(document.createElement('br'));
78
79             var text = document.createElement('text');
80             text.textContent = address
81             infowincontent.appendChild(text);
82             var icon = customLabel[type] || {};
83             var marker = new google.maps.Marker({
84                 map: map,
85                 position: point,
86                 label: icon.label
87             });
88             marker.addListener('click', function() {
89                 infoWindow.setContent(infowincontent);
90                 infoWindow.open(map, marker);
91             });
92         });
93     });
94 }
95 function downloadUrl(url, callback) {
96     var request = window.ActiveXObject ?
97         new ActiveXObject('Microsoft.XMLHTTP') :
98         new XMLHttpRequest;
99     request.onreadystatechange = function() {
100         if (request.readyState == 4) {
101             request.onreadystatechange = doNothing;
102             callback(request, request.status);
103         }
104     };
105     request.open('GET', url, true);
106     request.send(null);
107 }
108 function doNothing() {}
109 </script>
110 <script defer
111 src="https://maps.googleapis.com/maps/api/js?key=*****wyffnVaQUx8hBDeE5FxE&callback=initMap">
112 </script>
113 </body>
114 </html>

```